`SpHMC`: Spectral Hamiltonian Monte Carlo

Haoyi Xiong[†*], Kafeng Wang[§*], Jiang Bian[$*], Zhanxing Zhu[‡], Cheng-Zhong Xu[§], Zhishan Guo[$], Jun Huan[†]
[†] Big Data Lab, Baidu Inc. & National Engineering Laboratory for Deep Learning Technology and Applications, China
[§] Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences & University of Chinese Academy of Sciences, China
[‡] Deep Learning Laboratory, Peking University & Beijing Institute of Big Data Research, China
[$] Department of Electrical and Computer Engineering, University of Central Florida, United States
*Equal Contribution
{v_wangkafeng, xionghaoyi, v_bianjiang, huanjun}@baidu.com

Abstract

Stochastic Gradient Hamiltonian Monte Carlo (SGHMC) methods have been widely used to sample from certain probability distributions, incorporating (kernel) density derivatives and/or given datasets. Instead of exploring new samples from kernel spaces, this piece of work proposed a novel SGHMC sampler, namely *Spectral Hamiltonian Monte Carlo* (`SpHMC`), that produces the high dimensional sparse representations of given datasets through sparse sensing and SGHMC. Inspired by compressed sensing, we assume all given samples are low-dimensional measurements of certain high-dimensional sparse vectors, while a continuous probability distribution exists in such high-dimensional space. Specifically, given a dictionary for sparse coding, `SpHMC` first derives a novel likelihood evaluator of the probability distribution from the loss function of LASSO, then samples from the high-dimensional distribution using stochastic Langevin dynamics with derivatives of the logarithm likelihood and Metropolis–Hastings sampling. In addition, new samples in low-dimensional measuring spaces can be regenerated using the sampled high-dimensional vectors and the dictionary. Extensive experiments have been conducted to evaluate the proposed algorithm using real-world datasets. The performance comparisons on three real-world applications demonstrate the superior performance of `SpHMC` beyond baseline methods.

Introduction

Stochastic Gradient Hamiltonian Monte Carlo (SGHMC) algorithms (Chen, Fox, and Guestrin 2014; Ma, Chen, and Fox 2015) have been widely used to support a wide spectrum of applications, such as Bayesian learning (Andrieu et al. 2003), generative sampling, and data augmentation (Antoniou, Storkey, and Edwards 2017). All these tasks require efficient samplers, such as SGHMC, that are capable of drawing samples/parameters from certain prior/posterior distributions. Generally, given the derivative evaluation of the probability density function, SGHMC methods can sample from the distribution through a number of iterations, incorporating a discrete-time second-order Langevin dynamics.

While the evaluation of density derivative is indispensable for SGHMC, the density or its derivative is frequently not known. For example, for some data augmentation applications, a training dataset is given without knowing the probability distributions. One has to generate new samples from the unknown distributions using the dataset and further boost the performance of learning. In such a scenario, existing solutions often use Kernel Density Estimators, such as Gaussian Kernels, to first learn the distribution from the given datasets, then leverage SGHMC to draw new samples from the distributions modeled by kernels. Such solutions, namely *Kernel Hamiltonian Monte Carlo* (Strathmann et al. 2015; Strathmann 2018), have been used to handle the cases that likelihood or density of data is intractable.

Although the Kernel density estimator provides the sampler an evaluator of probability density, the performance of kernel-based solutions is usually limited mainly due to the following three factors: *1) Dimensionality Curse.* it is difficult to scale-up the density estimation on dimensionality, while kernel density estimation over high-dimensional data is frequently inaccurate (Bengio, Delalleau, and Roux 2006); *2) Bandwidth Selection.* When using Gaussian kernels, it consumes time significantly to select an optimal bandwidth for Kernel due to the dimensionality and size of dataset (Raykar and Duraiswami 2006); and *3) Computational Complexity.* It is quite time consuming to compute the derivatives of Kernel density model (Sasaki, Noh, and Sugiyama 2015; Sasaki et al. 2016), i.e., gradients of LogSumExp functions.

Thus, there needs a method to replace kernel density estimation, while providing accurate derivative evaluation of probability density. To lower the computational complexity to compute the derivative of kernel density functions, Sasaki *et al.* (Sasaki, Noh, and Sugiyama 2015; Sasaki et al. 2016) proposed the direct density derivative estimator that learns a regression model through sampling from fitted kernels, then predicts the derivatives using the regression model. With such methods, one can draw samples from the distribution with low computational complexity. However, such methods still cannot handle the dimensionality curse of kernel methods, which may cause significant performance degradation (Strathmann 2018) due to the divergence between fitted and groundtruth distributions.

Our Contributions. From the previous discussions, it is clear that a new probabilistic model is required to define **(i)** the prior probability of samples and characterize **(ii)** the likelihood from the given dataset. In this work, we propose a novel SGHMC sampler, namely *Spectral Hamilto-*

nian Monte Carlo (SpHMC). Instead of exploring in the kernel spaces, SpHMC samples the high dimensional sparse representations from the given dataset (Donoho 2006). Inspired by compressed sensing (Donoho 2006), SpHMC assumes all samples in the datasets are the low-dimensional measurements of certain high-dimensional sparse vectors, based on certain dictionary (Rubinstein, Bruckstein, and Elad 2010). With the distribution of the given dataset, a continuous probability distribution exists in such a high-dimensional space. Thus, each sample drawn from the high-dimensional distribution corresponds to a potential low-dimensional measurement via the same dictionary (and reverse).

Specifically, with a dictionary for compressed sensing, e.g., a random Gaussian matrix (Candès, Romberg, and Tao 2006), SpHMC first proposes a novel log-density evaluator of the probability distribution that characterizes the given datasets. Such log-density evaluator is derived from the loss function of LASSO (Tibshirani 1996), with low complexity solutions for exact derivative computation. Then, given the log-density (derivative) evaluator, SpHMC can draw high-dimensional samples from the distribution using a second-order stochastic Langvien dynamics with a Metropolis–Hastings sampler.

Finally, with the high-dimensional samples drawn, two types of applications can be supported as follows. **(1)** Using the dictionary, one can project the high-dimensional samples to their low-dimensional samples. In this way, SpHMC performs as a generator that reproduces new datums from the datasets with intractable likelihoods. **(2)** In addition, one can directly use generated the high-dimensional samples to augment the machine learning algorithms with sparse coding. For example, some image classification tasks can be improved using compressed sensing, while they can further improved by incorporating the high-dimensional data generation based on the same dictionary. Extensive experiments are conducted to evaluate the proposed algorithms with the two applications using three real-world datasets including MNIST, Fashion MNIST and EMNIST. The performance comparisons on the two applications demonstrate the excellence of SpHMC beyond baseline methods.

