Reparameterized Importance Sampling for Robust Variational Bayesian Neural Networks

Yunfei Long [*1] **Zilin Tian** [*1] **Liguo Zhang** [1 2] **Huosheng Xu** [1]

Abstract

Mean-field variational inference (MFVI) methods provide computationally cheap approximations to the posterior of Bayesian Neural Networks (BNNs) when compared to alternatives like MCMC. However, applying MFVI to BNNs encounters limitations due to the Monte Carlo sampling problem. This problem stems from two main issues. *First*, most samples do not accurately represent the most probable weights. *Second*, random sampling from variational distributions introduces high variance in gradient estimates, which can hinder the optimization process, leading to slow convergence or even failure. In this paper, we introduce a novel sampling method called *Reparameterized Importance Sampling* (RIS) to estimate the first moment in neural networks, reducing variance during feed-forward propagation. We begin by analyzing the generalized form of the optimal proposal distribution and presenting an inexpensive approximation. Next, we describe the sampling process from the proposal distribution as a transformation that combines exogenous randomness with the variational parameters. Our experimental results demonstrate the effectiveness of the proposed RIS method in three critical aspects: improved convergence, enhanced predictive performance, and successful uncertainty estimation for out-of-distribution data.

1. Introduction

Bayesian Neural Networks (BNNs) can perform probabilistic predictions by incorporating a prior distribution on the network weights and then inferring a posterior distribution using Bayes' rule. This characteristic empowers BNNs to estimate the level of uncertainty in their predictions, which is a crucial consideration, especially in safety-critical systems. However, due to the computational intractability of exact inference for posteriors, approximate methods are commonly employed in practice. Mean-field variational inference (MFVI) (Blundell et al., 2015; Graves, 2011; Kingma et al., 2015; Dusenberry et al., 2020; Coker et al., 2022) stands out as a powerful paradigm for approximating the Bayesian posterior with flexible variational distributions. The optimization of this approximation revolves around minimizing the Kullback-Leibler (KL) divergence between the variational distribution and the actual posterior. To review, let $\mathcal{W} = \{w_i\}$ represent network weights, D represent the data, and place prior $p(\mathcal{W})$ on weights $\mathcal{W}$. MFVI introduces a variational distribution $\mathcal{Q}$ with variational parameters $\Theta = \{\theta_i\}$ to approximate the posterior distribution $p(\mathcal{W}|D)$. This variational distribution is factorized, $\mathcal{Q}(\mathcal{W}|\Theta) = \prod_i q(w_i|\theta_i)$, and the values of Θ are optimized by minimizing the following objective function.

$$\begin{aligned}\mathcal{L}(D,\Theta) =& \mathbb{E}_{\mathcal{Q}}\left[\log \mathcal{Q}(\mathcal{W}|\Theta) - \log p(\mathcal{W})\right] \\ &- \mathbb{E}_{\mathcal{Q}}\left[\log p(D|\mathcal{W})\right]\end{aligned} \tag{1}$$

where the first term serves as a regularizer, penalizing solutions where the posterior deviates from the prior, while the second term represents a model-fitting component. It is straightforward to verify that analytic results are available for the KL term between the $\mathcal{Q}$ and $p(\mathcal{W})$. However, due to the nonlinearity inherent in neural networks, computing the exact expectation of log-likelihood is unfeasible. MFVI employs Monte Carlo sampling to approximate this expectation, $\mathbb{E}_{\mathcal{Q}}\left[\log p(D|\mathcal{W})\right] \approx \frac{1}{M}\sum_{m=1}^{M} p\left(D|\mathcal{W}^{(m)}\right)$, M represent the sample size, $\mathcal{W}^{(m)}$ denotes the m-th Monte Carlo sample drawn from the variational posterior $\mathcal{Q}(\mathcal{W}|\Theta)$. Unfortunately, despite the advantages related to computational tractability, training BNNs using MFVI can be challenging primarily due to the Monte Carlo sampling problem. *Firstly*, the majority of weights sampled from variational distributions tend to fall in edge regions, far from the most probable weight. *Secondly*, variational BNNs optimize Θ using stochastic gradient descent methods that rely on the gradient of the Monte Carlo $\mathcal{L}$ estimate with respect to Θ.

[*]Equal contribution [1]College Of Computer Science And Technology, Harbin Engineering University, Harbin, Heilongjiang, China [2]Modeling and Emulation in E-Government National Engineering Laboratory, Harbin Engineering University, Harbin, Heilongjiang, China. Correspondence to: Liguo Zhang <zhangliguo@hrbeu.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

The Monte Carlo generating process of weights is the source of stochasticity in the gradient estimation process. High variance in the gradient estimators can impede or prevent the optimization process from progressing smoothly. These challenges become more pronounced when working with larger networks. This work has focused on the sampling of variational distributions and the optimization of variational parameters.

We consider the task of enhancing the robustness of BNNs by propagating the first moment during the feed-forward inference procedure. Using this deterministic approximation inference can effectively decrease gradient variance, but, unlike the complex moment estimate in Deterministic Variational Inference (DVI), we investigate employing Importance Sampling for estimating the first moment. This scheme offers three distinct advantages: i) It allows for the attainment of a more accurate estimator with only a small number of samples. ii) It eliminates all stochasticity resulting from Monte Carlo sampling. iii) It effectively reduces gradient variance while preserving the capacity to model uncertainty. To accomplish these benefits, we introduce the *Reparameterized Importance Sampling method*. Specifically, we develop a computationally cheap approximation for the optimal proposal distribution in Importance Sampling. This approximation relies on an analytical linear estimation of both its mean and variance. Subsequently, we merge the Importance Sampling method with the reparameterization trick to facilitate the efficient sampling of weights from the proposal distribution and simplify the computation of gradients for variational parameters. Applying this approximation to variational inference in multiple neural networks, we observe faster convergence, more stable optimization traces, and improved predictive performance compared to MFVI using Monte Carlo sampling.

