On Uncertainty, Tempering, and Data Augmentation in Bayesian Classification

Anonymous Author(s)
Affiliation
Address
email

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 data augmentation in approximate inference softens the likelihood, leading to underconfidence and profoundly misrepresenting our honest 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 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.

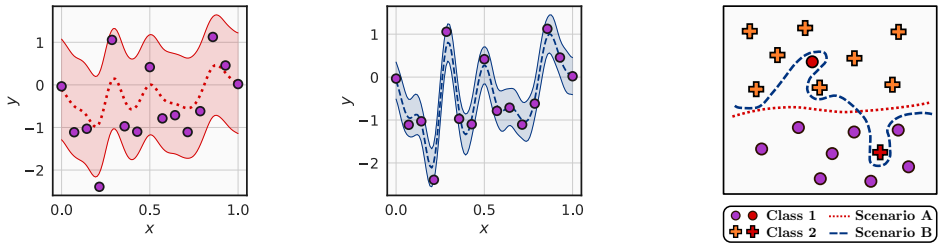
1 Introduction

Uncertainty is often compartmentalized into *epistemic* and *aleatoric uncertainty* [25, 30, 36]. 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 intrinsic irreducible uncertainty, often representing measurement noise in regression, or mislabeled training points in classification [e.g. 3]. 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 [49].

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 [47] 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 $\mathcal{N}(0, \sigma^2)$, with $\sigma^2 = 1$). Consequently, this model explains many of the observations with noise — such that

[1] Epistemic uncertainty also has a significant effect on predictive accuracy, see Wilson and Izmailov [52].

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.



(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.

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* [51] more honestly reflect our beliefs about aleatoric uncertainty than standard Bayesian posteriors. As an alternative to cold posteriors, we describe a *noisy Dirichlet model*, which can be viewed as an input dependent prior, to provide direct control over aleatoric uncertainty for BNN classifiers. This approach achieves competitive performance on image classification problems with a valid likelihood and no need for cold posteriors. 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. [29] and Fortuin et al. [12], 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 Hamiltonian Monte Carlo based MCMC [e.g. 33, 23, 40]. More recently, Wilson and Izmailov [52] show how Bayesian model averaging is especially compelling for *modern* deep networks. A number of recent works have focused

on making Bayesian deep learning practical [e.g., 53, 34, 42, 55, 52, 10, 9], often with better results than classical training, and essentially no additional runtime overhead.

Cold posterior effect. Wenzel et al. [51] 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. [29] showed that for *all* cases considered in Wenzel et al. [51] 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 [12, 41, 39]. Several works have suggested that misspecified priors explain cold posteriors, arguing that the current default choice of isotropic Gaussian priors in BNNs are inadequate [51, 54, 12]. However, Fortuin et al. [12] 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. [29] 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. [41] 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 mis-estimated. Nabarro et al. [39] 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.

Modeling label noise. Works have also focused on modifying the loss function in the context of label noise for classification. For example, Collier et al. [7, 8] proposed input dependent noise losses for label noise in classification, but did not focus on the uncertainty estimates, simply accuracy. Similarly, Zhang and Sabuncu [56] and Feng et al. [11] proposed modifications of the standard cross-entropy loss to provide increased robustness to label noise, but did not focus on the modelling of the inherent uncertainty in label noise settings.

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 [37], 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 $p(\mathcal{D} \mid w) = \prod_{i=1}^N p(y_i \mid x_i, w)$, and prior over parameters $p(w)$. For any novel input $x_\star$, we estimate the posterior predictive distribution via *Bayesian model averaging* (BMA) given by $p(y_\star \mid x_\star) = \int p(y_\star \mid x_\star, w)p(w \mid \mathcal{D})dw$. This integral 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., 52, 38].

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. [51], we then define a *cold posterior* for $T < 1$ as $p_{\text{cold}}(w \mid \mathcal{D}) \propto \exp\big[-\frac{1}{T}U(w)\big]$, 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. Grünwald [15]) only

raises the likelihood to a power $1/T$ as

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

Bayesian Classification. For a C-class classification problem, a standard choice of observation likelihood is the categorical distribution $p(y \mid x, w) = \text{Cat}([\pi_1, \pi_2, \ldots, \pi_C])$, where each class probability is computed using a softmax transform of logits $f(x; w) \in \mathbb{R}^C$ as $\pi_c \propto \exp\{[f(x; w)]_c\}$, 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 with SGLD, and then used to approximate the BMA using simple Monte Carlo.

