Implicit Generative Prior for Bayesian Neural Networks

Yijia Liu[1] and Xiao Wang[2]

[1,2]Department of Statistics, Purdue University

Abstract

Predictive uncertainty quantification is crucial for reliable decision-making in various applied domains. Bayesian neural networks offer a powerful framework for this task. However, defining meaningful priors and ensuring computational efficiency remain significant challenges, especially for complex real-world applications. This paper addresses these challenges by proposing a novel neural adaptive empirical Bayes (NA-EB) framework. NA-EB leverages a class of implicit generative priors derived from low-dimensional distributions. This allows for efficient handling of complex data structures and effective capture of underlying relationships in real-world datasets. The proposed NA-EB framework combines variational inference with a gradient ascent algorithm. This enables simultaneous hyperparameter selection and approximation of the posterior distribution, leading to improved computational efficiency. We establish the theoretical foundation of the framework through posterior and classification consistency. We demonstrate the practical applications of our framework through extensive evaluations on a variety of tasks, including the two-spiral problem, regression, 10 UCI datasets, and image classification tasks on both MNIST and CIFAR-10 datasets. The results of our experiments highlight the superiority of our proposed framework over existing methods, such as sparse variational Bayesian and generative models, in terms of prediction accuracy and uncertainty quantification.

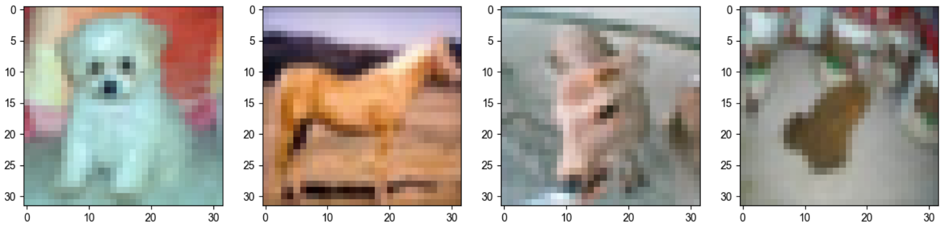
Key words: Deep neural networks; empirical Bayes; latent variable model; stochastic gradient method; variational inference

1 Introduction

Despite the remarkable accomplishments of deep neural networks (DNNs) in the field of artificial intelligence, they encounter numerous challenges. When utilized in the context of supervised learning, DNN models frequently struggle to accurately gauge uncertainty within training data and only provide a point estimate regarding class or prediction. The consequences of this limitation are profound, particularly when these models are entrusted with life-or-death decisions. In medical domains, for instance, experts may find it challenging to determine whether they should rely on automated diagnostic systems, and passengers in self-driving vehicles may not receive alerts to take control when the vehicle encounters situations it does not comprehend.

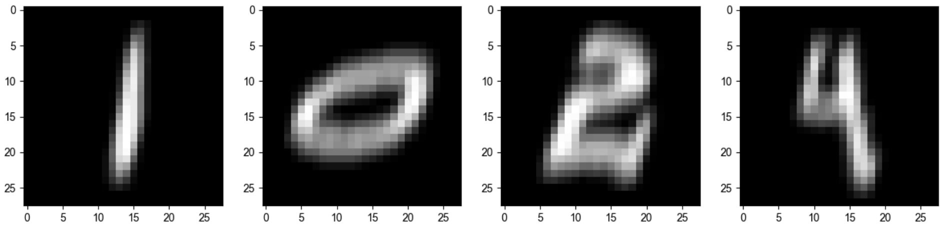
To illustrate the importance of predictive uncertainty, we present two real-world classification examples. First, in Figure 1, we compare the predicted probabilities of the ResNet-18 (He et al., 2016) classifier with our 95% Bayesian credible intervals on four test images from the CIFAR-10 dataset (`https://www.kaggle.com/c/cifar-10/`). This dataset comprises 60,000 32×32 color images across 10 classes. We observe that the standard DNN method often assigns high probabilities to incorrect classes. In contrast, our proposed approach, *Neural Adaptive Empirical Bayes* (NA-EB), offers prediction intervals for the likelihood of each class label. The widths of the prediction intervals reflect how sure or unsure NA-EB is about the correctness of its predictions while the point estimate of the likelihood generated by the standard DNN method does not convey uncertainty information. In the case of the deer image, our method produces similarly wide and overlapping prediction intervals for the two potential classes, respectively, implying the uncertainty in its predictions. In contrast, the standard DNN method assigns a high probability to the wrong class without accounting for uncertainty. Furthermore, in the case of the frog image, our method yields a relatively narrow prediction interval for the correct class, whereas the DNN method assigns a high probability estimate to the wrong class. This indicates that NA-EB not only quantifies predictive uncertainty but also enhances classification accuracy by implicitly specifying a correct prior for classifier weights. Furthermore, we present comparisons in a scenario of weaker data signals in Figure 2. Specifically, we employ a fully connected feedforward neural network as the base classifier on an artificially noisy dataset (Basu et al., 2017) created by introducing motion blur into the MNIST dataset (`http://yann.lecun.com/exdb/mnist/`). NA-EB generates narrow prediction intervals with high probability values for certain digits but wide overlapping intervals for others, indicating uncertainty in the model predictions. In contrast, for digits such as "two" and "four", the DNN method assigns high probabilities to incorrect classes without providing any information about the uncertainty regarding its belief. In summary, NA-EB offers a robust classifier capable of expressing its uncertainty through a full posterior distribution rather than a single point estimate. This suggests potential applications of NA-EB in the fields of medical diagnostics, such as the automated classification of diabetic retinopathy from retinal images. Uncertainty estimation is particularly critical in the medical domain, ensuring confident model predictions for screening automation while flagging uncertain cases for manual review by a medical expert. Bayesian deep learning has already demonstrated its significance in medical diagnostics (Worrall et al., 2016; Leibig et al., 2017; Kamnitsas et al., 2017; Ching et al., 2018), underscoring the potential relevance of NA-EB in such applications due to its enhanced performance in uncertainty quantification and prediction accuracy.

Under the Bayesian framework, given the prior of the weights of the classifier or the



True Label:	Dog	Horse	Deer	Frog
Predicted Label by Standard DNN:	Cat: 0.01% Dog: 99.93%	Deer: 0.01% Horse: 99.94%	Deer: 15.79% Dog: 52.84%	Cat: 99.95% Frog: 0.03%
Prediction Interval by NA-EB:	Cat: 0.01%-0.09% Dog: 99.78%-99.92%	Deer: 0.01%-0.16% Horse: 99.11%-99.96%	Deer: 30.25%-85.17% Dog: 14.67%-68.39%	Cat: 0.07%-5.33% Frog: 86.29%-97.34%

Figure 1: A comparison of classification results between the standard DNN method and NA-EB on four test images from the CIFAR-10 dataset.



True Label:	1	0	2	4
Predicted Label by Standard DNN:	1: 100.00% 7: 0.00%	0: 99.31% 9: 0.01%	2: 0.22% 3: 98.41%	4: 38.61% 9: 61.39%
Prediction Interval by NA-EB:	1: 100.00%-100.00% 7: 0.00%-0.00%	0: 99.99%-100.00% 9: 0.00%-0.01%	2: 89.58%-99.98% 3: 0.04%-0.13%	4: 36.14%-99.98% 9: 14.43%-63.86%

Figure 2: A comparison of classification results between the standard DNN method and NA-EB on four test images from the noisy MNIST with motion blur dataset.

regression model depending on the specific supervised learning task, the 95% Bayesian credible interval can be easily constructed based on the shortest interval that contains 95% of the predictive probability. However, when prior knowledge of the weights is not available, choosing a priori is a challenging task. Furthermore, when the weights are believed to reside in a low-dimensional manifold in higher-dimensional space, specifying the prior distribution of the weights is not straightforward since it may not have a tractable formula. For example, if we model the MNIST data, which contain 60,000

28×28 handwritten digit images, using a simple 2-hidden layer neural network with 784 input nodes, 800 nodes in each of two hidden layers and 10 nodes in the output layers, the model in total contains about 1.3 million parameters. It is not trivial to specify a prior distribution of 1.3 million dimensions. It is also reasonable to assume that these 1.3 million parameters reside in a low-dimensional manifold.

Bayesian DNNs (BNNs) have been extensively studied by (Neal, 1992; MacKay, 1995; Neal, 1996; Bishop, 1997; Lampinen and Vehtari, 2001; Bernardo and Smith, 2009). More recent developments with advanced algorithms can be found in (Sun et al., 2017; Mullachery et al., 2018; Hubin et al., 2018; Javid et al., 2020; Wilson and Izmailov, 2020; Izmailov et al., 2021). BNNs promise improved predictions and yield richer representations from cheap model averaging. The most widely used priors for BNNs are isotropic Gaussian (Neal, 1996; Hernández-Lobato and Adams, 2015; Louizos et al., 2017; Dusenberry et al., 2020; Immer et al., 2021). Gaussian priors have recently been shown to cause a cold posterior effect in BNNs. That is, the tempered posterior performs better than the true Bayesian posterior, suggesting prior misspecification (Wenzel et al., 2020). Another well-studied prior is placing sparsity-induced priors on network weights (Blundell et al., 2015; Molchanov et al., 2017; Ghosh et al., 2019; Bai et al., 2020), which can be used to aid compression of network weights. But sparsity-induced priors do not benefit prediction. In addition, Quinonero-Candela et al. (2005) demonstrated that many priors of convenience can lead to unreasonable predictive uncertainties. Although there are many methods available for Bayesian computing, such as MCMC (Neal, 1996; Graves, 2011), Langevin dynamics (Welling and Teh, 2011), and Hamiltonian methods (Springenberg et al., 2016), computing the posterior distribution is generally intractable, making it difficult to make inferences about the predictive distribution.

In this article, we propose a new class prior distributions called *implicit generative prior* to facilitate Bayesian modeling and inference. This idea is motivated by deep generative models in the machine learning literature (Goodfellow et al., 2014; Kingma and Welling, 2013). *Prescribed explicit priors* are those that provide an explicit distribution of the classifier's or the regression model's weights. Instead, the implicit generative prior represents the prior distribution through a function transformation of a known distribution to define a stochastic procedure that directly generates the parameter. We use a low-dimensional latent variable with a known prior distribution and transform it using a deterministic function with the hyperparameters. The deterministic transformation function is specified by a DNN, which takes the latent variable as the input and produces the weights as the output. This formulation is commonly seen in deep learning models such as the generative adversarial network (GAN, Goodfellow et al., 2014) and the variational autoencoder (VAE, Kingma and Welling, 2013).

In the context of Bayesian inference, we encounter two immediate challenges. First, the hyperparameter, the weights of the deterministic transformation function specified by a DNN, is high dimensional, making its selection a difficult task. Second, Bayesian computation becomes computationally intractable in many cases, particularly when dealing with DNNs of any practical size. To address these challenges, we propose a variational approach called Neural Adaptive Empirical Bayes (NA-EB). The key idea behind NA-EB is to leverage a variational posterior distribution with unknown hyperparameters, minimizing the Kullback–Leibler (KL) divergence from the true posterior distribution while simultaneously estimating the hyperparameters and approximating the posterior distribution. The notion of estimating the hyperparameters in the prior distribution from the data stems from empirical Bayes, a concept introduced by Herbert Robbins in the 1950s

(Robbins, 1992). Empirical Bayes methods have since been extensively studied and applied in various domains (Efron and Morris, 1973; Efron et al., 2001; Carlin and Louis, 2008; Atchadé, 2011; Efron, 2012). Instead, minimizing the KL divergence between the variational distribution and the true posterior distribution is derived from variational inference (Hinton and Van Camp, 1993; Blei and Lafferty, 2007; Blundell et al., 2015; Zhang et al., 2018). For training, we employ the gradient ascent algorithm along with Monte Carlo estimates to estimate both the variational posterior distribution and the unknown hyperparameters in the prior. This combination enables the NA-EB framework to be well suited for complex models and large-scale datasets. NA-EB shares some similarity with Atanov et al. (2018) that we both use a variational approximation for the true posterior distribution of the weights and propose an implicit generative prior for the weights. On the other hand, NA-EB is distinguished from Atanov et al. (2018) in terms of the objective function, the training procedure for the parameters of the implicit prior, the assumption about the parametric form of the implicit prior, and the establishment of theoretical guarantees.

Our contributions can be summarized as follows.

- We propose a novel approach to defining the prior distribution through a DNN transformation of a known low-dimensional distribution. This method provides a highly flexible prior that can capture complex high-dimensional distributions and incorporate intrinsic low-dimensional structure with ease. This approach represents a significant advancement in Bayesian modeling and inference, as it addresses the challenge of defining meaningful priors for neural networks, which has been a bottleneck in practical applications.

- Theoretically, we perform an asymptotic analysis in terms of posterior consistency (Bhattacharya et al., 2020; Bhattacharya and Maiti, 2021), which quantifies the quality of the resulting posterior as data are collected indefinitely. In contrast to this previous work, we establish the uniform posterior consistency for a class of nonlinear transformations when defining the prior. Furthermore, we establish the classification accuracy of variational Bayes DNNs. Therefore, our framework is guaranteed to provide a numerically stable and theoretically consistent solution.

- Empirically, we evaluate the finite sample performance through simulated data analysis and real data applications, including the classical two-spiral problem, synthetic regression data, 10 UCI datasets, MNIST dataset, noisy MNIST dataset, and CIFAR-10 dataset. We extensively compare our method with other methods including SGD (Rumelhart et al., 1986), variational Bayesian (Blundell et al., 2015), probabilistic back-propagation (Hernández-Lobato and Adams, 2015), dropout (Gal and Ghahramani, 2016), ensembles Bayesian (Lakshminarayanan et al., 2017), sparse variational Bayesian (Bai et al., 2020), variational Bayesian with collapsed bound (Tomczak et al., 2021), conditional generative models (Zhou et al., 2022), and diffusion models (Han et al., 2022). As a result, the proposed method outperforms these existing methods on predictive accuracy and uncertainty quantification in both regression and classification tasks as shown in Section 5. The source code implementations are available in the Appendix.

This paper is organized as follows. In Section 2, we give an overview of the proposed framework. In Section 3, we present the computational algorithm for NA-EB. In Section

4, we establish that our algorithm is theoretically guaranteed in terms of variational posterior consistency and classification accuracy. In Section 5, we illustrate the performance of our model through simulation studies and real-life data analysis. We conclude our paper in Section 6 with discussions. Further details about our algorithm, instructions for utilizing the code files, and the assumptions and the outline of the proofs of theorems and corollaries are given in the Appendix.

2 Implicit Generative Prior

Let $\mathcal{D}_n = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ represent the training data with sample size n, where $y_i \in \mathcal{Y}$ is the response of interest and $\boldsymbol{x}_i \in \mathcal{X} \subseteq \mathbb{R}^p$ is the covariate. As in the classical setting of supervised learning, $(\boldsymbol{x}_i, y_i)$, for $i = 1, \ldots, n$, are assumed to be i.i.d. from an unknown joint distribution $P_{\boldsymbol{x},y}$ on $\mathcal{X} \times \mathcal{Y}$. Consider a probabilistic model $p(y \mid \boldsymbol{x}; \boldsymbol{w})$, where $\boldsymbol{w} \in \mathscr{W} \subseteq \mathbb{R}^D$ denotes the vector of unknown parameter. For example, for regression, $p(y \mid \boldsymbol{x}; \boldsymbol{w})$ can be a Gaussian distribution with an unknown mean that is modeled by a DNN with weights $\boldsymbol{w}$. For classification, $p(y \mid \boldsymbol{x}; \boldsymbol{w})$ is the categorical distribution in which the success probabilities are modeled by an unknown multivariate function with parameter $\boldsymbol{w}$. Let $L(\mathcal{D}_n; \boldsymbol{w})$ be the joint conditional distribution of y_i given $\boldsymbol{x}_i$, where $\boldsymbol{w}$ is the unknown parameter. Bayesian inference will introduce a prior $\pi(\boldsymbol{w})$ on $\boldsymbol{w}$, and compute the posterior distribution of the weights given the training data such as $p(\boldsymbol{w} \mid \mathcal{D}_n) \propto \pi(\boldsymbol{w}) L(\mathcal{D}_n; \boldsymbol{w})$. The predictive distribution of a future y^* given a test data $\boldsymbol{x}^*$ can be obtained by

$$p(y^* \mid \boldsymbol{x}^*, \mathcal{D}_n) = \int p(y^* \mid \boldsymbol{x}^*; \boldsymbol{w}) p(\boldsymbol{w} \mid \mathcal{D}_n) d\boldsymbol{w}. \tag{1}$$

This is equivalent to an ensemble method, which uses an infinite number of models for prediction where the parameters of each model are sampled from the posterior distribution.

We specify $\pi(\boldsymbol{w})$ through a highly flexible *implicit* model defined by a two-step procedure, where firstly a latent variable $\boldsymbol{z} \in \mathscr{Z} \subseteq \mathbb{R}^r$ with $r \leq D$ is sampled from a fixed distribution $\pi_0(\boldsymbol{z})$, and then $\boldsymbol{z}$ is mapped to $\boldsymbol{w} = G_{\boldsymbol{\eta}}(\boldsymbol{z})$ via a *deterministic* transformation $G_{\boldsymbol{\eta}} : \mathbb{R}^r \to \mathbb{R}^D$ with hyperparameter $\boldsymbol{\eta} \in \mathbb{R}^{n_\eta}$. This defines a class of priors using the push-forward measure,

$$\Pi = \{G_{\boldsymbol{\eta}} \# \pi_0 : G_{\boldsymbol{\eta}} \in \mathcal{G}\},$$

where $\mathcal{G}$ is the function space for the transformation function. When $G_{\boldsymbol{\eta}}$ is invertible and $r = D$, we recover the familiar rule of transformation of probability distributions. The prior distribution $\pi(\boldsymbol{w})$ for this situation has an explicit formula by the change-of-variable technique. We are interested in developing more general and flexible cases where $G_{\boldsymbol{\eta}}$ is a nonlinear function with $r \leq D$. Under this circumstance, the explicit density of $\boldsymbol{w}$ is intractable, since the set $\{G_{\boldsymbol{\eta}}(\boldsymbol{z}) \leq \boldsymbol{w}\}$ cannot be determined. The advantages of the above formulation are the following: (a) $G_{\boldsymbol{\eta}}(\boldsymbol{z})$ with $\boldsymbol{z} \sim \pi_0$ can represent a wide range of distributions. In fact, for any continuous random vector $\boldsymbol{w}$ in a low-dimensional manifold, there always exists a $G_{\boldsymbol{\eta}}$ such that $G_{\boldsymbol{\eta}}(\boldsymbol{z})$ has the same distribution as $\boldsymbol{w}$ (Chen et al., 2022). (b) For many problems, it is reasonable to believe that the high-dimensional $\boldsymbol{w}$ lies in a low-dimensional manifold and the dimension r can be much smaller than D. (c) The mapping $G_{\boldsymbol{\eta}}$ is unconstrained, considerably simplifying the functional optimization problem.

Without loss of generality, we choose a normal distribution on each entry for $\boldsymbol{z} = (z_1, \ldots, z_r)^{\mathrm{T}}$ of the form

$$\pi_0(\boldsymbol{z}) = \prod_{j=1}^{r} \frac{1}{\sqrt{2\pi\zeta_j^2}} \exp\left\{-\frac{1}{2\zeta_j^2}(z_j - \mu_j)^2\right\}, \tag{2}$$

where $\boldsymbol{z} = (z_1, \ldots, z_r)^{\mathrm{T}}$ and each z_j requires a separate mean μ_j and variance ζ_j. We will provide conditions on $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_r)^{\mathrm{T}}$ and $\boldsymbol{\zeta} = (\zeta_1, \ldots, \zeta_r)^{\mathrm{T}}$ in Section 4 to ensure the posterior consistency of both Bayesian and variational approaches.

If $\boldsymbol{\eta}$ is known, the posterior distribution of $\boldsymbol{z}$ given $\mathcal{D}_n$ is

$$p_{\boldsymbol{\eta}}(\boldsymbol{z} \mid \mathcal{D}_n) = \frac{L(\mathcal{D}_n; G_{\boldsymbol{\eta}}(\boldsymbol{z}))\pi_0(\boldsymbol{z})}{\int L(\mathcal{D}_n; G_{\boldsymbol{\eta}}(\boldsymbol{z}))\pi_0(\boldsymbol{z})d\boldsymbol{z}}. \tag{3}$$

Selecting the hyperparameter $\boldsymbol{\eta}$, in particular, the high-dimensional hyperparameter, is very difficult. Instead, the empirical Bayes method will estimate $\boldsymbol{\eta}$ from the data based on the marginal likelihood, and obtain

$$\hat{\boldsymbol{\eta}} = \arg\max_{\boldsymbol{\eta}} L_{\mathtt{marg}}(\boldsymbol{\eta}; \mathcal{D}_n), \tag{4}$$

where the marginal likelihood is given by

$$L_{\mathtt{marg}}(\boldsymbol{\eta}; \mathcal{D}_n) = \int_{\mathscr{W}} L(\mathcal{D}_n; \boldsymbol{w})\pi(\boldsymbol{w})d\boldsymbol{w} = \int_{\mathscr{Z}} L(\mathcal{D}_n; G_{\boldsymbol{\eta}}(\boldsymbol{z}))\pi_0(\boldsymbol{z})d\boldsymbol{z}. \tag{5}$$

However, the main difficulty of the optimization problem (4) is evaluating (5) and its gradient with respect to $\boldsymbol{\eta}$. The exact evaluation of $L_{\mathtt{marg}}(\boldsymbol{\eta}; \mathcal{D}_n)$ is intractable since, in general, the integral is high-dimensional and does not have a closed form. In the next section, we introduce a novel algorithm based on the variational method to efficiently estimate $\boldsymbol{\eta}$ and approximate the posterior distribution.

3 The Algorithm

The conventional MCMC implementation suffers from high computational cost, which limits its use for large scale problems. To avoid computational issues, we adopt the variational approach to estimate $\boldsymbol{\eta}$ and derive the posterior distribution. The key difference between the current setting and the regular variational inference is that there involves an additional unknown hyperparameter $\boldsymbol{\eta}$ in the prior. Variational inference starts from a variational family, which is used to approximate the true posterior distribution $p_{\boldsymbol{\eta}}(\boldsymbol{z} \mid \mathcal{D}_n)$ in (3). Given several options, we work with a simple and computationally tractable variational family, which is a mean field Gaussian variational family of the form

$$\mathcal{Q} = \left\{ q_{\boldsymbol{\alpha}}(\boldsymbol{z}) : q_{\boldsymbol{\alpha}}(\boldsymbol{z}) = \prod_{j=1}^{r} \frac{1}{\sqrt{2\pi\varrho_j^2}} \exp\left\{-\frac{1}{2\varrho_j^2}(z_j - m_j)^2\right\} \right\},$$

where $\boldsymbol{\alpha} = (m_1, \ldots, m_r, \varrho_1, \ldots, \varrho_r)^{\mathrm{T}} \in \mathbb{R}^{2r}$ represents all unknown parameters in $q_{\boldsymbol{\alpha}}$.

Instead of maximizing the marginal log-likelihood $\log L_{\mathtt{marg}}(\boldsymbol{\eta}; \mathcal{D}_n)$ in (5), we maximize the evidence lower bound (ELBO), defined by

$$\mathcal{L}(\boldsymbol{\alpha}, \boldsymbol{\eta}) := \mathrm{ELBO}(\boldsymbol{\alpha}, \boldsymbol{\eta}) = \log L_{\mathtt{marg}}(\boldsymbol{\eta}; \mathcal{D}_n) - \mathrm{KL}\left(q_{\boldsymbol{\alpha}}(z) \parallel p_{\boldsymbol{\eta}}(z \mid \mathcal{D}_n)\right), \tag{6}$$

Algorithm 1 Stochastic gradient method for updating $\boldsymbol{\alpha}$ and $\boldsymbol{\eta}$

```
Input: Training data $\mathcal{D}_n = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, learning rate sequence $\{\beta^{(t)}\}$, sample size $H$
Output: Parameter estimates $(\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\eta}})$
1: Initialization: $\boldsymbol{\alpha}^{(0)}$, $\boldsymbol{\eta}^{(0)}$
2: for $t = 0, \dots, T-1$ do
3:     Simulate $H$ samples $\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(H)}$ from $q_{\boldsymbol{\alpha}^{(t)}}(.)$
4:     Compute
           $\nabla_{\boldsymbol{\alpha}^{(t)}} \mathcal{L}(\boldsymbol{\alpha}^{(t)}, \boldsymbol{\eta}^{(t)}) = H^{-1} \sum_{h=1}^{H} [\nabla_{\boldsymbol{\alpha}^{(t)}} \log\{q_{\boldsymbol{\alpha}^{(t)}}(\boldsymbol{z}^{(h)})\}] * (\mathrm{I})$
               where $(\mathrm{I}) = \log\{L(\mathcal{D}_n; G_{\boldsymbol{\eta}^{(t)}}(\boldsymbol{z}^{(h)}))\} + \log\{\pi_0(\boldsymbol{z}^{(h)}) / q_{\boldsymbol{\alpha}^{(t)}}(\boldsymbol{z}^{(h)})\}$
           $\nabla_{\boldsymbol{\eta}^{(t)}} \mathcal{L}(\boldsymbol{\alpha}^{(t)}, \boldsymbol{\eta}^{(t)}) = H^{-1} \sum_{h=1}^{H} \nabla_{\boldsymbol{\eta}^{(t)}} \log\{L(\mathcal{D}_n; G_{\boldsymbol{\eta}^{(t)}}(\boldsymbol{z}^{(h)}))\}$
5:     update
           $\boldsymbol{\alpha}^{(t+1)} = \boldsymbol{\alpha}^{(t)} + \beta^{(t)} \nabla_{\boldsymbol{\alpha}^{(t)}} \mathcal{L}(\boldsymbol{\alpha}^{(t)}, \boldsymbol{\eta}^{(t)})$
           $\boldsymbol{\eta}^{(t+1)} = \boldsymbol{\eta}^{(t)} + \beta^{(t)} \nabla_{\boldsymbol{\eta}^{(t)}} \mathcal{L}(\boldsymbol{\alpha}^{(t)}, \boldsymbol{\eta}^{(t)})$
6: end for
7: return $\hat{\boldsymbol{\alpha}} = \boldsymbol{\alpha}^{(T)}$, $\hat{\boldsymbol{\eta}} = \boldsymbol{\eta}^{(T)}$
```

where the last term in (6) is the KL divergence between the variation posterior $q_{\boldsymbol{\alpha}}(\boldsymbol{z})$ and true posterior $p_{\boldsymbol{\eta}}(\boldsymbol{z} \mid \mathcal{D}_n)$, which is always nonnegative. So, $\mathcal{L}(\boldsymbol{\alpha}, \boldsymbol{\eta})$ is a uniform lower bound for $\log L_{\mathtt{marg}}(\boldsymbol{\eta}; \mathcal{D}_n)$. If $\mathcal{L}(\boldsymbol{\alpha}, \boldsymbol{\eta})$ is taken as the objective function to maximize, then the result corresponds to variational inference. It is straightforward to demonstrate that the ELBO can be simplified as

$$\mathcal{L}(\boldsymbol{\alpha}, \boldsymbol{\eta}) = -\mathrm{KL}\left(q_{\boldsymbol{\alpha}}(\boldsymbol{z}) \parallel \pi_0(\boldsymbol{z})\right) + \mathbb{E}_{\boldsymbol{z} \sim q_{\boldsymbol{\alpha}}(\boldsymbol{z})}\left[\log\{L(\mathcal{D}_n; G_{\boldsymbol{\eta}}(\boldsymbol{z}))\}\right]. \tag{7}$$

Note that the ELBO in (7) can be evaluated efficiently. This is because the first KL term in (7) has an explicit solution, since both $q_{\boldsymbol{\alpha}}$ and π_0 are normal densities, and the second term can be unbiasedly estimated by the Monte Carlo average. Let $(\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\eta}})$ be the maximizer of (7). Then, $q^* := q_{\hat{\boldsymbol{\alpha}}}$ is the estimated variational posterior for $\boldsymbol{z}$ and the push-forward measure $\pi^* := G_{\hat{\boldsymbol{\eta}}} \# q^*$ is the empirical Bayes variational posterior for the weights $\boldsymbol{w}$, which is the approximation of the true posterior $p(\boldsymbol{w} \mid \mathcal{D}_n)$ in (1).