2. Background

Standard Mean-Field Variational Inference (MFVI) combines Monte Carlo sampling and the *reparameterization trick* to estimate the posterior distribution of Bayesian Neural Networks (BNNs). We begin by considering a feed-forward BNN with multiple hidden layers. For notational clarity, the factorized distribution on $\mathcal{W}$ is redefined as $\mathcal{Q}(\mathcal{W}|\Theta) = \prod_l q(w_l|\theta_l)$, where θ_l are the variational parameters of the q_l distribution, w_l represents the weights of the l-th layer that is considered independent from weights in other layers. Its variational posterior distribution, q_l, is typically represented by a diagonal Gaussian distribution, denoted as $\mathcal{N}\left(u_l, \sigma_l^2\right)$, with parameters $\theta_l = \left[u_l, \sigma_l^2\right]$. The reparameterization trick transforms the sampling procedure that generates weights w_l from $q(w_l|\theta_l)$ as a differentiable mapping t.

$$w_l = t(\epsilon; \theta_l) = \mu_l + \sigma_l^2 \odot \epsilon, \epsilon \sim N(0, I) \tag{2}$$

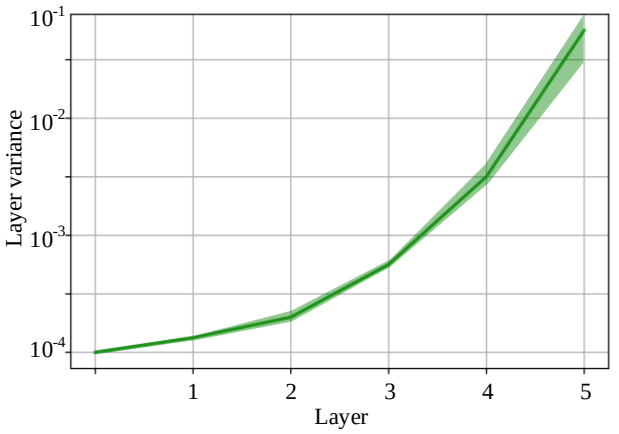


Figure 1. The variance in feed-forward neural network becomes higher as the number of layers increases.

where $\odot$ represents element-wise product, ϵ is independent of θ_l. For a single weight sampling, the gradient of variational parameters can be calculated by:

$$\nabla_{\theta_l} \triangleq \underbrace{-\nabla_{\theta_l}^{lg}\left[\log p(D|\mathcal{W})\right]}_{likelihood\ gradient} + \underbrace{\nabla_{\theta_l}^{rg}\left[\log\left(q(w_l|\theta_l) - p(w_l)\right)\right]}_{regularizer\ gradient}$$

For simplicity, we focus on the situation for w_l. Similar to the form of $\mathcal{L}$ (Eq. 1), the gradient ∇_{θ_l} consists of two terms including *likelihood gradient* and *regularizer gradient*.

Regularizer gradient We analyze the *regularizer gradient* of θ_l by decomposing it into the following form:

$$\nabla_{\theta_l}^{rg}\left[\log q(w_l|\theta_l) - p(w_l)\right] = \frac{\partial \log q(w_l|\theta_l)}{\partial w_l}\frac{\partial w_l}{\partial \theta_l} + \frac{\partial \log q(w_l|\theta_l)}{\partial \theta_l} - \frac{\partial \log p(w_l)}{\partial w_l}\frac{\partial w_l}{\partial \theta_l}$$

It is straightforward to calculate this gradient by taking the partial derivative of the log probability density function and differentiable mapping t. If we initialize the prior $p(w_l)$ as a diagonal Gaussian distribution with zero mean and unit variance, the regularizer gradients of u_l and σ_l^2 are given following:

$$\nabla_{\mu_l}^{rg}\left[\log q(w_l|\theta_l) - p(w_l)\right] = \epsilon \tag{3}$$

$$\nabla_{\sigma_l}^{rg}\left[\log q(w_l|\theta_l) - p(w_l)\right] \propto \epsilon^2 \tag{4}$$

Likelihood gradient The randomness in the estimate of the *regularizer gradient* is determined by the ϵ samples. Instead, the variance of likelihood gradient depends on

$\frac{\partial \log p(\mathcal{D}|\mathcal{W})}{\partial w_l}$, which is a multiplicative term over the gradient.

$$\nabla_{\mu_l}^{lg}\left[\log p\left(D|\mathcal{W}\right)\right] = \frac{\partial \log p\left(\mathcal{D}|\mathcal{W}\right)}{\partial w_l} \tag{5}$$

$$\nabla_{\sigma_l}^{lg}\left[\log p\left(D|\mathcal{W}\right)\right] \propto \frac{\partial \log p\left(\mathcal{D}|\mathcal{W}\right)}{\partial w_l}\epsilon \tag{6}$$

The variance of the back-propagating gradients $\frac{\partial \log p(\mathcal{D}|\mathcal{W})}{\partial w_l}$ will become higher as the number of layers increases. Comparing above Eqs. (3), (4), (5), (6), we empirically argue that the significant difference in gradient variance is a contributing factor to optimization prioritizing the proximity of variational posteriors to priors while overlooking critical model fit terms. Figure 1 shows the outputs variance becomes higher as the number of layers increases. In this work, we aim to reduce variance in feed-forward MFVI to reduce the gradients variance.

3. Moment Propagation in Mean-field Variational Inference

In a feed-forward BNN, the activation action of lth layer that maps z^{l-1} to z^l can be expressed as follows:

$$z_l = \delta(w_l z_{l-1}), w_l \sim q_l(w_l|\theta_l)^1 \tag{7}$$

where δ represents a non-linearity function (e.g. ReLU), w_l represents the weights of the l-th layer that are drawn from the variational distribution $q_l(w_l|\theta_l)$. The DVI(Wu et al., 2019) argues a Gaussian distribution for z_l and derives an approximate expression for calculating its first and second moments. This deterministic approximation to variational inference removes all stochasticity due to Monte Carlo sampling and reduces gradient variance to zero. In this work, we compute the likelihood value by sequentially computing and propagating the first moment estimate in the hidden layers.

$$\tilde{z}_l = \mathbb{E}_{w_l \sim q_l}\left[\delta\left(w_l \tilde{z}_{l-1}\right)\right]$$

In general, this expectation is approximated by using Monte Carlo samples from the variational distribution:

$$\tilde{z}_l \approx \frac{1}{M}\sum_{m=1}^{M} \delta\left(w_l^m \tilde{z}_{l-1}\right), w_l^m \sim q_l(w_l|\theta_l)$$

where M is the number of Monte Carlo samples. However, this approximation generally suffers from high-variance.

Importance Sampling (Tokdar & Kass, 2010; Borcherds, 2000) is an essential alternative to Monte Carlo sampling that highly reduces the estimate expectation variance. In general, we can approximate $\tilde{z}_l$ through following two steps.

[1]We introduce the forward propagation without bias and show in the Appendix that the case with bias is identical to this case.