Preliminaries and Problems

In this section, we first introduce the backgrounds of this work, then review the most relevant work.

Backgrounds

For a great number of machine learning tasks, sampling from a given distribution is frequently required. Among a wide range of samplers, Markov-Chain Monte Carlo (MCMC) (Andrieu et al. 2003) comprises a class of general-purpose sampling algorithms with fast mixing properties. Through incorporating a Markov chain derived from the distribution, MCMC can sample from the distribution by traversing the Markov chain via a number of steps. To construct the Markov chain from a distribution, the stochastic gradient Hamiltonian Monte Carlo (SGHMC) (Ma, Chen, and Fox 2015; Chen, Fox, and Guestrin 2014) has been proposed to use the discrete-time Langevin dynamics (Welling and Teh 2011) coupled by the gradient flow of logarithm density of the probability distribution (Bagnoli and Bergstrom 2005). The random jumps with the Langvein dynamics help SGHMC traverse the Markov chain and obtain samples.

Specifically, given the density $\mathrm{P}(\mathbf{X}) \propto \exp(-U(\mathbf{X}))$ of the desired distribution, where $U(\mathbf{X})$ is assumed to be convex, SGHMC draws a sequence of samples, e.g., $\mathbf{X}^1, \mathbf{X}^2, \ldots,$ from the distribution $\mathrm{P}(\mathbf{X})$ through discretizing the Langevin dynamics as follow:

$$\begin{cases} \mathrm{d}\mathbf{X} = \mathbf{r}\ \mathrm{d}t \\ \mathrm{d}\mathbf{r} = -\nabla U(\mathbf{X})\mathrm{d}t - \mathbf{B}\mathbf{r}\mathrm{d}t + \mathcal{N}(\mathbf{0}, 2\mathbf{B}\mathrm{d}t), \end{cases} \tag{1}$$

where r is a vector referring the momentum of the dynamics, $\mathbf{B}$ refers to a constant matrix that controls the influence of noise. The noise term $\mathcal{N}(0, 2\mathbf{B}\mathrm{d}t)$ generates a Gaussian noise i.i.d from $\mathcal{N}(\mathbf{0}, 2\mathbf{B})$ over the time t, so as to incorporate randomness during the sampling procedure. To obtain a sample, one can theoretically set the initial state of dynamics as the white noise, i.e., $\mathbf{X}(0) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $r(0) = \mathbf{0}$.

Thus, the derivative evaluator of the log-density is necessary for SGHMC. However, from many applications, such as data generation from datasets, the density function or the derivative evaluator is not accessible. One needs to first learn the distribution from the given dataset. A possible way is to use Kernel methods, among which the Gaussian kernel, aka., Radial basis function (RBF) kernel, is commonly used. Given a dataset $\{x_1, x_2, \ldots, x_n\}$ drawn from a distribution, one can fit a Gaussian kernel to approximate $\mathrm{P}(\mathbf{X})$ as follow.

$$\mathrm{P}(\mathbf{X}) \propto \sum_{i=1}^{n} \exp\left(-\frac{\|\mathbf{X} - x_i\|_2^2}{2h^2}\right), \tag{2}$$

where h refers to the bandwidth of the kernel (Raykar and Duraiswami 2006). With Gaussian kernels, the Kernel Hamiltonian Monte Carlo algorithms have been proposed (Strathmann 2018; Strathmann et al. 2015), while these methods suffer significant performance limitation when the given dataset size is large. It is quite time consuming to compute $-\nabla \log \mathrm{P}(\mathbf{X})$ using the density function modeled in Eq (2). Further selecting an appropriate setting of h is challenging (Raykar and Duraiswami 2006).

Related Work

With above two algorithms, one can successfully generate samples from datasets using stochastic gradient Hamiltonian Monte Carlo with kernels as approximation to the distribution. Given the procedure of Kernel Hamiltonian Monte Carlo, a number of pioneering studies have been done to improve the method. We sort them into four categories:

Dynamics and Averaging. Instead of using the second-order Langevin dynamics, the first-order dynamics, such as the one that stochastic gradient descent (SGD) algorithm behaves as, has been studied for approximately variational inference (Mandt, Hoffman, and Blei 2016) with an sampling-based procedure. The purpose of these dynamics is to minimize the Kullback-Leibler (KL) divergence between the stationary distribution of its continuous-time process and the posterior, while the sampled trajectories can be viewed as a approximation to the inferences. Compared to SGD, SGLD leverages a momentum term to expand the regions that the process explores. In addition to using alternative dynamics, averaging schemes are frequently used to further accelerate the search. Polyak and Juditysky (Polyak and Juditsky 1992) first proved the optimality of averaging for SGD-based inference. While Polyak average requires the storage of the whole trajectory traversed by SGD, the average based on sliding windows helps to lower the space complexity while also ensuring good performance (Mandt, Hoffman, and Blei 2017). Furthermore (Ahn et al. 2015) proposed to use a coupled dynamics for Bayesian inference of matrix factorization. Note that averaging has been frequently used to accelerate Bayesian inference (optimization) but rarely for data generation.

Preconditioning. The (vanilla) Stochastic Gradient Langvein Dynamic (SGLD) listed in Eq. (1) might be significantly influenced by the noise terms incorporated. To control the influence of Noise, Stochastic Gradient Fisher Scoring (SGFS) (Ahn, Korattikara, and Welling 2012) has been firstly proposed to use a positive-definite matrix $\mathbf{H}$ to precondition the dynamics as follow:

$$\begin{cases} \mathrm{d}\mathbf{X} = \mathbf{H}\mathbf{r}\mathrm{d}t \\ \mathrm{d}\mathbf{r} = -\nabla U(\mathbf{X})\mathrm{d}t - \mathbf{B}\mathbf{H}\mathbf{r}\mathrm{d}t + \mathcal{N}(\mathbf{0}, 2\mathbf{B}\mathrm{d}t). \end{cases} \tag{3}$$

The traditional SGLD can be viewed as a special case of SGFS with $\mathbf{H} = \mathbf{I}$. Please refer to Section 5.1 of (Mandt, Hoffman, and Blei 2017) for the analysis. Further (Li et al. 2016; Marceau-Caron and Ollivier 2017) preconditioned SGLD for Bayesian inference of deep networks.

