3rd Symposium on Advances in Approximate Bayesian Inference, 2020 1–15

Why Cold Posteriors? On the Suboptimal Generalization of Optimal Bayes Estimates

Chen Zeno CHENZENO@CAMPUS.TECHNION.AC.IL
Department of Electrical Engineering, Technion, Israel Institute of Technology, Haifa, Israel

Itay Golan ITAYGOLAN@GMAIL.COM
Department of Electrical Engineering, Technion, Israel Institute of Technology, Haifa, Israel

Ari Pakman ARIPAKMAN@GMAIL.COM
Department of Statistics and Center for Theoretical Neuroscience, Columbia University, New York, USA

Daniel Soudry DANIEL.SOUDRY@GMAIL.COM
Department of Electrical Engineering, Technion, Israel Institute of Technology, Haifa, Israel

Abstract

Recent works have shown that the predictive accuracy of Bayesian deep learning models exhibit substantial improvements when the posterior is raised to a $1/T$ power with $T < 1$. In this work, we explore several possible reasons for this surprising behavior.

1. Introduction

Many different approaches have been suggested to integrate neural networks with the toolbox of Bayesian inference (MacKay, 1992; Hinton and Camp, 1993; Neal, 1994; Graves, 2011; Welling and Teh, 2011; Ritter et al., 2018). In such Bayesian neural networks, we have, after training, not a single set of parameters $\mathbf{w}$ (weights), but an (approximate) posterior distribution over $\mathbf{w}$. Such a posterior distribution is very useful for many potential applications. For example, it enables uncertainty estimates over the network output; selection of hyper-parameters and models; and guided data collection (active learning).

More recently, there is a growing interest in understanding the properties of *tempered* posteriors, given by

$$p_T\left(\mathbf{w}|\mathbf{X},\mathbf{Y}\right) \propto \begin{cases} \left(p\left(\mathbf{Y}|\mathbf{X},\mathbf{w}\right)\right)^{1/T} p\left(\mathbf{w}\right) & \text{Partial tempering} \\ \left(p\left(\mathbf{Y}|\mathbf{X},\mathbf{w}\right) p\left(\mathbf{w}\right)\right)^{1/T} & \text{Full tempering} \end{cases} \tag{1}$$

where $(\mathbf{X},\mathbf{Y})$ is the observed dataset (input, output), $p\left(\mathbf{Y}|\mathbf{X},\mathbf{w}\right)$ is the likelihood, $p\left(\mathbf{w}\right)$ is the prior, and T is a 'temperature' parameter. This is motivated by the *cold posterior* effect, studied recently by Wenzel et al. (2020) and observed empirically in numerous previous works on Bayesian deep learning (e.g. Li et al. (2016); Zhang et al. (2018, 2020); Ashukha et al. (2020)). The observed effect is that in the posterior predictive distribution on input $\mathbf{x}$

$$p\left(y|\mathbf{x},\mathbf{X},\mathbf{Y}\right) = \int p\left(y|\mathbf{x},\mathbf{w}\right) p_T\left(\mathbf{w}|\mathbf{X},\mathbf{Y}\right) \mathrm{d}\mathbf{w}\,, \tag{2}$$

the use of $T < 1$ (with full tempering) commonly outperforms the standard, optimal Bayesian estimator with $T = 1$ in terms of predictive test accuracy. Note that the $1/T$ power in (1) with $T < 1$ artificially sharpens the posterior around models with high posterior probability by overcounting the data by a factor of $1/T$ (apart from the prior sharpening in the full tempering case). This result is surprising because, although many works argued that tempered posteriors improve posterior inference for misspecified models (arguably the case for neural network models), all these works required $T > 1$ (Jansen, 2013; Grünwald et al., 2017; Miller and Dunson, 2018).[1]

In this work, we examine several possible reasons for the cold posterior effect. After reviewing related works in Section 2, we consider first the possibility of a mismatch between the true prior and the one we use in practice. We explore two types of prior mismatch. First, in Section 3 we argue that good priors should be input-dependent, and show how this can lead to the observed cold posterior. In Section 4 we examine a prior mismatch due to depth, when the assumed model (student) is deeper than the actual model generating the labels (teacher). Finally, we demonstrate in Section 5 that the cold posterior effect is feasible, even without model mismatch in the case of a single teacher (the standard supervised learning setting) in some relevant cases, e.g. with heavy-tailed posterior distributions. Existing empirical evidence suggests the hypothesis of input-dependent priors to be most likely.

2. Related works

Recently Wenzel et al. (2020) conducted an extensive empirical and theoretical study of the cold posterior effect, ruling out several candidate reasons, such as an inaccurate inference method, or the non-formal likelihood functions used in deep learning models (e.g. data augmentation, batch normalization).

The work by Adlam et al. (2020) studied the cold posterior effect in Gaussian processes (GP) classification and regression, and argued that the high quality of the labels in academic benchmarks is not reflected in the high observation noise (called aleatoric uncertainty) assumed by the GP model. The effect of the $1/T$ power would be to reduce the observation noise of the model, thus adequating the latter to the data. A problem in this explanation for the regression case is that it requires an $1/T$ factor not only to affect the observation noise, but also the GP kernel — and it remains unclear why this should be the case. We provide an explanation of this simultaneous rescaling of noise and GP kernel in Section 4 and in Appendices B.3 and B.4, via a depth mismatch between the data-generating network and the one assumed for the model.

Anonymous (2021) formalizes the argument of Adlam et al. (2020) for classification by showing that the data overcounting implied by the power of $1/T$ in the likelihood in (1) is consistent with a training dataset that only includes examples for which $1/T$ human data labelers have agreed on the same label. However, to validate this proposal it remains to be seen if there is a cold-posterior effect when only a single labeler is used to create the dataset. In Section 3 we provide an alternative explanation to the issue of likelihood normalization, that still allows a single labeler to generate datasets with seemingly cold posteriors.

1. Note that these works considered the partial tempering case, but we do not expect much difference between the two cases in (1), as explained in Wenzel et al. (2020).

Additionally, note that all the above works explain the cold posterior effect as the result of model misspecification. However, in Section 5 we show that the effect is also possible without model misspecification.

