A Proofs for Section 3

In this section, we provide the proofs for Section 3 in the following order. We first prove the derivative of the empirical Gibbs loss in Proposition 2. Then, we show in Proposition 7 that for meaningful posteriors (depends on training data), the derivative won't be zero. Before proving Proposition 3 and Theorem 4, we first provide Proposition 8, stating an alternative expression of the derivative of the Bayes loss. The proofs of Proposition 3 and Theorem 4 then follow from that.

A.1 Proof of Proposition 2

We first show a slightly more general result of $\frac{d}{d\lambda}\mathbb{E}_{p_\lambda}[f(\boldsymbol{\theta})]$ 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 $\frac{d}{d\lambda}\left(p(D|\boldsymbol{\theta})^\lambda p(\boldsymbol{\theta})\right) = \ln(p(D|\boldsymbol{\theta}))p(D|\boldsymbol{\theta})^\lambda p(\boldsymbol{\theta})$, the derivative

$$\frac{d}{d\lambda}\mathbb{E}_{p_\lambda}[f(\boldsymbol{\theta})] = \mathbb{E}_{p_\lambda}[\ln p(D|\boldsymbol{\theta})f(\boldsymbol{\theta})] - \mathbb{E}_{p_\lambda}[\ln p(D|\boldsymbol{\theta})]\mathbb{E}_{p_\lambda}[f(\boldsymbol{\theta})] = \mathrm{COV}_{p_\lambda}(\ln p(D|\boldsymbol{\theta}), f(\boldsymbol{\theta}))\,, \tag{19}$$

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

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

A.2 Proposition 7

Proposition 7. *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

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

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 8 that the derivative 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 8. *The derivative of the Bayes loss of the tempered posterior p_λ can be expressed by*

$$\frac{d}{d\lambda}B(p_\lambda) = \hat{G}(\bar{p}_\lambda, D) - \hat{G}(p_\lambda, D)\,. \tag{20}$$

Proof. By definition,

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

where

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

due to Equation 19. 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

$$\frac{d}{d\lambda}B(p_\lambda) = \mathbb{E}_{p_\lambda}[\ln p(D|\boldsymbol{\theta})] - \mathbb{E}_\nu[\mathbb{E}_{\tilde{p}_\lambda}[\ln p(D|\boldsymbol{\theta})]] = \mathbb{E}_{p_\lambda}[\ln p(D|\boldsymbol{\theta})] - \mathbb{E}_{\bar{p}_\lambda}[\ln p(D|\boldsymbol{\theta})], \tag{21}$$

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

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

The last expression in Equation 21 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 8 together with Definition 1 give that the CPE takes place if and only if

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

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 8 that

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

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

B Proofs for Section 4

B.1 Proof of Proposition 5

First of all, by the definition in Equation 2, and assuming a data-independent prior $p(\boldsymbol{\theta}|\boldsymbol{X}) = p(\boldsymbol{\theta})$, the tempered posterior is given by

$$p_\lambda(\boldsymbol{\theta}|\boldsymbol{X},\boldsymbol{Y}) \propto p(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{\theta})^\lambda p(\boldsymbol{\theta})\,,$$

where the tempered likelihood fully factorizes as $p(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{\theta})^\lambda = \prod_{(\boldsymbol{y},\boldsymbol{x})\in(\boldsymbol{Y},\boldsymbol{X})} p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})^\lambda$. Let a similar but $\boldsymbol{y}$-independent function $k(\boldsymbol{\theta},\boldsymbol{X},\lambda) = \prod_{\boldsymbol{x}\in\boldsymbol{X}} \int p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})^\lambda \, d\boldsymbol{y}$.

Therefore, $p(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{\theta})^\lambda p(\boldsymbol{\theta}) = \frac{p(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{\theta})^\lambda}{k(\boldsymbol{\theta},\boldsymbol{X},\lambda)} \left(k(\boldsymbol{\theta},\boldsymbol{X},\lambda)p(\boldsymbol{\theta})\right)$, where we can let the new prior

$$q(\boldsymbol{\theta}|\boldsymbol{X},\lambda) \propto p(\boldsymbol{\theta})k(\boldsymbol{\theta},\boldsymbol{X},\lambda) = p(\boldsymbol{\theta}) \prod_{\boldsymbol{x}\in\boldsymbol{X}} \int p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})^\lambda \, d\boldsymbol{y}\,,$$

and the new posterior

$$q(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{\theta},\lambda) = \frac{p(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{\theta})^\lambda}{k(\boldsymbol{\theta},\boldsymbol{X},\lambda)} = \frac{\prod_{(\boldsymbol{y},\boldsymbol{x})\in(\boldsymbol{Y},\boldsymbol{X})} p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})^\lambda}{\prod_{\boldsymbol{x}\in\boldsymbol{X}} \int p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})^\lambda \, d\boldsymbol{y}} = \prod_{(\boldsymbol{y},\boldsymbol{x})\in(\boldsymbol{Y},\boldsymbol{X})} q(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})\,.$$