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 in 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 $f_y(x, w)$. We can think of the class probability vectors $f(x) = (f_1(x, w), \ldots, f_C(x, w))$ as latent variables, where the prior distribution $p(w)$ over the parameters of the network implies a joint prior distribution over $\{f(x)\}_{x \in \mathcal{D}}$. We will initially focus on the observation model. First, we note that we can introduce a uniform prior over the predicted class probabilities $f(x)$: $p(y \mid f(x)) = f_y(x) \propto \text{Dir.}(1, \ldots, 1)(f(x)) \cdot f_y(x)$, where Dir. denotes the Dirichlet distribution and $\text{Dir.}(1, \ldots, 1)$ is a uniform distribution over the class probabilities $f(x)$ (with support on the unit C-simplex), and the proportionality is with respect to the latent variables $f(x)$. We can view the right hand side of this expression as the unnormalized posterior in a multinomial model, where we use a uniform prior over the the parameters $f(x)$, and observe a single count of class y. Using the fact that the Dirichlet distribution is conjugate to the multinomial likelihood [e.g., 5, Ch. 2.2.1], we can rewrite the observation model as follows:

$$p(y \mid f(x)) \propto \text{Dir.}(1, \ldots, 1, \underbrace{2}_{\text{position } y}, 1, \ldots, 1)(f(x)). \tag{2}$$

Eq. (2) describes the distribution induced by the observation y on the predicted class probabilities for the corresponding input x. The posterior over the parameters w of the network can then be written as a product of Eq. (2) and the prior $p(w)$:

$$p(w \mid D) \propto p(w) \prod_{x,y \in \mathcal{D}} \text{Dir.}(1, \ldots, 1, \underbrace{2}_{\text{position } y}, 1, \ldots, 1)(f(x)). \tag{3}$$

Eq. (3) provides intuition for how BNNs estimate aleatoric uncertainty in classification. If we ignore the prior $p(w)$ and assume that the implied prior over $f(x)$ is uniform, then for an observation (x, y) the posterior over $f(x)$ is $\text{Dir.}(1, \ldots, 2, \ldots, 1)$ and the posterior mean for the probability of the correct class $\mathbb{E} f_y(x)$ is $\frac{2}{C+1}$. For example, for a dataset with 100 classes (e.g. CIFAR-100), the model on average will only be $2/101 \approx 2\%$ confident in the correct label on the *training data*!

In practice, the prior $p(w)$ will imply a non-trivial joint distribution over $\{f(x)\}_{x \in \mathcal{D}}$, so the actual posterior may be more (or less) confident than suggested by the analysis above. Furthermore, we have very limited understanding of the implied distribution [see 51, 52, 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 would lead to highly confident $f(x)$ is challenging, but we will show how modifying the likelihood can be viewed as providing an input-dependent prior that more fully reflects our beliefs about aleatoric uncertainty.

The expression in Eq. (3) suggests two natural ways of modifying the posterior to account for the aleatoric uncertainty: increasing the Dirichlet concentration for the observed class (equivalent to

posterior tempering) and 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 (see Section 3) in BNN classification can be written as $p_{temp}(w \mid D) \propto p(w) \prod_{x,y \in \mathcal{D}} f_y(x, w)^{1/T}$, where T is the temperature. Analogously to the derivation above, we can rewrite this posterior as

$$p_{temp}(w \mid D) \propto p(w) \prod_{x,y \in \mathcal{D}} \text{Dir.}\big(1, \ldots, 1, \underbrace{1 + 1/T}_{\text{position } y}, 1, \ldots, 1\big)(f(x)). \tag{4}$$

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 $f(x)$, the confidence in the correct labels on the train data under the tempered posterior $\mathbb{E} f_y(x) = \frac{1+1/T}{C+1/T}$. With 100 classes and at $T = 10^{-2}$ we get an average confidence of 50.5%, 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 F.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 [51, 41, 12]. 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. 46]. Thus, we may expect that tempered likelihoods with low temperatures will lead to optimal performance.

Is a Tempered Likelihood a Valid Likelihood? In classical Bayesian inference, the observation model is a distribution $p(y \mid x)$ over the labels conditioned on the input. Wenzel et al. [51] argue that the tempered softmax likelihood is in general not a valid likelihood because it does not sum to 1 over classes. However, Wenzel et al. [51] show, 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. (4), 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.

4.3 Noisy Dirichlet model: changing the prior over class probabilities

As we have seen in Eq. (4), 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*[2]. Indeed, consider the distribution

$$p_{ND}(w \mid D) \propto p(w) \prod_{x,y \in \mathcal{D}} \text{Dir.}(\alpha_\epsilon, \ldots, \alpha_\epsilon, \underbrace{\alpha_\epsilon + 1}_{\text{position } y}, \alpha_\epsilon, \ldots, \alpha_\epsilon)(f(x)), \tag{5}$$

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. [37] in the context of GP classification, where they designed a tractable approximation to this model. Using our running example, if $p(w)$ induces a uniform distribution over $f(x)$, for a problem with 100 classes and $\alpha_\epsilon = 10^{-2}$ we have expected confidence $\mathbb{E} f_y(x) = \frac{\alpha_\epsilon + 1}{C\alpha_\epsilon + 1} = 50.5\%$, same as for the tempered likelihood with $T = 10^{-2}$.

