A PROOFS FOR SECTION 3

In this section, we provide the proofs for Section 3 in the following order. We first prove the gradient of the empirical Gibbs loss in Theorem 2. Then, we show in Proposition 5 that for meaningful posteriors (depends on training data), the gradient won't be zero. Before proving Proposition 3 and Theorem 4, we first provide Proposition 6, stating an alternative expression of the gradient of the Bayes loss. The proofs of Proposition 3 and Theorem 4 then follow from that.

A.1 PROOF OF THEOREM 2

We first show a slightly more general result of $\nabla_\lambda \mathbb{E}_{p^\lambda}[f(\boldsymbol{\theta})]$ for any function $f(\boldsymbol{\theta})$ that is independent of λ. Recall that the posterior $p^\lambda(\boldsymbol{\theta}|D) \propto p(D|\boldsymbol{\theta})^\lambda p(\boldsymbol{\theta})$. With the fact that $\nabla_\lambda \left(p(D|\boldsymbol{\theta})^\lambda p(\boldsymbol{\theta})\right) = \ln(p(D|\boldsymbol{\theta}))p(D|\boldsymbol{\theta})^\lambda p(\boldsymbol{\theta})$, the gradient

$$\nabla_\lambda \mathbb{E}_{p^\lambda}[f(\boldsymbol{\theta})] = \mathbb{E}_{p^\lambda}[\ln p(D|\boldsymbol{\theta}) f(\boldsymbol{\theta})] - \mathbb{E}_{p^\lambda}[\ln p(D|\boldsymbol{\theta})]\mathbb{E}_{p^\lambda}[f(\boldsymbol{\theta})] = \mathrm{COV}_{p^\lambda}(\ln p(D|\boldsymbol{\theta}), f(\boldsymbol{\theta}))\,, \tag{14}$$

where we denote $\mathrm{COV}(X, Y)$ as the covariance of X and Y. Hence, the gradient of the empirical Gibbs loss

$$\nabla_\lambda \hat{G}(p^\lambda, D) = \nabla_\lambda \mathbb{E}_{p^\lambda}[-\ln p(D|\boldsymbol{\theta})] = \mathrm{COV}_{p^\lambda}(\ln p(D|\boldsymbol{\theta}), -\ln p(D|\boldsymbol{\theta})) = -\mathbb{V}_{p^\lambda}(\ln p(D|\boldsymbol{\theta}))\,.$$

A.2 PROPOSITION 5

Proposition 5. *For any $\lambda > 0$ and $D \neq \emptyset$, if the tempered posterior $p^\lambda(\boldsymbol{\theta}|D) \propto p(D|\boldsymbol{\theta})^\lambda p(\boldsymbol{\theta})$ satisfies $\mathbb{V}_{p^\lambda}(\ln P(D|\boldsymbol{\theta})) = 0$, then, $p^\lambda(\boldsymbol{\theta}|D) = p(\boldsymbol{\theta})$.*

Proof. First of all, note that the tempered posterior is defined as

$$p^\lambda(\boldsymbol{\theta}|D) = \frac{p(D|\boldsymbol{\theta})^\lambda p(\boldsymbol{\theta})}{\int_{\boldsymbol{\theta}} p(D|\boldsymbol{\theta})^\lambda p(\boldsymbol{\theta})}\,. \tag{15}$$

Then,

$$\mathbb{V}_{p^\lambda}(\ln p(D|\boldsymbol{\theta})) = 0 \implies \int_{\boldsymbol{\theta}} p^\lambda(\boldsymbol{\theta}|D)\left(\ln p(D|\boldsymbol{\theta}) - \mathbb{E}_{p^\lambda}[\ln p(D|\boldsymbol{\theta})]\right)^2 = 0$$

Thus, for any $\boldsymbol{\theta} \in \mathrm{supp}(p^\lambda)$, it verifies that

$$\ln p(D|\boldsymbol{\theta}) = \mathbb{E}_{p^\lambda}[\ln p(D|\boldsymbol{\theta})]\,.$$

That is, $\ln p(D|\boldsymbol{\theta})$ is constant in the support of p^λ. Let c denote such constant, then

$$p^\lambda(\boldsymbol{\theta}|D) = \frac{e^{c\lambda} p(\boldsymbol{\theta})}{\int_{\boldsymbol{\theta}} e^{c\lambda} p(\boldsymbol{\theta})} = \frac{e^{c\lambda} p(\boldsymbol{\theta})}{e^{c\lambda}\int_{\boldsymbol{\theta}} p(\boldsymbol{\theta})} = p(\boldsymbol{\theta})\,.$$

□

A.3 PROOF OF PROPOSITION 3 AND THEOREM 4

In order to prove Proposition 3 and Theorem 4, we first show in Proposition 6 that the gradient of the Bayes loss of the tempered posterior p^λ can be expressed by the difference between the empirical Gibbs loss of $\bar{p}^\lambda$ and the empirical Gibbs loss of p^λ.

