Piecewise Deterministic Markov Processes for Bayesian Neural Networks

Ethan Goan[1] **Dimitri Perrin**[1] **Kerrie Mengersen**[1] **Clinton Fookes**[1]

[1]Queensland University of Technolgy

Abstract

Inference on modern Bayesian Neural Networks (BNNs) often relies on a variational inference treatment, imposing violated assumptions of independence and the form of the posterior. Traditional MCMC approaches avoid these assumptions at the cost of increased computation due to its incompatibility to subsampling of the likelihood. New Piecewise Deterministic Markov Process (PDMP) samplers permit subsampling, though introduce a model-specific inhomogenous Poisson Process (IPPs) which is difficult to sample from. This work introduces a new generic and adaptive thinning scheme for sampling from these IPPs, and demonstrates how this approach can accelerate the application of PDMPs for inference in BNNs. Experimentation illustrates how inference with these methods is computationally feasible, can improve predictive accuracy, MCMC mixing performance, and provide informative uncertainty measurements when compared against other approximate inference schemes.

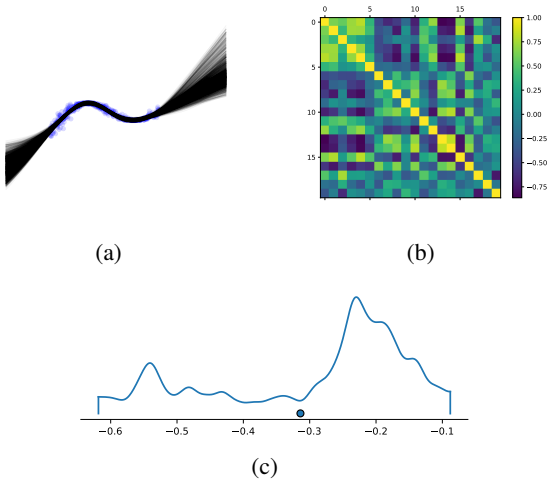


Figure 1: Example of correlations between the parameters in the first layer of a BNN for a simple regression task. Plot (a) samples of predictive posterior from proposed method, (b) correlation between all parameters on the first layer, (c) kernel density estimate for a single parameter.

1 INTRODUCTION

Since Hamiltonian Monte Carlo (HMC) was first developed for Bayesian inference Neal [2012], sampling methods have seen relatively little application to Bayesian Neural Networks (BNNs). Flexibility, inference diagnostics and asymptotic guarantees of HMC comes at the cost of computational complexity as each data point needs to be used to compute the entire likelihood, and to perform Metropolis Hastings corrections. As models and data sets have grown, this expense has not been offset by the considerable performance increase in computational hardware. A recent study found that the fitting of a HMC model for ResNet20 required a computational cost equivalent to 60 million SGD epochs to obtain only 240 samples from three chains Izmailov et al. [2021].

To circumnavigate the computational expense, much research has explored the application of approximate inference through the lens of Variational Inference (VI) Jordan et al. [1999], Wainwright and Jordan [2008], Blei et al. [2017] or through exploiting properties of SGD Mandt et al. [2017]. VI replaces the true target distribution with an approximate distribution that can be easily manipulated, typically using a mean-field approach where independence between parameters is assumed. These methods are attractive due to their reduced computational complexity and their amenability to stochastic optimisation. However, the suitability of these methods relies heavily on the expressiveness of the approximate posterior to accurately model the true distribution. Given the known correlations and frequent multi-

Proceedings of the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023), PMLR 216:712–722.

modal structure amongst parameters within BNNs Barber and Bishop [1998], MacKay [1995], a mean-field approximation can be unsuitable for accurate inference. Figure 1 illustrates these properties for a simple BNN. Stochastic gradient MCMC methods such as Stochastic Gradient Langevin Dynamics (SGLD) aim to address this issue but requires prohibitively small and decreasing learning rates to target the posterior that limits their applicability Nagapetyan et al. [2017].

This work explores a new set of "exact" inference methods based on Piecewise Deterministic Markov Processs (PDMPs) Davis [1984] to perform Bayesian inference. PDMP methods can maintain the true posterior as its invariant distribution during inference whilst permitting subsampling of the likelihood at each update. This property is attractive for BNNs which typically are of large dimension in terms of parameters and data sets. Furthermore, previous research has highlighted PDMP methods for favourable performance in terms of mixing and sampling efficiency Bouchard-Côté et al. [2018], Bierkens et al. [2019], Wu and Robert [2017], Bierkens et al. [2020]. The dynamics of these samplers are simple to simulate, though simulating the times to update these dynamics is controlled by an Inhongeneous Poisson Process (IPP) which can be difficult to sample. This work explores an adaptive procedure to approximate sampling of these event times to allow for approximate inference within the context of BNNs. The contributions of this paper are the following,

- Propose a novel adaptive thinning method for approximate sampling from IPPs events
- Develop a GPU-accelerated package for applying these methods to general models
- Evaluate the performance of these methods applied to computer vision tasks using BNNs;
- Evaluate the suitability of PDMP samplers for BNNs and investigate how they can improve predictive accuracy, calibration and posterior exploration when compared against SGLD.

MCMC methods have often been seen as computationally prohibitive for models with many parameters or where modern large data sets are used. It is hoped that this work will demonstrate that approximate inference using MCMC approaches for BNNs can be practically feasible, offer insightful results, and to show how we can leverage exact methods for approximate inference to more accurately target posterior distributions in BNNs.

2 PRELIMINARIES

Following the description from Fearnhead et al. [2018], PDMP are defined by three key components: piecewise deterministic dynamics, an event rate, and the transition kernel. For inference, the goal is to design these three key components such that we can use the properties of a PDMP to sample from the posterior distributions of our parameters ω. We represent the deterministic dynamics as $\Psi(\omega, \mathbf{v}, t)$, where $\mathbf{v}$ is an auxiliary velocity variable to guide posterior exploration with known distribution $\Phi(\mathbf{v})$ and t represents time. At random events, these dynamics are updated in accordance to a specified transition kernel. Upon an update event, the piecewise deterministic dynamics of the system update according to the kernel, and the state ω at the time of the update event serves as the starting position for the next segment such that they are all connected.

