On Uncertainty, Tempering, and Data Augmentation in Bayesian Classification

Sanyam Kapoor[*] New York University
Wesley J. Maddox[*] New York University
Pavel Izmailov[*] New York University
Andrew Gordon Wilson New York University

Abstract

Aleatoric uncertainty captures the inherent randomness of the data, such as measurement noise. In Bayesian regression, we often use a Gaussian observation model, where we control the level of aleatoric uncertainty with a noise variance parameter. By contrast, for Bayesian classification we use a categorical distribution with no mechanism to represent our beliefs about aleatoric uncertainty. Our work shows that explicitly accounting for aleatoric uncertainty significantly improves the performance of Bayesian neural networks. We note that many standard benchmarks, such as CIFAR-10, have essentially no aleatoric uncertainty. Moreover, we show that data augmentation in approximate inference softens the likelihood, leading to underconfidence and misrepresenting our beliefs about aleatoric uncertainty. Accordingly, we find that a cold posterior, tempered by a power greater than one, often more honestly reflects our beliefs about aleatoric uncertainty than no tempering — providing an explicit link between data augmentation and cold posteriors. We further show that we can match or exceed the performance of posterior tempering by using a Dirichlet observation model, where we explicitly control the level of aleatoric uncertainty, without any need for tempering.

1 Introduction

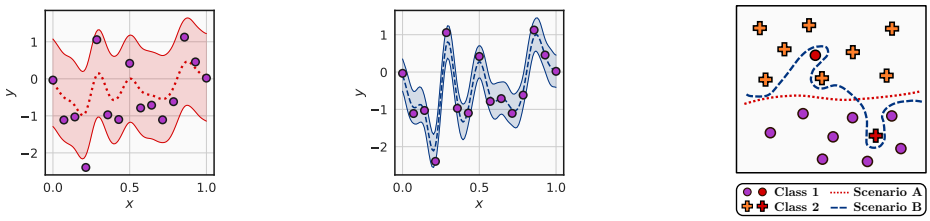
Uncertainty is often compartmentalized into *epistemic* and *aleatoric uncertainty* [27, 33, 41]. Epistemic uncertainty, sometimes called model uncertainty, is the reducible uncertainty about which solution is correct given limited information. Bayesian methods naturally represent epistemic uncertainty through a distribution over model parameters, leading to a posterior distribution over functions that are consistent with data. As we observe more data, this posterior concentrates around a single solution. Aleatoric uncertainty is the irreducible uncertainty in data, often representing measurement noise in regression, or mislabeled training points in classification [e.g. 4]. Although measurement noise can be reduced, for example, with better instrumentation, it is often a fixed property of the data we are given. Correctly expressing our assumptions about aleatoric uncertainty is crucial for achieving good predictive performance, both with Bayesian and non-Bayesian models [56].

In particular, our assumptions about aleatoric uncertainty profoundly affect predictive *accuracy*, not only predictive uncertainty.[1] In Figure 1(a,b) we show the predictive distributions of two Gaussian process (GP) regression models [54] trained on the same data and using the same RBF kernel function. The only difference between these models is their assumptions about the aleatoric uncertainty. The model in Figure 1(a) assumes a high observation noise (each point is corrupted by $\mathcal{N}(0, \sigma^2)$,

[*]Equal contribution.

[1]Epistemic uncertainty also has a significant effect on predictive accuracy, see Wilson and Izmailov [62].

36th Conference on Neural Information Processing Systems (NeurIPS 2022).



(a) GP regression, $\sigma^2 = 1$ (b) GP regression, $\sigma^2 = 10^{-2}$ (c) Classification

Figure 1: **Effect of aleatoric uncertainty.** In regression, we can express our assumptions about aleatoric uncertainty by setting the observation noise parameter σ^2. **(a)**: GP regression with high observation noise σ^2 explains many of the observations with noise. The datapoints are shown with purple circles, the dashed line shows the predictive mean, and the shaded region shows one standard deviation of the predictive distribution. **(b)**: GP regression on the same data, but with low observation noise σ^2 fits the training data nearly perfectly. **(c)**: In classification, we do not have a direct way to specify our assumptions about aleatoric uncertainty. We might use the same Bayesian neural network model if we know the data contains label noise (**Scenario A**) and if we know that there is no label noise (**Scenario B**), leading to poor performance in at least one of these scenarios.

with $\sigma^2 = 1$). Consequently, this model explains many of the observations with noise — such that the predictive mean does not closely fit the training datapoints. The model in Figure 1(b) assumes a low observation noise ($\sigma^2 = 10^{-2}$), and consequently the predictive mean runs through the training data, leading to very different predictions compared to the model in Figure 1(a).

Our assumptions about the aleatoric uncertainty are just as important in classification. We illustrate this point in Figure 1(c), where we fix a dataset and consider two possible scenarios. **Scenario A**: suppose we know that the data contains label noise, with some incorrectly labeled training examples. In this case, it is reasonable to believe that the data points shown in red in Figure 1(c) are highly likely to be incorrectly labeled, and we should opt for a solution with a decision boundary shown with a dotted red line. **Scenario B**: now, suppose that we know that the data are correctly labeled. In this case, we want our model to perfectly classify all the training data and would prefer a solution shown by the dashed blue line. Both of these solutions are reasonable descriptions of the data, and we can only make an informed choice between them by incorporating our assumptions about the aleatoric uncertainty. While in regression problems we can easily express our assumptions about aleatoric uncertainty, such as through a noise variance parameter in a Gaussian likelihood, in classification we do not have a direct way to express these beliefs. Indeed, in practice we might use the same Bayesian neural network (BNN) with the same likelihood and prior in both scenarios A and B above, necessarily leading to poor performance in at least one of these scenarios.

In this paper, we explicitly characterize aleatoric uncertainty in BNN classification. We show that *cold posteriors* [61] more honestly reflect our beliefs about aleatoric uncertainty than standard Bayesian posteriors. Further building on our analysis of aleatoric uncertainty in Bayesian classification, we describe an alternative to cold posteriors called the *noisy Dirichlet model*. The noisy Dirichlet model can be viewed as simply modifying the prior over the parameters of a BNN to put more mass on the solutions that make *highly confident* predictions on the training data, but only in scenarios with low aleatoric uncertainty. Unlike standard BNN classifiers, we get control over the aleatoric uncertainty through the prior. This approach achieves competitive performance on image classification problems with a valid Bayesian model and *removes the cold posterior effect*.

Finally, we show theoretically how data augmentation counterintuitively softens the likelihood, leading to solutions that underfit the training data. These results precisely characterize the empirical link between data augmentation and the success of cold posteriors observed by Izmailov et al. [31] and Fortuin et al. [14], and show that cold posteriors in fact serve to *correct* for the effects of data augmentation on aleatoric uncertainty.

2 Related Work

Early work on Bayesian neural networks focused on hyperparameter learning and mitigating overfitting, using Laplace approximations, variational methods, and MCMC [e.g. 37, 25, 47]. More

recently, Wilson and Izmailov [62] show how Bayesian model averaging is especially compelling for *modern* deep neural networks. A number of recent works have focused on making Bayesian deep learning practical [e.g., 63, 38, 49, 65, 12, 11], with improved performance, and essentially no overhead.

Modeling label noise. In the context of label noise for classification, prior works focus on modifying the loss function. For example, Collier et al. [9, 10] propose input-dependent noise losses for label noise in classification to improve accuracy, but do not focus on the uncertainty estimates. Similarly, Zhang and Sabuncu [66] and Feng et al. [13] propose modifications of the standard cross-entropy loss to provide increased robustness to label noise, but do not focus on modeling the inherent uncertainty in label noise settings.

Dirichlet-based uncertainty (DBU). Recent literature proposes neural networks where the output is a Dirichlet distribution, capturing both the epistemic and aleatoric uncertainty [41, 57, 40, 42, 7, 67, 46, 59, 58, 34]. These models are generally proposed as efficient alternatives to ensembles and Bayesian models, and targeted for improved uncertainty quantification, anomaly detection and adversarial robustness. Joo et al. [32] reformulate the problem of minimizing the softmax cross-entropy loss over the data into a collection of distribution matching problems. In this paper, we study the Dirichlet distributions associated with the posterior over the predictive class probabilities, which arise naturally in standard Bayesian classification models. Contrary to the methods in the DBU family, we do not modify the observation model or the structure of the network.

Cold posterior effect. Wenzel et al. [61] demonstrated with several examples that raising the posterior to a power $1/T$, with $T < 1$, in conjunction with SGLD inference, improves performance over classical training, but $T = 1$ can provide worse results than classical training, in what has been termed the *cold posterior effect*. However, in a study using HMC inference, Izmailov et al. [31] showed that for *all* cases considered in Wenzel et al. [61] there is no cold posterior effect when data augmentation is removed. The sufficiency of data augmentation to observe the cold posterior effect has been confirmed by several works using SG-MCMC inference [14, 48, 45]. Several works have suggested that misspecified priors explain cold posteriors, arguing that the current default choice of isotropic Gaussian priors in BNNs are inadequate [61, 64, 14]. However, Fortuin et al. [14] find that the cold posterior effect cannot be alleviated for convolutional neural networks by using heavy tailed or correlated priors under data augmentation. With HMC inference and no data augmentation, Izmailov et al. [31] also show that standard Gaussian priors perform similarly to a range of other priors, such as heavy-tailed logistic or mixtures of Gaussian priors, and generally are high performing. Noci et al. [48] argue that many different factors — likelihoods, priors, data augmentation — can cause cold posteriors, and that there may be no one cause in general. Aitchison [2] argue that BNN likelihoods are misspecified since many benchmark datasets have been carefully curated by human labelers. They also show that posterior tempering is less helpful in the presence of label noise. Adlam et al. [1] also consider model misspecification, showing a cold posterior effect can arise even with exact inference in GP regression when the aleatoric uncertainty is misestimated. Nabarro et al. [45] modify the likelihood to accommodate data augmentation, using as the observation model the average of the neural network outputs over augmentations, but still find a cold posterior effect. In concurrent work, Bachmann et al. [3] provide an alternative argument to how data augmentations introduce misspecification due to correlation between errors which cold posteriors correct for.

Our paper makes several distinctive contributions in the context of prior work: (1) we argue that standard likelihoods do not represent our beliefs about aleatoric uncertainty, and that standard benchmarks have essentially no aleatoric uncertainty; (2) we show how tempering and data augmentation specify aleatoric uncertainty, beyond curation; (3) we show the *precise way in which data augmentation with SGLD leads to underconfidence in the likelihood*, a counterintuitive result which finally resolves the empirical connection between data augmentation and cold posteriors; (4) we show that a $T < 1$, particularly with data augmentation, is a more honest reflection of our beliefs about aleatoric uncertainty than $T = 1$; (5) we show how the noisy Dirichlet likelihood, originally used for tractable GP classification [43], can naturally reflect our beliefs about aleatoric uncertainty, and for the first time remove the cold posterior effect in the presence of data augmentation.

3 Background

Bayesian Model Averaging. With Bayesian inference, we aim to infer the posterior distribution over parameters having observed a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ of input-output pairs, given by $p(w \mid \mathcal{D}) \propto p(\mathcal{D} \mid w)p(w)$ for a given observation likelihood under the i.i.d. assumption $p(\mathcal{D} \mid w) = \prod_{(x,y)\in\mathcal{D}} p(y \mid x, w)$, and prior over parameters $p(w)$. For any novel input $x_\star$, we compute the posterior predictive distribution via *Bayesian model averaging* (BMA) given by

$$p(y_\star \mid x_\star) = \int p(y_\star \mid x_\star, w)p(w \mid \mathcal{D})dw. \tag{1}$$

The BMA integral in Eq. (1) cannot generally be expressed in closed form for a Bayesian neural network, and is typically approximated with variational inference (VI), the Laplace approximation, or Markov Chain Monte Carlo (MCMC) [e.g., 62, 44].

Bayesian Classification. For a C-class classification problem, a standard choice of observation likelihood is the categorical distribution $p(y \mid x, w) = \mathrm{Cat}([\pi_1, \ldots, \pi_C])$, where each class probability is computed using a softmax transform of logits $g(x; w) = [g_1(x; w), \ldots, g_C(x; w)] \in \mathbb{R}^C$ as $\pi_c \propto f_c(x; w) = \exp\{g_c(x; w)\}$, and hence called the *softmax likelihood*. The prior over parameters $p(w)$ is often chosen to be an isotropic Gaussian $\mathcal{N}\left(0, \sigma^2 I\right)$ with some fixed variance σ^2. Approximate posterior samples are often obtained via SGLD [60], and used for a Monte Carlo estimator of the BMA in Eq. (1).

Cold Posteriors. Let $U(w) = -\log p(\mathcal{D} \mid w) - \log p(w)$ denote the posterior energy function of a BNN. Following Wenzel et al. [61], we define a *cold posterior* for $T < 1$ as

$$p_{\mathrm{cold}}(w \mid \mathcal{D}) \propto \exp\left\{-\frac{1}{T}U(w)\right\}, \tag{2}$$

which effectively raises both the likelihood and the prior to a power $1/T$. $T = 1$ recovers the standard Bayes' posterior. A *tempered likelihood posterior* (e.g. [17]) only raises the likelihood to a power $1/T$ as

$$p_{\mathrm{temp}}(w \mid \mathcal{D}) \propto p(\mathcal{D} \mid w)^{1/T} p(w). \tag{3}$$

4 Aleatoric Uncertainty in Bayesian Classification

We will now discuss how we represent aleatoric uncertainty in classification (Section 4.1), how tempering reduces aleatoric uncertainty (Section 4.2), and how we can explicitly represent aleatoric uncertainty without tempering with a noisy Dirichlet observation model (Section 4.3).