B.2 Proof of Proposition 6

The proof is made using differential entropy, i.e. assuming continuous target values $\boldsymbol{y}$. The only assumption is that Leibniz integral rule holds for $q(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta},\lambda)\ln q(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta},\lambda))$, verifying that

$$\frac{d}{d\lambda}\int \left(q(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta},\lambda)\ln q(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta},\lambda)\right)\, d\boldsymbol{y} = \int \frac{d}{d\lambda}\left(q(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta},\lambda)\ln q(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta},\lambda)\right)\, d\boldsymbol{y}\,.$$

In the case of supervised classification problems, we adopt the Shanon entropy, where equality holds naturally

$$\frac{d}{d\lambda}\sum_{\boldsymbol{y}\in\mathcal{Y}}\left(q(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta},\lambda)\ln q(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta},\lambda)\right) = \sum_{\boldsymbol{y}\in\mathcal{Y}}\frac{d}{d\lambda}\left(q(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta},\lambda)\ln q(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta},\lambda)\right).$$

From the definition of differential entropy, we got that

$$H(q(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta},\lambda)) = -\int q(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta},\lambda)\ln q(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta},\lambda)\, d\boldsymbol{y}\,.$$

Thus, taking derivative w.r.t. λ and exchanging derivative and integral leads to the following expression

$$\frac{d}{d\lambda}H(q(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta},\lambda)) = -\int \frac{d}{d\lambda}\left(q(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta},\lambda)\ln q(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta},\lambda)\right)\, d\boldsymbol{y} = -\int (\ln q(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta},\lambda)+1)\frac{d}{d\lambda}q(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta},\lambda)\, d\boldsymbol{y}\,.$$

Using that $\int \frac{d}{d\lambda}q(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta},\lambda)d\boldsymbol{y} = \frac{d}{d\lambda}\int q(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta},\lambda)d\boldsymbol{y} = 0$, simplifies the expression as

$$\frac{d}{d\lambda}H(q(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta},\lambda)) = -\int \ln q(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta},\lambda)\frac{d}{d\lambda}q(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta},\lambda)\, d\boldsymbol{y}\,.$$

Let us consider now the second term inside the integral. Using the derivative of the quotient rule leads to the following:

$$\frac{d}{d\lambda}q(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta},\lambda) = \frac{d}{d\lambda}\frac{p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})^\lambda}{\int p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})^\lambda\, d\boldsymbol{y}} = \frac{p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})^\lambda \ln p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})}{\int p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})^\lambda\, d\boldsymbol{y}} - \frac{p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})^\lambda \int p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})^\lambda \ln p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})\, d\boldsymbol{y}}{(\int p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})^\lambda\, d\boldsymbol{y})^2}\,.$$

Where, using the definition of $q(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta},\lambda)$, we got that

$$\frac{p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})^\lambda \ln p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})}{\int p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})^\lambda\, d\boldsymbol{y}} = q(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta},\lambda)\ln p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})\,,$$

and

$$\frac{p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})^\lambda \int p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})^\lambda \ln p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})\ d\boldsymbol{y}}{(\int p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})^\lambda\ d\boldsymbol{y})^2} = q(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta},\lambda)\int q(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta},\lambda)\ln p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})\ d\boldsymbol{y}$$
$$= q(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta},\lambda)\mathbb{E}_q[\ln p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})]\,.$$

As a result, we got that

$$\int \ln q(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta},\lambda)\frac{d}{d\lambda}q(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta},\lambda)\ d\boldsymbol{y} = \mathbb{E}_q[\ln p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})\ln q(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta},\lambda)] - \mathbb{E}_q[\ln q(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta},\lambda)]\mathbb{E}_q[\ln p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})]$$

Using $q(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta},\lambda)$ definition again:

$$\int \ln q(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta},\lambda)\frac{d}{d\lambda}q(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta},\lambda)\ d\boldsymbol{y} = \mathbb{E}_q[\ln p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})\ln\frac{p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})^\lambda}{\int p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})^\lambda}] - \mathbb{E}_q[\ln\frac{p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})^\lambda}{\int p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})^\lambda}]\mathbb{E}_q[\ln p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})]$$

Where, expanding the logarithms the denominators cancel each other, leading to

$$\int \ln q(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta},\lambda)\frac{d}{d\lambda}q(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta},\lambda)\ d\boldsymbol{y} = \mathbb{E}_q[\ln p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})\ln p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})^\lambda] - \mathbb{E}_q[\ln p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})^\lambda]\mathbb{E}_q[\ln p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})]$$
$$= \lambda\mathbb{V}(\ln p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})) \geq 0$$

As a result, the entropy is negative.