Gradient Estimation. As was mentioned, the derivative evaluator of the log-density is indispensable for gradient-based sampling, while the gradient computation for large high-dimensional datasets is quite time consuming. First of all, to lower the complexity with increasing size of datasets, noisy gradient estimation with mini-batch of samples has been widely used in (Chen, Fox, and Guestrin 2014; Ma, Chen, and Fox 2015; Strathmann 2018; Li, Zhang, and Li 2018). In addition to the direct estimation, some regression-based methods have been studied that can learn to predict the gradient (Sasaki, Noh, and Sugiyama 2015; Sasaki et al. 2016). Furthermore (Filippone and Engler 2015) studied to use conjugate gradient for sampling from Gaussian process with an unbiased solver. Most recent work replaces the common gradients with Fractional-order derivatives (Ye and Zhu 2018), so as to accelerate the Bayesian inference with a log-concave density.

Metropolis–Hastings Correction. The sequence sampled by MCMC can be corrected using Metropolis–Hastings algorithm (Hastings 1970), which rejects the new generated sample, if the new sample is not "with respect to the current sample, according to the distribution". Tons of work have been done to use Metropolis–Hastings or rejection-based method to correct the sampling sequence (Maclaurin and Adams 2014; Andrieu et al. 2003). The most recent work (Dwivedi et al. 2018) proved that bias caused by the step size can be appropriately corrected by using Metropolis–Hastings rejection with strong theoretical consequences under the log-concave assumption.

A comprehensive survey has been made in (Mandt, Hoffman, and Blei 2017). While most of above work intent to enable Bayesian inference with samplers, this paper aims at using gradient-based MCMC to generate datums through sampling from distributions.

Problem Formulation

Given an (unknown) p-dimensional probability distribution $\mathcal{P}$, one first draws n random i.i.d samples $y_1, y_2, \ldots, y_n$ from $\mathcal{P}$. Suppose there exists a $d \times p$ measurement matrix $\mathbf{A}$ and $d \ll p$. With the measurement matrix $\mathbf{A}$ and$y_1, y_2, \ldots, y_n$, one can obtain n d-dimensional (noisy) observations, such that $\forall y_i$

$$x_i = \mathbf{A}y_i + \varepsilon_i \text{ and } \varepsilon_i \sim \mathcal{N}(\mathbf{0}, \delta^2\mathbf{I}), \tag{4}$$

where δ controls the variance of white noise.

Problem. Given the matrix $\mathbf{A}$ and the d-dimensional observation set $\mathcal{D} = \{x_1, x_2, \ldots, x_n\}$, our problem is to generate a sequence of p-dimensional $\mathbf{X}^1, \mathbf{X}^2, \ldots, \mathbf{X}^K$ that fit the unknown probability distribution $\mathcal{P}$.

SpHMC: Spectral Sampler using Stochastic Gradient Hamiltonian Monte Carlo

In this work, we propose SpHMC—*the Spectral Sampler using Stochastic Gradient Hamiltonian Monte Carlo* with ℓ_1-penalized Log-Likelihood. In this section, we first present the overall algorithm design of SpHMC. Then, we introduce the detailed implementation of the algorithms.

The Algorithmic Framework

Given a dataset and a dictionary for sparse reconstruction, SpHMC first models the posterior distribution of data spectrum using the dictionary-based Gaussian likelihoods with a Laplacian prior, then adopt Stochastic Gradient Hamiltonian Monte Carlo (SGHMC) to draw samples from the distribution of spectrum.

Data Input. Algorithm 1 shows the overall design of SpHMC, where the input of algorithms include **(i)** $\mathcal{D}$–the set of original data samples, **(ii)** $\mathbf{A}$ the dictionary for sparse reconstruction in spectral spaces, **(iii)** T total number of iterations for SGLD exploration, **(iv)** K the number of generated samples required, **(v)** m the size of mini-batch, **(vi)** η the step-size for SGLD dynamics discretization, **(vii)** λ the parameter for ℓ_1 regularization for posterior modeling. The algorithm outputs $\mathbf{X}^1,\ \mathbf{X}^2, \ldots, \mathbf{X}^K$–the sequence of K generated samples in the spectral space of $\mathcal{D}$. The overall complexity of this algorithm is $\mathcal{O}(Km)$, while some samples may repeat in the generated sequence (i.e., $\#$ of unique samples $\leq K$). The algorithm is designed as follow.

Initialization. In line 1 of Algorithm 1, SpHMC initializes the whole sampling procedure by defining $\mathbf{X}^0$. A simple way to initialize is using the white noise i.i.d drawn from $\mathcal{N}(0, I)$. Yet another method would first draw an i.i.d sample from the dataset i.e., $z \overset{i.i.d}{\sim} \mathcal{D}$, then performs LASSO to recover its high-dimensional sparse representation using

z and the dictionary $\mathbf{A}$, such that

$$\mathbf{X}^0 \leftarrow \operatorname*{argmin}_{x} \frac{1}{2}\|z-\mathbf{A}x\|_2^2+\lambda\|x\|_1 . \tag{5}$$

In this way, we can start the sequence of sampling from the sparse representation of a (known) random sample.

Algorithm 1 Spectral Sampler using Stochastic Gradient Hamiltonian Monte Carlo

```
1:  procedure SpHMC($\mathcal{D}$, $\mathbf{A}$, $T$, $K$, $m$, $\eta$, $\lambda$)
2:      Initialization: $\mathbf{X}^0$
3:      /*Sequence Sampling*/
4:      for all $k = 1, 2, 3, \ldots, K$ do
5:          $\mathbf{r}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
6:          $\mathbf{X}_0 \leftarrow \mathbf{X}^{k-1}$
7:          /* Gradient-based sampling with $T$ Steps*/
8:          for all $t = 1, 2, 3, \ldots, T$ do
9:              $\mathbf{g}_t \leftarrow \mathrm{GradEst}(\mathbf{X}_{t-1}, \mathcal{D}, \mathbf{A}, m, \lambda)$
10:             $\mathbf{r}_t \leftarrow \mathbf{r}_{t-1} - \eta\mathbf{g}_t - \eta\mathbf{r}_{t-1}$
11:             $\mathbf{X}_t \leftarrow \mathbf{X}_{t-1} + \eta\mathbf{r}_t$
12:         end for
13:         /* Metropolis-Hastings Correction*/
14:         $\mathbf{X}^k \leftarrow \mathrm{MH}(\mathbf{X}^{k-1}, \mathbf{X}_T, T, \mathcal{D}, \lambda, \eta)$
15:     end for
16:     return $\mathbf{X}^1,\ \mathbf{X}^2,\ \mathbf{X}^3, \ldots, \mathbf{X}^K$
17: end procedure
```