4.1 How do we represent aleatoric uncertainty in classification?

Let us consider the Bayesian neural network posterior over parameters w in a classification problem: $p(w \mid D) \propto p(w) \prod_{(x,y)\in\mathcal{D}} f_y(x; w)$, where we denote the output of the softmax layer of the model corresponding to class y on input x as $f_y(x; w)$.

Alternatively, we can think of the class probability vectors $f(x) \triangleq f(x; w) = [f_1(x; w), \ldots, f_C(x; w)]$ as latent variables[2] in a $C - 1$ simplex Δ^{C-1}, such that $\sum_{c=1}^C f_c(x; w) = 1$. A prior distribution $p(w)$ over the parameters of the network implies a joint prior distribution over $f = \{f(x)\}_{x\in\mathcal{D}}$. We can then do inference directly in *function space* over the variables f, yielding a posterior $p(f \mid \mathcal{D})$ equal to the distribution over f implied by the the parameter space posterior $p(w \mid \mathcal{D})$ [see e.g. 54, Ch. 2.2]. We will use this function space posterior to analyze the assumptions on aleatoric uncertainty encoded by the Bayesian classification models.

Let us consider the likelihood $p(y \mid f(x))$ for a single observation (x, y):

$$\begin{aligned} p(y \mid f(x)) &= f_y(x) \propto_{f(x)} C! \cdot f_y(x) \\ &= \mathrm{Dir.}(1, \ldots, 1, \underbrace{2}_{\text{position } y}, 1, \ldots, 1)(f(x)), \end{aligned} \tag{4}$$

[2] We use $f(x; w)$ and $f(x)$ interchangeably to denote the latent variable corresponding to the class probability vector on the input x. We use $f(x; w)$ to emphasize the dependence on the weights w.

where the left hand side is equal to the right hand side of Eq. (4) up to a scalar multiplier $C!$ (normalizing constant of the Dirichlet distribution), which does not depend on $f(x)$.[3] We note that the algebra in Eq. (4) is the same as the algebra used to show the conjugacy of the Dirichlet prior and the categorical likelihood [e.g. 6, Ch. 2.2]. Eq. (4) describes the distribution induced by the observation y on the predicted class probabilities for the corresponding input x by the likelihood. Consequently, the joint posterior over the class probability vectors $f = \{f(x)\}_{x \in \mathcal{D}}$ with the full dataset can then be written as

$$p(f \mid \mathcal{D}) \propto_f \quad p(f) \prod_{(x,y) \in \mathcal{D}} \text{Dir.}(1, \ldots, 1, \underbrace{2}_{\text{position } y}, 1, \ldots, 1)(f(x)). \tag{5}$$

Eq. (5) provides intuition for how BNNs estimate aleatoric uncertainty in classification. If we further assume that the implied prior $p(f)$ over $f(x)$ for each x is uniform on Δ^{C-1} and independent across different inputs x, then for an observation (x, y) the posterior over $f(x)$ is Dir.$(1, \ldots, 2, \ldots, 1)$ and the posterior mean for the probability of the correct class y is $\mathbb{E}[f_y(x)] = 2/(C+1)$. For example, with a dataset containing 100 classes (e.g. CIFAR-100), the model on average will only be less than 2% confident in the correct label on the *training data*!

In practice, however, the prior $p(w)$ will imply a non-trivial joint distribution $p(f)$ over $\{f(x; w)\}_{x \in \mathcal{D}}$, so the actual posterior may be more (or less) confident than suggested by the analysis above. Moreover, we have a very limited understanding of the implied distribution $p(f)$ [see 61, 62, for empirical analysis]. In particular, most practitioners use simple $p(w) = \mathcal{N}(0, \alpha^2 I)$ priors, regardless of the amount of label noise in the data. Explicitly constructing a prior in the parameter space that leads to highly confident predictive distributions over $f(x)$ is challenging, but we will show how modifying the likelihood can be viewed as providing an input-dependent prior that more accurately reflects our beliefs about aleatoric uncertainty in Section 4.3.

Eq. (5) suggests two natural ways of modifying the posterior to account for the aleatoric uncertainty: (i) increasing the Dirichlet concentration for the observed class (equivalent to likelihood tempering), or (ii) decreasing the concentration for the unobserved classes (noisy Dirichlet model). Next, we describe both of these approaches in detail.

4.2 Likelihood tempering reduces aleatoric uncertainty

The tempered likelihood posterior over the parameters w in Eq. (3) for BNN classification can be written as $p_{\text{temp}}(w \mid \mathcal{D}) \propto_w p(w) \prod_{(x,y) \in \mathcal{D}} f_y(x; w)^{1/T}$, where T is the temperature. Analogous to the derivation above, we can rewrite the corresponding function space posterior as

$$p_{\text{temp}}(f \mid \mathcal{D}) \propto_f \quad p(f) \prod_{(x,y) \in \mathcal{D}} \text{Dir.}\big(1, \ldots, 1, \underbrace{1 + 1/T}_{\text{position } y}, 1, \ldots, 1\big)(f(x)), \tag{6}$$

In other words, the tempered posterior corresponds to the same model as the regular posterior, but assumes that we observed $1/T$ counts of class y for each input x in the training data. Assuming the prior $p(w)$ implies a uniform distribution over $f(x; w)$, the confidence in the correct labels on the train data under the tempered posterior $\mathbb{E}[f_y(x)] = (1 + 1/T)/(C + 1/T)$. For a dataset with 100 classes, at $T = 10^{-2}$ we get a much higher average confidence of more than 50% on the training data, much higher than the 2% for the standard observation model.

Tempering the Likelihood vs Posterior. Prior work has mostly considered tempering the full Bayesian posterior as opposed to just the likelihood (see Section 3). In Appendix G.2 we show that tempering the full Bayesian posterior is almost always equivalent to changing the prior distribution, and tempering the likelihood, and that likelihood tempering recovers the cold posterior effect.

How should we think about tempering? Prior work has asserted that posterior tempering sharply deviates from the Bayesian paradigm and the cold posterior effect is highly problematic [61, 48, 14]. We, on the other hand, argue that likelihood tempering is in fact a practical way to incorporate our assumptions about aleatoric uncertainty. Relatedly, Aitchison [2] argue that the cold posterior effect can be caused by data curation. For example, CIFAR-10 and ImageNet datasets have relatively very little label noise and are carefully curated [e.g. 53]. Thus, we may expect that tempered likelihoods with low temperatures will lead to optimal performance.

[3] We use notation $\propto_\theta$ to denote proportionality with respect to θ.

Is a Tempered Likelihood a Valid Likelihood? The observation model is a distribution $p(y \mid x)$ over the labels conditioned on the input. The tempered softmax likelihood is in general not a valid likelihood because it does not sum to 1 over classes. But Wenzel et al. [61] argue that the tempered softmax likelihood with $T < 1$ can be interpreted as a valid likelihood if we introduce a new class which is not observed in the training data. Moreover, from Eq. (6), we can naturally interpret the tempered likelihood as using the multinomial observation model, assuming $1/T$ counts of the label are observed for each of the training datapoints, which is uncontroversial and perfectly valid.

Smoothed softmax model. Guo et al. [19] show that softmax temperature scaling can be used to fix the uncertainty calibration of trained neural networks. Analogously, we can simply divide the logits of the model by a scalar temperature T, before computing the softmax likelihood, resulting in the *smoothed softmax model*, a valid Bayesian model with a tunable temperature parameter. In Appendix F, we show that the smoothed softmax behaves differently than the tempered softmax likelihood, and is insufficient to model the aleatoric uncertainty in Bayesian classification. In particular, Figure 10 shows that the smoothed softmax likelihood does not address the cold posterior effect.

4.3 Noisy Dirichlet model: changing the prior over class probabilities

As we have seen in Eq. (6), likelihood tempering increases the posterior confidence by increasing the Dirichlet concentration for the observed class y from 2 to $1 + 1/T$. We can achieve a similar effect by *decreasing the concentration of the unobserved classes*[4]. Indeed, consider the distribution

$$p_{\mathrm{ND}}(f \mid \mathcal{D}) \propto_f \; p(f) \prod_{(x,y)\in\mathcal{D}} \mathrm{Dir.}(\alpha_\epsilon, \ldots, \alpha_\epsilon, \underbrace{\alpha_\epsilon + 1}_{\text{position } y}, \alpha_\epsilon, \ldots, \alpha_\epsilon)(f(x)), \tag{7}$$

where ND in p_{ND} stands for *noisy Dirichlet* and α_ϵ is a tunable parameter. The noisy Dirichlet model was originally proposed by Milios et al. [43] in the context of Gaussian process classification, where they designed a tractable approximation to this model. Using our running example, if $p(w)$ induces a uniform distribution over $f(x; w)$, for a problem with 100 classes and $\alpha_\epsilon = 10^{-2}$ we have expected confidence $\mathbb{E}[f_y(x)] = (\alpha_\epsilon + 1)/(C\alpha_\epsilon + 1)$, slightly more than 50%, similar to the tempered likelihood with $T = 10^{-2}$.

Going back to the parameter space, we can write the posterior distribution over w corresponding to the noisy Dirichlet model as

$$p_{\mathrm{ND}}(w \mid \mathcal{D}) \propto_w \; p(w) \prod_{(x,y)\in\mathcal{D}} \mathrm{Dir.}(\alpha_\epsilon, \ldots, \alpha_\epsilon, \underbrace{\alpha_\epsilon + 1}_{\text{position } y}, \alpha_\epsilon, \ldots, \alpha_\epsilon)(f(x; w)). \tag{8}$$

The noisy Dirichlet model exactly corresponds to a different prior. Using the conjugacy of the Dirichlet and multinomial distributions [e.g., 6, Ch. 2.2.1], we can rewrite p_{ND} from Eq. (8) as follows:

$$\begin{aligned} p_{\mathrm{ND}}(w \mid \mathcal{D}) \propto_w \; p(w) \prod_{(x,y)\in\mathcal{D}} \mathrm{Dir.}(\alpha_\epsilon, \ldots, \alpha_\epsilon)(f(x; w)) \cdot f_y(x; w) = \\ q_{\mathrm{ND}}(w) \prod_{(x,y)\in\mathcal{D}} f_y(x; w). \end{aligned} \tag{9}$$

We can interpret $q_{\mathrm{ND}}(w) \triangleq p(w) \cdot \prod_{x\in\mathcal{D}} \mathrm{Dir.}(\alpha_\epsilon, \ldots, \alpha_\epsilon)(f(x; w))$ as a prior over the parameters w of the model. Indeed, $q_{ND}(w)$ does not depend on the labels y, and simply forces the predicted class probabilities to be confident in any one of the classes for the training data. Then, the only difference between the noisy Dirichlet model in Eq. (9) and the standard Bayesian neural network for classification is the choice of the prior over the parameters: instead of using a generic prior $p(w)$ such as the standard Gaussian distribution, we use the noisy Dirichlet prior q_{ND}. See the *EmpCov* prior in Izmailov et al. [30] for another example of a prior that depends on the training inputs but not training labels.

The noisy Dirichlet prior is intuitively appealing: in many practical settings a priori we believe that the aleatoric uncertainty on the training data is low, and the model should be confident in one of the

[4] While we discuss increasing the confidence of the model here, we can equivalently decrease the confidence of the model by *increasing* the concentration parameters for the unobserved classes.

classes. At the same time, we would not want to enforce low aleatoric uncertainty everywhere in the input space, as we do not expect that all the possible inputs to the model should be classified with high confidence.

The noisy Dirichlet model is a valid Bayesian model. In short, the noisy Dirichlet model corresponds to using a *valid likelihood* — the softmax likelihood — with a prior that explicitly enforces the model predictions to be confident on the training data. In Section 6.2 we will see that the noisy Dirichlet model removes the cold posterior effect.

Gaussian approximation. Milios et al. [43] considered the distribution in Eq. (9) in the context of Gaussian process classification, but with a different goal: they aimed to create a regression likelihood which would approximate the softmax likelihood. We describe their Gaussian approximation which amounts to solving a regression problem in the space of logits $z(x)$ with a Gaussian observation model in Appendix D. While theoretically we do not need to use the Gaussian approximation for Bayesian neural networks and can directly use Eq. (9), we found that in practice the approximation is much more stable numerically. We use the Gaussian approximation in all experiments.

5 The Effect of Data Augmentation

Data augmentation is a key ingredient of modern deep learning. Surprisingly, however, data augmentation has not been studied in the context of Bayesian methods until recently. Several works have shown that data augmentation often causes the cold posterior effect in BNNs [31, 64, 14, 45]. We show that data augmentation, as applied in practice[5], is closely connected to likelihood tempering with $T > 1$. In this case, the cold posteriors counterbalance the effect of data augmentation.