3. Prior mismatch due to input-dependent prior leads to cold posterior

The notion of input-dependent priors is natural if we assume that the input data $\mathbf{X}$ (the regressors) influence our prior knowledge of the model before any observations $\mathbf{Y}$ are made. In such a case, the natural form of Bayes rule is

$$p\left(\mathbf{w}|y,\mathbf{x}\right) \propto p\left(y|\mathbf{w},\mathbf{x}\right) p\left(\mathbf{w}|\mathbf{x}\right). \tag{3}$$

While in Bayesian deep learning it is common to assume that weights and inputs are independent i.e. $p(\mathbf{w}|\mathbf{x}) = p(\mathbf{w})$, we argue that there is much to gain by allowing a data-dependent prior. For example, in some cases, it was shown that learning the dependence of $\mathbf{w}$ on $\mathbf{x}$ with non-predictive losses (using e.g. information theory objectives) reduces (Grandvalet and Bengio, 2005) or obviates (Ji et al., 2019) the need to learn from the targets $\mathbf{y}$.

In this section, we show that if the likelihood function is assumed of the form

$$p_T\left(y|\mathbf{x},\mathbf{w}\right) = \frac{p\left(y|\mathbf{x},\mathbf{w}\right)^{1/T}}{\sum_{y'} p\left(y'|\mathbf{x},\mathbf{w}\right)^{1/T}}, \tag{4}$$

then a particular $\mathbf{X}$-dependent prior leads naturally to a tempered posterior. This form of likelihood function was suggested in Wilson and Izmailov (2020); Wenzel et al. (2020). However, with the standard Gaussian prior on the weights, this likelihood does not lead to good performance.

We observe training data $\mathbf{Y} = \left[\mathbf{y}_1, \cdots, \mathbf{y}_N\right], \mathbf{X} = \left[\mathbf{x}_1, \cdots, \mathbf{x}_N\right]$, where $\mathbf{y}_i$ are one-hot vectors representing categorical data. We denote by y_i the index of the single element of $\mathbf{y}_i$ which is equal to one. Let $\left\{u_i\left(\mathbf{x},\mathbf{w}\right)\right\}_{i=1}^K$ be the logits of the output of a neural network with parameters vector $\mathbf{w}$ and input data $\mathbf{x}$. Then the likelihood (4) for K-class classification with class label $\mathbf{y}$ is

$$p_T\left(\mathbf{y}|\mathbf{x},\mathbf{w}\right) = \frac{\left[\frac{\exp\left(u_y(\mathbf{x},\mathbf{w})\right)}{\sum_{j=1}^K \exp\left(u_j(\mathbf{x},\mathbf{w})\right)}\right]^{1/T}}{\sum_{i=1}^K \left[\frac{\exp\left(u_i(\mathbf{x},\mathbf{w})\right)}{\sum_{j=1}^K \exp\left(u_j(\mathbf{x},\mathbf{w})\right)}\right]^{1/T}} = \frac{\exp\left(\frac{u_y(\mathbf{x},\mathbf{w})}{T}\right)}{\sum_{i=1}^K \exp\left(\frac{u_i(\mathbf{x},\mathbf{w})}{T}\right)}. \tag{5}$$

Assuming the following input-dependent prior distribution

$$p\left(\mathbf{w}|\mathbf{X}\right) \propto \prod_{n=1}^N \left(\sum_{i=1}^K \left[\frac{\exp\left(u_i\left(\mathbf{x}_n,\mathbf{w}\right)\right)}{\sum_{j=1}^K \exp\left(u_j\left(\mathbf{x}_n,\mathbf{w}\right)\right)}\right]^{1/T}\right) \mathcal{N}\left(\mathbf{w}|0,T\Sigma_w)\right). \tag{6}$$

and applying Bayes' rule

$$p\left(\mathbf{w}|\mathbf{X},\mathbf{Y}\right) = \frac{p\left(\mathbf{Y}|\mathbf{X},\mathbf{w}\right) p\left(\mathbf{w}|\mathbf{X}\right)}{p\left(\mathbf{Y}|\mathbf{X}\right)}, \tag{7}$$

the resulting log posterior is

$$\log p\left(\mathbf{w}|\mathbf{X},\mathbf{Y}\right)=\frac{1}{T}\sum_{n=1}^{N}\log\left[\frac{\exp\left(u_{y_n}\left(\mathbf{x}_n,\mathbf{w}\right)\right)}{\sum_{j=1}^{K}\exp\left(u_j\left(\mathbf{x}_n,\mathbf{w}\right)\right)}\right]+\frac{1}{T}\log\left(\mathcal{N}\left(\mathbf{w}|0,\Sigma_w\right)\right)+C\,. \tag{8}$$

Now we will show that (8) is equivalent to the tempered posterior distribution in the case of classification. In the case of classification the commonly used likelihood function is

$$p\left(\mathbf{y}|\mathbf{x},\mathbf{w}\right)=\frac{\exp\left(u_y\left(\mathbf{x},\mathbf{w}\right)\right)}{\sum_{i=1}^{K}\exp\left(u_i\left(\mathbf{x},\mathbf{w}\right)\right)} \tag{9}$$

and the commonly used prior is $p\left(\mathbf{w}\right)=\mathcal{N}\left(\mathbf{w}|0,\Sigma_w\right)$. The fully tempered log posterior is

$$\log p\left(\mathbf{w}|\mathbf{X},\mathbf{Y}\right)=\frac{1}{T}\sum_{n=1}^{N}\log\left[\frac{\exp\left(u_{y_n}\left(\mathbf{x}_n,\mathbf{w}\right)\right)}{\sum_{j=1}^{K}\exp\left(u_j\left(\mathbf{x}_n,\mathbf{w}\right)\right)}\right]+\frac{1}{T}\log\left(\mathcal{N}\left(\mathbf{w}|0,\Sigma_w\right)\right)+C\,, \tag{10}$$

so (8) and (10) are equivalent. Note the particular input-dependent prior we propose in (6), which has a sum over the K indices of y_i. For $T<1$ it has a similar flavor as the information theory objectives mentioned above (Grandvalet and Bengio, 2005; Ji et al., 2019), encouraging the network to be confident in its most likely prediction when no label is observed. In Appendix A we introduce the full generative model.

4. Prior mismatch due to depth leads to cold posterior

In this section, we show that the cold posterior effect also occurs when the network used as a model ('student network') is deeper than the data-generating network ('teacher network'). To avoid inexact inference methods, we focus on models with closed-form solutions for the Bayesian optimal estimator to demonstrate this effect. We focus here first on deep and wide *linear* models, and extend the results to neural network Gaussian process (NNGP) and neural tangent kernel (NTK) with ReLU activation functions in Appendices B.3 and B.4.