C Proofs for Section 6

C.1 Proof of Equation 14

Note that

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

where the second equality is by applying Equation 19. By taking $\lambda = 1$, we obtain the desired derivative.

C.2 Proof of Equation 16

Recall from the proof of Theorem 8 that

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

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,

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

D Experiment details for Bayesian linear regression on synthetic data with exact inference

In this section we detail the settings of the toy experiment using synthetic data and exact Bayesian linear regression in Figure 2. We also show extra results of the derivative of Gibbs loss and Bayes loss w.r.t to λ approximated by samples.

To begin, we will outline the data-generating process for the synthetic data used in the experiment shown in Figure 2 and Figure 7. 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 2. 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.

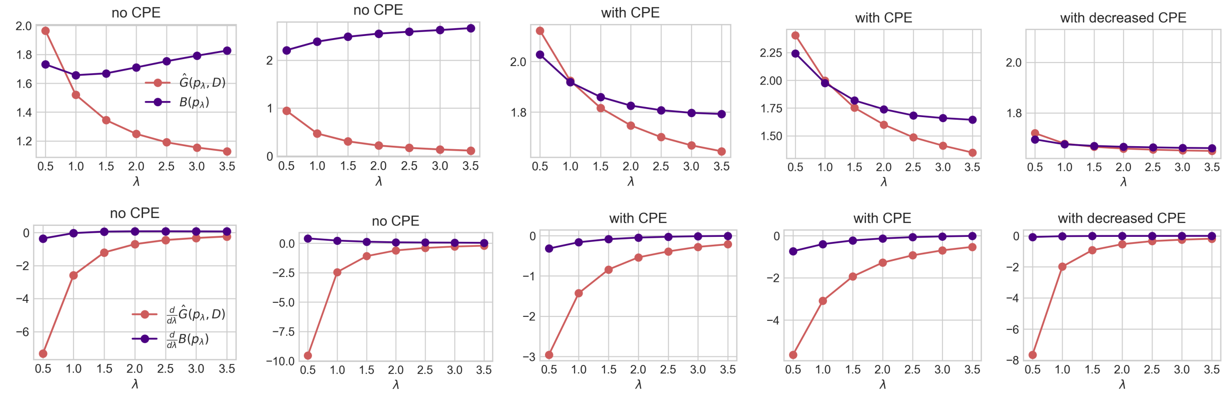
1. No misspecification: likelihood $p(y|\boldsymbol{x}, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}^T\boldsymbol{x}, 1.0)$, prior $p(\boldsymbol{\theta}) = \mathcal{N}(0, 2)$. This is the baseline for comparison.
2. Misspecified likelihood I: likelihood $p(y|\boldsymbol{x}, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}^T\boldsymbol{x}, 0.15)$ (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 $p(\boldsymbol{\theta}) = \mathcal{N}(0, 2)$. 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 2, where we show the Gibbs loss $\hat{G}(p_\lambda, D)$ (training) and the Bayes loss $B(p_\lambda)$ (testing) with respect to λ, we now show their derivatives $\frac{d}{d\lambda}\hat{G}(p_\lambda, D)$ (Equation 5) and $\frac{d}{d\lambda}B(p_\lambda)$ (Equation 20) respectively in Figure 7. 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 7, we could clearly see that the derivatives perfectly characterize the losses in all four settings.

E Experiment details for Bayesian neural networks on image data with approximate inference

In this section, we first present in Appendix E.1 the architectures of the small and large CNNs used in this paper. As promised in the main text, we then provide results on additional image datasets trained with



(a) No likelihood or prior misspecification (b) Misspecified likelihood I (c) Misspecified likelihood II (d) Only misspecified prior (e) Same as (c) but with 50 data points

Figure 7: **The derivatives $\frac{d}{d\lambda}\hat{G}(p_\lambda, D)$ (Equation 5) and $\frac{d}{d\lambda}B(p_\lambda)$ (Equation 20) characterize the Gibbs loss $\hat{G}(p_\lambda, D)$ and the Bayes loss $B(p_\lambda)$ perfectly.**

Stochastic Gradient Langevin Dynamics (SGLD) (Welling & Teh, 2011) in Appendix E.2. Lastly, we provide additional results using mean-field variational inference (MFVI) (Blei et al., 2017) on MNIST, where we observe that the results of MFVI align with the ones with SGLD.