Sampling Sparse Representation from Spectral Distributions. In Lines 4–15 of Algorithm 1, SpHMC leverages a loop of K iterations to generate a sequence of K samples, where each iteration performs a iterative process of T steps to obtain the next sample generated in the sequence. Specifically, in each iteration (e.g., the k^{th} iteration), SpHMC first setups $\mathbf{r}_0$ using the white noise and $\mathbf{X}_0$ with the previous sampling result $\mathbf{X}^{k-1}$ in the sequence, so as to initialize $\mathbf{r}$ and $\mathbf{X}$ of the SGLD dynamics. Then, SpHMC walks T steps of discrete-time SGLD with noisy gradient estimator (in line 9) to reach the potentially next sample $\mathbf{X}_T$. Further, a rejection-based Metropolis-Hastings correction is given to decide whether (1) to accept $\mathbf{X}_T$ as $\mathbf{X}^k$ or (2) to reject $\mathbf{X}_T$ while reusing $\mathbf{X}^{k-1}$ as $\mathbf{X}^k$. The design and implementation of the noisy gradient estimator and Metropolis-Hastings correction would be introduced in the next sections.

Finally, as shown in line 16 of the Algorithm 1, SpHMC outputs the sequence as the sampling results. Note that such sampling procedure depends on the initial status such as $\mathbf{X}^0$. Then it is better to repeat the whole algorithm multiple times while setting a relevant small K for each run, to balance the bias and the time complexity.

Data Reconstruction via Compression. Note that the output of Algorithm 1 $\mathbf{X}^1, \mathbf{X}^2, \ldots, \mathbf{X}^K$ are a sequence of sparse representations based on the dictionary $\mathbf{A}$. To obtain the generated datums, which can be observed in low dimension, one can use $\mathbf{A}$ to compress the sampling results, such that for $\forall\ \mathbf{X}^k$ for $1 \leq k \leq K$ one can compute $\mathbf{A}\mathbf{X}^k$ as the k^{th} generated sample of low-dimensional observations. Note that some simple soft-thresholding strategies can applied here to de-noise the data (e.g., images) and improve the reconstruction.

Gradient Estimation for Negative Log-Density

Algorithm 2 demonstrates the design of (noisy) gradient estimator used in the SGLD dynamics and sampling. Specifically, we define the posterior probability distribution of the high-dimensional sparse representation of the dataset $\mathcal{D}$ using Bayes' theorem as follow:

$$\mathbb{P}(\mathbf{X}|\mathcal{D}) \propto \mathbb{P}(\mathbf{X})\mathbb{P}(\mathcal{D}|\mathbf{X}) = \mathbb{P}(\mathbf{X}) \prod_{\forall x\in\mathcal{D}} \mathbb{P}(x|\mathbf{X}) , \tag{6}$$

where $\mathbb{P}(\mathbf{X})$ refers to the prior distribution and $\mathbb{P}(x_i|\mathbf{X})$ is the likelihood of $x \in \mathcal{D}$ given the high-dimensional sparse representation. Then, we define the prior distribution of $\mathbf{X}$ as the Laplacian distribution such that

$$\mathbb{P}(\mathbf{X}) \propto \exp(-\lambda\|\mathbf{X}\|_1) . \tag{7}$$

Further, the likelihood of low-dimensional observation $\forall x \in \mathcal{D}$ given the high-dimensional sparse representation $\mathbf{X}$ can be modeled using Gaussian distribution, such that

$$\mathbb{P}(x|\mathbf{X}) \propto \exp\left(-\frac{1}{|\mathcal{D}|}\|x-\mathbf{A}\mathbf{X}\|_2^2\right) . \tag{8}$$

In this way the negative log-density can be modeled as

$$-\log\ \mathbb{P}(\mathbf{X}|\mathcal{D}) = \frac{1}{|\mathcal{D}|}\sum_{\forall x\in\mathcal{D}} \|x-\mathbf{A}\mathbf{X}\|_2^2 + \lambda\|\mathbf{X}\|_1 + c , \tag{9}$$

where c is a constant. We find above function above actually averages the LASSO losses using all samples in the dataset $\mathcal{D}$. The gradient of the negative log-density can be written as

$$\begin{aligned}&\nabla\left(-\log\ \mathbb{P}(\mathbf{X}|\mathcal{D})\right)\\ &= \frac{1}{|\mathcal{D}|}\sum_{\forall x\in\mathcal{D}} \mathbf{A}^\top\left(\mathbf{A}\mathbf{X}-x\right) + \lambda\cdot\mathrm{sign}(\mathbf{X}) ,\end{aligned} \tag{10}$$

where $\mathrm{sign}(\cdot) \rightarrow \{\pm 1\}$ is the signal function. According to Lines 3–4 of Algorithm 2, the vector $\mathbf{g}_m$ indeed is a noisy estimation of the gradient with a mini-batch of m random samples drawn from $\mathcal{D}$, where the noise in the gradient estimation can be well controlled by m.

Algorithm 2 Noisy Gradient Estimation for ℓ_1-Penalized Log-Likelihood with Random Mini-Batch

```
1: procedure GradEst($\mathbf{X}$, $\mathcal{D}$, $\mathbf{A}$, $m$, $\lambda$)
2:     /*Noisy gradient estimation*/
3:     $M \leftarrow$ draw $m$ i.i.d samples from $\mathcal{D}$
4:     $\mathbf{g}_m \leftarrow \mathbf{A}^\top\mathbf{A}\mathbf{X} - \frac{1}{m}\sum_{\forall x\in M}\mathbf{A}^\top x + \lambda\cdot\mathrm{sign}(\mathbf{X})$
5:     return $\mathbf{g}_m$
6: end procedure
```

Linearized Metropolis-Hastings Correction

Algorithm 3 presents the design of the Metropolis-Hastings correction mechanism used in Line 14 of Algorithm 1. The major input of this algorithm includes **(i)** the previous generated sample $\mathbf{X}$ (i.e., $\mathbf{X}^k$ in Algorithm 1), **(ii)** the current approaching sample $\mathbf{Z}$ (i.e., $\mathbf{X}_T$ in Algorithm 1), **(iii)** the number of iterations required T, and **(iv)** the step size of SGLD η. `SpHMC` aims at correcting the bias caused by the discretization of dynamics with step size η. The output of `SpHMC` would assign to the newly generated sample $\mathbf{X}^k$.

