Bayesian Neural Network Priors Revisited

Vincent Fortuin∗
ETH Zürich
fortuin@inf.ethz.ch

Adrià Garriga-Alonso∗
University of Cambridge
ag919@cam.ac.uk

Florian Wenzel
Humboldt University of Berlin

Gunnar Rätsch
ETH Zürich

Richard E. Turner
University of Cambridge

Mark van der Wilk†
Imperial College London

Laurence Aitchison†
University of Bristol

Abstract

Isotropic Gaussian priors are the de facto standard for modern Bayesian neural network inference. However, such simplistic priors are unlikely to either accurately reflect our true beliefs about the weight distributions, or to give optimal performance. We study summary statistics of (convolutional) neural network weights in networks trained using SGD. We find that in certain circumstances, these networks have heavy-tailed weight distributions, while convolutional neural network weights often display strong spatial correlations. Building these observations into the respective priors, we get improved performance on MNIST classification. Remarkably, we find that using a more accurate prior partially mitigates the cold posterior effect, by improving performance at high temperatures corresponding to exact Bayesian inference, while having less of an effect at small temperatures.

1 Introduction

In a Bayesian neural network (BNN), we put a prior \( p(w) \) over the neural network parameters, and compute the posterior distribution over parameters conditioned on training data, \( p(w|x, y) = p(y|w, x)p(w)/p(y|x) \). This procedure should give considerable advantages for reasoning about predictive uncertainty, which is especially relevant in the small data setting. Crucially, to perform Bayesian inference, we need to choose a prior that accurately reflects our beliefs about the parameters before seeing any data [Bayes, 1763, Gelman et al., 2013]. However, the most common choice of the prior for BNN weights is the simplest one: the isotropic Gaussian. Isotropic Gaussians are used across almost all fields of Bayesian deep learning, ranging from variational inference [Dusenberry et al., 2020], to sampling-based inference using SGLD [Zhang et al., 2019], and even to infinite networks [Lee et al., 2017, Garriga-Alonso et al., 2018]. This is troubling, since isotropic Gaussian priors are almost certainly too simplistic. Indeed, artificially reducing posterior uncertainty using "cold" posteriors has been found to improve performance in Bayesian neural networks (BNNs) [Wenzel et al., 2020a]. This is surprising, because if the prior and likelihood are accurately reflecting our beliefs, the Bayesian solution really should be optimal [Gelman et al., 2013]. This raises the possibility that either the prior [Wenzel et al., 2020a] or likelihood [Aitchison, 2020], (or both) are ill-specified.

In this work, we study empirically whether isotropic Gaussian priors are indeed suboptimal for BNNs and whether this can explain the cold posterior effect. We analyse the performance of different BNN

∗Equal contribution.
†Equal contribution.
priors for different network architectures and compare them to the empirical weight distributions of maximum-likelihood solutions. We conclude that isotropic priors with heavier tails than the Gaussian are better suited for fully connected neural networks (FCNNs), while correlated Gaussian priors are better suited in the case of convolutional neural networks (CNNs). Thus, we would recommend the use of these priors instead of the widely-used isotropic Gaussians. While these priors can partially reduce the cold posterior effect in FCNNs, it remains more elusive in CNNs.

1.1 Contributions

Our main contributions are

- An analysis of the empirical weight distributions of SGD-trained neural networks with different architectures, suggesting that FCNNs learn heavy-tailed weight distributions (Sec. 3.1), while CNN weight distributions show significant correlations (Sec. 3.2).
- An empirical study of Bayesian FCCN performance, suggesting that heavy-tailed priors can outperform the widely-used Gaussian priors (Sec. 4.1).
- An empirical study of Bayesian CNN performance, suggesting that correlated Gaussian priors can outperform the isotropic ones (Sec. 4.2).
- An empirical study of the cold posterior effect in these models, suggesting that it can be reduced by choosing better priors in FCNNs, while the case is less clear in CNNs (Sec. 4).

1.2 Related Work

Previous work has investigated the performance implications of different neural network priors [Ghosh and Doshi-Velez, 2017, Wu et al., 2018, Atanov et al., 2018, Nalisnick, 2018, Overweg et al., 2019, Farquhar et al., 2019, Cui et al., 2020, Hafner et al., 2020, Ober and Aitchison, 2020]. However, none of this work uses the empirical weight distributions of SGD-trained networks to inform BNN priors. For a more in-depth discussion see Appendix A.