An IPP with rate function $\lambda(\omega(t), \mathbf{v}(t))$ governs the update times for the dynamics. All rate functions in this work rely upon the negative joint log probability of the model,

$$U(\omega) = -\log\Big(p(\omega)p(\mathcal{D}|\omega)\Big), \tag{1}$$

where $p(\omega)$ is a prior or reference measure and $p(\mathcal{D}|\omega)$ is our likelihood, If these three components are suitably defined, these processes can sample from a given posterior distribution. For derivations on how to design these components to target a posterior distribution, the reader can refer to Fearnhead et al. [2018], Vanetti et al. [2017], Davis [1993]. We now introduce the samplers used within this work.

2.1 BOUNCY PARTICLE SAMPLER

The dynamics of the Bouncy Particle Sampler (BPS) Bouchard-Côté et al. [2018] are given by $\Psi(\omega, \mathbf{v}, t) = \omega^i + \mathbf{v}^i t$, where the superscripts indicate a deterministic segment. The velocity remains constant within these segments and the parameter space is explored linearly. The velocity is updated at event times given by $\tau \sim$ IPP$(\lambda(\omega(t)), \mathbf{v})$, where,

$$\lambda(\omega(t), \mathbf{v}) = \max\{0, \nabla U(\omega) \cdot \mathbf{v}^i\}. \tag{2}$$

Once an event time is sampled, the state of our variable "bounces" according to a lossless inelastic Newtonian collision,

$$\mathbf{v}^{i+1} = \mathbf{v}^i - 2\frac{\nabla U(\omega^{i+1}) \cdot \mathbf{v}^i}{\|\nabla U(\omega^{i+1})\|^2}\nabla U(\omega^{i+1}) \tag{3}$$

where ω^{i+1} represents the end of the previous segment at time τ, and serves as the starting position for the following segment. The BPS provides linear dynamics that are simple to simulate, though relies only on local gradient information, which can lead to inefficient exploration for BNNs. Preconditioning can allow us to address this.

2.2 PRECONDITIONED BPS

To accelerate posterior exploration in directions of interest, we can precondition the gradients to include more infor-

mation about the structure of our posterior space. Introduction of a preconditioning matrix A results in new dynamics of $\Psi(\omega, \mathbf{v}, t) = \omega^i + A\mathbf{v}^i t$, and a new event rate, $\lambda(\omega(t), \mathbf{v}) = \max\{0, \mathbf{v} \cdot A\nabla U(\omega + \mathbf{v}t)\}$. Upon events, the velocity is updated according to,

$$\mathbf{v}^{i+1} = \mathbf{v}^i - 2\frac{A\nabla U(\omega^{i+1}) \cdot \mathbf{v}^i}{\|A\nabla U(\omega^{i+1})\|^2} A\nabla U(\omega^{i+1}). \quad (4)$$

With careful choice of A, exploration along certain axis can be appropriately scaled. Pakman et al. [2017] propose a preconditioner similar to Li et al. [2015], though our preliminary experimentation found inconsistent results when applied to BNNs. Instead, we opt to build on the approach of Bertazzi and Bierkens [2020], where we use variance information of our samples to precondition our dynamics. We choose the preconditioner such that $A = \text{diag}\big(\Sigma^{\frac{1}{2}}\big)$, where Σ is the estimated covariance in our sample found during a warm-up period. As such, we refer to this sampler as the σBPS .

2.3 BOOMERANG SAMPLER

The Boomerang Sampler Bierkens et al. [2020] introduces non-linear dynamics for both parameter and velocity terms, and the inclusion of a Gaussian reference measure for the parameters and velocity $\mathcal{N}(\omega_\star, \Sigma_\star) \otimes \mathcal{N}(0, \Sigma_\star)$. The first term in this reference measure can be seen as a replacement for the prior in the joint probability over parameters and the second as the known distribution for the velocity component. The parameters $\omega_\star$ and $\Sigma_\star$ can be specified as traditional prior, or can be specified in an empirical approach where they are learnt from the data. Within this work, we will set $\omega_\star$ to the MAP estimate. In the original paper, $\Sigma_\star$ is set to the inverse of the Hessian, however, this can be computationally prohibitive for BNNs. Instead, we sum over the first order gradients at $\omega_\star$, and then compute the derivative with respect to this sum that is then inverted and scaled such that,

$$\Sigma_\star = \gamma\Big[\sum_{i=0}^{N-1} \nabla_\omega \sum_{j=0}^{P-1} \nabla_\omega p(\mathcal{D}_i|\omega_\star)_j\Big]^{-1} \quad (5)$$

where N is the number of mini-batches present, P is the number of parameters, subscript j indicates summation over parameter gradients in our model and γ is a hyperparameter to adjust the scale as needed. This can be seen as a weighted stochastic average to the inverse of a Hessian matrix diagonal.

Unlike the BPS samplers, the velocity does not remain constant between events. The dynamics of the Boomerang sampler for ω and $\mathbf{v}$ within events are given by $\Psi(\omega, \mathbf{v}, t)_\omega = \omega_\star - (\omega^i - \omega_\star)\cos(t) + \mathbf{v}^i \sin(t)$, $\Psi(\omega, \mathbf{v}, t)_\mathbf{v} = -(\omega^i - \omega_\star)\sin(t) + \mathbf{v}^i \cos(t)$, where the subscripts denote the parameter and velocity trajectory within the deterministic segment. The event rate is the same as the BPS, and the starting velocity for the next segment is updated upon events as,

$$\mathbf{v}^{i+1} = \mathbf{v}^i - 2\frac{\nabla U(\omega^{i+1}) \cdot \mathbf{v}^i}{\left\|\Sigma_\star^{\frac{1}{2}} \nabla U(\omega^{i+1})\right\|^2} \Sigma_\star \nabla U(\omega^{i+1}). \quad (6)$$

2.4 VELOCITY REFRESHMENT

All of the samplers introduced fail to target the posterior explicitly when using the above dynamics alone. Introduction of a refreshment step rectifies this, which is governed by a homogeneous PP $\tau_{ref} \sim \lambda(\lambda_{ref})$. When $\tau_{ref} < \tau$, the velocity is instead randomly sampled from the known reference distribution $\Phi(\mathbf{v})$, and τ_{ref} is used for the update event time. For BPS samplers in this work, we use a refreshment distribution of the form $\mathcal{N}(0, \sigma^2)$, where σ is a hyper-parameter to be set, and the Boomerang sampler requires $\Phi(\mathbf{v}) = \mathcal{N}(0, \Sigma_\star)$. A summary of PDMP algorithms for inference is described in Algorithm 1.