Firstly, we sample weights w_l from the proposal distribution $r_l\left(w_l\right)$ and perform non-linear calculations. *Secondly*, we calculate importance weight as a correction for variance. We can rewrite $\tilde{z}_l$ in the following expectation form:

$$\begin{aligned}\tilde{z}_l &= \mathbb{E}_{r_l}\left[\frac{q\left(w_l|\theta_l\right)}{r_l\left(w_l\right)}\delta\left(w_l\tilde{z}_{l-1}\right)\right] \\ &\approx \frac{1}{M}\sum_{m=1}^{M}\frac{q\left(w_l^m|\theta_l\right)}{r_l\left(w_l^m\right)}\delta\left(w_l^m\tilde{z}_{l-1}\right), w_l^m \sim r_l\left(w_l\right)\end{aligned} \tag{8}$$

The importance sampling performs well when an optimal proposal distribution is used. A reasonable criterion for choosing a proposal distribution is to opt for one that minimizes the variance of $\tilde{z}_l$.

$$var_{r_l}\left[f\left(w_l\right)\gamma_l\right] = \mathbb{E}_{r_l}\left[f^2\left(w_l\right)\gamma_l^2\right] - \left[\mathbb{E}_{q_l}\left[f\left(w_l\right)\right]\right]^2$$

where $f\left(w_l\right) = \delta\left(w_l\tilde{z}_{l-1}\right), \gamma_l = \frac{q(w_l|\theta_l)}{r_l(w_l)}$ represents the importance weight.

Proposition 1. We can obtain the lower bound of $var_{r_l}\left[f\left(w_l\right)\gamma_l\right]$ when

$$r_l^*\left(w_l\right) \propto \left|f\left(w_l\right)\right| q_l\left(w_l\right) \tag{9}$$

Proof. As:

$$\begin{aligned}\mathbb{E}_{rl}\left[f\left(w_l\right)^2\gamma_l^2\right] &\geqslant \left[\mathbb{E}_{rl}\left|f\left(w_l\right)\gamma_l\right|\right]^2 \\ &= \left(\int\left|f\left(w_l\right)\right|q_l\left(w_l\right)dw_l\right)^2\end{aligned}$$

For $r_l = r_l^*$, we have

$$\begin{aligned}\mathbb{E}_{r_l^*}\left[f\left(w_l\right)^2\gamma_l^2\right] &= \int\frac{f\left(w_l\right)^2 q\left(w_l\right)^2}{\left|f\left(w_l\right)\right|q\left(w_l\right)}dw_l\int\left|f\left(w_l\right)\right|q(w_l)dw_l \\ &= \left(\int\left|f\left(w_l\right)\right|q\left(w_l\right)dw_l\right)^2\end{aligned}$$

Notably, $var_{r_l^*}\left[f\left(w_l\right)\gamma_l\right] = 0$ when $f(w_l) \geqslant 0$ for all w_l. In this case, $r_l^*\left(w_l\right) = f\left(w_l\right)q_l\left(w_l\right)/\tilde{z}_l$, where $\tilde{z}_l = \int f\left(w_l\right)q_l\left(w_l\right)dw_l = \mathbb{E}_{q_l}\left[f\left(w_l\right)\right]$, and it has that:

$$var_{r_l^*}\left[f\left(w_l\right)\gamma_l\right] = var_{r_l^*}\left[\frac{f\left(w_l\right)q_l\left(w_l\right)}{f\left(w_l\right)q_l\left(w_l\right)}\tilde{z}_l\right] = 0$$

Unfortunately, calculating the optimal proposal distribution can be quite challenging, primarily because of the intractable integral involving non-linear functions $|f\left(w_l\right)|$. In the next section, our objective is to create a readily practical and highly accurate approximation for this optimal proposal distribution.

4. Reparameterized Importance Sampling

4.1. Approximating the Optimal Proposal Distribution

Having established that the optimal proposal distribution $r_l^*(w_l)$ follows the form given by Eq. (9), we empirically constrain it to a multivariate Gaussian distribution with a mean of $\overset{*}{u}$ and diagonal variance of $\overset{*}{\sigma}^2$, according to the central limit theorem. Referring to the definitions of mean and variance, we can derive:

$$\overset{*}{\mu} = \mathbb{E}_{r_l}\left[w_l\right], \overset{*}{\sigma}^2 = \mathbb{E}_{r_l}\left[w_l^2\right] - \left[\mathbb{E}_{r_l}\left[w_l\right]\right]^2 \tag{10}$$

Combing Eqs. (9) (10) leads to a new transformation as:

$$\begin{aligned}\overset{*}{\mu} &= \int q\left(w_l|\theta_l\right)\left|f\left(w_l\right)\right| w_l dw_l \\ &= \mathbb{E}_{q_l}\left[\left|f\left(w_l\right)\right| w_l\right]\end{aligned} \tag{11}$$

$$\overset{*}{\sigma}^2 = \mathbb{E}_{q_l}\left[\left(\left|f\left(w_l\right)\right| w_l\right)^2\right] - \left[\mathbb{E}_{q_l}\left[\left|f\left(w_l\right)\right| w_l\right]\right]^2 \tag{12}$$

In general, exact computation of the above expressions is not feasible due to the intractable integrals involved. Therefore, we introduce practical approximations for these expressions. To simplify notation, we define all the non-linear functions related to w_l as $h(w_l)$. Moreover, the function $h(w_l)$ is certainly differentiable with respect to w_l, even in the presence of an absolute value function. We then linearize h about some value u_l

$$\tilde{h}\left(w_l\right) = h\left(\mu_l\right) + h'\left(\mu_l\right)\left(w_l - \mu_l\right) + \frac{h''\left(\mu_l\right)}{2}\left(w_l - \mu_l\right)^2 \tag{13}$$

where we have employed the Taylor series to approximate the non-linear function h, h' and h'' refer to the first and the second-order derivative of $h(w_l)$ with respect to w_l, both evaluated at u_l. Recalling the *reparameterization trick*, we can perceive it as a transformation of the random variable ϵ while keeping θ_l fixed.

$$\begin{aligned}\tilde{h}_{\theta_l}\left(\epsilon\right) &= h\left(\mu_l\right) + h'\left(\mu_l\right)\left(t\left(\epsilon;\theta_l\right) - \mu_l\right) \\ &\quad + \frac{h''\left(\mu_l\right)}{2}\left(t\left(\epsilon;\theta_l\right) - \mu_l\right)^2 \\ &= h\left(\mu_l\right) + h'\left(\mu_l\right)\sigma_l^2 \odot \epsilon + \frac{h''\left(\mu_l\right)}{2}\left(\sigma_l^2 \odot \epsilon\right)^2\end{aligned} \tag{14}$$