In practice, the gradient ascent will be adopted to obtain the estimates, and our algorithm computes the gradients as

$$\begin{aligned} \nabla_{\boldsymbol{\alpha}} \mathcal{L}(\boldsymbol{\alpha}, \boldsymbol{\eta}) &= \mathbb{E}_{\boldsymbol{z} \sim q_{\boldsymbol{\alpha}}(\boldsymbol{z})} \Big([\nabla_{\boldsymbol{\alpha}} \log\{q_{\boldsymbol{\alpha}}(\boldsymbol{z})\}] \left[\log\{L(\mathcal{D}_n; G_{\boldsymbol{\eta}}(\boldsymbol{z}))\} + \log\left\{ \tfrac{\pi_0(\boldsymbol{z})}{q_{\boldsymbol{\alpha}}(\boldsymbol{z})} \right\} \right] \Big), \\ \nabla_{\boldsymbol{\eta}} \mathcal{L}(\boldsymbol{\alpha}, \boldsymbol{\eta}) &= \mathbb{E}_{\boldsymbol{z} \sim q_{\boldsymbol{\alpha}}(\boldsymbol{z})} \left[\nabla_{\boldsymbol{\eta}} \log\{L(\mathcal{D}_n; G_{\boldsymbol{\eta}}(\boldsymbol{z}))\} \right]. \end{aligned} \tag{8}$$

The detailed algorithm is given in Algorithm 1. This is an iterative algorithm that updates the estimated $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\eta}}$ at each iteration. In practice, for variational parameters $(\varrho_1, \dots, \varrho_r)$, we apply the reparameterization trick $\varrho_j = \log(1 + e^{\rho_j})$, for $j = 1, \dots, r$, and update the quantities ρ_j in each step instead of ϱ_j as this guarantees non-negative estimates of standard deviation ϱ_j (Ranganath et al., 2014; Blundell et al., 2015). Further details about the algorithm can be found in the Appendix.

4 Theoretical Analysis

In this section, we investigate the theoretical properties of the variational posterior. We have established the uniform variational posterior consistency over a class of transformations $G_{\boldsymbol{\eta}}$ with some smoothness conditions on $G_{\boldsymbol{\eta}}$. We have also established the classification accuracy of the variational posterior.

4.1 Consistency of variational posterior

We will mainly focus on binary classification, and the conditional density of y given $\boldsymbol{x}$ under the truth is

$$\ell_0(y, \boldsymbol{x}) = \exp\{y f_0(\boldsymbol{x}) - \log(1 + e^{f_0(\boldsymbol{x})})\}, \tag{9}$$

where $f_0 : \mathbb{R}^p \mapsto \mathbb{R}$ is an unknown continuous function of the log-odds ratio. Let $f_{\boldsymbol{w}}$ be a neural network approximation of f_0 with network weights $\boldsymbol{w}$. Write

$$\ell_{\boldsymbol{w}}(y, \boldsymbol{x}) = \exp\{y f_{\boldsymbol{w}}(\boldsymbol{x}) - \log(1 + e^{f_{\boldsymbol{w}}(\boldsymbol{x})})\}. \tag{10}$$

Without loss of generality, assume $\boldsymbol{x}_i \sim Unif[0,1]^p$, for $i = 1, 2, \ldots, n$, which implies $p(\boldsymbol{x}) = 1$ and $p(\boldsymbol{x} \mid \boldsymbol{w}) = 1$. Define the Hellinger neighborhood of the true density function $g_0 = \ell_0$ under the true model f_0 as

$$\mathcal{U}_\varepsilon = \big\{\boldsymbol{w} : d_{\mathrm{H}}(\ell_0, \ell_{\boldsymbol{w}}) < \varepsilon\big\}, \tag{11}$$

where the Hellinger distance $d_{\mathrm{H}}(\ell_0, \ell_{\boldsymbol{w}})$ is expressed by

$$d_{\mathrm{H}}(\ell_0, \ell_{\boldsymbol{w}}) = \left[\frac{1}{2}\int_{\boldsymbol{x}\in[0,1]^p} \sum_{y\in\{0,1\}} \left\{\sqrt{\ell_0(y,\boldsymbol{x})} - \sqrt{\ell_{\boldsymbol{w}}(y,\boldsymbol{x})}\right\}^2 d\boldsymbol{x}\right]^{\frac{1}{2}}.$$

Similarly, the Kullback–Leibler neighborhood of the true density function ℓ_0 under the truth f_0 is defined as

$$\mathcal{N}_\varepsilon = \big\{\boldsymbol{w} : d_{\mathrm{KL}}(\ell_0, \ell_{\boldsymbol{w}}) < \varepsilon\big\},$$

where the KL distance $d_{\mathrm{KL}}(\ell_0, \ell_{\boldsymbol{w}})$ is given by

$$d_{\mathrm{KL}}(\ell_0, \ell_{\boldsymbol{w}}) = \int_{\boldsymbol{x}\in[0,1]^p} \sum_{y\in\{0,1\}} \left[\log\left\{\frac{\ell_0(y,\boldsymbol{x})}{\ell_{\boldsymbol{w}}(y,\boldsymbol{x})}\right\}\ell_0(y,\boldsymbol{x})\right] d\boldsymbol{x}.$$

In the following, we use the notation P_0 to denote the true distribution of $\mathcal{D}_n = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ under the true density ℓ_0. Regarding the asymptotic analysis of NA-EB, we assume that both r and D, the dimensions of $\boldsymbol{z}$ and $\boldsymbol{w}$, respectively, depend on n and thus rewrite $r = r_n$, $D = D_n$.

Assume $G_{\boldsymbol{\eta}} \in \mathcal{G}$ and we put some smoothness conditions on the functions in $\mathcal{G}$ through the constraints on the Jacobian of the function. Details of the conditions are given in Assumption 1 of the Appendix. The intuition is to improve the stability of the model, which avoids the situation where infinitesimal perturbations amplify and have substantial impacts on the performance of the output of $G_{\boldsymbol{\eta}}$. In practice, a Jacobian regularization can be added to the objective function, and a computationally efficient algorithm has been implemented by Hoffman et al. (2019). The prior parameters in (2) satisfy $\|\boldsymbol{\mu}\|_2^2 = o(n)$ and

$$\|\boldsymbol{\zeta}\|_\infty = O(n), \quad \|\boldsymbol{\zeta}^*\|_\infty = O(1), \tag{12}$$

where $\boldsymbol{\zeta}^* = (1/\zeta_1, \ldots, 1/\zeta_r)^{\mathrm{T}}$. This assumption, which is Assumption 2 of the Appendix, imposes restrictions on the prior parameters so that the KL distance between the variational posterior q^* and the true posterior $p(\boldsymbol{w} \mid \mathcal{D}_n)$ is negligible.

Theorem 4.1. *Suppose $r_n \sim n^a$, $D_n \sim n^u$, $0 < a \leq u < 1$. Then, under Assumptions 1 and 2 in the Appendix,*

$$\sup_{G_{\boldsymbol{\eta}} \in \mathcal{G}} \pi^*(\mathcal{U}_\varepsilon^c) \xrightarrow{P_0} 0.$$

Theorem 4.1 indicates that, under some regularity conditions, for any $G_{\boldsymbol{\eta}} \in \mathcal{G}$ and any $\nu > 0$, $\pi^*(\mathcal{U}_\varepsilon^c) < \nu$ with probability tending to 1 as $n \to \infty$. Under the conditions of Theorem 4.1 with less restrictive assumptions on $G_{\boldsymbol{\eta}}$, it can be proved that the true posterior satisfies $p(\mathcal{U}_\varepsilon^c \mid \mathcal{D}_n) < 2e^{-n\varepsilon^2/2}$ with probability tending to 1 as $n \to \infty$ as shown in Theorem F.1 part 1 of the Appendix. This implies that the probability of the ε-small Hellinger neighborhood of the true function ℓ_0 for the true posterior increases at a rate of $1 - 2e^{-n\varepsilon^2/2}$ in contrast to the slow rate of $1 - \nu$ for the variational posterior. On the other hand, the consistency of the variational posterior requires more conditions on the implicit model $G_{\boldsymbol{\eta}}$ than that of the true posterior, since the Bayesian posterior is hard to compute due to the intractable integrals.

Although we have established the uniform posterior variational consistency for any transformation function $G_{\boldsymbol{\eta}}$, our numerical experiences demonstrate that better predictive performance is achieved using the estimated $\boldsymbol{\eta}$ of our algorithm than a randomly or manually selected $\boldsymbol{\eta}$.

To establish the consistency of the variational posterior in a shrinking Hellinger neighborhood of ℓ_0, we need to modify Assumptions 1 and 2. Compared to Assumptions 1 and 2, the square of the Frobenius norm of the Jacobian in Assumption 3 and the rate of growth of L_2 norm of the prior mean parameter in Assumption 4 are allowed to grow slower since the consistency of the variational posterior in a shrinking Hellinger neighborhood of ℓ_0 is more restrictive in nature. Furthermore, it requires the existence of a neural network solution that converges to the true function ℓ_0 at a sufficiently fast rate while ensuring controlled growth of the L_2 norm of its coefficients. These assumptions are summarized in Assumptions 3, 4, and 5 of the Appendix.

Theorem 4.2. *Suppose $r_n \sim n^a$, $D_n \sim n^u$, $0 < a \leq u < 1$ and $\epsilon_n^2 \sim n^{-\delta}$, $0 < \delta < 1 - u \leq 1 - a$. Then, under Assumptions 3, 4, and 5 of the Appendix,*

$$\sup_{G_{\boldsymbol{\eta}} \in \mathcal{G}} \pi^*(\mathcal{U}_{\varepsilon\epsilon_n}^c) \xrightarrow{P_0} 0.$$

A restatement of Theorem 4.2 is for any $G_{\boldsymbol{\eta}} \in \mathcal{G}$ and any $\nu > 0$, $\pi^*(\mathcal{U}_{\varepsilon\epsilon_n}^c) < \nu$ with probability tending to 1 as $n \to \infty$. Under the conditions of Theorem 4.2 with less restrictive assumptions on $G_{\boldsymbol{\eta}}$, it can be established that the true posterior satisfies $p(\mathcal{U}_{\varepsilon\epsilon_n}^c \mid \mathcal{D}_n) < 2e^{-n\varepsilon^2\epsilon_n^2/2}$ with probability tending to 1 as $n \to \infty$ as shown in Theorem F.2 part 1 of the Appendix. This implies that the probability of the $\varepsilon\epsilon_n$-small Hellinger neighborhood of the true function ℓ_0 for the true posterior increases at the rate of $1 - 2e^{-n\varepsilon^2\epsilon_n^2/2}$ in contrast to the slow rate of $1 - \nu$ for the variational posterior.

4.2 Classification accuracy

In this subsection, we establish the classification accuracy of the variational posterior. A classifier C is a Borel-measurable function $C : \mathbb{R}^p \mapsto \{0, 1\}$, which assigns a sample

$\boldsymbol{x} \in \mathbb{R}^p$ to the class $C(\boldsymbol{x})$. The misclassification error risk of a classifier C is given by

$$R(C) = \int_{\mathbb{R}^p \times \{0,1\}} \mathbb{1}(C(\boldsymbol{x}) \neq y) dP_{\boldsymbol{x},y}. \tag{13}$$

where $\mathbb{1}(\cdot)$ denotes the indicator function. The Bayes classifier is defined as

$$C^{\mathsf{Bayes}}(\boldsymbol{x}) = \begin{cases} 1, & \sigma(f_0(\boldsymbol{x})) \geq 1/2 \\ 0, & \text{otherwise}, \end{cases} \tag{14}$$

where $\sigma(x) = e^x/(1+e^x)$ is the sigmoid function.

As the predictive distribution of Algorithm 1 can be estimated in (16) of the Appendix, the classifier based on the variational posterior is given by

$$\hat{C}(\boldsymbol{x}) = \begin{cases} 1, & \sigma(\hat{f}(\boldsymbol{x})) \geq 1/2 \\ 0, & \text{otherwise}, \end{cases} \tag{15}$$

where $\hat{f}(\boldsymbol{x}) = \sigma^{-1}(\int \sigma(f_{G_{\hat{\boldsymbol{\eta}}}(\boldsymbol{z})}(\boldsymbol{x}))q^*(\boldsymbol{z})d\boldsymbol{z})$ is the variational estimator of $f_0(\boldsymbol{x})$.

Although the Bayes classifier is optimal in terms of minimizing the test error in (13) (Hastie et al., 2009), it is not useful in practice, as the truth f_0 is unknown. We compare the classification accuracy of the Bayes classifier in (14) and the variational classifier in (15) in Corollaries 4.1 and 4.2 under different assumptions on the prior parameters and the deterministic transformation function $G_{\boldsymbol{\eta}}$.

Corollary 4.1. *Under the conditions of Theorem 4.1,*

$$\sup_{G_{\boldsymbol{\eta}} \in \mathcal{G}} \left| R(\hat{C}) - R(C^{\mathsf{Bayes}}) \right| \xrightarrow{P_{\boldsymbol{x},y}} 0.$$

Corollary 4.1 can be rephrased as for any $G_{\boldsymbol{\eta}} \in \mathcal{G}$ and any $\nu > 0$, $|R(\hat{C}) - R(C^{\mathsf{Bayes}})| < \nu$ with probability tending to 1 as $n \to \infty$. As part 2 of Theorem F.1 in the Appendix shows that the true posterior gives the classification accuracy at the same consistency rate under the conditions of Theorem 4.1 with less restrictive assumptions on $G_{\boldsymbol{\eta}}$ which indicates that there is no harm using the variational posterior approximation.

Corollary 4.2. *Under the conditions of Theorem 4.2, for every* $0 \leq \kappa \leq \frac{2}{3}$

$$\sup_{G_{\boldsymbol{\eta}} \in \mathcal{G}} \epsilon_n^{-\kappa} \left| R(\hat{C}) - R(C^{\mathsf{Bayes}}) \right| \xrightarrow{P_{\boldsymbol{x},y}} 0.$$

A restatement of Corollary 4.2 is for any $G_{\boldsymbol{\eta}} \in \mathcal{G}$ and any $\nu > 0$, $0 \leq \kappa \leq \frac{2}{3}$, $|R(\hat{C}) - R(C^{\mathsf{Bayes}})| < \nu\epsilon_n^{\kappa}$ with probability tending to 1 as $n \to \infty$. As part 2 of Theorem F.2 in the Appendix shows, under the conditions of Theorem 4.2 with less restrictive assumptions on $G_{\boldsymbol{\eta}}$, the true posterior satisfies $\epsilon_n^{-\kappa}|R(\hat{C}) - R(C^{\mathsf{Bayes}})| \xrightarrow{P_{\boldsymbol{x},y}} 0$ for every $0 \leq \kappa \leq 1$.

5 Numerical Results

In this section, we evaluate our NA-EB in regression and classification experiments. For these two classical types of supervised learning problems, we illustrate the performance of our proposed framework and algorithm in terms of predictive accuracy and predictive uncertainty with real data analysis and synthetic datasets, respectively.

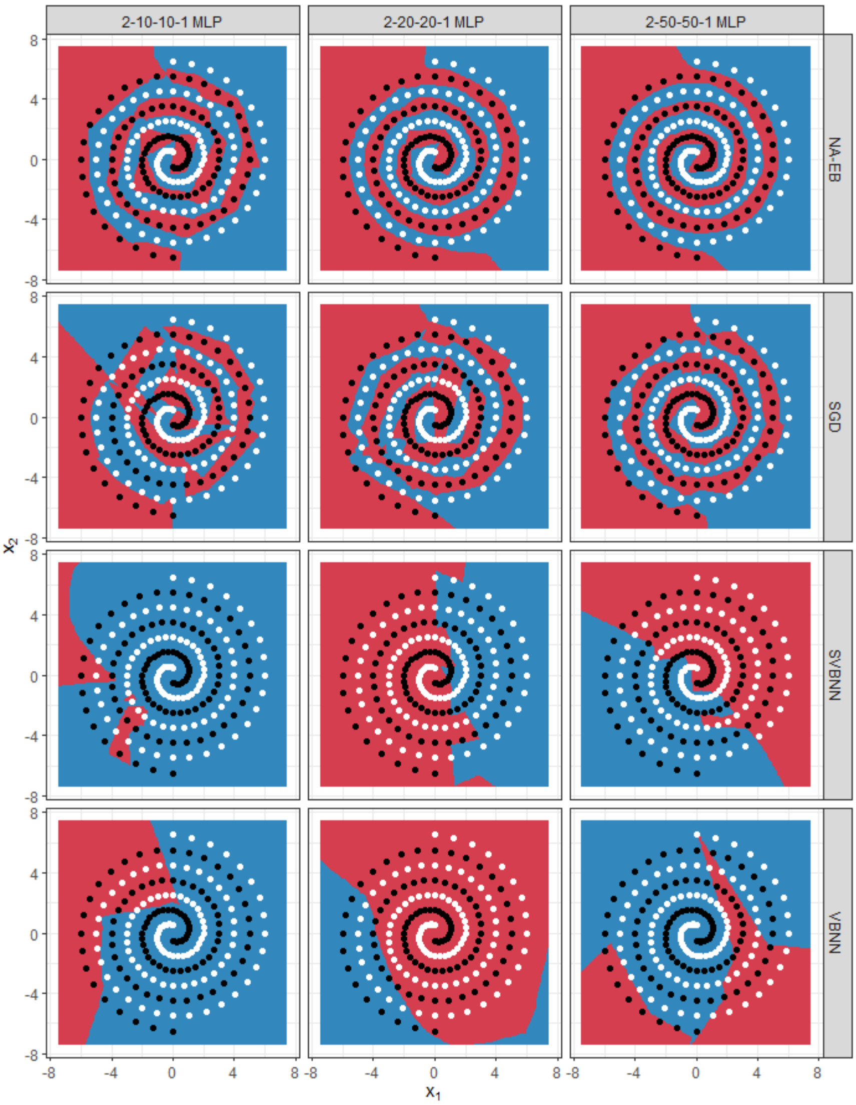


Figure 3: NA-EB, SGD, VBNN, and SVBNN classifying maps using 2-10-10-1 MLP, 2-20-20-1 MLP, and 2-50-50-1 MLP respectively. Black and white points denote training data for two spirals. Red and blue regions indicate the two classified classes.

5.1 Two-spiral problem

Consider the classification problem of learning a mapping for the two-spiral dataset $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{194}$ in which the samples $(\boldsymbol{x}_i, y_i) \in \mathbb{R}^2 \times \mathbb{R}$ are generated from:

$$\begin{aligned}
\boldsymbol{x}_{i,1} &= 6.5 \times (-1)^{\{i \pmod 2\}} \times \left[1 - \tfrac{i-\{i \pmod 2\}}{208}\right] \times \sin\left(\tfrac{[i-\{i \pmod 2\}]\pi}{32}\right), \\
\boldsymbol{x}_{i,2} &= 6.5 \times (-1)^{\{i \pmod 2\}} \times \left[1 - \tfrac{i-\{i \pmod 2\}}{208}\right] \times \cos\left(\tfrac{[i-\{i \pmod 2\}]\pi}{32}\right), \\
y_i &= i \pmod 2,
\end{aligned}$$

where $i \pmod 2$ is the remainder after dividing i by 2, for $i = 1, \ldots, 194$, and the sample points on two intertwined spirals are shown in Figure 3.

Since no structural information is assumed for the mapping, a feedforward neural network, which is also known as the multiple layer perceptron (MLP), can be used to

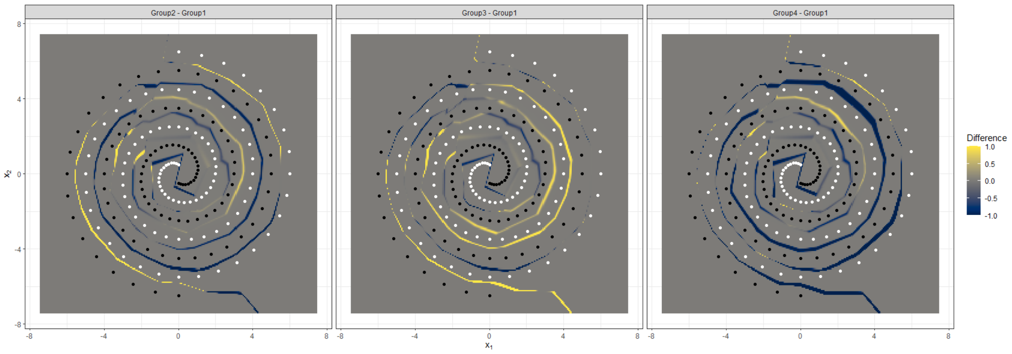


Figure 4: Difference maps of the predicted probability from four groups of weights sampled from the variational posterior of NA-EB. The yellow and dark blue areas in each map indicate the regions classified as different class by two groups of weights, respectively.

distinguish between points. We adopt fully connected two-hidden-layer MLPs consisting of two input units, h hidden units for both layers, one output unit, and the ReLu activation function denoted by the 2-h-h-1 MLP ($h = 10, 20, 50$). Besides, we employ a one-hidden-layer MLP comprising 128 rectified linear units as the deterministic transformation function $G_{\boldsymbol{\eta}}$. For comparison, the results of a standard neural network optimized by stochastic gradient descent via backpropagation (SGD) (Rumelhart et al., 1986), a variational Bayesian algorithm (VBNN) (Blundell et al., 2015), and a sparse variational BNN (SVBNN) (Bai et al., 2020) are also reported in Figure 3. The comparison indicates that NA-EB outperforms SGD, SVBNN, and VBNN in terms of predictive performance. NA-EB can find perfect solutions that distinguish between points on two intertwined spirals with smooth boundaries for different architectures of MLPs. However, SGD performs rather poorly as the number of hidden units of MLPs decreases. We also observe in this example that VBNN and SVBNN have not converged well as it requires the MLP to learn a highly nonlinear separation of the input space.

Furthermore, we sample weights from the learned variational posterior distribution multiple times and then compare their classification maps to illustrate the uncertainty in the weights. In particular, we generate 4 groups of weights $\{\boldsymbol{w}^{(1)}, \boldsymbol{w}^{(2)}, \boldsymbol{w}^{(3)}, \boldsymbol{w}^{(4)}\}$ of the 2-20-20-1 MLP from the variational posterior and display the maps of differences between predicted probability from these groups $\{p(f_{\boldsymbol{w}^{(v)}}(\boldsymbol{x}) = 1 \mid \boldsymbol{x}) - p(f_{\boldsymbol{w}^{(1)}}(\boldsymbol{x}) = 1 \mid \boldsymbol{x})\}_{v=2}^{4}$ respectively in Figure 4. As can be seen, the absolute values of the probability differences tend to be 1 in the middle of two intertwined spirals and 0 elsewhere. This indicates that the variational posterior prefers to be uncertain in the middle of two spirals, as it is reasonable to classify this region as either of two classes.

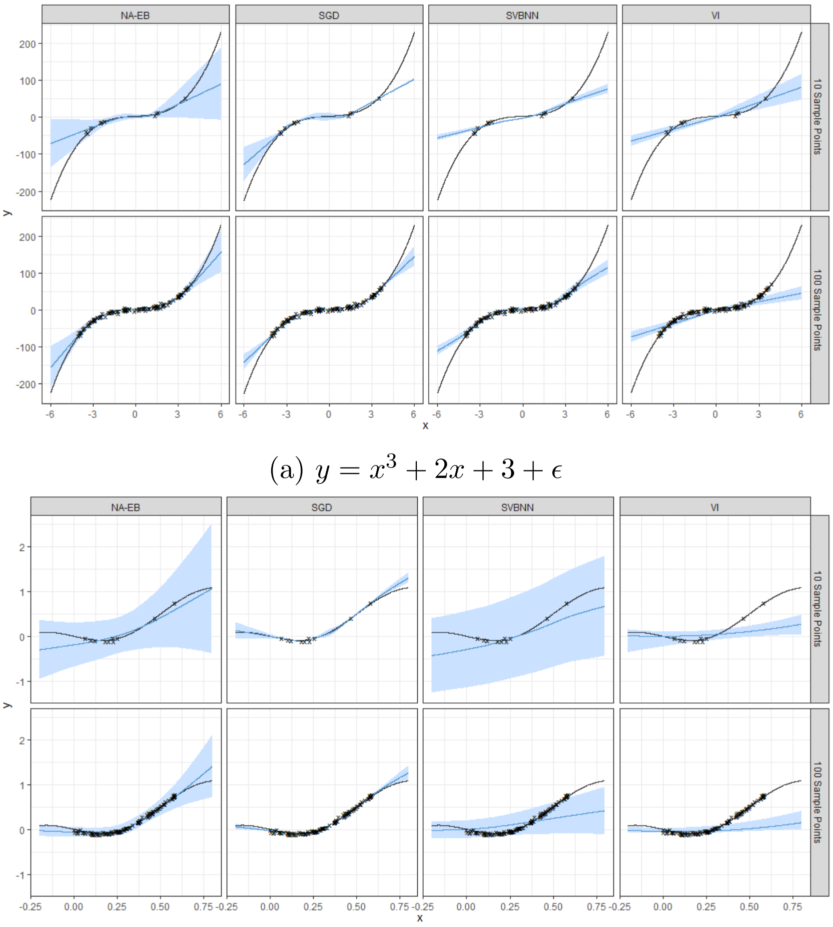
5.2 Synthetic 1D Experiments

In this subsection, we consider synthetic regression problems and demonstrate the predictive distribution obtained by NA-EB in toy datasets. We generate two datasets that consist of different nonlinear functions. We sample the inputs x uniformly from the interval $[-4, 4]$ and then generate training data from the first curve:

$$y = x^3 + 2x + 3 + \epsilon,$$

where $\epsilon \sim N(0, 9)$. We also generate sample points from the second curve:

$$y = x - 0.3\sin(2\pi x) + \epsilon,$$



(a) $y = x^3 + 2x + 3 + \epsilon$

(b) $y = x - 0.3\sin(2\pi x) + \epsilon$

Figure 5: Regression of two toy datasets with credible intervals using 10 sample points and 100 sample points made by NA-EB, SGD, SVBNN and VI.

where the inputs x are uniformly sampled from the interval $[0, 0.6]$ and $\epsilon \sim N(0, 0.02^2)$. We compare our method with the standard stochastic gradient descent via backpropagation (SGD) (Rumelhart et al., 1986), with the sparse variational BNN (SVBNN) (Bai et al., 2020) and with a variational inference (VI) approach (Graves, 2011). The neural network architecture includes two hidden layers with 100 and 50 hidden units for each hidden layer. Besides, we use a one-hidden-layer MLP with 128 rectified linear units as the deterministic transformation function $G_{\boldsymbol{\eta}}$. We use 300 training epochs for all methods on the training data.

Figures 5a and 5b show the predictions and credible intervals generated by each method. Noisy training samples from two datasets are shown as black crosses, with true data-generating functions depicted by black continuous lines and mean predictions shown as dark blue lines. The light blue areas represent credible intervals at ±3 standard deviations. In the sample interval, NA-EB and SGD give mean predictions that are much closer to the true data-generating function than VI and SVBNN given the same sample size for both datasets. On the other hand, in the regions of the input space where there are no data, NA-EB generates a diverged confidence interval reflecting that there are many possible extrapolations, while SGD fits a specific curve with the variance almost reduced to zero. This indicates that NA-EB prefers to be uncertain when nearby data is unavailable, in contrast to a standard neural network that can be overly confident. Furthermore, as the number of sample points increases, the prediction accuracy of NA-EB improves, and the approximated uncertainty of NA-EB decreases.

5.3 UCI datasets

We further evaluate the predictive accuracy and uncertainty quantification of NA-EB on real-world datasets for regression tasks. We adopt the same set of 10 UCI regression benchmark datasets (Dua and Graff, 2017) as well as the experimental protocol proposed in Hernández-Lobato and Adams (2015) and followed by Gal and Ghahramani (2016); Lakshminarayanan et al. (2017); Han et al. (2022). These datasets are available at `https://archive.ics.uci.edu/datasets`.