Algorithm 1: Application of PDMP samplers for Inference

Result: Samples from posterior distribution

while *Sampling* **do**

```
// Simulate event time
// event times in this work
   simulated with Algorithm 2
```

$\tau \sim \text{PP}(\lambda(\omega, \mathbf{v}))$;

```
// Simulate time of refresh event
```

$\tau_{\text{ref}} \sim \text{PP}(\lambda_{\text{ref}})$;

$\tau^i = \min(\tau, \tau_{\text{ref}})$;

```
// find end of current
   piecewise-deterministic
   segment, which will form start
   of next segment
```

$\omega^{i+1} = \Psi(\omega, \mathbf{v}, \tau^i)_\omega$;

if $\tau^i = \tau$ **then**

```
// update according kernel
```

$\mathbf{v}^{i+1} = R(\omega^{i+1}, \mathbf{v}^i)$;

else

```
// refresh velocity from known
   distribution
```

$\mathbf{v}^{i+1} \sim \Phi(\mathbf{v})$;

end

end

2.5 PROBLEMS WITH THE EVENT RATE

With the deterministic dynamics illustrated in these samplers, the main challenge in implementation of these methods is due to the sampling of the event times. Analytic sampling from IPP$\big(\lambda(t)\big)$ requires being able to invert the

integral of the event rate w.r.t. time,

$$\Lambda(t) = \int_0^\tau \lambda(t)dt = \int_0^\tau \max\{0, \mathbf{v} \cdot A\nabla U(\omega(\mathbf{v}, t)\}dt, \tag{7}$$

where $A = \mathbf{I}$ for the BPS and Boomerang samplers. Inverting the above integral is feasible only for simple models. A general case for sampling from IPPs is available through thinning Lewis and Shedler [1979]. This requires introducing an additional rate function that we can sample from $\mu(t)$ that is also a strict upper bound on the event rate function of interest such that $\mu(t) \geq \lambda(t)$ for all $t \geq 0$.

The efficiency of any thinning scheme relies on the tightness of the upper bound; the greater the difference between the upper bound and the true rate, the more likely a proposed time will be rejected when sampling. Pakman et al. [2017] propose a Bayesian linear regression method to generate an upper bound suitable for thinning, though require the calculation of variance within gradients to formulate a suitable upper bound. They calculate this variance empirically, which requires computing the gradient for each data point individually within a mini-batch. This computation prohibits use for BNNs where automatic differentiation software is used. Furthermore, the solution to the regression requires matrix inversion which can be numerically unstable without a strong prior, which limits its application for accelerating sampling within larger models. In the next section, we address this issue by instead introducing an interpolation-based scheme for creating efficient and adaptive approximate upper bounds that avoids excessive gradient computations and the numeric instability of matrix inversion.

3 ADAPTIVE BOUNDS FOR SAMPLERS

3.1 SAMPLING FROM IPPS WITH LINEAR EVENT RATES

Our goal is to create a piecewise-linear envelope $h(t)$ that will serve as an approximate upper bound of our true event rate, where each segment in $h(t)$ is represented by $a_i t + b_i$. This envelope will serve as the event rate for a proposal IPP that will be suitable for use with the thinning method of Lewis and Shedler [1979]. Acceptance of an event time t is given by,

$$U \leq \frac{\lambda(t)}{h(t)}, \tag{8}$$

where $U \sim \text{Uniform}[0, 1]$. We begin by building on the work of Klein and Roberts [1984] to demonstrate how to sample times from an IPP with a piecewise-linear event rate which we can use with thinning.

Within our proposal IPP with rate $h(t)$, we wish to generate the next event time t_i given the previous event t_{i-1}. The probability of events occuring within the range of $[t_{i-1}, t_i]$ is given by Devroye [2006],

$$F(x) = 1 - \exp\{-(\Lambda(t_i) - \Lambda(t_{i-1}))\} \tag{9}$$

We can solve this expression for t_i by,

$$t_i = \Lambda^{-1}(t_{i-1} - U) \tag{10}$$

where $U \sim$ Uniform[0,1]. For linear segments, the solution to this system can be written as Klein and Roberts [1984],

$$t_i = \big(- b_i + \sqrt{b_i^2 + a_i^2 t_{i-1}^2 + 2a_i b_i t_{i-1} \log(1 - U)}\big)/a_i. \tag{11}$$

This provides a framework for sampling from IPPs with a linear event rate. We now describe how we create a piecewise-linear envelope for a proposal process that can be used for thinning.

3.2 PIECEWISE INTERPOLATION FOR EVENT THINNING

We begin by introducing a modified event rate for which we will form our envelope,

$$\hat{\lambda}(\omega(t), \mathbf{v}) = \max\{0, \alpha\nabla U(\omega) \cdot \mathbf{v}^i\}, \tag{12}$$

where $\alpha \geq 1$ is a positive scaling factor to control the tightness of the approximate bound on the rate. The use of $\hat{\lambda}$ for creating our envelope is valid, since for values of $\alpha \geq 1$, $\hat{\lambda}(t) \geq \lambda(t)$. The scaling factor included in this event rate is designed to provide flexibility to end users with respect to computational time and bias that will be introduced during inference. The closer α is to one, the lower the probability for rejection of proposed event times, but the greater the probability that the generated event rate will not be a strict upper bound.

Our goal is to create a piecewise-linear upper bound suitable for proposing event times using Equation 11. To achieve this we have two growing sets, one for proposed event times $T = \{t_0, ..., t_n\}$ and the value of the adjusted event rates at these times $L = \{\hat{\lambda}(t_0), \ldots, \hat{\lambda}(t_n)\}$ for which we can create a set of functions,

$$h(t) = a_i t + b, \qquad t \geq t_i. \tag{13}$$

The values for a_i and b_i are found by interpolating between the points $(t_{i-1}, \hat{\lambda}(t_{i-1}))$ and $(t_i, \hat{\lambda}(t_i))$.