E.1 Architectures of small/large CNN

Small CNN The small CNN is similar to LeNet-5, but with 107786 parameters in total:

1. Convolutional layer 1. Input channels: 1 (assuming grayscale images), output channels: 6, kernel size: 5x5, padding: 2, activation: ReLU.
2. Average pooling layer 1. Kernel size: 2x2, stride: 2.
3. Convolutional layer 2. Input channels: 6, output channels: 16, kernel size: 5x5, padding: 2, activation: ReLU.
4. Average pooling layer 2. Kernel size: 2x2, stride: 2.
5. Flattening layer. Flattens the output from the previous layers.
6. Fully connected layer 1. Input features: 784 (16 channels * 7 * 7), output features: 120, activation: ReLU.
7. Fully connected layer 2. Input features: 120, output features: 84, activation: ReLU.
8. Fully connected layer 3 (output layer). Input features: 84, output features: num_classes (specified during instantiation).

Large CNN The large CNN is similar to the small CNN, but with 545546 parameters in total:

1. Convolutional layer 1. Input channels: 1 (assuming grayscale images), output channels: 6, kernel size: 5x5, padding: 2, activation: ReLU.
2. Average pooling layer 1. Kernel size: 2x2, stride: 2.
3. Convolutional layer 2. Input channels: 6, output channels: 16, kernel size: 5x5, padding: 2, activation: ReLU.

4. Average pooling layer 2. Kernel size: 2x2, stride: 2.
5. Convolutional layer 3. Input channels: 16, output channels: 120, kernel size: 5x5, padding: 2, activation: ReLU.
6. Flattening layer. Flattens the output from the previous layers.
7. Fully connected layer 1. Input features: 5880 (120 channels × 7 × 7), output features: 84, activation: ReLU.
8. Fully connected layer 2 (output layer). Input features: 84, output features: num_classes (specified during instantiation).

In all the convolutional layers, no stride = 1 and padding is set to *same*.

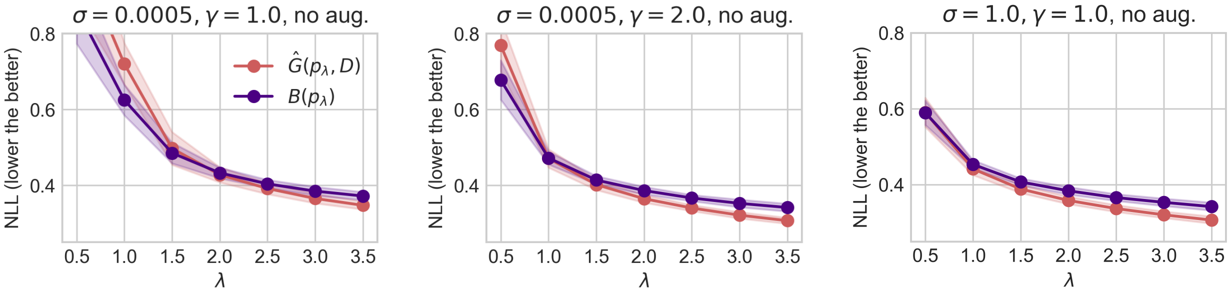
E.2 Stochastic Gradient Langevin Dynamics (SGLD)

Our experiments using SGLD are categorized into 4 groups:

1. Bayesian CNNs (small and large) on MNIST (Figures 3 - 6 in the main text)
2. Bayesian CNNs (small and large) on Fashion-MNIST (Appendix E.2.1)
3. Bayesian ResNets (18 and 50) on CIFAR-10 (Appendix E.2.2)
4. Bayesian ResNets (18 and 50) on CIFAR-100 (Appendix E.2.3)

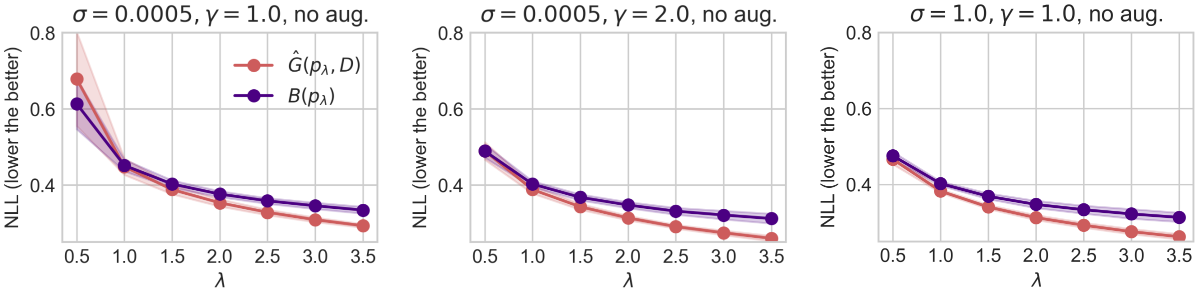
where each group evaluates the effect of underfitting on a small model and a large model. Note that as we follow the standard ResNet-18 and ResNet-50, the details of the architectures are omitted. They have around 11 million and 23 million parameters, respectively. We implement with PyTorch (Paszke et al., 2019) and 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. Experiments were conducted on NVIDIA A100 GPU, with each trial taking around 30 hours.