We observe training data $(\mathbf{x}_i,y_i)\in\mathbb{R}^d\times\mathbb{R}$ for $i=1,\ldots,N$ from a model

$$\mathbf{x}_i\sim\mathcal{N}\left(0,\mathbf{\Sigma}\right),\quad \epsilon_i\sim\mathcal{N}\left(0,\sigma^2\right),\quad y_i=\left(\prod_{l=1}^{L}\mathbf{W}^l\right)\mathbf{x}_i+\epsilon_i\,,$$

where $\mathbf{W}^l\in\mathbb{R}^{n_l\times n_{l-1}}, n_0=d, n_L=1$, and the samples are drawn independently. We consider the case of an infinite width network. The prior distribution of W^l_{ij} is $\mathcal{N}\left(0,\frac{\sigma_w^2}{n_{l-1}}\right)$ i.i.d. In the linear case, the neural network is equivalent to a linear predictor. Thus, to calculate the Bayesian predictor we only need the posterior of $\mathbf{z}=\prod_{l=1}^{L}\mathbf{W}^l$. To calculate the Bayesian estimator of $\mathbf{z}$ we only need to find its prior distribution. According to the central limit theorem (see Appendix B.1 for additional details) the prior distribution of the equivalent linear model $\mathbf{z}$ is given by $\mathbb{N}\left(0,\frac{\sigma_w^{2L}}{d}\mathbf{I}\right)$. Thus, we can calculate the fully tempered posterior distribution of $\mathbf{z}$

$$p\left(\mathbf{z}|\mathbf{Y},\mathbf{X}\right)\propto\left[p\left(\mathbf{Y}|\mathbf{X},\mathbf{z}\right)p\left(\mathbf{z}|\mathbf{X}\right)\right]^{\frac{1}{T}}\propto\exp\left(\frac{-1}{2T\sigma^2}\sum_{n=1}^{N}\left(y_n-\mathbf{z}^\top\mathbf{x}_n\right)^2\right)\exp\left(\frac{-d}{2T^L\sigma_w^{2L}}\mathbf{z}^\top\mathbf{z}\right).$$

The posterior distribution of $\mathbf{z}$ is equivalent to the posterior of Bayesian linear regression with prior variance $\frac{T^L \sigma_w^{2L}}{d}$ and noise variance $T\sigma^2$, therefore the estimator is

$$\hat{y}(\mathbf{x}) = \mathbb{E}(y|\mathbf{x}, \mathbf{Y}, \mathbf{X}) = \frac{1}{\sigma^2}\mathbf{Y}^\top \mathbf{X} \left(\frac{d}{T^{L-1}\sigma_w^{2L}} \mathbf{I} + \frac{1}{\sigma^2}\mathbf{X}^\top \mathbf{X} \right)^{-1} \mathbf{x}. \tag{11}$$

As can be seen from (11), changing the temperature T is equivalent to changing the variance of the prior distribution. We demonstrate the effect of the prior mismatch on the optimal temperature using a wide linear neural network with L_t layers as a teacher network, and a wide linear neural network with L_s layers as a student network. Therefore, we get prior mismatch if $L_t \neq L_s$. In this case, the tempered posterior can compensate for the prior mismatch, and the optimal temperature is (see Appendix B.2 for the full derivation)

$$T_{\text{opt}} = \sigma_w^{2\left(\frac{L_t - L_s}{L_s - 1}\right)}.$$

The average MSE is presented in figure 1 (see Appendix B.5 for implementation details). The results demonstrate that for $\sigma_w > 1$ (as used in Wenzel et al. (2020)) we get the cold posterior effect when using a student network deeper than the teacher network. In Appendix B.3 and B.4 we show that this phenomenon also occurs in non-linear models (neural network Gaussian process (NNGP) and neural tangent kernel (NTK) with ReLU activation functions).

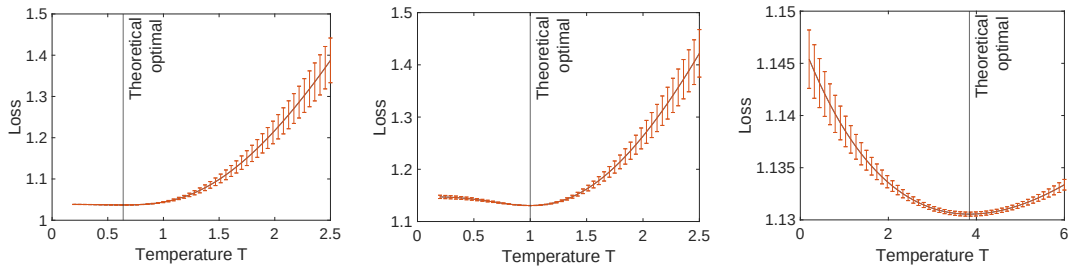

Figure 1: Average MSE for 10000 Monte Carlo samples of training sets. **(a)** prior model - 4 layers, true prior model - 2 layers. **(b)** prior model - 4 layers, true prior model - 4 layers. **(c)** prior model - 2 layers, true prior model - 4 layers.

The hypothesis in this section that the cold posterior effect originates from prior mismatch due to depth (the student network is deeper than the teacher network) implies a simple prediction: in a model which does not use bias, changing the variance of the prior is equivalent to changing the depth of the model. Assuming the hypothesis is correct, we should observe less cold posterior effect when we change the variance of the prior distribution (we do not expect much difference when we do use bias term). However, Wenzel et al. (2020) (in section 5.2) examines the effect of prior variance on the cold posterior effect. The experiment shows that the cold posterior effect is present for all tested choices of the prior variance. This seems to contradict the hypothesis's prediction. On the other hand, this type of prior mismatch is likely to occur since in Bayesian deep learning we usually use over-parameterized models along with Gaussian prior.