At the beginning of every deterministic PDMP segment, the sets T and L will be empty. To initialise the sets and create our first linear segment, we evaluate the event rate at two points, $t_0 = 0$ and $t = t_{init}$, where $t_{init} > t_0$. To evaluate the values for a_0 and b_0, we interpolate between these two segments. Once the values for the first linear segment are found, t_0 and $\hat{\lambda}(t_0)$ are appended to their corresponding sets, and t_{init} and $\hat{\lambda}(t_{init})$ are discarded. With this initial

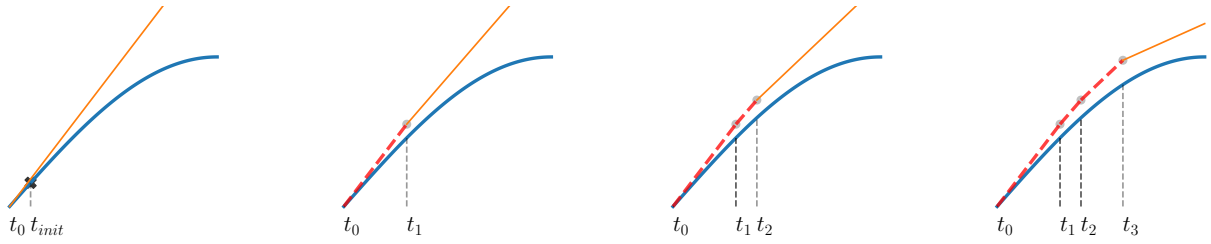


Figure 2: Example of the progression of the proposed envelope scheme used for thinning. The blue line represents the true event rate, orange section depicts the active regions for which we sample a new proposal time, and the red section depicts previous segments in the envelope. Starting from the left, an initial segment is found through interpolation between time points t_0 and t_{init}. In the next segment, active regions of the envelope are found by interpolating between the two prior points, which extends to create a new segment to propose times. This process continues until a proposed time is accepted from thinning.

linear segment, we can propose a time t_i through Equation 11. This proposed time is either accepted or rejected from Equation 8.

If the proposed time is accepted, then the dynamics of the PDMP sampler are updated at the given event time and the sets T and L are cleared, ready to be re-initialised for the new dynamics. If the time is rejected, the proposed time t_i and envelope evaluation $\hat{\lambda}(t_i)$ are appended to their respective sets, and a new linear segment is calculated to interpolate between this rejected proposal and the previous elements in the sets T and L. The rejected proposal time will serve as the new starting point (t_{i-1}) for the new linear segment to propose the next time using Equation 11. This will continue until the proposed event time is accepted. This process depicted visually in Figure 2 and summarised in Algorithm 2.

Within this work, we limit ourselves to models where the envelope provided by $h(t)$ will only be an approximate upper bound, meaning bias will likely be introduced during inference. Diagnosis and correction of this can be identified through the acceptance ratio $\lambda(t)/h(t)$; if this value is greater than one, the condition of $h(t)$ being a local upper bound is violated. The amount of potential bias introduced can be mitigated by increasing the scaling factor α in Equation 12 at the expense of increasing computation load. This property is investigated in Supp. Material A. In the following sections, we evaluate the proposed event thinning scheme for BNNs to identify the suitability of different samplers for inference in these challenging models, and how they can outperform other stochastic approximation methods in terms of calibration, posterior exploration, sampling efficiency and predictive performance.

Algorithm 2: Sampling event rate using proposed adaptive thinning method.

Result: Proposed PDMP Event Time τ
Initialize T, L;
Evaluate $(0, \lambda(0)), (t_{init}, \lambda(t_{init}))$;
$i = 1$;
Compute a_i, b_i;
$T_0 \leftarrow 0, L_0 \leftarrow \lambda(t)$;
Discard $t_{init}, \lambda(t_{init})$;
while *not accepted* **do**
  `// propose event time with` t_{i-1}, a_i `and` b_i
  $t_i \sim PP\Big(h(t)\Big)$;
  $u \sim$ Uniform$[0, 1]$;
  if $u \leq \lambda(t_i)/h(t)$ **then**
    `// sample is accepted`
    $\tau = t_i$;
    accepted = True;
  end
  else
    `// increment counter`
    $i+ = 1$;
    $T_i \leftarrow t_i, L_i \leftarrow \lambda(t_i)$;
    `// update linear segment`
    a_i, b_i = update(L, T);
  end
end

4 RELATED WORK

The samplers used within this work require the use of an additional reference process to provide velocity refreshments. The Generalised BPS Wu and Robert [2017] is an updated variant of the BPS algorithm that incorporates a stochas-

tic update of the velocity which alleviates the need for a refreshment process. Simulations have shown comparable performance to the BPS for simple models and how it can reduce the need for fine-tuning the reference parameter τ_{ref}.

Another prominent sampler is the Zig-Zag Process (ZZP) [Bierkens et al., 2019], where at events the dynamics of a single parameter are updated. For the one-dimensional case, this sampler represents the same process as the BPS. This sampler has shown favourable results in terms of mixing performance and can achieve ergodicity for certain models where the BPS cannot. A key characteristic of this method is that each parameter is assigned an individual event rate, making implementation for high-dimensional BNN models challenging.

Another class of algorithms designed for subsampling are discrete stochastic MCMC methods Wenzel et al. [2020], Chen et al. [2014], Ma et al. [2015], Welling and Teh [2011], Li et al. [2015]. These models have shown favourable performance, with a recent variant achieving comparable predictive accuracy on the ImageNet data set Heek and Kalchbrenner [2019]. Compared to algorithms related to PDMPs, it has been shown that high variance related to naive subsampling limits these methods to provide only an approximation to the posterior Betancourt [2015]. The bias that is introduced due to subsampling can be controlled by reducing the step-size for these methods at the expense of mixing performance and posterior explorationNagapetyan et al. [2017], Brosse et al. [2018], Teh et al. [2016]. We investigate the effect of this property for SGLD and compare performance with PDMP samplers in the following section.

5 EXPERIMENTS

We now validate the performance of PDMPs using the proposed event sampling method on a number of synthetic and real-world data sets for regression and classification. To analyse performance for predictive tasks, the predictive posterior needs to be evaluated. In this work, we discretise samples from the trajectory to allow for Monte Carlo integration,

$$p(y^*|x^*, \mathcal{D}) = \int \pi(\omega) p(y^*|\omega, x^*) d\omega \approx \frac{1}{N} \sum_{i=1}^{N} p(y^*|\omega_i, x^*) \quad \omega_i \sim \pi(\omega), \quad (14)$$

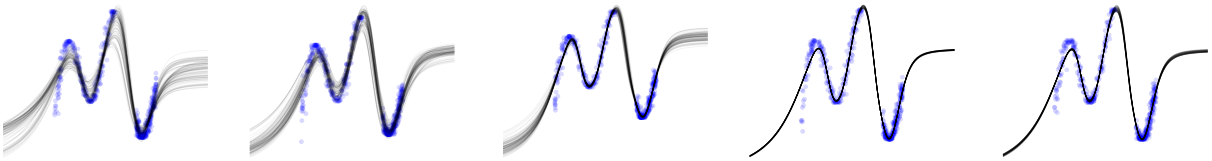
where parameter samples of $\omega^{(i)}$ are taken from the values encountered at event times. Experimentation is first conducted on synthetic data sets to allow us to easily visualise predictive performance and uncertainty in our models, followed by more difficult classification tasks with Bayesian Convolutional Neural Networks (CNNs) on real data sets. For all experimentation, we set our scaling factor from Equation 12 to $\alpha = 1.0$ to promote computational efficiency. To enable these experiments, we deliver a Python package titled Tensorflow PDMP (TPDMP). This package utilises the Tensorflow Probability library Dillon et al. [2017], allowing for hardware acceleration and graph construction of all our models to accelerate computation. We deliver kernels to implement the BPS, σBPS , and Boomerang sampler with our proposed event thinning scheme. Code is available at https://github.com/egstatsml/tpdmp.git.