[2] While for concreteness 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.

Using the conjugacy of the Dirichlet and multinomial distributions, we can rewrite p_{ND} as follows:

$$p_{ND}(w \mid D) \propto p(w) \prod_{x,y \in \mathcal{D}} \text{Dir.}(\alpha_\epsilon, \ldots, \alpha_\epsilon)(f(x)) \cdot f_y(x) = q_{ND}(w) f_y(w). \tag{6}$$

We can interpret $q_{ND}(w) = p(w) \cdot \prod_{x \in \mathcal{D}} \text{Dir.}(\alpha_\epsilon, \ldots, \alpha_\epsilon)(f(x))$ 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. 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 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.

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. [37] considered the distribution in Eq. (6) 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 C. While theoretically we do not need to use the Gaussian approximation for Bayesian neural networks and can directly use Eq. (6), 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 [29, 54, 12, 39]. We show that data augmentation, as applied in practice[3], 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 B). In prior work, e.g. in Wenzel et al. [51], 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

$$\sum_{i=1}^{N} \mathbb{E}_t[\nabla_w \log p(y_i | t(x_i))] + \nabla_w \log p(w) = \sum_{i=1}^{N} \sum_{j=1}^{K} [\nabla_w \log p(y_i | t_j(x_i))^{\frac{1}{K}}] + \nabla_w \log p(w), \tag{7}$$

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

$$p_{\text{aug}}(w \mid \mathcal{D}) \propto p(w) \prod_{i=1}^{N} \prod_{j=1}^{K} p(y_i \mid t_j(x_i))^{1/K}. \tag{8}$$

We can interpret the limiting distribution in Eq. (8) 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

[3] 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.

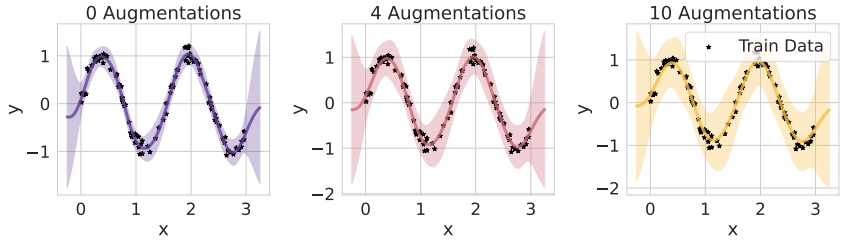


Figure 2: **Effect of data augmentation on GP regression.** As we increase the number K of augmentations of the dataset, a GP model fit becomes more diffuse. Here the original training data is shown with black stars, and the augmented datapoints $t_j(x) = x + \tau_j$ are not shown. Both the predictive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

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. (8), 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.

What about Other Inference Procedures? While the derivation in Eq. (8) comes directly from studying the stochastic gradient evaluated in SGMCMC, we can equivalently derive the same posterior distribution in stochastic variational inference [24] as we show in Appendix H.

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. [39] for a related discussion.

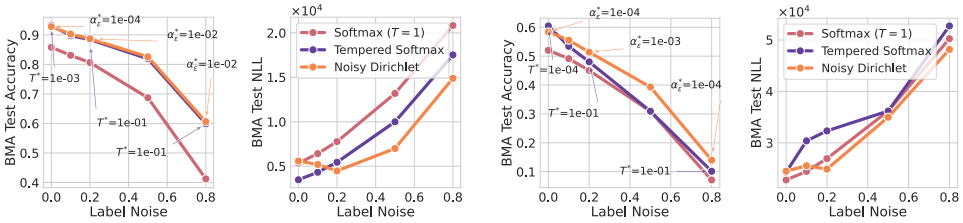
6 Experiments

In this section we provide empirical support for the observations presented in Sections 4 and 5. In Appendix D 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.

6.1 Visualizing the Effect of Data Augmentation

To illustrate our analysis in Section 5 we show the posterior in Eq. (8) 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. (8) for $K = 1, 4$ and 10 augmentations per datapoint in Figure 2. As the predictions on the train data are independent from the predictions



(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.

on the augmented datapoints, the posterior in Eq. (8) 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 the noisy Dirichlet model does not require tempering to achieve optimal performance. Then, 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. 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 [18] and the CIFAR-10 [31] and Tiny Imagenet [32] datasets. We use the SGLD sampler with a cyclical learning rate schedule [50, 55] to sample from the posterior. We provide details on the hyper-parameters in Appendix K.

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. [51], 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.