Each data set is randomly divided into training and test sets with 90% and 10% of the data, respectively. The splitting process is repeated 20 times for all datasets except that for Year dataset and Protein dataset, we do the train-test splitting only one and five times, respectively, due to their large sample sizes. Also, we normalize all datasets so that the input features and targets have zero mean and unit variance in the training set, and remove the normalization for evaluation. Furthermore, for both the Kin8nm and Naval dataset, we multiply the response variable by 100 which is the same as Han et al. (2022) to match the scale of other datasets. We compare our method with four BNN methods: probabilistic back-propagation (PBP) (Hernández-Lobato and Adams, 2015), the dropout uncertainty (MC Dropout) approach (Gal and Ghahramani, 2016), the deep ensembles (Ensembles) Bayesian method (Lakshminarayanan et al., 2017), and the sparse variational BNN (SVBNN) (Bai et al., 2020). We also compare our method with two deep generative model approaches: the generative conditional distribution sampler (GCDS) (Zhou et al., 2022) and classification and regression diffusion models (CARD) (Han et al., 2022). To be consistent with the results summarized in Han et al. (2022), we adopt the same experimental setup: a two-hidden-layer network architecture with 100 and 50 hidden units, the ReLU activation function, the Adam optimizer, the batch size varied case by case and 500 training epochs. Besides, we specify the deterministic transformation function $G_{\boldsymbol{\eta}}$ by a two-hidden-layer MLP comprising 128 rectified linear units for each layer.

Recent BNN approaches (Hernández-Lobato and Adams, 2015; Gal and Ghahramani, 2016; Lakshminarayanan et al., 2017) employ the negative log-likelihood (NLL) to quantify uncertainty. However, NLL computation assumes a Gaussian density for the conditional distributions $p(y \mid \boldsymbol{x}; \boldsymbol{w})$ for all $\boldsymbol{x}$, which may not hold for real-world datasets. Therefore, we adopt the quantile interval coverage error (QICE) metric proposed by Han et al. (2022) as a measure of uncertainty for regression tasks. Simultaneously, for classification tasks that will be discussed in detail in the subsequent sections, we maintain the use of NLL to quantify uncertainty, aligning with the metrics reported in classification experiments of Han et al. (2022). As stated in Han et al. (2022), QICE is defined as the mean absolute error between the proportion of true data contained within each quantile interval of generated samples of size N and the ideal proportion:

$$\text{QICE} := \frac{1}{M} \sum_{m=1}^{M} \left| r_m - \frac{1}{M} \right|, \quad \text{where } r_m = \frac{1}{N} \sum_{n=1}^{N} \mathbb{1}(y_n \geq \hat{y}_n^{\texttt{low}_\texttt{m}}) \mathbb{1}(y_n \leq \hat{y}_n^{\texttt{high}_\texttt{m}}),$$

where $\hat{y}_n^{\texttt{low}_\texttt{m}}$ and $\hat{y}_n^{\texttt{high}_\texttt{m}}$ represent the low and high percentiles of the mth quantile interval, respectively, of our choice for the predicted y_n outputs, for $n = 1, \ldots, N$, given the same $\boldsymbol{x}$

Table 1: Test RMSE of UCI regression tasks. Boldface indicates the method with the smallest test RMSE.

Dataset	Average Test RMSE and Standard Errors						
	PBP	MC Dropout	Ensembles	SVBNN	GCDS	CARD	NA-EB
# Parameters/Hyperparameters	$\sim 2D$	$\sim D$	$\sim 5D$	$\sim 3D$	$\sim 2D$	$\sim D$	$\sim 100D$
Boston	2.89 ± 0.74	3.06 ± 0.96	3.17 ± 1.05	3.17 ± 0.57	2.75 ± 0.58	2.61 ± 0.63	**2.18 ± 0.27**
Concrete	5.55 ± 0.46	5.09 ± 0.60	4.91 ± 0.47	5.57 ± 0.47	5.39 ± 0.55	4.77 ± 0.46	**3.87 ± 0.59**
Energy	1.58 ± 0.21	1.70 ± 0.22	2.02 ± 0.32	1.92 ± 0.19	0.64 ± 0.09	0.52 ± 0.07	**0.39 ± 0.05**
Kin8nm	9.42 ± 0.29	7.10 ± 0.26	8.65 ± 0.47	9.37 ± 0.26	8.88 ± 0.42	**6.32 ± 0.18**	6.74 ± 0.13
Naval	0.41 ± 0.08	0.08 ± 0.03	0.09 ± 0.01	0.21 ± 0.05	0.14 ± 0.05	0.02 ± 0.00	**0.02 ± 0.00**
Power	4.10 ± 0.15	4.04 ± 0.14	4.02 ± 0.15	4.01 ± 0.18	4.11 ± 0.16	3.93 ± 0.17	**3.52 ± 0.14**
Protein	4.65 ± 0.02	4.16 ± 0.12	4.45 ± 0.02	4.30 ± 0.05	4.50 ± 0.02	3.73 ± 0.01	**3.65 ± 0.04**
Wine	0.64 ± 0.04	0.62 ± 0.04	0.63 ± 0.04	0.62 ± 0.04	0.66 ± 0.04	0.63 ± 0.04	**0.57 ± 0.04**
Yacht	0.88 ± 0.22	0.84 ± 0.27	1.19 ± 0.49	1.10 ± 0.27	0.79 ± 0.26	0.65 ± 0.25	**0.23 ± 0.05**
Year	8.86 ± NA	8.77 ± NA	8.79 ± NA	8.87 ± NA	9.20 ± NA	**8.70 ± NA**	8.76 ± NA

Table 2: Test QICE (in %) of UCI regression tasks. Boldface indicates the method with the smallest test QICE.

Dataset	Average Test QICE and Standard Errors						
	PBP	MC Dropout	Ensembles	SVBNN	GCDS	CARD	NA-EB
Boston	3.50 ± 0.88	3.82 ± 0.82	3.37 ± 0.00	4.18 ± 1.24	11.73 ± 1.05	3.45 ± 0.83	**3.36 ± 0.73**
Concrete	2.52 ± 0.60	4.17 ± 1.06	2.68 ± 0.64	3.50 ± 0.76	10.49 ± 1.01	**2.30 ± 0.66**	2.51 ± 0.66
Energy	6.54 ± 0.90	5.22 ± 1.02	3.62 ± 0.58	5.49 ± 0.58	7.41 ± 2.19	4.91 ± 0.94	**4.89 ± 0.82**
Kin8nm	1.31 ± 0.25	1.50 ± 0.32	1.17 ± 0.22	5.87 ± 0.45	7.73 ± 0.80	**0.92 ± 0.25**	1.38 ± 0.26
Naval	4.06 ± 1.25	12.50 ± 1.95	6.64 ± 0.60	**0.78 ± 0.28**	5.76 ± 2.25	0.80 ± 0.21	3.90 ± 1.06
Power	**0.82 ± 0.19**	1.32 ± 0.37	1.09 ± 0.26	1.07 ± 0.28	1.77 ± 0.33	0.92 ± 0.21	1.00 ± 0.33
Protein	1.69 ± 0.09	2.82 ± 0.41	2.17 ± 0.16	1.22 ± 0.21	2.33 ± 0.18	**0.71 ± 0.11**	0.96 ± 0.19
Wine	2.22 ± 0.64	2.79 ± 0.56	2.37 ± 0.63	2.55 ± 0.65	3.13 ± 0.79	3.39 ± 0.69	**2.15 ± 0.71**
Yacht	6.93 ± 1.74	10.33 ± 1.34	7.22 ± 1.41	8.40 ± 1.70	5.01 ± 1.02	8.03 ± 1.17	**4.99 ± 1.42**
Year	2.96 ± NA	2.43 ± NA	2.56 ± NA	1.64 ± NA	1.61 ± NA	**0.53 ± NA**	1.52 ± NA

input. Ideally, when the learned conditional distribution perfectly matches the true one, the QICE value should be 0. QICE is an empirical metric that does not impose Gaussian restrictions or any specific parametric form on the conditional distribution. Similar to NLL, it relies on the summary statistics of samples from the learned distribution to empirically evaluate the similarity between the learned and true conditional distributions. To be consistent with the results presented in Han et al. (2022), we use the same parameter $M = 10$ quantile intervals to calculate QICE.

Tables 1 and 2 summarize the average test root mean squared error (RMSE) and QICE with their standard deviation across all splits for each method, respectively. We observe that NA-EB obtains the best results in 8 out of 10 datasets in terms of RMSE and 4 out of 10 for QICE, and it is competitive with the best method for the remaining datasets. It should be noted that although we do not explicitly optimize our model with respect to MSE or QICE, we still outperform existing models trained with these objectives. We also list the number of unknowns for all alternative methods in Table 1 for fair comparison. It is important to note that we assume that all methods employ the same base classifier, with the weights of this classifier being of dimension D. The key difference between NA-EB and other BNN methods is how we define the prior. In NA-EB, we define the implicit prior by $\boldsymbol{w} = G_{\boldsymbol{\eta}}(\boldsymbol{z})$, where $\boldsymbol{z}$ follows a standard Gaussian, and $\boldsymbol{\eta}$ is the so-called hyperparameter in statistics. A subjective Bayesian will choose a known $\boldsymbol{\eta}$, but the performance is poor in this setting. Instead, we incorporate the concept of empirical Bayes to estimate these hyperparameters from the data. This will cause another approximately $100D$ hyperparameters in our UCI examples.

Table 3: Test RMSE of 4 UCI regression datasets based on NA-EB using different latent dimensions. Boldface indicates the latent dimension with the smallest test RMSE.

Dataset	Sample Size	Feature Dimension	Average Test RMSE and Standard Errors			
			Latent Dimension			
			20	60	100	140
Yacht	308	6	0.25 ± 0.09	**0.23 ± 0.05**	0.23 ± 0.07	0.24 ± 0.08
Wine	1599	11	0.59 ± 0.04	0.60 ± 0.04	**0.57 ± 0.04**	0.58 ± 0.04
Power	9568	4	3.56 ± 0.15	**3.52 ± 0.14**	3.52 ± 0.18	3.59 ± 0.16
Protein	45730	9	3.73 ± 0.03	**3.65 ± 0.04**	3.72 ± 0.03	3.72 ± 0.02

Table 4: RMSE of 4 UCI regression datasets based on NA-EB using different parameterizations of the deterministic transformation function $G_{\boldsymbol{\eta}}$ with or without Jacobian regularization. Boldface indicates the architecture of $G_{\boldsymbol{\eta}}$ with the smallest RMSE.

Dataset	Average Test RMSE and Standard Errors			
	Architecture of Transformation Function $G_{\boldsymbol{\eta}}$			
	r-128-D MLP	r-64-64-D MLP	r-128-128-D MLP	r-128-128-D MLP Jacobian Regularization
Yacht	0.45 ± 0.12	0.24 ± 0.09	**0.23 ± 0.05**	0.33 ± 0.12
Wine	0.60 ± 0.04	0.60 ± 0.04	**0.57 ± 0.04**	0.60 ± 0.05
Power	3.70 ± 0.19	3.62 ± 0.21	**3.52 ± 0.14**	3.54 ± 0.17
Protein	3.75 ± 0.06	3.68 ± 0.06	**3.65 ± 0.04**	3.72 ± 0.05

Furthermore, we conduct experiments on UCI regression datasets to demonstrate that NA-EB is robust to the dimension r of the latent variable $\boldsymbol{z}$. We select four datasets with various sample sizes and feature dimensions and consider different latent dimensions in an appropriate range. As shown in Table 3, the average test RMSE changes insignificantly with different latent dimensions. With our network architecture, the empirical results show that NA-EB obtains the best result when the latent dimension is approximately 10 times the feature dimension and its predictive performance is robust within a suitable range of the latent dimension.

We further investigate the impact of different parameterizations of the deterministic transformation function $G_{\boldsymbol{\eta}}$ on the model performance. As described previously, we employ a fully connected feed-forward neural network with r input nodes and D output nodes as $G_{\boldsymbol{\eta}}$. We explore different parameterizations of $G_{\boldsymbol{\eta}}$ by changing the number of hidden layers and the number of hidden units within a layer. The test RMSE results using different architectures of $G_{\boldsymbol{\eta}}$ obtained from 20 runs on the four UCI datasets are summarized in the first three columns of Table 4. Our findings indicate that the r-128-128-D MLP architecture consistently achieves the lowest test RMSE results across all select UCI datasets. This suggests that higher complexity of this architecture allows for a more versatile and flexible representation of $G_{\boldsymbol{\eta}}$, resulting in better predictive accuracy. However, it's worth noting that the test RMSE results are quite similar between the two-hidden-layer MLPs transformations, implying the robust predictive performance of NA-EB within a reasonable range of parameterization complexities of $G_{\boldsymbol{\eta}}$.

In addition, as the asymptotic analysis of NA-EB in Section 4 suggests, the square of the Frobenius norm of the input-output Jacobian $\|J(\boldsymbol{z})\|_F^2 = \|\partial G_{\boldsymbol{\eta}}/\partial \boldsymbol{z}\|_F^2$ shall be constrained. To effectively incorporate Jacobian regularization into the training process,

Table 5: Test error rates (in %) of MNIST classification tasks with different base classifiers. Boldface indicates the method with the smallest test error.

Base Classifier	Test Error			
	SGD	VBNN	SVBNN	NA-EB
400-400 MLP	1.83	1.36	1.40	**1.24**
800-800 MLP	1.84	1.34	1.37	**1.22**
1200-1200 MLP	1.88	1.32	1.36	**1.21**
LeNet-5	1.14	31.01	4.08	**0.91**

we optimize a joint loss function that aligns with (5) from Hoffman et al. (2019):

$$\mathcal{L}_{\texttt{joint}}(\boldsymbol{\alpha}, \boldsymbol{\eta}) = \mathcal{L}(\boldsymbol{\alpha}, \boldsymbol{\eta}) + \frac{\lambda_{\texttt{JR}}}{2} \|J(\boldsymbol{z})\|_F^2,$$

where $\mathcal{L}(\boldsymbol{\alpha}, \boldsymbol{\eta})$ is the ELBO as specified in (6) of our paper and $\lambda_{\texttt{JR}}$ is a hyperparameter that controls the relative significance of the Jacobian regularizer. Hoffman et al. (2019) provides a computationally efficient implementation of Jacobian regularization $J(\boldsymbol{z})$. We follow its PyTorch implementation available at `https://github.com/facebookresearch/jacobian_regularizer` and set values of the hyperparameter $\lambda_{\texttt{JR}} = 0.1$ by default. The test RMSE incorporating Jacobian regularization on the four datasets is reported in the last column of Table 4. Interestingly, the numerical results reveal that both approaches based on the same transformation function architecture yield similar outcomes. Consequently, we opt to proceed without Jacobian regularization to reduce computational costs.

5.4 MNIST dataset

We demonstrate our experimental results on the MNIST digits dataset, consisting of 60,000 training and 10,000 test pixel images of size 28 by 28. Similarly to Lakshminarayanan et al. (2017), our motivation for classification is to improve the performance of a base classifier in terms of accuracy through a generative model on the benchmark datasets instead of achieving the state-of-the-art predictive performance, as the latter is strongly related to network architecture design. Here, we shall focus on improving the performance of an ordinary feed-forward neural network of various sizes without using convolutions, distortions, etc. Besides, we also provide empirical results for NA-EB based on the LeNet-5 (LeCun et al., 1998) architecture involving convolutional networks to highlight its efficacy in the robustness of the choice of base classifier architectures. Specifically, we repeat the protocol and the network architecture of Blundell et al. (2015) that an MLP of two hidden layers of h rectified linear units and a softmax output layer with 10 units, denoted by the h-h MLP ($h = 400, 800, 1200$), is trained with the Adam optimizer using a learning rate of 10^{-4} and minibatches of size 128 for 600 training epochs. On the other hand, we use the Adam optimizer, set the learning rate to 10^{-3}, employ minibatches of size 32, and conduct training over 100 epochs in the experiment based on the LeNet-5 architecture. In both experiments utilizing different base classifiers, we use a feed-forward neural network with one hidden layer of 128 rectified linear units as the deterministic transformation $G_{\boldsymbol{\eta}}$ for NA-EB. Following Blundell et al. (2015), we preprocess the pixels by dividing the values by 126.

Table 6: Test NLL of MNIST classification tasks with different base classifiers. Boldface indicates the method with the smallest test NLL.

Base Classifier	Test NLL		
	VBNN	SVBNN	NA-EB
400-400 MLP	0.118	0.144	**0.057**
LeNet-5	0.877	0.102	**0.034**

Table 7: The CPU time (in s) of MNIST classification tasks with different base classifiers. Boldface indicates the method with the shortest CPU time.

Base Classifier	Epochs	CPU Time			
		SGD	VBNN	SVBNN	NA-EB
400-400 MLP	600	**8366.89**	47201.29	352780.21	130016.78
LeNet-5	100	**2328.70**	10088.73	44615.34	13957.58

As summarized in Table 5, we compare the test error of our proposed method for different base classifier architectures with the performance of SGD (Simard et al., 2003), sparse variational BNN (SVBNN) (Bai et al., 2020) and variational BNN (VBNN) reported in Blundell et al. (2015). Here, the inclusion of SGD serves as a baseline reference to assess predictive accuracy of NA-EB and other variational BNN methods. Meanwhile, the learning curves on the test set for these methods based on a network with two layers of 1200 rectified linear units and on the LeNet-5 architecture are presented in Figures 6 and 7, respectively. In addition, the test NLL results based on the two base classifiers for NA-EB, VBNN (Blundell et al., 2015) and SVBNN (Bai et al., 2020) are summarized in Table 6. We further report the CPU time cost on a 2.30 GHz computer for each method in Table 7 to compare the computational speed of NA-EB against other variational BNN methods. All these evaluations, carried out using different base classifiers for each method, adhere to the same training setup, respectively, as previously described. From the results presented in Tables 5, 6, and 7, it is evident that regardless of the choice of the base classifier, NA-EB consistently outperforms other BNN methods in terms of predictive accuracy and uncertainty quantification. Additionally, NA-EB ranks the second place among the BNN methods in terms of computation time as it requires more parameters than VBNN to characterize the implicit prior distribution. However, it's noteworthy that when LeNet-5 is used as the base classifier, VBNN tends to converge to a local minimum across trials, considering all hyperparameter options for the mixing coefficient π and variances σ_1^2, σ_2^2 listed in Section 5.1 of Blundell et al. (2015), as shown in Figure 7. On the contrary, NA-EB achieves the lowest test error with reasonable computational cost. Moreover, as can be seen from Figure 6, NA-EB converges the fastest and eventually obtains the lowest test error after 600 epochs when the MLP is employed as the base classifier. SVBNN and VBNN converge to larger test errors at similar rates.

We further investigate the effectiveness of our approach in scenarios with weaker data signals. Specifically, we assess the performance of our method using three artificial datasets collectively referred to as n-MNIST (noisy MNIST) (Basu et al., 2017). These datasets are created by introducing (1) additive white Gaussian noise (AWGN), (2) motion blur, and (3) a combination of AWGN and reduced contrast to the MNIST dataset. In the n-MNIST dataset with AWGN, we employ Gaussian noise with a signal-to-noise

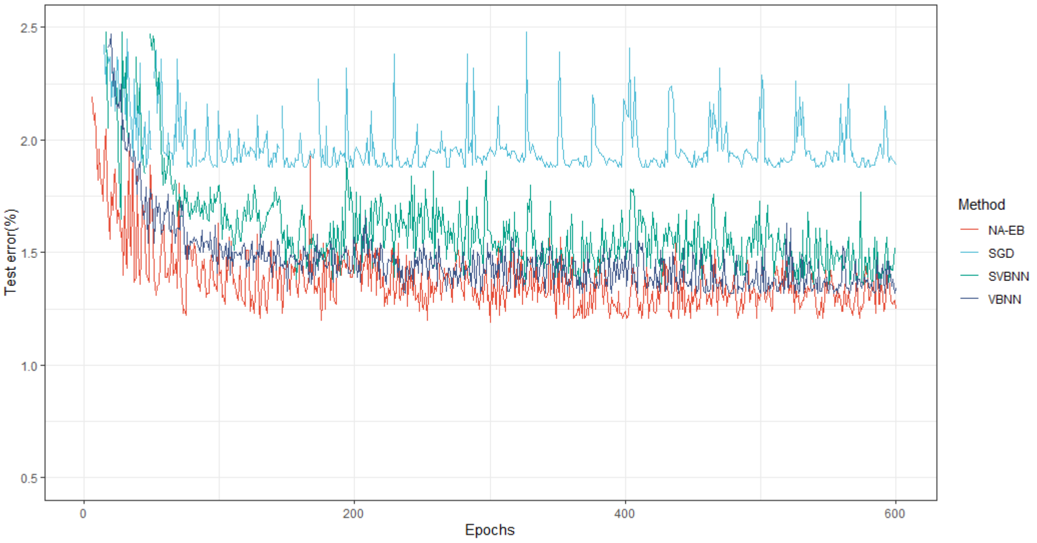


Figure 6: Test error on MNIST based on the 1200-1200 MLP classifier as training progresses: NA-EB, SGD, SVBNN and VBNN.

Table 8: Test error rates (in %) and NLL of classification tasks on different noisy MNIST datasets. Boldface indicates the method with the smallest test error or the smallest NLL.

Dataset	SGD	VBNN		SVBNN		NA-EB	
	Test Error	Test Error	Test NLL	Test Error	Test NLL	Test Error	Test NLL
n-MNIST with AWGN	4.98	4.79	0.17	5.87	0.29	**4.75**	**0.13**
n-MNIST with Motion Blur	1.99	1.96	0.11	1.65	0.28	**1.47**	**0.07**
n-MNIST with reduced contrast and AWGN	8.06	7.23	0.23	9.84	1.27	**7.15**	**0.18**

ratio of 9.5, simulating significant background clutter. For the n-MNIST dataset with motion blur, we apply a motion blur filter to emulate the linear motion of a camera by 5 pixels at an angle of 15 degrees counterclockwise. In the n-MNIST dataset with reduced contrast and AWGN, we scale down the contrast range to half and apply AWGN with a signal-to-noise ratio of 12. This emulates background clutter along with a significant change in lighting conditions. We repeat the training protocol for MNIST classification tasks and summarize the test performance of our method, along with SGD (Simard et al., 2003), VBNN (Blundell et al., 2015), and SVBNN (Bai et al., 2020). All these methods are based on a 400-400 MLP base classifier. The results for the three noisy MNIST datasets are presented in Table 8. It is evident that our method improves the test accuracy and achieves the lowest test NLL regardless of types of noise. Additionally, we present four samples from the noisy MNIST dataset with motion blur in Figure 2 to highlight the significance of predicting uncertainty. In the left two panels, the prediction intervals for the corresponding true labels are narrow and centered around 100%, demonstrating that NA-EB is highly confident about its predictions. Conversely, NA-EB provides relatively wide prediction intervals for the right two figures, particularly in the rightmost panel. Specifically, the top two most likely labels assigned by NA-EB have overlapping and broad prediction intervals, signifying uncertainty in its predictions. In contrast, the standard DNN method assigns an overly confident high prediction probability to an incorrect label in the right two panels.

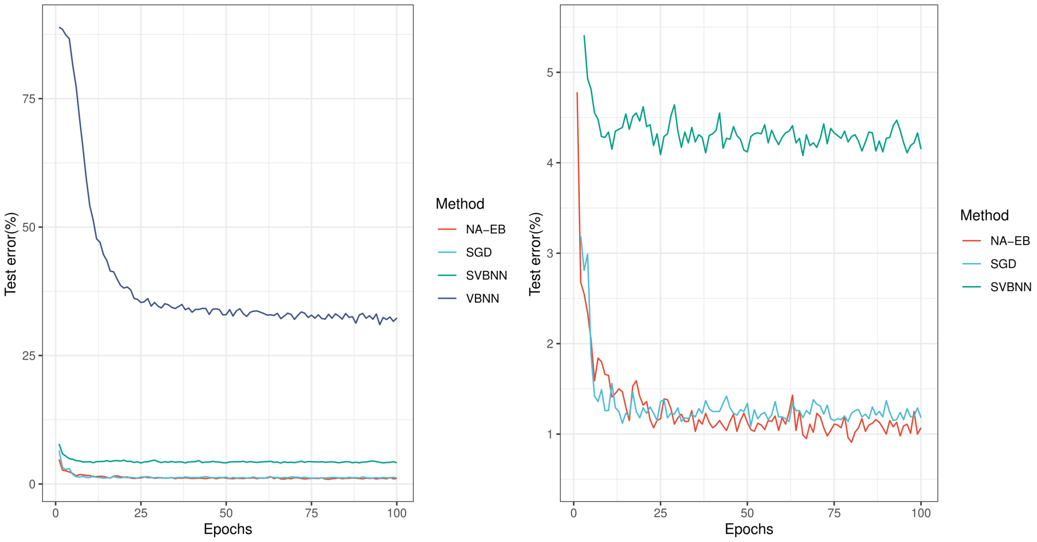


Figure 7: Left: Test error on MNIST based on the LeNet-5 classifier as training progresses: NA-EB, SGD, SVBNN and VBNN. Right: An enlarged figure of the left panel for NA-EB, SGD and SVBNN.

5.5 CIFAR-10 dataset

We evaluate the performance of NA-EB on the CIFAR-10 dataset. This dataset comprises 50,000 training images and 10,000 test images, each sized at 32 × 32 pixels and containing 3 color channels, evenly distributed across 10 distinct classes. We employ ResNet-18 (He et al., 2016), a larger CNN architecture, as the base classifier and optimize the loss function for 200 epochs using the default SGD optimizer with a learning rate of 0.01 and momentum of 0.9. We examine the predictive accuracy and uncertainty quantification of NA-EB in terms of test error and test NLL, respectively. We compare NA-EB with other BNN methods, as reported by Han et al. (2022) in Table 9. Specifically, CMV-MF-VI, CM-MF-VI and CV-MF-VI are variants of a recent BNN work (Tomczak et al., 2021) that employs a novel and tighter ELBO to conduct variational inference in BNNs. Our observations reveal that NA-EB achieves enhanced test accuracy, primarily due to its highly flexible implicit prior. Furthermore, our NLL result, while not the primary objective of NA-EB, is competitive with some of the best performing methods in this regard.

To gain a more tangible grasp of the significance of predictive uncertainty, we juxtapose the predicted probabilities obtained from the ResNet-18 classifier with our 95% Bayesian credible intervals for four test images sourced from the CIFAR-10 dataset. In the case of the deer image, the standard DNN method assigns a high probability to the wrong class without considering uncertainty. Conversely, NA-EB yields prediction intervals of similar width and overlapping for the two potential classes, suggesting uncertainty in its predictions. Moreover, for the frog image, NA-EB provides a relatively narrow prediction interval for the correct class, while the DNN method assigns a high probability estimate to the wrong class. This highlights that NA-EB not only quantifies predictive uncertainty but also improves classification accuracy by implicitly specifying a correct prior for classifier weights. From the examples of the noisy MNIST dataset and CIFAR-10 dataset, it is evident that the widths of the prediction intervals indicate the level of certainty or uncertainty NA-EB holds regarding the accuracy of its predictions, whereas

Table 9: Test error (in %) and NLL of CIFAR-10 classification tasks. Boldface indicates the method with the smallest test error or the smallest NLL.

Metrics	CMV-MF-VI	CM-MF-VI	CV-MF-VI	MF-VI	MC Dropout	MAP	CARD	NA-EB
Test Error	13.75	13.34	20.22	22.92	16.36	15.31	9.07	**8.40**
Test NLL	0.41	**0.39**	0.59	0.68	0.49	0.93	0.46	0.49

the point estimate of likelihood produced by the standard DNN method lacks uncertainty information. It is reasonable to conclude that the superior performance of NA-EB, both in terms of predictive accuracy and uncertainty quantification, can be extended to other classification tasks within the medical domain, such as automated diagnostics on medical images, where uncertainty estimation is of particular importance.

6 Conclusion

In this paper, we propose a general implicit generative prior for Bayesian inference. In particular, we develop the NA-EB framework to address the unsolved challenges in Bayesian DNNs. Meaningful priors for neural network weights are defined implicitly through a deep neural network transformation of a low-dimensional known distribution. The posterior for complex models with a large data volume can be efficiently computed using the proposed stochastic gradient method. We rigorously analyze the theoretical property of the algorithm and demonstrate that NA-EB outperforms many existing methods in a variety of numerical examples. Furthermore, NA-EB also takes advantage of recent developments in computational techniques. Our programming code for NA-EB is implemented using the PyTorch machine learning framework (Paszke et al., 2019), which allows us to construct complex neural networks and compute the gradient of functions using the automatic differentiation technique. Additionally, the NA-EB code can be easily run on modern computational hardware, such as graphics processing units, significantly reducing the computing time for large-scale datasets.

NA-EB has adopted the empirical Bayes framework to estimate the hyperparameter $\boldsymbol{\eta}$, which brings an additional unknown quantity into the model, especially for very large models. We will explore techniques to reduce the unknown-quantity count in NA-EB while maintaining its advantages in uncertainty quantification. This could involve investigating model compression techniques or exploring alternative network architectures.

