Abstract

Markov Chain Monte Carlo (MCMC) algorithms do not scale well for large datasets leading to difficulties in Neural Network posterior sampling. In this paper, we propose Penalty Bayesian Neural Networks - PBNNs, as a new algorithm that allows the evaluation of the likelihood using subsampled batch data (mini-batches) in a Bayesian inference context towards addressing scalability. PBNN avoids the biases inherent in other naive subsampling techniques by incorporating a penalty term as part of a generalization of the Metropolis Hastings algorithm. We show that it is straightforward to integrate PBNN with existing MCMC frameworks, as the variance of the loss function merely reduces the acceptance probability. By comparing with alternative sampling strategies on both synthetic data and the MNIST dataset, we demonstrate that PBNN achieves good predictive performance even for small mini-batch sizes of data. We show that PBNN provides a novel approach for calibrating the predictive distribution by varying the mini-batch size, significantly reducing predictive overconfidence.

1. INTRODUCTION

The development of robust and reliable Deep Neural Networks (DNNs) has led to a significant interest in the application of Uncertainty Quantification (UQ) techniques. UQ is essential for assessing the confidence in predictions made by DNNs, especially in critical applications such as autonomous driving and medical diagnosis. Despite its importance, designing effective UQ methods that accurately compute the predictive distribution by marginalizing over DNN parameters remains a challenge (Gawlikowski et al., 2023).

Bayesian inference methods provide a principled approach to obtain the posterior of network parameters. These methods, which include Variational inference and Markov Chain Monte Carlo (MCMC) sampling, the latter regarded as being the gold standard (Izmailov et al., 2021), face significant scalability issues. Specifically, MCMC’s requirement to evaluate the log-likelihood over the entire dataset at each iteration makes it impractical for large systems and datasets. This challenge hinders the practical application of such posterior sampling techniques in Bayesian Neural Networks (BNNs). Additionally, uninformative priors are commonly used to prevent overconfidence (overfitting), adversely affecting the predictive performance of BNNs. A current field of research is dedicated to developing BNN-specific priors as reviewed by Fortuin (2022).

Towards addressing these challenges, we propose a variant of MCMC which we call Penalty Bayesian Neural Network (PBNN). PBNN leverages a novel approach of using subsampled batch data (mini-batches) while incorporating a “penalty method” for BNN posterior sampling. This strategy significantly reduces predictive
overconfidence and avoids the biases inherent in naive subsampling techniques. PBNN adapts the penalty method (Ceperley and Dewing, 1999), originally developed in the context of statistical and computational physics, e.g. in the work of Pierleoni et al. (2004), to efficiently sample distributions with loss functions affected by statistical noise. The use of this method in PBNN enables unbiased posterior sampling by explicitly accounting for the variance of the loss associated with subsampled data batches (mini-batches).

The issue of introduction of strong bias in the posterior sampling process when done naively has been explored in previous work in the context of Bayesian inference resulting in the proposal of several methodologies addressing scalability (Bardenet et al., 2017; Korattikara et al., 2014; Seita et al., 2018). PBNN distinguishes itself by ensuring unbiased sampling of the posterior for any given mini-batch size under the assumption that the distribution generated by the mini-batches is normal and of known variance. Although the later condition might be slightly violated in practice, as the variance must be estimated itself, our approach represents a departure from other methods such as SGMCMC (HMC and Langevin) and Barker acceptance test that attempt to control bias post-sampling and reducing it below a threshold.

The remainder of the article is structured as follows: we begin with an overview of fundamental concepts and intuitions underlying the sampling of posterior distributions through Bayesian inference, we then delve into some related works, including a review of the Stochastic Gradient Langevin Dynamic (SGLD) algorithm (Welling and Teh, 2011) and its implications for noise in loss computation. Subsequently, we introduce PBNN as a novel, unbiased posterior sampling strategy and show its benefits focusing on its ability to effectively mitigate the predictive overconfidence effects.

We proceed to discuss some practical details on evaluating the noisy loss, and its uncertainty, highlighting the compatibility of PBNN with state of the art MCMC proposal distribution for the Markov chain such as Langevin samplers. We demonstrate PBNN’s predictive performance by testing it on a learning task designed to trigger overfitting. This demonstration not only highlights PBNN’s effectiveness but also introduces the mini-batch size as an efficient calibration parameter. We conclude by discussing the impact of mini-batch size, as a parameter, on acceptance rates and overall performance.

2. BACKGROUND

In the following, we consider a vector $\theta$ that describes the parameters of a model, representing the weights and biases of a neural network hereafter. We define $p(\theta)$ as a prior distribution over this set of parameters. Commonly used priors are Gaussian and Laplace that correspond respectively to L2 and L1 regularizations. We refer to $p(y|x, \theta)$ as the probability of a target $y$ given a data input $x$ and a parameter vector $\theta$. The uncertainty over the parameters $\theta$ given a collection of observed data $D$ is captured by the posterior distribution which is defined as:

$$p(\theta|D) = \frac{p(\theta) \prod_{i=1}^{N} p(y_i|x_i, \theta)}{p(D)}$$

(1)
where the observations in the data set $D = \{(y_i, x_i)\}_{i=1}^N$ are iid. Up to a constant, $p(\theta|D) \propto e^{-L_N(\theta)}$ where the loss $L_N(\theta)$ is the negative log of the posterior:

$$L_N(\theta) = -\log p(\theta) - \frac{N}{N} \sum_{i=1}^N \log p(y_i|x_i, \theta)$$

and the last term is the negative log-likelihood. This choice of the loss is only illustrative and does not reduce the generality of PBNN as we could have also considered an unsupervised setup where $D = \{x_i\}_{i=1}^N$ and $L_N(\theta) = -\log p(\theta) - \sum_{i=1}^N \log p(x_i|\theta)$.