Proposition 6. *The gradient of the Bayes loss of the tempered posterior p^λ can be expressed by*

$$\nabla_\lambda B(p^\lambda) = \hat{G}(\bar{p}^\lambda, D) - \hat{G}(p^\lambda, D)\,. \tag{16}$$

Proof. By definition,

$$\nabla_\lambda B(p^\lambda) = \nabla_\lambda \mathbb{E}_\nu\left[-\ln \mathbb{E}_{p^\lambda}[p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})]\right] = -\mathbb{E}_\nu\left[\nabla_\lambda \ln \mathbb{E}_{p^\lambda}[p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})]\right],$$

where

$$\nabla_\lambda \ln \mathbb{E}_{p^\lambda}[p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})] = \frac{\nabla_\lambda \mathbb{E}_{p^\lambda}[p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})]}{\mathbb{E}_{p^\lambda}[p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})]} = \frac{\mathrm{COV}_{p^\lambda}(\ln p(D|\boldsymbol{\theta}), p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta}))}{\mathbb{E}_{p^\lambda}[p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})]}$$

due to Equation (14). By expanding the covariance, the above formula further equals to

$$\frac{\mathbb{E}_{p^\lambda}[\ln p(D|\boldsymbol{\theta})p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})]-\mathbb{E}_{p^\lambda}[\ln p(D|\boldsymbol{\theta})]\mathbb{E}_{p^\lambda}[p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})]}{\mathbb{E}_{p^\lambda}[p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})]}=\mathbb{E}_{\tilde{p}^\lambda}[\ln p(D|\boldsymbol{\theta})]-\mathbb{E}_{p^\lambda}[\ln p(D|\boldsymbol{\theta})],$$

where the probability distribution $\tilde{p}^\lambda(\boldsymbol{\theta}|D,(\boldsymbol{y},\boldsymbol{x}))\propto p^\lambda(\boldsymbol{\theta}|D)p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})$. Put everything together, we have

$$\nabla_\lambda B(p^\lambda)=\mathbb{E}_{p^\lambda}[\ln p(D|\boldsymbol{\theta})]-\mathbb{E}_\nu\mathbb{E}_{\tilde{p}^\lambda}[\ln p(D|\boldsymbol{\theta})]=\mathbb{E}_{p^\lambda}[\ln p(D|\boldsymbol{\theta})]-\mathbb{E}_{\bar{p}^\lambda}[\ln p(D|\boldsymbol{\theta})], \quad (17)$$

where

$$\bar{p}^\lambda(\boldsymbol{\theta}|D)=\mathbb{E}_\nu[\tilde{p}^\lambda(\boldsymbol{\theta}|D,(\boldsymbol{y},\boldsymbol{x}))]=\mathbb{E}_\nu\left[\frac{p^\lambda(\boldsymbol{\theta}|D)p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})}{\mathbb{E}_{p^\lambda}[p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})]}\right].$$

The last equality is because

$$\begin{aligned}\mathbb{E}_\nu\mathbb{E}_{\tilde{p}^\lambda}[\ln p(D|\boldsymbol{\theta})]&=\int_{(\boldsymbol{y},\boldsymbol{x})}\nu(\boldsymbol{y},\boldsymbol{x})\int_{\boldsymbol{\theta}}\tilde{p}^\lambda(\boldsymbol{\theta}|D,(\boldsymbol{y},\boldsymbol{x}))\ln p(D|\boldsymbol{\theta})d\boldsymbol{\theta}d(\boldsymbol{y},\boldsymbol{x})\\&=\int_{\boldsymbol{\theta}}\left(\int_{(\boldsymbol{y},\boldsymbol{x})}\nu(\boldsymbol{y},\boldsymbol{x})\tilde{p}^\lambda(\boldsymbol{\theta}|D,(\boldsymbol{y},\boldsymbol{x}))d(\boldsymbol{y},\boldsymbol{x})\right)\ln p(D|\boldsymbol{\theta})d\boldsymbol{\theta}\\&=\int_{\boldsymbol{\theta}}\mathbb{E}_\nu[\tilde{p}^\lambda(\boldsymbol{\theta}|D,(\boldsymbol{y},\boldsymbol{x}))]\ln p(D|\boldsymbol{\theta})d\boldsymbol{\theta}\\&=\mathbb{E}_{\bar{p}^\lambda}[\ln p(D|\boldsymbol{\theta})].\end{aligned}$$