Crucially, the expectation of $\tilde{h}\left(w_l\right)$ under the variational distribution of w_l is equivalent to the expectation of $\tilde{h}_{\theta_l}\left(\epsilon\right)$ under the distribution on ϵ (for example, $N\left(0, I\right)$). Thus, we can obtain this expectation by calculating:

$$\begin{aligned}&\mathbb{E}_{q_l}\left[\tilde{h}\left(w_l\right)\right] = \mathbb{E}\left[\tilde{h}_{\theta_l}\left(\epsilon\right)\right] \\ &= \mathbb{E}\left[h\left(\mu_l\right) + h'\left(\mu_l\right)\sigma_l^2 \odot \epsilon d\epsilon + \frac{h''\left(\mu_l\right)}{2}\left(\sigma_l^2 \odot \epsilon\right)^2\right] \\ &= h\left(\mu_l\right) + h'\left(\mu_l\right)\sigma_l^2 \mathbb{E}[\epsilon] + \frac{h''\left(\mu_l\right)}{2}\mathbb{E}\left[\left(\sigma_l^2 \odot \epsilon_l\right)^2\right] \\ &= h\left(\mu_l\right) + \frac{h''\left(\mu_l\right)}{2}\left(\sigma_l^2\right)^2\end{aligned} \tag{15}$$

Algorithm 1 The first moment propagation in l-th via Reparameterized Importcance Sampling

1: Variational posterior $q_l(w_l)$ parameters $\theta_l = \left(\mu_l, \sigma_l^2\right)$, the first moment of last layer $\tilde{z}_{l-1}$.
2: Approximate the optimal proposal distribution $r_l(w_l|\overset{*}{\mu}_l, \overset{*}{\sigma}_l^2)$:
- Using Eq. (15) in Eq. (11) to calculate the mean $\overset{*}{\mu}_l$ by set $h(w_l) = |f(w_l)|w_l$
- UsingEq. (15) in Eq. (12) to calculate the variance $\overset{*}{\sigma}_l^2$ by set $h(w_l) = (|f(w_l)|w_l)^2$

3: **for** $m = 1$ to M **do**
4: Sample $\epsilon^m \sim \mathcal{N}\left(\frac{\overset{*}{\mu}_l - \mu_l}{\sigma_l^2}, \frac{\overset{*}{\sigma}_l^2}{\sigma_l^2}\right)$.
5: Let $w^m = \mu_l + \sigma_l^2 \odot \epsilon^m$
6: Calculate $\gamma_l^m = \frac{q(w_l^m|\theta_l)}{r_l\left(w_l^m\right)} \approx \frac{1}{|f(w_l)|}$
7: Calculate $\delta\left(w_l^m \tilde{z}_{l-1}\right)$
8: **end for**
9: Calculate $\tilde{z}_l \approx \frac{1}{M}\sum_{m=1}^{M} \gamma_l^m \delta\left(w_l^m \tilde{z}_{l-1}\right)$

Note that even though this uses quadratic term of standard deviation, it is a squared term of variance. Combining Eqs. (11) (12) (15) provides closed-form approximations for the mean $\overset{*}{u}$ and $\overset{*}{\sigma}^2$. These approximations depend on the function $h(w_l)$, which can be defined as $|f\left(w_l\right)|$, $|f\left(w_l\right)| w_l$ or $|f\left(w_l\right)| w_l^2$, respectively.

4.2. Defining Distribution on Exogenous Randomness

Now that we have constructed an approximation of the optimal proposal distribution, $r_l\left(w_l|\overset{*}{u}, \overset{*}{\sigma}^2\right)$, with high correlation to the variational parameters u_l, σ_l^2, we can employ it in Eq. (12) to obtain a lower-variance estimator of the first moment.

$$\tilde{z}_l \approx \frac{1}{M}\sum_{m=1}^{M} \frac{q\left(w_l^m|\theta_l\right)}{r_l^*\left(w_l^m\right)}\delta\left(w_l^m \tilde{z}_{l-1}\right), w_l^m \sim \mathcal{N}\left(\overset{*}{\mu}, \overset{*}{\sigma}^2\right) \tag{16}$$

The weights w_l for the l-th layer are generated from the proposal distribution. These weights are subsequently utilized in estimating the first moment through the computation of Monte Carlo (MC) averages. We define the sampling procedure as a transformation of the random variable ϵ. Tthe reparameterization trick produces the weights w_l by employing a differentiable mapping function t:

$$w_l = \overset{*}{\mu} + \overset{*}{\sigma}^2 \odot \epsilon, \epsilon \sim \mathcal{N}(0, I)$$

However, we have observed that the likelihood gradients of variational parameters have a complex derivation, which

can be expressed as:

$$\begin{gathered} Monte\ Carlo\ sampling \\ \frac{\partial w_l}{\partial \mu_l} = I, \frac{\partial w_l}{\partial \sigma_l} = \epsilon \\ \downarrow \\ Importance\ sampling \\ \frac{\partial w_l}{\partial \mu_l} = \frac{\partial \overset{*}{\mu}}{\partial \mu_l} + \frac{\partial \overset{*}{\sigma}^2}{\partial \mu_l}\epsilon, \frac{\partial w_l}{\partial \sigma_l} = \frac{\partial \overset{*}{\mu}}{\partial \sigma_l} + \frac{\partial \overset{*}{\sigma}^2}{\partial \sigma_l}\epsilon \end{gathered}$$

Therefore, we introduce a **Reparameterized Importance Sampling** (RIS) method, which allows us to apply importance sampling in conjunction with the reparameterization trick. By specifying a particular distribution for ϵ, the weights sampled from the proposal distribution can be represented as a transformation of ϵ combined with the variational parameters.