This algorithm first estimates the *transition probability* from $\mathbf{X}$ to $\mathbf{Z}$ and reverse. Then, the algorithm output $\mathbf{X}$ or $\mathbf{Z}$ depending on whether $\mathbf{X}$ would be accepted or rejected respectively. In Lines 2–7 of Algorithm 3, `SpHMC` evaluates the negative log-density and the derivatives on $\mathbf{X}$ and $\mathbf{Z}$ based on the whole dataset $\mathcal{D}$ respectively. Then it estimates the ratio α between the transition probability from $\mathbf{X}$ to $\mathbf{Z}$ and the transition probability from $\mathbf{Z}$ to $\mathbf{X}$ (in line 9). Such ratio is upper-bounded by 1. Then, like other Metropolis-Hastings algorithms, `SpHMC` randomly draws $\gamma \sim \mathrm{Uniform}[0, 1]$ and compares γ to α to make the decision for acceptance/rejection. Note that the newly approaching sample $\mathbf{Z}$ would be rejected when the ratio α, between the transition probabilities from $\mathbf{X}$ to $\mathbf{Z}$ and reverse, is not greater than the random number γ.

Algorithm 3 Linearized Metropolis-Hastings Correction

```
1:  procedure MH($\mathbf{X}$, $\mathbf{Z}$, T, $\mathcal{D}$, $\lambda$, $\eta$)
2:      /*Function and Derivative Evaluation*/
3:      $f_x \leftarrow \frac{1}{2|\mathcal{D}|}\sum_{\forall x \in \mathcal{D}} \|\mathbf{AX} - x\|_2^2 + \lambda\|\mathbf{AX}\|_1$
4:      $f_z \leftarrow \frac{1}{2|\mathcal{D}|}\sum_{\forall x \in \mathcal{D}} \|\mathbf{AZ} - x\|_2^2 + \lambda\|\mathbf{AZ}\|_1$
5:      $\nabla f_x \leftarrow \frac{1}{|\mathcal{D}|}\sum_{\forall x \in \mathcal{D}} \mathbf{A}^\top(\mathbf{AX} - x) + \lambda \cdot \mathrm{sign}(\mathbf{X})$
6:      $\nabla f_z \leftarrow \frac{1}{|\mathcal{D}|}\sum_{\forall x \in \mathcal{D}} \mathbf{A}^\top(\mathbf{AZ} - x) + \lambda \cdot \mathrm{sign}(\mathbf{Z})$
7:      $\Delta \leftarrow \eta T$
8:      /*Transition Probability Estimation*/
9:      $\alpha = \mathrm{min.}\left\{1, \frac{\exp(-f_z - \|\mathbf{X}-\mathbf{Z}+\Delta\cdot\nabla f_z\|_2^2/4\Delta)}{\exp(-f_x - \|\mathbf{Z}-\mathbf{X}+\Delta\cdot\nabla f_x\|_2^2/4\Delta)}\right\}$
10:     $\gamma \overset{i.i.d}{\sim} \mathrm{Uniform}[0, 1]$
11:     if $\alpha > \gamma$ then
12:         return $\mathbf{Z}$ /* Accept $\mathbf{Z}$*/
13:     else
14:         return $\mathbf{X}$ /* Reject $\mathbf{Z}$*/
15:     end if
16: end procedure
```

To lower the complexity of estimation, the transition probability estimation is linearized, e.g., $\Delta \cdot \nabla f_x$ or $\Delta \cdot \nabla f_z$, using the aggregated step size $\Delta = \eta T$ without considering the second-order dynamics. In this way, we don't need to calculate the aggregation of the T steps. On the other hand, When $T = 1$, the proposed algorithm would perform exactly same as (Dwivedi et al. 2018) with strong theoretical guarantee to eliminate bias caused by the finite step-size η.

Experiments and Empirical Validation

We propose to evaluate `SpHMC` using three real-world benchmark datasets including MNIST, Fashion MNIST, and EMNIST for two applications such as data synthesis & regeneration, spectral data augmentation for supervised machine learning and image classification.