The last expression in Equation (17) further equals to $\hat{G}(\bar{p}^\lambda,D)-\hat{G}(p^\lambda,D)$ by definition. □

A.3.1 PROOF OF PROPOSITION 3

Note that for any distribution ρ, we have $\hat{G}(\rho,D):=\mathbb{E}_\rho[-\ln p(D|\boldsymbol{\theta})]\geq\min_{\boldsymbol{\theta}}-\ln p(D|\boldsymbol{\theta})$. On the other hand, Proposition 6 together with Definition 1 give that the CPE takes place if and only if

$$\nabla_\lambda B(p^\lambda)_{|\lambda=1}=\hat{G}(\bar{p}^{\lambda=1},D)-\hat{G}(p^{\lambda=1},D)<0.$$

Therefore, it is not possible to have $\hat{G}(p^{\lambda=1},D)\not>\min_{\boldsymbol{\theta}}-\ln p(D|\boldsymbol{\theta})$ and, at the same time, $\hat{G}(\bar{p}^{\lambda=1},D)<\hat{G}(p^{\lambda=1},D)$ because $\hat{G}(\bar{p}^{\lambda=1},D)\geq\min_{\boldsymbol{\theta}}-\ln p(D|\boldsymbol{\theta})$.

A.3.2 PROOF OF THEOREM 4

It's easy to see from Proposition 6 that

$$\nabla_\lambda B(p^\lambda)_{|\lambda=1}=\hat{G}(\bar{p}^{\lambda=1},D)-\hat{G}(p^{\lambda=1},D)=0$$

if and only if $\hat{G}(\bar{p}^{\lambda=1},D)=\hat{G}(p^{\lambda=1},D)$.

B PROOFS FOR SECTION 5

B.1 PROOF OF EQ. (9)

Note that

$$\nabla_\lambda G(p^\lambda)=\nabla_\lambda\mathbb{E}_{p^\lambda}[L(\boldsymbol{\theta})]=\mathrm{COV}_{p^\lambda}(\ln p(D|\boldsymbol{\theta}),L(\boldsymbol{\theta}))=\mathrm{COV}_{p^\lambda}(-\hat{L}(D,\boldsymbol{\theta}),L(\boldsymbol{\theta})),$$

where the second equality is by applying Equation (14). By taking $\lambda=1$, we obtain the desired gradient.

B.2 PROOF OF EQ. (11)

Recall from the proof of Theorem 6 that

$$\nabla_\lambda B(p^\lambda) = \mathbb{E}_{p^\lambda}[\ln p(D|\boldsymbol{\theta})] - \mathbb{E}_{\bar{p}^\lambda}[\ln p(D|\boldsymbol{\theta})] = \mathbb{E}_{\bar{p}^\lambda}[\hat{L}(D,\boldsymbol{\theta})] - \mathbb{E}_{p^\lambda}[\hat{L}(D,\boldsymbol{\theta})],$$

where $\bar{p}^\lambda(\boldsymbol{\theta}|D) = \mathbb{E}_\nu\left[\tilde{p}^\lambda(\boldsymbol{\theta}|D,(\boldsymbol{y},\boldsymbol{x}))\right]$ (Equation (6)), and $\tilde{p}^\lambda(\boldsymbol{\theta}|D,(\boldsymbol{y},\boldsymbol{x})) \propto p^\lambda(\boldsymbol{\theta}|D)p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})$ is the distribution obtained by updating the posterior p^λ with one new sample $(\boldsymbol{y},\boldsymbol{x})$.

Therefore,

$$\mathbb{E}_{\bar{p}^\lambda}[\hat{L}(D,\boldsymbol{\theta})] = \mathbb{E}_\nu\mathbb{E}_{\tilde{p}^\lambda}[\hat{L}(D,\boldsymbol{\theta})] = \mathbb{E}_\nu\left[\mathbb{E}_{p^\lambda}\left[\frac{p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})}{\mathbb{E}_{p^\lambda}[p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})]}\hat{L}(D,\boldsymbol{\theta})\right]\right].$$

By Fubini's theorem, the above formula further equals to

$$\mathbb{E}_{p^\lambda}\left[\mathbb{E}_\nu\left[\frac{p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})}{\mathbb{E}_{p^\lambda}[p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})]}\hat{L}(D,\boldsymbol{\theta})\right]\right] = \mathbb{E}_{p^\lambda}\left[\mathbb{E}_\nu\left[\frac{p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})}{\mathbb{E}_{p^\lambda}[p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})]}\right]\hat{L}(D,\boldsymbol{\theta})\right] = \mathbb{E}_{p^\lambda}\left[-S_{p^\lambda}(\boldsymbol{\theta})\cdot\hat{L}(D,\boldsymbol{\theta})\right].$$

