How Good is the Bayes Posterior in Deep Neural Networks Really?

Florian Wenzel[*1] **Kevin Roth**[*+2] **Bastiaan S. Veeling**[*+3 1] **Jakub Świątkowski**[4+] **Linh Tran**[5+]
Stephan Mandt[6+] **Jasper Snoek**[1] **Tim Salimans**[1] **Rodolphe Jenatton**[1] **Sebastian Nowozin**[7+]

Abstract

During the past five years the Bayesian deep learning community has developed increasingly accurate and efficient approximate inference procedures that allow for Bayesian inference in deep neural networks. However, despite this algorithmic progress and the promise of improved uncertainty quantification and sample efficiency there are—as of early 2020—no publicized deployments of Bayesian neural networks in industrial practice. In this work we cast doubt on the current understanding of Bayes posteriors in popular deep neural networks: we demonstrate through careful MCMC sampling that the posterior predictive induced by the Bayes posterior yields systematically worse predictions compared to simpler methods including point estimates obtained from SGD. Furthermore, we demonstrate that predictive performance is improved significantly through the use of a "cold posterior" that overcounts evidence. Such cold posteriors sharply deviate from the Bayesian paradigm but are commonly used as heuristic in Bayesian deep learning papers. We put forward several hypotheses that could explain cold posteriors and evaluate the hypotheses through experiments. Our work questions the goal of accurate posterior approximations in Bayesian deep learning: If the true Bayes posterior is poor, what is the use of more accurate approximations? Instead, we argue that it is timely to focus on understanding the origin of the improved performance of cold posteriors.

CODE: https://github.com/google-research/google-research/tree/master/cold_posterior_bnn

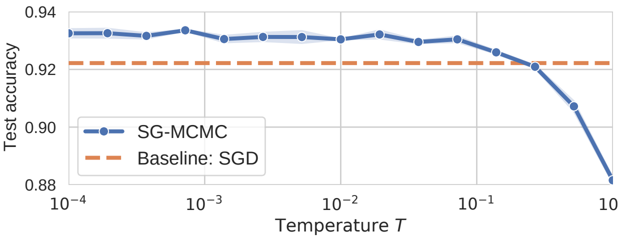

Figure 1. The "**cold posterior**" effect: for a ResNet-20 on CIFAR-10 we can improve the generalization performance significantly by cooling the posterior with a temperature $T \ll 1$, deviating from the Bayes posterior $p(\boldsymbol{\theta}|\mathcal{D}) \propto \exp(-U(\boldsymbol{\theta})/T)$ at $T = 1$.

1. Introduction

In supervised deep learning we use a training dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1,\dots,n}$ and a probabilistic model $p(y|x, \boldsymbol{\theta})$ to minimize the regularized cross-entropy objective,

$$L(\boldsymbol{\theta}) := -\frac{1}{n} \sum_{i=1}^{n} \log p(y_i|x_i, \boldsymbol{\theta}) + \Omega(\boldsymbol{\theta}), \tag{1}$$

where $\Omega(\boldsymbol{\theta})$ is a regularizer over model parameters. We approximately optimize (1) using variants of stochastic gradient descent (SGD), (Sutskever et al., 2013). Beside being efficient, the SGD minibatch noise also has generalization benefits (Masters & Luschi, 2018; Mandt et al., 2017).

1.1. Bayesian Deep Learning

In Bayesian deep learning we do not optimize for a *single* likely model but instead want to discover *all* likely models. To this end we approximate the *posterior distribution* over model parameters, $p(\boldsymbol{\theta}|\mathcal{D}) \propto \exp(-U(\boldsymbol{\theta})/T)$, where $U(\boldsymbol{\theta})$ is the *posterior energy function*,

$$U(\boldsymbol{\theta}) := -\sum_{i=1}^{n} \log p(y_i|x_i, \boldsymbol{\theta}) - \log p(\boldsymbol{\theta}), \tag{2}$$

and T is a *temperature*. Here $p(\boldsymbol{\theta})$ is a *proper* prior density function, for example a Gaussian density. If we scale $U(\boldsymbol{\theta})$ by $1/n$ and set $\Omega(\boldsymbol{\theta}) = -\frac{1}{n} \log p(\boldsymbol{\theta})$ we recover $L(\boldsymbol{\theta})$ in (1). Therefore $\exp(-U(\boldsymbol{\theta}))$ simply gives high probability to models which have low loss $L(\boldsymbol{\theta})$. Given $p(\boldsymbol{\theta}|\mathcal{D})$ we *predict* on a new instance x by averaging over all likely models,

$$p(y|x, \mathcal{D}) = \int p(y|x, \boldsymbol{\theta})\, p(\boldsymbol{\theta}|\mathcal{D})\, \mathrm{d}\boldsymbol{\theta}, \tag{3}$$

[*] Equal contribution [+] Work done while at Google [1] Google Research [2] ETH Zurich [3] University of Amsterdam [4] University of Warsaw [5] Imperial College London [6] University of California, Irvine [7] Microsoft Research. Correspondence to: Florian Wenzel <florianwenzel@google.com>.

Proceedings of the 37th International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020. Copyright 2020 by the author(s).

where (3) is also known as *posterior predictive* or *Bayes ensemble*. Solving the integral (3) exactly is not possible. Instead, we approximate the integral using a sample approximation, $p(y|x, \mathcal{D}) \approx \frac{1}{S} \sum_{s=1}^{S} p(y|x, \boldsymbol{\theta}^{(s)})$, where $\boldsymbol{\theta}^{(s)}$, $s = 1, \ldots, S$, is approximately sampled from $p(\boldsymbol{\theta}|\mathcal{D})$.

The remainder of this paper studies a surprising effect shown in Figure 1, the "*Cold Posteriors*" effect: for deep neural networks the Bayes posterior (at temperature $T = 1$) works poorly but by cooling the posterior using a temperature $T < 1$ we can significantly improve the prediction performance.

> **Cold Posteriors**: among all temperized posteriors the best posterior predictive performance on holdout data is achieved at temperature $T < 1$.

1.2. Why Should Bayes ($T = 1$) be Better?

Why would we expect that predictions made by the *ensemble model* (3) could improve over predictions made at a single well-chosen parameter? There are three reasons: 1. *Theory*: for several models where the predictive performance can be analyzed it is known that the posterior predictive (3) can dominate common point-wise estimators based on the likelihood, (Komaki, 1996), even in the case of misspecification, (Fushiki et al., 2005; Ramamoorthi et al., 2015); 2. *Classical empirical evidence*: for classical statistical models, averaged predictions (3) have been observed to be more robust in practice, (Geisser, 1993); and 3. *Model averaging*: recent deep learning models based on deterministic model averages, (Lakshminarayanan et al., 2017; Ovadia et al., 2019), have shown good predictive performance.

Note that a large body of work in the area of Bayesian deep learning in the last five years is motivated by the assertion that predicting using (3) is desirable. We will confront this assertion through a simple experiment to show that our understanding of the Bayes posterior in deep models is limited. Our work makes the following **contributions**:

- We demonstrate for two models and tasks (ResNet-20 on CIFAR-10 and CNN-LSTM on IMDB) that the Bayes posterior predictive has poor performance compared to SGD-trained models.
- We put forth and systematically examine hypotheses that could explain the observed behaviour.
- We introduce two new diagnostic tools for assessing the approximation quality of stochastic gradient Markov chain Monte Carlo methods (SG-MCMC) and demonstrate that the posterior is accurately simulated by existing SG-MCMC methods.

