

**Code:** github.com/google-research/google-research/tree/master/cold posterior bnn

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

Florian Wenzel (Google Research Berlin)

Joint first authors: Kevin Roth, Bas Veeling, and: Jakub Swiatkowski, Linh Tran, Stephan Mandt, Jasper Snoek, Tim Salimans, Rodolphe Jenatton, Sebastian Nowozin



#### Bayesian Deep Learning



Jianyi Zhang

Duke University

jz318@duke.edu

**Goal:** enable **Bayesian inference for deep networks** to improve robustness of predictions!

Active research field where most work focuses on **improving approximate inference** to get closer to the Bayes posterior

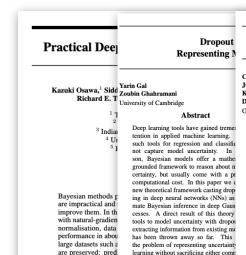
Noisy Natur

estimates lead to more efficient ext

active learning, and intrinsic motivat

forcement learning.

1. Introduction



complexity or test accuracy. We perfor

tensive study of the properties of drop

certainty. Various network architectures

linearities are assessed on tasks of re-

and classification, using MNIST as an

of-distribution data :

This work enables p

principles. A PyToro

#### Guodong Zhang \*12 Charles Blundell Julien Cornebise Koray Kavukcuoglu Abstract Daan Wierstra Variational Bayesian neural nets comb Google DeepMind ibility of deep learning with Bayesian estimation. Unfortunately, there is between cheap but simple variation Abstrac (e.g. fully factorized) or expensive : cated inference procedures. We show We introduce a new, efficie gradient ascent with adaptive weight r backpropagation-compatible itly fits a variational posterior to ma ing a probability distribution evidence lower bound (ELBO). This a neural network, called Bay lows us to train full-covariance, fully regularises the weights by or matrix-variate Gaussian variations using noisy versions of natural grad pression cost, known as the and K-FAC, respectively, making it ergy or the expected lower bo scale up to modern ConvNets. On: likelihood. We show that t gression benchmarks, our noisy K-FA of regularisation yields comp makes better predictions and match to dropout on MNIST class nian Monte Carlo's predictive varia demonstrate how the learnt than existing methods. Its improved

weights can be used to imp

in non-linear regression prob

weight uncertainty can be

exploration-exploitation trac-

ment learning.

# Bayesia BAY

Jonathan Heek Google Brain Amsterdam jheek@google.com

Bayesian inference promises to groun ral networks. It promises to be robus cedure and the space of hyperparame uncertainty that can enhance decisio fairness. Markov Chain Monte Carlé ence by generating samples from the ters. Despite the theoretical advantag between MCMC and optimization me has so far lagged behind optimization We aim to fill this gap and introduce that estimates and is able to sample fr dynamically adjusts the amount of meter undate in order to compensate f

#### CYCLICAL STOCHASTIC GRADIENT MCMC FOR BAYESIAN DEEP LEARNING

Chunvuan Li

Ruqi Zhang Cornell University

University at Buffalo, SUNY

changyou@buffalo.edu

rz297@cornell.edu chunyl@microsoft.com

Changvou Chen Andrew Gordon

Andrew Gordon Wilson
New York University
andrewgw@cims.nyu.edu

#### ABSTRACT

The posteriors over neural network weights are high dimensional and multimodal. Each mode typically characterizes a meaningfully different representation of the data. We develop Cyclical Stochastic Gradient MCMC (SG-MCMC) to automatically explore such distributions. In particular, we propose a cyclical stepsize schedule, where larger steps discover new modes, and smaller steps characterize each mode. We also prove non-asymptotic convergence of our proposed algorithm. Moreover, we provide extensive experimental results, including ImageNet, to demonstrate the scalability and effectiveness of cyclical SG-MCMC in learning complex multimodal distributions, especially for fully Bayesian inference with modern deep neural networks.