First, we consider the interaction between data augmentation and SGLD (or, equivalently, any other SGMCMC method). With SGLD, we aim to construct an unbiased estimator of the gradient of the posterior log-density (see Appendix C). In prior work, e.g. in Wenzel et al. [61], the stochastic gradient is estimated using a randomly augmented mini-batch of the data. For a minibatch of the full dataset $\mathcal{D}_m = \{x_i, y_i\}_{i=1}^m \subset \mathcal{D}$, and a finite set of augmentations $\mathcal{T} = \{t_1, \ldots, t_K\}$, this stochastic gradient is given by $\nabla \widetilde{U}(w) = \frac{N}{m} \sum_{(x_i,y_i)\in\mathcal{D}_m} \nabla_w \log p(y_i \mid t_j(x_i)) + \nabla_w \log p(w)$, where the transformations t_j are sampled uniformly from $\mathcal{T}$. There are two sources of randomness in $\nabla \widetilde{U}(w)$ — the choice of the mini-batch $\mathcal{D}_m$ and the choice of augmentation t_j used for each x_i. The limiting distribution of SGLD is determined by the expectation of the stochastic gradient above, given by

$$\sum_{i=1}^{N} \mathbb{E}_t[\nabla_w \log p(y_i|t(x_i))] + \nabla_w \log p(w) = \sum_{i=1}^{N} \sum_{j=1}^{K} [\nabla_w \log p(y_i|t_j(x_i))^{\frac{1}{K}}] + \nabla_w \log p(w), \tag{10}$$

where $t_j \sim \mathcal{T}$ are augmentations sampled uniformly from $\mathcal{T}$. Therefore, we conclude that the limiting distribution of SGLD under data augmentation is given by

$$p_{\text{aug}}(w \mid \mathcal{D}) \propto_w \quad p(w) \prod_{i=1}^{N} \prod_{j=1}^{K} p(y_i \mid t_j(x_i))^{1/K}. \tag{11}$$

We can interpret the limiting distribution in Eq. (11) as a *tempered likelihood posterior* for a new dataset $\mathcal{D}' = \{(t_j(x_i), y_i)\}_{i,j}$ which contains all augmentations of every data point from the original dataset $\mathcal{D}$. Furthermore, the likelihood is tempered with a temperature $K > 1$, corresponding to a *warm posterior*. In other words, the limiting distribution of SGLD with data augmentation corresponds to a posterior over the augmented dataset with a *softened* likelihood, leading to less confidence about the labels. By applying a cold temperature $T = 1/K$ to the posterior in Eq. (11), we can recover the standard Bayesian posterior on the augmented dataset $\mathcal{D}'$. In Section 6.2, we explore the effect of data augmentation on aleatoric uncertainty in practice and show that it does indeed soften the likelihood and lead to increased aleatoric uncertainty.

[5] Data augmentation is a practical procedure, where at each iteration of optimization or sampling, we draw random augmentation for each object in each mini-batch of the data. Consequently, we can only reason about the effect of the data augmentation on the limiting distributions of approximate inference procedures: data augmentation is not defined for *true* Bayesian neural networks.

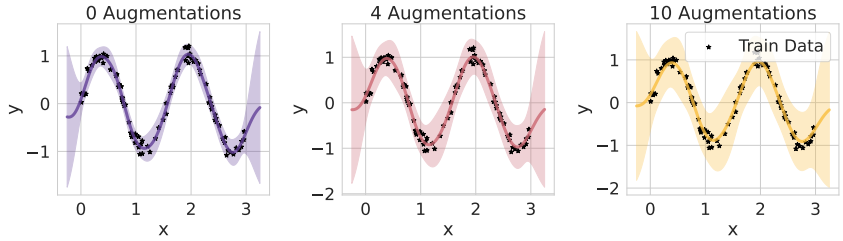


Figure 2: **Effect of data augmentation on GP regression.** As we increase the number K of augmentations of the dataset, a GP model fit becomes more diffuse. Here the original training data is shown with black stars, and the augmented datapoints $t_j(x) = x + \tau_j$ are not shown. Both the predictive mean and confidence change with the number K of augmentations.

What about Other Inference Procedures? While the derivation in Eq. (11) comes directly from studying the stochastic gradient evaluated in SGMCMC, we can equivalently derive the same posterior distribution in stochastic variational inference [26] as we show in Appendix I.

Should we always use temperature $T = 1/K$? While using the temperature $T = 1/K$ recovers the standard Bayesian posterior on the augmented dataset $\mathcal{D}'$, it is not necessarily a correct approach to modeling data augmentation. Indeed, the standard posterior $p(w \mid \mathcal{D}') \propto p(w) \prod_{i=1}^{N} \prod_{j=1}^{K} p(y_i \mid t_j(x_i))$ assumes independence across both augmentations and data points. In practice, however, we share the same label y_i across all the augmented versions of the image; treating the observations as independent then leads to *underestimating* the aleatoric uncertainty. Indeed, consider the extreme scenario, where all the augmentations $t_j(x)$ simply return x. In this case, treating the observations $(t_j(x), y)$ as independent will simply raise the likelihood to a power of K, while in reality we have not received any additional information from the augmentation policy. Correcting the likelihood by a factor of $1/K$ is the correct approach only when the predictions on $t_j(x)$ are completely independent from each other (see Section 6.1). See also Nabarro et al. [45] for a related discussion.

6 Experiments

In this section we provide empirical support for the observations presented in Sections 4 and 5. First, in Section 6.1 we visualize the effect of data augmentation on the limiting distribution of SGMCMC using a Gaussian process regression model. Next, in Section 6.2 we report the results for BNNs on image classification problems. In Appendix E we additionally illustrate the effects of tempering, noisy Dirichlet model and data augmentation using a Bayesian neural network on a synthetic 2-D classification problem. Appendix K provides detailed numerical results for reference. All results report the mean and one standard deviation over 3 trials. The code to reproduce experiments is available at github.com/activatedgeek/understanding-bayesian-classification.

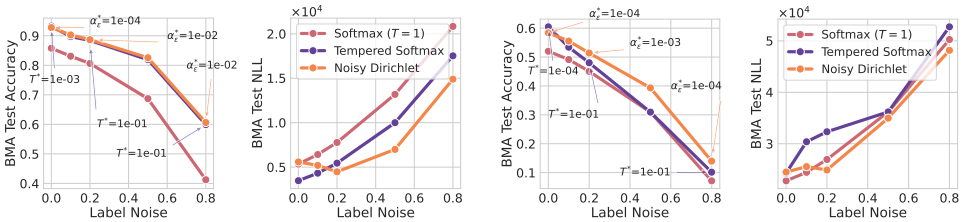
6.1 Visualizing the Effect of Data Augmentation

To illustrate our analysis in Section 5 we show the posterior in Eq. (11) under data augmentation in a Gaussian process (GP) regression model. We use a GP with a RBF kernel, noting that data points where $k(x, x') \approx 0$ are effectively independent from each other, which occurs when $\|x - x'\| \gg \delta$. Consequently, the implied correlations between inputs beyond the distance δ are zero. We use augmented samples that are given by $t_j(x) = x + \tau_j$, for vectors $\{\tau_j\}_{j=1}^{K}$ such that $\|\tau_j\| \gg \delta$, assuming that $k(x, t_j(x')) = 0$ for all datapoints x, x' and all augmentations t_j. Thus, the outputs of the GP on the training data are independent from the outputs on the augmented datapoints.

We show the posterior near the original data in Eq. (11) for $K = 1, 4$ and 10 augmentations per datapoint in Figure 2. As the predictions on the train data are independent from the predictions on the augmented datapoints, the posterior in Eq. (11) corresponds to tempering the posterior on the original training data with a warm temperature equal to the number K of augmentations. As a result, increasing K softens the likelihood, leading to a less confident fit on the training data.

6.2 Image Classification with Bayesian Neural Networks

In this section we experimentally verify the results of the analysis in Sections 4 and 5 for Bayesian neural networks in image classification problems. First, we show that both the noisy Dirichlet model and tempered softmax likelihood can be successfully used to express our beliefs about the amount of label noise in the data. Then, we show that the noisy Dirichlet model does not require tempering to achieve optimal performance. Finally, we show that data augmentation softens the likelihood for BNNs, and that the optimal temperature depends on the complexity of the data augmentation policy. For all experiments we use a ResNet-18 model [20] and the CIFAR-10 [35] and Tiny Imagenet [36] datasets. We use the SGLD sampler with a cyclical learning rate schedule [60, 65] to sample from the posterior. We provide details on the hyper-parameters in Appendix L.



(a) CIFAR-10 (b) Tiny Imagenet

Figure 3: **Label noise in image classification.** BMA test accuracy and negative log likelihood for the softmax, tempered softmax and noisy Dirichlet model **(a)** on CIFAR-10 and **(b)** Tiny Imagenet. Accounting for the label noise via the noisy Dirichlet model or the tempered softmax likelihood significantly improves accuracy accross the board. Moreover, the best performance is achieved by different values of temperature T^* in the tempered softmax likelihood or noise α_ϵ^* in the noisy Dirichlet model, i.e. no one model describes the data optimally across all levels of label noise.

Modeling label noise. In classification problems the aleatoric uncertainty corresponds to the amount of label noise in the data. Throughout the paper we argued that aleatoric uncertainty is misrepresented by standard Bayesian neural networks, and that likelihood tempering and the noisy Dirichlet model are compelling approaches to incorporate information about the amount of label noise. In Figure 3, we show the BMA test accuracy and negative log-likelihood for the standard softmax likelihood, tempered softmax likelihood and the noisy Dirichlet model on CIFAR-10 and Tiny Imagenet under varying amounts of label noise. We plot the results for the best performing temperature $T \in [10^{-5}, 10]$ or noise $\alpha_\epsilon \in [10^{-6}, 10^{-1}]$ for the noisy Dirichlet model. On both datasets and across label noises, we can significantly improve performance over the softmax likelihood by explicitly modeling the aleatoric uncertainty with either tempering or the noisy Dirichlet model. Moreover, different amounts of label noise require different values of T and α_ϵ parameters.

No cold posterior effect in the noisy Dirichlet model. We explore the effect of posterior tempering on the standard softmax classification likelihood and the noisy Dirichlet model with noise parameter $\alpha_\epsilon = 10^{-6}$. In Figure 4 we show the results on the CIFAR-10 dataset. As reported by Wenzel et al. [61], for the standard softmax likelihood, tempering is required to achieve optimal performance, with $T = 10^{-3}$ providing the best results. For the noisy Dirichlet model on the other hand, tempering does not significantly improve the results with roughly constant performance across the temperature values. In particular, the noisy Dirichlet model achieves near-optimal results at $T = 1$. These results agree with our analysis in Section 4: both tempering and the noisy Dirichlet model are alternative ways of expressing our beliefs about the aleatoric uncertainty in the data; with the noisy Dirichlet model, we can achieve strong results without any need for tempering.

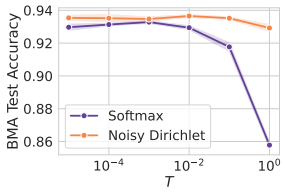
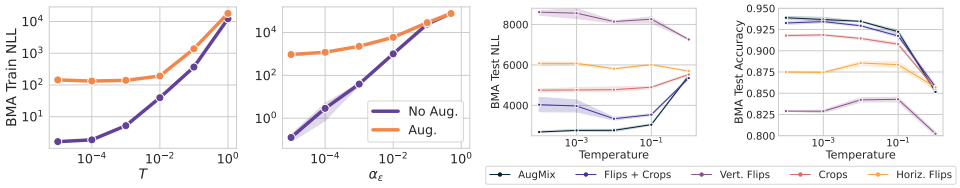


Figure 4: BMA test accuracy for the noisy Dirichlet model with $\alpha_\epsilon = 10^{-6}$ and the softmax likelihood as a function of temperature on CIFAR-10. The noisy Dirichlet model shows no cold posterior effect.

Data augmentation leads to underfitting on train data. In Figure 5a we show the BMA train negative log-likelihood for the models trained with and without data augmentation. Across temperatures for the tempered softmax and α_ϵ parameters for the noisy Dirichlet model, adding data augmentations reduces the quality of the



(a) Softmax & noisy Dirichlet

(b) Augmentations vs Tempering

Figure 5: **Effect of data augmentation on BNN image classification.** For all plots we use ResNet-18 on the CIFAR-10 dataset. **(a)**: The BMA negative log-likelihood on the train data for the tempered softmax model across different temperatures T and the noisy Dirichlet model across different values of noise α_ϵ, with and without data augmentation. As predicted in Section 5, data augmentation softens the likelihood, leading to a more diffuse fit on the train data. Tempering or reducing the noise parameter α_ϵ is needed to counteract the effect of data augmentation. **(b)** BMA test accuracy and NLL for various augmentation policies. As predicted in Section 5, more complex policies corresponding to a higher effective number of augmentations K require lower temperatures.

fit on the original training data. This result is analogous to the GP results in Section 6.1 and our analysis in Section 5: data augmentation softens the likelihood, leading to a more diffuse fit.

Complex augmentations require lower temperatures. In Section 5 we showed that under data augmentation the likelihood is effectively tempered with a temperature K, assuming the predictions on the augmented datapoints are completely independent from the original datapoints. In practice however, the augmentations can be close to the original images, and the corresponding predictions can be highly correlated. In Figure 5b, we consider five separate types of augmentations for ResNet-18 on CIFAR-10 at various posterior temperatures: horizontal and vertical flips, random crops, combinations of flips and crops, and AugMix [23] — an augmentation policy employing a very diverse set of transformations. We find that, as predicted by the analysis in Section 5, the optimal temperatures are different for different policies: in terms of test NLL, the simple policies (vertical and horizontal flips) corresponding to $K = 2$ work best at warmer temperatures $T = 1$, intermediate policies (crops and crops+flips) corresponding to $K \approx 100$ require lower temperatures $T \in [10^{-2}, 10^{-1}]$, and the most complex AugMix policy requires the lowest temperature $T \leq 10^{-4}$.

Robustness to corruptions. Finally, we consider robustness to dataset corruptions by considering the CIFAR-10C dataset [22], evaluating our models with both tempered softmax (TS) and noisy Dirichlet models. A full plot is shown in Appendix K; summarizing those results, we see that TS has the best performance at $T = 10^{-3}$ with an accuracy of 93.2 (with $T = 1$ again under-performing), while again the noisy Dirichlet model performs well, with an accuracy of 92.6 (at $\alpha_\epsilon = 10^{-4}$).