2. Cold Posteriors Perform Better

We now examine the quality of the posterior predictive for two simple deep neural networks. We will describe details of the models, priors, and approximate inference methods in Section 3 and Appendix A.1 to A.3. In particular, we will study the accuracy of our approximate inference and the influence of the prior in great detail in Section 4 and Section 5.2, respectively. Here we show that temperized Bayes ensembles obtained via low temperatures $T < 1$ outperform the true Bayes posterior at temperature $T = 1$.

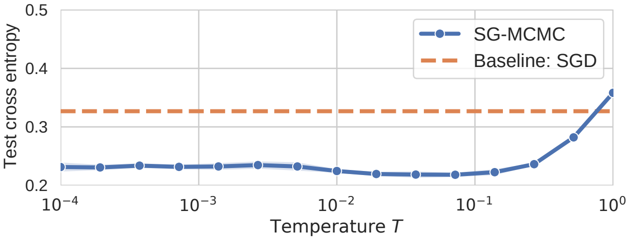

Figure 2. Predictive performance on the CIFAR-10 test set for a cooled ResNet-20 Bayes posterior. The SGD baseline is separately tuned for the same model (Appendix A.2).

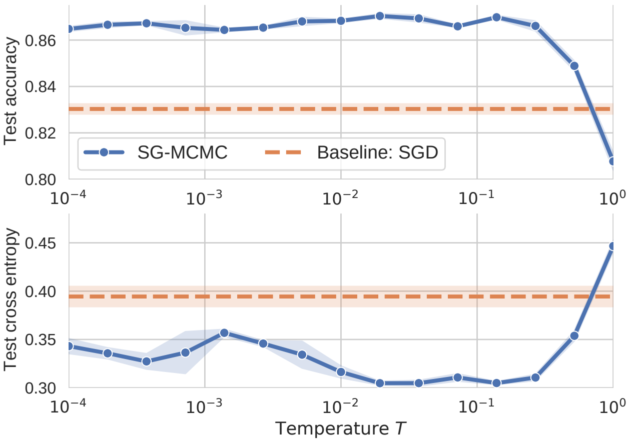

Figure 3. Predictive performance on the IMDB sentiment task test set for a tempered CNN-LSTM Bayes posterior. Error bars are ± one standard error over three runs. See Appendix A.4.

2.1. Deep Learning Models: ResNet-20 and LSTM

ResNet-20 on CIFAR-10. Figure 1 and 2 show the test accuracy and test cross-entropy of a Bayes prediction (3) for a ResNet-20 on the CIFAR-10 classification task.[1] We can clearly see that both accuracy and cross-entropy are significantly improved for a temperature $T < 1/10$ and that this trend is consistent. Also, surprisingly this trend holds all the way to small $T = 10^{-4}$: the test performance obtained from an ensemble of models at temperature $T = 10^{-4}$ is superior to the one obtained from $T = 1$ and better than the performance of a single model trained with SGD. In Appendix G we show that the uncertainty metrics Brier score (Brier, 1950) and expected calibration error (ECE) (Naeini et al., 2015) are also improved by cold posteriors.

[1] A similar plot is Figure 3 in (Baldock & Marzari, 2019) and another is in the appendix of (Zhang et al., 2020).

CNN-LSTM on IMDB text classification. Figure 3 shows the test accuracy and test cross-entropy of the tempered prediction (3) for a CNN-LSTM model on the IMDB sentiment classification task. The optimal predictive performance is again achieved for a tempered posterior with a temperature range of approximately $0.01 < T < 0.2$.

2.2. Why is a Temperature of $T < 1$ a Problem?

There are two reasons why cold posteriors are problematic. *First*, $T < 1$ corresponds to artificially sharpening the posterior, which can be interpreted as overcounting the data by a factor of $1/T$ and a rescaling[2] of the prior as $p(\theta)^{\frac{1}{T}}$. This is equivalent to a Bayes posterior obtained from a dataset consisting of $1/T$ replications of the original data, giving too strong evidence to individual models. For $T = 0$, all posterior probability mass is concentrated on the set of maximum a posteriori (MAP) point estimates. *Second*, $T = 1$ corresponds to the true Bayes posterior and performance gains for $T < 1$ point to a deeper and potentially resolvable problem with the prior, likelihood, or inference procedure.

2.3. Confirmation from the Literature

Should the strong performance of tempering the posterior with $T \ll 1$ surprise us? It certainly is an observation that needs to be explained, but it is not new: if we comb the literature of Bayesian inference in deep neural networks we find broader evidence for this phenomenon.

Related work that uses $T < 1$ posteriors in SG-MCMC. The following table lists work that uses SG-MCMC on deep neural networks and tempers the posterior.[3]

Reference	Temperature T
(Li et al., 2016)	$1/\sqrt{n}$
(Leimkuhler et al., 2019)	$T < 10^{-3}$
(Heek & Kalchbrenner, 2020)	$T = 1/5$
(Zhang et al., 2020)	$T = 1/\sqrt{50000}$

Related work that uses $T < 1$ posteriors in Variational Bayes. In the variational Bayes approach to Bayesian neural networks, (Blundell et al., 2015; Hinton & Van Camp, 1993; MacKay et al., 1995; Barber & Bishop, 1998) we optimize the parameters τ of a variational distribution $q(\boldsymbol{\theta}|\tau)$ by maximizing the evidence lower bound (ELBO),

$$\mathbb{E}_{\boldsymbol{\theta}\sim q(\boldsymbol{\theta}|\tau)}\left[\sum_{i=1}^{n} \log p(y_i|x_i,\boldsymbol{\theta})\right] - \lambda D_{\mathrm{KL}}(q(\boldsymbol{\theta}|\tau)\|p(\boldsymbol{\theta})). \quad (4)$$

For $\lambda = 1$ this directly minimizes $D_{\mathrm{KL}}(q(\boldsymbol{\theta}|\tau) \,\|\, p(\boldsymbol{\theta}|\mathcal{D}))$ and thus for sufficiently rich variational families will closely approximate the true Bayes posterior $p(\boldsymbol{\theta}|\mathcal{D})$. However, in practice researchers discovered that using values $\lambda < 1$ provides better predictive performance, with common values shown in the following table.[4]

Reference	KL term weight λ in (4)
(Zhang et al., 2018)	$\lambda \in \{1/2, 1/10\}$
(Bae et al., 2018)	tuning of λ, unspecified
(Osawa et al., 2019)	$\lambda \in \{1/5, 1/10\}$
(Ashukha et al., 2020)	λ from 10^{-5} to 10^{-3}

In Appendix E we show that the KL-weighted ELBO (4) arises from tempering the likelihood part of the posterior.

From the above list we can see that the cold posterior problem has left a trail in the literature, and in fact we are not aware of *any* published work demonstrating well-performing Bayesian deep learning at temperature $T = 1$. We now give details on how we perform accurate Bayesian posterior inference in deep learning models.

3. Bayesian Deep Learning in Practice

In this section we describe how we achieve efficient and accurate simulation of Bayesian neural network posteriors. This section does not contain any major novel contribution but instead combines existing work.