As an example of a common loss, using a one-dimensional homoscedastic Gaussian likelihood in a regression task and a Gaussian prior over the model’s parameters, we obtain the familiar squared-error loss function:

$$L_N(\theta) \propto \lambda \|\theta\|^2 + \sum_{i=1}^N (y_i - f_\theta(x))^2 + \text{cst}$$

where $f_\theta(x)$ is the prediction of the neural network and $\lambda$ tunes the strength of the L2 regularization.

The parameters $\theta$ of the network are usually optimized through Stochastic Gradient Descent (SGD), as defined below:

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta L_n(\theta_t)$$

$\eta \in \mathbb{R}^+$ is called the learning rate. Stochastic gradient descent algorithms aim at maximizing the posterior distribution (MAP) using mini-batches of data from the whole dataset $D$. The new stochastic loss $L_n(\theta_t)$ is defined as:

$$L_n(\theta) = -\log p(\theta) - \frac{n}{N} \sum_{i=1}^n \log p(y_i|x_i, \theta)$$

where $n \leq N$ corresponds to the size of the mini-batch. This mini-batch contains data sampled without replacement from dataset $D$ such that by definition $\langle L_n(\theta) \rangle = L_N(\theta)$.

In order to capture the epistemic uncertainty of $\theta$ given $D$, we want to obtain samples $\theta$ that are distributed according to Equation (1) instead of a point estimate $\theta_{t \to \infty}$ that minimizes the loss. In the special case of deep learning, the class of models and techniques that target this posterior distribution are called Bayesian Neural Networks (BNNs).

As standard in Bayesian inference, the prediction of a BNN corresponds to a distribution of possible values $y$ given an input $x$ and conditional on the training data set $D$, defined as:

$$p(y|x, D) = \int p(y|x, \theta)p(\theta|D)d\theta$$

This is called the predictive posterior distribution and is calculated by marginalizing the product of the likelihood and posterior distribution over the model’s parameters. We can approximate this distribution by computing a Monte Carlo estimate of an expected value:

$$p(y|x, D) = \mathbb{E}_{\theta \sim p(\theta|D)}[p(y|x, \theta)] \simeq \frac{1}{L} \sum_{i=1}^L p(y|x, \theta(i))$$
where $\theta^{(1)}, ..., \theta^{(L)}$ are $L$ independent and identically distributed (iid) random parameters sampled from the posterior distribution that we can obtain for instance using a MCMC algorithm.

In order to calibrate the predictive posterior, one can use Safe Bayes approaches (Wilson and Izmailov, 2020) such as considering $n \in \mathbb{R}^+$ as a tunable hyper-parameter without using any data subsampling strategy. The loss then writes:

$$L_N^N(\theta) = -\log p(\theta) - \frac{n}{N} \sum_{i=1}^{N} \log p(y_i|x_i, \theta)$$

This technique is called tempering, as the $N/n$ factor can be interpreted as a temperature using the analogy with statistical physics. The temperature parameter is known to help in cases of model misspecification where there is no $\theta$ for which it can be assumed that the observations $(y_i, x_i)$ are iid according to $p(y|x, \theta)$.

3. RELATED WORK

In the context of sampling the posterior distribution, we introduce in this section some relevant literature that studied how to take into account a noisy estimate of either $L_N^N(\theta)$ or $\nabla_{\theta} L_N^N(\theta)$ computed from a subset of the data. The link between neural network noisy gradient and sampling a BNN posterior distribution is discussed in Section 6.

**Stochastic Gradient Langevin Dynamics** Welling and Teh (2011) showed that the iterates $\theta_t$ will converge to samples from the target posterior distribution in Equation (1) as they anneal the stepsize by adding the right amount of noise to a standard stochastic gradient optimization algorithm. This is known as the Stochastic Gradient Langevin Dynamics (SGLD) where the parameter update is given by:

$$\theta_{t+1} = \theta_t - \eta_t \nabla_{\theta} L_N^N(\theta_t) + \sqrt{2\eta_t} \epsilon_t$$

where $\epsilon_t$ is a random vector containing centered normally distributed components. No rejection step is required for a vanishing step size. Chen et al. (2014) later extended this idea to HMC sampler.

**Noisy Posterior Sampling Bias** Due to a potentially high variance of the stochastic gradients $\nabla_{\theta} L_N^N(\theta_t)$, Brosse et al. (2018) showed that the SGLD algorithm has an invariant probability measure which in general significantly departs from the target posterior for any non vanishing stepsize $\eta_t$. Furthermore, a recent work from Garriga-Alonso and Fortuin (2021) suggests that recent versions of SGLD implementing an additional Metropolis-Hastings rejection step do not improve this issue, because the resulting acceptance probability is likely to vanish too.