7 Discussion

A correct representation of aleatoric uncertainty is crucial to achieve strong performance with Bayesian neural networks. Standard Bayesian classifiers have no explicit mechanism to represent our beliefs about aleatoric uncertainty, which can lead to inadequate fits to the training data. This effect is exaggerated by data augmentation, which softens the likelihood, making the models under-confident on the training data. Posterior tempering is a simple and practical way to correct for this misspecification and express our beliefs about the aleatoric uncertainty: most benchmark datasets have very low levels of label noise, and we can express this belief by tempering the softmax likelihood. We also showed that we can achieve a similar effect without tempering, by using a prior that forces the model to be confident on the training datapoints with the noisy Dirichlet model.

Acknowledgements

We would like to thank Micah Goldblum and Wanqian Yang for helpful comments. This research is supported by an Amazon Research Award, NSF I-DISRE 193471, NIH R01DA048764-01A1, NSF IIS-1910266, and NSF 1922658 NRT-HDR: FUTURE Foundations, Translation, and Responsibility for Data Science. This work is supported in part through the NYU IT High Performance Computing resources, services, and staff expertise.

References

[1] Ben Adlam, Jasper Snoek, and Samuel L Smith. Cold posteriors and aleatoric uncertainty. *arXiv preprint arXiv:2008.00029*, 2020. 3, 22

[2] Laurence Aitchison. A statistical theory of cold posteriors in deep neural networks. In *International Conference on Learning Representations*, 2020. 3, 5, 21, 23

[3] Gregor Bachmann, Lorenzo Noci, and Thomas Hofmann. How tempering fixes data augmentation in bayesian neural networks. *arXiv preprint arXiv:2205.13900*, 2022. 3

[4] Lucas Beyer, Olivier J Hénaff, Alexander Kolesnikov, Xiaohua Zhai, and Aäron van den Oord. Are we done with imagenet? *arXiv preprint arXiv:2006.07159*, 2020. 1

[5] Lukas Biewald. Experiment tracking with weights and biases, 2020. URL https://www.wandb.com/. Software available from wandb.com. 29

[6] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006. 5, 6

[7] Bertrand Charpentier, Daniel Zügner, and Stephan Günnemann. Posterior network: Uncertainty estimation without ood samples via density-based pseudo-counts. *Advances in Neural Information Processing Systems*, 33:1356–1367, 2020. 3

[8] Tianqi Chen, Emily Fox, and Carlos Guestrin. Stochastic gradient hamiltonian monte carlo. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1683–1691, Bejing, China, 22–24 Jun 2014. PMLR. 18

[9] Mark Collier, Basil Mustafa, Efi Kokiopoulou, Rodolphe Jenatton, and Jesse Berent. A simple probabilistic method for deep classification under input-dependent label noise. *arXiv preprint arXiv:2003.06778*, 2020. 3

[10] Mark Collier, Basil Mustafa, Efi Kokiopoulou, Rodolphe Jenatton, and Jesse Berent. Correlated input-dependent label noise in large-scale image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1551–1560, 2021. 3

[11] Erik Daxberger, Agustinus Kristiadi, Alexander Immer, Runa Eschenhagen, Matthias Bauer, and Philipp Hennig. Laplace redux-effortless bayesian deep learning. *Advances in Neural Information Processing Systems*, 34, 2021. 3

[12] Michael Dusenberry, Ghassen Jerfel, Yeming Wen, Yian Ma, Jasper Snoek, Katherine Heller, Balaji Lakshminarayanan, and Dustin Tran. Efficient and scalable bayesian neural nets with rank-1 factors. In *International conference on machine learning*, pages 2782–2792, 2020. 3

[13] Lei Feng, Senlin Shu, Zhuoyi Lin, Fengmao Lv, Li Li, and Bo An. Can cross entropy loss be robust to label noise? In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 2206–2212, 2021. 3

[14] Vincent Fortuin, Adrià Garriga-Alonso, Florian Wenzel, Gunnar Rätsch, Richard Turner, Mark van der Wilk, and Laurence Aitchison. Bayesian neural network priors revisited. *arXiv preprint arXiv:2102.06571*, 2021. 2, 3, 5, 7, 17, 22, 23

[15] Jacob R Gardner, Geoff Pleiss, David Bindel, Kilian Q Weinberger, and Andrew Gordon Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. *Advances in Neural Information Processing Systems*, 2018:7576–7586, 2018. 29

[16] Joyee Ghosh, Yingbo Li, and Robin Mitra. On the use of cauchy prior distributions for bayesian logistic regression. *Bayesian Analysis*, 13(2):359–383, 2018. 23

[17] Peter Grünwald. The safe bayesian. In *International Conference on Algorithmic Learning Theory*, pages 169–183. Springer, 2012. 4

[18] Peter Grünwald and Thijs Van Ommen. Inconsistency of bayesian inference for misspecified linear models, and a proposal for repairing it. *Bayesian Analysis*, 12(4):1069–1103, 2017. 21

[19] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2017. 6, 20

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016. 9, 29

[21] Jonathan Heek and Nal Kalchbrenner. Bayesian inference for large scale image classification. *arXiv preprint arXiv:1908.03491*, 2019. 22

[22] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019. 10, 27

[23] Dan Hendrycks, Norman Mu, Ekin D Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. *arXiv preprint arXiv:1912.02781*, 2019. 10

[24] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 20

[25] Geoffrey E Hinton and Drew Van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the sixth annual conference on Computational learning theory*, pages 5–13, 1993. 2

[26] Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 2013. 8

[27] Stephen C. Hora. Aleatory and epistemic uncertainty in probability elicitation with an example from hazardous waste management. *Reliability Engineering & System Safety*, 54:217–223, 1996. 1

[28] W Ronny Huang, Zeyad Emam, Micah Goldblum, Liam Fowl, Justin K Terry, Furong Huang, and Tom Goldstein. Understanding generalization through visualizations. 2020. 20

[29] Alexander Immer, Matthias Bauer, Vincent Fortuin, Gunnar Rätsch, and Khan Mohammad Emtiyaz. Scalable marginal likelihood estimation for model selection in deep learning. In *International Conference on Machine Learning*, pages 4563–4573. PMLR, 2021. 25

[30] Pavel Izmailov, Patrick Nicholson, Sanae Lotfi, and Andrew Gordon Wilson. Dangers of bayesian model averaging under covariate shift. *arXiv preprint arXiv:2106.11905*, 2021. 6, 17, 18, 23

[31] Pavel Izmailov, Sharad Vikram, Matthew D Hoffman, and Andrew Gordon Wilson. What are bayesian neural network posteriors really like? *arXiv preprint arXiv:2104.14421*, 2021. 2, 3, 7, 16, 17, 20, 24, 30

[32] Taejong Joo, Uijung Chung, and Minji Seo. Being bayesian about categorical probability. In *ICML*, 2020. 3

[33] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *NIPS*, 2017. 1, 16

[34] Anna-Kathrin Kopetzki, Bertrand Charpentier, Daniel Zügner, Sandhya Giri, and Stephan Günnemann. Evaluating robustness of predictive uncertainty estimation: Are dirichlet-based models reliable? In *International Conference on Machine Learning*, pages 5707–5718. PMLR, 2021. 3

[35] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009. 9, 14

[36] Ya Le and Xuan S. Yang. Tiny imagenet visual recognition challenge. 2015. 9, 14

[37] David J. C. MacKay. A practical bayesian framework for backpropagation networks. *Neural Computation*, 4:448–472, 1992. 2

[38] Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gordon Wilson. A simple baseline for bayesian uncertainty in deep learning. *Advances in Neural Information Processing Systems*, 32, 2019. 3

[39] Wesley J Maddox, Gregory Benton, and Andrew Gordon Wilson. Rethinking parameter counting in deep models: Effective dimensionality revisited. *arXiv preprint arXiv:2003.02139*, 2020. 19, 29

[40] Andrey Malinin and Mark Gales. Reverse kl-divergence training of prior networks: Improved uncertainty and adversarial robustness. *Advances in Neural Information Processing Systems*, 32, 2019. 3

[41] Andrey Malinin and Mark John Francis Gales. Predictive uncertainty estimation via prior networks. In *NeurIPS*, 2018. 1, 3

[42] Andrey Malinin, Bruno Mlodozeniec, and Mark Gales. Ensemble distribution distillation. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BygSP6Vtvr. 3

[43] Dimitrios Milios, Raffaello Camoriano, Pietro Michiardi, Lorenzo Rosasco, and Maurizio Filippone. Dirichlet-based gaussian processes for large-scale calibrated classification. In *NeurIPS*, 2018. 3, 6, 7, 16, 18

[44] Kevin P. Murphy. *Probabilistic Machine Learning: Advanced Topics*. MIT Press, 2023. URL probml.ai. 4

[45] Seth Nabarro, Stoil Ganev, Adrià Garriga-Alonso, Vincent Fortuin, Mark van der Wilk, and Laurence Aitchison. Data augmentation in bayesian neural networks and the cold posterior effect. *arXiv preprint arXiv:2106.05586*, 2021. 3, 7, 8, 17

[46] Jay Nandy, Wynne Hsu, and Mong Li Lee. Towards maximizing the representation gap between in-domain & out-of-distribution examples. *Advances in Neural Information Processing Systems*, 33:9239–9250, 2020. 3

[47] Radford M. Neal. Bayesian learning for neural networks. 1995. 2

[48] Lorenzo Noci, Kevin Roth, Gregor Bachmann, Sebastian Nowozin, and Thomas Hofmann. Disentangling the roles of curation, data-augmentation and the prior in the cold posterior effect. *ArXiv*, abs/2106.06596, 2021. 3, 5, 17

[49] Kazuki Osawa, Siddharth Swaroop, Mohammad Emtiyaz E Khan, Anirudh Jain, Runa Eschenhagen, Richard E Turner, and Rio Yokota. Practical deep learning with bayesian principles. *Advances in neural information processing systems*, 32, 2019. 3

[50] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, 2019. 29

[51] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011. 29

[52] John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999. 20

[53] Geoff Pleiss, Tianyi Zhang, Ethan Elenberg, and Kilian Q Weinberger. Identifying mislabeled data using the area under the margin ranking. *Advances in Neural Information Processing Systems*, 33:17044–17056, 2020. 5

[54] Carl Edward Rasmussen and Christopher K Williams. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006. 1, 4

[55] Simo Särkkä and A. Solin. Applied stochastic differential equations. 2019. 18

[56] Robin Senge, Stefan Bösner, Krzysztof Dembczynski, Jörg Haasenritter, Oliver Hirsch, Norbert Donner-Banzhoff, and Eyke Hüllermeier. Reliable classification: Learning classifiers that distinguish aleatoric and epistemic uncertainty. *Inf. Sci.*, 255:16–29, 2014. 1

[57] Murat Sensoy, Lance Kaplan, and Melih Kandemir. Evidential deep learning to quantify classification uncertainty. *Advances in neural information processing systems*, 31, 2018. 3

[58] Murat Sensoy, Lance Kaplan, Federico Cerutti, and Maryam Saleki. Uncertainty-aware deep classifiers using generative models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5620–5627, 2020. 3

[59] Weishi Shi, Xujiang Zhao, Feng Chen, and Qi Yu. Multifaceted uncertainty estimation for label-efficient deep learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 17247–17257. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/c80d9ba4852b67046bee487bcd9802c0-Paper.pdf. 3

[60] 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. Citeseer, 2011. 4, 9, 18

[61] Florian Wenzel, Kevin Roth, Bastiaan Veeling, Jakub Swiatkowski, Linh Tran, Stephan Mandt, Jasper Snoek, Tim Salimans, Rodolphe Jenatton, and Sebastian Nowozin. How good is the bayes posterior in deep neural networks really? In *International Conference on Machine Learning*, pages 10248–10259. PMLR, 2020. 2, 3, 4, 5, 6, 7, 9, 17, 20, 22, 24, 25

[62] Andrew Gordon Wilson and Pavel Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. *arXiv preprint arXiv:2002.08791*, 2020. 1, 3, 4, 5, 17, 24

[63] Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P Xing. Deep kernel learning. In *Artificial intelligence and statistics*, 2016. 3

[64] Chen Zeno, Itay Golan, Ari Pakman, and Daniel Soudry. Why cold posteriors? on the suboptimal generalization of optimal bayes estimates. In *Third Symposium on Advances in Approximate Bayesian Inference*, 2020. 3, 7, 20

[65] Ruqi Zhang, Chunyuan Li, Jianyi Zhang, Changyou Chen, and Andrew Gordon Wilson. Cyclical stochastic gradient MCMC for bayesian deep learning. *arXiv preprint arXiv:1902.03932*, 2019. 3, 9, 18, 22

[66] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, 31, 2018. 3

[67] Xujiang Zhao, Feng Chen, Shu Hu, and Jin-Hee Cho. Uncertainty aware semi-supervised learning on graph data. *Advances in Neural Information Processing Systems*, 33:12827–12836, 2020. 3

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] Appendix A.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] Appendix A.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Eq. (10), Appendices G and I.
   (b) Did you include complete proofs of all theoretical results? [Yes] See Eq. (10), Appendices G and I
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Unless otherwise stated, mean and standard deviation over 3 trials.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Appendix L.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] CIFAR-10 [35], TinyImagenet [36].
   (b) Did you mention the license of the assets? [Yes] CIFAR-10, MIT License, TinyImagenet, MIT License [36]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Appendix for On Uncertainty, Tempering, and Data Augmentation in Bayesian Classification

The Appendix is structured as follows:

- In Appendix A, we discuss broader impacts and limitations of our work.
- In Appendix B, we provide more context for our key observations in the main text.
- In Appendix C, we give more description of stochastic gradient langevin dynamics (SGLD) which we use for our inference procedure in this work.
- In Appendix D, we describe the gaussian approximation to the noisy dirichlet model of Milios et al. [43].
- In Appendix E, we describe another synthetic experiment relating data augmentation, inference, and posterior tempering.
- In Appendix F, we describe how to sharpen the softmax likelihood with logit smoothing, which is our smoothed softmax baseline.
- In Appendix G, we further connect cold and tempered likelihood posteriors.
- In Appendix H, we visualize the noisy dirichlet likelihood for observed and unobserved classes.
- In Appendix I, we extend the interplay of SGHMC with data augmentation (from Section 5) to variational inference.
- In Appendix J, we give preliminary description to a proper likelihood accounting for data augmentation.
- In Appendix K, we describe further experimental results on synthetic problems and the interplay of cold and tempered posteriors.
- In Appendix L, we describe experimental details for all of our experiments.