7 Acknowledgments

The authors would like to thank the anonymous referees, an Associate Editor and the Editor for their constructive comments that improved the quality of this paper. The second author was supported in part by NSF Grant SES-2316428.

Appendix

A Algorithm Details

We provide more details for Algorithm 1. First, all variational parameters $m_1, \dots, m_r$, $\rho_1, \dots, \rho_r$ are initialized by i.i.d. uniform $Unif[-1, 1]$. All components of $\boldsymbol{\eta}$ in $G_{\boldsymbol{\eta}}$ are initialized by a uniform distribution following the default initialization in PyTorch 1.9.0. Then, in each iteration, we compute the Monte Carlo estimates of the gradients in (8) by generating H samples from the current variational posterior distribution. In terms of calculating $\nabla_{\rho_j^{(t)}} \mathcal{L}(\boldsymbol{\alpha}_t, \boldsymbol{\eta}_t)$, for $j = 1, \dots, r$, we apply the chain rule, $\nabla_{\rho_j} \log\{q_{\boldsymbol{\alpha}}(\boldsymbol{z})\} = [\nabla_{\varrho_j} \log\{q_{\boldsymbol{\alpha}}(\boldsymbol{z})\}|_{\varrho_j = \log(1+e^{\rho_j})}] e^{\rho_j}(1+e^{\rho_j})^{-1}$, where the term in the first curly brackets is the derivative of $\log[q_\alpha(z)]$ with respect to ϱ_j evaluated at the point $\varrho_j = \log(1 + e^{\rho_j})$ and the rest is the derivative of ϱ_j with respect to ρ_j. For the next step, we update the estimates of next iteration adopting the gradient descent method whereas the chosen learning rate sequence $\{\beta^{(t)}\}$ follows the Robbins–Monro conditions for convergence (Ranganath et al., 2014). The stopping criteria for Algorithm 1 is set to be a maximum iteration step.

We further investigate an additional improvement on the above algorithm. In the era of big data, our algorithm can also adopt minibatch optimization. The training data $\mathcal{D}_n$ is randomly split into a partition of B subsets $\mathcal{D}_1, \dots, \mathcal{D}_B$, and each gradient is averaged over one subset of the data iteratively (Graves, 2011). For example, one can partition $\mathcal{D}_n$ based on $\tau = (\tau_1, \dots, \tau_B) \in [0, 1]^B$ with $\sum_{b=1}^B \tau_b = 1$. The objective function in (7) is changed to

$$\mathcal{L}_b(\boldsymbol{\alpha}, \boldsymbol{\eta}) = -\tau_b \mathrm{KL}(q_{\boldsymbol{\alpha}} \parallel \pi_0) + \mathbb{E}_{\boldsymbol{z} \sim q_{\boldsymbol{\alpha}}(\boldsymbol{z})}[\log\{L(\mathcal{D}_b; G_{\boldsymbol{\eta}}(\boldsymbol{z}))\}].$$

This is reasonable since $\mathbb{E}_B[\sum_{b=1}^B \mathcal{L}_b(\boldsymbol{\alpha}, \boldsymbol{\eta})] = \mathcal{L}(\boldsymbol{\alpha}, \boldsymbol{\eta})$, where the expectation is taken over the random partitioning of minibatches. In particular, we adopt the scheme $\tau_b = 2^{B-b}/(2^B - 1)$ in our algorithm which puts more weights on the first few minibatches as little data are seen.

As we obtain the estimated DNN transformation function $G_{\hat{\boldsymbol{\eta}}}$ and the variational posterior distribution $q_{\hat{\boldsymbol{\alpha}}}(\boldsymbol{z})$ from Algorithm 1, the predictive distribution can be obtained by multiple forward passes on the network while sampling from the weight posteriors. Given a new input $\boldsymbol{x}^*$, the predictive distribution $p(y^* \mid \boldsymbol{x}^*, \mathcal{D}_n)$ can be estimated by

$$\hat{p}(y^* \mid \boldsymbol{x}^*, \mathcal{D}_n) = \frac{1}{M} \sum_{m=1}^{M} p(y^* \mid \boldsymbol{x}^*, G_{\hat{\boldsymbol{\eta}}}(\tilde{\boldsymbol{z}}_m)) \tag{16}$$

with $\tilde{\boldsymbol{z}}_m$ sampled from $q_{\hat{\boldsymbol{\alpha}}}(\boldsymbol{z})$ independently, for $m = 1, \dots, M$, and M being the number of Monte Carlo samples.

B Instructions for utilizing code files

We provide source code of numerical experiments in Section 5 at `https://github.com/yjliu7/Neural-Adaptive-Empirical-Bayes`. The implementation of NA-EB relies on the PyTorch deep learning framework. In terms of data, we use simulated data in *two_spiral.py* for the two-spiral problem and utilize the MNIST dataset provided by the *torchvision* package. The ten UCI datasets can be downloaded from `https:`

//archive.ics.uci.edu/datasets. Regarding the model, we specify the deterministic transformation function $G_{\boldsymbol{\eta}}$ by a fully connected feedforward neural network in *deterministic_transformation.py.* We provide a Gaussian variational sampler for the latent variable $\boldsymbol{z}$ in *gaussian_variational.py.* The base classifier $f_{\boldsymbol{w}}$ and the loss function $\mathcal{L}(\boldsymbol{\alpha}, \boldsymbol{\eta})$ are given in *Bayes.py.* Furthermore, we include detailed explanations and comments within each Python file for reference.

C Assumptions for Theoretical Analysis

To take advantage of the representation power of the DNN, we approximate f_0 by a DNN $f_{\boldsymbol{w}}$, defined in the following way:

$$\begin{aligned} & f_{\boldsymbol{w}} = A_d \circ \sigma \circ A_{d-1} \circ \cdots \circ \sigma \circ A_1, && (17) \\ & A_1 \in \mathcal{A}_p^N,\ A_2, \ldots, A_{d-1} \in \mathcal{A}_N^N,\ A_d \in \mathcal{A}_N^1, && (18) \\ & \sigma((x_1, \ldots, x_N)^{\mathrm{T}}) = (\sigma(x_1), \ldots, \sigma(x_N))^{\mathrm{T}}, \end{aligned}$$

where N and d are the width and depth of the DNN, respectively, $\mathcal{A}_{n_1}^{n_2}$ stands for affine transformations $\mathbb{R}^{n_1} \mapsto \mathbb{R}^{n_2}$ of the form $\boldsymbol{x}_{n_1 \times 1} \mapsto \boldsymbol{w}_{n_2 \times n_1} \boldsymbol{x} + \boldsymbol{b}_{n_2 \times 1}$, σ is the nonlinear activation function, and $\boldsymbol{w} \in \mathscr{W} \subseteq \mathbb{R}^D$ represents all the parameters in the DNN. In (18) a constant width N is used in each layer merely for simplicity of notations; in practice the dimensions of A_i can vary from layer to layer. Other popular modes such as CNN (Krizhevsky et al., 2017) and ResNet (He et al., 2016) are special cases of (17). DNN models allow for great freedom in selecting the activation function. Popular choices include the rectified linear unit (Glorot et al., 2011), $\sigma(x) = \max\{x, 0\}$, and its smoothed version $\sigma(x) = \log\{1 + \exp(x)\} \approx \max\{x, 0\}$.

Due to the universal approximation property of DNNs (Hornik et al., 1989a), for any $\epsilon > 0$, there exists a DNN $f_{\boldsymbol{\Upsilon}}$ of the form (17) with weights $\boldsymbol{\Upsilon} \in \mathscr{W} \subset \mathbb{R}^{D_n}$ such that $\|f_0 - f_{\boldsymbol{\Upsilon}}\|_\infty \leq \varepsilon/4$, where $\|\cdot\|_\infty$ is the L_∞ norm of a function defined as $\|f\|_\infty = \sup_{\boldsymbol{x} \in \mathcal{X}} |f(\boldsymbol{x})|$. In addition, there exists an $\boldsymbol{s} = (s_1, \ldots, s_{r_n})^{\mathrm{T}} \in \mathscr{Z} \subset \mathbb{R}^{r_n}$ such that $\boldsymbol{\Upsilon} = G_{\boldsymbol{\eta}}(\boldsymbol{s})$. A neighborhood of $\boldsymbol{s}$ is defined as

$$\mathcal{P}_\varepsilon = \left\{ \boldsymbol{z} : |z_j - s_j| < \frac{\sqrt{\varepsilon}}{8\sqrt{r_n n^{1-u}}\{D_n + (p+1)\|\boldsymbol{\Upsilon}\|_1\}}, j = 1, \ldots, r_n \right\}, \tag{19}$$

where $\|\cdot\|_1$ is the L_1 norm of a vector defined as $\|\boldsymbol{\Upsilon}\|_1 = \sum_{k=1}^{D_n} |\Upsilon_k|$ and u is given in Theorems 4.1 and 4.2 measuring the order of magnitude of D_n. Let $\{\mathcal{F}_n\}$ denote a sequence of sieves defined as

$$\mathcal{F}_n = \{\boldsymbol{z} : |z_j| \leq \tilde{C}_n, j = 1, \ldots, r_n\}, \tag{20}$$

where $\tilde{C}_n$ is a parameter related to n. In the following, we write $G_{\boldsymbol{\eta}} = (G_{\boldsymbol{\eta}1}, G_{\boldsymbol{\eta}2}, \ldots, G_{\boldsymbol{\eta}D_n})^{\mathrm{T}}$ where $G_{\boldsymbol{\eta}k} : \mathscr{Z} \mapsto \mathbb{R}$ is the deterministic transformation function such that $w_k = G_{\boldsymbol{\eta}k}(\boldsymbol{z})$, for $k = 1, \ldots, D_n$.

In Theorem 4.1, we establish that under some conditions, the variational posterior concentrates in ε-small Hellinger neighborhoods of the true density ℓ_0. The conditions on the deterministic transformation $G_{\boldsymbol{\eta}}$ and prior parameters are summarized below.

Assumption 1. *The deterministic function $G_{\boldsymbol{\eta}}$ is twice differentiable at $\boldsymbol{z} \in \mathscr{Z}$. For $\boldsymbol{z} \in \mathcal{F}_n \cup \mathcal{P}_\varepsilon$ with $\tilde{C}_n = e^{n^b/r_n}$ where b is a constant, the first-order derivative satisfies*

$\|\partial G_{\boldsymbol{\eta}}/\partial \boldsymbol{z}\|_F^2 = o(\varepsilon n^{1-u})$ where u is given in Theorem 4.1 and $\|\cdot\|_F$ denotes the Frobenius norm of a matrix $\mathbf{A}$ defined as $\|\mathbf{A}\|_F = (\sum_i \sum_j |a_{ij}|^2)^{1/2}$. The second-order derivative satisfies $\{\sum_{k=1}^{D_n}(\partial^2 G_{\boldsymbol{\eta}k}/\partial z_j^2)^2\}^{1/2} = o(\sqrt{D_n}\sum_{k=1}^{D_n}(\partial G_{\boldsymbol{\eta}k}/\partial z_j)^2)$ at $\boldsymbol{s}$, for $j = 1, \ldots, r_n$.

Assumption 2. *The prior parameters in* (2) *satisfy the assumption* (12) *and $\|\boldsymbol{\mu}\|_2^2 = o(n)$, where $\|\cdot\|_2$ is the L_2 norm of a vector defined as $\|\boldsymbol{\mu}\|_2 = (\sum_{j=1}^r \mu_j^2)^{1/2}$.*

Assumption 1 puts some smoothness constraints on the Jacobian of $G_{\boldsymbol{\eta}}$. The intuition is to improve the stability of the model, which avoids the situation where infinitesimal perturbations amplify and have substantial impacts on the performance of the output of $G_{\boldsymbol{\eta}}$. In practice, a Jacobian regularization can be added to the objective function, and a computationally efficient algorithm has been implemented by Hoffman et al. (2019). Furthermore, the second-order derivative condition in Assumption 1 is mild, since it only requires that the square root of the mean square of the second-order derivative be insignificant compared to the sum of the squared first-order derivative at a fixed point $\boldsymbol{s}$. As long as a substantial fraction of $\partial G_{\boldsymbol{\eta}k}/\partial z_j$ among $k = 1, \ldots, D_n$ evaluated at this fixed point is nonzero, the condition is satisfied. Assumption 2 imposes restrictions on the prior parameters so that the KL distance between the variational posterior q^* and the true posterior $p(\boldsymbol{w} \mid \mathcal{D}_n)$ is negligible.

In Theorem 4.2, we establish the consistency of variational posterior for shrinking neighborhood sizes of the true density ℓ_0. However, since Theorem 4.2 is more restrictive in nature than Theorem 4.1, it requires additional assumptions on the approximation of the neural network solution to the true function f_0. Therefore, we modify Assumptions 1 and 2 as follows.

Assumption 3. *The deterministic function $G_{\boldsymbol{\eta}}$ is twice differentiable at $\boldsymbol{z} \in \mathscr{Z}$. For $\boldsymbol{z} \in \mathcal{F}_n \cup \mathcal{P}_{\varepsilon\epsilon_n^2}$ with $\tilde{C}_n = e^{n^b \epsilon_n^2/r_n}$ where b is a constant, the first-order derivative satisfies $\|\partial G_{\boldsymbol{\eta}}/\partial \boldsymbol{z}\|_F^2 = o(\varepsilon\epsilon_n^2 n^{1-u})$ where u is given in Theorem 4.2. The second-order derivative satisfies $\{\sum_{k=1}^{D_n}(\partial^2 G_{\boldsymbol{\eta}k}/\partial z_j^2)^2\}^{1/2} = o(\sqrt{D_n}\sum_{k=1}^{D_n}(\partial G_{\boldsymbol{\eta}k}/\partial z_j)^2)$ at $\boldsymbol{s}$, for $j = 1, \ldots, r_n$.*

Assumption 4. *The prior parameters in* (2) *satisfy the assumption* (12) *and $\|\boldsymbol{\mu}\|_2^2 = o(n\epsilon_n^2)$.*

Assumption 5. *There exists a sequence of neural network functions $f_{\boldsymbol{\Upsilon}}$ of the form* (17) *with $\boldsymbol{\Upsilon} = G_{\boldsymbol{\eta}}(\boldsymbol{s})$ satisfying $\|f_0 - f_{\boldsymbol{\Upsilon}}\|_\infty = o(\epsilon_n^2)$, $\|\boldsymbol{\Upsilon}\|_2^2 = o(n\epsilon_n^2)$, $\|\boldsymbol{s}\|_2^2 = o(n\epsilon_n^2)$.*

Compared to Assumptions 1 and 2, the square of the Frobenius norm of the Jacobian in Assumption 3 and the rate of growth of L_2 norm of the prior mean parameter in Assumption 4 are allowed to grow slower since the consistency of the variational posterior in a shrinking Hellinger neighborhood of ℓ_0 is more restrictive in nature. Furthermore, Assumption 5 requires the existence of a neural network solution that converges to the true function ℓ_0 at a sufficiently fast rate while ensuring controlled growth of the L_2 norm of its coefficients.

D Overview of The Proof

We focus mainly on the proof for the binary classification in (9). Without loss of generality, we consider $f_{\boldsymbol{w}}$ in (17) as a single layer neural network: $f_{\boldsymbol{w}}(\boldsymbol{x}) = \sum_{j=1}^d a_j \sigma(\boldsymbol{\omega}_j{}^{\mathrm{T}}\boldsymbol{x} + b_j)$ where $\boldsymbol{w} = (\boldsymbol{\omega}_1^{\mathrm{T}}, \ldots, \boldsymbol{\omega}_d^{\mathrm{T}}, b_1, \ldots, b_d, a_1, \ldots, a_d)^{\mathrm{T}}$, $a_j, b_j \in \mathbb{R}$, $\boldsymbol{\omega}_j \in \mathbb{R}^p$, $j = 1, \ldots, d$. We briefly outline the main steps in the proof of Theorems 4.1 and 4.2 and Corollaries 4.1 and 4.2. We borrow a few steps and notations from Bhattacharya et al. (2020).

To establish the consistency of variational posterior, we use the following inequality as the foundation of the proof for Theorems 4.1 and 4.2 and its derivations are given in detail in the proof of Theorem 4.1 in the Appendix:

$$-q^*(\mathcal{V}_\varepsilon^c)A \leq d_{\mathrm{KL}}(q^*, p_{\boldsymbol{\eta}}(\cdot \mid \mathcal{D}_n)) + |B| + \log 2, \tag{21}$$

where $\mathcal{V}_\varepsilon$ is defined in (23) and

$$A = \log\left\{\int_{\mathcal{V}_\varepsilon^c} \tfrac{L(\mathcal{D}_n; f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})})}{L_0}\pi_0(\boldsymbol{z})d\boldsymbol{z}\right\}, \quad B = -\log\left\{\int \tfrac{L(\mathcal{D}_n; f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})})}{L_0}\pi_0(\boldsymbol{z})d\boldsymbol{z}\right\}. \tag{22}$$

Then we get the following main steps towards the proof:

1. We construct the upper bound of the first term A by decomposing its exponential term as

   $$e^A = \int_{\mathcal{V}_\varepsilon^c \cap \mathcal{F}_n} \{L(\mathcal{D}_n; f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})})/L_0\}\pi_0(\boldsymbol{z})d\boldsymbol{z} + \int_{\mathcal{V}_\varepsilon^c \cap \mathcal{F}_n^c} \{L(\mathcal{D}_n; f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})})/L_0\}\pi_0(\boldsymbol{z})d\boldsymbol{z},$$

   where $\{\mathcal{F}_n\}_{n=1}^\infty$ is a suitably chosen sequence of sieves (see the proof of Proposition 5 in the Appendix), which gives the lower bound of $-q^*(\mathcal{V}_\varepsilon^c)A$.

2. We control the second quantity B by the rate at which the prior π_0 gives mass to a shrinking KL neighborhood of the true density ℓ_0 (see step 1 (c) of the proof of Proposition 6 in the Appendix for Theorem 4.1 and step 2 (c) of the proof of Proposition 6 in the Appendix for Theorem 4.2).

3. We construct the upper bound for $d_{\mathrm{KL}}(q^*, p_{\boldsymbol{\eta}}(\cdot \mid \mathcal{D}_n))$ by bounding $d_{\mathrm{KL}}(q, p_{\boldsymbol{\eta}}(\cdot \mid \mathcal{D}_n))$ below by

   $$d_{\mathrm{KL}}(q, \pi_0) + \left|\int \log\{L(\mathcal{D}_n; f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})})/L_0\}q(\boldsymbol{z})d\boldsymbol{z}\right| + \left|\log\left[\int \{L(\mathcal{D}_n; f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})})/L_0\}\pi_0(\boldsymbol{z})d\boldsymbol{z}\right]\right|$$

   for a suitable $q \in \mathcal{Q}$ (see the proof of part 1 of Proposition 6 in the Appendix for Theorem 4.1 and the proof of part 2 of Proposition 6 in the Appendix for Theorem 4.2).

4. Based on the relation (21) and the results from steps 1, 2 and 3, we can control $q^*(\mathcal{V}_\varepsilon^c)$, which is equal to $\pi^*(\mathcal{U}_\varepsilon^c)$.

In addition, in terms of the proof for Corollaries 4.1 and 4.2, we first derive that the difference in classification accuracy $R(\hat{C}) - R(C^{\mathsf{Bayes}})$ is bounded above by the logit links $\hat{f}(\boldsymbol{x})$ and $f_0(\boldsymbol{x})$. We further bound the logit links by $d_{\mathrm{H}}(\hat{\ell}, \ell_0)$. Finally, we establish that $R(\hat{C}) - R(C^{\mathsf{Bayes}})$ is bounded by a constant with probability tending to 1 as the sample size increases to infinity.

E Theoretical Properties of Variational Posterior

E.1 Proof of Theorem 4.1

First, we define the Hellinger neighborhood of the true function density function $g_0 = \ell_0$ in terms of $\boldsymbol{z}$ as

$$\mathcal{V}_\varepsilon = \{\boldsymbol{z} : G_{\boldsymbol{\eta}}(\boldsymbol{z}) \in \mathcal{U}_\varepsilon\}. \tag{23}$$

In view of (11) and (23), we notice that

$$\pi^*(\mathcal{U}_\varepsilon^c) = q^*(\mathcal{V}_\varepsilon^c).$$

Therefore, we now turn to prove the posterior consistency of $q^*(\boldsymbol{z})$.
Next, we construct the main inequality (21) related to $q^*(\boldsymbol{z})$ as follows:

$$\begin{aligned}
d_{\mathrm{KL}}(q^*, p_{\boldsymbol{\eta}}(\cdot \mid \mathcal{D}_n)) &= \int_{\mathcal{V}_\varepsilon} q^*(\boldsymbol{z}) \log\left\{\frac{q^*(\boldsymbol{z})}{p_{\boldsymbol{\eta}}(\boldsymbol{z} \mid \mathcal{D}_n)}\right\} d\boldsymbol{z} + \int_{\mathcal{V}_\varepsilon^c} q^*(\boldsymbol{z}) \log\left\{\frac{q^*(\boldsymbol{z})}{p_{\boldsymbol{\eta}}(\boldsymbol{z} \mid \mathcal{D}_n)}\right\} d\boldsymbol{z} \\
&= -q^*(\mathcal{V}_\varepsilon) \int_{\mathcal{V}_\varepsilon} \frac{q^*(\boldsymbol{z})}{q^*(\mathcal{V}_\varepsilon)} \log\left\{\frac{p_{\boldsymbol{\eta}}(\boldsymbol{z} \mid \mathcal{D}_n)}{q^*(\boldsymbol{z})}\right\} d\boldsymbol{z} \\
&- q^*(\mathcal{V}_\varepsilon^c) \int_{\mathcal{V}_\varepsilon^c} \frac{q^*(\boldsymbol{z})}{q^*(\mathcal{V}_\varepsilon^c)} \log\left\{\frac{p_{\boldsymbol{\eta}}(\boldsymbol{z} \mid \mathcal{D}_n)}{q^*(\boldsymbol{z})}\right\} d\boldsymbol{z} \\
&\geq q^*(\mathcal{V}_\varepsilon) \log\left\{\frac{q^*(\mathcal{V}_\varepsilon)}{p_{\boldsymbol{\eta}}(\mathcal{V}_\varepsilon \mid \mathcal{D}_n)}\right\} + q^*(\mathcal{V}_\varepsilon^c) \log\left\{\frac{q^*(\mathcal{V}_\varepsilon^c)}{p_{\boldsymbol{\eta}}(\mathcal{V}_\varepsilon^c \mid \mathcal{D}_n)}\right\} \\
&\geq q^*(\mathcal{V}_\varepsilon) \log\{q^*(\mathcal{V}_\varepsilon)\} + q^*(\mathcal{V}_\varepsilon^c) \log\{q^*(\mathcal{V}_\varepsilon^c)\} - q^*(\mathcal{V}_\varepsilon^c) \log\{p_{\boldsymbol{\eta}}(\mathcal{V}_\varepsilon^c \mid \mathcal{D}_n)\} \\
&\geq -q^*(\mathcal{V}_\varepsilon^c) \log\{p_{\boldsymbol{\eta}}(\mathcal{V}_\varepsilon^c \mid \mathcal{D}_n)\} - \log 2 \\
&= -q^*(\mathcal{V}_\varepsilon^c) \log\left\{\int_{\mathcal{V}_\varepsilon^c} \frac{L(\mathcal{D}_n; f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})})}{L_0} \pi_0(\boldsymbol{z}) d\boldsymbol{z}\right\} \\
&+ q^*(\mathcal{V}_\varepsilon^c) \log\left\{\int \frac{L(\mathcal{D}_n; f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})})}{L_0} \pi_0(\boldsymbol{z}) d\boldsymbol{z}\right\} - \log 2,
\end{aligned}$$

where the third inequality holds by Jensen's inequality, the fourth step follows since $p_{\boldsymbol{\eta}}(\mathcal{V}_\varepsilon \mid \mathcal{D}_n) \leq 1$ and the fifth step follows since $x\log(x) + (1-x)\log(1-x) \geq -\log 2$. In the above proof we have assumed $q^*(\mathcal{V}_\varepsilon) > 0$, $q^*(\mathcal{V}_\varepsilon^c) > 0$. If $q^*(\mathcal{V}_\varepsilon^c) = 0$, there is nothing to prove. If $q^*(\mathcal{V}_\varepsilon) = 0$, then we will get $\varepsilon^2 = o_{P_0}(1)$ which is a contradiction.
By Assumption 2, the prior parameters satisfy

$$\|\boldsymbol{\mu}\|_2^2 = o(n), \quad \|\boldsymbol{\zeta}\|_\infty = O(n), \quad \|\boldsymbol{\zeta}^*\|_\infty = O(1), \quad \boldsymbol{\zeta}^* = 1/\boldsymbol{\zeta}.$$

Note $r_n \sim n^a$, $0 < a < 1$ and $D_n \sim d_n \sim n^u$, $0 < u < 1$ which implies $r_n \log(n) = o(n)$, $D_n \log(n) = o(n)$.
By part 1 of Proposition 6,

$$d_{\mathrm{KL}}(q^*, p_{\boldsymbol{\eta}}(\cdot \mid \mathcal{D}_n)) = o_{P_0}(n). \tag{24}$$

By step 1 (c) of the proof of Proposition 6,

$$B = o_{P_0}(n). \tag{25}$$

Since $r_n \sim n^a$, $r_n \log(n) = o(n^b)$, $a < b < 1$ and $D_n \sim n^u$, $D_n \log(n) = o(n^v)$, $u < v < 1$, using Proposition 5 with $\epsilon_n = 1$, we have

$$-q^*(\mathcal{V}_\varepsilon^c) A \geq n\varepsilon^2 q^*(\mathcal{V}_\varepsilon^c) - \log 2 + o_{P_0}(1) = n\varepsilon^2 q^*(\mathcal{V}_\varepsilon^c) + O_{P_0}(1). \tag{26}$$

Therefore, using (24), (25) and (26) in (21), we obtain

$$n\varepsilon^2 q^*(\mathcal{V}_\varepsilon^c) + O_{P_0}(1) \leq o_{P_0}(n) + o_{P_0}(n) \Rightarrow \pi^*(\mathcal{U}_\varepsilon^c) = q^*(\mathcal{V}_\varepsilon^c) = o_{P_0}(1).$$

E.2 Proof of Theorem 4.2

We assume the relation (21) holds with A and B same as in (22). We also turn to prove the posterior consistency of $q^*(\boldsymbol{z})$, as $\pi^*(\mathcal{U}^c_{\varepsilon\epsilon_n}) = q^*(\mathcal{V}^c_{\varepsilon\epsilon_n})$ where $\mathcal{V}_{\varepsilon\epsilon_n}$ is defined in (23). Note $r_n \sim n^a$, $0 < a < 1$, $D_n \sim n^u$, $0 < u < 1$ and $\epsilon_n^2 \sim n^{-\delta}$, $0 < \delta < 1 - u < 1 - a$. This implies $r_n \log(n) = o(n\epsilon_n^2)$, $D_n \log(n) = o(n\epsilon_n^2)$.
By Assumption 4, the prior parameters satisfy

$$\|\boldsymbol{\mu}\|_2^2 = o(n\epsilon_n^2), \quad \|\boldsymbol{\zeta}\|_\infty = O(n), \quad \|\boldsymbol{\zeta}^*\|_\infty = O(1), \quad \boldsymbol{\zeta}^* = 1/\boldsymbol{\zeta}.$$

In addition, by Assumption 5,

$$\|f_0 - f_{\boldsymbol{\Upsilon}}\|_\infty = o(\epsilon_n^2), \quad \|\boldsymbol{\Upsilon}\|_2^2 = o(n\epsilon_n^2), \quad \|\boldsymbol{s}\|_2^2 = o(n\epsilon_n^2).$$

By part 2 of Proposition 6,

$$d_{\mathrm{KL}}(q^*, p_{\boldsymbol{\eta}}(\cdot \mid \mathcal{D}_n)) = o_{P_0}(n\epsilon_n^2). \tag{27}$$

By step 2 (c) of the proof of Proposition 6,

$$B = o_{P_0}(n\epsilon_n^2). \tag{28}$$

Since $r_n \sim n^a$, $r_n \log(n) = o(n^b\epsilon_n^2)$, $a + \delta < b < 1$ and $D_n \sim n^u$, $D_n \log(n) = o(n^v\epsilon_n^2)$, $u + \delta < v < 1$, it follows, by using Proposition 5 that

$$-q^*(\mathcal{V}^c_{\varepsilon\epsilon_n})A \geq n\varepsilon^2\epsilon_n^2 q^*(\mathcal{V}^c_{\varepsilon\epsilon_n}) - \log 2 + o_{P_0}(1) = n\varepsilon^2\epsilon_n^2 q^*(\mathcal{V}^c_{\varepsilon\epsilon_n}) + O_{P_0}(1). \tag{29}$$