5.1 REGRESSION AND BINARY CLASSIFICATION WITH SYNTHETIC DATA

To visualise predictive performance and uncertainty estimation, regression and binary classification tasks are formed on synthetic data sets. Description of the networks used for these tasks is described in Supp. Material E. Before sampling, a MAP estimate was first found using stochastic optimisation, and was used to initialise each sampler. 2,000 samples were generated using each sampling method, with each sampler initialised from the same MAP estimate. The σBPS requires an additional warmup period to identify suitable values for the preconditioner. We achieve this by performing 1,000 initial samples using the BPS, and standard deviation parameters used for the preconditioner are estimated from these samples using the Welford algorithm Welford [1962]. These preconditioner values are then fixed throughout the sampling process. For the Boomerang Sampler, the preconditioner scaling factor from Equation 5 is set to $\gamma = 500.0$. The PDMP methods are compared against SGLD which starts with a learning rate that decays to zero as required Welling and Teh [2011], Nagapetyan et al. [2017], and with no decay of the learning rate as is commonly done in practice (SGLD-ND). Examples of the predictive posterior distribution for regression and binary classification are shown in Figures 3 and 4 respectively, with full analysis in Supp. Material B. All PDMP models are fit with the proposed adaptive event thinning procedure.

Results from these experiments affirm that inference from the PDMP models is suitable for predictive reasoning, with low variance seen within the range of observed data and greater variance as distance from observed samples increases. We similarly see an increase in uncertainty along the decision boundary, which is a desireable property. This is in contrast to SGLD, which is unable to offer suitable predictive uncertainty, even in the case for larger non-decreasing learning rates. This highlights the known limitations of SGLD, that with a decaying learning rate it can fail to explore the posterior, and with a larger non-decreasing learning rate will converge to dynamics offered by traditional SGD Brosse et al. [2018], Nagapetyan et al. [2017].

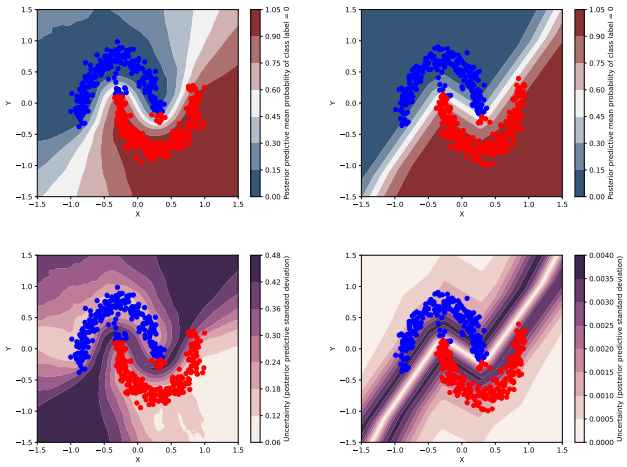
These tests indicate promising performance in terms of predictive accuracy and uncertainty estimates. To further demonstrate classification performance, we move to larger and more complicated models for performing classification



(a) BPS (b) σBPS (c) Boomerang (d) SGLD (e) SGLD-ND

Figure 3: Examples of the different PDMP samplers using the proposed event thinning procedure on synthetic regression task compared against SGLD with decaying learning rate and constant learning rate (SGLD-ND).

on real-world data sets.



(a) BPS (b) SGLD

Figure 4: Examples of the predictive mean and variance for synthetic classification task. Left column illustrates results using the BPS and the right using SGLD. We see increased uncertainty for the BPS around the decision boundary, whilst SGLD shows greater certainty.

5.1.1 UCI-Datasets

We further evaluate the performance of the PDMP samplers enabled by the proposed event sampling scheme on datasets from the UCI repository Newman et al. [1998]. In Table 1, we show performance metrics on the Boston houses dataset, with the Naval, Energy, Yacht, and Concrete datasets evaluated in Supp. Material E.2. Each model is fit with 2,000 samples. For these experiments, we further include the naive Stochastic Gradient Hamiltonian Monte Carlo (SGHMC) Chen et al. [2014]. Predictive performance of these models is measured with Mean Squared Error (MSE) and Negative Log-Likelihood (NLL). Sampling efficiency is evaluated with Effective Sample Size (ESS) Robert et al. [1999]. Due to the high dimension of our models, we perform PCA on returned samples and project them onto the first principal component to report ESS on the direction of greatest variance within samples.

Table 1: Summary of predictive performance using PDMP samplers with the proposed event time sampling methods on the Boston Houses dataset. Negative log-likelihood (NLL) and Mean Squared Error (MSE) are reported. Effective sample size (ESS) is measured over the first principal component of samples. Results are shown over 5 independent runs with standard deviations reported.

Inference	**NLL** $\downarrow$	**MSE** $\downarrow$	**ESS** $\uparrow$
BPS	51.26 ± 0.19	3.81 ± 0.08	2.73 ± 0.03
σBPS	51.14 ± 0.10	3.76 ± 0.05	2.74 ± 0.05
Boomerang	51.40 ± 0.32	3.87 ± 0.14	$\mathbf{1974.73 \pm 34.83}$
SGLD-ND	$\mathbf{51.07 \pm 0.00}$	$\mathbf{3.73 \pm 0.00}$	2.87 ± 0.00
SGHMC	51.08 ± 0.08	3.74 ± 0.03	2.72 ± 0.01

From these results, we see SGLD and SGHMC provide a slight improvement in terms of MSE and NLL, though we see that the Boomerang Sampler considerably outperforms both of these methods in terms of sample efficiency. This result follows from the previous sections where we see that SGLD frequently converges to the SGD solution space, whilst the PDMP samplers can explore the posterior space. Additional results in Supp. Material E.2 further validate these results.