#### ABSTRACT

Microsoft Research, Redmond

1 Introduction



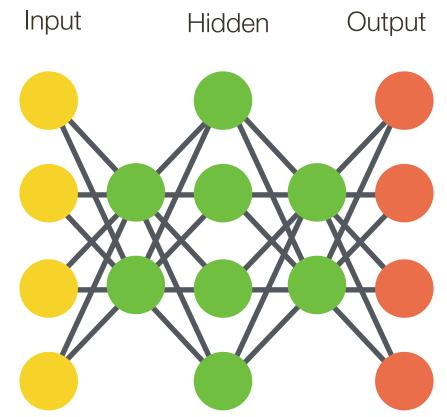
# But is the Bayes posterior actually good?



**Neural Network** 

$$p(\mathcal{D}|\boldsymbol{\theta}) = p(y_i|x_i,\boldsymbol{\theta})$$

Different models obtained by different  $oldsymbol{ heta}$ 



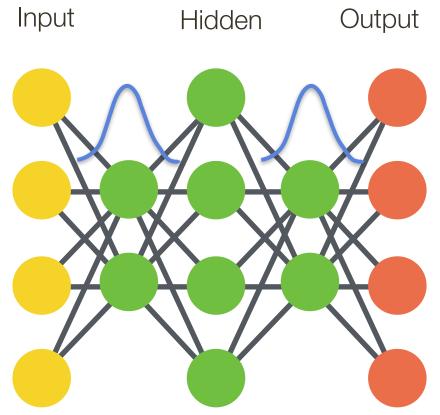


## **Bayesian** Neural Network

$$p(\boldsymbol{\theta}, \mathcal{D}) = p(\underline{y_i}|\underline{x_i}, \boldsymbol{\theta}) p(\boldsymbol{\theta})$$

Posterior: Distribution over likely models given the data

$$p(\boldsymbol{\theta}|\mathcal{D})$$

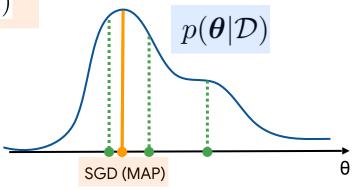


#### **BNNs: Predictions**



In standard deep learning we optimize  $U(\theta)$ 

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



BNNs use samples from the posterior (ensemble of models)

$$\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3, \dots \sim p(\boldsymbol{\theta}|\mathcal{D}) \propto \exp(-U(\boldsymbol{\theta}))$$

#### **BNNs:** Predictions

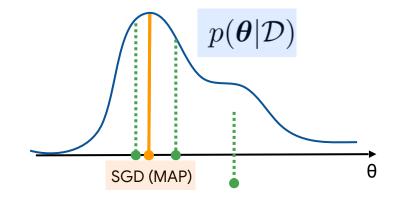


#### Predict by using an average of models

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

$$pprox \sum p(y|x, \boldsymbol{\theta}_s)$$

$$\approx \sum_{s} p(y|x, \boldsymbol{\theta}_{s})$$
$$\boldsymbol{\theta}_{1}, \boldsymbol{\theta}_{2}, \boldsymbol{\theta}_{3}, \dots \sim p(\boldsymbol{\theta}|\mathcal{D})$$



In this talk: A model is good if it predicts well (e.g. low cross entropy loss)



#### **Promises of BNNs\*:**

- Robustness in generalization
- Better uncertainty quantification (calibration)
- Enables new deep learning applications (continual learning, sequential decision making, ...)

<sup>\* [</sup>e.g., Neal 1995, Gal et al. 2016, Wilson 2019, Ovadia et al. 2019].



# But in practice BNNs are rarely used!





#### In practice:

- Often, the Bayes posterior is worse than SGD point estimates
- But Bayes predictions can be improved by the use of the

#### **Cold Posterior\***

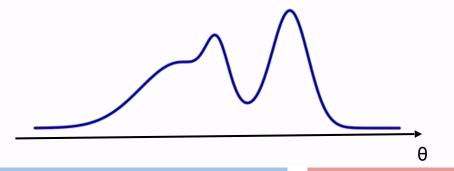
$$p(\boldsymbol{\theta}|\mathcal{D}) \propto \exp(-U(\boldsymbol{\theta})/T)$$

For temperature *T<1*: We **sharpen the posterior** (over-count evidence)

<sup>\*</sup>Explicitly (or implicitly) **used by most recent Bayesian DL papers** [e.g., Li et al. 2016, Zhang et al. 2020, Ashukha et al. 2020].