A Limitations and Broader Impacts

Broader Impacts

Our work is primarily theoretical and not applied in nature so we do not see any specific negative or societal effects from our work. However, Bayesian deep learning has been applied to both medical diagnoses and autonomous driving [33], while our work is intended to make these types of modelling more robust and reliable by making the inferred probabilities more accurate.

Limitations

We see the following limitations for our work:

- As the cold posterior effect has primarily been demonstrated on image datasets, our work almost exclusively focuses on residual architectures on noiseless image datasets.
- Our work primarily focuses on SGHMC (although we show similar results for SVI) under data augmentation due to the computational expense and difficulties of running HMC with large models under data augmentation [31].
- Our work solely focuses on understanding the interactions between data augmentation and posterior tempering, but does not provide guarantees of robustness to label noise or fixes to other modelling issues.

B Our Observations in the Context of Prior Work

Noci et al. [48]: Data augmentation is sufficient but not necessary for CPE. Noci et al. [48] confirm the findings in Izmailov et al. [31] and Fortuin et al. [14] that data augmentation plays a significant role in the cold posterior effect. In particular, Izmailov et al. [31] show that removing data augmentation from the models in Wenzel et al. [61] alleviates the cold posterior effect in all of their examples. However, Noci et al. [48] also show that it is sometimes possible to achieve CPE without data augmentation, for example by subsampling the dataset. Similarly, Wilson and Izmailov [62] note that many types of misspecification could lead to a cold posterior effect.

In our work, we provide a more nuanced perspective on the cold posterior effect and data augmentation: data augmentation leads to a poor representation of aleatoric uncertainty, which can be addressed by tempering the posterior. We note that BNNs can misrepresent aleatoric uncertainty even without data augmentation, and can have a reasonable representation of aleatoric uncertainty in the presence of data augmentation. However, in the vast majority of practical scenarios where the cold posterior effect has been demonstrated, it appears due to the use of data augmentation. Thus data augmentation is *largely* responsible for the cold posterior effect, and we have shown here that this is precisely because augmentation changes our representation of aleatoric uncertainty to be underconfident, and tempering can correct for this underconfidence, more honestly representing our beliefs.

Finally, Izmailov et al. [30] show that BNNs have issues under covariate shift, due to the posterior not contracting sufficiently along some directions in the parameter space. The same issue occurs when BNNs are applied to extremely small datasets, which may affect the results on data subsampling presented in Noci et al. [48].

Nabarro et al. [45]: the lack of a CPE without DA is an artifact of using the wrong model (i.e. without DA). Similar to our discussion in Section 5, Nabarro et al. [45] point out that tempering does not always provide a principled approach to data augmentation. They instead devise a new observation model, where the model outputs are averaged over all the augmented datapoints. With this observation model, they still observe the cold posterior effect. These results are interesting in that they show that a valid likelihood is not sufficient to remove the cold posterior effect. However, these results do not rule out the possibility that the proposed likelihood is a poor description of the data generation process: indeed, in reality the labels for the training image are not produced by considering all the possible augmentations of these images.

Generally, our results suggest that the cold posterior effect is caused by a poor representation of the aleatoric uncertainty in the BNNs. It can happen with or without data augmentation, with valid or invalid likelihoods. A valid likelihood is not necessarily a well-specified likelihood. However, we believe that the interpretation that the lack of CPE in models without data augmentation is an artifact of misspecification is highly questionable: when our beliefs about the aleatoric uncertainty in the data are correctly captured by the model, we should not need posterior tempering to achieve strong results; it is the need for tempering which is an indication of misspecification.

Noci et al. [48]: Cold posteriors are unlikely to arise from a single simple cause. Noci et al. [48] show examples where the cold posterior effect arises in isolation from data augmentation, data curation, or prior misspecification. Consequently, they argue that no one cause can fully explain the cold posterior effect. While we agree with this general argument, which is also aligned with the discussion in Wilson and Izmailov [62], we note that all of the considered causes are directly related to aleatoric uncertainty. Indeed, in this work we have shown that data augmentation significantly affects the level of aleatoric uncertainty assumed by the model. The data curation is also directly connected to aleatoric uncertainty, as curated datasets are expected to have low label noise. Finally, in BNNs the prior over the parameters specifies the assumptions about aleatoric uncertainty, and tempering can be used to correct for the effect of the prior. We believe that our results in this paper provide a compelling explanation for the observations in Noci et al. [48].

Summary. Overall, properly representing aleatoric uncertainty is a challenging but fundamentally important consideration in Bayesian classification. We have shown that posterior tempering provides a mechanism to more honestly represent our beliefs about aleatoric uncertainty, especially in the presence of data augmentation. In general, as in Wilson and Izmailov [62], we should not be alarmed if $T = 1$ is not optimal in sophisticated models on complex real-world datasets. Moreover, we have shown how other mechanisms to represent aleatoric uncertainty, such as the noisy Dirichlet model,

do not suffer from a cold posterior effect in the presence of data augmentation. Indeed, while an interesting phenomenon, cold posteriors should not be conflated with the success or failure of Bayesian deep learning. In general, approximate inference in Bayesian neural networks has been making great strides forward, often providing better performance at a comparable computational cost to standard methods. However, there *are* practical challenges to the adoption of Bayesian deep learning. For example, Izmailov et al. [30] shows that Bayesian neural networks can profoundly degrade in performance under a wide range of relatively minor distribution shifts — behaviour which could affect applicability on virtually any real-world problem, since train and test rarely come from exactly the same distribution. While their *EmpCov* prior provides a partial remedy, there is still much work to be done. For example one could develop priors that protect against covariate shift by accounting for linear dependencies in the internal representations of the network.

C Stochastic Gradient Langevin Dynamics (SGLD).

For large-scale neural networks, exact posterior inference remains intractable. To approximate samples from the posterior $p(w \mid \mathcal{D})$ over parameters of a neural network, we simulate the time-discretized Langevin stochastic differential equation (SDE) [55, 60]

$$\begin{aligned} \Delta w &= \mathbf{M}^{-1}\mathbf{m}\epsilon, \\ \Delta \mathbf{m} &= -\nabla_w \widetilde{U}(w)\epsilon - \gamma \mathbf{m}\epsilon + \sqrt{2\gamma T}\eta, \text{ where } \eta \sim \mathcal{N}\left(\mathbf{0}, \mathbf{M}\right), \end{aligned} \tag{12}$$

where $\mathbf{m}$ are the auxiliary momentum variables, $\mathbf{M}$ is the mass matrix which acts as a preconditioner (often identity), γ is the friction parameter, T is the temperature, $\Delta t = \epsilon$ is the time discretization (step size), and $\nabla_w \widetilde{U}(w)$ is an unbiased estimator of the gradient $\nabla_w U(w)$ using only a subset of the dataset $\mathcal{D}$ for computational efficiency. For $\epsilon \to 0$ in the limit of time $t \to \infty$, simulating Eq. (12) produces a trajectory distributed according to the stationary distribution $\exp\{-U(w)/T\}$, which is exactly the posterior

$$p_{\text{cold}}(w \mid \mathcal{D}) \propto \exp\Bigg\{ -\frac{1}{T} \underbrace{\left(-\log p(\mathcal{D} \mid w) - \log p(w)\right)}_{U(w)} \Bigg\}, \tag{13}$$

we desire in Section 3. When $\gamma = 0$, Eq. (12) describes the Stochastic Gradient Langevin Dynamics (SGLD) [60], and otherwise it describes the Stochastic Gradient Hamiltonian Monte Carlo (SGHMC) where $1 - \gamma$ represents the momentum [8]. Furthermore, we can sample from Eq. (3) by setting $T = 1$ in Eq. (12) and raising only the likelihood to a power T. In addition, it is often beneficial to use a cyclical time-stepping schedule for ϵ in Eq. (12), as proposed by Zhang et al. [65] for cyclical-SGLD (cSGLD) when $\gamma = 0$, and cyclical-SGHMC (cSGHMC) when $\gamma > 0$.

D Gaussian Approximation to the noisy Dirichlet Model.

Milios et al. [43] considered the distribution in Eq. (9) in the context of Gaussian process classification, but with a different goal: they aimed to create a regression likelihood which would approximate the softmax likelihood. They further approximated the Dirichlet distribution Dir.$(f(x))$ over the class probabilities $f(x)$ with a product of independent Gaussian distributions over the logits $z(x)$:

$$\begin{aligned} p_{NDG}(w \mid D) &\propto p(w) \prod_{x,y \in \mathcal{D}} \prod_{c=1}^{C} \mathcal{N}(z_c(x) \mid \mu_c, \sigma_c^2), \text{ with} \\ \alpha_c &= 1 + \alpha_\epsilon \cdot I[c = y], \;\; \sigma_c^2 = \log(1/\alpha_c + 1), \;\; \mu_c = \log(\alpha_c) - \frac{\sigma_c^2}{2}, \end{aligned} \tag{14}$$

where $I[c = y]$ is the indicator function equal to 1 for $c = y$ and 0 for $c \neq y$. Here *NDG* stands for *noisy Dirichlet Gaussian* approximation. While theoretically we do not need to use the Gaussian approximation in Eq. (14) for Bayesian neural networks and can directly use Eq. (9), we found that in practice the approximation is much more stable numerically. Indeed, the approximation in Eq. (14) amounts to solving a regression problem in the space of logits $z(x)$ with a Gaussian observation model. In the experiments in this paper we will use the p_{NDG} model.

Visual comparison. In Figure 6 (left), we show the CDF of the posterior distribution over the confidence f_y for the correct class y with the standard softmax likelihood, tempered likelihood and

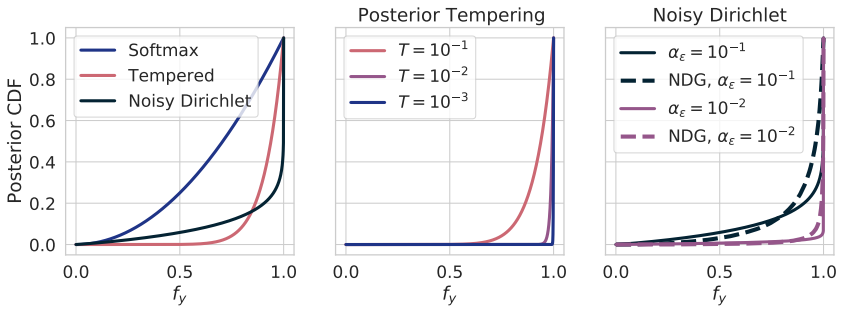
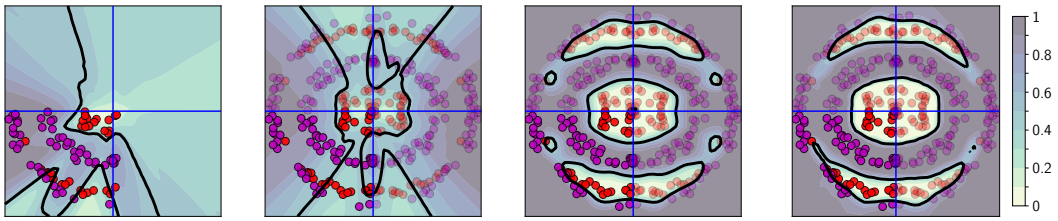


Figure 6: **Comparison of Bayesian classification models.** The probability CDF of the posterior over the confidence f_y for the correct class y with the standard cross entropy likelihood, tempered likelihood and the noisy Dirichlet model in a binary classification problem. **Left**: Tempering and noisy Dirichlet models both allow to concentrate the posterior on solutions that are confident in the correct label. **Middle**: Lower likelihood temperatures lead to higher concentration on confident solutions. **Right**: In the noisy Dirichlet model, lower values of the noise parameter α_ϵ also lead to higher confidence.



(a) Softmax Lik. (SL) (b) SL+Data Aug.(DA) (c) SL+DA+Tempering (d) Noisy Dirichlet+DA

Figure 7: **Insufficient Sharpness of Softmax Likelihood.** Decision boundary of BNN classifiers on a synthetic problem. In each panel, the decision boundary is shown with a black line, datapoints are shown with magenta (class 0) and red (class 1) circles. We consider data augmentation – random flip about the axes shown with blue lines; the augmented datapoints are shown semi-transparently. We use HMC to approximate the posterior in each of the panels. **(a)** A standard softmax likelihood provides a reasonable but imperfect fit of the training data. **(b)** Adding data augmentation leads to a more diffuse likelihood which is underconfident, and therefore an even worse fit. **(c)** Tempering with $T = 0.1$ sharpens the softmax likelihood leading to a perfect fit on the train data. **(d)** A noisy Dirichlet model, provides a similar fit without the need for tempering.

the noisy Dirichlet model for a binary classification problem. Here, we assume that the prior over $\{f(x)\}_{x\in\mathcal{D}}$ is uniform and independent across x for the purpose of visualization. Both tempering and the noisy Dirichlet model indeed push the predictions f_y to be much more confident in the correct class y compared to the standard softmax likelihood. In Figure 6 (middle) we show the effect of the value of temperature: lower temperatures correspond to more confident predictions. Similarly, in Figure 6 (right) lower values of α_ϵ lead to more confident predictions in the noisy Dirichlet model. The Gaussian approximation NDG matches the CDF of the noisy Dirichlet model well, but smooths it slightly, making it more amenable to numerical sampling.