Thus, using (27), (28) and (29) in (21), we get

$$n\varepsilon^2\epsilon_n^2 q^*(\mathcal{V}^c_{\varepsilon\epsilon_n}) + O_{P_0}(1) \leq o_{P_0}(n\epsilon_n^2) + o_{P_0}(n\epsilon_n^2) \Rightarrow \pi^*(\mathcal{U}^c_{\varepsilon\epsilon_n}) = q^*(\mathcal{V}^c_{\varepsilon\epsilon_n}) = o_{P_0}(1).$$

E.3 Proof of Corollary 4.1

Note that

$$\begin{aligned}
R(\hat{C}) - R(C^{\mathsf{Bayes}}) &= E_{\boldsymbol{x}}\left[E_{y|\boldsymbol{x}}\left\{\mathbb{1}(\hat{C}(\boldsymbol{x}) \neq y) - \mathbb{1}(C^{\mathsf{Bayes}}(\boldsymbol{x}) \neq y)\right\}\right] \\
&= E_x\left(E_{y|\boldsymbol{x}}\left[\left\{\mathbb{1}(\hat{C}(\boldsymbol{x}) = 0) - \mathbb{1}(C^{\mathsf{Bayes}}(\boldsymbol{x}) = 0)\right\}\sigma(f_0(\boldsymbol{x}))\right]\right) \\
&+ E_{\boldsymbol{x}}\left(E_{y|\boldsymbol{x}}\left[\left\{\mathbb{1}(\hat{C}(\boldsymbol{x}) = 1) - \mathbb{1}(C^{\mathsf{Bayes}}(\boldsymbol{x}) = 1)\right\}\{1 - \sigma(f_0(\boldsymbol{x}))\}\right]\right) \\
&= 2E_{\boldsymbol{x}}\left[\mathbb{1}(\hat{C}(\boldsymbol{x}) \neq C^{\mathsf{Bayes}}(\boldsymbol{x}))\left|\sigma(f_0(\boldsymbol{x})) - \frac{1}{2}\right|\right] \\
&= 2E_{\boldsymbol{x}}\left[\mathbb{1}(\sigma(\hat{f}(\boldsymbol{x})) \geq \frac{1}{2}, \sigma(f_0(\boldsymbol{x})) < \frac{1}{2})\left|\sigma(f_0(\boldsymbol{x})) - \frac{1}{2}\right|\right] \\
&+ 2E_{\boldsymbol{x}}\left[\mathbb{1}(\sigma(\hat{f}(\boldsymbol{x})) < \frac{1}{2}, \sigma(f_0(\boldsymbol{x})) \geq \frac{1}{2})\left|\sigma(f_0(\boldsymbol{x})) - \frac{1}{2}\right|\right] \\
&\leq 2E_{\boldsymbol{x}}\left[\left|\sigma(f_0(\boldsymbol{x})) - \sigma(\hat{f}(\boldsymbol{x}))\right|\right].
\end{aligned} \tag{30}$$

Let

$$\hat{\ell}(y, \boldsymbol{x}) = \int \ell_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}(y, \boldsymbol{x}) q^*(\boldsymbol{z}) d\boldsymbol{z}. \tag{31}$$

Then

$$\begin{aligned}
d_{\mathrm{H}}(\hat{\ell}, \ell_0) &= d_{\mathrm{H}}\left(\int \ell_{G_{\boldsymbol{\eta}}(\boldsymbol{z})} q^*(\boldsymbol{z}) d\boldsymbol{z}, \ell_0\right) \\
&\le \int d_{\mathrm{H}}(\ell_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}, \ell_0) q^*(\boldsymbol{z}) d\boldsymbol{z} \quad \text{by Jensen's inequality} \\
&= \int_{\mathcal{V}_\varepsilon} d_{\mathrm{H}}(\ell_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}, \ell_0) q^*(\boldsymbol{z}) d\boldsymbol{z} + \int_{\mathcal{V}_\varepsilon^c} d_{\mathrm{H}}(\ell_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}, \ell_0) q^*(\boldsymbol{z}) d\boldsymbol{z} \\
&\le \varepsilon + o_{P_0}(1),
\end{aligned}$$

where the last inequality is a consequence of Theorem 4.1.
Taking $\varepsilon \to 0$, we get $d_{\mathrm{H}}(\hat{\ell}, \ell_0) = o_{P_0}(1)$.
Note that $\hat{f}(\boldsymbol{x}) = \sigma^{-1}(\hat{\ell}(y, \boldsymbol{x})) = \log\{\hat{\ell}(0, \boldsymbol{x})/\hat{\ell}(1, \boldsymbol{x})\}$, then

$$\begin{aligned}
2d_{\mathrm{H}}(\hat{\ell}, \ell_0) &= \int_{\boldsymbol{x}\in[0,1]^p} \sum_{y\in\{0,1\}} \left\{\sqrt{\hat{\ell}(y, \boldsymbol{x})} - \sqrt{\ell_0(y, \boldsymbol{x})}\right\}^2 d\boldsymbol{x} \\
&= 2 - 2\int_{\boldsymbol{x}\in[0,1]^p} \sum_{y\in\{0,1\}} \sqrt{\hat{\ell}(y, \boldsymbol{x})\ell_0(y, \boldsymbol{x})} d\boldsymbol{x} \\
&= 2 - 2\int_{\boldsymbol{x}\in[0,1]^p} \sum_{y\in\{0,1\}} \exp\left[\frac{1}{2}\left\{y\hat{f}(\boldsymbol{x}) - \log(1 + e^{\hat{f}(\boldsymbol{x})}) + yf_0(\boldsymbol{x}) - \log(1 + e^{f_0(\boldsymbol{x})})\right\}\right] d\boldsymbol{x} \\
&= 2 - 2\int_{\boldsymbol{x}\in[0,1]^p} \left[\sqrt{\sigma(\hat{f}(\boldsymbol{x}))\sigma(f_0(\boldsymbol{x}))} + \sqrt{\{1 - \sigma(\hat{f}(\boldsymbol{x}))\}\{1 - \sigma(f_0(\boldsymbol{x}))\}}\right] d\boldsymbol{x} \\
&\ge 2 - 2\int_{\boldsymbol{x}\in[0,1]^p} \sqrt{1 - \left\{\sqrt{\sigma(f_0(\boldsymbol{x}))} - \sqrt{\sigma(\hat{f}(\boldsymbol{x}))}\right\}^2} d\boldsymbol{x} \\
&\ge \int_{\boldsymbol{x}\in[0,1]^p} \left\{\sqrt{\sigma(f_0(\boldsymbol{x}))} - \sqrt{\sigma(\hat{f}(\boldsymbol{x}))}\right\}^2 d\boldsymbol{x} \\
&\ge \frac{1}{4}\int_{\boldsymbol{x}\in[0,1]^p} \{\sigma(f_0(\boldsymbol{x})) - \sigma(\hat{f}(\boldsymbol{x}))\}^2 d\boldsymbol{x},
\end{aligned} \tag{32}$$

where the fifth step holds because

$$\begin{aligned}
\left\{\sqrt{ab} + \sqrt{(1-a)(1-b)}\right\}^2 &= 2ab + 1 - a - b + 2\sqrt{ab(1-a)(1-b)} \\
&= 2\sqrt{ab}\{\sqrt{ab} + \sqrt{(1-a)(1-b)}\} + 1 - a - b \\
&\le 2\sqrt{ab}\{\frac{a+b}{2} + \frac{(1-a)+(1-b)}{2}\} + 1 - a - b \\
&= 2\sqrt{ab} + 1 - a - b \\
&= 1 - (\sqrt{a} - \sqrt{b})^2.
\end{aligned}$$

The sixth and seventh steps hold because $\sqrt{1-x} < 1 - x/2$ and $|a - b| \le |\sqrt{a} + \sqrt{b}||\sqrt{a} - \sqrt{b}| \le 2|\sqrt{a} - \sqrt{b}|$ respectively.
By the Cauchy–Schwartz inequality and (32),

$$\begin{aligned}
\int_{\boldsymbol{x}\in[0,1]^p} |\sigma(f_0(\boldsymbol{x})) - \sigma(\hat{f}(\boldsymbol{x}))| d\boldsymbol{x} &\le \left[\int_{\boldsymbol{x}\in[0,1]^p} \{\sigma(f_0(\boldsymbol{x})) - \sigma(\hat{f}(\boldsymbol{x}))\}^2 d\boldsymbol{x}\right]^{\frac{1}{2}} \\
&\le 2\sqrt{2} d_{\mathrm{H}}(\hat{\ell}, \ell_0) = o_{P_0}(1).
\end{aligned} \tag{33}$$

The proof of part 2 is completed by (30).

E.4 Proof of Corollary 4.2

Let $D_n \sim n^u$ and $\epsilon_n \sim n^{-\delta}$, $0 < \delta < 1-u$. This implies $D_n \log(n) = o(n\epsilon_n^2)$. Additionally, $D_n \log(n) = o(n^v \epsilon_n^2)$, $u + \delta < v < 1$. This implies $D_n \log(n) = o(n^v \epsilon_n^{2\kappa})$, $0 \leq \kappa \leq 1$. Thus, using Proposition 5 with $\epsilon_n = \epsilon_n^\kappa$, we have

$$-q^*(\mathcal{V}^c_{\varepsilon\epsilon_n^\kappa})A \geq n\varepsilon^2\epsilon_n^{2\kappa} q^*(\mathcal{V}^c_{\varepsilon\epsilon_n^\kappa}) - \log 2 + o_{P_0}(1) = n\varepsilon^2\epsilon_n^{2\kappa} q^*(\mathcal{V}^c_{\varepsilon\epsilon_n^\kappa}) + O_{P_0}(1).$$

This together with (27), (28) and (21) implies

$$q^*(\mathcal{V}^c_{\varepsilon\epsilon_n^\kappa}) = o_{P_0}(\epsilon_n^{2-2\kappa}).$$

By (31),

$$\begin{aligned}
d_{\mathrm{H}}(\hat{\ell}, \ell_0) &= d_{\mathrm{H}}\left(\int \ell_{G_{\boldsymbol{\eta}}(\boldsymbol{z})} q^*(\boldsymbol{z}) d\boldsymbol{z}, \ell_0\right) \\
&\leq \int d_{\mathrm{H}}(\ell_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}, \ell_0) q^*(\boldsymbol{z}) d\boldsymbol{z} \quad \text{by Jensen's inequality} \\
&= \int_{\mathcal{V}_{\varepsilon\epsilon_n^\kappa}} d_{\mathrm{H}}(\ell_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}, \ell_0) q^*(\boldsymbol{z}) d\boldsymbol{z} + \int_{\mathcal{V}^c_{\varepsilon\epsilon_n^\kappa}} d_{\mathrm{H}}(\ell_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}, \ell_0) q^*(\boldsymbol{z}) d\boldsymbol{z} \\
&\leq \varepsilon\epsilon_n^\kappa + o_{P_0}(\epsilon_n^{2-2\kappa}).
\end{aligned}$$

Dividing by ϵ_n^κ on both sides, we have

$$\frac{1}{\epsilon_n^\kappa} d_{\mathrm{H}}(\hat{\ell}, \ell_0) = o_{P_0}(1) + o_{P_0}(\epsilon_n^{2-3\kappa}) = o_{P_0}(1), \quad 0 \leq \kappa \leq 2/3$$

By (33), for every $0 \leq \kappa \leq 2/3$,

$$\frac{1}{\epsilon_n^\kappa} \int_{\boldsymbol{x} \in [0,1]^p} |\sigma(f_0(\boldsymbol{x})) - \sigma(\hat{f}(\boldsymbol{x}))| d\boldsymbol{x} \leq \frac{1}{\epsilon_n^\kappa} 2\sqrt{2} d_{\mathrm{H}}(\hat{\ell}, \ell_0) = o_{P_0}(1).$$

The proof completes following (30).

F Theoretical Properties of True Posterior

Theorem F.1. *Suppose $r_n \sim n^a$, $D_n \sim n^u$, $0 < a \leq u < 1$. Assume that the deterministic function $G_{\boldsymbol{\eta}}$ is differentiable at $\boldsymbol{z} \in \mathscr{Z}$ and the first-order derivative satisfies $\|\partial G_{\boldsymbol{\eta}}/\partial \boldsymbol{z}\|_F^2 = o(\varepsilon n^{1-u})$ for $\boldsymbol{z} \in \mathcal{P}_\varepsilon$ defined in* (19). *Besides, the prior parameters in* (2) *satisfy Assumption 2. Then, as R is defined in* (13),

1. $P_0\left(p(\mathcal{U}^c_\varepsilon \mid \mathcal{D}_n) \leq 2e^{-n\varepsilon^2/2}\right) \to 1, \quad n \to \infty.$

2. $P_0\left(\left|R(\hat{C}) - R(C^{\mathsf{Bayes}})\right| \leq 4\sqrt{2}\varepsilon\right) \to 1, \quad n \to \infty.$

Proof. The proof of this theorem is comprised of two parts.
Proof of part 1. In view of (11) and (23), we note that

$$p(\mathcal{U}_\varepsilon^c \mid \mathcal{D}_n) = p_{\boldsymbol{\eta}}(\mathcal{V}_\varepsilon^c \mid \mathcal{D}_n). \tag{34}$$

Also, note that,

$$\begin{aligned} p_{\boldsymbol{\eta}}(\mathcal{V}_\varepsilon^c \mid \mathcal{D}_n) &= \frac{\int_{\mathcal{V}_\varepsilon^c} L(\mathcal{D}_n; f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})})\pi_0(\boldsymbol{z})d\boldsymbol{z}}{\int L(\mathcal{D}_n; f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})})\pi_0(\boldsymbol{z})d\boldsymbol{z}} \\ &= \frac{\int_{\mathcal{V}_\varepsilon^c} \{L(\mathcal{D}_n; f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})})/L_0\}\pi_0(\boldsymbol{z})d\boldsymbol{z}}{\int \{L(\mathcal{D}_n; f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})})/L_0\}\pi_0(\boldsymbol{z})d\boldsymbol{z}}. \end{aligned} \tag{35}$$

By Assumption 2, the prior parameters satisfy

$$\|\boldsymbol{\mu}\|_2^2 = o(n), \quad \|\boldsymbol{\zeta}\|_\infty = O(n), \quad \|\boldsymbol{\zeta}^*\|_\infty = O(1), \quad \boldsymbol{\zeta}^* = 1/\boldsymbol{\zeta}.$$

Note $r_n \sim n^a$, $0 < a < 1$ which implies $r_n \log(n) = o(n)$. And $D_n \sim n^u$, $0 < u < 1$ which implies $D_n \log(n) = o(n)$. Thus, the conditions of Proposition 2 hold with $\epsilon_n = 1$.

$$\begin{aligned} &P_0\left(\int \frac{L(\mathcal{D}_n; f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})})}{L_0}\pi_0(\boldsymbol{z})d\boldsymbol{z} \le e^{-nv}\right) \\ &\le P_0\left(\left|\log\left\{\int \frac{L(\mathcal{D}_n; f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})})}{L_0}\pi_0(\boldsymbol{z})d\boldsymbol{z}\right\}\right| > nv\right) \to 0, \quad n \to \infty, \end{aligned} \tag{36}$$

where the above convergence follows from (67) in step 1 (c) of the proof of Proposition 6. Additionally, since $r_n \log(n) = o(n^b)$, $a < b < 1$, the conditions of Proposition 5 hold with $\epsilon_n = 1$.

$$P_0\left(\int_{\mathcal{V}_\varepsilon^c} \frac{L(\mathcal{D}_n; f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})})}{L_0}\pi_0(\boldsymbol{z})d\boldsymbol{z} \ge 2e^{-n\varepsilon^2}\right) \to 0, \quad n \to \infty, \tag{37}$$

where the last equality follows from (60) with $\epsilon_n = 1$ in the proof of Proposition 5. Using (36) and (37) with (34) and (35), we get

$$P_0\left(p(\mathcal{U}_\varepsilon^c \mid \mathcal{D}_n) \ge 2e^{-n(\varepsilon^2-\nu)/2}\right) = P_0\left(p_{\boldsymbol{\eta}}(\mathcal{V}_\varepsilon^c \mid \mathcal{D}_n) \ge 2e^{-n(\varepsilon^2-\nu)/2}\right) \to 0, \quad n \to \infty.$$

Taking $\nu = \varepsilon^2/2$, the proof is completed.
Proof of part 2. Let

$$\hat{\ell}(y, \boldsymbol{x}) = \int \ell_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}(y, \boldsymbol{x}) p_{\boldsymbol{\eta}}(\boldsymbol{z} \mid \mathcal{D}_n)d\boldsymbol{z}. \tag{38}$$

Then

$$\begin{aligned} d_{\mathrm{H}}(\hat{\ell}, \ell_0) &= d_{\mathrm{H}}\left(\int \ell_{G_{\boldsymbol{\eta}}(\boldsymbol{z})} p_{\boldsymbol{\eta}}(\boldsymbol{z} \mid \mathcal{D}_n)d\boldsymbol{z}, \ell_0\right) \\ &\le \int d_{\mathrm{H}}(\ell_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}, \ell_0) p_{\boldsymbol{\eta}}(\boldsymbol{z} \mid \mathcal{D}_n)d\boldsymbol{z} \quad \text{by Jensen's inequality} \\ &= \int_{\mathcal{V}_\varepsilon} d_{\mathrm{H}}(\ell_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}, \ell_0) p_{\boldsymbol{\eta}}(\boldsymbol{z} \mid \mathcal{D}_n)d\boldsymbol{z} + \int_{\mathcal{V}_\varepsilon^c} d_{\mathrm{H}}(\ell_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}, \ell_0) p_{\boldsymbol{\eta}}(\boldsymbol{z} \mid \mathcal{D}_n)d\boldsymbol{z} \\ &\le \varepsilon + 2e^{-n\varepsilon^2/2} \\ &\le 2\varepsilon \quad \text{with probability tending to 1 as } n \to \infty, \end{aligned}$$

where the last inequality is a consequence of part 1 of Theorem F.1.
The proof of part 2 is complete after (30) and (33). □

Theorem F.2. *Suppose $r_n \sim n^a$, $D_n \sim n^u$, $0 < a \leq u < 1$, $\epsilon_n^2 \sim n^{-\delta}$, $0 < \delta < 1-u \leq 1-a$. Assume that the deterministic function $G_{\boldsymbol{\eta}}$ is differentiable at $\boldsymbol{z} \in \mathscr{Z}$ and that the first-order derivative satisfies $\|\partial G_{\boldsymbol{\eta}}/\partial \boldsymbol{z}\|_F^2 = o(\varepsilon\epsilon_n^2 n^{1-u})$ for $\boldsymbol{z} \in \mathcal{P}_{\varepsilon\epsilon_n^2}$ defined in (19). In addition, the prior parameters in (2) satisfy Assumption 4. Then*

1. $P_0\left(p(\mathcal{U}_{\varepsilon\epsilon_n}^c \mid \mathcal{D}_n) \leq 2e^{-n\epsilon_n^2\varepsilon^2/2}\right) \to 1, \quad n \to \infty.$

2. $P_0\left(\left|R(\hat{C}) - R(C^{\mathsf{Bayes}})\right| \leq 4\sqrt{2}\varepsilon\epsilon_n\right) \to 1, \quad n \to \infty.$

Proof. The proof of this theorem consists of two parts.
Proof of part 1. In view of (11) and (23), we notice that

$$p(\mathcal{U}_{\varepsilon\epsilon_n}^c \mid \mathcal{D}_n) = p_{\boldsymbol{\eta}}(\mathcal{V}_{\varepsilon\epsilon_n}^c \mid \mathcal{D}_n). \tag{39}$$

By Assumption 4, the prior parameters satisfy

$$\|\boldsymbol{\mu}\|_2^2 = o(n\epsilon_n^2), \quad \|\boldsymbol{\zeta}\|_\infty = O(n), \quad \|\boldsymbol{\zeta}^*\|_\infty = O(1), \quad \boldsymbol{\zeta}^* = 1/\boldsymbol{\zeta}.$$

Also, by Assumption 5,

$$\|f_0 - f_{\boldsymbol{\Upsilon}}\|_\infty = o(\epsilon_n^2), \quad \|\boldsymbol{\Upsilon}\|_2^2 = o(n\epsilon_n^2), \quad \|\boldsymbol{z}\|_2^2 = o(n\epsilon_n^2).$$

Note $r_n \sim n^a$, $0 < a < 1$, $D_n \sim n^u$, $0 < u < 1$ and $\epsilon_n^2 \sim n^{-\delta}$, $0 < \delta < 1-u < 1-a$, therefore, $r_n \log(n) = o(n\epsilon_n^2)$, $D_n \log(n) = o(n\epsilon_n^2)$. Thus, the conditions of Proposition 2 hold.

$$\begin{aligned} &P_0\left(\int \frac{L(\mathcal{D}_n; f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})})}{L_0}\pi_0(\boldsymbol{z})d\boldsymbol{z} \leq e^{-n\epsilon_n^2 v}\right) \\ &\leq P_0\left(\left|\log\left\{\int \frac{L(\mathcal{D}_n; f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})})}{L_0}\pi_0(\boldsymbol{z})d\boldsymbol{z}\right\}\right| > n\epsilon_n^2 v\right) \to 0, \quad n \to \infty, \end{aligned} \tag{40}$$

where the above convergence follows from (70) in step 2 (c) of the proof of Proposition 6. Also, since $r_n \log(n) = o(n^b\epsilon_n^2)$, $a + \delta < b < 1$, $D_n \log(n) = o(n^v\epsilon_n^2)$, $u + \delta < v < 1$, the conditions of Proposition 5 are satisfied.

$$P_0\left(\int_{\mathcal{V}_\varepsilon^c} \frac{L(\mathcal{D}_n; f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})})}{L_0}\pi_0(\boldsymbol{z})d\boldsymbol{z} \geq 2e^{-n\epsilon_n^2\varepsilon^2}\right) \to 0, \quad n \to \infty, \tag{41}$$

where the last equality follows from (60) in the proof of Proposition 5.
Using (40) and (41) with (39) and (35), we get

$$P_0\left(p(\mathcal{U}_\varepsilon^c \mid \mathcal{D}_n) \geq 2e^{-n\epsilon_n^2(\varepsilon^2-\nu)/2}\right) = P_0\left(p_{\boldsymbol{\eta}}(\mathcal{V}_\varepsilon^c \mid \mathcal{D}_n) \geq 2e^{-n\epsilon_n^2(\varepsilon^2-\nu)/2}\right) \to 0, \quad n \to \infty.$$

Taking $\nu = \varepsilon^2/2$, we complete the proof.
Proof of part 2. By (38), we get

$$\begin{aligned} d_{\mathrm{H}}(\hat{\ell}, \ell_0) &= d_{\mathrm{H}}\left(\int \ell_{G_{\boldsymbol{\eta}}(\boldsymbol{z})} p_{\boldsymbol{\eta}}(\boldsymbol{z} \mid \mathcal{D}_n)d\boldsymbol{z}, \ell_0\right) \\ &\leq \int d_{\mathrm{H}}(\ell_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}, \ell_0)p_{\boldsymbol{\eta}}(\boldsymbol{z} \mid \mathcal{D}_n)d\boldsymbol{z} \quad \text{by Jensen's inequality} \\ &= \int_{\mathcal{V}_{\varepsilon\epsilon_n}} d_{\mathrm{H}}(\ell_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}, \ell_0)p_{\boldsymbol{\eta}}(\boldsymbol{z} \mid \mathcal{D}_n)d\boldsymbol{z} + \int_{\mathcal{V}_{\varepsilon\epsilon_n}^c} d_{\mathrm{H}}(\ell_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}, \ell_0)p_{\boldsymbol{\eta}}(\boldsymbol{z} \mid \mathcal{D}_n)d\boldsymbol{z} \\ &\leq \varepsilon\epsilon_n + 2e^{-2n\epsilon_n^2\varepsilon^2} \\ &\leq 2\varepsilon \quad \text{with probability tending to 1 as } n \to \infty, \end{aligned}$$

where the last inequality is a consequence of part 1 of Theorem F.2 and $\epsilon_n \sim n^{-\delta}$. Dividing by ϵ_n on both sides, we get

$$\epsilon_n^{-1} d_{\mathrm{H}}(\hat{\ell}, \ell_0) \leq 2\varepsilon \quad \text{with probability tending to 1 as } n \to \infty.$$

The remainder of the proof is followed by (30) and (33). □

G Lemmas

Lemma 1. *Assume that the deterministic function $G_{\boldsymbol{\eta}}$ is twice differentiable at $\boldsymbol{z} \in \mathscr{Z}$ and the second-order derivative satisfies*

$$\{\sum_{k=1}^{D_n} (\partial^2 G_{\boldsymbol{\eta} k}/\partial z_j^2)^2\}^{1/2} = o(\sqrt{D_n} \sum_{k=1}^{D_n} (\partial G_{\boldsymbol{\eta} k}/\partial z_j)^2),$$

for $j = 1, \ldots, r_n$. For

$h(\boldsymbol{z}) = \int_{\boldsymbol{x}\in[0,1]^p} [\sigma(f_0(\boldsymbol{x}))\{f_0(\boldsymbol{x}) - f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}(\boldsymbol{x})\} + \log\{1 - \sigma(f_0(\boldsymbol{x}))\} - \log\{1 - \sigma(f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}(\boldsymbol{x}))\}] d\boldsymbol{x}$,

we have

$$\sum_{j=1}^{r_n} |(\nabla^2 h(\boldsymbol{z}))_{jj}| \leq D_n(2c^2 + 1) \left|\left|\partial G_{\boldsymbol{\eta}}/\partial \boldsymbol{z}\right|\right|_F^2,$$

where $\mathbf{A}_{jj}$ denotes the jth diagonal entry of a matrix $\mathbf{A}$ and c is a constant.