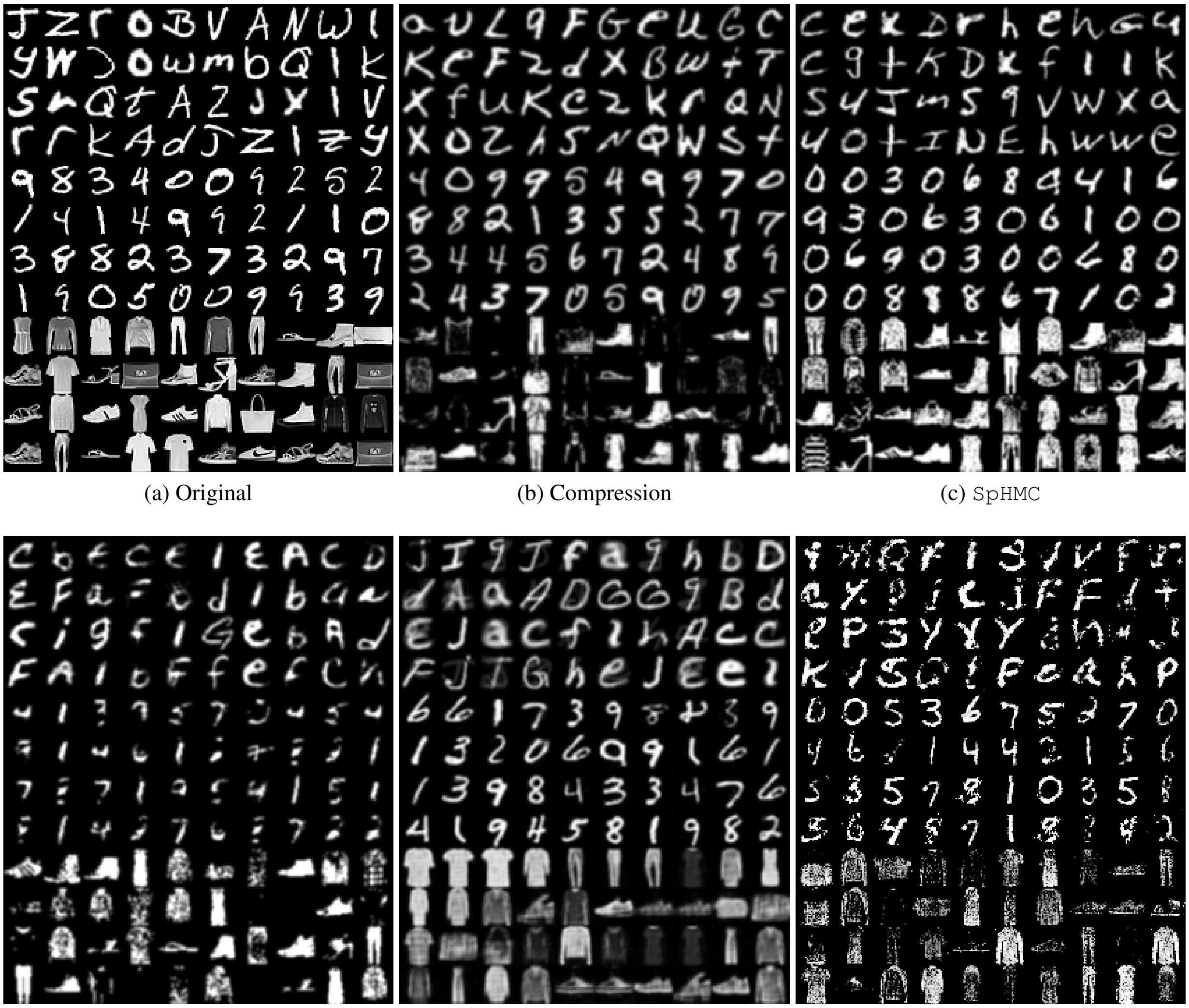
Image Data Set Synthesis & Regeneration

To better understand the performance of `SpHMC` for image data generation, we evaluate the proposed algorithms with baselines using the visionary datasets. All these datasets consist of images with 28×28 pixels at gray-scale. In our research, we consider each image as a vector of 784 dimensions, where each dimension is scaled from 0 to 255.

Baselines and Setup We include following methods as the baseline algorithms for comparison. The work that are most relevant to our work is Kernel Hamiltonian Monte Carlo (Kernel HMC) (Strathmann et al. 2015; Strathmann 2018), which enables Bayesian sampling without the derivative evaluator of the (log) posterior density. In addition to Kernel HMC, we also want to measure the affect the Metropolis-Hasting (MH) correction to the performance of sampling. Thus, we also include the option to disable/enable the MH correction during the sampling procedure. With MH correction disabled, all samples that are traversed by the dynamics will be accepted as the data generation. In this way, we provide three key baseline algorithms: *i) `SpHMC` without MH correction*, *ii) Kernel HMC*, and *iii) Kernel HMC without MH correction*. The comparison between `SpHMC` to its variant without MH correction demonstrates the improvement made through rejecting low probability samples for data generation, while the comparison to *Kernel HMC* illustrates the effectiveness of our fantastic intuition that leverages sparse coding/reconstruction in Monte Carlo settings (rather than compressed sensing). We also compare `SpHMC` to a conditional Generative Adversarial Network (GAN) (Goodfellow et al. 2014; Mirza and Osindero 2014).

Further, we setup the proposed algorithm `SpHMC` (and `SpHMC` without MH correction) with a 784×3136 Gaussian random noise matrix as the measurement $\mathbf{A}$. With such measurement matrix, `SpHMC` can reconstruct a higher dimensional (i.e., 4 times of original dimensions) probability distribution that characterizes the given datasets as its low dimensional observations. Note that the performance of `SpHMC` can be further improved with a fine-tuned the measurement matrix $\mathbf{A}$ given through dictionary learning, though a Gaussian matrix is capable of providing certain theoretical consequence. All other parameters such as regularization term λ, step size η and batch size m for both `SpHMC` and baselines are all tuned best with repeat trials. All generated images (except GAN) are filtered by the same Gaussian smoother to rescale each pixel at 0–255 range.

Results Figures 1(a)–(f) demonstrate the comparison of original images and generated images by algorithms. Specifically, Figures 1(a) refers to the images randomly drawn from the three datasets, where the top four rows are from EMNIST datasets, the four rows in the middle are drawn from the MNIST datasets, and the bottom four rows are from Fashion MNIST dataset. In Figure 1 (b), we demonstrate the results based on *Compression*, where for each image, we first draw a random image from the original dataset,



(a) Original

(b) Compression

(c) SpHMC

(d) SpHMC w/o MH

(e) Kernel HMC

(f) GAN

Figure 1: Examples of Data Synthesis and Generation

then we use LASSO and measurement matrix $\mathbf{A}$ to reconstruct its high-dimensional representation, further we use $\mathbf{A}$ to obtain its low-dimensional observation. We expect to observe the affect the reconstruction and compression to the original image. The comparison between compression to the original shows that the shape of letters and digits for EMNIST and MNIST datasets can be well preserved through the reconstruction-and-compression procedure, while the image quality for Fashion MNIST image reconstruction is poor.

Figures (c)–(h) presents the generated images of the three datasets using SpHMC, SpHMCwithout MH correction, Kernel HMC, and GAN, respectively. All these models have successfully generated images that are visible and human understandable. It is quite subjective to judge or compare the quality of these images. We can see that the original images indeed incorporate some local patterns, such as shadows and textures of materials. Most of sampling methods can well reconstructs the shapes while missing the supports to the local patterns, SpHMC seems working well for both shape and pattern recovery. Comparing SpHMC to SpHMC w/o MH (the SpHMC wit MH Correction disabled), the images procedure by SpHMC can be reviewed as a subset of images by SpHMC w/o MH, which has been carefully selected by the transition probabilities. The images sampled by SpHMC are rendered with higher sharpness, while the edges of SpHMC w/o MH are usually blurred. Kernel HMC is with the similar drawback, where the produced images are lack of detailed local patterns and with blurred edges. The images produced by CGAN are generally with good quality and local patterns reconstructed. From a Bayesian sampling perspectives, we doubt whether CGAN or the general Generative Adversarial Net can sample from the original distribution of images consistently. Recent study (Richardson and Weiss 2018) shows that GAN failed to capture/cover the whole distribution fro sample generation with significant biases.