Temperature: 1.00



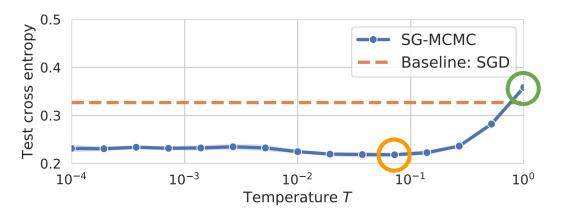
#### **Cold Posterior**

$$p(\boldsymbol{\theta}|\mathcal{D}) \propto \exp(-U(\boldsymbol{\theta})/T)$$

For temperature *T<1*: We **sharpen the posterior** (over-count evidence)

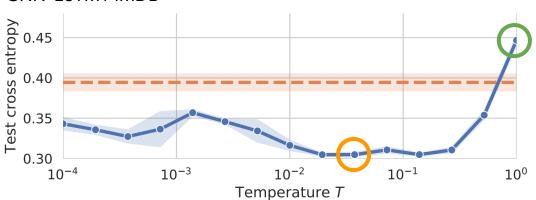
#### ResNet-20 / CIFAR-10





- True Bayes posterior
- Optimal cold posterior

#### CNN-LSTM / IMDB



Florian Wenzel, 15 June 2020



13

# The cold posterior sharply deviates from the Bayesian paradigm.

What is the use of more accurate posterior approximations if the posterior is poor?



# Our paper: Hypothesis for the origin of the improved performance of cold posteriors

#### Inference

Inaccurate SDE Simulation?

Bias of SG-MCMC?

Minibatch noise (which is not Gaussian)?

Bias-variance tradeoff induced by cold posterior?

#### Likelihood

Dirty likelihoods?

(batch-normalization, dropout, data augmentation)

#### Prior

Current priors used for BNN parameters are poor?

The effect becomes stronger with increasing model depths and capacity?



# Our paper: Hypothesis for the origin of the improved performance of cold posteriors



#### Inference

Inaccurate SDE Simulation?

Bias of SG-MCMC?

Minibatch noise (which is not Gaussian)?

Bias-variance tradeoff induced by cold posterior?



#### Likelihood

Dirty likelihoods?

(batch-normalization, dropout, data augmentation)



#### Prior

Current priors used for BNN parameters are poor?

The effect becomes stronger with increasing model depths and capacity?

15



# Our paper: Hypothesis for the origin of the improved performance of cold posteriors

#### Inference

Inaccurate SDE Simulation?

Bias of SG-MCMC?

Minibatch noise (which is not Gaussian)?

Bias-variance tradeoff induced by cold posterior?

#### Likelihood

Dirty likelihoods?

(batch-normalization, dropout, data augmentation)

#### Prior

Current priors used for BNN parameters are poor?

The effect becomes stronger with increasing model depths and capacity?

#### Inference: Is it accurate?



1. How to compute the posterior (inference)?

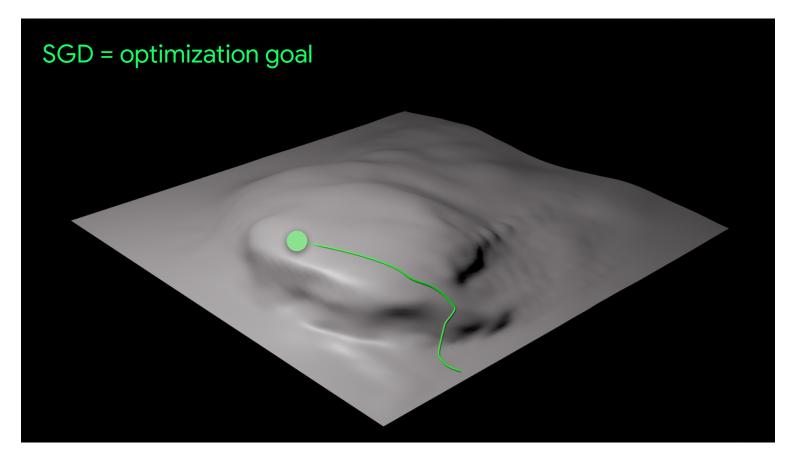
#### Sample from the posterior using <u>SG-MCMC</u> methods