Proof. Note that

$$\begin{aligned} \nabla^2 h(\boldsymbol{z}) = &-\int_{\boldsymbol{x}\in[0,1]^p} \underbrace{\{\sigma(f_0(\boldsymbol{x})) + \sigma(f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}(\boldsymbol{x}))\}}_{g_1(\boldsymbol{x})} \nabla^2 f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}(\boldsymbol{x}) d\boldsymbol{x} \\ &- \int_{\boldsymbol{x}\in[0,1]^p} \underbrace{\{\sigma(f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}(\boldsymbol{x}))\}\{1 - \sigma(f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}(\boldsymbol{x}))\}}_{g_2(\boldsymbol{x})} \nabla f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}(\boldsymbol{x}) \{\nabla f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}(\boldsymbol{x})\}^{\mathrm{T}} d\boldsymbol{x}. \end{aligned} \tag{42}$$

First, note that

$$(\nabla f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}(\boldsymbol{x}))_j = \frac{\partial f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}}{\partial z_j} = \sum_{k=1}^{D_n} \frac{\partial f_{\boldsymbol{w}}}{\partial w_k} \frac{\partial w_k}{\partial z_j},$$

where v_j denotes the jth element of a vector.

$$\begin{aligned} (\nabla f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}(\boldsymbol{x}) \{\nabla f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}(\boldsymbol{x})\}^{\mathrm{T}})_{jj} &= \left(\sum_{k=1}^{D_n} \frac{\partial f_{\boldsymbol{w}}}{\partial w_k} \frac{\partial w_k}{\partial z_j}\right) \left(\sum_{k=1}^{D_n} \frac{\partial f_{\boldsymbol{w}}}{\partial w_k} \frac{\partial w_k}{\partial z_j}\right) \\ &\leq \left\{\sum_{k=1}^{D_n} \left(\frac{\partial f_{\boldsymbol{w}}}{\partial w_k}\right)^2\right\} \left\{\sum_{k=1}^{D_n} \left(\frac{\partial G_{\boldsymbol{\eta} k}}{\partial z_j}\right)^2\right\} \\ &\leq D_n \max_t a_t^2 \sum_{k=1}^{D_n} \left(\frac{\partial G_{\boldsymbol{\eta} k}}{\partial z_j}\right)^2, \end{aligned} \tag{43}$$

where the second inequality holds by the Cauchy–Schwarz inequality. And the last inequality follows since $(\partial f_{\boldsymbol{w}}/\partial w_k)^2 \leq \max_t a_t^2$ by combining $0 < \sigma^{'}(\cdot) < 1$, $0 \leq x_{t^{'}} \leq 1$ and

$$\frac{\partial f_{\boldsymbol{w}}(\boldsymbol{x})}{\partial w_k} = \begin{cases} 1, & w_k = a_t \quad \text{for some } t = 1, \dots, d \\ a_t \sigma^{'}(\boldsymbol{\omega}_t^{\mathrm{T}} \boldsymbol{x} + b_t) x_{t^{'}}, & w_k = \omega_{tt^{'}} \quad \text{for some } t = 1, \dots, d, t^{'} = 1, \dots, p \\ a_t \sigma^{'}(\boldsymbol{\omega}_t^{\mathrm{T}} \boldsymbol{x} + b_t), & w_k = b_t \quad \text{for some } t = 1, \dots, d. \end{cases}$$

On the other hand,

$$(\nabla^2 f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}(\boldsymbol{x}))_{jj} = \underbrace{\left(\frac{\partial w_1}{\partial z_j}, \dots, \frac{\partial w_{D_n}}{\partial z_j}\right) \nabla_{\boldsymbol{w}}^2 f_{\boldsymbol{w}}(\boldsymbol{x}) \left(\frac{\partial w_1}{\partial z_j}, \dots, \frac{\partial w_{D_n}}{\partial z_j}\right)^{\mathrm{T}}}_{\text{(I)}} + \underbrace{\sum_{k=1}^{D_n} \frac{\partial f_{\boldsymbol{w}}(\boldsymbol{x})}{\partial w_k} \frac{\partial^2 w_k}{\partial z_j^2}}_{\text{(II)}}.$$

Note that

$$\text{(I)} \leq \lambda_{\mathtt{max}} \left\| \left(\frac{\partial w_1}{\partial z_j}, \dots, \frac{\partial w_{D_n}}{\partial z_j}\right)^{\mathrm{T}} \right\|_2^2 \leq D_n \max_t |a_t| \sum_{k=1}^{D_n} \left(\frac{\partial G_{\boldsymbol{\eta}k}}{\partial z_j}\right)^2,$$

where $\lambda_{\mathtt{max}}$ is the largest singular value of $\nabla_{\boldsymbol{w}}^2 f_{\boldsymbol{w}}(\boldsymbol{x})$ and the first inequality is based on the spectral norm of a matrix. The last inequality follows by the Gershgorin circle theorem where $\sum_{k_2=1}^{D_n} (\nabla_{\boldsymbol{w}}^2 f_{\boldsymbol{w}}(\boldsymbol{x}))_{k_1 k_2} \leq D_n \max_t |a_t|$ for any $k_1 = 1, \dots, D_n$, since

$$\begin{aligned} &(\nabla_{\boldsymbol{w}}^2 f_{\boldsymbol{w}}(\boldsymbol{x}))_{k_1 k_2} \\ &= \frac{\partial^2 f_{\boldsymbol{w}}(\boldsymbol{x})}{\partial w_{k_1} \partial w_{k_2}} \\ &= \begin{cases} \sigma^{'}(\boldsymbol{\omega}_t^{\mathrm{T}} \boldsymbol{x} + b_t) x_{t^{'}}, & w_{k_1} = \omega_{tt^{'}}, w_{k_2} = a_t \quad \text{for some } t = 1, \dots, d, t^{'} = 1, \dots, p \\ a_t \sigma^{''}(\boldsymbol{\omega}_t^{\mathrm{T}} \boldsymbol{x} + b_t) x_{t^{'}} x_{\tilde{t}}, & w_{k_1} = \omega_{tt^{'}}, w_{k_2} = \omega_{t\tilde{t}} \quad \text{for some } t = 1, \dots, d, t^{'}, \tilde{t} = 1, \dots, p \\ a_t \sigma^{''}(\boldsymbol{\omega}_t^{\mathrm{T}} \boldsymbol{x} + b_t) x_{t^{'}}, & w_{k_1} = \omega_{tt^{'}}, w_{k_2} = b_t \quad \text{for some } t = 1, \dots, d, t^{'} = 1, \dots, p \\ \sigma^{'}(\boldsymbol{\omega}_t^{\mathrm{T}} \boldsymbol{x} + b_t), & w_{k_1} = b_t, w_{k_2} = a_t \quad \text{for some } t = 1, \dots, d \\ a_t \sigma^{''}(\boldsymbol{\omega}_t^{\mathrm{T}} \boldsymbol{x} + b_t) x_{t^{'}}, & w_{k_1} = b_t, w_{k_2} = \omega_{tt^{'}} \quad \text{for some } t = 1, \dots, d, t^{'} = 1, \dots, p \\ a_t \sigma^{''}(\boldsymbol{\omega}_t^{\mathrm{T}} \boldsymbol{x} + b_t), & w_{k_1} = b_{t_1}, w_{k_2} = b_t \quad \text{for some } t = 1, \dots, d \\ 0, & \text{otherwise} \end{cases} \\ &\leq \max_t |a_t|, \end{aligned}$$

where the last inequality follows using $0 < \sigma^{'}(\cdot) < 1$, $0 < \sigma^{''}(\cdot) < 1$, $0 \leq x_{t^{'}} \leq 1$.
Also, note that

$$\text{(II)} \leq \sqrt{\left\{\sum_{k=1}^{D_n} \left(\frac{\partial f_{\boldsymbol{w}}}{\partial w_k}\right)^2\right\} \left\{\sum_{k=1}^{D_n} \left(\frac{\partial^2 G_{\boldsymbol{\eta}k}}{\partial z_j^2}\right)^2\right\}} \leq \sqrt{D_n \max_t a_t^2 \sum_{k=1}^{D_n} \left(\frac{\partial^2 G_{\boldsymbol{\eta}k}}{\partial z_j^2}\right)^2},$$

where the first inequality follows by the Cauchy–Schwarz inequality.
Therefore, we get

$$(\nabla^2 f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}(\boldsymbol{x}))_{jj} \leq D_n \max_t |a_t| \sum_{k=1}^{D_n} \left(\frac{\partial G_{\boldsymbol{\eta}k}}{\partial z_j}\right)^2 + \sqrt{D_n \max_t a_t^2 \sum_{k=1}^{D_n} \left(\frac{\partial^2 G_{\boldsymbol{\eta}k}}{\partial z_j^2}\right)^2}. \tag{44}$$

Note $|g_1(\boldsymbol{x})| \leq 2$, $|g_2(\boldsymbol{x})| \leq 1$ and combining (43) and (44) and replacing (42), we get

$$\begin{aligned}
\sum_{j=1}^{r_n} |(\nabla^2 h(\boldsymbol{z}))_{jj}| &\leq \sum_{j=1}^{r_n} \left\{ D_n(\max_t a_t^2 + \max_t |a_t|) \sum_{k=1}^{D_n} \left(\frac{\partial G_{\boldsymbol{\eta}k}}{\partial z_j} \right)^2 \right\} \\
&\quad + \sum_{j=1}^{r_n} \left\{ \max_t |a_t| \sqrt{D_n \sum_{k=1}^{D_n} \left(\frac{\partial^2 G_{\boldsymbol{\eta}k}}{\partial z_j^2} \right)^2 } \right\} \\
&\lesssim \sum_{j=1}^{r_n} \left\{ D_n(\max_t a_t^2 + \max_t |a_t|) \sum_{k=1}^{D_n} \left(\frac{\partial G_{\boldsymbol{\eta}k}}{\partial z_j} \right)^2 \right\} \\
&\leq D_n(2 \max_t a_t^2 + 1) \sum_{j=1}^{r_n} \sum_{k=1}^{D_n} \left(\frac{\partial G_{\boldsymbol{\eta}k}}{\partial z_j} \right)^2 \\
&\leq D_n(2c^2 + 1) \left|\left| \frac{\partial G_{\boldsymbol{\eta}}}{\partial \boldsymbol{z}} \right|\right|_F^2,
\end{aligned}$$

where the second inequality follows by

$$\{\sum_{k=1}^{D_n} (\partial^2 G_{\boldsymbol{\eta}k} / \partial z_j^2)^2\}^{1/2} = o(\sqrt{D_n} \sum_{k=1}^{D_n} (\partial G_{\boldsymbol{\eta}k} / \partial z_j)^2)$$

and the third inequality in the above step uses $|x| < x^2 + 1$. The last inequality follows by letting $\max_t |a_t| < c$. □

Lemma 2. *Let $\tilde{\mathcal{F}}_n = \{\sqrt{\ell} : \ell_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}(y, \boldsymbol{x}), \boldsymbol{z} \in \mathcal{F}_n\}$ where $\ell_{G_{\boldsymbol{\eta}}}(y, \boldsymbol{x}) = \ell_{\boldsymbol{w}}(y, \boldsymbol{x})$ is the same as in (10) and $\mathcal{F}_n = \{\boldsymbol{z} : |z_j| \leq \tilde{C}_n, j = 1, \ldots, r_n\}$ is defined in (20). Assume that $w_k = G_{\boldsymbol{\eta}k}(\boldsymbol{z})$ satisfies $|w_k| \leq C_n$, for $k = 1, \ldots, D_n$. Furthermore, assume that the deterministic function $G_{\boldsymbol{\eta}}$ is differentiable at $\boldsymbol{z} \in \mathscr{X}$ and the first-order derivative satisfies $\|\partial G_{\boldsymbol{\eta}} / \partial \boldsymbol{z}\|_F^2 = o(\varepsilon \epsilon_n^2 n^{1-u})$ for $\boldsymbol{z} \in \mathcal{F}_n$. Then*

$$\int_{\varepsilon^2/8}^{\sqrt{2}\varepsilon} \sqrt{H(u, \tilde{\mathcal{F}}_n, \|\cdot\|_2)} du \lesssim \varepsilon \sqrt{2 r_n \{\log(r_n) + \log(D_n) + \log(C_n) + \log(\tilde{C}_n) - \log(\varepsilon)\}},$$

where $H(u, \tilde{\mathcal{F}}_n, \|\cdot\|_2)$ is the natural logarithm of the bracketing number defined in Pollard (1990) and explained in detail in the proof below.

Proof. As stated in Pollard (1990), for any two functions l and u, we define the bracket $[l, u]$ as the set of all functions f such that $l \leq f \leq u$. Then an ε-bracket is defined as a bracket with $\|u - l\| \leq \varepsilon$ where $\|\cdot\|$ is a metric. Define the bracketing number of a set of functions $\mathcal{F}^*$ as the minimum number of ε-brackets needed to cover $\mathcal{F}^*$, and denote it by $N(\varepsilon, \mathcal{F}^*, \|\cdot\|)$. Finally, the bracketing entropy, denoted by $H(\varepsilon, \mathcal{F}^*, \|\cdot\|)$, is the natural logarithm of the bracketing number.

Note that by Lemma 4.1 in Pollard (1990),

$$N(\varepsilon, \mathcal{F}_n, \|\cdot\|_\infty) \leq (\frac{3\tilde{C}_n}{\varepsilon})^{r_n}.$$

For $\boldsymbol{z}_1, \boldsymbol{z}_2 \in \mathcal{F}_n$, let $\tilde{\ell}(u) = \sqrt{\ell_{u\boldsymbol{z}_1 + (1-u)\boldsymbol{z}_2}(y, \boldsymbol{x})}$.

Following equation (52) in Bhattacharya and Maiti (2021), we get

$$\sqrt{\ell_{\boldsymbol{z}_1}(y, \boldsymbol{x})} - \sqrt{\ell_{\boldsymbol{z}_1}(y, \boldsymbol{x})} \leq r_n \sup_j \left| \frac{\partial \tilde{\ell}}{\partial z_j} \right| \|\boldsymbol{z}_1 - \boldsymbol{z}_2\|_\infty \leq F(\boldsymbol{x}, y) \|\boldsymbol{z}_1 - \boldsymbol{z}_2\|_\infty \tag{45}$$

where the upper bound $F(\boldsymbol{x}, y) = r_n D_n C_n/2$. This is because $|\partial\tilde{\ell}/\partial z_j|$ is bounded as shown below:

$$\begin{aligned}
\left|\frac{\partial\tilde{\ell}}{\partial z_j}\right| &= \left|\frac{1}{2}\frac{\partial f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}(\boldsymbol{x})}{\partial z_j}\sqrt{\left\{y - \frac{e^{f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}(\boldsymbol{x})}}{1+e^{f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}(\boldsymbol{x})}}\right\}\exp\left\{yf_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}(\boldsymbol{x}) - \log(1+e^{f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}(\boldsymbol{x})})\right\}}\right| \\
&= \frac{1}{2}\sqrt{\left\{y - \frac{e^{f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}(\boldsymbol{x})}}{1+e^{f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}(\boldsymbol{x})}}\right\}\exp\left\{yf_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}(\boldsymbol{x}) - \log(1+e^{f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}(\boldsymbol{x})})\right\}}\left|\sum_{k=1}^{D_n}\frac{\partial f_{\boldsymbol{w}}(\boldsymbol{x})}{\partial w_k}\frac{\partial G_{\boldsymbol{\eta}k}}{\partial z_j}\right| \\
&\leq \frac{1}{2}\sqrt{\frac{e^{f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}(\boldsymbol{x})}}{1+e^{f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}(\boldsymbol{x})}}}\sqrt{\frac{1}{1+e^{f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}(\boldsymbol{x})}}}\left|\sum_{k=1}^{D_n}\frac{\partial f_{\boldsymbol{w}}(\boldsymbol{x})}{\partial w_k}\frac{\partial G_{\boldsymbol{\eta}k}}{\partial z_j}\right| \\
&\leq \frac{1}{2}\sqrt{\frac{e^{f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}(\boldsymbol{x})}}{1+e^{f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}(\boldsymbol{x})}}}\sqrt{\frac{1}{1+e^{f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}(\boldsymbol{x})}}}\sqrt{\left\{\sum_{k=1}^{D_n}\left(\frac{\partial f_{\boldsymbol{w}}(\boldsymbol{x})}{\partial w_k}\right)^2\right\}\left\{\sum_{k=1}^{D_n}\left(\frac{\partial G_{\boldsymbol{\eta}k}}{\partial z_j}\right)^2\right\}}.
\end{aligned}$$

Also, note that

$$\left|\frac{\partial f_{\boldsymbol{w}}(\boldsymbol{x})}{\partial w_k}\right| \leq \begin{cases} 1, & w_k = a_t \quad \text{for some } t = 1,\dots,d \\ |a_t\sigma'(\boldsymbol{\omega}_t^{\mathrm{T}}\boldsymbol{x} + b_t)x_{t'}|, & w_k = \omega_{tt'} \quad \text{for some } t = 1,\dots,d, t' = 1,\dots,p \\ |a_t\sigma'(\boldsymbol{\omega}_t^{\mathrm{T}}\boldsymbol{x} + b_t)|, & w_k = b_t \quad \text{for some } t = 1,\dots,d. \end{cases}$$

Using $|\sigma'(\cdot)| \leq 1$, $|x_{t'}| \leq 1$, $|a_t| \leq C_n$,

$$\left|\frac{\partial f_{\boldsymbol{w}}(\boldsymbol{x})}{\partial w_k}\right| \leq C_n.$$

Thus, using $e^{f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}(\boldsymbol{x})}/(1+e^{f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}(\boldsymbol{x})}) \leq 1$, $1/(1+e^{f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}(\boldsymbol{x})}) \leq 1$, $\|\partial G_{\boldsymbol{\eta}}/\partial\boldsymbol{z}\|_F^2 = o(\varepsilon\epsilon_n^2 n^{1-u})$, we get

$$\left|\frac{\partial\tilde{\ell}}{\partial z_j}\right| \leq \frac{1}{2}\sqrt{D_n C_n^2 \varepsilon\epsilon_n^2 n^{1-u}}.$$

By using $D_n \sim n^u$, $\epsilon_n^2 \sim n^\delta$, $0 < \delta < 1-u$ and letting $\varepsilon = 1$, the bound $F(\boldsymbol{x}, y)$ follows. In view of (45) and Theorem 2.7.11 in Vaart and Wellner (1996), we have

$$N(\varepsilon, \tilde{\mathcal{F}}_n, \||\cdot\|_2) \leq (\frac{3r_n D_n C_n\tilde{C}_n}{2\varepsilon})^{r_n} \Rightarrow H(\varepsilon, \tilde{\mathcal{F}}_n, \|\cdot\|_2) \leq r_n \log\left(\frac{r_n D_n C_n \tilde{C}_n}{\varepsilon}\right).$$

Using Lemma 7.3 in Bhattacharya et al. (2020) with $M_n = r_n D_n C_n \tilde{C}_n$, we get

$$\int_0^\varepsilon \sqrt{H(u, \tilde{\mathcal{F}}_n, \|\cdot\|_2)}du \lesssim \varepsilon\sqrt{r_n[\log(r_n D_n C_n \tilde{C}_n) - \log(\varepsilon)]}.$$

Therefore,

$$\begin{aligned}
\int_{\varepsilon^2/8}^{\sqrt{2}\varepsilon} \sqrt{H(u, \tilde{\mathcal{F}}_n, \|\cdot\|_2)}du &\leq \int_0^{\sqrt{2}\varepsilon} \sqrt{H(u, \tilde{\mathcal{F}}_n, \|\cdot\|_2)}du \\
&\lesssim \sqrt{2}\varepsilon\sqrt{r_n\{\log(r_n D_n C_n \tilde{C}_n) - \log(\sqrt{2}\varepsilon)\}}.
\end{aligned}$$

The proof follows by noting that $\log(\sqrt{2}\varepsilon) \geq \log(\varepsilon)$. □

H Propositions

Proposition 1. *Let $q(\boldsymbol{z}) = N(\boldsymbol{s}, \mathbf{I}_{r_n}/\sqrt{n})$ and $\pi_0(\boldsymbol{z}) = N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma} = \mathrm{diag}(\boldsymbol{\zeta})$ and $\boldsymbol{\zeta}^* = 1/\boldsymbol{\zeta}$. Let $n\epsilon_n^2 \to \infty$, $r_n \log(n) = o(n\epsilon_n^2)$, $\|\boldsymbol{s}\|_2^2 = o(n\epsilon_n^2)$ and $\|\boldsymbol{\mu}\|_2^2 = o(n\epsilon_n^2)$, then for any $\nu > 0$,*

$$d_{\mathrm{KL}}(q, \pi_0) \le n\epsilon_n^2 \nu,$$

provided $\|\boldsymbol{\zeta}\|_\infty = O(n)$, $\|\boldsymbol{\zeta}^*\|_\infty = O(1)$.

Proof.

$$\begin{aligned}
d_{\mathrm{KL}}(q, \pi_0) &= \sum_{j=1}^{r_n} \left\{ \log(n\zeta_j) + \frac{1}{n\zeta_j^2} + \frac{(s_j - \mu_j)^2}{\zeta_j^2} - \frac{1}{2} \right\} \\
&\le \frac{r_n}{2}\{\log(n) - 1\} + \sum_{j=1}^{r_n} \log(\zeta_j) + \frac{1}{n}\sum_{j=1}^{r_n} \frac{1}{\zeta_j^2} + 2\sum_{j=1}^{r_n} \frac{s_j^2}{\zeta_j^2} + 2\sum_{j=1}^{r_n} \frac{\mu_j^2}{\zeta_j^2} - \frac{r_n}{2} \\
&\le \frac{r_n}{2}\{\log(n) - 1\} + r_n \log(\|\boldsymbol{\zeta}\|_\infty) + \frac{r_n}{n}\|\boldsymbol{\zeta}^*\|_\infty + 2\|\boldsymbol{s}\|_2^2 \|\boldsymbol{\zeta}^*\|_\infty + 2\|\boldsymbol{\mu}\|_2^2 \|\boldsymbol{\zeta}^*\|_\infty \\
&= o(n\epsilon_n^2),
\end{aligned}$$

where the second last inequality uses $\boldsymbol{\zeta}^* = 1/\boldsymbol{\zeta}$. The last equality follows since $\|\boldsymbol{\zeta}\|_\infty = O(n)$, $\|\boldsymbol{\zeta}^*\|_\infty = O(1)$, $r_n \log(n) = o(n\epsilon_n^2)$, $\|\boldsymbol{s}\|_2^2 = o(n\epsilon_n^2)$ and $\|\boldsymbol{\mu}\|_2^2 = o(n\epsilon_n^2)$. □

Proposition 2. *Let $\pi_0(\boldsymbol{z})$ be as in (2). Define*

$$\begin{aligned}
\mathcal{N}_\varepsilon &= \{\boldsymbol{w} : d_{\mathrm{KL}}(\ell_0, \ell_{\boldsymbol{w}}) < \varepsilon\}, \\
\mathcal{L}_\varepsilon &= \{\boldsymbol{z} : G_{\boldsymbol{\eta}}(\boldsymbol{z}) \in \mathcal{N}_\varepsilon\},
\end{aligned} \tag{46}$$

where

$$d_{\mathrm{KL}}(\ell_0, \ell_{\boldsymbol{w}}) = \int_{\boldsymbol{x} \in [0,1]^p} \left[\sigma(f_0(\boldsymbol{x}))\{f_0(\boldsymbol{x}) - f_{\boldsymbol{w}}(\boldsymbol{x})\} + \log\left\{ \frac{1 - \sigma(f_0(\boldsymbol{x}))}{1 - \sigma(f_{\boldsymbol{w}}(\boldsymbol{x}))} \right\} \right] d\boldsymbol{x}.$$

Let $\|f_0 - f_{\boldsymbol{\Upsilon}}\|_\infty \le \varepsilon\epsilon_n^2/4$, $n\epsilon_n^2 \to \infty$. Assume that the deterministic function $G_{\boldsymbol{\eta}}$ is differentiable at $\boldsymbol{z} \in \mathscr{Z}$ and the first-order derivative satisfies $\|\partial G_{\boldsymbol{\eta}}/\partial \boldsymbol{z}\|_F^2 = o(\varepsilon\epsilon_n^2 n^{1-u})$ for $\boldsymbol{z} \in \mathcal{P}_{\varepsilon\epsilon_n^2}$ defined in (19). If $r_n \log(n) = o(n\epsilon_n^2)$, $D_n \log(n) = o(n\epsilon_n^2)$, $\|\boldsymbol{s}\|_2^2 = o(n\epsilon_n^2)$, $\|\boldsymbol{\Upsilon}\|_2^2 = o(n\epsilon_n^2)$, $\|\boldsymbol{\mu}\|_2^2 = o(n\epsilon_n^2)$, then for any $\nu > 0$,

$$\int_{\boldsymbol{z} \in \mathcal{L}_{\varepsilon\epsilon_n^2}} \pi_0(\boldsymbol{z}) d\boldsymbol{z} \ge e^{-n\epsilon_n^2 \nu},$$

provided $\|\boldsymbol{\zeta}\|_\infty = O(n)$, $\|\boldsymbol{\zeta}^*\|_\infty = O(1)$ where $\boldsymbol{\zeta}^* = 1/\boldsymbol{\zeta}$.

Proof. Let $f_{\boldsymbol{\Upsilon}}(\boldsymbol{x}) = \sum_{j=1}^d a_j^{\boldsymbol{\Upsilon}} \sigma((\boldsymbol{\omega}_j^{\boldsymbol{\Upsilon}})^{\mathrm{T}} \boldsymbol{x} + b_j^{\boldsymbol{\Upsilon}})$ be the neural network such that

$$\|f_{\boldsymbol{\Upsilon}} - f_0\|_1 \le \frac{\varepsilon\epsilon_n^2}{4}, \tag{47}$$

and let $\boldsymbol{\Upsilon} = G_{\boldsymbol{\eta}}(\boldsymbol{s})$. Such a neural network exists since $\|f_{\boldsymbol{\Upsilon}} - f_0\|_1 \le \|f_{\boldsymbol{\Upsilon}} - f_0\|_\infty \le \varepsilon\epsilon_n^2/4$. Next define a neighborhood $\mathcal{M}_{\varepsilon\epsilon_n^2}$ as follows:

$$\mathcal{M}_{\varepsilon\epsilon_n^2} = \left\{ \boldsymbol{w} : |w_k - \Upsilon_k| < \frac{\varepsilon\epsilon_n^2}{8\{D_n + (p+1)\|\boldsymbol{\Upsilon}\|_1\}},\ k = 1, \ldots, D_n \right\}.$$

For every $\boldsymbol{w} \in \mathcal{M}_{\varepsilon\epsilon_n^2}$, by Lemma 7.7 in Bhattacharya et al. (2020), we have

$$\|f_{\boldsymbol{w}} - f_{\boldsymbol{\Upsilon}}\|_1 \leq \frac{\varepsilon\epsilon_n^2}{2}. \tag{48}$$

Combining (47) and (48), we get for $\boldsymbol{w} \in \mathcal{M}_{\varepsilon\epsilon_n^2}$, $\|f_{\boldsymbol{w}} - f_0\|_1 \leq \varepsilon\epsilon_n^2/2$.
Thus, in view of Lemma 7.8 in Bhattacharya et al. (2020), $d_{\mathrm{KL}}(\ell_0, \ell_{\boldsymbol{w}}) < \varepsilon\epsilon_n^2$.
Thus, for every $\boldsymbol{w} \in \mathcal{M}_{\varepsilon\epsilon_n^2}$, we have $\boldsymbol{w} \in \mathcal{N}_{\varepsilon\epsilon_n^2}$.
Now define a neighborhood $\mathcal{P}_{\varepsilon\epsilon_n^2}$ as follows, which is the same as (19):

$$\mathcal{P}_{\varepsilon\epsilon_n^2} = \left\{ \boldsymbol{z} : |z_j - s_j| < \frac{\sqrt{\varepsilon\epsilon_n^2}}{8\sqrt{r_n n^{1-u}}\{D_n + (p+1)\|\boldsymbol{\Upsilon}\|_1\}}, j = 1, \ldots, r_n \right\}.$$

Then, for $k = 1, \ldots, D_n$,

$$\begin{aligned}
|w_k - \Upsilon_k| &= |G_{\boldsymbol{\eta}k}(\boldsymbol{z}) - G_{\boldsymbol{\eta}k}(\boldsymbol{s})| \\
&= |\left(\nabla G_{\boldsymbol{\eta}k}(\boldsymbol{\xi})\right)^{\mathrm{T}}(\boldsymbol{z} - \boldsymbol{s})| \\
&= \left| \sum_{j=1}^{r_n} \frac{\partial G_{\boldsymbol{\eta}k}}{\partial \boldsymbol{\xi}_j}(z_j - s_j) \right| \\
&\leq \sqrt{\left\{ \sum_{j=1}^{r_n} \left(\frac{\partial G_{\boldsymbol{\eta}k}}{\partial \boldsymbol{\xi}_j} \right)^2 \right\} \left\{ \sum_{j=1}^{r_n} (z_j - s_j)^2 \right\}} \\
&< \sqrt{\left\| \frac{\partial G_{\boldsymbol{\eta}}}{\partial \boldsymbol{z}} \right\|_F^2 \left[\frac{\sqrt{\varepsilon\epsilon_n^2}}{8\sqrt{n^{1-u}}\{D_n + (p+1)\|\boldsymbol{\Upsilon}\|_1\}} \right]^2} \\
&= \frac{\varepsilon\epsilon_n^2}{8\{D_n + (p+1)\|\boldsymbol{\Upsilon}\|_1\}},
\end{aligned}$$

where $\boldsymbol{\xi} \in \mathscr{Z}$ and the last inequality follows by the Cauchy–Schwarz inequality.
Thus, for every $\boldsymbol{z} \in \mathcal{P}_{\varepsilon\epsilon_n^2}$, $\boldsymbol{w} = G_{\boldsymbol{\eta}}(\boldsymbol{z}) \in \mathcal{M}_{\varepsilon\epsilon_n^2} \subseteq \mathcal{N}_{\varepsilon\epsilon_n^2}$. Additionally, by (46), for every $\boldsymbol{z} \in \mathcal{P}_{\varepsilon\epsilon_n^2}$, $\boldsymbol{z} \in \mathcal{L}_{\varepsilon\epsilon_n^2}$. Therefore,

$$\int_{\boldsymbol{z} \in \mathcal{L}_{\varepsilon\epsilon_n^2}} \pi_0(\boldsymbol{z}) d\boldsymbol{z} \geq \int_{\boldsymbol{z} \in \mathcal{P}_{\varepsilon\epsilon_n^2}} \pi_0(\boldsymbol{z}) d\boldsymbol{z}.$$