3.1. Posterior Simulation using Langevin Dynamics

To generate approximate parameter samples $\boldsymbol{\theta} \sim p(\boldsymbol{\theta} \,|\, \mathcal{D})$ we consider *Langevin dynamics* over parameters $\boldsymbol{\theta} \in \mathbb{R}^d$ and momenta $\mathbf{m} \in \mathbb{R}^d$, defined by the Langevin stochastic differential equation (SDE),

$$\mathrm{d}\,\boldsymbol{\theta} = \mathbf{M}^{-1}\,\mathbf{m}\,\mathrm{d}t, \quad (5)$$

$$\mathrm{d}\,\mathbf{m} = -\nabla_{\boldsymbol{\theta}} U(\boldsymbol{\theta})\,\mathrm{d}t - \gamma\mathbf{m}\,\mathrm{d}t + \sqrt{2\gamma T}\,\mathbf{M}^{1/2}\,\mathrm{d}\mathbf{W}. \quad (6)$$

Here $U(\boldsymbol{\theta})$ is the *posterior energy* defined in (2), and $T > 0$ is the *temperature*. We use $\mathbf{W}$ to denote a standard multivariate Wiener process, which we can loosely understand as a generalized Gaussian distribution (Särkkä & Solin, 2019; Leimkuhler & Matthews, 2016). The *mass matrix* $\mathbf{M}$ is a preconditioner, and if we use no preconditioner then $\mathbf{M} = I$, such that all $\mathbf{M}$-related terms vanish from the equations. The

[2] E.g., using a Normal prior with temperature T results in a Normal distribution with scaled variance by a factor of T.

[3] For (Li et al., 2016) the tempering with $T = 1/\sqrt{n}$ arises due to an implementation mistake. For (Heek & Kalchbrenner, 2020) we communicated with the authors, and tempering arises due to overcounting data by a factor of 5, approximately justified by data augmentation, corresponding to $T = 1/5$. For (Zhang et al., 2020) the original implementation contains inadvertent tempering, however, the authors added a study of tempering in a revision.

[4] For (Osawa et al., 2019) scaling with λ arises due to their use of a "data augmentation factor" $\rho \in \{5, 10\}$.

friction parameter $\gamma > 0$ controls both the strength of coupling between the moments $\mathbf{m}$ and parameters $\boldsymbol{\theta}$ as well as the amount of injected noise (Langevin, 1908; Leimkuhler & Matthews, 2016). For any friction $\gamma > 0$ the SDE (5–6) has the same limiting distribution, but the choice of friction *does* affect the speed of convergence to this distribution. Simulating the continuous Langevin SDE (5–6) produces a trajectory distributed according to $\exp(-U(\boldsymbol{\theta})/T)$ and the Bayes posterior is recovered for $T = 1$.

3.2. Stochastic Gradient MCMC (SG-MCMC)

Bayesian inference now corresponds to simulating the above SDE (5–6) and this requires numerical discretization. For efficiency *stochastic gradient Markov chain Monte Carlo* (SG-MCMC) methods further approximate $\nabla_{\boldsymbol{\theta}} U(\boldsymbol{\theta})$ with a minibatch gradient (Welling & Teh, 2011; Chen et al., 2014). For a minibatch $B \subset \{1, 2, \ldots, n\}$ we first compute the minibatch average gradient $\tilde{G}(\boldsymbol{\theta})$,

$$\nabla_{\boldsymbol{\theta}} \tilde{G}(\boldsymbol{\theta}) := -\frac{1}{|B|} \sum_{i \in B} \nabla_{\boldsymbol{\theta}} \log p(y_i | x_i, \boldsymbol{\theta}) - \frac{1}{n} \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}), \tag{7}$$

and approximate $\nabla_{\boldsymbol{\theta}} U(\boldsymbol{\theta})$ with the unbiased estimate $\nabla_{\boldsymbol{\theta}} \tilde{U}(\boldsymbol{\theta}) = n \nabla_{\boldsymbol{\theta}} \tilde{G}(\boldsymbol{\theta})$. Here $|B|$ is the minibatch size and n is the training set size; in particular, note that the log prior scales with $1/n$ regardless of the batch size.

The SDE (5–6) is defined in continuous time ($\mathrm{d}t$), and in order to solve the dynamics numerically we have to discretize the time domain (Särkkä & Solin, 2019). In this work we use a simple first-order symplectic Euler discretization, (Leimkuhler & Matthews, 2016), as first proposed for (5–6) by (Chen et al., 2014). Recent work has used more sophisticated discretizations, (Chen et al., 2015; Shang et al., 2015; Heber et al., 2019; Heek & Kalchbrenner, 2020). Applying the symplectic Euler scheme to (5–6) gives the discrete time update equations,

$$\begin{aligned} \mathbf{m}^{(t)} &= (1 - h\gamma)\, \mathbf{m}^{(t-1)} - hn \nabla_{\boldsymbol{\theta}} \tilde{G}(\boldsymbol{\theta}^{(t-1)}) && (8) \\ &\quad + \sqrt{2\gamma h T}\, \mathbf{M}^{1/2}\, \mathbf{R}^{(t)}, && (9) \\ \boldsymbol{\theta}^{(t)} &= \boldsymbol{\theta}^{(t-1)} + h\, \mathbf{M}^{-1} \mathbf{m}^{(t)}, && (10) \end{aligned}$$

where $\mathbf{R}^{(t)} \sim \mathcal{N}_d(0, I_d)$ is a standard Normal vector.

In (8–10), the parameterization is in terms of step size h and friction γ. These quantities are different from typical SGD parameters. In Appendix B we establish an exact correspondence between the SGD learning rate ℓ and momentum decay parameters β and SG-MCMC parameters. For the symplectic Euler discretization of Langevin dynamics, we derive this relationship as $h := \sqrt{\ell/n}$, and $\gamma := (1 - \beta)\sqrt{n/\ell}$, where n is the total training set size.

3.3. Accurate SG-MCMC Simulation

In practice there remain two sources of error when following the dynamics (8–10):

- *Minibatch noise*: $\nabla_{\boldsymbol{\theta}} \tilde{U}(\boldsymbol{\theta})$ is an unbiased estimate of $\nabla_{\boldsymbol{\theta}} U(\boldsymbol{\theta})$ but contains additional estimation variance.
- *Discretization error*: we incur error by following a continuous-time path (5–6) using discrete steps (8–10).

We use two methods to reduce these errors: *preconditioning* and *cyclical time stepping*.

Layerwise Preconditioning. Preconditioning through a choice of matrix $\mathbf{M}$ is a common way to improve the behavior of optimization methods. Li et al. (2016) and Ma et al. (2015) proposed preconditioning for SG-MCMC methods, and in the context of molecular dynamics the use of a matrix $\mathbf{M}$ has a long tradition as well, (Leimkuhler & Matthews, 2016). Li's proposal is an adaptive preconditioner inspired by RMSprop, (Tieleman & Hinton, 2012). Unfortunately, using the discretized Langevin dynamics with a preconditioner $\mathbf{M}(\boldsymbol{\theta})$ that depends on $\boldsymbol{\theta}$ compromises the correctness of the dynamics.[5] We propose a simpler preconditioner that limits the frequency of adaptating $\mathbf{M}$: after a number of iterations we estimate a new preconditioner $\mathbf{M}$ using a small number of batches, say 32, but without updating any model parameters. This preconditioner then remains fixed for a number of iterations, for example, the number of iterations it takes to visit the training set once, i.e. one epoch. We found this strategy to be highly effective at improving simulation accuracy. For details, please see Appendix D.

Cyclical time stepping. The second method to improve simulation accuracy is to decrease the discretization step size h. Chen et al. (2015) studied the consequence of both minibatch noise and discretization error on simulation accuracy and showed that the overall simulation error goes to zero for $h \searrow 0$. While lowering the step size h to a small value would also make the method slow, recently Zhang et al. (2020) propose to perform *cycles* of iterations $t = 1, 2, \ldots$ with a high-to-low step size schedule $h_0\, C(t)$ described by an initial step size h_0 and a function $C(t)$ that starts at $C(1) = 1$ and has $C(L) = 0$ for a cycle length of L iterations. Such cycles retain fast simulation speed in the beginning while accepting simulation error. Towards the end of each cycle however, a small step size ensures an accurate simulation. We use the cosine schedule from (Zhang et al., 2020) for $C(t)$, see Appendix A.