Not covered: Approximate posterior using variational inference

2. Does inaccurate inference lead to the cold posterior effect?

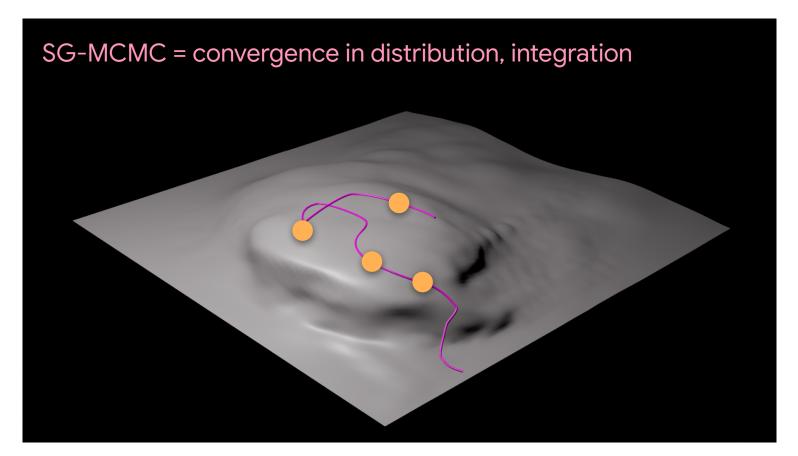
#### **SG-MCMC:** Stochastic Gradient Markov Chain Monte Carlo





#### **SG-MCMC:** Stochastic Gradient Markov Chain Monte Carlo





#### **Stochastic Gradient Markov Chain Monte Carlo**



Langevin Dynamics: one-slide refresher

$$d\theta = \mathbf{M}^{-1}\mathbf{m}dt$$

$$d\mathbf{m} = -\nabla_{\theta}U(\theta)dt - \gamma\mathbf{m}dt + \sqrt{2\gamma T}\mathbf{M}^{1/2}d\mathbf{W}$$

- Simulating SDE has **stationary distribution** proportional to exp(-U(θ) / T) [Langevin, 1908], [Leimkuhler and Matthews, "Molecular Dynamics", 2016]
- Parameters  $\theta$ , moments m, mass matrix M > 0, friction  $\gamma$  > 0
- "Solving SDE" 

  ⇔ obtain one random continuous-time path

#### **Stochastic Gradient Markov Chain Monte Carlo**



Symplectic Euler (Discretized version of SDE)

$$\mathbf{m}^{(t)} = (1 - h\gamma)\mathbf{m}^{(t-1)} - hn\nabla_{\boldsymbol{\theta}}\tilde{G}\left(\boldsymbol{\theta}^{(t-1)}\right) + \sqrt{2\gamma hT}\mathbf{M}^{1/2}\mathcal{N}\left(0, I\right)$$
$$\boldsymbol{\theta}^{(t)} = \boldsymbol{\theta}^{(t-1)} + h\mathbf{M}^{-1}\mathbf{m}^{(t)}$$