Proposition 2. Let w be a random variable following a distribution denoted as $r(w|\overset{*}{u}, \overset{*}{\sigma}^2)$. Now, consider another distribution with mean u and variance σ^2. In the context of this second distribution, we can analyze the properties of the random variable w:

$$w = u + \sigma^2 \odot \epsilon, \epsilon \sim \mathcal{N}\left(\frac{\overset{*}{\mu} - \mu}{\sigma^2}, \frac{\overset{*}{\sigma}^2}{\sigma^2}\right) \tag{17}$$

Proof.

$$\begin{aligned} \mathbb{E}\left[\mu + \sigma^2 \odot \epsilon\right] &= \mu + \sigma^2 \mathbb{E}\left(\epsilon\right) \\ &= \mu + \sigma^2 \frac{\overset{*}{\mu} - \mu}{\sigma^2} \\ &= \overset{*}{\mu} = \mathbb{E}_{r(w)}\left[w\right] \end{aligned}$$

$$\begin{aligned} var\left[\mu + \sigma^2 \odot \epsilon\right] &= var\left[\sigma^2 \odot \epsilon\right] \\ &= \mathbb{E}\left[\left(\sigma^2 \odot \epsilon\right)^2\right] - \left(\mathbb{E}\left[\sigma^2 \odot \epsilon\right]\right)^2 \\ &= \sigma^4 \left(\frac{\overset{*}{\mu} - \mu}{\sigma^2}\right)^2 + \sigma^4 (\frac{\overset{*}{\sigma}^2}{\sigma^2})^2 - (\overset{*}{\mu} - \mu)^2 \\ &= \overset{*}{\sigma}^2 = var_{r(w)}\left[w\right] \end{aligned}$$

To compute the likelihood gradient with respect to the variational parameters, we employ Proposition 1 to address the challenge of sampling weights directly from the proposal distribution. Specifically, we simplify the distribution for ϵ to be parameter-free, enabling us to efficiently calculate gradients.

$$\begin{gathered} Importance\ sampling \\ \frac{\partial w_l}{\partial \mu_l} = \frac{\partial \overset{*}{\mu}}{\partial \mu_l} + \frac{\partial \overset{*}{\sigma}^2}{\partial \mu_l}\epsilon, \frac{\partial w_l}{\partial \sigma_l} = \frac{\partial \overset{*}{\mu}}{\partial \sigma_l} + \frac{\partial \overset{*}{\sigma}^2}{\partial \sigma_l}\epsilon \\ , \\ \downarrow \\ RIS \\ \frac{\partial w_l}{\partial u_l} = I, \frac{\partial w_l}{\partial \sigma_l} = \epsilon \end{gathered}$$

Algorithm 1 entails a detailed description of how the first moment propagates across each layer.

5. Related Works

Approximations to the posterior of Neural Networks can be traced back to David MacKay's 1992 work (MacKay, 1992). In this groundbreaking study, MacKay introduced the concept of Occam's Razor in Bayesian modeling, which refers to that simpler models are more likely to be correct. However, at that time, the approach relied on the relatively crude Laplace approximation, suitable only for small and shallow neural networks. In the early days of Bayesian neural networks, there were two primary approximation methods. On the one hand, Hinton and van Camp introduced the Variational Bayes (VB) approach for posterior inference (Hinton & van Camp), aiming to minimize the description length. On the other hand, Neal developed efficient gradient-based Monte Carlo techniques, specifically Hamiltonian Monte Carlo(Neal, 1992).

After more than a decade of relative silence in the field of Bayesian Neural Networks (BNNs), Graves revitalized Variational Bayes (VB) by introducing Monte Carlo variational inference (MCVI) (Graves, 2011) as a practical and scalable method for optimizing the VB objective function. This development marked a resurgence in modern BNN research. However, despite advancements in more accurate and efficient inference techniques, some scenarios still witness BNNs trained with Mean-field variational inference (MFVI) yielding posterior distributions that underperform a baseline trained with standard stochastic gradient descent. Additionally, MFVI struggles with scalability when applied to large-scale neural networks. To address these challenges, researchers have explored various methods. Notably, the "Reparameterization trick" proposed by Kingma and Welling has gained popularity for efficiently reducing gradient estimator variance (Kingma et al., 2015). Blundell et al. introduced the "Bayes by Backprop" method using Mean-field variational inference (MFVI) to approximate BNN posteriors, incorporating the "Reparameterization trick" for weight sampling (Blundell et al., 2015). Wu et al. introduced "Deterministic Variational Inference" (DVI), a deterministic approximation to variational inference in neural networks. DVI eliminates Monte Carlo sampling-induced stochasticity by propagating moments of the distribution for activations, thereby stabilizing BNN training (Wu et al., 2019). Zhang et al. tackled the randomness associated with Monte Carlo sampling by training BNNs with an Adversarial Distribution. An intuitive strategy to improve BNN initialization involves specifying "informed weight priors" extracted from pre-trained deterministic neural networks with equivalent architecture (Rossi et al., 2019; Krishnan et al., 2020; Wu et al., 2019). Alternatively, the issues mentioned can also be mitigated by artificially reducing posterior uncertainty through the use of "cold posteriors" (Wilson & Izmailov, 2020; Wenzel et al., 2020; Zhang et al., 2018; Bae et al., 2018).

Markov chain Monte Carlo (MCMC) approaches using stochastic gradient methods constitute a diverse and effective family of approximation methods for Bayesian neural networks. However, their development has been hindered by computational inefficiency. These approaches rely on unbiased log-likelihood values to estimate approximate posterior parameters. A seminal contribution in this domain was the introduction of Stochastic Gradient Langevin Dynamics (SGLD) by Welling, known for its simplicity and efficiency in implementation (Welling & Teh, 2011). Recent advancements have focused on enhancing efficiency in cases of correlated posteriors, achieved through the estimation of the Fisher Information Matrix (Ahn et al., 2012). Chen et al. extended Hamiltonian Monte Carlo (HMC) to accommodate the stochastic gradient scenario, further broadening the applicability of these methods (Chen et al., 2014). Ma et al. and Gong et al. have provided comprehensive characterizations of Stochastic Gradient Markov Chain Monte Carlo (SGMCMC) approaches, offering valuable insights into their workings (Ma et al., 2015; Gong et al., 2018).

Furthermore, there are a few methods that aim to provide "inexpensive" approximations to the Bayesian posterior. While these methods may lack a rigorous theoretical foundation, they are straightforward to implement. For instance, Gal and Ghahramani proposed that dropout can be seen as an approximate equivalent to variational inference, offering a computationally efficient approach to Bayesian modeling (Gal & Ghahramani, 2016). Similarly, Bootstrap posteriors have been introduced as a versatile, dependable, and accurate technique for posterior inference, drawing on principles from predictive statistics (Harris, 1989; Fushiki et al., 2005; Lakshminarayanan et al., 2017). It's worth noting that training an ensemble of bootstrap posteriors typically incurs a higher computational cost compared to training a single model.