We integrate these two techniques together into a practical SG-MCMC procedure, Algorithm 1. When no preconditioning and no cosine schedule is used ($\mathbf{M} = I$ and $C(t) = 1$ in all iterations) and $T(t) = 0$ this algorithm is equivalent

[5] Li et al. (2016) derives the required correction term, which however is expensive to compute and omitted in practice.

Algorithm 1: Symplectic Euler Langevin scheme.

```
Function SymEulerSGMCMC($\tilde{G}$, $\boldsymbol{\theta}^{(0)}$, $\ell$, $\beta$, $n$, $T$)
    Input: $\tilde{G} : \Theta \to \mathbb{R}$ mean energy function estimate;
           $\boldsymbol{\theta}^{(0)} \in \mathbb{R}^d$ initial parameter; $\ell > 0$ learning
           rate; $\beta \in [0, 1)$ momentum decay; $n$ total
           training set size; $T(t) \geq 0$ temperature
           schedule
    Output: Sequence $\boldsymbol{\theta}^{(t)}$, $t = 1, 2, \ldots$
    $h_0 \leftarrow \sqrt{\ell / n}$                          // SDE time step
    $\gamma \leftarrow (1 - \beta)\sqrt{n / \ell}$                    // friction
    Sample $\mathbf{m}^{(0)} \sim \mathcal{N}_d(0, I_d)$
    $\mathbf{M} \leftarrow I$                                  // Initial $\mathbf{M}$
    for $t = 1, 2, \ldots$ do
        if new epoch then
            $\mathbf{m}_c \leftarrow \mathbf{M}^{-1/2}\,\mathbf{m}^{(t-1)}$
            $\mathbf{M} \leftarrow$ EstimateM($\tilde{G}, \boldsymbol{\theta}^{(t-1)}$)
            $\mathbf{m}^{(t-1)} \leftarrow \mathbf{M}^{1/2}\,\mathbf{m}_c$
        $h \leftarrow C(t)\,h_0$                // Cyclic modulation
        Sample $\mathbf{R}^{(t)} \sim \mathcal{N}_d(0, I_d)$            // noise
        $\mathbf{m}^{(t)} \leftarrow (1 - h\gamma)\,\mathbf{m}^{(t-1)} - hn\nabla_{\boldsymbol{\theta}}\tilde{G}(\boldsymbol{\theta}^{(t-1)}) +$
            $\sqrt{2\gamma h T(t)}\,\mathbf{M}^{1/2}\,\mathbf{R}^{(t)}$
        $\boldsymbol{\theta}^{(t)} \leftarrow \boldsymbol{\theta}^{(t-1)} + h\,\mathbf{M}^{-1}\mathbf{m}^{(t)}$
        if end of cycle then
            yield $\boldsymbol{\theta}^{(t)}$              // Parameter sample
```

to *Tensorflow*'s SGD with momentum (Appendix C).

Coming back to the Cold Posteriors effect, what could explain the poor performance at temperature $T = 1$? With our Bayesian hearts, there are only three possible areas to examine: the inference, the prior, or the likelihood function.

4. Inference: Is it Accurate?

Both the Bayes posterior and the cooled posteriors are all intractable. Moreover, it is plausible that the high-dimensional posterior landscape of a deep network may lead to difficult-to-simulate SDE dynamics (5–6). Our approximate SG-MCMC inference method further has to deal with minibatch noise and produces only a finite sample approximation to the predictive integral (3). Taken together, could the Cold Posteriors effect arise from a poor inference accuracy?

4.1. Hypothesis: Inaccurate SDE Simulation

Inaccurate SDE Simulation Hypothesis: the SDE (5–6) is poorly simulated.

To gain confidence that our SG-MCMC method simulates the posterior accurately, we introduce diagnostics that previously have not been used in the SG-MCMC context:

- **Kinetic temperatures** (Appendix I.1): we report per-variable statistics derived from the moments $\mathbf{m}$. For these so called *kinetic temperatures* we know the exact sampling distribution under Langevin dynamics and compute their 99% confidence intervals.
- **Configurational temperatures** (Appendix I.2): we report per-variable statistics derived from $\langle \boldsymbol{\theta}, \nabla_{\boldsymbol{\theta}} U(\boldsymbol{\theta}) \rangle$. For these *configurational temperatures* we know the expected value under Langevin dynamics.

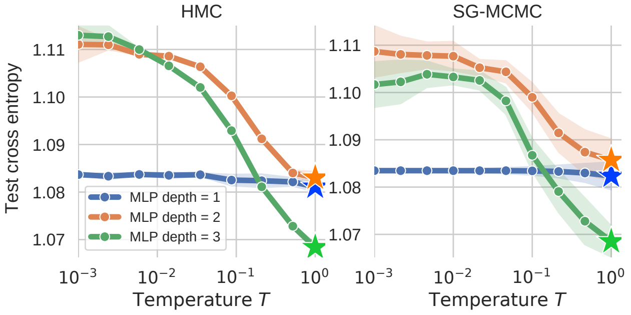

Figure 4. HMC (left) agrees closely with SG-MCMC (right) for synthetic data on multilayer perceptrons. A star indicates the optimal temperature for each model: for the synthetic data sampled from the prior there are no cold posteriors and both sampling methods perform best at $T = 1$.

We propose to use these diagnostics to assess simulation accuracy of SG-MCMC methods. We introduce the diagnostics and our new results in detail in Appendix I.

Inference Diagnostics Experiment: In Appendix J we report a detailed study of simulation accuracy for both models. This study reports accurate simulation for both models when both preconditioning and cyclic time stepping are used. We can therefore with reasonably high confidence rule out a poor simulation of the SDE. All remaining experiments in this paper also pass the simulation accuracy diagnostics.

4.2. Hypothesis: Biased SG-MCMC

Biased SG-MCMC Hypothesis: Lack of accept/reject Metropolis-Hastings corrections in SG-MCMC introduces bias.

In Markov chain Monte Carlo it is common to use an additional accept-reject step that corrects for bias in the sampling procedure. For MCMC applied to deep learning this correction step is too expensive and therefore omitted in SG-MCMC methods, which is valid for small time steps only, (Chen et al., 2015). If accept-reject is computationally feasible the resulting procedure is called *Hamiltonian Monte Carlo* (HMC) (Neal et al., 2011; Betancourt & Girolami, 2015; Duane et al., 1987; Hoffman & Gelman, 2014). Because it provides unbiased simulation, we can consider HMC the *gold standard*, (Neal, 1995). We now compare gold standard HMC against SG-MCMC on a small example where comparison is feasible. We provide details of our HMC setup in Appendix O.

HMC Experiment: we construct a simple setup using a multilayer perceptron (MLP) where by construction $T = 1$ is optimal; such Bayes optimality must hold in expectation if the data is generated by the prior and model that we use for inference, (Berger, 1985). Thus, we can ensure that if the cold posterior effect is observed it must be due to a problem in our inference method. We perform all inference without minibatching ($|B| = n$) and test MLPs of varying number of one to three layers, ten hidden units each, and using the ReLU activation. As HMC implementation we use `tfp.mcmc.HamiltonianMonteCarlo` from *Tensorflow Probability* (Dillon et al., 2017; Lao et al., 2020): Details for our data and HMC are in Appendix N–O.

In Figure 4 the SG-MCMC results agree very well with the HMC results with optimal predictions at $T = 1$, i.e. no cold posteriors are present. For the cases tested we conclude that SG-MCMC is almost as accurate as HMC and the lack of accept-reject correction cannot explain cold posteriors. Appendix O further shows that SG-MCMC and HMC are in good agreement when inspecting the KL divergence of their resulting predictive distributions.