2 Background: the Cold Posterior Effect

When performing inference in Bayesian models, we can temper the posterior by a positive temperature $T$, giving

$$\log p(w|x, y)^1 = \frac{1}{T}[\log p(y|w, x) + \log p(w)] + Z(T)$$  \hspace{1cm} (1)$$

for a normalizing constant $Z(T)$. Setting $T = 1$ gives the standard Bayesian posterior. The temperature parameter can be easily handled when simulating Langevin dynamics, used in molecular dynamics and MCMC [Leimkuhler and Matthews, 2012].

In their recent work, Wenzel et al. [2020a] have drawn attention to the effect that when the posterior is cooled down in BNNs (i.e., setting $T \ll 1$), the performance of the models often increases. Testing different hypotheses for potential problems with the inference, likelihood and prior, they conclude that the BNN priors (which were Gaussian in their experiments) are misspecified and could be one of the main causes of the “cold posterior effect”. Reversing this argument, we can hypothesize that choosing better priors for BNNs should lead to a less pronounced cold posterior effect, which we can use to evaluate different candidate priors.

3 Empirical analysis of neural network weights

We trained fully connected neural networks (FCNNs) and convolutional neural networks (CNNs) with SGD on the task of MNIST handwritten digit recognition [LeCun et al., 1998] and analysed the empirical weight distributions. Intuitively, the more a BNN prior deviates from the empirical weight distribution reached by this training, the less probability mass the posterior will assign to the maximum-likelihood (ML) solution. This can be easily seen, since the posterior probability for the ML solution is $p(w_{ML}|x, y) \propto p(y|w_{ML}, x) p(w_{ML})$, which scales linearly with the prior probability $p(w_{ML})$. Since BNNs are generally rather believed to be underfitting [Neal, 1995, Wenzel]...
Figure 1: Empirical marginal weight distributions of FCNNs and CNNs trained with SGD on MNIST. We show marginal weight histograms and Q-Q plots with different distributions. It can be seen that heavy-tailed distributions (e.g., Laplace) yield a better fit than Gaussians for the empirical weight marginals.

Figure 2: Distributions of off-diagonal elements in the empirical covariances of the weights of FCNNs and CNNs trained with SGD on MNIST. The empirical distributions are plotted as histograms, while the idealized random Gaussian weights are overlaid in orange. We see that the covariances of the empirical weights are more heavy-tailed than for the Gaussian weights.

3.1 Are neural network weights heavy tailed?

We can see in Figure 1 that the weight values of the FCNNs and CNNs follow a more heavy-tailed distribution than a Gaussian. Judging from the Q-Q plots, they seem to be better approximated by a Laplace distribution. This suggests that BNN priors for these models might benefit from being more heavy-tailed than isotropic Gaussians.

3.2 Are neural network weights correlated?

In contrast to the isotropic Gaussian, the empirical weight distributions of FCNNs show some significant correlations among rows and columns of the weight matrices (Fig. B.1). This can be seen especially well by comparing the distribution of off-diagonal elements of the empirical covariance matrices (Fig. 2). We see that the empirical weights (blue histograms) have a more heavy-tailed distribution of off-diagonal elements than the randomly sampled Gaussian weights (orange kernel density estimate).

Interestingly, this is also true for CNN weights (Fig. B.2). The off-diagonal elements of their empirical covariance matrices are also more heavy-tailed than for isotropic Gaussian weights (Fig. 2). Crucially, most of these correlations seem to occur spatially, that is, between weights within the same CNN...
filter (Fig. 3). This could potentially be due to the smoothness and translation equivariance properties of natural images [Simoncelli, 2009].

These findings suggest that better priors could be designed by explicitly taking this correlation structure into account. We hypothesize that multivariate distributions with non-diagonal covariance matrices could be good candidates for CNN priors, especially when the covariances are large for neighboring pixels within the CNN filters (see Sec. 4.2).

4 Empirical study of Bayesian neural network priors

We again performed experiments on the MNIST handwritten digit data set [LeCun et al., 1998]. We compare Bayesian FCNNs and Bayesian CNNs on this task. For the BNN inference, we use Stochastic Gradient Markov Chain Monte Carlo (SG-MCMC), following Wenzel et al. [2020a] and Zhang et al. [2019]. Additional experimental results can be found in Appendix B, inference diagnostics in Appendix C, and implementation details in Appendix F.