SGD with Momentum

Gaussian Noise

scaled by temperature

#### **Stochastic Gradient Markov Chain Monte Carlo**

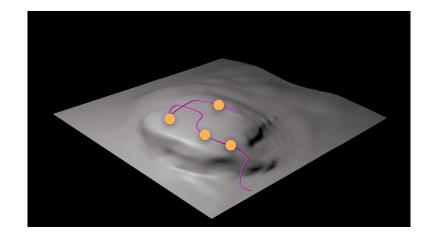


The discretization scheme leads to

#### approximate samples from the posterior

$${m heta}^{(1)}, {m heta}^{(2)}, {m heta}^{(3)}, \dots$$

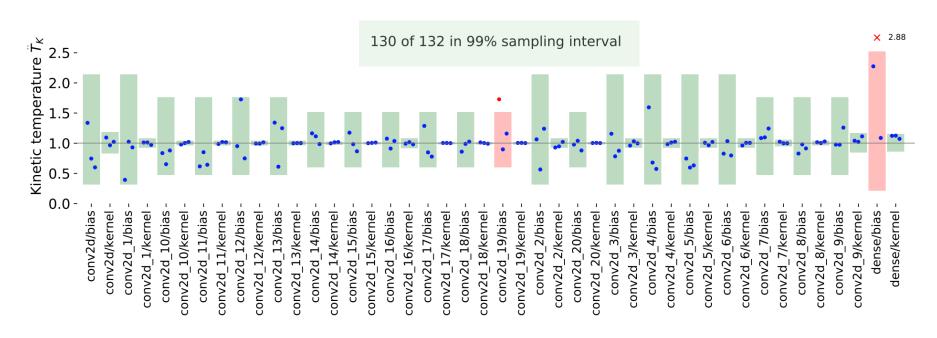
Is this accurate enough?



#### **Novel diagnostics** for SG-MCMC



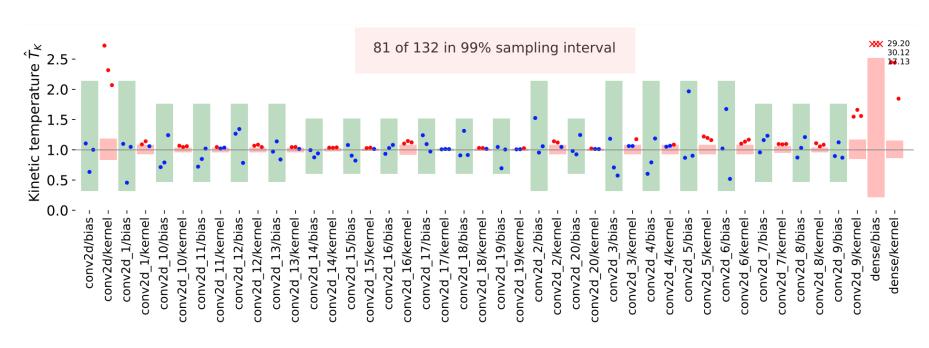
#### Diagnostics check out!



#### **Novel diagnostics** for SG-MCMC



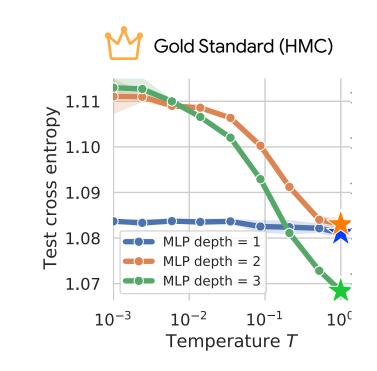
#### Diagnostics fail.

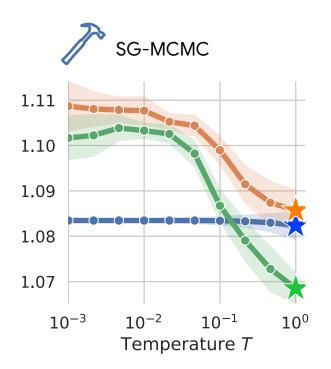


## SG-MCMC works well enough!



#### Synthetic data generated from an MLP





## SG-MCMC inference works well enough!



#### Inference

Inaccurate SDE Simulation?



Bias of SG-MCMC?

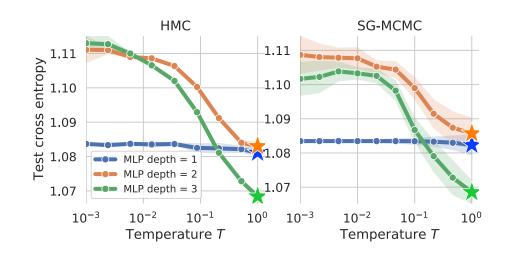


Minibatch noise (which is not Gaussian)?



Bias-variance tradeoff induced by cold posterior?





### SG-MCMC inference works well enough!



#### Inference

Inaccurate SDE Simulation?



Bias of SG-MCMC?