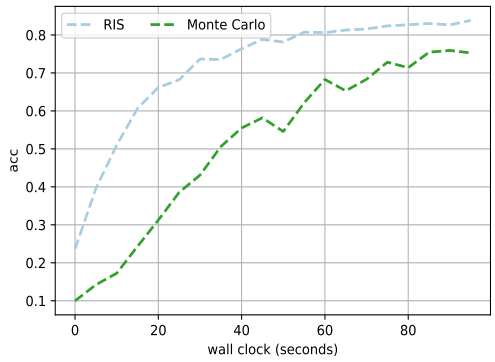
6. Experimental Results

We validate the effectiveness of the proposed approximate inference method regarding three aspects, including convergence improvement, model performance enhancement, and out-of-distribution data uncertainty estimation. Our experiments on real-world applications include LeNet architecture (LeCun et al., 1998) for MNIST digit dataset, ResNet20, ResNet56 architecture(He et al., 2016), on CIFAR-10 and CIFAR-100 datasets (Krizhevsky et al., 2009). We implement the above Bayesian architecture and train them with RIS and with standard Monte Carlo sampling (we consider vanilla MFVI with the local reparameterization trick and 'cold posterior'), under the PyTorch framework, on a Titan RTX 28G device, and using the same random seeds.

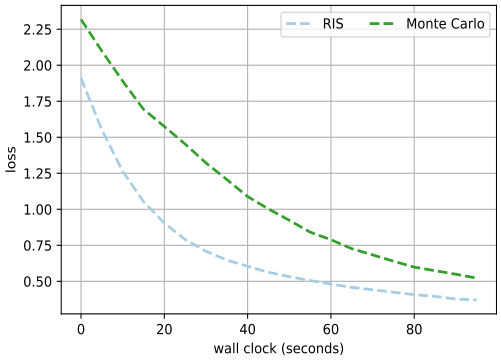
6.1. Comparison with Baseline MFVI

6.1.1. Optimizer Convergence and Stability

We compare the optimization traces for the Bayesian Neural Network trained with RIS and Monte Carlo sampling respectively. At each iteration, we estimate the true accuracies and losses value using 10 Monte Carlo samples. The accuracies represent how well the model recognizes the validation set during training. The losses represent the Kullback-Leibler (KL) divergence between the variational distribution and the posterior. We optimize the two loss objectives using adam (Kingma & Ba, 2014) for same step size.



(a)



(b)

Figure 2. Optimization trace of applying MFVI to approximate the posterior of the neural network. We run the standard Monte Carlo sampling (green line) and the RIS (light blue line) with 10 samples.

Figure 2 compares optimization traces for BNNs trained with RIS and standard Monte Carlo Sampling. We can observe that the RIS makes early progress and converge quickly. We also find that at the same time point, RIS improves accuracy by up to 35% and reduces loss by up to 45% compared to Monte Carlo sampling.

Table 1. Comparison of classification accuracies for different models trained with the proposed method and Monte Carlo samplings sampling on various data sets.

Dataset	Bayesian Model	Sampling Method	
		Monte Carlo	RIS
Cifar-10	ResNet20	83.56 ± 0.45	87.37 ± 0.26
	ResNet56	84.16 ± 0.38	88.25 ± 0.14
Cifar-100	ResNet20	52.47 ± 1.46	55.62 ± 0.82
	ResNet56	54.62 ± 1.05	58.75 ± 0.72
Mnist	LeNet	99.24 ± 0.12	99.56 ± 0.03

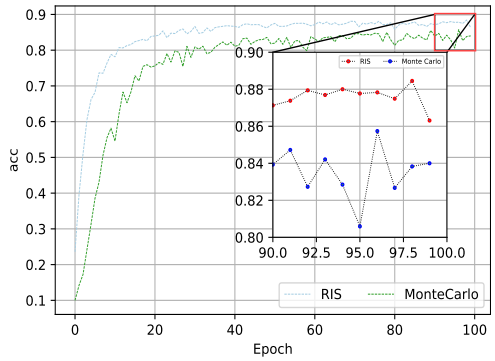
6.1.2. Combination on Predictive Performance

We demonstrate that the proposed approximate inference method improves the model's performance by comparing the classification accuracies. To obtain predictive distributions of various models, we sample from the posterior distribution of the weights and perform stochastic forward passes during the inference phase. We compare these results against models that utilize the Monte Carlo sampling method. Table 1 presents classification accuracies for various models. BNNs trained using our proposed approximate inference methos achieve better predictive accuracies as compared to the MFVI with Monte Carlo sampling. Figure 3(a) displays the accuracy rates for Bayesian ResNet-20 models trained on the CIFAR-10 dataset. We can easily notice that the curve of accuracy rates for the our approximation method exhibits a smoother trend compared to the MFVI with Monte Carlo sampling. This confirms that our proposed method helps to stabilize BNNs training and reduce variance due to Monte Carlo sampling. Following (Zhang et al., 2022), Figure 3(b) displays a comparison of the accuracies of 1000 models sampled from Bayesian neural networks.

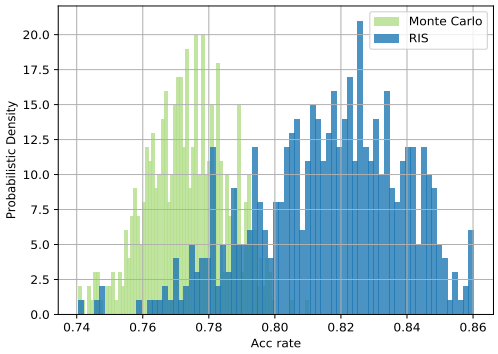
Table 2. Comparison of classification accuracy of ResNet-20 trained using the proposed method and trained using Monte Carlo sampling at different sampling times. The experiment is constructed on the CIFAR-10 dataset.

Dataset	model	MC samples	method	Accuracy
CIFAR-10	ResNet-20	10	MFVI (tempered)	82.74 ± 0.52
CIFAR-10	ResNet-20	10	RIS	87.07 ± 0.27
CIFAR-10	ResNet-20	100	MFVI (tempered)	85.38 ± 0.35
CIFAR-10	ResNet-20	100	RIS	87.74 ± 0.23

Additionally, we compared the predictive performance with the MFVI method at different sampling times. The number of samplings is 10 and 100, respectively. The results are summarized in table 2 We find that the performance of our method is insensitive to the number of Monte Carlo samples. On the other hand, although the accuracy of the baseline method improves with an increasing number of samples, there remains a gap between its performance and our method.



(a)



(b)

Figure 3. (a) Accuracy rates of Bayesian ResNet20 trained with using RIS and with Monte Carlo samplings sampling during a 100-epoch training.(b) Accuracy comparison of 1000 models sampled from Bayesian neural networks using RIS and Monte Carlo sampling.