**Adaptive Subsampling Approach** Bardenet et al. (2014) showed one way to cope with the bias that is introduced by the data subsampling in the MCMC computation of a BNN posterior distribution. They propose an approximate implementation of the accept/reject step of Metropolis-Hastings algorithm while providing guarantees to coincide with this step based on the full dataset with a probability superior to a user-specified tolerance level.
Failures of Data Set Splitting Inference  Other works have exploited parallel computing to scale Bayesian inference to large datasets by using a two-step approach. First, a MCMC computation is run in parallel on $K$ (sub)posteriors defined on data partitions following $p(\theta|D) \propto \prod_{i=1}^{K} p(\theta)^{1/K} e^{-\mathcal{L}_n^i}$, where $i$ corresponds to the index of a randomly chosen mini-batch containing $n = N/K$ data. Then, a server combines local results. While efficient, this framework is very sensitive to the quality of sub-posterior sampling as shown by Souza et al. (2022).

4. PENALTY BAYESIAN NEURAL NETWORK

4.1. Biased Posterior Sampling Because Of Mini-Batches

We can draw iid samples from the posterior distribution defined in Equation (1) using MCMC through exploration of the state space of $\theta$ using Markov chains. It is well known that the detailed balance equation is a sufficient but not necessary condition ensuring that this Markov process possesses a stationary distribution corresponding to Equation (1). Detailed balance is given by:

$$A(\theta', \theta) q(\theta | \theta') e^{-\Delta(\theta', \theta)} = A(\theta, \theta') q(\theta | \theta')$$

(10)

where $A(\theta', \theta)$ corresponds to the probability of accepting a move from the parameter set $\theta$ to $\theta'$ suggested by the proposal distribution $q(\theta' | \theta)$. For the sake of brevity in notation we consider in this section a random walk algorithm corresponding to drawing a centered reduced normal random variable $\epsilon$ and computing $\theta' = \theta + \sqrt{2\eta} \epsilon$ such that the proposal distribution is symmetric $q(\theta' | \theta) = q(\theta | \theta')$. Using the Metropolis–Hastings (MH) algorithm, the acceptance then writes:

$$A(\theta', \theta) = \min \left(1, e^{-\Delta(\theta', \theta)} \right) \text{ with } \Delta(\theta', \theta) = \mathcal{L}_N^n(\theta') - \mathcal{L}_N^n(\theta)$$

(11)

We would rather compute the loss differences over random mini-batches instead of the full dataset $D$. Consequently, we introduce a random variable $\delta(\theta', \theta)$:

$$\delta(\theta', \theta) = \mathcal{L}_n^N(\theta') - \mathcal{L}_n^N(\theta)$$

(12)

that is a noisy estimate of the loss defined in Equation (11) such that $\Delta(\theta', \theta) = \langle \delta(\theta', \theta) \rangle$. As shown in the Figures 1 and 2(c), naively replacing $\Delta$ by $\delta$ in Equation (11) prevents MCMC algorithms from accurately sampling the target posterior distribution. We need to take into account the noise in the updated loss difference $\delta(\theta', \theta)$ caused by subsampling the data. This is in stark contrast to SGD algorithms (cf. Equation (4)) where a similar stochasticity has been shown to be beneficial for the optimization.

4.2. Noise Penalty Theory

In the context of statistical physics and computational chemistry, Ceperley and Dewing (1999) generalized the Metropolis-Hastings random walk algorithm to the situation where the energy difference $\Delta$ (loss difference in our case) is noisy and can only be estimated.
They showed that it is possible to sample the target distribution if we assume that we can write \( \delta(\theta', \theta) \) as equal to the loss difference \( \Delta(\theta', \theta) \) plus a Gaussian noise, meaning:

\[
\delta(\theta', \theta) \sim \mathcal{N}(\delta; \Delta(\theta', \theta), \sigma^2(\theta', \theta)) \tag{13}
\]

We can sample the target distribution under significant Gaussian noise by applying a penalty term \( -\sigma^2(\theta', \theta)/2 \) to the noisy difference in the acceptance ratio \( A \) such that

\[
A(\theta', \theta) = \min \left( 1, e^{-\delta(\theta', \theta) - \sigma^2(\theta', \theta)/2} \right) \tag{14}
\]

One can then show that detailed balance is satisfied on average, which is a sufficient condition for the Markov chain to sample the target distribution.

\[
\int d\delta A(\theta, \theta') \mathcal{N}(\delta; \Delta(\theta', \theta), \sigma^2(\theta', \theta)) = e^{-\Delta(\theta', \theta)} \int d\delta A(\theta', \theta) \mathcal{N}(\delta; \Delta(\theta, \theta'), \sigma^2(\theta, \theta')) \tag{15}
\]