Let $\delta = \sqrt{\varepsilon\epsilon_n^2}/[8\sqrt{r_n n^{1-u}}\{D_n + (p+1)\|\boldsymbol{\Upsilon}\|_1\}]$, then

$$\begin{aligned}
\int_{\boldsymbol{z} \in \mathcal{P}_{\varepsilon\epsilon_n^2}} \pi_0(\boldsymbol{z}) d\boldsymbol{z} &= \prod_{j=1}^{r_n} \int_{s_j - \delta}^{s_j + \delta} \frac{1}{\sqrt{2\pi\zeta_j^2}} e^{-\frac{(z_j - \mu_j)^2}{2\zeta_j^2}} dz_j \\
&= \prod_{j=1}^{r_n} \frac{2\delta}{\sqrt{2\pi\zeta_j^2}} e^{-\frac{(\tilde{s}_j - \mu_j)^2}{2\zeta_j^2}}, \quad \tilde{s}_j \in [s_j - \delta, s_j + \delta] \qquad (49) \\
&= \prod_{j=1}^{r_n} e^{-\left\{ -\frac{1}{2}\log\left(\frac{2}{\pi}\right) - \log(\delta) + \log(\zeta_j) + \frac{(\tilde{s}_j - \mu_j)^2}{2\zeta_j^2} \right\}},
\end{aligned}$$

where the second last equality holds by the mean value theorem.
Note that $\tilde{s}_j \in [s_j - \delta, s_j + \delta]$, since $\delta \to 0$, therefore,

$$\frac{(\tilde{s}_j - \mu_j)^2}{2\zeta_j^2} \leq \frac{\max\{(s_j - \mu_j - 1)^2, (s_j - \mu_j + 1)^2\}}{2\zeta_j^2} \leq \frac{(s_j - \mu_j)^2}{\zeta_j^2} + \frac{1}{\zeta_j^2},$$

where the last inequality follows since $(a+b)^2 \leq 2(a^2 + b^2)$. Therefore,

$$\begin{aligned}\sum_{j=1}^{r_n} \frac{(\tilde{s}_j - \mu_j)^2}{2\zeta_j^2} &\leq 2\sum_{j=1}^{r_n} \frac{s_j^2}{\zeta_j^2} + 2\sum_{j=1}^{r_n} \frac{\mu_j^2}{\zeta_j^2} + \sum_{j=1}^{r_n} \frac{1}{\zeta_j^2} \\ &\leq 2(\|\boldsymbol{s}\|_2^2 + \|\boldsymbol{\mu}\|_2^2 + 1)\|\boldsymbol{\zeta}^*\|_\infty \\ &\leq n\nu\epsilon_n^2,\end{aligned} \tag{50}$$

since $\|\boldsymbol{s}\|_2^2 = o(n\epsilon_n^2)$, $\|\boldsymbol{\mu}\|_2^2 = o(n\epsilon_n^2)$ and $\|\boldsymbol{\zeta}^*\|_\infty = O(1)$ and $n\epsilon_n^2 \to \infty$. Furthermore,

$$\begin{aligned}-\log(\delta) + \log(\zeta_j) &= \log 8 + \log\{D_n + (p+1)\|\boldsymbol{\Upsilon}\|_1\} + \log(r_n n^{1-u}) + \log(\zeta_j) - \frac{1}{2}\log(\varepsilon\epsilon_n^2) \\ &\leq \log 8 + \log\{D_n + (p+1)\sqrt{D_n}\|\boldsymbol{\Upsilon}\|_2\} + \log(r_n n^{1-u}) + \log(\zeta_j) \\ &\quad - \frac{1}{2}\log(\varepsilon) - \log(\epsilon_n) \\ &\leq \log 8 + \log(D_n) + \log(1 + \|\boldsymbol{\Upsilon}\|_2) + \log(r_n n^{1-u}) + \log(\zeta_j) \\ &\quad - \frac{1}{2}\log(\varepsilon) - \log(\epsilon_n),\end{aligned}$$

where the second inequality is an outcome of the Cauchy–Schwarz inequality and the third inequality follows since $p + 1 \leq \sqrt{D_n}$, $n \to \infty$. Therefore,

$$\begin{aligned}&\sum_{j=1}^{r_n} -\frac{1}{2}\log\left(\frac{2}{\pi}\right) - \log(\delta) + \log(\zeta_j) \\ &\leq r_n \log 8 + r_n \log(D_n) + r_n \log(1 + \|\boldsymbol{\Upsilon}\|_2) + r_n \log(r_n n^{1-u}) + r_n \log(\|\boldsymbol{\zeta}\|_\infty) \\ &- \frac{1}{2} r_n \log(\varepsilon) - r_n \log(\epsilon_n) \\ &\leq n\nu\epsilon_n^2,\end{aligned} \tag{51}$$

where the last inequality follows since $D_n \log(n) = o(n\epsilon_n^2)$, $r_n \log(n) = o(n\epsilon_n^2)$, $\|\boldsymbol{\zeta}\|_\infty = O(n)$, $\|\boldsymbol{\Upsilon}\|_2 = o(\sqrt{n}\epsilon_n) = o(n)$ and $1/n\epsilon_n^2 = o(1)$ which implies $-2\log(\epsilon_n) = o(\log(n))$. Combining (50) and (51) and replacing (49), the proof follows. □

Proposition 3. *Let $q(\boldsymbol{z}) \sim N(\boldsymbol{s}, \mathbf{I}_{r_n}/\sqrt{n})$. Define*

$$h(\boldsymbol{z}) = \int_{\boldsymbol{x} \in [0,1]^p} \left[\sigma(f_0(\boldsymbol{x}))\{f_0(\boldsymbol{x}) - f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}(\boldsymbol{x})\} + \log\left\{\frac{1 - \sigma(f_0(\boldsymbol{x}))}{1 - \sigma(f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}(\boldsymbol{x}))}\right\}\right] d\boldsymbol{x}.$$

Assume that the deterministic function $G_{\boldsymbol{\eta}}$ is twice differentiable at $\boldsymbol{z} \in \mathscr{Z}$ and the first-order derivative satisfies $\|\partial G_{\boldsymbol{\eta}}/\partial \boldsymbol{z}\|_F^2 = o(\varepsilon\epsilon_n^2 n^{1-u})$ at $\boldsymbol{s}$, and the second-order derivative satisfies $\{\sum_{k=1}^{D_n}(\partial^2 G_{\boldsymbol{\eta}k}/\partial z_j^2)^2\}^{1/2} = o(\sqrt{D_n}\sum_{k=1}^{D_n}(\partial G_{\boldsymbol{\eta}k}/\partial z_j)^2)$ at $\boldsymbol{s}$, for $j = 1, \ldots, r_n$. Let $\|f_0 - f_{G_{\boldsymbol{\eta}}(\boldsymbol{s})}\|_\infty \leq \varepsilon\epsilon_n^2/4$ where $n\epsilon_n^2 \to \infty$. If $r_n \log(n) = o(n\epsilon_n^2)$, $D_n \sim n^u$, $\|\boldsymbol{\Upsilon}\|_2^2 = o(n\epsilon_n^2)$, then

$$\int h(\boldsymbol{z})q(\boldsymbol{z})d\boldsymbol{z} \leq \varepsilon\epsilon_n^2,$$

provided $\|\boldsymbol{\zeta}\|_\infty = O(n)$, $\|\boldsymbol{\zeta}^*\|_\infty = O(1)$ where $\boldsymbol{\zeta}^* = 1/\boldsymbol{\zeta}$.

Proof. Since $h(\boldsymbol{z})$ is a KL-distance, $h(\boldsymbol{z}) > 0$. We shall thus establish an upper bound. Let $\mathcal{A} = \{\boldsymbol{z} : \cap_{j=1}^{r_n} |z_j - s_j| \le \sqrt{\varepsilon\epsilon_n^2/r_n}\}$, then

$$\int h(\boldsymbol{z})q(\boldsymbol{z})d\boldsymbol{z} = \int_{\mathcal{A}} h(\boldsymbol{z})q(\boldsymbol{z})d\boldsymbol{z} + \int_{\mathcal{A}^c} h(\boldsymbol{z})q(\boldsymbol{z})d\boldsymbol{z}. \tag{52}$$

By the Taylor expansion, the first term is equal to

$$\begin{aligned}
&= \int_{\mathcal{A}} \left\{ h(\boldsymbol{s}) + (\boldsymbol{z} - \boldsymbol{s})^{\mathrm{T}} \nabla h(\boldsymbol{s}) + \frac{1}{2}(\boldsymbol{z} - \boldsymbol{s})^{\mathrm{T}} \nabla^2 h(\boldsymbol{s})(\boldsymbol{z} - \boldsymbol{s}) \right\} q(\boldsymbol{z})d\boldsymbol{z} + o(\varepsilon\epsilon_n^2) \\
&\le |h(\boldsymbol{s})| + \frac{1}{2}\int_{\mathcal{A}} (\boldsymbol{z} - \boldsymbol{s})^{\mathrm{T}} \nabla^2 h(\boldsymbol{s})(\boldsymbol{z} - \boldsymbol{s}) q(\boldsymbol{z})d\boldsymbol{z} + o(\varepsilon\epsilon_n^2) \\
&= \frac{\varepsilon\epsilon_n^2}{2} + \frac{1}{2}\int_{\mathcal{A}} (\boldsymbol{z} - \boldsymbol{s})^{\mathrm{T}} \nabla^2 h(\boldsymbol{s})(\boldsymbol{z} - \boldsymbol{s}) q(\boldsymbol{z})d\boldsymbol{z} + o(\varepsilon\epsilon_n^2) \qquad (53) \\
&= \frac{\varepsilon\epsilon_n^2}{2} + o(\varepsilon\epsilon_n^2) \\
&\le \frac{3\varepsilon\epsilon_n^2}{4},
\end{aligned}$$

where the second step holds because $q(\boldsymbol{z})$ is symmetric around $\boldsymbol{s}$. The third step holds in view of Lemma 7.8 in Bhattacharya et al. (2020) and the fact that $\boldsymbol{s}$ satisfies $\|f_{G_{\boldsymbol{\eta}}(\boldsymbol{s})} - f_0\|_\infty \le \varepsilon\epsilon_n^2/4$.
The final step is justified next. With $J = \{1, \ldots, r_n\}$, let $\nabla^2 h(\boldsymbol{s}) = ((b_{jj'}))_{j \in J, j' \in J}$,

$$\int_{\mathcal{A}} (\boldsymbol{z} - \boldsymbol{s})^{\mathrm{T}} \nabla^2 h(\boldsymbol{s})(\boldsymbol{z} - \boldsymbol{s}) q(\boldsymbol{z})d\boldsymbol{z} = \sum_{j=1}^{r_n} b_{jj} \int_{|z_j - s_j| \le \sqrt{\varepsilon\epsilon_n^2/r_n}} (z_j - s_j)^2 q(z_j)dz_j,$$

where the cross-covariance terms disappear since z_j are independent and $q(\boldsymbol{z})$ is symmetric around $\boldsymbol{s}$. Therefore,

$$\int_{\mathcal{A}} (\boldsymbol{z} - \boldsymbol{s})^{\mathrm{T}} \nabla^2 h(\boldsymbol{s})(\boldsymbol{z} - \boldsymbol{s}) q(\boldsymbol{z})d\boldsymbol{z} \le \sum_{j=1}^{r_n} b_{jj} \int (z_j - s_j)^2 q(z_j)dz_j = \frac{1}{n}\sum_{j=1}^{r_n} |b_{jj}|.$$

Using Lemma 1, we get

$$\int_{\mathcal{A}} (\boldsymbol{z} - \boldsymbol{s})^{\mathrm{T}} \nabla^2 h(\boldsymbol{s})(\boldsymbol{z} - \boldsymbol{s}) q(\boldsymbol{z})d\boldsymbol{z} \le \frac{1}{n}\left\{ D_n(2c^2 + 1) \left\| \frac{\partial G_{\boldsymbol{\eta}}}{\partial \boldsymbol{z}} \right\|_F^2 \right\} = o(\varepsilon\epsilon_n^2),$$

where the last equality holds since $D_n \sim n^u$ and $\|\partial G_{\boldsymbol{\eta}}/\partial \boldsymbol{z}\|_F^2 = o(\varepsilon\epsilon_n^2 n^{1-u})$.
We next handle the second term in (52). Using Lemma 7.8 in Bhattacharya et al. (2020), note that

$$\begin{aligned}
\int_{\mathcal{A}^c} h(\boldsymbol{z})q(\boldsymbol{z})d\boldsymbol{z} &\le 2\int_{\mathcal{A}^c} \left(\int_{\boldsymbol{x} \in [0,1]^p} |f_0(\boldsymbol{x}) - f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}(\boldsymbol{x})| d\boldsymbol{x} \right) q(\boldsymbol{z})d\boldsymbol{z} \\
&\le 2\int_{\boldsymbol{x} \in [0,1]^p} |f_0(\boldsymbol{x})| d\boldsymbol{x} \int_{\mathcal{A}^c} q(\boldsymbol{z})d\boldsymbol{z} + 2\int_{\mathcal{A}^c} \int_{\boldsymbol{x} \in [0,1]^p} |f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}(\boldsymbol{x})| d\boldsymbol{x} q(\boldsymbol{z})d\boldsymbol{z}.
\end{aligned}$$

First, note that using $|\sigma(\cdot)| \le 1$, we get $|f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}(\boldsymbol{x})| \le \sum_{t=1}^d |a_t^{\Upsilon}|$. Thus, $|f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}(\boldsymbol{x})| \le \sum_{t=1}^d |a_t^{\Upsilon}| + \sum_{t=1}^d |a_t - a_t^{\Upsilon}|$ which implies

$$\frac{1}{2}\int_{\mathcal{A}^c} h(\boldsymbol{z})q(\boldsymbol{z})d\boldsymbol{z} \le Q(\mathcal{A}^c) \left(\int_{\boldsymbol{x} \in [0,1]^p} |f_0(\boldsymbol{x})d\boldsymbol{x}| + \sum_{t=1}^{r_n} |a_t^{\Upsilon}| \right) + \int_{\mathcal{A}^c} \left(\sum_{t=1}^{d} |a_t - a_t^{\Upsilon}| \right) q(\boldsymbol{z})d\boldsymbol{z}. \tag{54}$$

First, note that $\mathcal{A}^c = \cup_{j=1}^{r_n} \mathcal{A}_j^c$ where $\mathcal{A}_j = \{|z_j - s_j| \leq \sqrt{\varepsilon\epsilon_n^2/r_n}\}$. Therefore,

$$\begin{aligned} Q(\mathcal{A}^c) = Q(\cup_{j=1}^{r_n}\mathcal{A}_j^c) \leq \sum_{j=1}^{r_n} Q(\mathcal{A}_j^c) &= \sum_{j=1}^{r_n} \int_{|z_j - s_j| > \sqrt{\varepsilon\epsilon_n^2/r_n}} q(z_j) dz_j \\ &= 2r_n \left\{ 1 - \Phi\left(\sqrt{\frac{n\varepsilon\epsilon_n^2}{r_n}} \right) \right\}. \end{aligned} \tag{55}$$

Using (55) in the first term of (54), we get

$$\begin{aligned} Q(\mathcal{A}^c) \left\{ \int_{\boldsymbol{x}\in[0,1]^p} |f_0(\boldsymbol{x})d\boldsymbol{x}| + \sum_{t=1}^{r_n} |a_t^{\boldsymbol{\Upsilon}}| \right\} &\lesssim 2(\|f_0\|_1 + \|\boldsymbol{\Upsilon}\|_1) r_n \left\{ 1 - \Phi\left(\sqrt{\frac{n\varepsilon\epsilon_n^2}{r_n}} \right) \right\} \\ &\leq 2(\|f_0\|_1 + \sqrt{r_n}\|\boldsymbol{\Upsilon}\|_2) r_n \left\{ 1 - \Phi\left(\sqrt{\frac{n\varepsilon\epsilon_n^2}{r_n}} \right) \right\} \\ &\leq 4n\epsilon_n^2 r_n \left\{ 1 - \Phi\left(\sqrt{\frac{n\varepsilon\epsilon_n^2}{r_n}} \right) \right\} \\ &\leq 4nr_n \left\{ 1 - \Phi\left(\sqrt{\frac{n\varepsilon\epsilon_n^2}{r_n}} \right) \right\} \\ &\sim 4nr_n \sqrt{\frac{r_n}{n\varepsilon\epsilon_n^2}} e^{-\frac{n\varepsilon\epsilon_n^2}{2r_n}} \quad \text{by Mill's ratio} \\ &\leq 4nr_n e^{-\frac{n\varepsilon\epsilon_n^2}{2r_n}} = o(n\epsilon_n^2), \end{aligned} \tag{56}$$

where the second step follows by the Cauchy–Schwartz inequality, the third step is satisfied because $\|\boldsymbol{\Upsilon}\|_2 = o(\sqrt{n\epsilon_n^2})$ and $\sqrt{r_n} = o(\sqrt{n\epsilon_n^2})$ and $\|f_0\|_1$ are fixed and the fourth step is satisfied because $\epsilon_n^2 \leq 1$. The last equality in the above steps holds because

$$-\frac{n\epsilon_n^2}{r_n} + \log(r_n) + \log(n) - \log(\varepsilon) \leq -\frac{n\epsilon_n^2}{r_n} + 3\log(n) = -\log(n) \left\{ \frac{n\epsilon_n^2}{r_n \log(n)} - 3 \right\} \to -\infty, \tag{57}$$

where the first inequality holds since $r_n \leq n$.

For the second term in (54),

$$\begin{aligned}
\int_{\mathcal{A}_c}\left(\sum_{t=1}^{d}|a_t - a_t^{\Upsilon}|\right)q(\boldsymbol{z})d\boldsymbol{z} &= \sum_{t=1}^{d}\int_{\mathcal{A}^c}|a_t - a_t^{\Upsilon}|q(\boldsymbol{z})d\boldsymbol{z}\\
&\leq \sum_{k=1}^{D_n}\int_{\mathcal{A}^c}|w_k - w_k^{\Upsilon}|q(\boldsymbol{z})d\boldsymbol{z}\\
&\leq \sum_{k=1}^{D_n}\sum_{j=1}^{r_n}\int_{\mathcal{A}^c}|z_j - s_j|q(\boldsymbol{z})d\boldsymbol{z}\\
&= \sum_{k=1}^{D_n}\sum_{j=1}^{r_n}\int_{|z_j-s_j|>\sqrt{\varepsilon\epsilon_n^2/r_n}}\sqrt{\frac{n}{2\pi}}|z_j - s_j|e^{-\frac{n}{2}|z_j-s_j|^2}dz_j\\
&= \frac{2}{\sqrt{n}}\sum_{k=1}^{D_n}\sum_{j=1}^{r_n}\int_{\sqrt{\varepsilon\epsilon_n^2/r_n}}^{\infty}\frac{u}{\sqrt{2\pi}}e^{-\frac{1}{2}u^2}du\\
&\leq 2D_n r_n e^{-\frac{n\varepsilon\epsilon_n^2}{2r_n}} = o(\varepsilon\epsilon_n^2),
\end{aligned}\tag{58}$$

where the last equality is a consequence of (57) and $D_n \sim n^u$.
Combining (54), (56) and (58), we get

$$\int_{\mathcal{A}^c} h(\boldsymbol{z})q(\boldsymbol{z})d\boldsymbol{z} = o(\varepsilon\epsilon_n^2) \leq \frac{\varepsilon\epsilon_n^2}{4}.$$

This together with (53) completes the proof. □

Proposition 4. *Let $n\epsilon_n^2 \to \infty$. Suppose $\pi_0(\boldsymbol{z})$ satisfies (2) with $\|\boldsymbol{\mu}\|_2^2 = o(n\epsilon_n^2)$ and $\|\boldsymbol{\zeta}\|_\infty = O(n)$. Suppose that for some $0 < b < 1$, $r_n\log(n) = o(n^b\epsilon_n^2)$, then for $\tilde{C}_n = e^{n^b\epsilon_n^2/r_n}$ and $\mathcal{F}_n$ as in (20), we have for any $\varepsilon > 0$,*

$$\int_{\boldsymbol{z}\in\mathcal{F}_n^c}\pi_0(\boldsymbol{z})d\boldsymbol{z} \leq e^{-n\varepsilon\epsilon_n^2}.$$

Proof. Let $\mathcal{F}_{jn} = \{z_j : |z_j| \leq \tilde{C}_n\}$.

$$\mathcal{F}_n = \cap_{j=1}^{r_n}\mathcal{F}_{jn} \Rightarrow \mathcal{F}_n^c = \cap_{j=1}^{r_n}\mathcal{F}_{jn}^c.$$

Note that

$$\begin{aligned}
\int_{\boldsymbol{z}\in\mathcal{F}_n^c}\pi_0(\boldsymbol{z})d\boldsymbol{z} &\leq \sum_{j=1}^{r_n}\int_{\mathcal{F}_{jn}^c}\frac{1}{\sqrt{2\pi\zeta_j^2}}e^{-\frac{(z_j-\mu_j)^2}{2\zeta_j^2}}dz_j\\
&= \sum_{j=1}^{r_n}\int_{-\infty}^{-\tilde{C}_n}\frac{1}{\sqrt{2\pi\zeta_j^2}}e^{-\frac{(z_j-\mu_j)^2}{2\zeta_j^2}}dz_j + \sum_{j=1}^{r_n}\int_{\tilde{C}_n}^{\infty}\frac{1}{\sqrt{2\pi\zeta_j^2}}e^{-\frac{(z_j-\mu_j)^2}{2\zeta_j^2}}dz_j\\
&= \sum_{j=1}^{r_n}\left\{1-\Phi\left(\frac{\tilde{C}_n-\mu_j}{\zeta_j}\right)\right\} + \sum_{j=1}^{r_n}\left\{1-\Phi\left(\frac{\tilde{C}_n+\mu_j}{\zeta_j}\right)\right\}.
\end{aligned}$$

Since $\|\boldsymbol{\mu}\|_2^2 = o(n\epsilon_n^2)$, this implies $\|\boldsymbol{\mu}\|_\infty = o(\sqrt{n}\epsilon_n)$. Also, $\|\boldsymbol{\zeta}\|_\infty = O(n)$, which implies for some $M > 0$,

$$\min\left\{\frac{|\tilde{C}_n - \mu_j|}{\zeta_j}, \frac{|\tilde{C}_n + \mu_j|}{\zeta_j}\right\} \geq \frac{(\tilde{C}_n - \sqrt{n})}{nM} \geq e^{\log(\tilde{C}_n) - 2\log(n)} - \frac{1}{\sqrt{n}M} \sim e^{R_n \log(n)} \to \infty, \tag{59}$$

where the last asymptotic relation holds because $1/\sqrt{n} \to 0$ and $R_n = (n^b\epsilon_n^2)/\{r_n \log(n)\} - 2 \to \infty$ since $r_n \log(n) = o(n^b\epsilon_n^2)$.
Thus, using Mill's ratio, we get

$$\begin{aligned}
\int_{\boldsymbol{z}\in\mathcal{F}_n^c} \pi_0(\boldsymbol{z})d\boldsymbol{z} &\lesssim \sum_{j=1}^{r_n} \frac{\zeta_j}{\tilde{C}_n - \mu_j} e^{-\frac{(\tilde{C}_n - \mu_j)^2}{2\zeta_j^2}} + \sum_{j=1}^{r_n} \frac{\zeta_j}{\tilde{C}_n + \mu_j} e^{-\frac{(\tilde{C}_n + \mu_j)^2}{2\zeta_j^2}} \\
&\leq 2r_n e^{-\frac{(\tilde{C}_n - \sqrt{n})^2}{2n^2M^2}} \lesssim e^{-\varepsilon n\epsilon_n^2},
\end{aligned}$$

where the last asymptotic inequality holds because

$$\frac{(\tilde{C}_n - \sqrt{n})^2}{2n^2M^2} - \log(2r_n) \gtrsim \frac{1}{2}e^{2R_n \log(n)} - 2\log(n) = n\left\{\frac{e^{R_n}}{2} - \frac{2\log(n)}{n}\right\} \geq \varepsilon n\epsilon_n^2.$$

In the above step, the first asymptotic inequality holds due to (59) and $r_n \leq n$. The last inequality holds since $R_n \to \infty$ and $\log(n)/n \to 0$. □

Proposition 5. *Let $n\epsilon_n^2 \to \infty$. Suppose $r_n \log(n) = o(n^b\epsilon_n^2)$ for some $0 < b < 1$, $D_n \log(n) = o(n^v\epsilon_n^2)$ for some $0 < v < 1$ and $\pi_0(\boldsymbol{z})$ satisfies (2) with $\|\boldsymbol{\mu}\|_2^2 = o(n\epsilon_n^2)$. Assume that $w_k = G_{\boldsymbol{\eta}k}(\boldsymbol{z})$ satisfies $|w_k| \leq C_n$, for $k = 1, \ldots, D_n$. Furthermore, assume that the deterministic function $G_{\boldsymbol{\eta}}$ is differentiable at $\boldsymbol{z} \in \mathscr{Z}$ and the first-order derivative satisfies $\|\partial G_{\boldsymbol{\eta}}/\partial \boldsymbol{z}\|_F^2 = o(\varepsilon\epsilon_n^2 n^{1-u})$ for $\boldsymbol{z} \in \mathcal{F}_n$. Then, for every $\varepsilon > 0$,*

$$\log\left\{\int_{\mathcal{V}_{\varepsilon\epsilon_n}^c} \frac{L(\mathcal{D}_n; f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})})}{L_0}\pi_0(\boldsymbol{z})d\boldsymbol{z}\right\} \leq \log 2 - \varepsilon^2 n\epsilon_n^2 + o_{P_0}(1).$$

Proof. In this direction, we first show

$$P_0\left(\int_{\mathcal{V}_{\varepsilon\epsilon_n}^c} \frac{L(\mathcal{D}_n; f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})})}{L_0}\pi_0(\boldsymbol{z})d\boldsymbol{z} > 2e^{-\varepsilon^2 n\epsilon_n^2}\right) \to 0, \quad n \to \infty. \tag{60}$$

Note that

$$\begin{aligned}
&P_0\left(\int_{\mathcal{V}_{\varepsilon\epsilon_n}^c} \frac{L(\mathcal{D}_n; f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})})}{L_0}\pi_0(\boldsymbol{z})d\boldsymbol{z} > 2e^{-\varepsilon^2 n\epsilon_n^2}\right) \\
&\leq P_0\left(\int_{\mathcal{V}_{\varepsilon\epsilon_n}^c \cap \mathcal{F}_n} \frac{L(\mathcal{D}_n; f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})})}{L_0}\pi_0(\boldsymbol{z})d\boldsymbol{z} > e^{-\varepsilon^2 n\epsilon_n^2}\right) \\
&+ P_0\left(\int_{\mathcal{F}_n^c} \frac{L(\mathcal{D}_n; f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})})}{L_0}\pi_0(\boldsymbol{z})d\boldsymbol{z} > e^{-\varepsilon^2 n\epsilon_n^2}\right).
\end{aligned}$$

Using Lemma 2 with $\varepsilon = \varepsilon\epsilon_n$ and $C_n = e^{n^v\epsilon_n^2/r_n}$, $\tilde{C}_n = e^{n^b\epsilon_n^2/r_n}$,

$$
\begin{aligned}
&\int_{\varepsilon^2/8}^{\sqrt{2}\varepsilon} \sqrt{H_(u, \tilde{\mathcal{F}}_n, \|\cdot\|_2)}du \\
&\lesssim \varepsilon\epsilon_n\sqrt{2r_n\{\log(r_n) + \log(D_n) + \log(C_n) + \log(\tilde{C}_n) - \log(\epsilon_n)\}} \\
&\leq \varepsilon\epsilon_n O(\max\{\sqrt{r_n\log(r_n)}, \sqrt{r_n\log(D_n)}, \sqrt{r_n\log(C_n)}, \sqrt{r_n\log(\tilde{C}_n)}, \sqrt{-\log(\epsilon_n)}\}) \\
&\leq \varepsilon\epsilon_n \max\{o(\sqrt{n\epsilon_n}), o(\sqrt{n\epsilon_n}), O(\sqrt{n^v}\epsilon_n), O(\sqrt{n^b}\epsilon_n), O(\sqrt{\log(n)})\} \\
&\leq \varepsilon^2\epsilon_n^2\sqrt{n}.
\end{aligned}
$$

The first inequality in the third step follows because $D_n \leq n$, $D_n\log(n) = o(n\epsilon_n)$ and $r_n\log(n) = o(n\epsilon_n)$, $r_n\log(C_n) = r_n(n^v\epsilon_n^2/r_n)$ and $r_n\log(\tilde{C}_n) = r_n(n^b\epsilon_n^2/r_n)$, $1/\epsilon_n^2 = o(n)$, then $-\log(\epsilon_n^2) \leq \log(n)$. The second inequality in the third step follows since $n^v/n = o(1)$, $n^b/n = o(1)$ and $\log(n) = o(n\epsilon_n^2)$.