E.2.1 Small and Large CNNs via SGLD on Fashion-MNIST



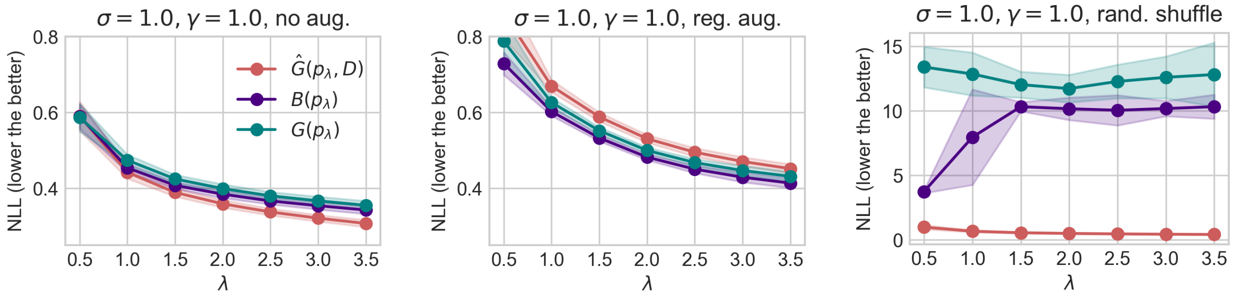
(a) Narrow prior and standard softmax (b) Narrow prior and tempered softmax (c) Standard prior and standard softmax

Figure 8: Extended results of Figure 3 using small CNN via SGLD on Fashion-MNIST.



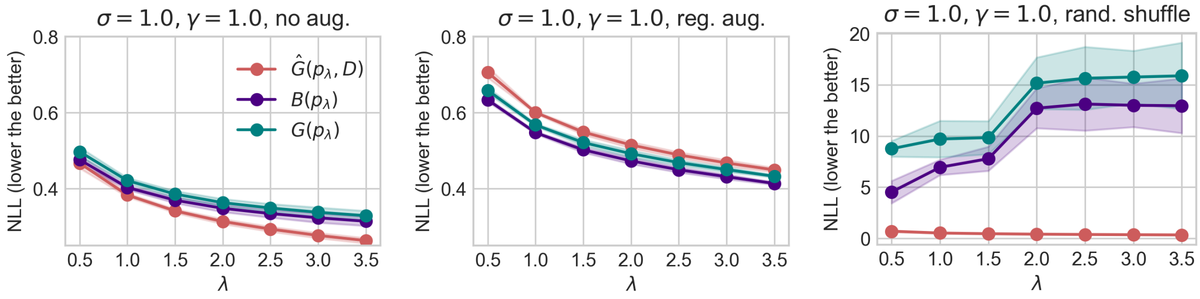
(a) Narrow prior and standard softmax (b) Narrow prior and tempered softmax (c) Standard prior and standard softmax

Figure 9: Extended results of Figure 4 using large CNN via SGLD on Fashion-MNIST.



(a) Standard prior and standard softmax (b) Random crop and horizontal flip (c) Pixels randomly shuffled

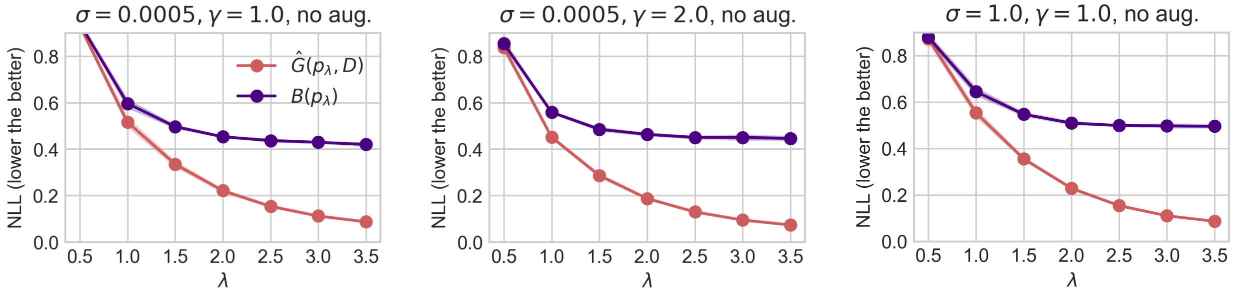
Figure 10: Extended results of Figure 5 using small CNN via SGLD on Fashion-MNIST.