5.2 MULTI-CLASS CLASSIFICATION

We now evaluate the performance of the proposed sampling procedures on the popular MNIST LeCun et al. [1998], Fashion MNIST Xiao et al. [2017], SVHN Netzer et al. [2011], CIFAR-10 and CIFAR-100 Krizhevsky and Hinton [2009] data sets using CNNs. For MNIST and Fashion-MNIST, the LeNet5 architecture was used whilst for SVHN, CIFAR-10, and CIFAR-100 the modified ResNet20 architecture from Wenzel et al. [2020] was used. Each parameter was again supplied a standard normal prior.

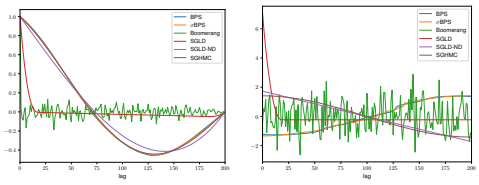
Similar to the experiments on regression, a MAP estimate is found and used to initialise each sampler. 2,000 samples for each model are then generated, though a thinning factor of 10 is used to reduce the number of returned samples used for

Table 2: Summary of predictive performance using PDMP samplers with the proposed event time sampling methods. Negative log-likelihood (NLL) is reported, along with calibration measured using the expected calibration error (ECE) Guo et al. [2017]. Effective sample size (ESS) is measured over the first principal component of samples. Mean and standard deviation in results presented over 5 independent runs.

Inference	ACC ↑	NLL ↓	ECE ↓	ESS ↑
		MNIST		
BPS	**0.99 ± 0.01**	62.63 ± 5.60	1.05 ± 0.12	2.70 ± 0.03
σBPS	0.97 ± 0.03	51.72 ± 10.85	0.88 ± 0.41	2.74 ± 0.04
Boomerang	**0.99 ± 0.00**	0.18 ± 0.05	**0.02 ± 0.00**	**138.17 ± 47.08**
SGLD	**0.99 ± 0.00**	6.10 ± 0.00	0.09 ± 0.00	19.88 ± 0.02
SGLD-ND	**0.99 ± 0.00**	77.05 ± 0.00	1.52 ± 0.00	3.40 ± 0.00
SGHMC	**0.99 ± 0.00**	**0.14 ± 0.02**	**0.02 ± 0.00**	2.71 ± 0.00
		Fashon-MNIST		
BPS	**0.91 ± 0.00**	16.79 ± 1.65	0.44 ± 0.02	2.74 ± 0.02
σBPS	0.90 ± 0.00	**3.43 ± 1.00**	0.32 ± 0.02	2.79 ± 0.12
Boomerang	**0.91 ± 0.00**	3.82 ± 0.29	**0.31 ± 0.01**	**200.00 ± 0.00**
SGLD	**0.91 ± 0.00**	5.53 ± 0.00	0.30 ± 0.00	19.85 ± 0.02
SGLD-ND	**0.91 ± 0.00**	69.17 ± 0.01	1.58 ± 0.00	3.59 ± 0.00
SGHMC	**0.91 ± 0.00**	4.63 ± 0.13	0.34 ± 0.00	2.71 ± 0.00
		SVHN		
BPS	0.95 ± 0.00	35.35 ± 5.91	0.61 ± 0.10	2.69 ± 0.02
σBPS	0.95 ± 0.00	**0.36 ± 0.11**	**0.19 ± 0.00**	2.74 ± 0.05
Boomerang	0.95 ± 0.00	0.50 ± 0.07	**0.19 ± 0.00**	**186.33 ± 21.10**
SGLD	0.95 ± 0.00	7.01 ± 10.12	0.24 ± 0.10	16.61 ± 6.44
SGLD-ND	**0.96 ± 0.00**	27.32 ± 0.08	0.44 ± 0.00	3.73 ± 0.00
SGHMC	0.95 ± 0.00	0.47 ± 0.05	**0.19 ± 0.00**	2.71 ± 0.00
		CIFAR-10		
BPS	0.79 ± 0.01	42.03 ± 2.42	1.18 ± 0.07	2.82 ± 0.10
σBPS	0.79 ± 0.00	**5.93 ± 6.43**	0.70 ± 0.05	2.75 ± 0.08
Boomerang	0.81 ± 0.00	6.71 ± 2.20	**0.64 ± 0.06**	**200.00 ± 0.00**
SGLD	0.81 ± 0.00	13.31 ± 0.01	0.85 ± 0.00	19.83 ± 0.00
SGLD-ND	**0.82 ± 0.00**	31.12 ± 0.13	0.89 ± 0.00	4.04 ± 0.00
SGHMC	0.80 ± 0.00	13.84 ± 0.27	0.92 ± 0.02	2.71 ± 0.00
		CIFAR-100		
BPS	0.57 ± 0.01	42.45 ± 1.63	2.48 ± 0.09	2.69 ± 0.02
σBPS	0.63 ± 0.00	8.27 ± 0.37	1.39 ± 0.00	2.78 ± 0.08
Boomerang	**0.64 ± 0.00**	**6.85 ± 0.86**	**1.35 ± 0.01**	**162.21 ± 43.74**
SGLD	**0.64 ± 0.00**	12.40 ± 0.07	1.45 ± 0.00	20.40 ± 0.07
SGLD-ND	**0.64 ± 0.00**	11.10 ± 0.05	1.42 ± 0.00	2.83 ± 0.00
SGHMC	**0.64 ± 0.00**	12.34 ± 0.10	1.45 ± 0.00	2.71 ± 0.00

prediction to 200. For these models, we measure predictive performance and calibration through the Accuracy, NLL, and Expected Calibration Error (ECE) Guo et al. [2017], and similarly measure sampling efficiency using the ESS with on samples after performing PCA. A full description of the models used, and experiment parameters is shown in Supp. Material E.3. Table 2 summarises the results of these experiments.

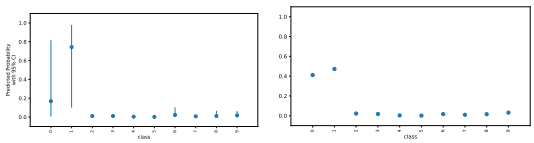
These results highlight favourable performance for certain samplers. The BPS sampler is unable to provide calibrated predictions, whilst the σBPS and Boomerang samplers consistently provide calibrated predictive performance and reduction in NLL. Most importantly, we note the Boomerang



(a) ACF first principal component (b) Trace first principal component

Figure 5: Example of ACF and trace plots for first principal component of the samples from network fit on SVHN dataset.