4.3. Hypothesis: Stochastic Gradient Noise

Minibatch Noise Hypothesis: gradient noise from minibatching causes inaccurate sampling at $T = 1$.

Gradient noise due to minibatching can be heavy-tailed and non-Gaussian even for large batch sizes, (Simsekli et al., 2019). Our SG-MCMC method is only justified if the effect of noise will diminish for small time steps. We therefore study the influence of batch size on predictive performance through the following experiment.

Batchsize Experiment: we repeat the original ResNet-20/CIFAR-10 experiment at different temperatures for batch sizes in $\{32, 64, 128, 256\}$ and study the variation of the predictive performance as a function of batch size. Figure 5 and Figure 6 show that while there is a small variation between different batch sizes $T < 1$ remains optimal for all batch sizes. Therefore minibatch noise alone cannot explain the observed poor performance at $T = 1$.

For both ResNet and CNN-LSTM the best cross-entropy is achieved by the smallest batch size of 32 and 16, respectively. The smallest batch size has the *largest* gradient noise. We can interpret this noise as an additional heat source that increases the effective simulation temperature. However, the noise distribution arising from minibatching is anisotropic, (Zhu et al., 2019), and this could perhaps aid generalization. We will not study this hypothesis further here.

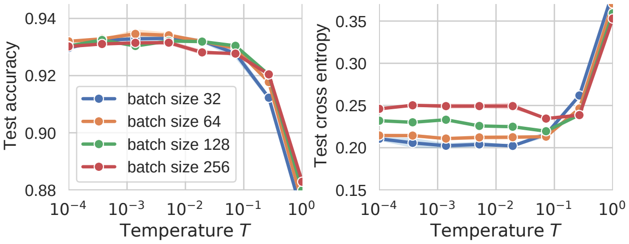

Figure 5. Batch size dependence of the ResNet-20/CIFAR-10 ensemble performance, reporting mean and standard error (3 runs): for all batch sizes the optimal predictions are obtained for $T < 1$.

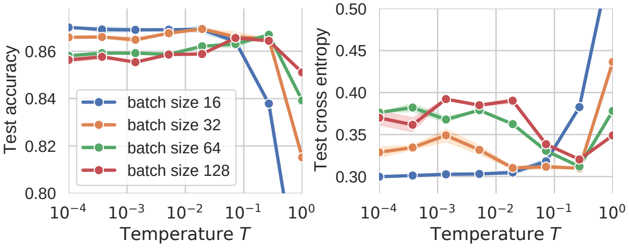

Figure 6. Batch size dependence of the CNN-LSTM/IMDB ensemble performance, reporting mean and standard error (3 runs): for all batch sizes, the optimal performance is achieved at $T < 1$.

4.4. Hypothesis: Bias-Variance Trade-off

Bias-variance Tradeoff Hypothesis: For $T = 1$ the posterior is diverse and there is high variance between model predictions. For $T \ll 1$ we sample nearby modes and reduce prediction variance but increase bias; the variance dominates the error and reducing variance ($T \ll 1$) improves predictive performance.

If this hypothesis were true then simply collecting more ensemble members, $S \to \infty$, would reduce the variance to arbitrary small values and thus fix the poor predictive performance we observe at $T = 1$. Doing so would require running our SG-MCMC schemes for longer—potentially for much longer. We study this question in detail in Appendix F and conclude by an asymptotic analysis that the amount of variance cannot explain cold posteriors.

5. Why Could the Bayes Posterior be Poor?

With some confidence in our approximate inference procedure what are the remaining possibilities that could explain the cold posterior effect? The remaining two places to look at are the likelihood function and the prior.

5.1. Problems in the Likelihood Function?

For Bayesian deep learning we use the same likelihood function $p(y|x, \boldsymbol{\theta})$ as we use for SGD. Therefore, because the same likelihood function works well for SGD it appears an unlikely candidate to explain the cold posterior effect. However, current deep learning models use a number of techniques—such as data augmentation, dropout, and batch normalization—that are not formal likelihood functions. This observations brings us to the following hypothesis.

Dirty Likelihood Hypothesis: Deep learning practices that violate the likelihood principle (batch normalization, dropout, data augmentation) cause deviation from the Bayes posterior.

In Appendix K we give a theory of "*Jensen posteriors*" which describes the likelihood-like functions arising from modern deep learning techniques. We report an experiment (Appendix K.4) that—while slightly inconclusive—demonstrates that cold posteriors remain when a clean likelihood is used in a suitably modified ResNet model; the CNN-LSTM model already had a clean likelihood function.

5.2. Problems with the Prior $p(\boldsymbol{\theta})$?

So far we have used a simple Normal prior, $p(\boldsymbol{\theta}) = \mathcal{N}(0, I)$, as was done in prior work (Zhang et al., 2020; Heek & Kalchbrenner, 2020; Ding et al., 2014; Li et al., 2016; Zhang et al., 2018). But is this a good prior?

One could hope, that perhaps with an informed and structured model architecture, a simple prior could be sufficient in placing prior beliefs on suitable functions, as argued by Wilson (2019). While plausible, we are mildly cautious because there are known examples where innocent looking priors have turned out to be unintentionally highly informative.[6] Therefore, with the cold posterior effect having a track record in the literature, perhaps $p(\boldsymbol{\theta}) = \mathcal{N}(0, I)$ could have similarly unintended effects of placing large prior mass on undesirable functions. This leads us to the next hypothesis.

Bad Prior Hypothesis: The current priors used for BNN parameters are inadequate, unintentionally informative, and their effect becomes stronger with increasing model depths and capacity.

To study the quality of our prior, we study typical functions obtained by sampling from the prior, as is good practice in model criticism, (Gelman et al., 2013).

Prior Predictive Experiment: for our ResNet-20 model we generate samples $\boldsymbol{\theta}^{(i)} \sim p(\boldsymbol{\theta}) = \mathcal{N}(0, I)$ and look at the induced predictive distribution $\mathbb{E}_{x \sim p(x)}[p(y|x, \boldsymbol{\theta}^{(i)})]$ for each parameter sample, using the real CIFAR-10 training images. From Figure 7 we see that typical prior draws produce concentrated class distributions, indicating that the $\mathcal{N}(0, I)$ distribution is a poor prior for the ResNet-20 likelihood. From Figure 8 we can see that the average predictions obtained from such concentrated functions remain close to the uniform class distribution. Taken together, from a subjective Bayesian view $p(\boldsymbol{\theta}) = \mathcal{N}(0, I)$ is a *poor prior*: typical functions produced by this prior place a high probability the same few classes for all x. In Appendix L we carry out another prior predictive study using He-scaling priors, (He et al., 2015), which leads to similar results.

Prior Variance σ Scaling Experiment: in the previous experiment we found that the standard Normal prior is poor. Can the Normal prior $p(\boldsymbol{\theta}) = \mathcal{N}(0, \sigma)$ be fixed by using a more appropriate variance σ? For our ResNet-20 model we employ Normal priors of varying variances. Figure 12 shows that the cold posterior effect is present for all variances considered. Further investigations for known scaling laws in deep networks is given in Appendix L. The cold posterior effect cannot be resolved by using the right scaling of the Normal prior.