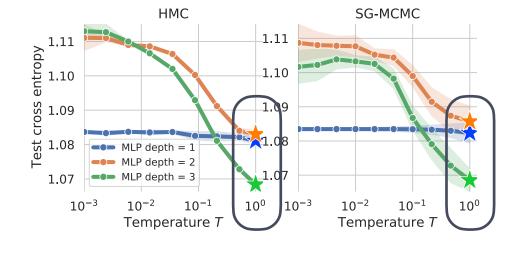


Minibatch noise (which is not Gaussian)?



Bias-variance tradeoff induced by cold posterior?





If the model is **well-specified**, T=1 is optimal.



But for real-world data T<1 is better!

Florian Wenzel, 15 June 2020



#### The cold posterior effect

# Why does the cold posterior perform better than the true Bayes posterior?

#### Problems with the prior?



#### Prior

Current priors used for BNN parameters are poor?

$$p(\boldsymbol{\theta}) = \mathcal{N}(0, I)$$

The effect becomes stronger with increasing model depths and capacity?



Draw from prior

$$\boldsymbol{\theta}^{(i)} \sim p(\boldsymbol{\theta}) = \mathcal{N}(0, I)$$

Induced predictive distribution

$$\mathbb{E}_{x \sim p(x)} \left[ p\left(y|x, \boldsymbol{\theta}^{(i)}\right) \right]$$



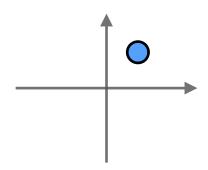
Draw from prior

$$\boldsymbol{\theta}^{(i)} \sim p(\boldsymbol{\theta}) = \mathcal{N}(0, I)$$

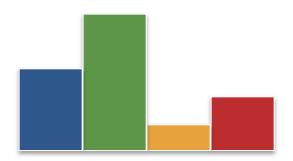
Induced predictive distribution

$$\mathbb{E}_{x \sim p(x)} \left[ p\left(y|x, \boldsymbol{\theta}^{(i)}\right) \right]$$

Model parameters



Class Probabilities





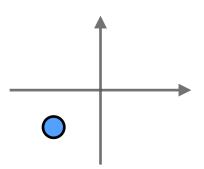
#### Draw from prior

$$\boldsymbol{\theta}^{(i)} \sim p(\boldsymbol{\theta}) = \mathcal{N}(0, I)$$

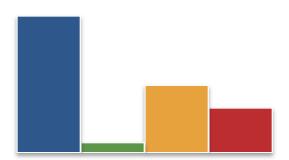
#### Induced predictive distribution

$$\mathbb{E}_{x \sim p(x)} \left[ p\left(y|x, \boldsymbol{\theta}^{(i)}\right) \right]$$

#### Model parameters



#### Class Probabilities





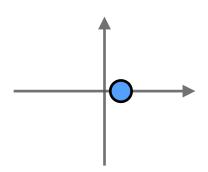
Draw from prior

$$\boldsymbol{\theta}^{(i)} \sim p(\boldsymbol{\theta}) = \mathcal{N}(0, I)$$

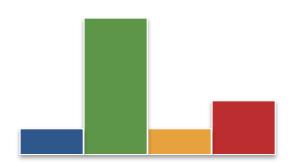
Induced predictive distribution

$$\mathbb{E}_{x \sim p(x)} \left[ p\left(y|x, \boldsymbol{\theta}^{(i)}\right) \right]$$