Table 3. Comparison of classification accuracies for ResNet-20 trained with the proposed method and other SOTA methods on the CIFAR-10 dataset.

Dataset	model	Method	Accuracy
CIFAR-10	ResNet-20	MFVI	22.46 ± 0.70
CIFAR-10	ResNet-20	MFVI(tempered))	83.56 ± 0.45
CIFAR-10	ResNet-20	SWAG	86.80 ± 0.10
CIFAR-10	ResNet-20	VOGN	85.36 ± 0.25
CIFAR-10	ResNet-20	GLM	84.35 ± 0.18
CIFAR-10	ResNet-20	Adversarial Sampling	86.33 ± 0.45
CIFAR-10	ResNet-20	RIS	**87.37 ± 0.26**
CIFAR-10	ResNet-20	HMC	**90.02 ± 0.26**

6.2. Compare with SOTA BNNs

We compare the results using our approximation with that of variants of MFVI and other previous SOTA methods. These methods include SWAG (Maddox et al., 2019), VOGN (Osawa et al., 2019), GLM (Immer et al., 2021) and Adversarial sampling (Zhang et al., 2022). All these method do not use data augmentation and all models in the following table use a ResNet-20 architecture. We find that our method attains a better predictive accuracy than these methods, except for the HMC method. The results are summarized in table 3 Although the HMC method currently provides the best approximation of the Bayesian posterior, it consumes more computational resources.

6.3. Uncertainty Estimation to Out-of-Distribution Data

Uncertainty estimation is crucial ability for the Bayesian Neural Networks. The total uncertainty of a model at an input point x^* is typically measured by the predictive entropy.

$$\mathcal{H}\left(y^*|x^*,\mathcal{W}\right) = \sum_y -p\left(y^*|x^*,\mathcal{W}\right)\log p\left(y^*|x^*,\mathcal{W}\right)$$

It can be broken down into two main types of uncertainty: on the one hand, *Aleatoric* uncertainty captures inherent noise in the observations, which would not be further reduced even if additional data were to be collected.

$$\mathcal{H}_{ale}\left(y^*|x^*,\mathcal{W}\right) = \mathbb{E}_{p(\mathcal{W}|D)}\left[\mathcal{H}\left(y^*|x^*,\mathcal{W}\right)\right]$$

Epistemic uncertainty, on the other hand, refers to the ignorance of model parameters regarding the collected data. This type of uncertainty can be reduced with more data and is often known as model uncertainty.

$$\begin{aligned}\mathcal{H}_{epi}\left(y^*|x^*,\mathcal{W}\right) &= \mathcal{H}\left(y^*|x^*,\mathcal{W}\right) \\ &\quad - \mathbb{E}_{p(\mathcal{W}|D)}\left[\mathcal{H}\left(y^*|x^*,\mathcal{W}\right)\right]\end{aligned}$$

We evaluate the uncertainty estimation ability of Bayesian neural networks trained using our proposed approximate inference method to identify out-of-distribution data. Out-of-distribution samples are data points that fall far off from the training data distribution. We use CIFAR-10 as the in-distribution samples to train a Bayesian ResNet-20 model and use other images in CIFAR-100 as the out-of distribution samples which were not used during the training phase. Figure 4 shows the density histograms of aleatoric and epistemic uncertainty estimates. It's clear that the out-of-distribution samples have higher uncertainty values than the in-distribution samples.

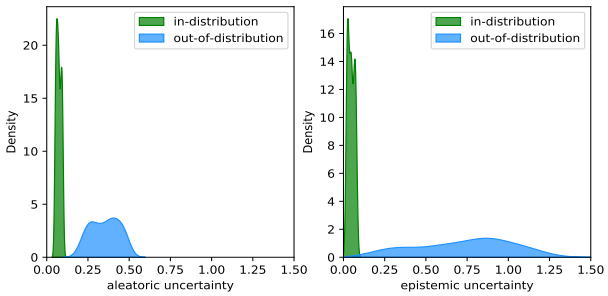


Figure 4. Density histograms obtained from in- and out-of-distribution samples.

7. Conclusions

In this paper, we have optimized the current Mean-Field Variational Inference (MFVI) method to efficiently obtain an approximate posterior for neural networks. To achieve this, we have proposed a novel Reparameterized Importance Sampling technique for estimating the first moment of each activation layer in neural networks. Specifically, we have devised a cost-effective approximation expression for the optimal proposal distribution and represented the sampling procedure from the proposal distribution as a transformation of exogenous randomness combined with the variational parameters. Our proposed method not only enhances convergence and stability but also significantly improves the performance of Bayesian Neural Networks (BNNs). Furthermore, our experiments have shown that the uncertainty estimations derived from models trained using the proposed method are highly reliable for identifying out-of-distribution data.

This method described in this work is specifically designed for Gaussian approximating families used in mean-field variational inference. However, we still incorporate the 'cold posterior' strategy to prevent convergence to the prior during the training of BNNs. Looking ahead, we are aiming to tailor our technique to accommodate diverse initializations of the prior. Our ultimate goal is to bolster the utilization of the mean-field variational inference method within the realm of neural networks. We aspire to make significant con-

tributions to the ongoing research in uncertainty modeling and posterior inference.

Acknowledgments

This work was supported by the National Key R&D Program of China (2021YFC3320302).

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Ahn, S., Korattikara, A., and Welling, M. Bayesian posterior sampling via stochastic gradient fisher scoring. *arXiv preprint arXiv:1206.6380*, 2012.

Bae, J., Zhang, G., and Grosse, R. Eigenvalue corrected noisy natural gradient. *arXiv preprint arXiv:1811.12565*, 2018.

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. Weight uncertainty in neural network. In *International conference on machine learning*, pp. 1613–1622. PMLR, 2015.

Borcherds, P. Importance sampling: an illustrative introduction. *European Journal of Physics*, 21(5):405, 2000.

Chen, T., Fox, E., and Guestrin, C. Stochastic gradient hamiltonian monte carlo. In *International conference on machine learning*, pp. 1683–1691. PMLR, 2014.

Coker, B., Bruinsma, W. P., Burt, D. R., Pan, W., and Doshi-Velez, F. Wide mean-field bayesian neural networks ignore the data. In *International Conference on Artificial Intelligence and Statistics*, pp. 5276–5333. PMLR, 2022.

Dusenberry, M., Jerfel, G., Wen, Y., Ma, Y., Snoek, J., Heller, K., Lakshminarayanan, B., and Tran, D. Efficient and scalable bayesian neural nets with rank-1 factors. In *International conference on machine learning*, pp. 2782–2792. PMLR, 2020.