In order to demonstrate the validity of the noise penalty method, let us first consider the case where the integration over the noise \( \delta \) is performed exactly, either analytically or via numerical quadrature. Then, a Monte Carlo proposition \( \theta \rightarrow \theta' \) is accepted with probability

\[
\overline{A}(\theta', \theta) = \int d\delta \mathcal{N}(\delta; \Delta(\theta', \theta), \sigma^2(\theta', \theta)) A(\theta', \theta) \tag{16}
\]

and a Markov chain Monte Carlo based on \( \overline{A}(\theta', \theta) \) satisfies the detailed balance condition, Eq. (15), or

\[
\overline{A}(\theta', \theta) p(\theta|\mathcal{D}) = \overline{A}(\theta, \theta') p(\theta'|\mathcal{D}) \tag{17}
\]

as can be checked explicitly Ceperley and Dewing (1999). Under the usual assumption of irreducibility of \( \overline{A}(\theta', \theta) \), the corresponding MCMC converges to a unique stationary distribution, \( p(\theta|\mathcal{D}) \). However, this result of the MCMC cannot depend on the underlying quadrature used to perform the integration over the noise, and a statistical evaluation of Eq. (16) at each step of the Markov chain must converge to the same distribution. Therefore, the noise penalty methods gives access to an unbiased sampling of \( p(\theta|\mathcal{D}) \) under our assumption of a normal distribution of \( \delta(\theta', \theta) \) with known variance \( \sigma^2(\theta', \theta) \).

Figure 1 shows the benefits of adding the noise penalty on a synthetic linear regression example. Typically, increasing the size \( n \) of the mini-batches diminishes the magnitude of the noise, i.e. reduces the variance \( \sigma^2(\theta', \theta) \) and therefore increases the acceptance \( A(\theta', \theta) \).

From Equation (14) one can immediately identify a drawback of PBNN leading to an exponential reduction of the acceptance as the variance \( \sigma^2(\theta', \theta) \) is always non-negative. In Section 6, we show an extension of the penalty method to non-symmetric proposal distributions \( q(\theta'|\theta) \) (Welling and Teh, 2011; Neal, 2010) that aims at increasing the acceptance, thus mitigating this drawback. Additionally, in the case of BNNs, the variance \( \sigma^2(\theta', \theta) \) is unknown and can only be estimated. While it is possible to account for noisy variances (Ceperley and Dewing, 1999), we will not explore this extension further.

4.3. Expected Posterior Sampling Using Mini-Batches

As shown in Figure 1, it is possible to target the usual posterior distribution as defined by the loss given in Equation (2) using the noise penalty. Since the noise penalty requires
Figure 1: Illustration of posterior predictive distributions, defined in Equation (7), computed for a linear regression task. The coloured areas correspond to the mean of the distributions ± one standard deviation. The blue area is computed by naively replacing $\Delta$ by $\delta$ in the MH algorithm from Equation (11). The noisy loss difference $\delta$ as defined by Equation (12) is computed on a single mini-batch of 2 data points. The yellow area shows a similar computation, that incorporates the noise penalty term as defined in Equation (14). It correctly matches the red area corresponding to the analytical Bayesian linear regression with Gaussian prior and known variance (Bishop, 2007).

the computation of $\sigma^2(\theta, \theta')$, i.e. the expected variance of the loss differences over multiple mini-batches, we would like to show in the following that it is more interesting to target the posterior distribution defined by the mean loss $L_n(\theta)$. This loss corresponds to the average loss given an infinite number of mini-batches.

$$L_n(\theta) = \langle L_n^n(\theta) \rangle$$

(18)

One can write this mean loss using the expected value of the likelihood as shown in Equation (19). In practice, computing this quantity is intractable since it would require an infinite amount of data.

$$L_n(\theta) = -\log p(\theta) - n \mathbb{E}[\log p(y_i|x_i, \theta)]$$

(19)

It is important to distinguish between $L_n(\theta)$ and $L_N^N(\theta) = \langle L_n^N(\theta) \rangle$, where the latter represents the mean loss over all mini-batches within a single dataset. Specifically, when $n = N$, computing $L_N(\theta)$ is comparative to treating the entire training dataset as a single large mini-batch and computing its expected loss as $\langle L_N^N(\theta) \rangle = L_N(\theta)$. Applying the MH algorithm as defined by Equation (11) is impossible as the loss difference $\Delta$ given by:

$$\Delta(\theta', \theta) = L_n(\theta') - L_n(\theta)$$

(20)

is unknown and generally intractable. We can, however, compute a noisy estimate $\delta$ of $\Delta$ based on the losses of the mini-batches:

$$\delta(\theta', \theta) = L_n^n(\theta') - L_n^n(\theta)$$

(21)
where it is evident that \( \langle \delta(\theta', \theta) \rangle = \Delta(\theta', \theta) \). In this case, varying the size \( n \) of the mini-batches corresponds to changing the target poster distribution that we wish to sample. This relates to empirical Bayes and tempering techniques shown in the appendix (see Section A). However, these well establish techniques do not target the distribution defined by Equation (19), resulting in qualitatively different results compared to PBNN (cf. Figure 4(c) in Appendix A). As shown in Equation (8), tempering the posterior distribution requires the computation of the likelihood over the full data set, a process we aim to avoid.