5. Cold posterior effect without model mismatch

In this section, we will demonstrate that when the prior and the posterior distributions are heavy-tailed, using tempered posterior with $T < 1$ is optimal with high probability, in the setting of supervised learning. In this setting, we use a given dataset (e.g. CIFAR-10, ImagNet) to estimate the label for a new unlabeled data, which is equivalent in the Bayesian setting to have a fixed $\mathbf{X}$ and only one sample of the teacher network to generate $\mathbf{y}$. To make a clear demonstration we consider an extreme case of heavy-tailed distribution where for *every* sampled dataset $T < 1$ is outperforming the optimal Bayesian estimator. Let consider the case where the prior and the posterior distributions of the parameter are equal to the distribution of an absolute value of a Cauchy random variable. In the case of linear model, the Bayesian estimator is $\hat{y} = \hat{w}x$, where

$$\hat{w}(T) \propto \int_{-\infty}^{\infty} |w| \frac{1}{(1+w^2)^{1/T}} \mathrm{d}w = \begin{cases} \infty & \text{if } T = 1 \\ \frac{T}{1-T} & \text{if } T < 1 \end{cases} . \tag{12}$$

Therefore, for a single dataset the MSE is[2]

$$\mathrm{MSE}(w^*) = (w^* - \hat{w}(T))^2 + \sigma^2 = \begin{cases} \infty & \text{if } T = 1 \\ \text{const} & \text{if } T < 1 \end{cases} . \tag{13}$$

Meaning that for *every* sampled dataset the Bayesian estimator with $T < 1$ outperforms the optimal Bayesian estimator. In Appendix C.1 we demonstrate that heavy-tailed prior and posterior distributions exist in the case of narrow and deep neural network and show that with high probability $T < 1$ is optimal. In Appendix C.3 we also demonstrate similar results for a bi-modal distribution.

However, if we sample correctly from such posterior distributions (which have rare events with high error) then we expect that only a few predictions would have a high error, while most would be correct. In contrast, when we examined the SG-MCMC samples (see Appendix C.4) obtained in Wenzel et al. (2020) we found that all of the predictions of the Bayesian ensemble had high error. This seems to contradict the hypothesis in this section, and suggest that some model misspecification is necessary to generate the cold posterior effect.

6. Conclusions

In this work, we suggested and examined several hypotheses to explain the cold posterior effect. In the case of classification we can construct a new likelihood function with tempered soft-max (Hinton et al., 2015) using temperatures $T < 1$. We suggested a specific input-dependent prior which leads naturally to a tempered posterior. Next, we considered a prior mismatch due to a deeper student network than the teacher network. We show that prior mismatch due to network depth results in the cold posterior effect, both for linear and non-linear models. Lastly, we demonstrated that in the setting of supervised learning (with only one sample of the teacher network) the cold posterior effect is feasible. We conclude, based on existing empirical observations, that the most likely explanation is the hypothesis of input-dependent prior.

2. Assuming Gaussian noise with variance of σ^2 and $x \sim \mathcal{N}(0, 1)$.

References

Ben Adlam, Jasper Snoek, and Samuel L. Smith. Cold posteriors and aleatoric uncertainty. In *ICML 2020 Workshop on Uncertainty and Robustness in Deep Learning*. 2020.

Anonymous. A statistical theory of cold posteriors in deep neural networks. In *Submitted to International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=Rd138pWXMvG. under review.

Arsenii Ashukha, Alexander Lyzhov, Dmitry Molchanov, and Dmitry Vetrov. Pitfalls of in-domain uncertainty estimation and ensembling in deep learning. In *ICLR*, 2020.

Alberto Bietti and Julien Mairal. On the inductive bias of neural tangent kernels. In *NeurIPS*. 2019.

Youngmin Cho and Lawrence K. Saul. Kernel methods for deep learning. In *NeurIPS*. 2009.

Alexander G. de G. Matthews, Jiri Hron, Mark Rowland, Richard E. Turner, and Zoubin Ghahramani. Gaussian process behaviour in wide deep neural networks. In *International Conference on Learning Representations*, 2018.

Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Advances in neural information processing systems*, pages 529–536, 2005.

Alex Graves. Practical variational inference for neural networks. In *Advances in Neural Information Processing Systems 24*. 2011.

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

G E Hinton and D Van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In *COLT '93*, 1993.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Arthur Jacot, Franck Gabriel, and Clement Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems 31*. 2018.

Lennard Jansen. *Robust Bayesian inference under model misspecification*. PhD thesis, Master's thesis, Leiden University, 2013.

Xu Ji, João F Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9865–9874, 2019.

Chunyuan Li, Changyou Chen, David Carlson, and Lawrence Carin. Preconditioned stochastic gradient langevin dynamics for deep neural networks. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, page 1788–1794. AAAI Press, 2016.

D J C MacKay. A practical Bayesian framework for backpropagation networks. *Neural computation*, 472(1):448–472, 1992.

Alexander G de G Matthews, Mark Rowland, Jiri Hron, Richard E Turner, and Zoubin Ghahramani. Gaussian process behaviour in wide deep neural networks. *arXiv preprint arXiv:1804.11271*, 2018.

Henry H Mattingly, Mark K Transtrum, Michael C Abbott, and Benjamin B Machta. Maximizing the information learned from finite data selects a simple model. *Proceedings of the National Academy of Sciences*, 115(8):1760–1765, 2018.

Jeffrey W Miller and David B Dunson. Robust bayesian inference via coarsening. *Journal of the American Statistical Association*, 2018.

R M Neal. *Bayesian Learning for Neural Networks*. PhD thesis, Dept. of Computer Science, University of Toronto, 1994.

Hippolyt Ritter, Aleksandar Botev, and David Barber. A scalable laplace approximation for neural networks. In *6th International Conference on Learning Representations, ICLR 2018-Conference Track Proceedings*, volume 6. International Conference on Representation Learning, 2018.

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

Florian Wenzel, Kevin Roth, Bastiaan S Veeling, Jakub Świątkowski, Linh Tran, Stephan Mandt, Jasper Snoek, Tim Salimans, Rodolphe Jenatton, and Sebastian Nowozin. How good is the Bayes posterior in deep neural networks really? *ICML*, 2020.

Andrew Gordon Wilson and Pavel Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. *arXiv preprint arXiv:2002.08791*, 2020.

Guodong Zhang, Shengyang Sun, David Duvenaud, and Roger Grosse. Noisy natural gradient as variational inference. *ICML*, 2018.

Ruqi Zhang, Chunyuan Li, Jianyi Zhang, Changyou Chen, and Andrew Gordon Wilson. Cyclical Stochastic Gradient MCMC for Bayesian Deep Learning. *International Conference on Learning Representations*, 2020.