On the other hand, since

$$\mathbb{E}_{p^\lambda}[-S_{p^\lambda}(\boldsymbol{\theta})] = \mathbb{E}_{p^\lambda}\left[\mathbb{E}_\nu\left[\frac{p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})}{\mathbb{E}_{p^\lambda}[p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})]}\right]\right] = \mathbb{E}_\nu\left[\mathbb{E}_{p^\lambda}\left[\frac{p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})}{\mathbb{E}_{p^\lambda}[p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})]}\right]\right] = 1\,,$$

we have

$$\mathbb{E}_{p^\lambda}[\hat{L}(D,\boldsymbol{\theta})] = \mathbb{E}_{p^\lambda}[\hat{L}(D,\boldsymbol{\theta})]\mathbb{E}_{p^\lambda}[-S_{p^\lambda}(\boldsymbol{\theta})]\,.$$

By putting them altogether,

$$\nabla_\lambda B(p^\lambda) = \mathbb{E}_{p^\lambda}\left[-S_{p^\lambda}(\boldsymbol{\theta})\cdot\hat{L}(D,\boldsymbol{\theta})\right] - \mathbb{E}_{p^\lambda}[\hat{L}(D,\boldsymbol{\theta})]\mathbb{E}_{p^\lambda}[-S_{p^\lambda}(\boldsymbol{\theta})] = -\mathrm{COV}\left(\hat{L}(D,\boldsymbol{\theta}), S_{p^\lambda}(\boldsymbol{\theta})\right)\,.$$

C EXPERIMENT DETAILS

Our experimental results can be divided into two parts. In Appendix C.1, we detail the settings of the toy experiment using synthetic data and exact Bayesian linear regression in Figure 1. We also show extra results of the gradients of Gibbs loss and Bayes loss w.r.t to λ approximated by samples. In Appendix C.2, we introduce the settings of the Bayesian CNN (LeCun et al., 1998) experiments on MNIST (LeCun and Cortes, 2010), as well as extra results of the Bayesian CNN on Fashion-MNIST (Xiao et al., 2017), and Bayesian ResNets (He et al., 2016) on CIFAR-10 and CIFAR-100 (Krizhevsky, 2009). Particularly, stochastic gradient Langevin dynamics (SGLD) (Welling and Teh, 2011) is used for inference. At the end, we also show extra results of mean-field variational inference (MFVI) (Blei et al., 2017) on MNIST.

C.1 BAYESIAN LINEAR REGRESSION ON SYNTHETIC DATA WITH EXACT INFERENCE

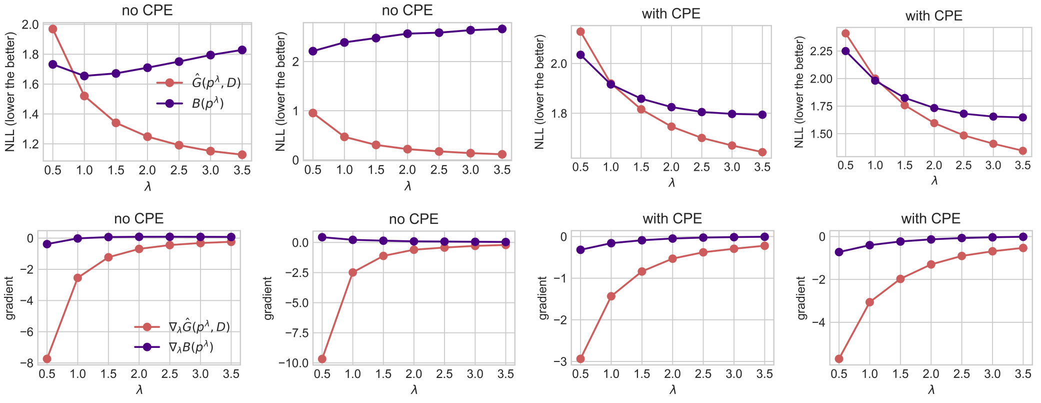
To begin, we will outline the data-generating process for the synthetic data used in the experiment. We sample x uniformly from the $[-1,1]$ interval and pass it through a Fourier transformation to construct the input of the data. That is, for a sampled x, the input $\boldsymbol{x}$ is constructed by a 10-dimensional Fourier basis function $\boldsymbol{\phi}(x) = [g_1(x),...,g_K(x)]^T$ for $K = 10$, where the basis functions are defined as follows: $g_1(x) = \frac{1}{\sqrt{2\pi}}$, and for other odd values of k, $g_k(x) = \frac{1}{\sqrt{\pi}}\sin(kx)$, whereas for even values of k, $g_k(x) = \frac{1}{\sqrt{\pi}}\cos(kx)$. The distribution of the output $y \in \mathbb{R}$ given an input $\boldsymbol{x}$, denoted as $\nu(y|\boldsymbol{x})$, follows a Normal distribution with mean $\mathbf{1}^T\boldsymbol{x}$ and variance 1.0, where $\mathbf{1}$ is an all-ones vector. That is, $\nu(y|\boldsymbol{x}) = \mathcal{N}(\mathbf{1}^T\boldsymbol{x}, 1.0)$.