The noise penalty method is particularly relevant for BNNs as these networks do not scale well for large dataset. Additionally, their common uninformative prior makes it difficult to calibrate the Bayesian predictive marginal distribution. Similar to Safe Bayes approaches (see Section A), one can use the mini-batch size \( n \) as a calibration hyper-parameter. The example on a synthetic non-linear regression task from Figure 2 shows the benefits of targeting the posterior distribution given by Equation (19) instead of Equation (2). It shows an example of a MCMC algorithm that uses a noise penalty in the acceptance Equation (30). The figure shows the expected posterior predictive distributions defined as:

\[
p(y|x, n) = \frac{\int \frac{p(y|x, \theta)e^{-\mathcal{L}_n(\theta)}}{e^{-\mathcal{L}_n(\theta)}d\theta} \approx \frac{1}{L} \sum_{i=1}^{L} p(y|x, \theta^{(i)})}{\int e^{-\mathcal{L}_n(\theta)}d\theta}
\]

where \( \theta^{(i)} \) should be iid samples obtained from the posterior distribution which is proportional to \( e^{-\mathcal{L}_n(\theta)} \). In practice, obtaining such a set of samples using a MCMC algorithm as defined in Equation (30) challenges the iid assumption since \( \delta(\theta', \theta) \) are computed on a limited amount of mini-batches. As shown in the rest of this article, empirical results show that this assumption is not an obstacle to the use of this method.

Even in the situation where \( \delta(\theta', \theta) \) is not distributed according to a Gaussian distribution as required by Equation (13), we can estimate its variance following:

\[
\sigma^2(\theta', \theta) \approx \frac{1}{M-1} \sum_{i=1}^{M} \left( \mathcal{L}_{n,i}^{n}(\theta') - \mathcal{L}_{n,i}^{n}(\theta) - \delta(\theta', \theta) \right)^2
\]

where \( i \) denotes the index of a randomly chosen mini-batch.

4.4. Large Number Of Mini-Batches M Scenario

In the case of a large number of mini-batches \( M \) i.e a large available dataset and small mini-batches \( n \), it is convenient to define \( \delta(\theta', \theta) \) as an empirical average over the loss difference:

\[
\delta(\theta', \theta) = \frac{1}{M} \sum_{i=1}^{M} \left( \mathcal{L}_{n,i}^{n}(\theta') - \mathcal{L}_{n,i}^{n}(\theta) \right)
\]

with \( M \) strictly smaller than the total number of available mini-batches \( N/n \). By definition, this average is an unbiased estimator such that \( \langle \delta(\theta', \theta) \rangle = \Delta(\theta', \theta) \). Importantly, the central limit theorem ensures that \( \delta(\theta', \theta) \) approaches a normal distribution for large \( M \) as required by the noise penalty setup in the Equation (13).
Penalty Bayesian Neural Networks

(a) BNN prediction corresponding to Equation (6) with \(N = 2000\).

(b) PBNN prediction, \(n = 5\) and \(M = 5\).

(c) Prediction without noise penalty using Equation (12), \(n = 10\).

(d) PBNN prediction, \(n = 10\) and \(M = 5\).

Figure 2: Illustration of posterior predictive distributions, defined in Equation (22). Shaded regions indicate predictive means ± one, two and three standard deviations. We use a homoscedastic Gaussian likelihood (cf. loss defined in Equation (3)). The noise in the data is small such that the visible posterior distribution variance is due to the epistemic uncertainty. The training data set contains \(N = 2000\) points meaning that at each step, only a fraction of the data points is used by the PBNN. We emphasize that none of these images are expected to match Figure 2(a) as each targets a different posterior distribution.

Using the definition from Equation (24), the variance of the random variable \(\delta(\theta', \theta)\) strictly decreases with the number of mini-batches \(M\). This is very convenient since, as shown in Equation (14), the loss difference \(\delta(\theta', \theta)\) must dominate the always positive variance \(\sigma^2(\theta', \theta)/2\) in order to obtain a reasonable acceptance \(A(\theta', \theta)\) i.e sufficiently greater than zero.

While \(\sigma^2(\theta', \theta)\) is unknown, we can obtain an unbiased estimate for it as:

\[
\sigma^2(\theta', \theta) \simeq \frac{1}{M(M-1)} \sum_{i=1}^{M} \left( \mathcal{L}_{n,i}^a(\theta') - \mathcal{L}_{n,i}^a(\theta) - \delta(\theta', \theta) \right)^2
\]

In the following, we do not take into account the error over the estimation of the variance \(\sigma^2(\theta', \theta)\). This assumption is based on the hypothesis that the variations of \(\sigma^2\) as a function of \(\theta'\) and \(\theta\) predominate over the estimation noise. Alternatively, a significant number of mini-batches is required to minimize this error. Leading order corrections in this noise are discussed in Ceperley and Dewing (1999).
5. PBNN ALGORITHM

5.1. Penalty Bayesian Neural Network Posterior Sampling Algorithm

Given Equation (24) and Equation (25) we can design Algorithm 1 that is able to sample BNN posterior distributions while restricting the evaluation of the likelihood to \(n\) data points and \(M\) small mini-batches at each iteration step. Figure 2(b) and Figure 2(d) are drawn using this algorithm.