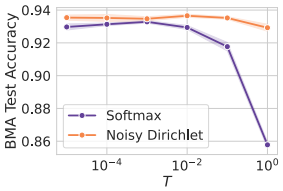
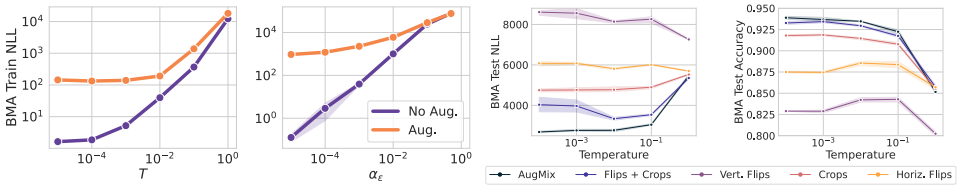


Figure 4: BMA test accuracy for the noisy Dirichlet model with $\alpha_\epsilon = 10^{-6}$ and the softmax likelihood as a function of temperature on CIFAR-10. The noisy Dirichlet model shows no cold posterior effect.

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 like-



(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.

lihood 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.

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 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 [21] — 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 [20], evaluating our models with both tempered softmax (TS) and noisy Dirichlet models. A full plot is shown in Appendix J; 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.

References

[1] Ben Adlam, Jasper Snoek, and Samuel L Smith. Cold posteriors and aleatoric uncertainty. *arXiv preprint arXiv:2008.00029*, 2020. 3, 18

[2] Laurence Aitchison. A statistical theory of cold posteriors in deep neural networks. In *International Conference on Learning Representations*, 2020. 3, 5, 17, 19

[3] 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

[4] Lukas Biewald. Experiment tracking with weights and biases, 2020. URL https://www.wandb.com/. Software available from wandb.com. 22

[5] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006. 4

[6] 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. 14

[7] 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

[8] 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

[9] 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

[10] 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

[11] 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

[12] 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, 6, 18, 19, 23

[13] 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. 22

[14] 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. 19

[15] Peter Grünwald. The safe bayesian. In *International Conference on Algorithmic Learning Theory*, pages 169–183. Springer, 2012. 3

[16] 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. 17

[17] 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. 16

[18] 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. 8, 22

[19] Jonathan Heek and Nal Kalchbrenner. Bayesian inference for large scale image classification. *arXiv preprint arXiv:1908.03491*, 2019. 18

[20] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019. 9

[21] 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. 9

[22] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 16

[23] 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

[24] Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 2013. 7

[25] 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

[26] W Ronny Huang, Zeyad Emam, Micah Goldblum, Liam Fowl, Justin K Terry, Furong Huang, and Tom Goldstein. Understanding generalization through visualizations. 2020. 16

[27] 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. 21

[28] Pavel Izmailov, Patrick Nicholson, Sanae Lotfi, and Andrew Gordon Wilson. Dangers of bayesian model averaging under covariate shift. *arXiv preprint arXiv:2106.11905*, 2021. 19, 23, 24

[29] 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, 6, 15, 19, 22, 23

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

[31] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009. 8, 13

[32] Ya Le and Xuan S. Yang. Tiny imagenet visual recognition challenge. 2015. 8, 13

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

[34] 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

[35] Wesley J Maddox, Gregory Benton, and Andrew Gordon Wilson. Rethinking parameter counting in deep models: Effective dimensionality revisited. *arXiv preprint arXiv:2003.02139*, 2020. 15

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

[37] Dimitrios Milios, Raffaello Camoriano, Pietro Michiardi, Lorenzo Rosasco, and Maurizio Filippone. Dirichlet-based gaussian processes for large-scale calibrated classification. In *NeurIPS*, 2018. 3, 5, 6, 14

[38] Kevin P. Murphy. *Probabilistic Machine Learning: Advanced Topics*. MIT Press, 2023. URL probml.ai. 3

[39] 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, 6, 7, 23

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

[41] 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, 23, 24

[42] 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

[43] 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. 22

[44] 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. 22

[45] 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. 16

[46] 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

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

[48] Simo Särkkä and A. Solin. Applied stochastic differential equations. 2019. 14

[49] 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

[50] 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. 8, 14

[51] 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, 8, 16, 18, 19, 20, 23

[52] Andrew Gordon Wilson and Pavel Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. *arXiv preprint arXiv:2002.08791*, 2020. 1, 2, 3, 4, 19, 23, 24

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

[54] 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, 6, 16

[55] 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, 8, 14, 18

[56] 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

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. 7, Appendices F, H
   (b) Did you include complete proofs of all theoretical results? [Yes] See Eq. 7, Appendices F, H
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 K.
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 [31], TinyImagenet [32].
   (b) Did you mention the license of the assets? [Yes] CIFAR-10, MIT License, TinyImagenet, MIT License [32]
   (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]