E Synthetic Experiment

We generate the data from a modified version of the two spirals dataset [see e.g. 39], where we restrict the data to the $x_1 < 0, x_2 < 0$ quadrant. The exact code used to generate the data is available in the Appendix L.1.

In Figure 7(a) we visualize the data and the decision boundary of a Bayesian neural network. For the Bayesian neural network, we use an iid Gaussian prior $\mathcal{N}(0, 0.3^2)$ over the parameters of the

model; we use full batch Hamiltonian Monte Carlo (HMC) to sample from the posterior, following Izmailov et al. [31]. With the chosen prior, the model is not able to fit the data perfectly, but provides a reasonable fit of the training data.

In Figure 7(b) we consider the effect of data augmentation. Specifically, we apply random flips about the x_1 and x_2 axes. In the figure, the augmented datapoints are shown semi-transparently. To evaluate the network under data augmentation, we run HMC to sample from the posterior distribution in Eq. (11). The Bayesian neural network provides a lower quality fit to the data compared to the fit without data augmentation shown in Figure 7(a). Indeed, according to our analysis in Section 5, the observation model is softened by the data augmentation, leading the model to fit the training data poorly.

According to Section 4, we can sharpen the model leading to a much better fit on the train data by using the tempered likelihood as in Figure 7(c) or the noisy Dirichlet observation model as in Figure 7(d). With both of these approaches we achieve a near-perfect fit on the training data, which is desirable if we assume low aleatoric unceratinty. For further details on this experiment, please see Appendix L.1.

To intuitively grasp the impact of aleatoric uncertainty on inference, we first develop a toy illustration. Consider the *two spirals* binary classification problem [28] where, for training we generate 50 samples with both $x_1, x_2 < 0$. Figure 7 visualizes the posterior predictive density for various settings.

By naively learning from the training data using the softmax likelihood, we find that the predictive density is poorly calibrated to the effect that it does not learn anything meaningful except in the region containing the training data, as visualized in Figure 7a. But, by virtue of our a priori knowledge about the mirror symmetry across both axes, we can create augmented data, i.e. for every (x, y) pair in the training data, we create augmentations $(\pm x, \pm y)$ for use in training. With this modification, we find that the predictive density surface in Figure 7b appears to improve but remains very diffuse, implying underconfidence.

But in fact, we can fix the predictive density surface (and hence the performance) by accounting for aleatoric uncertainty. In Figure 7c we see that tempering allows us to sharpen the softmax likelihood such that the final predictive density does not remain diffuse. We are implicitly correcting for aleatoric uncertainty which was artificially inflated due to data augmentations. Alternatively, by using a noisy Dirichlet likelihood which allows explicit control over the amount of aleatoric uncertainty in our data observation model, we are able to achieve a similar effect as shown in Figure 7d.

This toy illustration is representative of how modern machine learning models for classification are trained in practice, and provide a first direct understanding of the interaction of aleatoric uncertainty with tempering and data augmentation.

F Sharpening with the Smoothed Softmax Likelihood

A natural approach to consider sharpening the softmax likelihood is by smoothing the logits. The *tempered softmax likelihood* for a given class observation $y = c$ conditional on input x is given by,

$$\frac{1}{T} \log p(y = c \mid x) = \log \frac{\exp\{f_c(x; w)/T\}}{(\sum_{j=1}^{C} \exp\{f_j(x; w)\})^{1/T}} \tag{15}$$

where $f_j(x; w) \in \mathbb{R}$ represents the logit for the jth class of a total of C classes. A properly normalized distribution, however, leads to what we term the *smoothed softmax likelihood*,

$$\log p_T(y = c \mid x) = \log \frac{\exp\{f_c(x; w)/T\}}{\sum_{j=1}^{C} \exp\{f_j(x; w)/T\}}. \tag{16}$$

Versions of this likelihood have appeared in prior literature, especially when considering distillation [24] and temperature (e.g. Platt) scaling [52, 19]. More recently, Zeno et al. [64] have pointed out a connection between Eq. (16) and the cold posteriors Eq. (3) studied by Wenzel et al. [61].

Insufficiency of smoothed softmax. Consider a toy coinflip example — we are interested in inferring the probability of heads $w \in [0, 1]$ using a single observation of a heads $\mathcal{D} = \{H\}$.

The posterior over w, under the assumption of a uniform prior $p(w) = \mathrm{U}[0, 1]$ is given by $p(w \mid \mathcal{D}) \propto p(H \mid w) = \theta$. The smoothed softmax likelihood corresponds to rescaling the logit $\ell(w) = \log\{w/(1-w)\}$ as $p_T(H \mid w) = \sigma(\ell(w)/T)$, where $\sigma(z) = 1/(1 + \exp\{-z\})$ is the sigmoid function [6].

Using the tempered softmax likelihood by raising to a power $1/T$ where $T < 1$, we effectively sharpen the density function and force the posterior to concentrate more sharply around 1 owing to the single observation. On the other hand, using a smoothed softmax likelihood ends up spreading the posterior mass very differently, to the effect that it undermines the evidence from observation. This behavior is visualized in Figure 8. In fact, no matter what temperature we use, the smoothed softmax likelihood cannot be made arbitrarily sharp.

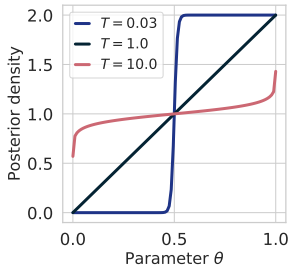


Figure 8: A smoothed likelihood Eq. (16) is insufficient to achieve arbitrary sharpness (Appendix F). An arbitrarily sharp likelihood would concentrate around a single value of θ rather than being effectively uniform over ranges.

In Figure 10, we empirically verify with ResNet-18 on CIFAR-10 that no matter the value of temperature T, a smoothed softmax likelihood does not outperform a tempered likelihood. Moreover, it does not mitigate the cold posterior effect. Therefore, we can return to our discussion of cold and tempered posteriors.

Smoothed softmax and reparameterization. Intuitively, for Bayesian neural networks, the smoothed softmax likelihood corresponds to a simple reparameterization of the standard softmax likelihood BNN, where the last layer parameters are all divided by a scalar constant T. This model is equivalent to simply rescaling the prior over the last layer parameters. In particular, for BNNs with zero-mean iid Gaussian priors, this reparameterization is equivalent to dividing the standard deviation of the prior over the last layer parameters by T. Such rescaling has a very limited effect on the properties of the model, especially given the homogeneity of the ReLU activations and the scale invariance due to normalization layers, which may provide further intuition for why the smoothed softmax likelihood is insufficient for modeling aleatoric uncertainty in Bayesian classification.

G Connecting Cold Posteriors and Tempered Likelihood Posteriors

In this appendix, we explicitly spell out the connections between cold, tempered, and Bayesian posteriors, focusing first on the shape of the posterior with Bayesian linear models before moving into describing how the prior changes with a cold posterior.

G.1 Connections in Bayesian Linear Models

We begin by taking a look at the connection between the Bayesian, cold, and tempered posteriors under linear models. For a dataset $\mathcal{D} = \{X, y\}$ consisting of d-dimensional inputs $X = [x_1; x_2; \ldots; x_N] \in \mathbb{R}^{N \times d}$, corresponding outputs $y \in \mathbb{R}^N$, a known observation noise variance σ^2, and prior over parameters $w \sim \mathcal{N}(0, \alpha \mathbf{I})$, the posterior over parameters is given by,

$$p(w \mid \mathcal{D}) = \mathcal{N}(\left(X^\top X + \alpha^{-1}\sigma^2 \mathbf{I}\right)^{-1} X^\top y, \left(\sigma^{-2} X^\top X + \alpha^{-1}\mathbf{I}\right)^{-1}). \tag{17}$$

Following Grünwald and Van Ommen [18],who study $T > 1$ in the context of model misspecfication, the tempered posterior for an explicit temperature T is,

$$p_{\text{temp}}(w \mid \mathcal{D}) = \mathcal{N}(\left(\tfrac{1}{T} X^\top X + \alpha^{-1} T \sigma^2 \mathbf{I}\right)^{-1} X^\top y, \left(\tfrac{1}{T}\sigma^{-2} X^\top X + \alpha^{-1}\mathbf{I}\right)^{-1}). \tag{18}$$

Further, for a *cold posterior*, where both the likelihood and the prior are raised to a temperature, we can follow Aitchison [2] and rewrite the prior over w as $\mathcal{N}(0, \alpha T \mathbf{I})$, giving the posterior as,

$$p_{\text{cold}}(w \mid \mathcal{D}) = \mathcal{N}(\left(X^\top X + \alpha^{-1}\sigma^2 \mathbf{I}\right)^{-1} X^\top y, T\left(\sigma^{-2} X^\top X + \alpha^{-1}\mathbf{I}\right)^{-1}). \tag{19}$$

Thus, "cold" posteriors in this setting reproduce the same posterior mean as the standard Bayesian posterior, only varying the value of the posterior covariance matrix. We visually demonstrate this effect for the posterior confidence ellipses over θ in Figure 9 left.

[6] Note here that $\sigma(\ell(\theta)) = \theta$

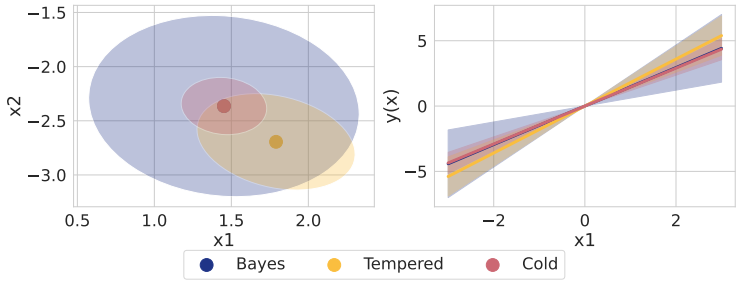


Figure 9: **Left:** Posterior confidence ellipses for Bayesian linear regression in both Bayesian, tempered, and cold frameworks. **Right:** Posterior functions for all three approaches. The cold posterior simply rescales the prior, producing more confident fits.

For $T < 1$, we tend to increase the concentration of the posterior around the posterior mean. In contrast, for $T > 1$, we tend to decrease the concentration of the posterior around the posterior mean. Propagating the posterior distribution on w to the resulting distribution over the function $y_\star = w^\top x_\star$ for novel inputs $x_\star$, the cold posterior over $y_\star$ has the same mean as the Bayesian posterior, while only the posterior covariance is rescaled by a factor of T. We also demonstrate this effect in Figure 9 right, showing that the slope of the cold model is the same as the slope of the fully Bayesian model, only their posterior variances are different. However, the slope of the tempered model is different, owing to an effectively increased noise term.

A similar finding is produced by Adlam et al. [1] in the context of Gaussian process regression, where similarly cold posteriors simply rescale the posterior variance of the GP posterior, while tempering the posterior tends to increase the noise level.

G.2 Cold Posteriors and Sharp Likelihoods

While constructing a cold posterior Eq. (2) involves tempering both the likelihood and the prior, the cold posterior effect still remains even with a tempered likelihood posterior Eq. (3) alone. Many classes of prior distributions, specifically any prior that is bounded almost everywhere, continue to be proper prior distributions when raised to a power $1/T$. We formalize this statement in Theorem 1.

Theorem 1. *For proper prior distributions, $p(w)$, that have bounded density functions, e.g. $p(w) \leq M$ almost everywhere, then for $T \leq 1$, $p(w)^{1/T}/\mathcal{I}$ is a proper prior distribution, where $\mathcal{I} = \int p(\theta)^{1/T} d\theta < \infty$.*

Proof. As $p(\theta)$ is bounded above by M, then $p(\theta)/M \leq 1$ for all x. As $1/T \geq 1$, then $(p(\theta)/M)^{1/T} \leq p(\theta)/M \leq 1$. By construction $p(\theta)/M$ is integrable as $p(w)$ is integrable, so then $(p(w)/M)^{1/T}$ must also be integrable implying that $p(w)^{1/T}$ is integrable. To make $p(w)^{1/T}$ a distribution, we only need to construct a normalization constant $\mathcal{I}$. ■

Theorem 1 shows that the cold posterior effect in the Bayesian deep learning literature can be equivalently thought of in terms of a *tempered likelihood* posterior alone. The only difference is that the actual priors being used are different from the stated priors in that they have tighter variances. See Appendix G for further examples and discussion.

The above result unifies several tempered likelihood findings [65, 21] to those with the cold posterior studies [61, 14]. Using a ResNet-18 on CIFAR-10, we empirically demonstrate such an equivalence in Figure 10. A rescaled prior and step size ϵ can match the performance of a cold posterior. To match the gradient scaling, we must downscale the step size by the square root of the

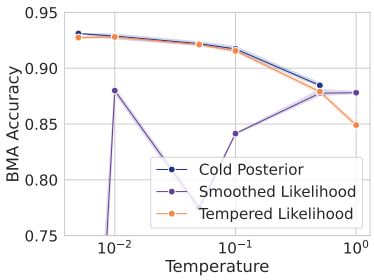


Figure 10: ResNet-18 on CIFAR-10. Cold posteriors are matched by tempered posteriors, but not by the smoothed likelihood.

implied temperature, while we match the prior variance by setting it to ϵT. We give specific examples of these priors in the next section.