Training Set Size n Scaling Experiment: the posterior energy $U(\boldsymbol{\theta})$ in (2) sums over all n data log-likelihoods but adds $\log p(\boldsymbol{\theta})$ only once. This means that the influence of $\log p(\boldsymbol{\theta})$ vanishes at a rate of $1/n$ and thus the prior will exert its strongest influence for small n. We now study what happens for small n by comparing the Bayes predictive under a $\mathcal{N}(0, I)$ prior against performing SGD maximum a posteriori (MAP) estimation on the *same* log-posterior.[7]

Figure 9 and Figure 10 show the predictive performance for ResNet-20 on CIFAR-10 and CNN-LSTM on IMDB, respectively. These results differ markedly between the two models and datasets: for ResNet-20 / CIFAR-10 the Bayes posterior at $T = 1$ degrades gracefully for small n, whereas SGD suffers large losses in test cross-entropy for small n. For CNN-LSTM / IMDB predictions from the Bayes posterior at $T = 1$ deteriorate quickly in both test accuracy and cross entropy. In all these runs SG-MCMC and SGD/MAP work with the same $U(\boldsymbol{\theta})$ and the difference is between integration and optimization. The results are inconclusive but somewhat implicate the prior in the cold posterior effect: as n becomes small there is an increasing difference between the cross-entropy achieved by the Bayes prediction and the SGD estimate, for large n the SGD estimate performs better.

Capacity Experiment: we consider a MLP using a $\mathcal{N}(0, I)$ prior and study the relation of the network capacity to the cold posterior effect. We train MLPs of varying depth (number of layers) and width (number of units per layer) at different temperatures on CIFAR-10. Figure 11 shows that for increasing capacity the cold posterior effect becomes more prominent. This indicates a connection between model capacity and strength of the cold posterior effect.

[6] A shocking example in the Dirichlet-Multinomial model is given by Nemenman et al. (2002). Importantly the unintended effect of the prior was not recognized when the model was originally proposed by Wolpert & Wolf (1995).

[7] For SGD we minimize $U(\boldsymbol{\theta})/n$.

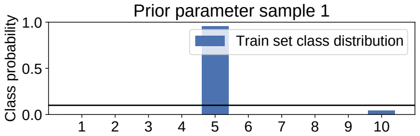

Figure 7. ResNet-20/CIFAR-10 typical prior predictive distributions for 10 classes under a $\mathcal{N}(0, I)$ prior averaged over the entire training set, $\mathbb{E}_{x\sim p(x)}[p(y|x, \boldsymbol{\theta}^{(i)})]$. Each plot is for one sample $\boldsymbol{\theta}^{(i)} \sim \mathcal{N}(0, I)$ from the prior. Given a sample $\boldsymbol{\theta}^{(i)}$ the average training data class distribution is highly concentrated around the same classes for all x.

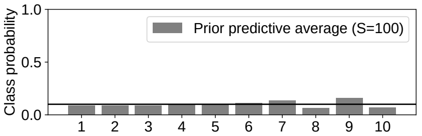

Figure 8. ResNet-20/CIFAR-10 prior predictive $\mathbb{E}_{x\sim p(x)}[\mathbb{E}_{\boldsymbol{\theta}\sim p(\boldsymbol{\theta})}[p(y|x, \boldsymbol{\theta})]]$ over 10 classes, estimated using $S = 100$ prior samples $\boldsymbol{\theta}^{(i)}$ and all training images.

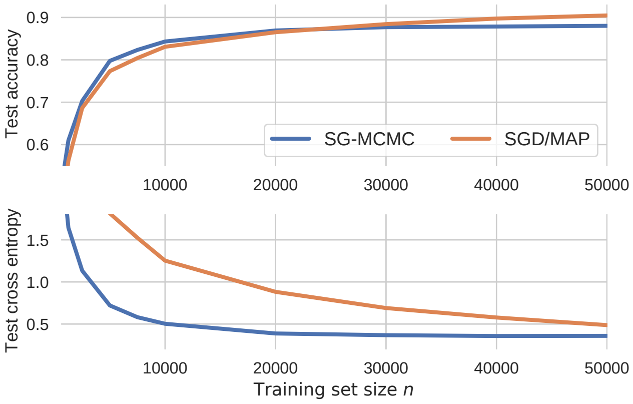

Figure 9. ResNet-20/CIFAR-10 predictive performance as a function of training set size n. The Bayes posterior ($T = 1$) degrades gracefully as n decreases, whereas SGD/MAP performs worse.

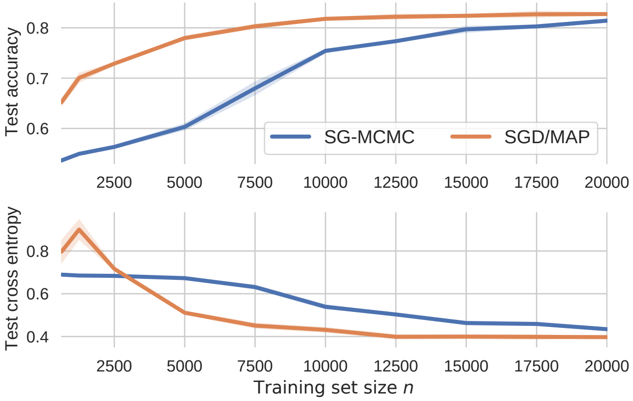

Figure 10. CNN-LSTM/IMDB predictive performance as a function of training set size n. The Bayes posterior ($T = 1$) suffers more than the SGD performance, indicating a problematic prior.

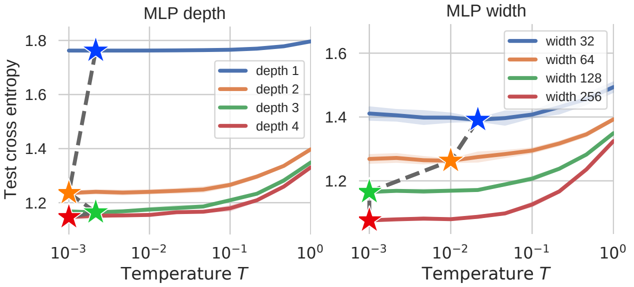

Figure 11. MLP of different capacities (depth and width) on CIFAR-10. Left: we fix the width to 128 and vary the depth. Right: we fix the depth to 3 and vary the width. Increasing capacity lowers the optimal temperature.

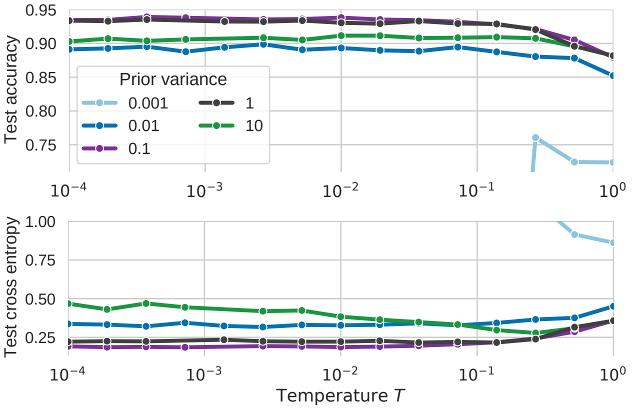

Figure 12. ResNet-20/CIFAR-10 predictive performance as a function of temperature T for different priors $p(\boldsymbol{\theta}) = \mathcal{N}(0, \sigma)$. The cold posterior effect is present for all choices of the prior variance σ. For all models the optimal temperature is significantly smaller than one and for $\sigma = 0.001$ the performance is poor for all temperatures. There is no "simple" fix of the prior.

5.3. Inductive Bias due to SGD?

> **Implicit Initialization Prior in SGD**: The inductive bias from initialization is strong and beneficial for SGD but harmed by SG-MCMC sampling.