4.1 Bayesian FCNN performance with different priors

Following our observations from the empirical weight distributions (Sec. 3.1), we hypothesized that heavy-tailed priors should work better than Gaussian ones for Bayesian FCNNs. We tested this by performing BNN inference with the same network architecture as above using different priors (for details about the priors, see Appendix D). We report the predictive error and log likelihood on the MNIST test set. We follow Ovadia et al. [2019] in reporting the calibration of the uncertainty estimates on rotated MNIST digits and the out-of-distribution (OOD) detection accuracy on FashionMNIST [Xiao et al., 2017]. For more details about our evaluation metrics, we refer to Appendix E.

We observe that the heavy-tailed priors do indeed outperform the Gaussian one for all metrics except for the calibration error (Fig. 4). This suggests that Gaussian priors over the weights of feedforward networks induce poor priors in the function space and inhibit the posterior from assigning probability mass to high-likelihood solutions, such as the SGD solutions analysed above (Sec. 3). Moreover, we find that the cold posterior effect is less pronounced when using heavy-tailed priors.

4.2 Bayesian CNN performance with different priors

We repeated the same experiment for Bayesian CNNs. Following our observations from the empirical weights (Sec. 3.1), in this case we might also expect the heavy-tailed priors to outperform the Gaussian one. The results in terms of performance alone are less striking here than in the FCNN experiments, and when cooling down the posterior, the Gaussian prior often outperforms the heavy-tailed ones (Fig. 5). However, in the true posterior ($T = 1$), we see that the heavy-tailed priors generally perform slightly better than the Gaussian, even though this effect might not be significant. Moreover, we again observe that the cold posterior effect is less pronounced with heavy-tailed priors.
We have shown that especially in fully-connected BNNs, heavy-tailed non-Gaussian priors can yield a better performance across many metrics and also fit the empirical weight distributions of maximum-likelihood solutions better. Moreover, they seem to partially alleviate the cold posterior effect.

In contrast, in convolutional BNNs, the performance benefit of heavy-tailed priors seems less obvious, although they also fit the empirical weights better and alleviate the cold posterior effect. Moreover, CNNs seem to exhibit significant correlations in the empirical weight distributions, especially between weights within a filter. Including such correlations into the prior improves the performance, but does not seem to alleviate the cold posterior effect.

Apart from the marginal weight priors, following our correlation analysis (Sec. 3.2), we would expect to improve the prior when introducing weight correlations. We did this by defining a multivariate Gaussian prior with nonzero covariance between weights within each CNN filter, where the covariance is defined by a radial basis function (RBF) kernel, such that it decays smoothly with increasing distance in pixel space. We describe this prior in more detail in Appendix D. For this correlated prior, we observe that it does indeed improve the performance compared to the isotropic Gaussian one (Fig. 6). However, the cold posterior effect is not reduced as significantly as in the previous experiments and thus remains more elusive for CNNs.

5 Conclusion

We have shown that especially in fully-connected BNNs, heavy-tailed non-Gaussian priors can yield a better performance across many metrics and also fit the empirical weight distributions of maximum-likelihood solutions better. Moreover, they seem to partially alleviate the cold posterior effect.
References


A Detailed Related Work

Empirical analysis of weight distributions. There is some history in neuroscience of analysing the statistics of data to inform inductive priors for learning algorithms, especially when it comes to vision [Simoncelli, 2009]. For instance, it has been noted that correlations help in modeling natural images [Srivastava et al., 2003], as well as sparsity in the parameters [Field, 1987; Olshausen and Field, 1997], and the use of Gaussian scale mixtures [Wainwright and Simoncelli, 1999; Lyu and Simoncelli, 2008]. In the context of machine learning, the empirical weight distributions of standard neural networks have also been studied before [Bellido and Piesler, 1993; Go and Lee, 1999], but these works have not systematically compared different architectures and did not use their insights to inform Bayesian prior choices.