Thus, to understand the cold posterior effect, we may focus on the tempered likelihood term in Eq. (3). Tempering the likelihood with $T < 1$ is sharpening the density function to facilitate a more accurate observation model per our beliefs. It is then only natural to use a likelihood which allows direct control of the sharpness, which we introduce next. We also note that the likelihood and the prior tend to interact here. For a fixed prior, a sharper likelihood will tend to produce a sharper posterior, while decreasing the variance on the prior will additionally tend to produce a sharper (more confident) posterior. This is exactly the effect produced by likelihood tempering with $T < 1$; tempering the likelihood produces a sharper likelihood which therefore produces a sharper posterior.

More Cold Priors

Theorem 1 covers many common prior choices in the Bayesian deep learning literature. For *Gaussian and Laplace priors*, the scaling term (variance in the case of the Gaussian prior and scale in the case of the Laplace prior) changes by a factor of ϵT where ϵ is the original scale. It also holds for correlated Gaussian priors such as those studied by Fortuin et al. [14], Izmailov et al. [30]. This effect was previously noted for Gaussian priors by Aitchison [2].

For *Student's t priors*, we show below that the degrees of freedom increases and the scale term changes producing a distribution with smaller variance than the original distribution. Similarly for *Cauchy priors*, the entire form of the distribution changes and in such a way that it now has finite mean and variance, which it did not before.

The log pdf for a non-central student T distribution[7] is

$$\log p(x) = \log \Gamma(v + 1/2) - \log \Gamma(v/2) - \log \sqrt{v\pi}\sigma - \frac{v+1}{2} \log \left(1 + \frac{x^2}{v\sigma^2}\right)$$

Rescaling by a temperature T gives

$$\frac{1}{T} \log p(x) = \frac{1}{T} \left(\log \Gamma(v + 1/2) - \log \Gamma(v/2) - \log \sqrt{v\pi}\sigma\right) - \frac{v+1}{2T} \log \left(1 + \frac{x^2}{v\sigma^2}\right)$$

and letting $\tilde{v} = \frac{v+1}{T} - 1$ produces

$$\frac{1}{T} \log p(x) = \text{const.} - \frac{\tilde{v}+1}{2} \log \left(1 + \frac{\tilde{v}}{\tilde{v}} \frac{x^2}{v\sigma^2}\right) = \text{const.} - \frac{\tilde{v}+1}{2} \log \left(1 + \frac{x^2}{\tilde{v}\tilde{\sigma}^2}\right)$$

with $\tilde{\sigma}^2 = v\sigma^2/\tilde{v} = \frac{v\sigma^2 T}{v+1-T}$.

For $v > 2$, the original variance was $\sigma^2 \frac{v}{v-2}$ and now it is $\sigma^2 \frac{T}{v+1-3T}$, which is less whenever $v > 2$ and so the variance exists. Note additionally that $\tilde{v} > v$ as well.

For a Cauchy distribution (or equivalently, follow the argument for the non-central t with $\eta = 1$), the tempered prior follows the form

$$p_T(x) \propto (1 + x^2)^{-1/T},$$

which produces a finite integral and therefore a proper prior for all $T \geq 1$. This class of distributions has finite means for any $T < 1$ and finite variances for $T < 2/3$, whereas the Cauchy distributions do not have finite means or variances. This change may have the effect of making posterior means exist for cold posteriors when they may not have existed before, as can be the case for logistic regression with Cauchy priors [16].

G.3 Discussion with the Cold Posterior Effect

These types of findings tend to suggest that "cold" priors have smaller variances (or spreads, if the variances do not exist) than their warm counterparts. This is most clearly the case for Gaussian and Laplace prior distributions where the prior variance is reduced from ϵ to ϵT. One natural question is if this finding is an explanation for the "cold" posterior effect, an analogue of prior-misspecification,

[7] https://en.wikipedia.org/wiki/Noncentral_t-distribution

as discussed by Wenzel et al. [61]. However, Wilson and Izmailov [62] found that standard Gaussian $\mathcal{N}(0, 1)$ priors tend to be reasonably well-calibrated and that varying priors tends to have minimal impact on down-stream accuracy and calibration. Izmailov et al. [31] verified this evidence with high-quality HMC samples, but without data augmentation.

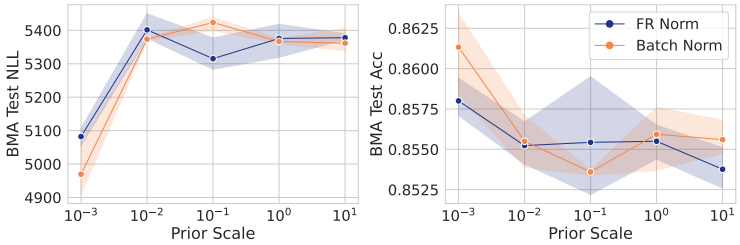


Figure 11: Priors of various scales for BNNs with temperature 1. No prior with the same step size is able to match the performance of the lower temperature.

In Figure 11, we show that no choice of prior alone with temperature 1 is able to reach the same accuracy as a tempered (or cold) baseline model. Thus, it is not merely (Gaussian) prior misspecification that causes cold posteriors to exist, and we need to look beyond this potential explanation.

H Visualizations of Noisy Dirichlet Likelihoods

In Figure 12, we show various visualization of the likelihood for the observed and unobserved classes for the noisy Dirichlet likelihood.

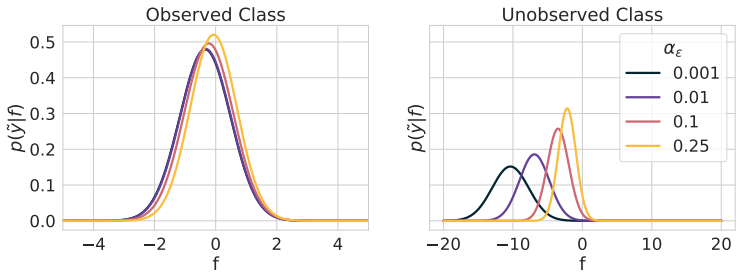


Figure 12: Dirichlet densities for various values of α_ϵ for both the observed class (**left**) and any unobserved classes (**right**). Higher values of α_ϵ tend to produce sharper likelihoods as the distributions corresponding to these values have lower variances. Here f is the equivalent to the logit, while $\tilde{y}$ is the rescaled observation.

I Targeting Distributions for Variational Inference

A naive attempt to incorporate data augmentation would be to simply stack all K copies of the data into the posterior and then to infer it; however, this approach would contract the posterior by too much as we would be considering NK data points rather than N data points. A second approach would be then to down-weight the augmentations so that we effectively see only N data points by raising the likelihood of each (augmented) data point to a power $1/K$, which produces exactly Eq. (11).

Variational inference with stochastic gradient descent using the evidence lower bound (ELBO) produces a similar gradient to stochastic gradient descent as

$$\nabla \text{ELBO}(\phi) = \frac{N}{m} \sum_{(x_i, y_i) \in \mathcal{D}_m} \mathbb{E}_{q_\phi(w)} \nabla_w \log p(y_i \mid t_j(x_i)) + \nabla_\phi \text{KL}(q_\phi(w) || p(w)), \tag{20}$$

switching the parameters of the variational distribution to ϕ and again with transformations t_j sampled uniformly from $\mathcal{T}$. The same two sources of randomness, sub-sampling and augmentation,

apply in this situation and computing the expected value of Eq. (20) produces a similar target as Eq. (10):

$$\mathbb{E}[\nabla \text{ELBO}(\phi)] = \sum_{i=1}^{N} \sum_{j=1}^{K} [\nabla_w \log p(y_i \mid t_j(x_i))^{1/K}] + \nabla_\phi \text{KL}(q_\phi(w) || p(w)), \tag{21}$$

implying that the full optimization problem VI is targeting uses a tempered likelihood in that term, e.g. $p(y_i \mid t_j(x_i))^{1/K}$. We note that variational inference for BNNs often does end up requiring tempering by downweighting the KL term in the ELBO, see Section 2.3 of Wenzel et al. [61] for several examples. While we do not consider Laplace approximations under data augmentation, they additionally tend to require some amount of likelihood down-weighting for high performance (see e.g. Immer et al. [29] for example), suggesting that they also tend to target a similar likelihood distribution.

J A Proper Likelihood for Data Augmentation

Treating data augmentations as independent samples from the true underlying data generating process is problematic, since the augmented samples are highly correlated with the originally observed samples. In Section 6.1, we see that stacking the augmented samples in to a single unified dataset is problematic in the sense that observation noise artificially inflated. For well-calibrated predictions, this is a mischaracterization of the data generating process.

To arrive at a proper likelihood accounting for data augmentations $t_j \in \mathcal{T}$, consider a likelihood which extends the usual observation likelihood with a consistency term,

$$p_{\text{aug}}(\mathcal{D} \mid w) = \prod_{i=1}^{N} p(y_i \mid x_i) \cdot \prod_{t_j \in \mathcal{T}} p\left(f(t_j(x_i); w)) \mid f(x_i; w)\right). \tag{22}$$

The augmentation likelihood p_{aug} properly accounts for aleatoric uncertainty when the augmentation is trivial, and not lead to underconfident predictions when the augmentations are uncorrelated. For a minibatch sample from the full dataset $\mathcal{D}_m \subset \mathcal{D}$ of size m, and a subset of k augmentations $\mathcal{T}_k \subset \mathcal{T}$, the valid unbiased stochastic gradient estimator for use with SGLD is given by,

$$\begin{aligned} \nabla \widetilde{U}_{\text{aug}}(w) = \frac{N}{m} \sum_{(x_i, y_i) \in \mathcal{D}_m} \nabla_w \log p(y_i \mid t_k(x_i)) + \nabla_w \log p(w) \\ + \cdot \frac{N}{m} \sum_{x_i \in \mathcal{D}_m} \frac{K}{k} \sum_{t_j \in \mathcal{T}_k} \nabla_w \log p\left(f(t_j(x_i); w)) \mid f(x_i; w)\right). \end{aligned} \tag{23}$$

Note that to use the augmentation likelihood in Eq. (22), we need to know the number K of possible augmentations that we consider.

K Further Experimental Results

K.1 Synthetic Examples

In the top left panel of Figure 13, we demonstrate that softmax scaling in the presence of label noise produces very underconfident predictions.

For GPs, varying the values of α_ϵ then vary the sharpness of the resulting posterior predictive distribution, as we show in Figure 13 for a two spirals problem with 150 data points and 20 noisy labels. The posterior predictive for $\alpha_\epsilon = 0.001$ is only confident in a very small region of the data, whereas the higher values of η are confident in a broader region indicating a sharper likelihood that enables more certainly about the observed data. This is in comparison to a Laplace approximated Bernoulli GP classifier that is less confident about the data than $\alpha_\epsilon = 0.1$ but performs similarly in terms of confidence to the Dirichlet with $\alpha_\epsilon = 0.25$, suggesting that too sharp of distributions can reduce some confidence.

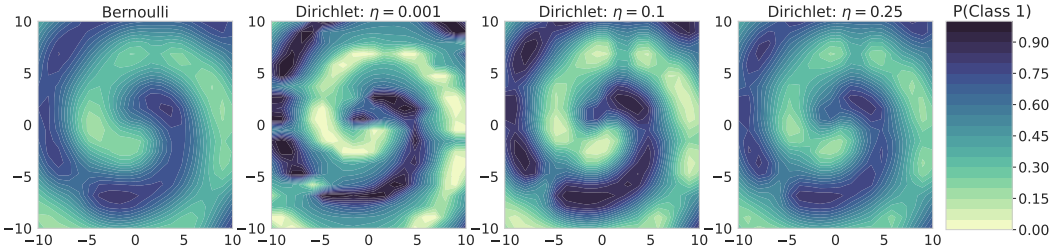
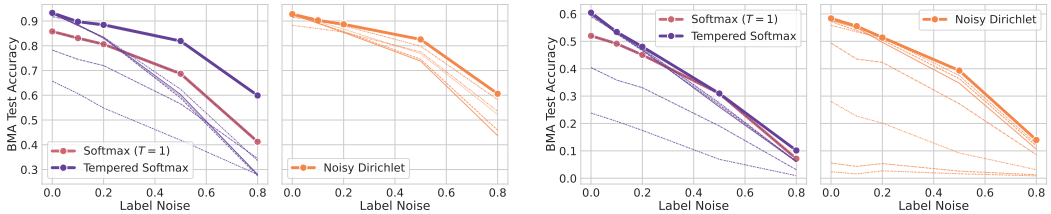


Figure 13: Modeling label noise explicitly using Eq. (14) and using tuned values of η enables better modelling of aleatoric uncertainty in the presence of label noise for GP classifiers on a two spirals problem with 150 data points and 20 flipped labels.

K.2 Label Noise in Image Classification

This section expands on the results presented in Figure 3 by also plotting the sub-optimal values of T and α_ϵ in Figure 14 to provide the reader a more complete picture of performance on various values of the parameters for both likelihoods considered in this work.



(a) CIFAR-10 (b) Tiny Imagenet

Figure 14: We expand on the results in Figure 3 by plotting performance curves for all temperatures T and noise values α_ϵ, with the optimal curve highlighted. Each thin line in purple represents the performance on temperatures $T \in \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 0.1, 1., 3., 10.\}$ for the tempered softmax likelihood, and each thin line in orange represents the performance on noise values $\alpha_\epsilon \in \{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 0.1\}$ for the noisy Dirichlet likelihood. Numerical data is provided in Tables 1 and 2.