(a) Standard prior and standard softmax (b) Random crop and horizontal flip (c) Pixels randomly shuffled

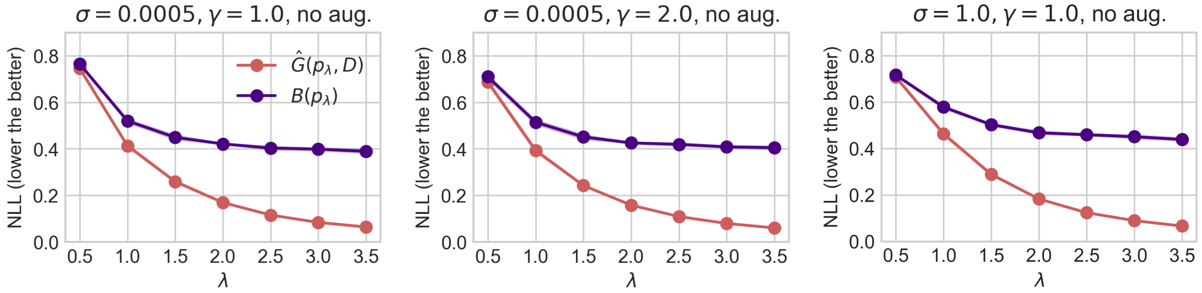
Figure 11: Extended results of Figure 6 using large CNN via SGLD on Fashion-MNIST.

E.2.2 ResNet-18 and ResNet-50 via SGLD on CIFAR-10



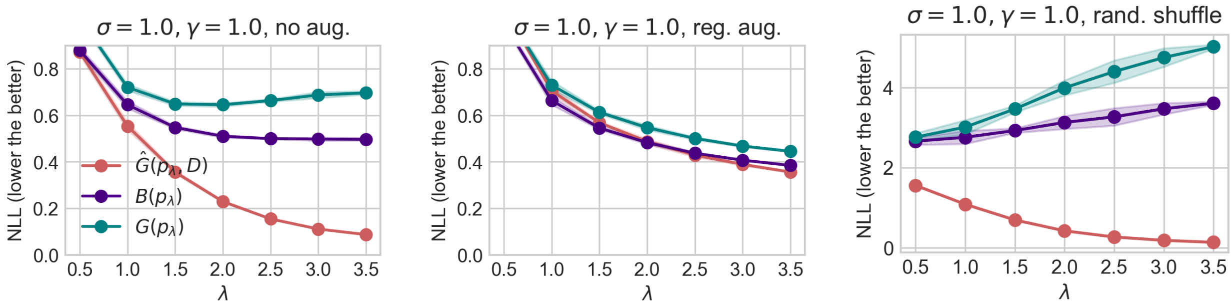
(a) Narrow prior and standard softmax (b) Narrow prior and tempered softmax (c) Standard prior and standard softmax

Figure 12: Extended results of Figure 3 using ResNet-18 via SGLD on CIFAR-10.



(a) Narrow prior and standard softmax (b) Narrow prior and tempered softmax (c) Standard prior and standard softmax

Figure 13: Extended results of Figure 4 using ResNet-50 via SGLD on CIFAR-10.

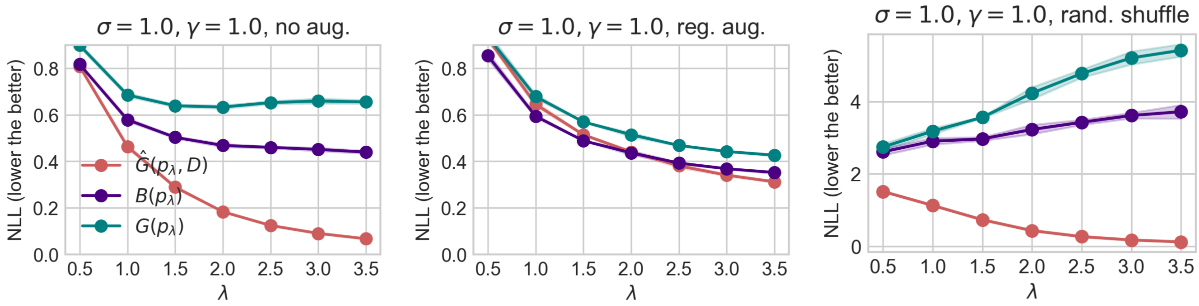


(a) Standard prior and standard softmax

(b) Random crop and horizontal flip

(c) Pixels randomly shuffled

Figure 14: Extended results of Figure 5 using ResNet-18 via SGLD on CIFAR-10.