In our experiment, the likelihood model and the prior model are defined differently for the four settings in Figure 1. To enable exact inference, both the likelihood and the prior are Gaussian, which gives a closed-form solution for the posterior predictive. This choice also provides convenience when studying the CPE: different values of λ on the likelihood term can be naturally absorbed into the Gaussian densities by adjusting the variance (dividing by λ) without hindering the exact inference step. We describe them in detail in the following.

1. No misspecification: likelihood $p(y|\boldsymbol{x}, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}^T\boldsymbol{x}, 1.0)$, prior $p(\boldsymbol{\theta}) = \mathcal{N}(0, 2)$. This is the baseline for comparison.
2. Misspecified likelihood I: likelihood $p(y|\boldsymbol{x}, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}^T\boldsymbol{x}, 0.15)$ (the order of Fourier transformation is $K = 20$, however note that it still contains the $K = 5$ data-generating process in its solution space), prior $p(\boldsymbol{\theta}) = \mathcal{N}(0, 2)$. In this case, the model is misspecified in a way that it has a smaller variance than the data-generating process.
3. Misspecified likelihood II: likelihood $p(y|\boldsymbol{x}, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}^T\boldsymbol{x}, 3.0)$, prior $p(\boldsymbol{\theta}) = \mathcal{N}(0, 2.0)$. In this case, the model is misspecified in a way that it has a larger variance than the data-generating process. This is similar to one of the scenarios where CPE was found: the curated data has a lower aleatoric uncertainty than the model (Aitchison, 2021).
4. Misspecified prior: likelihood $p(y|\boldsymbol{x}, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}^T\boldsymbol{x}, 1.0)$, prior $p(\boldsymbol{\theta}) = \mathcal{N}(0, 0.5)$. The prior is poorly specified in a way that it is tightly centered at 0 while the best $\boldsymbol{\theta}$ should be 1.

In all the experiments, every training set consists of only 5 samples. Since there are more parameters than the number of training data points, our setting falls within the "overparameterized" regime where CPE has been observed in Bayesian deep learning (Wenzel et al., 2020).

Continuing from Figure 1, where we show the Gibbs loss $\hat{G}(\rho, D)$ (training) and the Bayes loss $B(p^\lambda)$ (testing) with respect to λ, we now show their derivatives $\nabla_\lambda \hat{G}(\rho, D)$ (Eq. (5)) and $\nabla_\lambda B(p^\lambda)$ (Eq. (16)) respectively in Figure 4. Here the losses are included for a clearer depiction of the derivatives. To approximate the Bayes loss for generating the plot, we use 10000 data points sampled from the data-generating distribution. Also, the derivatives are approximated using 10000 samples from the exact posteriors. From Figure 4, we could clearly see that the derivatives perfectly characterize the losses in all four settings.



(a) No misspecification (b) Misspecified likelihood I (c) Misspecified likelihood II (d) Misspecified prior

Figure 4: **The derivatives $\nabla_\lambda \hat{G}(\rho, D)$ (Eq. (5)) and $\nabla_\lambda B(p^\lambda)$ (Eq. (16)) characterize the Gibbs loss $\hat{G}(\rho, D)$ and the Bayes loss $B(p^\lambda)$ perfectly.**

C.2 BAYESIAN NEURAL NETWORKS ON IMAGE DATA WITH APPROXIMATE INFERENCE

Briefly speaking, our experiments are divided into 4 groups:

1. Bayesian CNNs (large and small) on MNIST (**in the main text**)
2. Bayesian CNNs (large and small) on Fashion-MNIST
3. Bayesian ResNets (18 and 50) on CIFAR-10
4. Bayesian ResNets (18 and 50) on CIFAR-100

where each group compares a larger model and a smaller model to evaluate the effect of underfitting. Most of the results use stochastic gradient Langevin dynamics (SGLD) (Welling and Teh, 2011) for inference, but we also provide additional results with mean-field variational inference (MFVI) (Blei et al., 2017) later in Appendix C.2.2. We observe that the results of MFVI align with the ones with SGLD.