Appendix A. Derivations for section 3

In this section, we present a generative model which implies an input-dependent prior. Defining $r(\mathbf{x})$ as some distribution (of a single sample $\mathbf{x}$ in the input space), and

$$g_i(\mathbf{w}, \mathbf{x}) = r(\mathbf{x}) \left[\frac{\exp(u_i(\mathbf{x}, \mathbf{w}))}{\sum_{j=1}^{K} \exp(u_j(\mathbf{x}, \mathbf{w}))} \right]^{1/T}$$

$$g_i(\mathbf{w}) = \int g_i(\mathbf{w}, \mathbf{x})\, d\mathbf{x}\,,$$

we find the following generative model implies the input dependent prior:

$$p(\mathbf{w}) = \frac{(\sum_{i=1}^{K} g_i(\mathbf{w}))^N \mathcal{N}(\mathbf{w}|0, T\Sigma_w)}{\int (\sum_{i=1}^{K} g_i(\mathbf{w}))^N \mathcal{N}(\mathbf{w}|0, T\Sigma_w)\, \mathrm{d}\mathbf{w}} \tag{14}$$

$$p(\mathbf{y}|\mathbf{w}) = \frac{g_y(\mathbf{w})}{\sum_{j=1}^{K} g_j(\mathbf{w})} \tag{15}$$

$$p(\mathbf{x}|\mathbf{y}, \mathbf{w}) = \frac{g_y(\mathbf{w}, \mathbf{x})}{g_y(\mathbf{w})}\,. \tag{16}$$

Note that though the marginal prior $p(\mathbf{w})$ is dependent on the number of training data N, such dependence was considered in previous works (e.g. Mattingly et al. (2018)).

Appendix B. Derivations and implementation details for section 4

B.1. Calculation of the prior distribution of z - wide linear network

According to an extension of the Central Limit Theorem (CLT), derived in Theorem 4 in Matthews et al. (2018), the pre-activation of each layer (for any linear or sub-linear activation function) converge in distribution to a multivariate Gaussian as the layer widths $n_1, ..., n_{L-1}$ are taken to infinity in any order (with n_0, n_L finite). For linear activations, this also implies also that $\mathbf{z}$ is Gaussian. The first and second moments of $\mathbf{z}$ are:

$$\mathbb{E}(z_i) = 0 \tag{17}$$

$$\begin{aligned} \mathbb{E}(z_i z_j) = \mathbb{E}\Bigg(& \sum_{a_{L-1}=1}^{N_{L-1}} \cdots \sum_{a_1=1}^{n_1} W^L_{1,a_{L-1}} W^{L-1}_{a_{L-1},a_{L-2}} \cdots W^1_{a_1,i} \\ & \sum_{b_{L-1}=1}^{n_{L-1}} \cdots \sum_{b_1=1}^{n_1} W^L_{1,b_{L-1}} W^{L-1}_{b_{L-1},b_{L-2}} \cdots W^1_{b_1,j} \Bigg) \end{aligned} \tag{18}$$

$$= \sum_{a_{L-1}=1}^{n_{L-1}} \cdots \sum_{a_1=1}^{n_1} \sum_{b_{L-1}=1}^{n_{L-1}} \cdots \sum_{b_1=1}^{n_1} \delta_{a_{L-1},b_{L-1}} \frac{\sigma_w^2}{n_{L-1}} \delta_{a_{L-2},b_{L-2}} \frac{\sigma_w^2}{n_{L-2}} \cdots \delta_{i,j} \frac{\sigma_w^2}{n_0} \tag{19}$$

$$= \delta_{i,j} \frac{\sigma_w^{2L}}{n_0} \tag{20}$$

Therefore, according to the CLT theorem the prior distribution of $\mathbf{z}$ is

$$p\left(\mathbf{z}|\mathbf{X}\right)=\left(\frac{d}{2\pi\sigma_w^{2L}}\right)^{d/2}\exp\left(-\frac{d}{2\sigma_w^{2L}}\mathbf{z}^\top\mathbf{z}\right). \tag{21}$$

B.2. Calculation of the optimal temperature - prior mismatch in wide linear network

The MSE of the Bayesian optimal estimator is

$$\begin{aligned}\text{MSE}&=\mathbb{E}_{(\mathbf{x},y)}\left[\left(y-\hat{y}\left(\mathbf{x}\right)\right)^2|\mathbf{X},\mathbf{Y}\right]=\mathbb{E}_{\mathbf{z}}\left[\mathbb{E}_{(\mathbf{x},y)}\left[\left(y\left(\mathbf{z}\right)-\hat{y}\left(\mathbf{x}\right)\right)^2|\mathbf{z},\mathbf{X},\mathbf{Y}\right]|\mathbf{X},\mathbf{Y}\right] &(22)\\ &=\mathbb{E}_{\mathbf{z}}\left[\mathbb{E}_{(\mathbf{x},y)}\left[\left(\mathbf{z}^\top\mathbf{x}+\epsilon-\hat{\mathbf{z}}^\top\mathbf{x}\right)^2|\mathbf{z},\mathbf{X},\mathbf{Y}\right]|\mathbf{X},\mathbf{Y}\right] &(23)\\ &=\mathbb{E}_{\mathbf{z}}\left[\sigma^2+\frac{1}{N}\left\|\mathbf{z}-\hat{\mathbf{z}}\right\|^2|\mathbf{X},\mathbf{Y}\right] &(24)\end{aligned}$$

where

$$\hat{\mathbf{z}}=\frac{1}{\sigma^2}\left(\frac{d}{T^{L_s-1}\sigma_w^{2L_s}}\mathbf{I}+\frac{1}{\sigma^2}\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top\mathbf{Y}. \tag{25}$$

The posterior distribution of $\mathbf{z}$ is

$$p\left(\mathbf{z}|\mathbf{X},\mathbf{Y}\right)=\mathcal{N}\left(\frac{1}{\sigma^2}\left(\frac{d}{\sigma_w^{2L_t}}\mathbf{I}+\frac{1}{\sigma^2}\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top\mathbf{Y},\left(\frac{d}{\sigma_w^{2L_t}}\mathbf{I}+\frac{1}{\sigma^2}\mathbf{X}^\top\mathbf{X}\right)^{-1}\right) \tag{26}$$

therefore,

$$\begin{aligned}\text{MSE}=\sigma^2&+\frac{1}{N}\text{Tr}\left(\left(\frac{d}{\sigma_w^{2L_t}}\mathbf{I}+\frac{1}{\sigma^2}\mathbf{X}^\top\mathbf{X}\right)^{-1}\right)\\&+\frac{1}{N}\left\|\frac{1}{\sigma^2}\left(\frac{d}{T^{L_s-1}\sigma_w^{2L_s}}\mathbf{I}+\frac{1}{\sigma^2}\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top\mathbf{Y}-\frac{1}{\sigma^2}\left(\frac{d}{\sigma_w^{2L_t}}\mathbf{I}+\frac{1}{\sigma^2}\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top\mathbf{Y}\right\|^2.\end{aligned} \tag{27}$$

As can be seen, the optimal temperature is

$$T_{\text{opt}}=\sigma_w^{2\left(\frac{L_t-L_s}{L_s-1}\right)}. \tag{28}$$

B.3. Neural Network Gaussian Process (NNGP)

At initialization, neural networks are equivalent to NNGP infinite-width limit (de G. Matthews et al., 2018). Due to the non-linearity of NNGP we can no longer obtain the optimal temperature analytically. However, we numerically demonstrate that prior mismatch due to deeper prior than the true prior leads to cold posterior effect.