BNNs in practice. Since the inception of Bayesian neural networks, scholars have thought about choosing good priors for them, including hierarchical [MacKay, 1992] and heavy-tailed ones [Neal, 1995]. In the context of infinite-width limits of such networks [Lee et al., 2017; Matthews et al., 2018; Garriga-Alonso et al., 2018; Yang, 2019; Tsuchida et al., 2019], it has also been shown that heavy-tailed priors can lead to more interesting properties than finite-variance ones [Neal, 1995; Peluchetti et al., 2020]. Moreover, it has been shown that the activations in deep neural networks grow more heavy-tailed with increasing depth [Vladimirova et al., 2019] and that the popular dropout regularization is related to sparsity-inducing priors [Nalisnick et al., 2019]. Nonetheless, most state-of-the-art BNN methods still use simple isotropic Gaussian priors [Osawa et al., 2019; Zhang et al., 2019; Maddox et al., 2019; Wilson and Izmailov, 2020; Dusenberry et al., 2020]. It has been hypothesized that this could be one of the reasons why BNNs are still not convincingly outperforming standard neural networks on many tasks [Wenzel et al., 2020a].

Alternative BNN priors. While many interesting distributions have been proposed as variational posteriors for BNNs [Louizos and Welling, 2017; Farquhar et al., 2019; Swiatkowski et al., 2020; Dusenberry et al., 2020], these approaches have still relied on simple Gaussian priors. Although a few different priors have been proposed for BNNs, these were mostly designed for specific tasks [Atanov et al., 2018; Ghosh and Doshi-Velez, 2017; Overweg et al., 2019; Nalisnick, 2018; Cui et al., 2020; Hafner et al., 2020] or relied heavily on non-standard inference methods [Sun et al., 2019; Ma et al., 2019; Karaletsos and Bui, 2020; Pearce et al., 2020].

Our work. In this work we explicitly study the question of whether non-Gaussian priors are generally useful (or even necessary) for Bayesian neural networks. We make an attempt to answer this question for different neural network architectures and also compare the priors to the empirical distributions of weights in networks trained via maximum likelihood. Moreover, we used reliable Markov Chain Monte Carlo (MCMC) BNN inference [Neal, 1992; Zhang et al., 2019; Wenzel et al., 2020a] in our experiments to be able to make claims about the true posteriors of the models. This is in contrast to many of the works mentioned above, which used variational inference approaches for sake or their computational benefits.

B Additional experimental results

B.1 Covariance matrices

Here we report the full covariance matrices for the layers that were analysed above (Sec. 3.2). The covariances of the FCNN weights are shown in Figure B.1 and of the CNN weights in Figure B.2.

B.2 Empirical weight results for the other layers

In Section 3 we exemplarily report results for the respective second layers of our FCNN and CNN. Here, we report the same results for the other layers for sake of completeness. The FCNN results are shown in Figures B.3, B.4, B.5, and B.6 and the CNN results in Figures B.7 and B.8.
We show the estimated temperatures in Figures C.1, C.2, and C.3, as a mean ± two standard errors. The desired temperature is shown as a dotted horizontal line. The kinetic temperatures generally agree well with the true temperatures, so our sampler works as expected there. The configurational temperature estimates have a higher variance than the kinetic ones. Their error bars mostly include the true temperatures, but their means are usually too low. Thus, the sampler is still within the tolerance levels of working correctly there, but there could be some small inaccuracies.

B.3 BNN performances of untempered posteriors

Here we report the BNN performances from the experiments in Section 4 for the true Bayesian posteriors \( T = 1 \) for ease of comparison. The FCNN performances with heavy-tailed and Gaussian priors are shown in Figure B.9, the CNN performances with heavy-tailed and Gaussian priors in Figure B.10 and the CNN performances with isotropic and correlated Gaussian priors in Figure B.11.

C Inference diagnostics

In order to check the correctness of our SG-MCMC inference, we estimated the temperature of the sampler using the two different methods from Wenzel et al. [2020a], namely the kinetic temperature and the configurational temperature.

The kinetic temperature is derived from the sampler’s momentum \( m \in \mathbb{R}^d \). The inner product \( \frac{1}{2}m^T M^{-1} m \), for the (in this case diagonal) mass matrix \( M \), is an estimate of the scaled variance of the momenta, and should, in expectation, be equal to the desired temperature. In contrast, the configurational temperature is \( \frac{1}{2} \theta^T \nabla H(\theta, m) \). In expectation, this should also equal \( T \). Using subsets of a parameter or momentum also yields estimators of the temperature.

In both cases, we estimate the mean and its standard error from a weighted average of parameters or momenta. That is, for each separate NN weight matrix or bias vector, we estimate its kinetic and configurational temperature using the expressions above. Then, we take their average, weighted by how many elements each matrix or vector has, and approximate its weighted standard error.