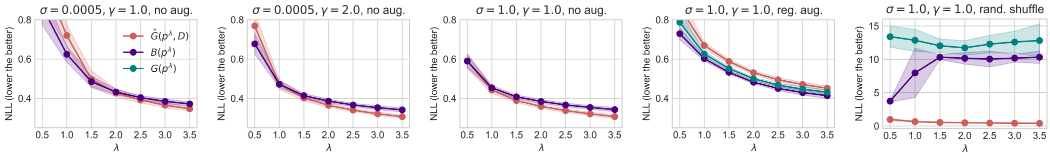
The large CNN we use has a similar architecture to LeNet-5, but with a much larger number of parameters. These are the details of the architecture:

1. Convolutional layer with 6 output channels, kernel size of 5 and *ReLU* activation.
2. MaxPooling layer of kernel size 2 and stride 2.
3. Convolutional layer with 16 output channels, kernel size of 5 and *ReLU* activation.
4. MaxPooling layer of kernel size 2 and stride 2.
5. Convolutional layer with 120 output channels, kernel size of 5 and *ReLU* activation.
6. Dense layer with output dimension of 84 and *ReLU* activation.
7. Dense layer with output dimension of 10 (the number of classes).

In all the convolutional layers, no stride $= 1$ and padding is set to *same*. This model uses a total of 545546 parameters. The small CNN has a similar architecture with a total number of 107786 parameters. In the meanwhile, we do not show the architectures of ResNet-18 and ResNet-50 as they are very common. They have around 11 million and 23 million parameters.

C.2.1 STOCHASTIC GRADIENT LANGEVIN DYNAMICS (SGLD)

We use PyTorch (Paszke et al., 2019) for the experiments with SGLD. We train the model using cyclical learning rate SGLD (cSGLD) (Zhang et al., 2019) for 1000 epochs. We set the learning rate to 1e-6 with a momentum term of 0.99. We run cSGLD for 10 trials and collect 10 samples for each trial, respectively. Experiments were performed on NVIDIA A100 GPU, where 1 trial took around 30 hours.



(a) Narrow prior and standard softmax

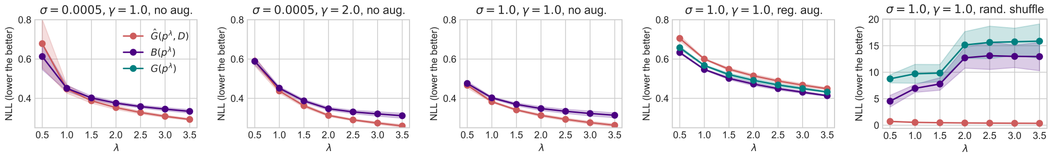
(b) Narrow prior and tempered softmax

(c) Standard prior and standard softmax

(d) Random crop and horizontal flip

(e) Pixels randomly shuffled

Figure 5: **Experimental illustrations for the arguments in Sections 4 and 5 using small CNN via SGLD on Fashion-MNIST.** Figures 5a to 5c illustrate the arguments in Section 4, while Figures 5c to 5e illustrate the arguments in Section 5. Figure 5c uses the standard prior ($\sigma = 1$) and the standard softmax ($\gamma = 1$) for the likelihood without applying DA. Figure 5a follows a similar setup except for using a narrow prior. Figure 5b uses a narrow prior as in Figure 5a but with a tempered softmax that results in a lower aleatoric uncertainty. Figure 5d follows the setup as in Figure 5c but with standard DA methods applied, while Figure 5e uses fabricated DA. We report the training loss $\hat{G}(p^\lambda, D)$ and the testing losses $B(p^\lambda)$ and $G(p^\lambda)$ from 10 samples of the small Convolutional neural network (CNN) via Stochastic Gradient Langevin Dynamics (SGLD). We show the mean and standard deviation across three different seeds. For additional experimental details, refer to Appendix C.



(a) Narrow prior and standard softmax

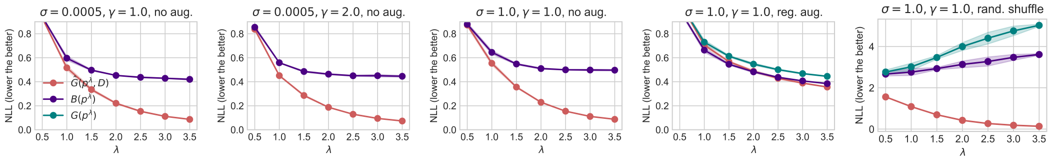
(b) Narrow prior and tempered softmax

(c) Standard prior and standard softmax

(d) Random crop and horizontal flip

(e) Pixels randomly shuffled

Figure 6: **Experimental illustrations for the arguments in Sections 4 and 5 using large CNN via SGLD on Fashion-MNIST.** The experiment setup is similar to the setups in Figure 5 but with a large CNN. Please refer to Appendix C for further details on the model.