(a) Image Class 0

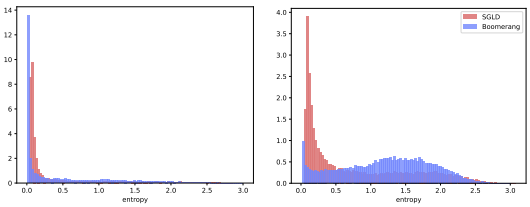


(b) Boomerang Sampler (c) SGLD

Figure 6: Examples from predictive posterior for difficult-to-classify samples from SVHN. Top row shows the original image and the bottom row shows the predictive distribution for the Boomerang sampler and SGLD. The mean for each class represented by the dot, and the 95% credible intervals shown with the error bars.

sampler consistently outperforms other samplers in terms of effective sample size, whilst also promoting competitive or improved predictive accuracy. This highlights the potential for the Boomerang sampler for probabilistic inference within neural networks.

With measures of predictive performance and ESS within our models, we wish to further investigate the mixing properties of the samplers presented within to identify how well the posterior space is being explored. ESS only gives a measure to approximate the number of independent samples within our MCMC chain, though we are also interested in how well the support for the posterior is being explored. Given the large number of parameters seen within a BNN, it is infeasible to evaluate the coordinate trace and autocorrelation plots for individual parameters as is typically done for MCMC models. Instead, we again perform PCA to reduce the dimension of our data and investigate the trace plots of the first principal component as illustrated in Figure. 5. From these figures, we can identify strong correlation between samples from the BPS, σBPS , SGHMC, and SGLD-ND solutions. SGLD offers reduced correlation in samples, how-



(a) In-distribution (b) OOD

Figure 7: Entropy histograms comparing SGLD and Boomerang sampler fit on the CIFAR-10 dataset. OOD data represented by SVHN. We see the predictive entropy from the Boomerang sampler increases as desired for OOD data, whilst SGLD remains overly confident for erroneous samples.

ever as seen in the trace plot, samples fail to explore the posterior and instead converge to a steady state, whilst the Boomerang sampler provides considerably reduced correlation and more favourable mixing. Convergence of the SGLD samples can be attributed to the reduction in learning rate required to target the posterior. We verify this result in Supp. Material C, where we provide further analysis into results from all networks and remaining principal components. The effect of this convergence in terms of predictive uncertainty is illustrated within Figure 6, where the PDMP sampler is able to provide more meaningful uncertainty estimates for difficult-to-classify samples, and the SGLD predictive results converge to that similar of a point estimate. Additional examples of the predictive distributions is shown in Supp. Material H.

Probabilistic methods have shown favourable performance in terms of Out of Distribution (OOD) detection Grathwohl et al. [2019], Maddox et al. [2019]. Given the point-estimate-like nature of the results returned by SGLD, we wish to compare with results from the Boomerang sampler to see if both can offer similar performance for OOD data. We see in Figure 7 that the Boomerang sampler offers greater entropy for OOD data, indicating a desireable increase in aleatoric uncertainty. Additional analysis is provided in Supp. Material G. Given the consistent predictive performance, quality of uncertainty estimates, and posterior exploration, we would recommend researchers wishing to apply MCMC methods for BNNs consider the use of the Boomerang sampler.

6 DISCUSSION AND LIMITATIONS

Whilst the PDMP methods have shown favourable performance in terms of predictive accuracy, calibration and uncertainty in BNNs, there are certain challenges with fitting them. The PDMP samplers used within this work are designed to target the joint distribution,

$$p(\omega, \mathbf{v}) = \pi(\omega)\Phi(\mathbf{v}) \tag{15}$$

where $\pi(\omega)$ is the target posterior and $\Phi(\mathbf{v})$ is the distribution of the auxiliary velocity components which must be set by users in the form of the refreshment distribution. For the BPS and σBPS samplers, it has been shown that with a reference distribution may be a Gaussian or restricted to the unit hypersphere Bouchard-Côté et al. [2018]. For the Boomerang sampler, the velocity distribution is designed with respect to a reference measure to ensure invariance to the target distribution, such that $\Phi(\mathbf{v}) = \mathcal{N}(0, \Sigma_\star)$, where $\Sigma_\star$ is the same factor used to precondition the dynamics. The choice in distribution used for the velocity component has an explicit effect on the mixing capabilities of the models when applied to BNNs. We demonstrate this effect in Supp. Material D.1. We find that a velocity distribution with too much variance can cause effects similar to that of divergences seen in HMC and NUTS. Furthermore, we see that with variance set too low, the samplers can fail to explore the posterior sufficiently to provide the desired meaningful uncertainty estimates. A similar effect can be seen for the choice of refreshment rate, which we investigate in Supp. Material D.2. We highlight these limitations as areas for future research to enable robust application of PDMP methods for BNNs.

The Boomerang sampler as implemented within this work and the original paper is probabilistic, though is not purely Bayesian. This is due to the reference measure for the velocity being identified through the data itself. A strictly Bayesian approach can be recovered by setting the reference measure and associated preconditioner matrix from a prior distribution, though we would lose some favourable sampling performance offered by this sampler. We can view the approach implemented within similar to an empirical Bayes, where we are gleaning information about the prior (reference measure for the Boomerang sampler), from the data itself. Given the difficulty of specifying a meaningful and informative prior, and the success seen when using emperical priors for BNNs Krishnan et al. [2020], we believe the use of such an approach for the Boomerang sampler is justified.

7 CONCLUSION

Within this work, we demonstrate how PDMPs can be used for BNNs. We provide a flexible piecewise linear bound to enable sampling of event times within these frameworks that permits inference in BNNs. A GPU-accelerated software package is offered to increase the availability of PDMPs for a wide array of models. Experimentation on BNNs for regression and classification indicates comparable or improved predictive performance and calibration, though were able to consistently improve sampling efficiency and uncertainty estimation when compared against existing stochastic inference methods.

References

David Barber and Christopher M Bishop. Ensemble learning in bayesian neural networks. *Nato ASI Series F Computer and Systems Sciences*, 168:215–238, 1998.

Andrea Bertazzi and Joris Bierkens. Adaptive schemes for piecewise deterministic monte carlo algorithms. *arXiv preprint arXiv:2012.13924*, 2020.