**Algorithm 1 PBNN Algorithm**

1. Initialize a parameter vector \(\theta_t \leftarrow \theta_0\)

2. For \(t = 1\) to maximum number of iterations
   (a) Sample a new configuration \(\theta' \sim q(\theta' | \theta_t)\)
   (b) Compute the noisy loss difference \(\delta(\theta', \theta_t)\) from Equation (24) (or Equation (21))
   (c) Compute the corresponding \(\sigma^2(\theta', \theta_t)\) from Equation (25) (or Equation (23))
   (d) Compute the acceptance \(A(\theta', \theta_t) \leftarrow \min\left(1, \frac{q(\theta_t | \theta')}{q(\theta' | \theta_t)} e^{-\delta(\theta', \theta_t) - \sigma^2(\theta', \theta_t)/2}\right)\)
   (e) Sample a uniform random variable \(u \sim U(0,1)\)
   (f) If \(u \leq A(\theta', \theta_t)\)
       \(\theta_{t+1} \leftarrow \theta'\)
   otherwise
       \(\theta_{t+1} \leftarrow \theta_t\)
   (g) increment \(t\) and return to step 2a

5.2. Prediction Calibration Using The Mini-Batch Size \(n\)

As introduced in Section 4.3, the mini-batch size \(n\) is a hyperparameter that we can use to calibrate the posterior predictive distribution. PBNN’s ability to evaluate the likelihood over small mini-batches even in the presence of strong noise allows us to achieve this. A model is calibrated if, in the long-run, the proportion of forecast \(\times\) percent credible intervals that succeed in covering the actual value of the predicted quantity turns out to be \(\times\) percent (Degroot and Fienberg, 2018).

The reliability diagrams in Figure 3 display the correspondence or discrepancy between a model prediction uncertainty and the observed frequencies in a test set of data. Ideal calibration corresponds to the \(y = x\) diagonal: we see that PBNN mini-batch size can help calibrate a model prediction which is overconfident.

Note that, partial BNNs (Sharma et al., 2023) are used in the experiments shown in Figure 3 to induce overconfidence. Only the last layer is stochastic, such that the usual posterior distribution (Equation (1)), given the full available dataset does not capture the
total uncertainty (see misspecification in the Appendix A). In Figure 3, BNN and PBNN share the same non-stochastic layers.

**Figure 3:** Reliability diagrams on test data. BNN prediction corresponds to Equation (7) whereas the prediction of PBNN is computed using Equation (22). MNIST classifiers obtain a similar accuracy test score: 93.2% using a PBNN, and 93.6% using a BNN. The architecture of the softmax classifier is a single hidden layer containing 20 neurons.

6. PENALIZED LANGEVIN DYNAMICS

6.1. Metropolis Adjusted Langevin Algorithm

In Algorithm 1, we are free to choose a proposal distribution \( q(\theta' | \theta) \). It is well known that choosing a non-symmetric proposal distribution can speed up the mixing of the Markov Chain and help BNN scale to larger systems by maximizing the acceptance \( A(\theta', \theta) \). Assuming a sufficiently small step from \( \theta \) to \( \theta' \), the Metropolis-Hastings acceptance writes:

\[
A(\theta', \theta) = \min \left( 1, \frac{q(\theta | \theta')}{q(\theta' | \theta)} e^{-\Delta(\theta', \theta)} \right) \approx \min \left( 1, \frac{q(\theta | \theta')}{q(\theta' | \theta)} e^{-(\theta - \theta') \cdot (\nabla_\theta \mathcal{L}_n(\theta) + \nabla_\theta \mathcal{L}_n(\theta)) / 2} \right) \tag{26}
\]

where we have Taylor-expanded the loss \( \mathcal{L}(\theta) \) around \( \theta' \). The maximization of \( A(\theta', \theta) \) leads to a drift in the Gaussian proposal distribution that corresponds to the gradient of the loss:

\[
q(\theta' | \theta) = \mathcal{N}(\theta'; \theta - \eta \nabla_\theta \mathcal{L}_n(\theta), 2\eta) \tag{27}
\]

Sampling a new state \( \theta' \) from the proposal distribution \( q(\theta' | \theta) \) exactly corresponds to drawing a centered reduced normal random variable \( \epsilon \), and computing:

\[
\theta' = \theta - \eta \nabla_\theta \mathcal{L}_n(\theta) + \sqrt{2\eta} \epsilon \tag{28}
\]
Equation (28) is equivalent to a Langevin dynamic and gives rise to the celebrated Metropolis-Adjusted Langevin algorithm (MALA) and Smart Monte Carlo (Rossky et al., 1978).

6.2. Penalized MALA

In order to use a noisy gradient, $\nabla_\theta L_n(\theta) \simeq \nabla_\theta L_n^n(\theta)$, as an approximation of the Gaussian mean’s drift, one can design a proposal distribution $q(\theta' | \theta)$ such that:

$$q(\theta' | \theta) = N(\theta' ; \theta - \eta \nabla_\theta L_n^n(\theta), 2\eta)$$ (29)

and set a non-zero step size $\eta$ while computing the Metropolis-Hastings acceptance:

$$A(\theta', \theta) = \min \left( 1, \frac{q(\theta | \theta')}{q(\theta' | \theta)} e^{-\delta(\theta', \theta) - \sigma^2(\theta', \theta)/2} \right)$$ (30)

which also satisfies the detailed balance in Equations (10) and (15) on average. The non-trivial term in the Metropolis-Hastings acceptance then writes:

$$\log \left( \frac{q(\theta | \theta')}{q(\theta' | \theta)} e^{-\delta(\theta', \theta) - \sigma^2(\theta', \theta)/2} \right) = -\frac{1}{4\eta} \left\| \eta \left( \nabla_\theta L_n^n(\theta') + \nabla_\theta L_n^n(\theta) \right) - \sqrt{2\eta} \epsilon \right\|^2 + \frac{1}{4\eta} \left\| \sqrt{2\eta} \epsilon \right\|^2 - \delta(\theta', \theta) - \sigma^2(\theta', \theta)/2$$ (31)