We observe training data $(\mathbf{x}_i,y_i)\in\mathbb{R}^d\times\mathbb{R}$ for $i=1,\ldots,N$ from a model

$$\mathbf{x}_i\sim\mathcal{N}\left(0,\mathbf{\Sigma}\right),\quad\epsilon_i\sim\mathcal{N}\left(0,\sigma^2\right),\quad y_i=f\left(\mathbf{x}_i\right)+\epsilon_i, \tag{29}$$

where (Cho and Saul, 2009; de G. Matthews et al., 2018)

$$f(\mathbf{x}) \sim \mathcal{GP}\left(0, K^{L-1}(\mathbf{x},\mathbf{x})\right) \tag{30}$$

$$K^{l}(\mathbf{x},\mathbf{x}') = \sigma_b^2 + \frac{\sigma_w^2}{2\pi}\sqrt{K^{l-1}(\mathbf{x},\mathbf{x})\,K^{l-1}(\mathbf{x}',\mathbf{x}')}\left(\sin\theta_{x,x'}^{l-1} + \left(\pi - \theta_{x,x'}^{l-1}\right)\cos\theta_{x,x'}^{l-1}\right) \tag{31}$$

$$\theta_{x,x'}^{l} = \cos^{-1}\left(\frac{K^{l}(\mathbf{x},\mathbf{x}')}{\sqrt{K^{l}(\mathbf{x},\mathbf{x})\,K^{l}(\mathbf{x}',\mathbf{x}')}}\right) \tag{32}$$

$$K^{0}(\mathbf{x},\mathbf{x}') = \sigma_b^2 + \sigma_w^2\left(\frac{\mathbf{x}\cdot\mathbf{x}'}{d}\right) \tag{33}$$

and the samples are drawn independently. We define the feature matrix $\mathbf{X}$ and the label vector $\mathbf{Y}$ as follows

$$\mathbf{X} = \begin{bmatrix}\mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_N\end{bmatrix}^{\top} \in \mathbb{R}^{N\times d}, \quad \mathbf{Y} = \begin{bmatrix}y_1 & y_2 & \cdots & y_N\end{bmatrix}^{\top} \in \mathbb{R}^{N}. \tag{34}$$

we can write the joint distribution

$$\begin{bmatrix}\mathbf{Y}\\ y\end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix}K^{L-1}(\mathbf{X},\mathbf{X}) + \sigma^2\mathbf{I} & K^{L-1}(\mathbf{X},\mathbf{x})\\ K^{L-1}(\mathbf{x},\mathbf{X}) & K^{L-1}(\mathbf{x},\mathbf{x}) + \sigma^2\end{bmatrix}\right). \tag{35}$$

So the predictive distribution is given by

$$p(y|\mathbf{x},\mathbf{X},\mathbf{Y}) = \mathcal{N}\left(y|\mu_y,\sigma_y^2\right), \tag{36}$$

where

$$\mu_y = K^{L-1}(\mathbf{x},\mathbf{X})\left(K^{L-1}(\mathbf{X},\mathbf{X}) + \sigma^2\mathbf{I}\right)^{-1}\mathbf{Y} \tag{37}$$

$$\sigma_y^2 = K^{L-1}(\mathbf{x},\mathbf{x}) + \sigma^2 - K^{L-1}(\mathbf{x},\mathbf{X})\left(K^{L-1}(\mathbf{X},\mathbf{X}) + \sigma^2\mathbf{I}\right)^{-1}K^{L-1}(\mathbf{X},\mathbf{x}) \tag{38}$$

The equivalent of cold posterior in GP is obtain using tempered kernel and tempered noise in the training set (without tempered noise in the test data point) and we obtain

$$\hat{y} = K_T^{L-1}(\mathbf{x},\mathbf{X})\left(K_T^{L-1}(\mathbf{X},\mathbf{X}) + T\sigma^2\mathbf{I}\right)^{-1}\mathbf{Y} \tag{39}$$

Similar to the previous subsection, we demonstrate the effect of the prior mismatch on the optimal temperature using a Gaussian process with neural network kernel with L_t layers as a teacher network, and a Gaussian process with neural network kernel with L_s layers as a student network. The MSE of the Bayesian optimal estimator is

$$\text{MSE} = \mathbb{E}_{(\mathbf{x},y)}\left[\left(y - \hat{y}(\mathbf{x})\right)^2|\mathbf{X},\mathbf{Y}\right] = \mathbb{E}_{\mathbf{x}}\left[\mathbb{E}_y\left[\left(y - \hat{y}(\mathbf{x})\right)^2|\mathbf{x},\mathbf{X},\mathbf{Y}\right]\right] \tag{40}$$

$$= \mathbb{E}_{\mathbf{x}}\left[\sigma_y^2(\mathbf{x}) + \left(\mu_y(\mathbf{x}) - \hat{y}(\mathbf{x})\right)^2\right]. \tag{41}$$

The average MSE is presented in figure 2 (See Appendix B.5 for implementation details). Similar to the case of a wide linear neural network, the results demonstrate that when using a student network which is deeper than the teacher network we get the cold posterior effect.

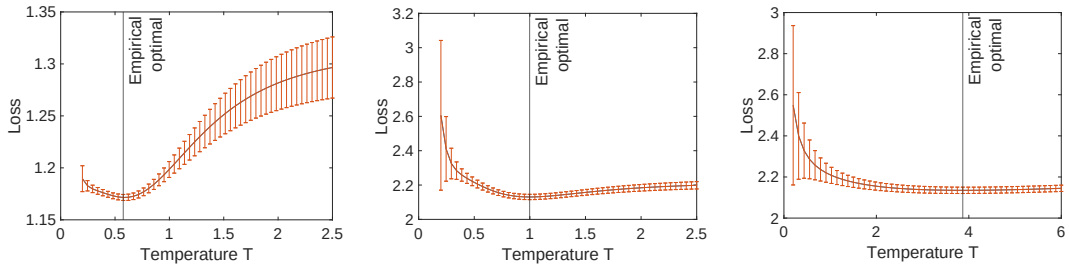

Figure 2: The average MSE for 100 Monte Carlo samples of training sets. **(a)** prior model - 3 hidden layers, true prior model - 1 hidden layers. **(b)** prior model - 3 hidden layers, true prior model - 3 hidden layers. **(c)** prior model - 1 hidden layers, true prior model - 3 hidden layers.