Model parameters

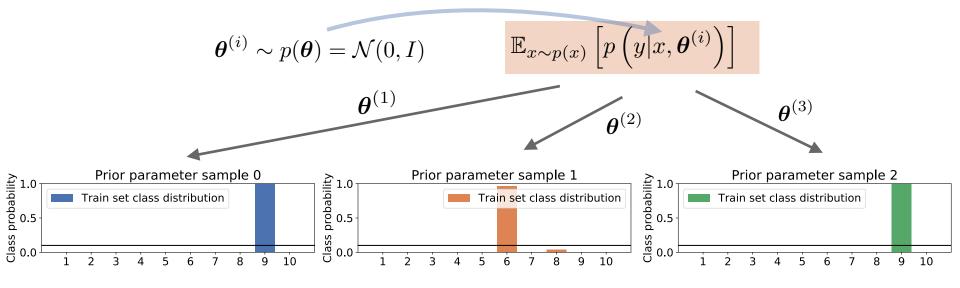


Class Probabilities

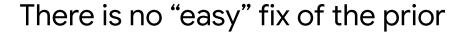


#### Prior Predictive Experiment: ResNet-20 / CIFAR-10



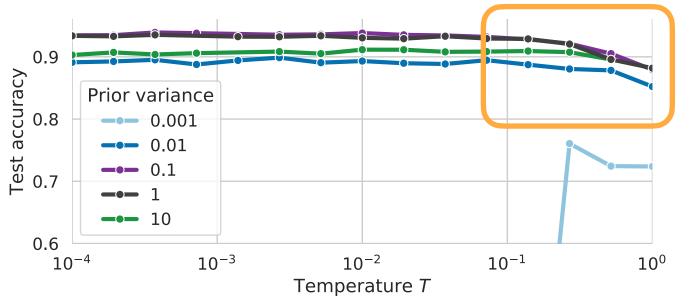


Each network drawn from the prior maps all images to one class!





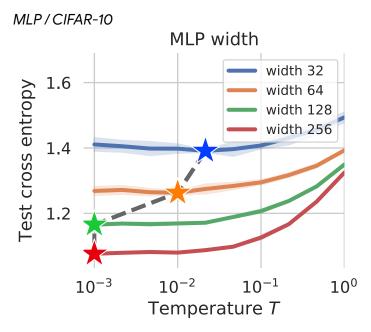
Final performance of different variances  $\sigma$  used for the prior  $\theta \sim \mathcal{N}(0,\sigma)$ 



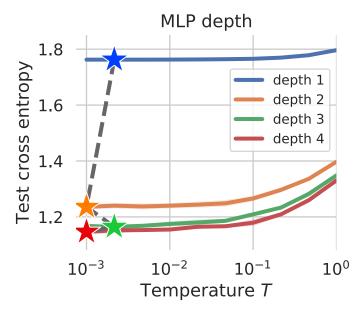
ResNet-20 / CIFAR-10



# The cold posterior effect becomes stronger with increasing capacity







(fixed width=128)

### Summary



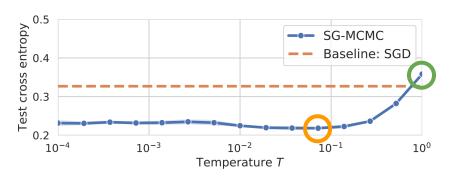
37

SG-MCMC is accurate enough.

Cold posteriors work.

More work on priors for deep nets is needed.

#### ResNet-20 / CIFAR-10



- True Bayes posterior
- Optimal cold posterior



Code: github.com/google-research/
google-research/tree/master/
cold\_posterior\_bnn

More info/feedback:
<a href="mailto:www.florianwenzel.com">www.florianwenzel.com</a>
<a href="mailto:florianwenzel.com">florianwenzel@google.com</a>

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

Florian Wenzel<sup>\*1</sup> Kevin Roth<sup>\*+2</sup> Bastiaan S. Veeling<sup>\*+31</sup> Jakub Świątkowski <sup>4+</sup> Linh Tran<sup>5+</sup> Stephan Mandt<sup>6+</sup> Jasper Snoek<sup>1</sup> Tim Salimans<sup>1</sup> Rodolphe Jenatton<sup>1</sup> Sebastian Nowozin<sup>1</sup>

#### 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 vields 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.



In supervised deen learning we use a training dataset

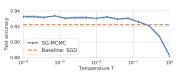


Figure 1. The "**cold posterior**" effect: for a ResNet-20 on CIFAR-10 we can improve the generalization performance significantly by cooling the posterior with a temperature  $T \ll 1$ , deviating from the Bayes posterior  $p(\theta|\mathcal{D}) \propto \exp(-U(\theta)/T)$  at T=1.

to minimize the regularized cross-entropy objective,

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

where  $\Omega(\theta)$  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,  $p(\theta|\mathcal{D}) \propto \exp(-U(\theta)/T)$ , where  $U(\theta)$  is the *posterior energy function*.

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

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