6.3. Unadjusted Langevin Algorithm

For large-sized models, biased samplers like the Unadjusted Langevin Algorithm (ULA) are known to be very effective. This is because they skip the rejection step i.e set $A(\theta', \theta) = 1$ for a sufficiently small step size $\eta$ resulting in an unadjusted Langevin sampling as:

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta L_n(\theta_t) + \sqrt{2\eta} \epsilon_t$$ (32)

Note that in Equation (32), new states are automatically accepted. Once again, in order to model a noisy estimate of the loss, it is tempting to replace the drift $\nabla_\theta L_n(\theta_t)$ with an estimate such that:

$$\eta \nabla_\theta L_n(\theta) = \eta \nabla_\theta L_n^n(\theta) + \eta \sigma(\theta)$$ (33)

For a vanishing step size $\eta \to 0$, one may then expect that the additional noise term, $\eta \sigma(\theta)$, gets negligible compared to the random noise of order $\eta^{1/2}$ in Equation (32) as suggested by SGLD (Welling and Teh, 2011). However, the uncertainty of the loss gradient $\sigma(\theta)$ does in general not result in white noise, but is correlated between different parameters $\theta$. For non-vanishing $\eta$, the noisy loss gradient can thus trigger a significant departure from the target posterior (see Brosse et al. (2018) and Figures 4(a) and 4(b) in Appendix A).

7. CONCLUSION

Uncertainty quantification for the predictions of large size neural networks remains an open issue. In this work, we have introduced PBNN as a novel approach to enable data subsampling for Bayesian Neural Network without relying on a noisy estimate of the gradient such as those used in Stochastic Gradient Langevin Dynamic.
First, we demonstrated that a naive estimation of the likelihood based on noisy loss introduces a bias in the posterior sampling if not correctly accounted for. We then showed that a generalization of the Metropolis Hastings algorithm can eliminate the bias, allowing for exact posterior sampling even amid substantial noise. This approach necessitates an additional “noise penalty” that matches the variance of the noisy loss difference. The drawback of the method is that it exponentially suppresses the MCMC acceptance probability.

In practice, the noise penalty corresponds to replacing a single large data set by multiple smaller subsampled mini-batches associated with an uncertainty over their losses. Varying the size of the mini-batches enables a calibration of the predictive model. We have compared this benefit of PBNN to other techniques such as tempered Safe Bayes approaches.

Based on this calibration principle, we conducted numerical experiments demonstrating the robust predictive performance of PBNNs. We are optimistic that integrating data subsampling with other Monte Carlo acceleration techniques such as HMC (Neal, 2010) will enable the computation of uncertainties for previously unattainable models and data set sizes.

Lastly, PBNN appears to be particularly well-suited for scenarios where the data sets $D$ are distributed across multiple decentralized devices as in the typical federated learning setup. The key to its suitability lies in the noise penalty which is determined by the variance of losses across several mini-batches. These losses can be computed independently on each device’s data set. Consequently, PBNN should enable the possibility to compute uncertainty with separate data sets without exchanging them.

8. ACKNOWLEDGMENTS

This work has been supported by the French government under the ”France 2030” program, as part of the SystemX Technological Research Institute. Authors thank Victor Berger and Han Wang for useful comments and discussions.

References

Rémi Bardenet, Arnaud Doucet, and Chris Holmes. Towards scaling up markov chain monte carlo: an adaptive subsampling approach. In Eric P. Xing and Tony Jebara, editors, Proceedings of the 31st International Conference on Machine Learning, volume 32 of Proceedings of Machine Learning Research, pages 405–413, Beijing, China, 22–24 Jun 2014. PMLR. URL https://proceedings.mlr.press/v32/bardenet14.html.

Rémi Bardenet, Arnaud Doucet, and Chris Holmes. On Markov chain Monte Carlo methods for tall data. Journal of Machine Learning Research, 18(47):1–43, 2017. URL http://jmlr.org/papers/v18/15-205.html.

Christopher M. Bishop. Pattern Recognition and Machine Learning (Information Science and Statistics). Springer, 1 edition, 2007. ISBN 0387310738. URL http://www.amazon.com/Pattern-Recognition-Learning-Information-Statistics/dp/0387310738%3FSubscriptionId%3D13CT5CVB80YFWJEPWS02%26tag%3Dws%26linkCode%3Dxm2%26camp%3D2025%26creative%3D165953%26creativeASIN%3D0387310738.
Nicolas Brosse, Alain Durmus, and Eric Moulines. The promises and pitfalls of Stochastic Gradient Langevin Dynamics. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/335d1b90bfa4ee70b39d08a4ae0cf2d-Paper.pdf.

D. M. Ceperley and M. Dewing. The penalty method for random walks with uncertain energies. The Journal of Chemical Physics, 110(20):9812–9820, 05 1999. ISSN 0021-9606. doi: 10.1063/1.478034. URL https://doi.org/10.1063/1.478034.