B.4. Neural Tangent Kernel (NTK)

Neural networks trained using continuous gradient descent on MSE loss are equivalent to NTK in the infinite-width limit (Jacot et al., 2018). Similar to the case of NNGP we numerically demonstrate that prior mismatch due to deeper prior than the true prior leads to cold posterior effect. In the case of NTK, the derivation of the optimal Bayesian estimator and the MSE is similar to the case of NNGP except for the kernel function. In this case, the kernel function is given by[3] (Cho and Saul, 2009; Jacot et al., 2018; Bietti and Mairal, 2019)

$$f\left(\mathbf{x}\right) \sim \mathcal{GP}\left(0, K^{L-1}\left(\mathbf{x},\mathbf{x}\right)\right) \tag{42}$$

$$K^{l}\left(\mathbf{x},\mathbf{x}'\right) = \Sigma^{l}\left(\mathbf{x},\mathbf{x}'\right) + K^{l-1}\left(\mathbf{x},\mathbf{x}'\right)\frac{\sigma_w^2}{2\pi}\left(\pi - \theta^{l-1}_{x,x'}\right) \tag{43}$$

$$\Sigma^{l}\left(\mathbf{x},\mathbf{x}'\right) = \frac{\sigma_w^2}{2\pi}\sqrt{\Sigma^{l-1}\left(\mathbf{x},\mathbf{x}\right)\Sigma^{l-1}\left(\mathbf{x}',\mathbf{x}'\right)}\left(\sin\theta^{l-1}_{x,x'} + \left(\pi - \theta^{l-1}_{x,x'}\right)\cos\theta^{l-1}_{x,x'}\right) \tag{44}$$

$$\theta^{l}_{x,x'} = \cos^{-1}\left(\frac{\Sigma^{l}\left(\mathbf{x},\mathbf{x}'\right)}{\sqrt{\Sigma^{l}\left(\mathbf{x},\mathbf{x}\right)\Sigma^{l}\left(\mathbf{x}',\mathbf{x}'\right)}}\right) \tag{45}$$

$$K^{0}\left(\mathbf{x},\mathbf{x}'\right) = \Sigma^{0}\left(\mathbf{x},\mathbf{x}'\right) = \sigma_w^2\left(\frac{\mathbf{x}\cdot\mathbf{x}'}{d}\right). \tag{46}$$

The average MSE is presented in figure 3 (see Appendix B.5 for implementation details). Similar to the case of NNGP, the results demonstrate that when using a student network which is deeper than the teacher network we get the cold posterior effect.

B.5. Implementation details

B.5.1. Deep and wide linear network

We use a wide linear network with $L_{teacher}$ as a teacher network and a a wide linear network with $L_{student}$ as a student network. For both we use prior of $\mathcal{N}\left(0,\sigma_w^2\right)$ with $\sigma_w = 1.4$.

3. Without the bias.

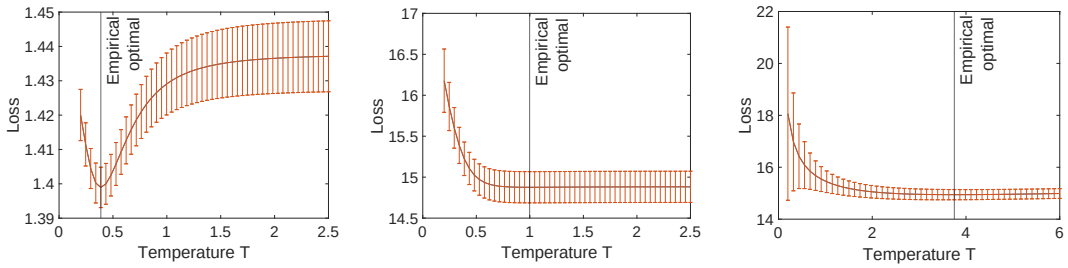

Figure 3: The average MSE for 100 Monte Carlo samples of training sets. **(a)** prior model - 3 hidden layers, true prior model - 1 hidden layers. **(b)** prior model - 3 hidden layers, true prior model - 3 hidden layers. **(c)** prior model - 1 hidden layers, true prior model - 3 hidden layers.

We use as an train set $N = 100$ samples of $(\mathbf{x}, y)$ where $x \in \mathbb{R}^d$ and $d = 100$, in addition $\mathbf{x} \sim \mathcal{N}\left(0, \frac{1}{N}\mathbf{I}\right)$. The additive noise sampled from $\mathcal{N}\left(0, \sigma^2\right)$ where $\sigma = 1$.

B.5.2. GAUSSIAN PROCESSES

For both NNGP and NTK we use a Gaussian process with kernel of $L_{teacher}$ layers as a teacher and a Gaussian process with neural network kernel of $L_{student}$ layers as a student. For both teacher and student we use prior of $\mathcal{N}\left(0, \sigma_w^2\right)$ with $\sigma_w = 2.6674$ and $\sigma_b = 0$. We use as an train set $N = 100$ samples of $(\mathbf{x}, y)$ where $x \in \mathbb{R}^d$ and $d = 100$, in addition $\mathbf{x} \sim \mathcal{N}\left(0, \frac{1}{N}\mathbf{I}\right)$. The additive noise sampled from $\mathcal{N}\left(0, \sigma^2\right)$ where $\sigma = 1$.