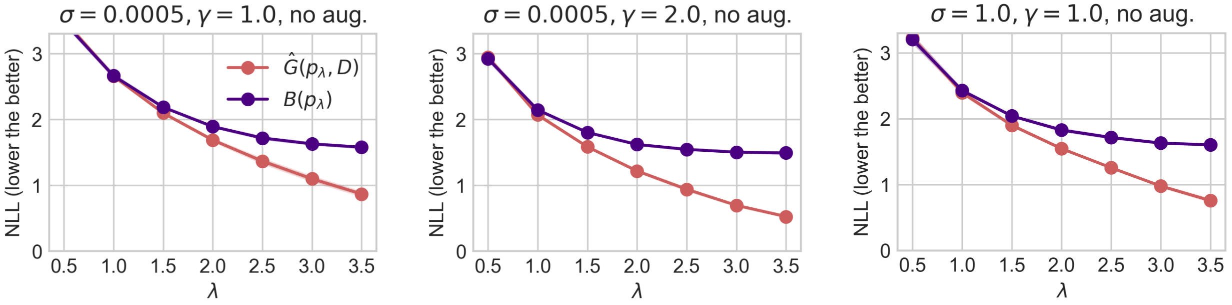
(a) Standard prior and standard softmax

(b) Random crop and horizontal flip

(c) Pixels randomly shuffled

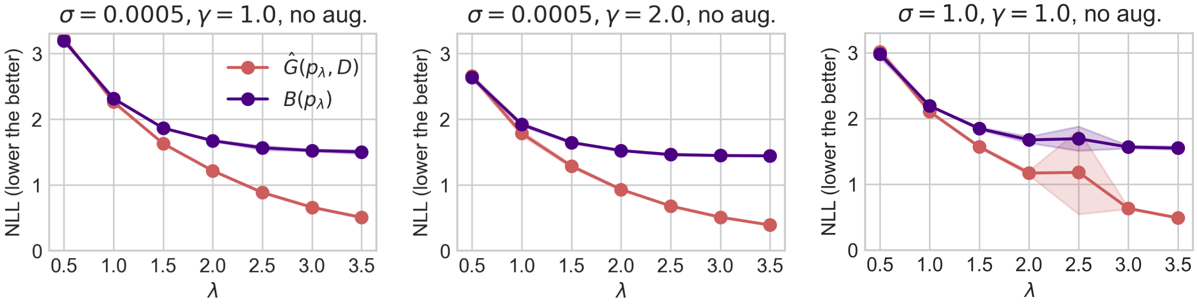
Figure 15: Extended results of Figure 6 using ResNet-50 via SGLD on CIFAR-10.

E.2.3 ResNet-18 and ResNet-50 via SGLD on CIFAR-100



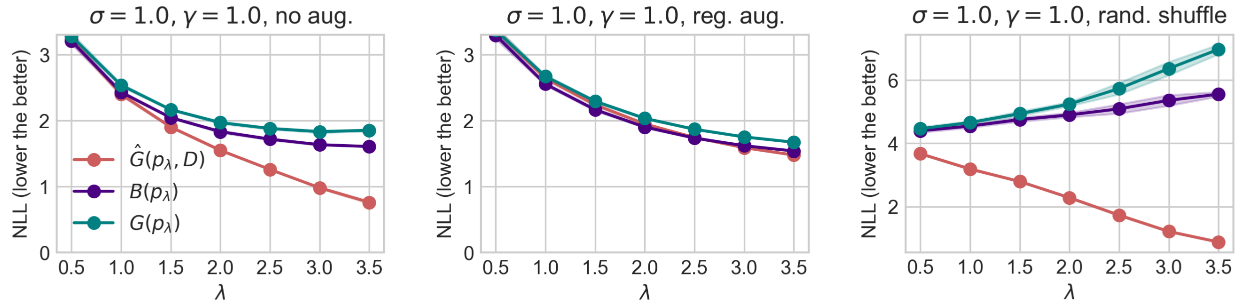
(a) Narrow prior and standard softmax (b) Narrow prior and tempered softmax (c) Standard prior and standard softmax

Figure 16: Extended results of Figure 3 using ResNet-18 via SGLD on CIFAR-100.