K.3 Learning Rate Rescaling

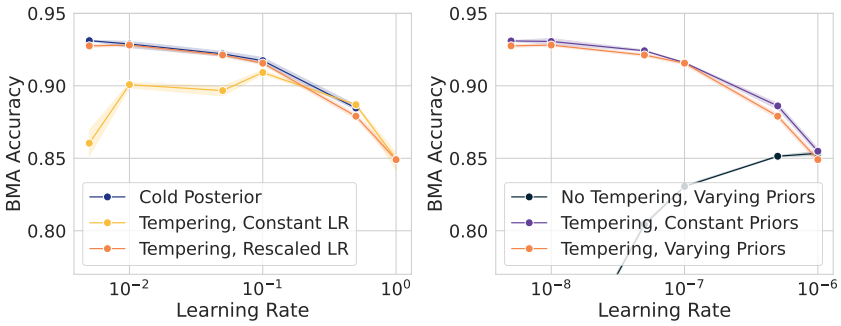


Figure 15: We empirically verify that a tempered likelihood combined with prior rescaling is sufficient to reproduce the cold posterior effect, as discussed in Appendix G.

In Figure 15, we demonstrate that cold posteriors are identical to tempered posteriors when using SGLD with a rescaled learning rate. Again, here, we use ResNet-18 on CIFAR-10. Furthermore, we also need to vary the scale of the Gaussian prior to produce identical results; however the scale of the prior changes the results minimally when using tempering.

Label Noise	T	Test BMA Accuracy (%)	Test NLL
0	0.00001	92.97 ± 0.19	3577.74 ± 36.69
	0.0001	93.13 ± 0.08	3599.74 ± 14.91
	0.001	$\mathbf{93.28} \pm 0.08$	3476.33 ± 18.85
	0.01	92.94 ± 0.12	3521.37 ± 32.88
	0.1	91.77 ± 0.26	3587.2 ± 56.83
	1	85.78 ± 0.19	5344.59 ± 31.25
	3	78.24 ± 0.13	7891.12 ± 48.70
	10	65.69 ± 0.36	12440.25 ± 61.04
0.1	0.00001	88.43 ± 0.08	5872.89 ± 104.96
	0.0001	88.15 ± 0.23	6000.17 ± 124.29
	0.001	88.49 ± 0.35	5885.82 ± 33.58
	0.01	88.23 ± 0.33	5777.52 ± 97.05
	0.1	$\mathbf{89.71} \pm 0.20$	4322.14 ± 35.02
	1	83.08 ± 0.44	6419.33 ± 69.41
	3	74.51 ± 0.15	9088.32 ± 35.38
	10	60.70 ± 1.01	13551.9 ± 81.70
0.2	0.00001	83.30 ± 0.37	8346.63 ± 87.89
	0.0001	83.21 ± 0.40	8425.63 ± 185.47
	0.001	82.97 ± 0.57	8345.62 ± 284.03
	0.01	83.49 ± 0.09	7920.99 ± 103.56
	0.1	$\mathbf{88.45} \pm 0.61$	5448.62 ± 125.64
	1	80.62 ± 0.21	7774.44 ± 27.49
	3	71.97 ± 0.46	10406.1 ± 57.89
	10	54.85 ± 1.37	15327.49 ± 295.44
0.5	0.00001	60.31 ± 1.10	17986.78 ± 364.34
	0.0001	58.87 ± 0.65	18381.33 ± 187.42
	0.001	59.60 ± 0.33	17816.36 ± 258.46
	0.01	62.40 ± 0.43	15499.53 ± 168.51
	0.1	$\mathbf{81.91} \pm 0.39$	9998.57 ± 185.99
	1	68.70 ± 0.55	13187.17 ± 73.68
	3	56.54 ± 0.40	16126.59 ± 78.99
	10	41.85 ± 0.54	19707.41 ± 91.98
0.8	0.00001	27.96 ± 1.22	29862.19 ± 1241.84
	0.0001	27.43 ± 1.36	31027.27 ± 382.76
	0.001	28.08 ± 0.65	29515.91 ± 637.08
	0.01	33.56 ± 1.43	22806.11 ± 1126.21
	0.1	$\mathbf{59.90} \pm 0.45$	17524.06 ± 81.75
	1	41.23 ± 0.36	20839.3 ± 58.07
	3	34.44 ± 1.29	21811.33 ± 68.24
	10	27.96 ± 1.20	23129.04 ± 21.34

Table 1: Performance of the standard softmax likelihood model on CIFAR-10, used to plot results in Figure 3(a). We show one standard deviation over 3 trials.

K.4 Corrupted CIFAR-10 Experiments

Finally, in Figure 16, we display the results for both tempered softmax and noisy Dirchlet for ResNet-18s on corrupted CIFAR-10 [22].

K.5 Data Cardinality and Underfitting

In Figure 17, we show that increasing the cardinality of the dataset improves the training fit, as measured by the average NLL (negative log likelihood) over the training set, showing that dataset cardinality alone cannot explain underfitting.

Label Noise	α_ϵ	Test BMA Accuracy (%)	Test NLL
0.	0.000001	92.75 ± 0.24	8080.56 ± 131.65
	0.00001	92.72 ± 0.10	6874.20 ± 36.90
	0.0001	$\mathbf{92.78} \pm 0.09$	5586.79 ± 86.25
	0.001	92.15 ± 0.07	4491.19 ± 117.73
	0.01	91.50 ± 0.07	3564.69 ± 50.87
	0.1	88.26 ± 0.07	6384.59 ± 15.93
0.1	0.000001	89.11 ± 0.29	11507.79 ± 345.42
	0.00001	89.11 ± 0.24	9638.78 ± 324.75
	0.0001	89.70 ± 0.06	7399.10 ± 64.58
	0.001	$\mathbf{90.28} \pm 0.37$	5195.21 ± 48.68
	0.01	90.25 ± 0.39	3955.32 ± 77.80
	0.1	86.93 ± 0.05	7194.98 ± 24.86
0.2	0.000001	85.36 ± 0.33	14221.11 ± 161.47
	0.00001	85.72 ± 0.40	11884.17 ± 241.67
	0.0001	86.71 ± 0.24	8865.18 ± 187.91
	0.001	88.09 ± 0.18	6006.20 ± 83.82
	0.01	$\mathbf{88.68} \pm 0.22$	4471.03 ± 62.10
	0.1	85.48 ± 0.01	8144.23 ± 11.36
0.5	0.000001	73.80 ± 1.12	19231.71 ± 760.60
	0.00001	74.54 ± 0.99	16510.63 ± 755.69
	0.0001	76.85 ± 0.33	12659.80 ± 540.86
	0.001	79.86 ± 0.29	8708.74 ± 136.45
	0.01	$\mathbf{82.56} \pm 0.19$	6995.20 ± 54.81
	0.1	77.55 ± 0.54	12331.45 ± 26.44
0.8	0.000001	43.71 ± 2.93	31334.21 ± 2914.70
	0.00001	45.89 ± 1.50	26160.53 ± 1253.68
	0.0001	52.24 ± 1.60	19088.42 ± 809.65
	0.001	58.25 ± 1.43	14744.07 ± 560.55
	0.01	$\mathbf{60.61} \pm 1.60$	14896.93 ± 296.23
	0.1	54.16 ± 0.78	19904.50 ± 187.42

Table 2: Performance of the noisy Dirichlet model on CIFAR-10, used to plot results in Figure 3(a). We show one standard deviation over 3 trials.

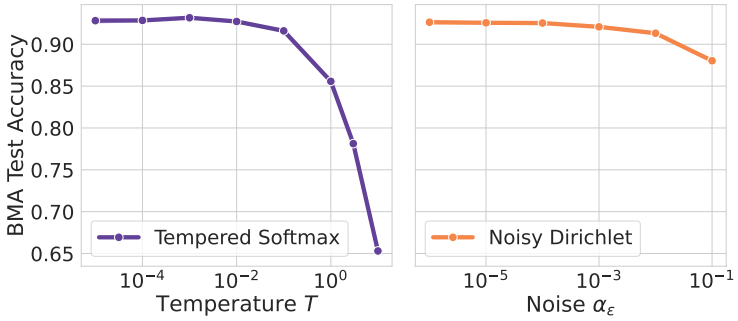


Figure 16: Test accuracy on corrupted CIFAR-10 for both tempered softmax (left) and noisy Dirichlet. Noisy Dirichlet performs very similar to the best performing softmax temperatures (10^{-3}) across a range of α_ϵ values.

K.6 Model Calibration

Section 4 is in general related to miscalibration. We, however, note that in standard neural networks, miscalibration typically means *overconfidence*, while in our work we show that Bayesian neural networks are typically *underconfident*. Both issues are related to model misspecification, where the model does not correctly represent the data-generating process. We show that we can address part of the model misspecification by explicitly encoding our assumptions about aleatoric uncertainty with likelihood tempering or the noisy Dirichlet model.

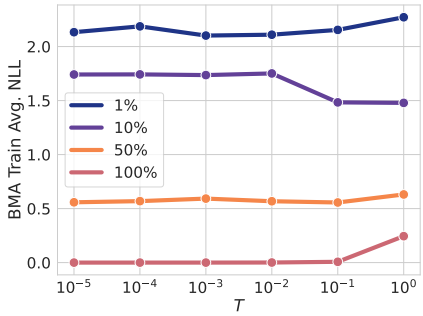


Figure 17: For various subsets of the unaugmented CIFAR-10 dataset, we visualize the average train NLL as a measure of the train data fit using ResNet-18. We find that increasing the cardinality of the dataset improves the fit over the training data. Conclusively, cardinality of the dataset cannot explain the underfitting as observed in Figure 5.

In Table 3, we present the test expected calibration error (ECE) on CIFAR-10 with 20% label noise for the model with standard softmax likelihood, tempered softmax likelihood and the noisy Dirichlet model. The noisy Dirichlet model provides the best calibration (i.e. lowest ECE), and suggests a more accurate description of the data-generating process.

Model	**Test BMA Accuracy** (%)	**Test ECE**
Noisy Dirichlet ($\alpha_\epsilon^\star = 0.01$)	**88.68**	**0.024**
Tempered Softmax ($T^\star = 0.1$)	88.45	0.144
Standard Softmax	80.62	0.179

Table 3: On CIFAR-10 with 20% label noise, we compare the test Expected Calibration Error (ECE) between three models — noisy Dirichlet with optimal noise parameter $\alpha_\epsilon^\star$ (see Table 1), tempered softmax with optimal temperature $T^\star$ (see Table 2), and standard softmax. We find that the noisy Dirichlet model, performs at least as well than the tempered softmax model, while being significantly better calibrated. In general, we expect models which more accurately represent the data-generating process to be better calibrated.

L Experimental Details

We use PyTorch for all experiments [50]. For the Gaussian process experiments, we used GPyTorch [15] for all but the Laplace bernoulli classifier where we used scikit-learn [51].

We used $\mathcal{N}(0, 1)$ priors unless otherwise stated and ran cyclical learning rate SGLD for 1000 epochs with an initial learning rate of $1e-6$ and a momentum term of 0.99. We used ResNet-18 [20] architectures unless otherwise stated and used the CIFAR-10[8].

For experiment management, we used wandb [5] to manage experiments. Experiments were performed on a mix of Nvidia RTX GPUs on internal servers and clusters as well as some RTX GPUs on AWS; we also used some AMD mi50 GPUs on an internal cluster. Each SGLD trial took roughly 10 hours to run, with a total of 3 trials per experiment. Over the course of all experiments including ones that did not make it into the paper, this is approximately 1 GPU year. Unless otherwise mentioned, all experiments used 3 random seeds and we plotted the mean with shading via one standard deviation.

L.1 Synthetic problem

We present the code used to generate the data for the synthetic problem considered in Appendix E in Listing 1 following Maddox et al. [39].

[8] https://www.cs.toronto.edu/~kriz/cifar.html dataset

For all models we use an iid Gaussian prior $\mathcal{N}(0, 0.3^2)$ over the parameters. We use HMC with a step size $3 \cdot 10^{-6}$ and the trajectory length is $\pi \cdot 0.3/2$, amounting to $150 \cdot 10^3$ leapfrog steps per iteration, following the advice in Izmailov et al. [31]. We run HMC for 100 iterations, discarding the first 10 samples as burn-in.

For data augmentation, we generate $K = 4$ augmentations of each datapoint. We run HMC on the distribution in Eq. (11). For the tempered likelihood, we use $T = 0.1$, and for the noisy Dirichlet model we use $\alpha_\epsilon = 10^{-5}$.

Listing 1: Data generation for the synthetic problem

```
import numpy as np

def twospirals(n_samples, noise=.5, random_state=920):
    """
     Returns the two spirals dataset.
    """
    onp.random.seed(random_state)

    n = np.sqrt(np.random.rand(n_samples,1)) * 600 * (2*np.pi)/360
    d1x = -1.5*np.cos(n)*n + np.random.randn(n_samples,1) * noise
    d1y =  1.5*np.sin(n)*n + np.random.randn(n_samples,1) * noise
    return (np.vstack((np.hstack((d1x,d1y)), np.hstack((-d1x,-d1y)))),
            np.hstack((np.zeros(n_samples), np.ones(n_samples))))

x, y = twospirals(n_samples=200, noise=0.6, random_state=920)

label_mask = (np.linalg.norm(x, axis=-1) > 13)
y[label_mask] = 1 - y[label_mask]

mask = np.logical_and(x[:, 0] < 0, x[:, 1] < 0)
x, y = x[mask], y[mask]
x_aug = np.concatenate([x, -x,
                        x * onp.array([[1., -1]]),
                        x * onp.array([[-1., 1]])])
y_aug = np.concatenate([y, y, y, y])
```