Appendix C. Derivations for section 5

C.1. Scalar Neural Network

Let consider the case of scalar neural network model with an absolute value activation function.

$$y = \left(\prod_{j=1}^{L} |w_j|\right) |x| + \epsilon, \quad \epsilon \sim \mathcal{N}\left(0, \sigma^2\right) , \tag{47}$$

where $\mathbf{w} \in \mathbb{R}^L$ and $L \gg 1$. The prior distribution of $\mathbf{w}$ is

$$p\left(\mathbf{w}\right) = \mathcal{N}\left(\mathbf{w}|0, \sigma_w^2 \mathbf{I}_d\right) . \tag{48}$$

To calculate the Bayesian predictor we only need the posterior of $z = \prod_{j=1}^{L} |w_j|$. According to the central limit theorem (CLT), the distribution of z is approximately (see Appendix C.2 for additional details)

$$z \sim \text{Lognormal}\left(\mu_z, \sigma_z^2\right) \tag{49}$$

where

$$\mu_z = L\mathbb{E}\left(\log\left(|w_i|\right)\right), \quad \sigma_z^2 = L\mathrm{Var}\left(\log\left(|w_i|\right)\right). \tag{50}$$

For simplicity, we assume that no training data is observed. In the case of tempered posterior the the Bayesian optimal estimator is

$$\hat{z} = T^{\frac{L}{2}} \exp\left(\mu_z + \frac{\sigma_z^2}{2}\right). \tag{51}$$

The sampled parameter is most likely to be around the mode of the distribution

$$z^* \approx \exp\left(\mu_z - \sigma_z^2\right). \tag{52}$$

Thus, with high probability for a single dataset the optimal temperature is

$$T_{\text{opt}} \approx \exp\left(-\frac{\sigma_z^2}{L}\right) = \exp\left(-\mathrm{Var}\left(\log\left(|w_i|\right)\right)\right) < 1. \tag{53}$$

C.2. Calculation of the prior distribution of z - scalar network

For large enough L the distribution of the sample average of $\tilde{w}_j = \log\left(|w_j|\right)$ is approximately

$$\frac{1}{L}\sum_{j=1}^{L} \log\left(|w_j|\right) \sim \mathcal{N}\left(\mu_{\tilde{w}}, \frac{\sigma_{\tilde{w}}^2}{L}\right), \tag{54}$$

Therefore the distribution of z is approximately

$$z \sim \text{Lognormal}\left(\mu_z, \sigma_z^2\right) \tag{55}$$

where

$$\mu_z = L\mathbb{E}\left(\log\left(|w_i|\right)\right), \quad \sigma_z^2 = L\mathrm{Var}\left(\log\left(|w_i|\right)\right). \tag{56}$$

C.3. Bi-modal distribution

Another example of distribution where with high probability $T < 1$ is outperforming the optimal Bayesian estimator is the Bernoulli distribution with $p \approx 1$. Let consider the following model

$$x \sim \mathcal{N}\left(0, 1\right), \quad \epsilon \sim \mathcal{N}\left(0, \sigma^2\right), \quad y = f\left(x, w\right) + \epsilon. \tag{57}$$

In addition, we assume that the prior distribution and the posterior distribution of the parameter are Bernoulli distribution. Therefore, the tempered posterior distribution is

$$w \sim \text{Bern}\left(q_T\right), \tag{58}$$

where

$$q_T = \frac{1}{1 + \left(p^{-1} - 1\right)^{1/T}}. \tag{59}$$

In this case the Bayesian optimal estimator is

$$\hat{y} = q_T f(x,1) + (1-q_T) f(x,0) \ . \tag{60}$$

Therefore, for a single data set we obtain

$$\mathrm{MSE}(w^*) = \mathbb{E}_x \left(f(x,w^*) - \hat{y}\right)^2 + \sigma^2 \tag{61}$$

$$= \sigma^2 + \mathbb{E}_x \left(f(x,0) - f(x,1)\right)^2 \begin{cases} (1-q_T)^2 & \text{if } w^* = 1 \\ q_T^2 & \text{if } w^* = 0 \end{cases} . \tag{62}$$

For example, if we have a prior distribution with $p(w=1) = 0.99$ then the sampled parameter is most likely to be $w^* = 1$. Therefore, with high probability the optimal temperature is $T = 0$ since then $q_T = 1$ and $\mathrm{MSE}(w^*) = \sigma^2$.

C.4. SG-MCMC samples for $T = 1$

In this subsection, we present the SG-MCMC samples obtained in the experiment of ResNet-20 on CIFAR-10 (Wenzel et al., 2020). As can be seen from figure 4, all of the predictions of the optimal Bayesian ensemble ($T = 1$) have high error compared to the Bayesian ensemble with $T = 0.01$ and the SGD baseline.

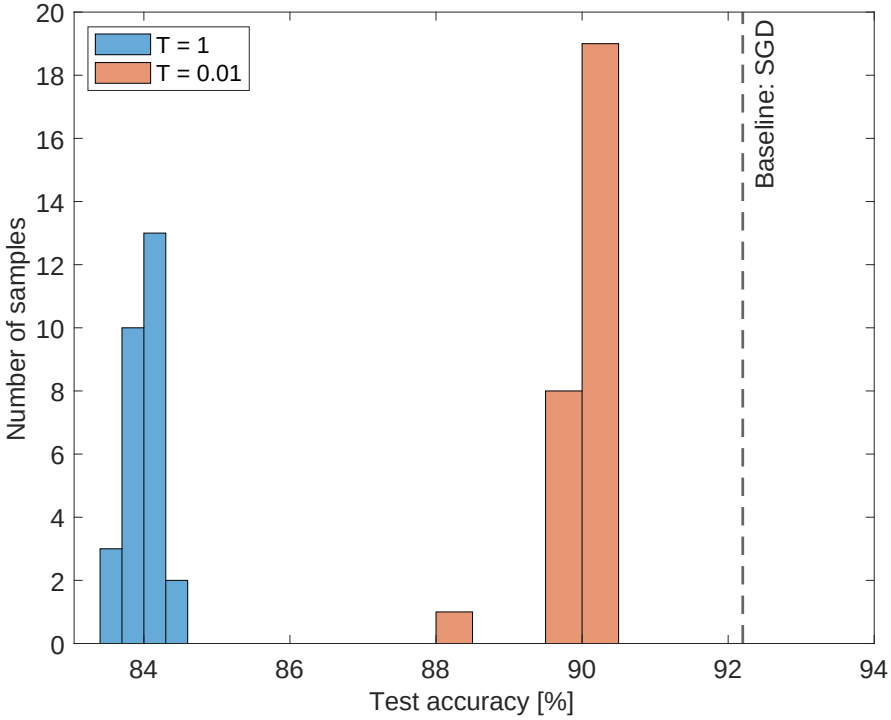

Figure 4: The histogram of the test accuracy of the SG-MCMC samples for $T = 1$ and $T = 0.01$. The number of samples is 28.