(a) Narrow prior and standard softmax

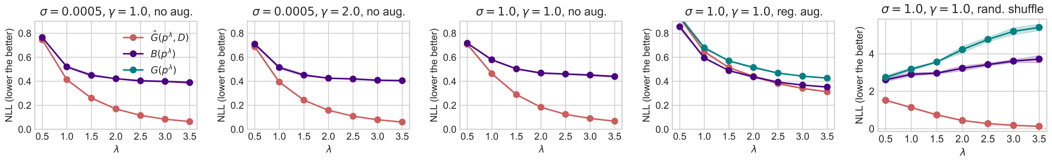
(b) Narrow prior and tempered softmax

(c) Standard prior and standard softmax

(d) Random crop and horizontal flip

(e) Pixels randomly shuffled

Figure 7: **Experimental illustrations for the arguments in Sections 4 and 5 using ResNet-18 via SGLD on CIFAR-10.** Figures 7a to 7c illustrate the arguments in Section 4, while Figures 7c to 7e illustrate the arguments in Section 5. Figure 7c uses the standard prior ($\sigma = 1$) and the standard softmax ($\gamma = 1$) for the likelihood without applying DA. Figure 7a follows a similar setup except for using a narrow prior. Figure 7b uses a narrow prior as in Figure 7a but with a tempered softmax that results in a lower aleatoric uncertainty. Figure 7d follows the setup as in Figure 7c but with standard DA methods applied, while Figure 7e uses fabricated DA. We report the training loss $\hat{G}(p^\lambda, D)$ and the testing losses $B(p^\lambda)$ and $G(p^\lambda)$ from 10 samples of ResNet-18 via Stochastic Gradient Langevin Dynamics (SGLD). We show the mean and standard deviation across three different seeds.



(a) Narrow prior and standard softmax

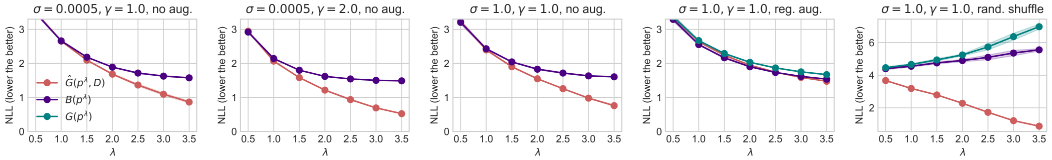
(b) Narrow prior and tempered softmax

(c) Standard prior and standard softmax

(d) Random crop and horizontal flip

(e) Pixels randomly shuffled

Figure 8: **Experimental illustrations for the arguments in Sections 4 and 5 using ResNet-50 via SGLD on CIFAR-10.** The experiment setup is similar to the setups in Figure 7 but with ResNet-50.



(a) Narrow prior and standard softmax

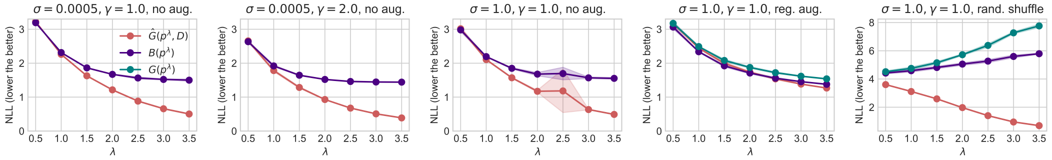
(b) Narrow prior and tempered softmax

(c) Standard prior and standard softmax

(d) Random crop and horizontal flip

(e) Pixels randomly shuffled

Figure 9: **Experimental illustrations for the arguments in Sections 4 and 5 using ResNet-18 via SGLD on CIFAR-100.** Figures 9a to 9c illustrate the arguments in Section 4, while Figures 9c to 9e illustrate the arguments in Section 5. Figure 9c uses the standard prior ($\sigma = 1$) and the standard softmax ($\gamma = 1$) for the likelihood without applying DA. Figure 9a follows a similar setup except for using a narrow prior. Figure 9b uses a narrow prior as in Figure 9a but with a tempered softmax that results in a lower aleatoric uncertainty. Figure 9d follows the setup as in Figure 9c but with standard DA methods applied, while Figure 9e uses fabricated DA. We report the training loss $\hat{G}(p^\lambda, D)$ and the testing losses $B(p^\lambda)$ and $G(p^\lambda)$ from 10 samples of ResNet-18 via Stochastic Gradient Langevin Dynamics (SGLD). We show the mean and standard deviation across three different seeds. For additional experimental details, please refer to Appendix C.