Optimizing neural networks via SGD with a suitable initialization is known to have a beneficial inductive bias leading to good local optima, (Masters & Luschi, 2018; Mandt et al., 2017). Does SG-MCMC perform worse due to decreasing the influence of that bias? We address this question by the following experiment. We first run SGD until convergence, then switch over to SG-MCMC sampling for 500 epochs (10 cycles), and finally switch back to SGD again. Figure 13 shows that SGD initialized by the last model of the SG-MCMC sampling dynamics recovers the same performance as vanilla SGD. This indicates that the beneficial initialization bias for SGD is not destroyed by SG-MCMC. Details can be found in Appendix H.

6. Alternative Explanations?

Are there other explanations we have not studied in this work?

Masegosa Posteriors. One exciting avenue of future exploration was provided to us after submitting this work: a compelling analysis of the failure to predict well under the Bayes posterior is given by Masegosa (2019). In his analysis he first follows Germain et al. (2016) in identifying the Bayes posterior as a solution of a loose PAC-Bayes generalization bound on the predictive cross-entropy. He then uses recent results demonstrating improved Jensen inequalities, (Liao & Berg, 2019), to derive alternative posteriors. These alternative posteriors are *not* Bayes posteriors and in fact explicitly encourage diversity among ensemble member predictions. Moreover, the alternative posteriors can be shown to dominate the predictive performance achieved by the Bayes posterior when the model is misspecified. We believe that these new "Masegosa-posteriors", while not explaining cold posteriors fully, may provide a more desirable approximation target than the Bayes posterior. In addition, the Masegosa-posterior is compatible with both variational and SG-MCMC type algorithms.

Tempered observation model? In (Wilson & Izmailov, 2020, Section 8.3) it is claimed that cold posteriors in one model correspond to untempered ($T = 1$) Bayes posteriors in a modified model by a simple change of the likelihood function. If this were the case, this would resolve the cold posterior problem and in fact point to a systematic way how to improve the Bayes posterior in many models. However, the argument in (Wilson & Izmailov, 2020) is wrong, which we demonstrate and discuss in detail in Appendix M.

7. Related Work on Tempered Posteriors

Statisticians have studied *tempered* or *fractional* posteriors for $T > 1$. Motivated by the behavior of Bayesian inference in *misspecified* models (Grünwald et al., 2017; Jansen, 2013) develop the *SafeBayes* approach and Bhattacharya et al. (2019) develops *fractional posteriors* with the goal of slowing posterior concentration. The use of multiple temperatures $T > 1$ is also common in Monte Carlo simulation in the presence of rough energy landscapes, e.g. (Earl & Deem, 2005; Sugita & Okamoto, 1999; Swendsen & Wang, 1986). However, the purpose of such tempering is to aid in accurate sampling at a desired target temperature, but not in changing the target distribution. (Mandt et al., 2016) studies temperature as a latent variable in the context of variational inference and shows that models often select temperatures different from one.

8. Conclusion

Our work has raised the question of cold posteriors but we did not fully resolve nor fix the cause for the cold posterior phenomenon. Yet our experiments suggest the following.

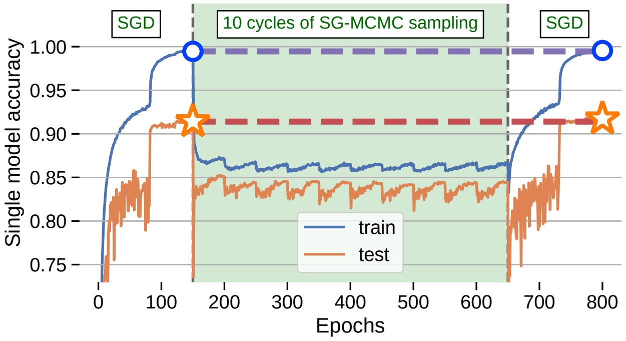

Figure 13. Do the SG-MCMC dynamics harm a beneficial initialization bias used by SGD? We first train a ResNet-20 on CIFAR-10 via SGD, then switch over to SG-MCMC sampling and finally switch back to SGD optimization. We report the single-model test accuracy of SGD and the SG-MCMC chain as function of epochs. SGD recovers from being initialized by the SG-MCMC state.

SG-MCMC is accurate enough: our experiments (Section 4–5) and novel diagnostics (Appendix I) indicate that current SG-MCMC methods are robust, scalable, and accurate enough to provide good approximations to parameter posteriors in deep nets.

Cold posteriors work: while we do not fully understand cold posteriors, tempered SG-MCMC ensembles provide a way to train ensemble models with improved predictions compared to individual models. However, taking into account the added computation from evaluating ensembles, there may be more practical methods, (Lakshminarayanan et al., 2017; Wen et al., 2019; Ashukha et al., 2020).

More work on priors for deep nets is needed: the experiments in Section 5.2 implicate the prior $p(\boldsymbol{\theta})$ in the cold posterior effect, although the prior may not be the only cause. Our investigations fail to produce a "simple" fix based on scaling the prior variance appropriately. Future work on suitable priors for Bayesian neural networks is needed, building on recent advances, (Sun et al., 2019; Pearce et al., 2019; Flam-Shepherd et al., 2017; Hafner et al., 2018).

Acknowledgements. We would like to thank Dustin Tran for reading multiple drafts and providing detailed feedback on the work. We also thank the four anonymous ICML 2020 reviewers for their detailed and helpful feedback.

References

Ashukha, A., Lyzhov, A., Molchanov, D., and Vetrov, D. Pitfalls of in-domain uncertainty estimation and ensembling in deep learning. In *Eigth International Conference on Learning Representations (ICLR 2020)*, 2020.

Bae, J., Zhang, G., and Grosse, R. Eigenvalue corrected noisy natural gradient. *arXiv preprint arXiv:1811.12565*, 2018.

Baldock, R. J. and Marzari, N. Bayesian neural networks at finite temperature. *arXiv preprint, arXiv:1904.04154*, 2019.

Barber, D. and Bishop, C. M. Ensemble learning for multilayer networks. In *Advances in neural information processing systems*, pp. 395–401, 1998.

Berger, J. O. *Statistical decision theory and Bayesian analysis*. Springer, 1985.

Betancourt, M. and Girolami, M. Hamiltonian Monte Carlo for hierarchical models. *Current trends in Bayesian methodology with applications*, 79:30, 2015.

Bhattacharya, A., Pati, D., Yang, Y., et al. Bayesian fractional posteriors. *The Annals of Statistics*, 47(1):39–66, 2019.

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. Weight uncertainty in neural network. 37:1613–1622, 2015.

Brier, G. W. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950.

Chen, C., Ding, N., and Carin, L. On the convergence of stochastic gradient MCMC algorithms with high-order integrators. In *Advances in Neural Information Processing Systems*, pp. 2278–2286, 2015.

Chen, T., Fox, E., and Guestrin, C. Stochastic gradient Hamiltonian Monte Carlo. In *International Conference on Machine Learning*, pp. 1683–1691, 2014.

Dillon, J. V., Langmore, I., Tran, D., Brevdo, E., Vasudevan, S., Moore, D., Patton, B., Alemi, A., Hoffman, M., and Saurous, R. A. Tensorflow distributions. *arXiv preprint arXiv:1711.10604*, 2017.

Ding, N., Fang, Y., Babbush, R., Chen, C., Skeel, R. D., and Neven, H. Bayesian sampling using stochastic gradient thermostats. In *Advances in neural information processing systems*, pp. 3203–3211, 2014.

Duane, S., Kennedy, A. D., Pendleton, B. J., and Roweth, D. Hybrid Monte Carlo. *Physics letters B*, 195(2):216–222, 1987.

Earl, D. J. and Deem, M. W. Parallel tempering: Theory, applications, and new perspectives. *Physical Chemistry Chemical Physics*, 7(23):3910–3916, 2005.