Fushiki, T., Komaki, F., and Aihara, K. Nonparametric bootstrap prediction. *Bernoulli*, 11(2):293–307, 2005.

Gal, Y. and Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059. PMLR, 2016.

Gong, W., Li, Y., and Hernández-Lobato, J. M. Meta-learning for stochastic gradient mcmc. *arXiv preprint arXiv:1806.04522*, 2018.

Graves, A. Practical variational inference for neural networks. *Advances in neural information processing systems*, 24, 2011.

Harris, I. R. Predictive fit for natural exponential families. *Biometrika*, 76(4):675–684, 1989.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hinton, G. and van Camp, D. Keeping neural networks simple by minimising the description length of weights. 1993. In *Proceedings of COLT-93*, pp. 5–13.

Immer, A., Korzepa, M., and Bauer, M. Improving predictions of bayesian neural nets via local linearization. In Banerjee, A. and Fukumizu, K. (eds.), *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 703–711. PMLR, 13–15 Apr 2021. URL https://proceedings.mlr.press/v130/immer21a.html.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kingma, D. P., Salimans, T., and Welling, M. Variational dropout and the local reparameterization trick. *Advances in neural information processing systems*, 28, 2015.

Krishnan, R., Subedar, M., and Tickoo, O. Specifying weight priors in bayesian deep neural networks with empirical bayes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 4477–4484, 2020.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Ma, Y.-A., Chen, T., and Fox, E. A complete recipe for stochastic gradient mcmc. *Advances in neural information processing systems*, 28, 2015.

MacKay, D. J. A practical bayesian framework for backpropagation networks. *Neural computation*, 4(3):448–472, 1992.

Maddox, W. J., Izmailov, P., Garipov, T., Vetrov, D. P., and Wilson, A. G. A simple baseline for bayesian uncertainty in deep learning. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/118921efba23fc329e6560b27861f0c2-Paper.pdf.

Neal, R. Bayesian learning via stochastic dynamics. *Advances in neural information processing systems*, 5, 1992.

Osawa, K., Swaroop, S., Jain, A., Eschenhagen, R., Turner, R. E., Yokota, R., and Khan, M. E. Practical deep learning with bayesian principles. *arXiv preprint arXiv:1906.02506*, 2019.

Rossi, S., Michiardi, P., and Filippone, M. Good initializations of variational Bayes for deep models. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5487–5497. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/rossi19a.html.

Tokdar, S. T. and Kass, R. E. Importance sampling: a review. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(1):54–60, 2010.

Welling, M. and Teh, Y. W. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 681–688, 2011.

Wenzel, F., Roth, K., Veeling, B. S., Świątkowski, J., Tran, L., Mandt, S., Snoek, J., Salimans, T., Jenatton, R., and Nowozin, S. How good is the bayes posterior in deep neural networks really? *arXiv preprint arXiv:2002.02405*, 2020.

Wilson, A. G. and Izmailov, P. Bayesian deep learning and a probabilistic perspective of generalization. *Advances in neural information processing systems*, 33:4697–4708, 2020.

Wu, A., Nowozin, S., Meeds, E., Turner, R. E., Hernández-Lobato, J. M., and Gaunt, A. L. Deterministic variational inference for robust bayesian neural networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=B1l08oAct7.

Zhang, G., Sun, S., Duvenaud, D., and Grosse, R. Noisy natural gradient as variational inference. In *International conference on machine learning*, pp. 5852–5861. PMLR, 2018.

Zhang, J., Hua, Y., Song, T., Wang, H., Xue, Z., Ma, R., and Guan, H. Improving bayesian neural networks by adversarial sampling. 2022.

A. Applicability to BNNs with Non-deterministic Bias

We describe the activation action of the l-th layer using an unbiased case. Theoretically, variational inference uses fully factorized Gaussian approximation, thus the biases will naturally be part of the weights and are independent of each other. Here, we discuss the case of using uncertain bias.

In a feed-forward BNN, the activation action of l-th layer that maps z^{l-1} to z^l can be expressed as follows:

$$z_l = \delta(w_l z_{l-1} + b_l), w_l \sim q(w_l|\theta_{w_l}), b_l \sim q(b_l|\theta_{b_l})$$

where w_l and b_l represents the weights and bias of the l-th layer, which are drawn from the variational distribution $q(w_l|\theta_{w_l})$ and $q(b_l|\theta_{b_l})$ respectively.

Using importance sampling:

$$\tilde{z}_l = \frac{1}{M}\sum_{m=1}^{M} \frac{q\left(w_l^m|\theta_{wl}\right) q\left(b_l^m|\theta_{bl}\right)}{r_l\left(w_l^m\right) r_l\left(b_l^m\right)} \delta\left(w_l^m \tilde{z}_{l-1} + b_l^m\right), w_l^m \sim r_{wl}\left(w_l\right), b_l^m \sim r_{bl}\left(b_l\right)$$

We set $W_l = [w_l, b_l], Z_{l-1} = [z_{l-1}, I]$, where $[,]$ represents the concatenation operation on the channels (), then $\tilde{z}_l$ can be expressed as:

$$\tilde{z}_l = \frac{1}{M}\sum_{m=1}^{M} \frac{q\left(W_l^m|\theta_l\right)}{r_l\left(W_l^m\right)} \delta\left(W_l^m \tilde{Z}_{l-1}\right), W_l^m \sim r_l\left(W_l\right)$$

Now, we can proceed to the next calculation, which shows that the proposed method is not limited by the bias.

B. Comparison under Using Bias

We demonstrate that the proposed approximate inference method improves the performance model with bias by comparing the classification accuracies. Table 4 presents classification accuracies for various models. We can find that BNNs trained using our proposed approximate inference methodology consistently outperform these using the MFVI with Monte Carlo sampling.

Table 4. Comparison of classification accuracies for different models with bias trained with the proposed method and Monte Carlo samplings sampling on various data sets.

Dataset	Bayesian Model	Sampling Method	
		Monte Carlo	RIS
Cifar-10	ResNet20	83.94 ± 0.39	88.05 ± 0.24
	ResNet56	85.02 ± 0.32	88.95 ± 0.11
Cifar-100	ResNet20	53.69 ± 1.46	56.12 ± 0.75
	ResNet56	55.02 ± 0.95	59.92 ± 0.71
Mnist	LeNet	99.25 ± 0.14	99.66 ± 0.04