We show the estimated temperatures in Figures C.1, C.2, and C.3 as a mean ± two standard errors. The desired temperature is shown as a dotted horizontal line. The kinetic temperatures generally agree well with the true temperatures, so our sampler works as expected there. The configurational temperature estimates have a higher variance than the kinetic ones. Their error bars mostly include the true temperatures, but their means are usually too low. Thus, the sampler is still within the tolerance levels of working correctly there, but there could be some small inaccuracies.

Following the code from https://stats.stackexchange.com/q/33959
D Priors for Bayesian Neural Networks

In this study, we contrast the widely used Gaussian priors with more heavy-tailed priors, especially Laplace and Student-t distributions. We chose these distributions mostly based on our observations regarding the empirical weight distributions of trained networks (see Fig. 1 and Sec. 3) and for their ease of implementation and optimization. We now give a quick overview over these different distributions and their most salient properties.

**Gaussian** The Gaussian distribution [Gauss, 1809] is the de-facto standard for BNN priors in recent work [Wenzel et al., 2020a, Wilson and Izmailov, 2020, Zhang et al., 2019]. Its probability density function (PDF) is

\[
p(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right)
\]

It is attractive, because it is the central limit of all finite-variance distributions [Billingsley, 1961] and the maximum entropy distribution for a given mean and scale [Bishop, 2006]. However, its tails are relatively light compared to some of the other distributions that we will consider.

**Laplace** The Laplace distribution [Laplace, 1774] has heavier tails than the Gaussian and is discontinuous at \(x = \mu\). Its PDF is

\[
p(x; \mu, b) = \frac{1}{2b} \exp\left(-\frac{|x - \mu|}{b}\right)
\]

It is often used in the context of lasso regression, where it encourages sparsity in the learned weights [Tibshirani, 1996].

**Student-t** The Student-t distribution characterizes the mean of a finite number of samples from a Gaussian distribution with respect to the true mean [Student, 1908]. Its PDF is

\[
p(x; \mu, \nu) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)\sqrt{\nu\pi}} \left(1 + \frac{(x - \mu)^2}{\nu}\right)^{-\frac{\nu+1}{2}},
\]
Figure B.5: Empirical covariances of the weights of the FCNN in the third layer. We see that they contain more systematic correlations than the isotropic Gaussian.

Figure B.6: Distributions of singular values of the weight matrices of the FCNN in the other layers. We see that the spectra of the empirical weights decay faster than the ones of the Gaussian weights.

where \( \Gamma \) is the gamma function and \( \nu \) are the degrees of freedom. The Student-t also arises as the marginal distribution over Gaussians with an inverse-Gamma prior over the variances \( \text{[Helmert, 1875, Lüroth, 1876]} \). For \( \nu \rightarrow \infty \), the Student-t distribution approaches the Gaussian. For any finite \( \nu \) it has heavier tails than the Gaussian. Its \( k \)’th moment is only finite for \( \nu > k \). The \( \nu \) parameter thus offers a convenient way to adjust the heaviness of the tails. Unless otherwise stated, we set \( \nu = 3 \) in our experiments, such that the distribution has rather heavy tails, while still having a finite mean and variance.

**Multivariate Gaussian with RBF covariance** For our correlated Bayesian CNN priors, we use multivariate Gaussian priors

\[
p(x; \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^{d} \det \Sigma}} \exp \left( -\frac{1}{2} (x - \mu)^{T} \Sigma^{-1} (x - \mu) \right),
\]

where \( d \) is the dimensionality.

In our experiments, we set \( \mu = 0 \) and define the covariance \( \Sigma \) to be block-diagonal, such that the covariance between weights in different filters is 0 and between weights in the same filter is given by an RBF kernel on the pixel distances. Formally, for the weights \( w_{i,j} \) and \( w_{i',j'} \) in filters \( i \) and \( i' \) and for pixels \( j \) and \( j' \), the covariance is

\[
\text{cov}(w_{i,j}, w_{i',j'}) = \begin{cases} 
\sigma^{2} \exp \left( -\frac{d(j,j')}{\lambda} \right) & \text{if } i = i', \\
0 & \text{else},
\end{cases}
\]

where \( d(\cdot, \cdot) \) is the Euclidean distance in pixel space and we set \( \sigma = \lambda = 1 \).