(a) Narrow prior and standard softmax

(b) Narrow prior and tempered softmax

(c) Standard prior and standard softmax

(d) Random crop and horizontal flip

(e) Pixels randomly shuffled

Figure 10: **Experimental illustrations for the arguments in Sections 4 and 5 using ResNet-50 via SGLD on CIFAR-100.** The experiment setup is similar to the setups in Figure 9 but with ResNet-50.

C.2.2 MEAN-FIELD VARIATIONAL INFERENCE (MFVI)

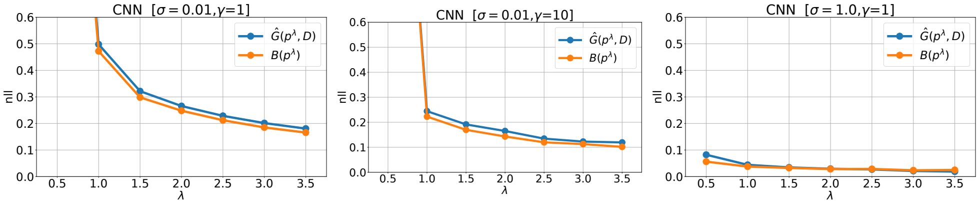
Experimental Settings: These experiments were run using Tensorflow (Abadi et al., 2015), Tensorflow Probability (Dillon et al., 2017) and Keras (Chollet et al., 2015). By default, we use zero-center Normal distributions, $\mathcal{N}(0, \sigma)$, as priors with different standard deviations, i.e., σ values. For the variational approximation, we use fully factorized Normal distributions, where both the mean and the standard deviation of each of them were the parameters to be learned by the variational algorithm. Although using an over-simplified family to approximate the true posterior, MFVI seems to achieve competitive results (Zhang and Nalisnick, 2021) compared to SGLD.

The convolutional neural network used for this experiment is a variational implementation of the network described above. This variational model uses a total of 1091092 parameters, double the number of parameters of the original model.

We use an Adam optimizer with a default learning rate 0.001, batch size = 100, and run during 100 epochs, which in our case, is enough to achieve convergence. The Keras global seed was set to 15. Other seeds were set, but similar results were obtained. Experiments were performed on Google Colab on a NVIDIA T4 GPU. The computation time was in the order of a few hours.

Prior Misspecification, Likelihood Misspecification and the CPE:

We run a similar experiment to the one reported in Figure 3 but using MFVI (Blei et al., 2017) as an approximate inference technique. The results of this experiment are reported in Figure 11. The conclusions are completely similar to the ones already discussed in Section 4.



(a) Baseline: "narrow" prior + standard softmax likelihood

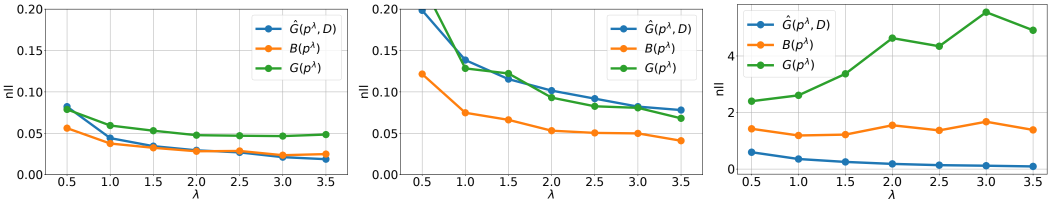
(b) "Narrow" prior + tempered softmax likelihood

(c) Standard prior + standard softmax likelihood

Figure 11: **CPE can be mitigated by a less misspecified model (Figure 11b) or imposing a less regularizing prior (Figure 11c).** We plot the training loss $\hat{G}(p^\lambda, D)$ and the testing loss $B(p^\lambda)$ with different priors and likelihood models. The parameter σ is the standard deviation of the isotropic Gaussian prior centered at zero, while the parameter γ serves as a smoothing parameter on the logits. All metrics are approximated using 10 samples drawn from the MFVI posterior.

Data Augmentation (DA) and the CPE:

As in the previous case, we ran a similar experiment to the one reported in Figure 3 but using MFVI (Blei et al., 2017) as an approximate inference technique. The results of this experiment are reported in Figure 12. The conclusions are very similar to the ones already discussed in Section 5.



(a) No augmentation

(b) Random crop and horizontal flip

(c) Pixels randomly shuffled

Figure 12: **CPE only occurs with "meaningful" augmentation (Figure 12b).** We plot the training loss $\hat{G}(p^\lambda, D)$ and the testing losses $B(p^\lambda)$ and $G(p^\lambda)$ with different augmentation methods. While Figure 11 shows no augmentation, Figure 12b and 12c show standard augmentation and an artificially designed "harmful" augmentation, where the pixels are shuffled randomly. All metrics are approximated using 10 samples drawn from the MFVI posterior.