Tianqi Chen, Emily Fox, and Carlos Guestrin. Stochastic gradient hamiltonian monte carlo. In Eric P. Xing and Tony Jebara, editors, Proceedings of the 31st International Conference on Machine Learning, volume 32 of Proceedings of Machine Learning Research, pages 1683–1691, Beijing, China, 22–24 Jun 2014. PMLR. URL https://proceedings.mlr.press/v32/chen14.html.

Morris H. Degroot and Stephen E. Fienberg. The Comparison and Evaluation of Forecasters. Journal of the Royal Statistical Society Series D: The Statistician, 32(1-2):12–22, 12 2018. ISSN 2515-7884. doi: 10.2307/2987588. URL https://doi.org/10.2307/2987588.

Vincent Fortuin. Priors in bayesian deep learning: A review. International Statistical Review, 90(3):563–591, 2022. doi: https://doi.org/10.1111/insr.12502. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/insr.12502.

Adrià Garriga-Alonso and Vincent Fortuin. Exact langevin dynamics with stochastic gradients. CoRR, abs/2102.01691, 2021. URL https://arxiv.org/abs/2102.01691.

Jakob Gawlikowski, Cedrique Rovile Njieutchu Tassi, Mohsin Ali, Jongseok Lee, Matthias Hunt, Jianxiang Feng, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, Muhammad Shahzad, Wen Yang, Richard Bamler, and Xiao Xiang Zhu. A survey of uncertainty in deep neural networks. Artificial Intelligence Review, 56(1):1513–1589, October 2023. ISSN 1573-7462. doi: 10.1007/s10462-023-10562-9. URL https://doi.org/10.1007/s10462-023-10562-9.

Pavel Izmailov, Sharad Vikram, Matthew D Hoffman, and Andrew Gordon Gordon Wilson. What are bayesian neural network posteriors really like? In Marina Meila and Tong Zhang, editors, Proceedings of the 38th International Conference on Machine Learning, volume 139 of Proceedings of Machine Learning Research, pages 4629–4640. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/izmailov21a.html.

Anoop Korattikara, Yutian Chen, and Max Welling. Austerity in mcmc land: Cutting the metropolis-hastings budget. In Eric P. Xing and Tony Jebara, editors, Proceedings of the 31st International Conference on Machine Learning, volume 32 of Proceedings of Machine Learning Research, pages 181–189, Beijing, China, 22–24 Jun 2014. PMLR. URL https://proceedings.mlr.press/v32/korattikara14.html.

Radford M. Neal. MCMC Using Hamiltonian Dynamics. Handbook of Markov Chain Monte Carlo, 54:113–162, 2010.
Carlo Pierleoni, David M. Ceperley, and Markus Holzmann. Coupled electron-ion monte carlo calculations of dense metallic hydrogen. *Phys. Rev. Lett.*, 93:146402, Sep 2004. doi: 10.1103/PhysRevLett.93.146402. URL https://link.aps.org/doi/10.1103/PhysRevLett.93.146402.

P. J. Rossky, J. D. Doll, and H. L. Friedman. Brownian dynamics as smart Monte Carlo simulation. *The Journal of Chemical Physics*, 69(10):4628–4633, 11 1978. ISSN 0021-9606. doi: 10.1063/1.436415. URL https://doi.org/10.1063/1.436415.

Daniel Seita, Xinlei Pan, Haoyu Chen, and John Canny. An efficient minibatch acceptance test for metropolis-hastings. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, *IJCAI-18*, pages 5359–5363. International Joint Conferences on Artificial Intelligence Organization, 7 2018. doi: 10.24963/ijcai.2018/753. URL https://doi.org/10.24963/ijcai.2018/753.

Mrinank Sharma, Sebastian Farquhar, Eric Nalisnick, and Tom Rainforth. Do bayesian neural networks need to be fully stochastic? In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent, editors, *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 7694–7722. PMLR, 25–27 Apr 2023. URL https://proceedings.mlr.press/v206/sharma23a.html.

Daniel A. De Souza, Diego Mesquita, Samuel Kaski, and Luigi Acerbi. Parallel MCMC Without Embarrassing Failures. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, pages 1786–1804. PMLR, May 2022. URL https://proceedings.mlr.press/v151/de-souza22a.html. ISSN: 2640-3498.

Max Welling and Yee Whye Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, *ICML’11*, page 681–688, Madison, WI, USA, 2011. Omnipress. ISBN 9781450306195.

Andrew G Wilson and Pavel Izmailov. Bayesian Deep Learning and a Probabilistic Perspective of Generalization. In *Advances in Neural Information Processing Systems*, volume 33, pages 4697–4708. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/322f62469c5e3c7dc3e58f5a4d1ea399-Abstract.html.
Appendix A. SGLD, Safe Bayes and PBNN

In principle SGLD can be used to compute Equation (22) in the same way as PBNN. However, because of the noise introduced by the data subsampling, SGLD requires a vanishing learning rate $\eta$ as shown in Figure 4. In the same figure, PBNN has a learning rate of $\eta = 0.1$: varying $\eta$ changes the MCMC acceptance and not the prediction which is independent of the step size.

Figure 4: SGLD results are very sensitive to the learning rate $\eta$. Figure 4(d) is used as a reference as it targets the same posterior as Figure 4(a) and Figure 4(b). We note that as expected, Safe Bayes approaches obtain a qualitatively different result from PBNN for the same mini-batch size.