**E Evaluation Metrics**

When using BNNs, practitioners might care about different outcomes. In some applications, the predictive accuracy might be the only metric of interest, while in other applications calibrated uncertainty estimates could be crucial. We therefore use a range of different metrics in our experiments in order to highlight the respective strengths and weaknesses of different priors. Moreover, we compare the priors to the empirical weight distributions of conventionally trained networks.
E.1 Empirical test performance

**Test error** The test error is probably the most widely used metric in supervised learning. It intuitively measures the performance of the model on a held-out test set and is often seen as an empirical approximation to the true generalization error. While it is often used for model selection, it comes with the risk of overfitting to the used test set [Bishop, 2006] and in the case of BNNs also fails to account for the predictive variance of the posterior.

**Test log-likelihood** The predictive log-likelihood also requires a test set for its evaluation, but it takes the predictive posterior variance into account. It can thus offer a built-in tradeoff between the mean fit and the quality of the uncertainty estimates. Moreover, it is a proper scoring rule [Gneiting and Raftery, 2007].

E.2 Uncertainty estimates

**Uncertainty calibration** Bayesian methods are often chosen for their superior uncertainty estimates, so many users of BNNs will not be satisfied with only fitting the posterior mean well. The calibration measures how well the uncertainty estimates of the model correlate with predictive performance. Intuitively, when the model is for instance 70% certain about a prediction, this prediction should be correct with 70% probability. Many deep learning models are not well calibrated, because they are often overconfident and assign too low uncertainties to their predictions [Ovadia et al., 2019; Wenzel et al., 2020b]. When the models are supposed to be used in safety-critical scenarios, it is often crucial to be able to tell when they encounter an input that they are not certain about [Kendall and Gal, 2017]. For these applications, metrics such as the expected calibration error [Naeini et al., 2015] might be the most important criteria.

**Out-of-distribution detection** The out-of-distribution (OOD) detection measures how well one can tell in-distribution and out-of-distribution examples apart based on the uncertainties. This is important when we believe that the model might be deployed under some degree of data set shift. In
Figure B.9: Performance of Bayesian FCNNs with different priors on MNIST in terms of different metrics. We see that the heavy-tailed priors perform better than the Gaussian ones, except in terms of expected calibration.

Figure B.10: Performance of Bayesian CNNs with different priors on MNIST in terms of different metrics. We see that the Gaussian prior performs as well as the other ones.

this case, the model should be able to detect these OOD examples and be able to reject them, that is, refuse to make a prediction on them.

F Implementation details

Training setup. For all the MNIST BNN experiments, we perform 20 cycles of SG-MCMC [Zhang et al., 2019] with 45 epochs each. We draw one sample each at the end of the respective last five epochs of each cycle. From these 100 samples, we discard the first 30 as a burn-in of the chain. Moreover, in each cycle, we only add Langevin noise in the last 15 epochs (similar to Zhang et al. [2019]). We start each cycle with a learning rate of 0.01 and decay to 0 using a cosine schedule. We use a mini-batch size of 128.

For the SGD experiments yielding the empirical weight distributions, we use the same settings, but do not add any Langevin noise. We also only take one sample at the very end of training.

FCNN architecture. For the FCNN experiments, we used a feedforward neural network with three layers, a hidden layer width of 100, and ReLU activations.

CNN architecture. For the CNN experiments, we use a convolutional network with two convolutional layers and one fully-connected layer. The hidden convolutional layers have 64 channels each and use $3 \times 3$ convolutions and ReLU activations. Each convolutional layer is followed by a $2 \times 2$ max-pooling layer.
Figure B.11: Performance of Bayesian CNNs with different priors on MNIST in terms of different metrics. We see that the correlated prior performs a bit better than the isotropic one.

Figure C.1: Temperature diagnostics of the MNIST experiment with FCNNs. The kinetic temperature estimates coincide with the desired temperature. The configurational temperature estimates are within error bars for all temperatures, but the mean is usually low. This is mild evidence that the sampler may not be entirely accurate.

Figure C.2: Temperature diagnostics of the MNIST experiment with CNNs and heavy-tailed priors. The conclusions are similar to Figure C.2.
Figure C.3: Temperature diagnostics of the MNIST experiment with CNNs and correlated priors. The conclusions are similar to Figure C.2 but here the kinetic temperature is slightly too hot.