Michael Betancourt. The fundamental incompatibility of scalable hamiltonian monte carlo and naive data subsampling. In *International Conference on Machine Learning*, ICML'15, pages 533–540. JMLR.org, 2015.

Joris Bierkens, Paul Fearnhead, Gareth Roberts, et al. The zig-zag process and super-efficient sampling for bayesian analysis of big data. *The Annals of Statistics*, 47(3):1288–1320, 2019.

Joris Bierkens, Sebastiano Grazzi, Kengo Kamatani, and Gareth Roberts. The boomerang sampler. *arXiv preprint arXiv:2006.13777*, 2020.

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

Alexandre Bouchard-Côté, Sebastian J Vollmer, and Arnaud Doucet. The bouncy particle sampler: A nonreversible rejection-free markov chain monte carlo method. *Journal of the American Statistical Association*, 113(522):855–867, 2018.

Nicolas Brosse, Alain Durmus, and Eric Moulines. The promises and pitfalls of stochastic gradient langevin dynamics. In *Advances in Neural Information Processing Systems*, pages 8268–8278, 2018.

Tianqi Chen, Emily Fox, and Carlos Guestrin. Stochastic gradient hamiltonian monte carlo. In *International conference on machine learning*, pages 1683–1691, 2014.

Mark HA Davis. Piecewise-deterministic markov processes: a general class of non-diffusion stochastic models. *Journal of the Royal Statistical Society: Series B (Methodological)*, 46(3):353–376, 1984.

Mark HA Davis. *Markov models & optimization*, volume 49. CRC Press, 1993.

Luc Devroye. Nonuniform random variate generation. *Handbooks in operations research and management science*, 13:83–121, 2006.

Joshua V Dillon, Ian Langmore, Dustin Tran, Eugene Brevdo, Srinivas Vasudevan, Dave Moore, Brian Patton, Alex Alemi, Matt Hoffman, and Rif A Saurous. Tensorflow distributions. *arXiv preprint arXiv:1711.10604*, abs/1711.10604, 2017.

Paul Fearnhead, Joris Bierkens, Murray Pollock, Gareth O Roberts, et al. Piecewise deterministic markov processes for continuous-time monte carlo. *Statistical Science*, 33 (3):386–412, 2018.

Will Grathwohl, Kuan-Chieh Wang, Joern-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. In *International Conference on Learning Representations*, 2019.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR, 2017.

Jonathan Heek and Nal Kalchbrenner. Bayesian inference for large scale image classification. *arXiv preprint arXiv:1908.03491*, 2019.

Pavel Izmailov, Sharad Vikram, Matthew D Hoffman, and Andrew Gordon Wilson. What are bayesian neural network posteriors really like? *arXiv preprint arXiv:2104.14421*, 2021.

Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2): 183–233, 1999.

Robert W. Klein and Stephen D. Roberts. A time-varying poisson arrival process generator. *Simulation*, 43:193–195, 1984.

Ranganath Krishnan, Mahesh Subedar, and Omesh Tickoo. Specifying weight priors in bayesian deep neural networks with empirical bayes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 4477–4484, 2020. URL `https://ojs.aaai.org/index.php/AAAI/article/view/5875`.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

PA W Lewis and Gerald S Shedler. Simulation of nonhomogeneous poisson processes by thinning. *Naval research logistics quarterly*, 26(3):403–413, 1979.

Chunyuan Li, Changyou Chen, David Carlson, and Lawrence Carin. Preconditioned stochastic gradient langevin dynamics for deep neural networks. *arXiv preprint arXiv:1512.07666*, 2015.

Yi-An Ma, Tianqi Chen, and Emily Fox. A complete recipe for stochastic gradient mcmc. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2917–2925. Curran Associates, Inc., 2015. URL http://papers.nips.cc/paper/5891-a-complete-recipe-for-stochastic-gradient-mcmc.pdf.

David JC MacKay. Probable networks and plausible predictions–a review of practical bayesian methods for supervised neural networks. *Network: computation in neural systems*, 6(3):469–505, 1995.

Wesley J. Maddox, Pavel Izmailov, Timur Garipov, Dmitry P. Vetrov, and Andrew Gordon Wilson. A simple baseline for bayesian uncertainty in deep learning. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13132–13143, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/118921efba23fc329e6560b27861f0c2-Abstract.html.

Stephan Mandt, Matthew D Hoffman, and David M Blei. Stochastic gradient descent as approximate bayesian inference. *Journal of Machine Learning Research*, 18:1–35, 2017.

Tigran Nagapetyan, Andrew B Duncan, Leonard Hasenclever, Sebastian J Vollmer, Lukasz Szpruch, and Konstantinos Zygalakis. The true cost of stochastic gradient langevin dynamics. *arXiv preprint arXiv:1706.02692*, 2017.

Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, page 5, 2011.

D.J. Newman, S. Hettich, C.L. Blake, and C.J. Merz. Uci repository of machine learning databases, 1998. URL http://www.ics.uci.edu/~mlearn/MLRepository.html.

Ari Pakman, Dar Gilboa, David Carlson, and Liam Paninski. Stochastic bouncy particle sampler. In *International Conference on Machine Learning*, pages 2741–2750, 2017.

Christian P Robert, George Casella, and George Casella. *Monte Carlo statistical methods*, volume 2. Springer, 1999.

Yee Whye Teh, Alexandre H Thiery, and Sebastian J Vollmer. Consistency and fluctuations for stochastic gradient langevin dynamics. *Journal of Machine Learning Research*, 17, 2016.

Paul Vanetti, Alexandre Bouchard-Côté, George Deligiannidis, and Arnaud Doucet. Piecewise-deterministic markov chain monte carlo. *arXiv preprint arXiv:1707.05296*, 2017.

Martin J Wainwright and Michael Irwin Jordan. *Graphical models, exponential families, and variational inference*, volume 1. Now Publishers Inc, 2008.

BP Welford. Note on a method for calculating corrected sums of squares and products. *Technometrics*, 4(3):419–420, 1962.

Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688, 2011.

Florian Wenzel, Kevin Roth, Bastiaan S. Veeling, Jakub Swiatkowski, Linh Tran, Stephan Mandt, Jasper Snoek, Tim Salimans, Rodolphe Jenatton, and Sebastian Nowozin. How good is the bayes posterior in deep neural networks really? In *International Conference of Machine Learning*, 2020.

Changye Wu and Christian P Robert. Generalized bouncy particle sampler. *arXiv preprint arXiv:1706.04781*, art. arXiv:1706.04781, Jun 2017.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.