In summary, we conclude that SpHMC is capable of gen-

Table 1: Error Comparison using SpHMC on MNIST, Fashion MNIST and EMNIST Datasets

	MNIST (LeCun, Cortes, and Burges 2010) ($n = 60,000,\ d = 784$)				
	SVM	ℓ_2-SVM	ℓ_1-Log Reg	ℓ_2-Log Reg	ℓ_2-Perceptron
SpHMC	**3.59 ± 0.00**	**5.74 ± 0.06**	**5.49 ± 0.01**	**5.23 ± 0.08**	**12.54 ± 0.49**
SpHMC w/o MH	3.61 ± 0.03	5.87 ± 0.19	5.50 ± 0.09	5.38 ± 0.06	18.84 ± 2.60
Original	5.96 ± 0.00	11.11 ± 1.29	8.06 ± 0.01	7.99 ± 0.00	16.62 ± 0.00
Compression	3.61 ± 0.00	5.84 ± 0.18	5.50 ± 0.00	5.32 ± 0.00	13.22 ± 0.00
Kernel HMC	6.14 ± 0.11	15.44 ± 2.40	34.14 ± 7.11	13.46 ± 0.79	29.42 ± 4.98
GAN	5.99 ± 0.09	11.33 ± 0.45	8.11 ± 0.14	8.15 ± 0.89	15.05 ± 1.27
	Fashion MNIST (Xiao, Rasul, and Vollgraf 2017) ($n = 60,000,\ d = 784$)				
	SVM	ℓ_2-SVM	ℓ_1-Log Reg	ℓ_2-Log Reg	ℓ_2-Perceptron
SpHMC	**13.09 ± 0.08**	**15.19 ± 0.27**	**14.33 ± 0.00**	**14.54 ± 0.08**	**19.82 ± 0.69**
SpHMC w/o MH	13.54 ± 0.08	15.42 ± 0.35	14.46 ± 0.08	14.69 ± 0.09	22.18 ± 2.62
Original	15.36 ± 0.00	19.88 ± 2.91	15.81 ± 0.03	15.89 ± 0.00	30.84 ± 0.00
Compression	13.20 ± 0.00	15.32 ± 0.26	14.34 ± 0.01	14.55 ± 0.00	25.09 ± 0.00
Kernel HMC	15.73 ± 0.18	27.07 ± 2.36	25.70 ± 0.78	24.99 ± 0.44	37.30 ± 7.73
GAN	15.39 ± 0.10	19.29 ± 1.26	15.99 ± 0.09	16.09 ± 0.18	34.17 ± 2.69
	EMNIST (Cohen et al. 2017) ($n = 124,800,\ d = 784$)				
	SVM	ℓ_2-SVM	ℓ_1-Log Reg	ℓ_2-Log Reg	ℓ_2-Perceptron
SpHMC	**13.14 ± 0.09**	**15.38 ± 0.11**	**17.64 ± 0.11**	**14.41 ± 0.07**	**19.83 ± 0.69**
SpHMC w/o MH	13.32 ± 0.06	22.71 ± 0.10	18.37 ± 0.13	18.43 ± 0.07	49.36 ± 2.48
Original	21.21 ± 0.00	37.33 ± 0.80	28.82 ± 0.00	29.10 ± 0.00	50.62 ± 0.00
Compression	13.21 ± 0.00	22.60 ± 0.17	18.33 ± 0.01	18.36 ± 0.00	49.43 ± 0.00
Kernel HMC	25.18 ± 0.19	42.54 ± 2.26	31.59 ± 0.09	31.99 ± 0.07	56.55 ± 2.80
GAN	22.17 ± 0.14	44.26 ± 1.70	33.16 ± 0.29	33.21 ± 0.40	57.83 ± 2.35

erating images. Actually. it samples from the distribution of sparse representation from the datasets, while one can easily convert these sparse representation back to the images using the dictionary (i.e., measurement matrix). Compared to the images produced Kernel Hamiltonian Monte Carlo and GAN, we cannot observe the significant drawback of SpHMC. We subjectively the images generated by SpHMC look better, as they preserve more local patterns and textures.

Data Augmentation

To further evaluate the quality of samples drawn, we will evaluate SpHMC, as a tool for data augmentation, using a wide range of linear classifiers including SVM, ℓ_2-SVM, ℓ_1-regularized Logistic Regression (entitled ℓ_1-Log. Reg.), ℓ_2-regularized Logistic Regression (entitled ℓ_2-Log. Reg.), and ℓ_1-regularized Perceptron (entitled ℓ_1-Perceptron). We don't intend to perform such comparison on top of neural networks, as these methods already incorporate data augmentation in their deep architectures.

In this experiment, we shuffle the original datasets with the images generated in the ratio of 6:1. Note that for the experiment based on SpHMC and compression, we use the spectral data (generated spectral samples by SpHMC and/or the one obtained by LASSO using the same dictionary) for training and testing rather than the images. All algorithms are tuned with the best hyper-parameters through 10 folder cross-validation on the training set. We repeat the experiments 5 times to estimate the accuracy with intervals.

Results and Comparison. Table 1 presents the testing error comparison of the six classifiers on the three datasets with various augmentations. SpHMC outperforms all baseline methods with higher accuracy, while it marginally improves the results of compression (which consists of the spectral representation of original data). It is obvious that, in the most cases, the upper interval of the error of SpHMC (indicating the worst case accuracy) is still lower than the lower interval of baselines (indicating the best case accuracy). The comparison shows that SpHMC significantly outperform the one based on original dataset, Kernel HMC and GAN with clearly higher accuracy. The comparison between SpHMC and Kernel HMC demonstrate the power of sampling sparse representation from space, rather than the spatial-temporal information of images. The comparison between SpHMC and SpHMC without MH correction shows the power of rejecting low probability samples to well augment the training set. The compression between SpHMC and Compression shows the effectiveness of Bayesian sampling for generating new spectral samples. We consider the advantage of SpHMC is due to the effectiveness combining sparse coding and Bayesian sampling.