Flam-Shepherd, D., Requeima, J., and Duvenaud, D. Mapping gaussian process priors to bayesian neural networks. In *NIPS Bayesian deep learning workshop*, 2017.

Fushiki, T. et al. Bootstrap prediction and Bayesian prediction under misspecified models. *Bernoulli*, 11(4):747–758, 2005.

Geisser, S. *An Introduction to Predictive Inference*. Chapman and Hall, New York, 1993.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. *Bayesian data analysis*. Chapman and Hall/CRC, 2013.

Germain, P., Bach, F., Lacoste, A., and Lacoste-Julien, S. Pac-bayesian theory meets bayesian inference. In *Advances in Neural Information Processing Systems*, pp. 1884–1892, 2016.

Grünwald, P., Van Ommen, T., et al. Inconsistency of Bayesian inference for misspecified linear models, and a proposal for repairing it. *Bayesian Analysis*, 12(4): 1069–1103, 2017.

Hafner, D., Tran, D., Lillicrap, T., Irpan, A., and Davidson, J. Noise contrastive priors for functional uncertainty. *arXiv preprint arXiv:1807.09289*, 2018.

He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

Heber, F., Trstanova, Z., and Leimkuhler, B. Tati-thermodynamic analytics toolkit: Tensorflow-based software for posterior sampling in machine learning applications. *arXiv preprint arXiv:1903.08640*, 2019.

Heek, J. and Kalchbrenner, N. Bayesian inference for large scale image classification. In *International Conference on Learning Representations (ICLR 2020)*, 2020.

Hinton, G. and Van Camp, D. Keeping neural networks simple by minimizing the description length of the weights. In *in Proc. of the 6th Ann. ACM Conf. on Computational Learning Theory*, 1993.

Hoffman, M. D. and Gelman, A. The no-u-turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15(1): 1593–1623, 2014.

Jansen, L. Robust Bayesian inference under model misspecification, 2013. Master thesis.

Komaki, F. On asymptotic properties of predictive distributions. *Biometrika*, 83(2):299–313, 1996.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems 30*. 2017.

Langevin, P. Sur la théorie du mouvement brownien. *Compt. Rendus*, 146:530–533, 1908.

Lao, J., Suter, C., Langmore, I., Chimisov, C., Saxena, A., Sountsov, P., Moore, D., Saurous, R. A., Hoffman, M. D., and Dillon, J. V. tfp.mcmc: Modern Markov chain Monte Carlo tools built for modern hardware, 2020.

Leimkuhler, B. and Matthews, C. *Molecular Dynamics*. Springer, 2016.

Leimkuhler, B., Matthews, C., and Vlaar, T. Partitioned integrators for thermodynamic parameterization of neural networks. *arXiv preprint arXiv:1908.11843*, 2019.

Li, C., Chen, C., Carlson, D., and Carin, L. Preconditioned stochastic gradient Langevin dynamics for deep neural networks. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

Liao, J. and Berg, A. Sharpening jensen's inequality. *The American Statistician*, 73(3):278–281, 2019.

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

MacKay, D. J. et al. Ensemble learning and evidence maximization. In *Proc. Nips*, volume 10, pp. 4083. Citeseer, 1995.

Mandt, S., McInerney, J., Abrol, F., Ranganath, R., and Blei, D. M. Variational tempering. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, AISTATS*, JMLR Workshop and Conference Proceedings, 2016.

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

Masegosa, A. R. Learning under model misspecification: Applications to variational and ensemble methods. *arXiv preprint, arXiv:19012.08335*, 2019.

Masters, D. and Luschi, C. Revisiting small batch training for deep neural networks. *arXiv preprint arXiv:1804.07612*, 2018.

Naeini, M. P., Cooper, G., and Hauskrecht, M. Obtaining well calibrated probabilities using bayesian binning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

Neal, R. M. *Bayesian learning for neural networks*. PhD thesis, University of Toronto, 1995.

Neal, R. M. et al. MCMC using Hamiltonian dynamics. *Handbook of Markov chain Monte Carlo*, 2(11):2, 2011.

Nemenman, I., Shafee, F., and Bialek, W. Entropy and inference, revisited. In *Advances in neural information processing systems*, pp. 471–478, 2002.

Osawa, K., Swaroop, S., Jain, A., Eschenhagen, R., Turner, R. E., Yokota, R., and Khan, M. E. Practical deep learning with Bayesian principles. *arXiv preprint arXiv:1906.02506*, 2019.

Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J. V., Lakshminarayanan, B., and Snoek, J. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems (NeurIPS 2019)*, 2019.

Pearce, T., Zaki, M., Brintrup, A., and Neely, A. Expressive priors in bayesian neural networks: Kernel combinations and periodic functions. *arXiv preprint arXiv:1905.06076*, 2019.

Ramamoorthi, R. V., Sriram, K., and Martin, R. On posterior concentration in misspecified models. *Bayesian Anal.*, 10 (4):759–789, 12 2015. doi: 10.1214/15-BA941.

Särkkä, S. and Solin, A. *Applied stochastic differential equations*, volume 10. Cambridge University Press, 2019.

Shang, X., Zhu, Z., Leimkuhler, B., and Storkey, A. J. Covariance-controlled adaptive Langevin thermostat for large-scale Bayesian sampling. In *Advances in Neural Information Processing Systems*, pp. 37–45, 2015.

Simsekli, U., Sagun, L., and Gurbuzbalaban, M. A tail-index analysis of stochastic gradient noise in deep neural networks. *arXiv preprint arXiv:1901.06053*, 2019.

Sugita, Y. and Okamoto, Y. Replica-exchange molecular dynamics method for protein folding. *Chemical physics letters*, 314(1-2):141–151, 1999.

Sun, S., Zhang, G., Shi, J., and Grosse, R. Functional variational Bayesian neural networks. *arXiv preprint arXiv:1903.05779*, 2019.

Sutskever, I., Martens, J., Dahl, G., and Hinton, G. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pp. 1139–1147, 2013.

Swendsen, R. H. and Wang, J.-S. Replica Monte Carlo simulation of spin-glasses. *Physical review letters*, 57 (21):2607, 1986.

Tieleman, T. and Hinton, G. Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude. Coursera: Neural Networks for Machine Learning, 2012.

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

Wen, Y., Tran, D., and Ba, J. BatchEnsemble: Efficient ensemble of deep neural networks via rank-1 perturbation. 2019. Bayesian deep learning workshop 2019.

Wilson, A. G. The case for Bayesian deep learning. *NYU Courant Technical Report*, 2019. Accessible at https://cims.nyu.edu/~andrewgw/caseforbdl.pdf.

Wilson, A. G. and Izmailov, P. Bayesian deep learning and a probabilistic perspective of generalization. *arXiv preprint arXiv:2002.08791*, 2020.

Wolpert, D. H. and Wolf, D. R. Estimating functions of probability distributions from a finite set of samples. *Physical Review E*, 52(6):6841, 1995.

Zhang, G., Sun, S., Duvenaud, D., and Grosse, R. Noisy natural gradient as variational inference. *International Conference on Machine Learning*, 2018.

Zhang, R., Li, C., Zhang, J., Chen, C., and Wilson, A. G. Cyclical stochastic gradient MCMC for Bayesian deep learning. In *International Conference on Learning Representations (ICLR 2020)*, 2020.

Zhu, Z., Wu, J., Yu, B., Wu, L., and Ma, J. The anisotropic noise in stochastic gradient descent: Its behavior of escaping from sharp minima and regularization effects. In *Proceedings of the 36th International Conference on Machine Learning (ICML 2019)*, 2019.