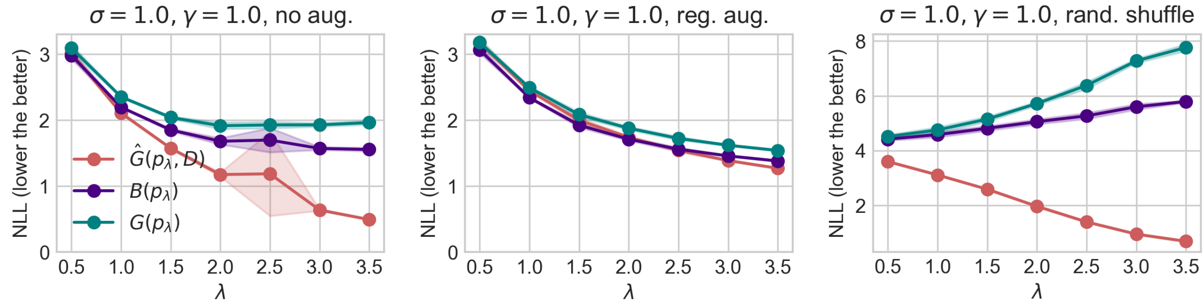
(a) Narrow prior and standard softmax (b) Narrow prior and tempered softmax (c) Standard prior and standard softmax

Figure 17: Extended results of Figure 4 using ResNet-50 via SGLD on CIFAR-100.



(a) Standard prior and standard softmax (b) Random crop and horizontal flip (c) Pixels randomly shuffled

Figure 18: Extended results of Figure 5 using ResNet-18 via SGLD on CIFAR-100.



(a) Standard prior and standard softmax (b) Random crop and horizontal flip (c) Pixels randomly shuffled

Figure 19: Extended results of Figure 6 using ResNet-50 via SGLD on CIFAR-100.

E.3 Mean-Field Variational Inference (MFVI)

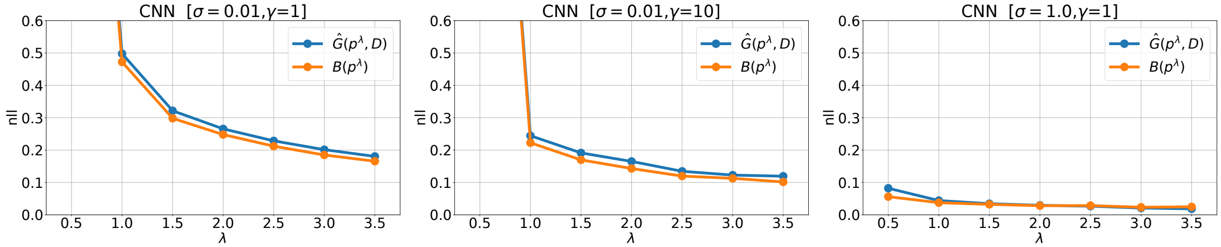
Experimental Settings: These experiments were run using Tensorflow (Abadi et al., 2015), Tensorflow Probability (Dillon et al., 2017) and Keras (Chollet et al., 2015). By default, we use zero-center Normal distributions, $\mathcal{N}(0, \sigma)$, as priors with different standard deviations, i.e., σ 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 also achieves competitive results (Zhang & 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 4 but using MFVI (Blei et al., 2017) as an approximate inference technique. The results of this experiment are reported in Figure 20. The conclusions are completely similar to the ones already discussed in Section 5.



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

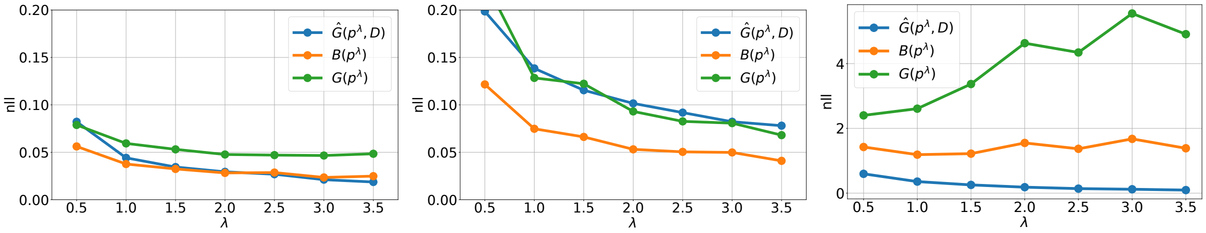
(b) "Narrow" prior + tempered softmax likelihood

(c) Standard prior + standard softmax likelihood

Figure 20: **CPE can be mitigated by a less misspecified model (Figure 20b) or imposing a less regularizing prior (Figure 20c).** 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 4 but using MFVI (Blei et al., 2017) as an approximate inference technique. The results of this experiment are reported in Figure 21. The conclusions are very similar to the ones already discussed in Section 6.



(a) No augmentation (b) Rand. crop and horiz. flip (c) Pixels randomly shuffled

Figure 21: **CPE only occurs with "meaningful" augmentation (Figure 21b).** 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 20 shows no augmentation, Figure 21b and 21c 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.