Discussion and Conclusion

In this paper, we propose SpHMC, which aims at sampling the sparse representations from the given dataset without the need of traceable likelihoods. Incorporating a dictionary (or measurement matrix) for sparse coding, SpHMC leverages a Stochastic Gradient Langevin Dynamics to traverse on a log-posterior density model derived from compressed sensing, where a sequence of samples can be generated via the trajectory sampled by the dynamics. The empirical validation, based three benchmark image datasets, shows SpHMC can draw the possible sparse representation from the spectral space of the images, while the sparse representations drawn can be used to generate new images. We compare the images generated by SpHMC with those based on Kernel HMC and GAN. While all generated images are visible and human understandable, the images produced by SpHMC seem to preserve more local patterns and textures. Moreover, our experiments indicate that the generated data could help to augment the original datasets for supervised learning tasks. The comparison shows SpHMC significantly outperforms the one augmented by Kernel HMC and GAN with higher classification accuracy.

Acknowledgement

We appreciate efforts made by reviewers and PC members to review, comment and improve this paper. Our research is supported by following grants: National Key R&D Program of China (2018YFB1004804); NSFC under grant U1401258, No. 61802387 and No. 61872010; Science and Technology Planning Project of Guangdong Province(2015B010129011); Shenzhen Discipline Construction Project for Urban Computing and Data Intelligence; National Science Foundation (NSF) CRII: CSR: NeuroMC—Parallel Online Scheduling of Mixed-Criticality Real-Time Systems via Neural Networks (# 1755965).

References

Ahn, S.; Korattikara, A.; Liu, N.; Rajan, S.; and Welling, M. 2015. Large-scale distributed bayesian matrix factorization using stochastic gradient mcmc. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 9–18. ACM.

Ahn, S.; Korattikara, A.; and Welling, M. 2012. Bayesian posterior sampling via stochastic gradient fisher scoring. *arXiv preprint arXiv:1206.6380.*

Andrieu, C.; De Freitas, N.; Doucet, A.; and Jordan, M. I. 2003. An introduction to mcmc for machine learning. *Machine learning* 50(1-2):5–43.

Antoniou, A.; Storkey, A.; and Edwards, H. 2017. Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340.*

Bagnoli, M., and Bergstrom, T. 2005. Log-concave probability and its applications. *Economic theory* 26(2):445–469.

Bengio, Y.; Delalleau, O.; and Roux, N. L. 2006. The curse of highly variable functions for local kernel machines. In *Advances in neural information processing systems*, 107–114.

Candès, E. J.; Romberg, J.; and Tao, T. 2006. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on information theory* 52(2):489–509.

Chen, T.; Fox, E.; and Guestrin, C. 2014. Stochastic gradient hamiltonian monte carlo. In *International Conference on Machine Learning*, 1683–1691.

Cohen, G.; Afshar, S.; Tapson, J.; and Schaik, A. V. 2017. Emnist: an extension of mnist to handwritten letters.

Donoho, D. L. 2006. Compressed sensing. *IEEE Transactions on information theory* 52(4):1289–1306.

Dwivedi, R.; Chen, Y.; Wainwright, M. J.; and Yu, B. 2018. Log-concave sampling: Metropolis-hastings algorithms are fast! *arXiv preprint arXiv:1801.02309.*

Filippone, M., and Engler, R. 2015. Enabling scalable stochastic gradient-based inference for gaussian processes by employing the unbiased linear system solver (ulisse). *arXiv preprint arXiv:1501.05427.*

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, 2672–2680.

Hastings, W. K. 1970. Monte carlo sampling methods using markov chains and their applications.

LeCun, Y.; Cortes, C.; and Burges, C. 2010. Mnist handwritten digit database. *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist* 2.

Li, C.; Chen, C.; Carlson, D.; and Carin, L. 2016. Preconditioned stochastic gradient langevin dynamics for deep neural networks. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 1788–1794. AAAI Press.

Li, Z.; Zhang, T.; and Li, J. 2018. Stochastic gradient hamiltonian monte carlo with variance reduction for bayesian inference. *arXiv preprint arXiv:1803.11159.*

Ma, Y.-A.; Chen, T.; and Fox, E. 2015. A complete recipe for stochastic gradient mcmc. In *Advances in Neural Information Processing Systems*, 2917–2925.

Maclaurin, D., and Adams, R. P. 2014. Firefly monte carlo: Exact mcmc with subsets of data. In *UAI*, 543–552.

Mandt, S.; Hoffman, M.; and Blei, D. 2016. A variational analysis of stochastic gradient algorithms. In *International Conference on Machine Learning*, 354–363.

Mandt, S.; Hoffman, M. D.; and Blei, D. M. 2017. Stochastic gradient descent as approximate bayesian inference. *The Journal of Machine Learning Research* 18(1):4873–4907.

Marceau-Caron, G., and Ollivier, Y. 2017. Natural langevin dynamics for neural networks. In *International Conference on Geometric Science of Information*, 451–459. Springer.

Mirza, M., and Osindero, S. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784.*

Polyak, B. T., and Juditsky, A. B. 1992. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization* 30(4):838–855.

Raykar, V. C., and Duraiswami, R. 2006. Fast optimal bandwidth selection for kernel density estimation. In *Proceedings of the 2006 SIAM International Conference on Data Mining*, 524–528. SIAM.

Richardson, E., and Weiss, Y. 2018. On gans and gmms. *arXiv preprint arXiv:1805.12462.*

Rubinstein, R.; Bruckstein, A. M.; and Elad, M. 2010. Dictionaries for sparse representation modeling. *Proceedings of the IEEE* 98(6):1045–1057.

Sasaki, H.; Noh, Y.-K.; Niu, G.; and Sugiyama, M. 2016. Direct density derivative estimation. *Neural computation* 28(6):1101–1140.

Sasaki, H.; Noh, Y.-K.; and Sugiyama, M. 2015. Direct density-derivative estimation and its application in kl-divergence approximation. In *Artificial Intelligence and Statistics*, 809–818.

Strathmann, H.; Sejdinovic, D.; Livingstone, S.; Szabo, Z.; and Gretton, A. 2015. Gradient-free hamiltonian monte carlo with efficient kernel exponential families. In *Advances in Neural Information Processing Systems*, 955–963.

Strathmann, H. 2018. *Kernel methods for Monte Carlo*. Ph.D. Dissertation, UCL (University College London).

Tibshirani, R. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)* 267–288.

Welling, M., and Teh, Y. W. 2011. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 681–688.

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

Ye, N., and Zhu, Z. 2018. Stochastic fractional hamiltonian monte carlo. In *27th International Joint Conference on Artificial Intelligence (IJCAI 2018)*.