By Theorem 1 in Wong and Shen (1995), for some constant $C > 0$, we have

$$
\begin{aligned}
P_0\left(\int_{\mathcal{V}_{\varepsilon\epsilon_n}^c\cap\mathcal{F}_n} \frac{L(\mathcal{D}_n; f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})})}{L_0}\pi_0(\boldsymbol{z})d\boldsymbol{z} > e^{-\varepsilon^2n\epsilon_n^2}\right) &\leq P_0\left(\sup_{\boldsymbol{z}\in\mathcal{V}_{\varepsilon\epsilon_n}^c\cap\mathcal{F}_n} \frac{L(\mathcal{D}_n; f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})})}{L_0} > e^{-\varepsilon^2n\epsilon_n^2}\right) \\
&\leq 4\exp(-C\varepsilon^2n\epsilon_n^2) \to 0.
\end{aligned}
\tag{61}
$$

Using Proposition 4 with $\varepsilon = 2\varepsilon$, we have

$$
\int_{\boldsymbol{z}\in\mathcal{F}_n^c} \pi_0(\boldsymbol{z})d\boldsymbol{z} \leq e^{-2n\varepsilon^2\epsilon_n^2}.
$$

Therefore, using Lemma 7.6 in Bhattacharya et al. (2020) with $\varepsilon = 2\varepsilon^2\epsilon_n^2$ and $\tilde{\varepsilon} = \varepsilon^2\epsilon_n^2$, we have

$$
P_0\left(\int_{\mathcal{F}_n^c} \frac{L(\mathcal{D}_n; f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})})}{L_0}\pi_0(\boldsymbol{z})d\boldsymbol{z} > e^{-\varepsilon^2n\epsilon_n^2}\right) \leq e^{-\varepsilon^2n\epsilon_n^2} \to 0. \tag{62}
$$

Combining (61) and (62), (60) follows.

Finally, to complete the proof, let (I) $= \log[\int_{\mathcal{V}_{\varepsilon\epsilon_n}^c}\{L(\mathcal{D}_n \mid G_{\boldsymbol{\eta}}(\boldsymbol{z}))/L_0\}\pi_0(\boldsymbol{z})d\boldsymbol{z}]$.

$$
\begin{aligned}
\text{(I)} &= \text{(I)} \times \mathbb{1}(\text{(I)} \leq \log 2 - \varepsilon^2\epsilon_n^2) + \text{(I)} \times \mathbb{1}(\text{(I)} \geq \log 2 - \varepsilon^2\epsilon_n^2) \\
&\leq \log 2 - \varepsilon^2\epsilon_n^2 + \underbrace{\text{(I)} * \mathbb{1}(\text{(I)} > \log 2 - \varepsilon^2\epsilon_n^2)}_{\text{(II)}} \\
&= \log 2 - \varepsilon^2\epsilon_n^2 + o_{P_0}(1),
\end{aligned}
$$

where the last equality follows from (60) as below

$$
P_0(|\text{(II)}| > \nu) \leq P_0(\mathbb{1}(\text{(I)} > \log 2 - \varepsilon^2\epsilon_n^2) = 1) = P_0(\text{(I)} > \log 2 - \varepsilon^2\epsilon_n^2) \to 0.
$$

□

Proposition 6. *Let* $\pi_0(\boldsymbol{z})$ *satisfy* (2) *with* $\|\boldsymbol{\zeta}\|_\infty = O(n)$ *and* $\|\boldsymbol{\zeta}^*\|_\infty = O(1)$, $\boldsymbol{\zeta}^* = 1/\boldsymbol{\zeta}$. *Assume that the deterministic function* $G_{\boldsymbol{\eta}}$ *is twice differentiable at* $\boldsymbol{z} \in \mathscr{Z}$ *and the first-order derivative satisfies* $\|\partial G_{\boldsymbol{\eta}}/\partial\boldsymbol{z}\|_F^2 = o(\varepsilon\epsilon_n^2n^{1-u})$ *at* $\boldsymbol{s}$, *and the second-order derivative satisfies*

$$
\{\sum_{k=1}^{D_n}(\partial^2G_{\boldsymbol{\eta}k}/\partial z_j^2)^2\}^{1/2} = o(\sqrt{D_n}\sum_{k=1}^{D_n}(\partial G_{\boldsymbol{\eta}k}/\partial z_j)^2)
$$

at $\boldsymbol{s}$, *for* $j = 1, \ldots, r_n$.

1. *If* $r_n \log(n) = o(n)$, $D_n \log(n) = o(n)$ *and* $\|\boldsymbol{\mu}\|_2^2 = o(n)$, *then*
$$d_{\mathrm{KL}}(q^*, p_{\boldsymbol{\eta}}(\cdot \mid \mathcal{D}_n)) = o_{P_0}(n).$$
2. *If* $r_n \log(n) = o(n\epsilon_n^2)$, $D_n \log(n) = o(n\epsilon_n^2)$ *and* $\|\boldsymbol{\mu}\|_2^2 = o(n\epsilon_n^2)$ *and there exists a neural network such that* $\|f_0 - f_{\boldsymbol{\Upsilon}}\|_\infty = o(\epsilon_n^2)$, $\|\boldsymbol{\Upsilon}\|_2^2 = o(n\epsilon_n^2)$, $\|\boldsymbol{s}\|_2^2 = o(n\epsilon_n^2)$, *then*
$$d_{\mathrm{KL}}(q^*, p_{\boldsymbol{\eta}}(\cdot \mid \mathcal{D}_n)) = o_{P_0}(n\epsilon_n^2).$$

Proof. For any $q \in \mathcal{Q}$,

$$\begin{aligned}
d_{\mathrm{KL}}(q, p_{\boldsymbol{\eta}}(\cdot \mid \mathcal{D}_n)) &= \int q(\boldsymbol{z}) \log\{q(\boldsymbol{z})\} d\boldsymbol{z} - \int q(\boldsymbol{z}) \log\{p_{\boldsymbol{\eta}}(\boldsymbol{z} \mid \mathcal{D}_n)\} d\boldsymbol{z} \\
&= \int q(\boldsymbol{z}) \log\{q(\boldsymbol{z})\} d\boldsymbol{z} - \int q(\boldsymbol{z}) \log\left\{ \frac{L(\mathcal{D}_n; f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}) \pi_0(\boldsymbol{z})}{\int L(\mathcal{D}_n; f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}) \pi_0(\boldsymbol{z}) d\boldsymbol{z}} \right\} d\boldsymbol{z} \\
&= d_{\mathrm{KL}}(q, \pi_0) - \int \log\left\{ \frac{L(\mathcal{D}_n; f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})})}{L_0} \right\} q(\boldsymbol{z}) d\boldsymbol{z} \\
&+ \log\left\{ \int \frac{L(\mathcal{D}_n; f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})})}{L_0} \pi_0(\boldsymbol{z}) d\boldsymbol{z} \right\} \\
&\le d_{\mathrm{KL}}(q, \pi_0) + \left| \int \log\left\{ \frac{L(\mathcal{D}_n; f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})})}{L_0} \right\} q(\boldsymbol{z}) d\boldsymbol{z} \right| \\
&+ \left| \log\left\{ \int \frac{L(\mathcal{D}_n; f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})})}{L_0} \pi_0(\boldsymbol{z}) d\boldsymbol{z} \right\} \right|.
\end{aligned} \tag{63}$$

Since q^* satisfies the minimized KL-distance to $p_{\boldsymbol{\eta}}(\cdot \mid \mathcal{D}_n)$ in the family $\mathcal{Q}$, therefore,

$$P_0(d_{\mathrm{KL}}(q^*, p_{\boldsymbol{\eta}}(\cdot \mid \mathcal{D}_n)) > \kappa) \le P_0(d_{\mathrm{KL}}(q, p_{\boldsymbol{\eta}}(\cdot \mid \mathcal{D}_n)) > \kappa), \tag{64}$$

for any $\kappa > 0$.

Proof of part 1. Note that $r_n \log(n) = o(n)$, $D_n \log(n) = o(n)$, $\|\boldsymbol{\mu}\|_2^2 = o(n)$, $\|\boldsymbol{\zeta}\|_\infty = O(n)$ and $\|\boldsymbol{\zeta}^*\|_\infty = O(1)$. We take $q(\boldsymbol{z}) = N(\boldsymbol{s}, \mathbf{I}_{r_n}/\sqrt{n})$ where $\boldsymbol{s}$ is defined next.

For $N \ge 1$, let $f_{\boldsymbol{\Upsilon}_N}$ be a neural network that satisfies $\|f_{\boldsymbol{\Upsilon}_N} - f_0\|_\infty \le \varepsilon/4$. The existence of such a neural network is always guaranteed by Hornik et al. (1989b). Define $\boldsymbol{\Upsilon}$ as

$$a_j^{\boldsymbol{\Upsilon}} = \begin{cases} a_j^{\boldsymbol{\Upsilon}_N}, & j = 1, \ldots, d_N \\ 0, & j = d_N + 1, \ldots, d \end{cases} \qquad \boldsymbol{\omega}_j^{\boldsymbol{\Upsilon}} = \begin{cases} \boldsymbol{\omega}_j^{\boldsymbol{\Upsilon}_N}, & j = 1, \ldots, d_N \\ 0, & j = d_N + 1, \ldots, d, \end{cases}$$

and let $d_N = r_n/2$, we could rewrite $\boldsymbol{\Upsilon}$ as

$$\boldsymbol{\Upsilon}_k = \begin{cases} (G_{\boldsymbol{\eta}}(\boldsymbol{s}))_k, & k = 1, \ldots, r_n \\ 0, & k = r_n + 1, \ldots, D_n, \end{cases}$$

where $(G_{\boldsymbol{\eta}}(\boldsymbol{s}))_k$ is the kth element of the vector $\boldsymbol{\Upsilon} = G_{\boldsymbol{\eta}}(\boldsymbol{s})$. This choice guarantees $\|f_{G_{\boldsymbol{\eta}}(\boldsymbol{s})} - f_0\|_\infty = \|f_{\boldsymbol{\Upsilon}} - f_0\|_\infty \le \varepsilon/4$.

Step 1 (a): Since $\|\boldsymbol{\Upsilon}\|_2^2 = \|\boldsymbol{\Upsilon}_N\|_2^2$ is bounded, which implies $\|\boldsymbol{\Upsilon}\|_2^2 = o(n)$.

By the expression of $\boldsymbol{\Upsilon} = G_{\boldsymbol{\eta}}(\boldsymbol{s})$, $G_{\boldsymbol{\eta}}(\cdot)$ is invertible at $\boldsymbol{s}$ and thus $\boldsymbol{s} = h(\boldsymbol{\Upsilon})$ where $G_{\boldsymbol{\eta}}(h(\boldsymbol{\Upsilon})) = \boldsymbol{\Upsilon}$.

Denote $\tilde{h}(\cdot) = h(\cdot) - h(\mathbf{0})$. By the Taylor expansion,

$$\begin{aligned}
\boldsymbol{s} &\approx h(\mathbf{0}) + (\boldsymbol{\Upsilon} - \mathbf{0})^{\mathrm{T}} \nabla h(\mathbf{0}) \\
&= h(\mathbf{0}) + \boldsymbol{\Upsilon}^{\mathrm{T}} \{\nabla G_{\boldsymbol{\eta}}(\boldsymbol{s}_0)\}^{-1},
\end{aligned}$$

where $\boldsymbol{s}_0 = h(\mathbf{0})$ the last equation follows by the inverse function theorem. Therefore,

$$\|\boldsymbol{s}\|_2^2 \leq \|\boldsymbol{s}_0\|_2^2 + \|\boldsymbol{\Upsilon}\|_2^2 \|\{\nabla G_{\boldsymbol{\eta}}(\boldsymbol{s}_0)\}^{-1}\|_F^2,$$

which implies $\|\boldsymbol{s}\|_2^2 = o(n)$ since $\|\boldsymbol{\Upsilon}\|_2^2 = o(n)$ and $\|\{\nabla G_{\boldsymbol{\eta}}(\boldsymbol{s}_0)\}^{-1}\|_F^2 = O(1)$.
Using Proposition 1, with $\epsilon_n = 1$, we get for any $\nu > 0$,

$$d_{\mathrm{KL}}(q, \pi_0) \leq n\nu,$$

where the above step follows by $\|\boldsymbol{s}\|_2^2 = o(n)$. Therefore,

$$P_0(d_{\mathrm{KL}}(q, \pi_0) > n\nu) = 0. \tag{65}$$

Step 1 (b): Next, note that

$$\begin{aligned} d_{\mathrm{KL}}(\ell_0, \ell_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}) &= \int_{\boldsymbol{x} \in [0,1]^p} \sigma(f_0(\boldsymbol{x})) \log\left\{\frac{\sigma(f_0(\boldsymbol{x}))}{\sigma(f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}(\boldsymbol{x}))}\right\} d\boldsymbol{x} \\ &+ \int_{\boldsymbol{x} \in [0,1]^p} \{1 - \sigma(f_0(\boldsymbol{x}))\} \log\left\{\frac{1 - \sigma(f_0(\boldsymbol{x}))}{1 - \sigma(f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}(\boldsymbol{x}))}\right\} d\boldsymbol{x} \\ &= \int_{\boldsymbol{x} \in [0,1]^p} \left[\sigma(f_0(\boldsymbol{x}))\{f_0(\boldsymbol{x}) - f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}(\boldsymbol{x})\} + \log\left\{\frac{1 - \sigma(f_0(\boldsymbol{x}))}{1 - \sigma(f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}(\boldsymbol{x}))}\right\}\right] d\boldsymbol{x}. \end{aligned}$$

Since $\|f_{\boldsymbol{\Upsilon}} - f_0\|_\infty \leq \varepsilon/4$, using Proposition 3 with $\epsilon_n = 1$ and $\varepsilon = \varepsilon$,

$$\int d_{\mathrm{KL}}(\ell_0, \ell_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}) q(\boldsymbol{z}) d\boldsymbol{z} \leq \varepsilon,$$

where the above step follows since $\|\boldsymbol{\Upsilon}\|_2^2 = \|\boldsymbol{\Upsilon}_N\|_2^2$ is bounded, which implies $\|\boldsymbol{\Upsilon}\|_2^2 = o(n)$. Therefore, by Lemma 7.4 in Bhattacharya et al. (2020),

$$P_0\left(\left|\int \log\left\{\frac{L(\mathcal{D}_n; f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})})}{L_0}\right\} q(\boldsymbol{z}) d\boldsymbol{z}\right| > n\nu\right) \leq \frac{\varepsilon}{\nu}. \tag{66}$$

Step 1 (c): Since $\|f_{\boldsymbol{\Upsilon}} - f_0\|_\infty \leq \varepsilon/4$, therefore, using Proposition 2 with $\epsilon_n = 1$ and $\nu = \varepsilon$, we get

$$\int_{\boldsymbol{z} \in \mathcal{L}_\varepsilon} \pi_0(\boldsymbol{z}) d\boldsymbol{z} \geq \exp(-n\varepsilon),$$

where the above step follows $\|\boldsymbol{\Upsilon}\|_2^2 = \|\boldsymbol{\Upsilon}_N\|_2^2$ is bounded which implies $\|\boldsymbol{\Upsilon}\|_2^2 = o(n)$ and $\|\boldsymbol{s}\|_2^2 = o(n)$.
Therefore, using Lemma 7.5 in Bhattacharya et al. (2020), we get

$$P_0\left(\left|\log\left\{\int \frac{L(\mathcal{D}_n; f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})})}{L_0} \pi_0(\boldsymbol{z}) d\boldsymbol{z}\right\}\right| > n\nu\right) \leq \frac{2\varepsilon}{\nu}. \tag{67}$$

Step 1 (d): From (63) and (64), we get

$$\begin{aligned} P_0(d_{\mathrm{KL}}(q^*, p_{\boldsymbol{\eta}}(\cdot \mid \mathcal{D}_n)) > 3n\nu) &\leq P_0(d_{\mathrm{KL}}(q, p_{\boldsymbol{\eta}}(\cdot \mid \mathcal{D}_n)) > n\nu) \\ &+ P_0\left(\left|\int \log\left\{\frac{L(\mathcal{D}_n; f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})})}{L_0}\right\} q(\boldsymbol{z}) d\boldsymbol{z}\right| > n\nu\right) \\ &+ P_0\left(\left|\log\left\{\int \frac{L(\mathcal{D}_n; f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})})}{L_0} \pi_0(\boldsymbol{z}) d\boldsymbol{z}\right\}\right| > n\nu\right) \\ &\leq \frac{2\varepsilon}{\nu}, \end{aligned}$$

where the last inequality is a consequence of (65), (66) and (67). Since ε is arbitrary, taking $\varepsilon \to 0$ completes the proof.

Proof of part 2. Note that $r_n \log(n) = o(n\epsilon_n^2)$, $D_n \log(n) = o(n\epsilon_n^2)$, $\|\boldsymbol{\mu}\|_2^2 = o(n\epsilon_n^2)$, $\|\boldsymbol{\zeta}\|_\infty = O(n)$ and $\|\boldsymbol{\zeta}^*\|_\infty = O(1)$. We take $q(\boldsymbol{z}) = N(\boldsymbol{s}, \mathbf{I}_{r_n}/\sqrt{n})$ where $\boldsymbol{s}$ is defined next. For $N \geq 1$, let $f_{\boldsymbol{\Upsilon}_N}$ be a neural network that satisfies $\|f_{\boldsymbol{\Upsilon}_N} - f_0\|_\infty \leq \varepsilon\epsilon_n^2/4$. The existence of such a neural network is always guaranteed by Hornik et al. (1989b). Define $\boldsymbol{\Upsilon}$ as

$$a_j^{\boldsymbol{\Upsilon}} = \begin{cases} a_j^{\boldsymbol{\Upsilon}_N}, & j = 1, \ldots, d_N \\ 0, & j = d_N + 1, \ldots, d \end{cases} \qquad \boldsymbol{\omega}_j^{\boldsymbol{\Upsilon}} = \begin{cases} \boldsymbol{\omega}_j^{\boldsymbol{\Upsilon}_N}, & j = 1, \ldots, d_N \\ 0, & j = d_N + 1, \ldots, d, \end{cases}$$

and let $d_N = r_n/2$, we could rewrite $\boldsymbol{\Upsilon}$ as

$$\boldsymbol{\Upsilon}_k = \begin{cases} (G_{\boldsymbol{\eta}}(\boldsymbol{s}))_k, & k = 1, \ldots, r_n \\ 0, & k = r_n + 1, \ldots, D_n, \end{cases}$$

where $(G_{\boldsymbol{\eta}}(\boldsymbol{s}))_k$ is the kth element of the vector $\boldsymbol{\Upsilon} = G_{\boldsymbol{\eta}}(\boldsymbol{s})$. This choice guarantees $\|f_{G_{\boldsymbol{\eta}}(\boldsymbol{s})} - f_0\|_\infty = \|f_{\boldsymbol{\Upsilon}} - f_0\|_\infty \leq \varepsilon\epsilon_n^2/4$.

Step 2 (a): Since $\|\boldsymbol{\Upsilon}\|_2^2 = \|\boldsymbol{\Upsilon}_N\|_2^2$ is bounded, this implies $\|\boldsymbol{\Upsilon}\|_2^2 = o(n\epsilon_n^2)$.

By the expression of $\boldsymbol{\Upsilon} = G_{\boldsymbol{\eta}}(\boldsymbol{s})$, $G_{\boldsymbol{\eta}}(\cdot)$ is invertible at $\boldsymbol{s}$ and thus $\boldsymbol{s}_n = h(\boldsymbol{\Upsilon})$ where $G_{\boldsymbol{\eta}}(h(\boldsymbol{\Upsilon})) = \boldsymbol{\Upsilon}$.

Denote $\tilde{h}(\cdot) = h(\cdot) - h(\mathbf{0})$. By the Taylor expansion,

$$\begin{aligned} \boldsymbol{s} &\approx h(\mathbf{0}) + (\boldsymbol{\Upsilon} - \mathbf{0})^{\mathrm{T}} \nabla h(\mathbf{0}) \\ &= h(\mathbf{0}) + \boldsymbol{\Upsilon}^{\mathrm{T}} \{\nabla G_{\boldsymbol{\eta}}(\boldsymbol{s}_0)\}^{-1}, \end{aligned}$$

where $\boldsymbol{s}_0 = h(\mathbf{0})$ the last equation follows by the inverse function theorem.

Therefore,

$$\|\boldsymbol{s}\|_2^2 \leq \|\boldsymbol{s}_0\|_2^2 + \|\boldsymbol{\Upsilon}\|_2^2 \|\{\nabla G_{\boldsymbol{\eta}}(\boldsymbol{s}_0)\}^{-1}\|_F^2,$$

which implies $\|\boldsymbol{s}\|_2^2 = o(n\epsilon_n^2)$ since $\|\boldsymbol{\Upsilon}\|_2^2 = o(n\epsilon_n^2)$ and $\|\{\nabla G_{\boldsymbol{\eta}}(\boldsymbol{s}_0)\}^{-1}\|_F^2 = O(1)$.

Using Proposition 1, we get for any $\nu > 0$,

$$d_{\mathrm{KL}}(q, \pi_0) \leq n\epsilon_n^2 \nu,$$

where the above step follows by $\|\boldsymbol{s}\|_2^2 = o(n\epsilon_n^2)$. Therefore,

$$P_0(d_{\mathrm{KL}}(q, \pi_0) > n\epsilon_n^2 \nu) = 0. \tag{68}$$

Step 2 (b): Since $\|f_{\boldsymbol{\Upsilon}} - f_0\|_\infty \leq \varepsilon\epsilon_n^2/4$ and $\|\boldsymbol{\Upsilon}\|_2^2 = o(n\epsilon_n^2)$, by Proposition 3,

$$\int d_{\mathrm{KL}}(\ell_0, \ell_{G_{\boldsymbol{\eta}}(\boldsymbol{z})}) q(\boldsymbol{z}) d\boldsymbol{z} \leq \varepsilon\epsilon_n^2,$$

Therefore, by Lemma 7.4 in Bhattacharya et al. (2020),

$$P_0\left(\left| \int \log \left\{ \frac{L(\mathcal{D}_n; f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})})}{L_0} \right\} q(\boldsymbol{z}) d\boldsymbol{z} \right| > n\epsilon_n^2 \nu \right) \leq \frac{\varepsilon}{\nu}. \tag{69}$$

Step 2 (c): Since $\|f_{\boldsymbol{\Upsilon}} - f_0\|_\infty \leq \varepsilon\epsilon_n^2/4$, $\|\boldsymbol{\Upsilon}\|_2^2 = o(n\epsilon_n^2)$ and $\|\boldsymbol{s}\|_2^2 = o(n\epsilon_n^2)$, by Proposition 2,

$$\int_{\boldsymbol{z} \in \mathcal{L}_\varepsilon} \pi_0(\boldsymbol{z}) d\boldsymbol{z} \geq \exp(-\varepsilon n\epsilon_n^2),$$

Therefore, using Lemma 7.5 in Bhattacharya et al. (2020), we get

$$P_0\left(\left|\log\left\{\int \frac{L(\mathcal{D}_n; f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})})}{L_0}\pi_0(\boldsymbol{z})d\boldsymbol{z}\right\}\right| > n\epsilon_n^2\nu\right) \le \frac{2\varepsilon}{\nu}. \tag{70}$$

Step 2 (d): From (63) and (64), we get

$$\begin{aligned}P_0(d_{\mathrm{KL}}(q^*, p_{\boldsymbol{\eta}}(\cdot \mid \mathcal{D}_n)) > 3n\epsilon_n^2\nu) &\le P_0(d_{\mathrm{KL}}(q, p_{\boldsymbol{\eta}}(\cdot \mid \mathcal{D}_n)) > n\epsilon_n^2\nu) \\ &+ P_0\left(\left|\int \log\left\{\frac{L(\mathcal{D}_n; f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})})}{L_0}\right\} q(\boldsymbol{z})d\boldsymbol{z}\right| > n\epsilon_n^2\nu\right) \\ &+ P_0\left(\left|\log\left\{\int \frac{L(\mathcal{D}_n; f_{G_{\boldsymbol{\eta}}(\boldsymbol{z})})}{L_0}\pi_0(\boldsymbol{z})d\boldsymbol{z}\right\}\right| > n\epsilon_n^2\nu\right) \\ &\le \frac{2\varepsilon}{\nu},\end{aligned}$$

where the last inequality is a consequence of (68), (69) and (70).
Since ε is arbitrary, taking $\varepsilon \to 0$ completes the proof. □

References

Atanov, A., Ashukha, A., Struminsky, K., Vetrov, D., and Welling, M. (2018). The deep weight prior. *arXiv preprint arXiv:1810.06943*.

Atchadé, Y. (2011). A computational framework for empirical bayes inference. *Statistics and Computing*, 21:463–473.

Bai, J., Song, Q., and Cheng, G. (2020). Efficient variational inference for sparse deep learning with theoretical guarantee. *Advances in Neural Information Processing Systems*, 33:466–476.

Basu, S., Karki, M., Ganguly, S., DiBiano, R., Mukhopadhyay, S., Gayaka, S., Kannan, R., and Nemani, R. (2017). Learning sparse feature representations using probabilistic quadtrees and deep belief nets. *Neural Processing Letters*, 45:855–867.

Bernardo, J. M. and Smith, A. F. (2009). *Bayesian theory*, volume 405. John Wiley & Sons.

Bhattacharya, S., Liu, Z., and Maiti, T. (2020). Variational bayes neural network: Posterior consistency, classification accuracy and computational challenges. *arXiv preprint arXiv:2011.09592*.

Bhattacharya, S. and Maiti, T. (2021). Statistical foundation of variational bayes neural networks. *Neural networks : the official journal of the International Neural Network Society*, 137:151–173.

Bishop, C. M. (1997). Bayesian neural networks. *Journal of the Brazilian Computer Society*, 4:61–68.

Blei, D. M. and Lafferty, J. D. (2007). A correlated topic model of science. *The annals of applied statistics*, 1(1):17–35.

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

Carlin, B. P. and Louis, T. A. (2008). *Bayesian methods for data analysis.* CRC Press.

Chen, Y., Gao, Q., and Wang, X. (2022). Inferential wasserstein generative adversarial networks. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 84(1):83–113.

Ching, T., Himmelstein, D. S., Beaulieu-Jones, B. K., Kalinin, A. A., Do, B. T., Way, G. P., Ferrero, E., Agapow, P.-M., Zietz, M., Hoffman, M. M., et al. (2018). Opportunities and obstacles for deep learning in biology and medicine. *Journal of The Royal Society Interface*, 15(141):20170387.

Dua, D. and Graff, C. (2017). UCI machine learning repository.

Dusenberry, M. W., Jerfel, G., Wen, Y., Ma, Y.-A., Snoek, J., Heller, K., Lakshminarayanan, B., and Tran, D. (2020). Efficient and scalable bayesian neural nets with rank-1 factors. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

Efron, B. (2012). *Large-scale inference: empirical Bayes methods for estimation, testing, and prediction*, volume 1. Cambridge University Press.

Efron, B. and Morris, C. (1973). Stein's estimation rule and its competitors—an empirical bayes approach. *Journal of the American Statistical Association*, 68(341):117–130.

Efron, B., Tibshirani, R., Storey, J. D., and Tusher, V. (2001). Empirical bayes analysis of a microarray experiment. *Journal of the American statistical association*, 96(456):1151–1160.

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

Ghosh, S., Yao, J., and Doshi-Velez, F. (2019). Model selection in bayesian neural networks via horseshoe priors. *J. Mach. Learn. Res.*, 20(182):1–46.

Glorot, X., Bordes, A., and Bengio, Y. (2011). Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 315–323.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, 27.

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

Han, X., Zheng, H., and Zhou, M. (2022). Card: Classification and regression diffusion models. *arXiv preprint arXiv:2206.07275.*

Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer.

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

Hernández-Lobato, J. M. and Adams, R. (2015). Probabilistic backpropagation for scalable learning of bayesian neural networks. In *International conference on machine learning*, pages 1861–1869. PMLR.

Hernández-Lobato, J. M. and Adams, R. P. (2015). Probabilistic backpropagation for scalable learning of bayesian neural networks. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 1861–1869. JMLR.org.

Hinton, G. E. and Van Camp, D. (1993). Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the sixth annual conference on Computational learning theory*, pages 5–13.

Hoffman, J., Roberts, D. A., and Yaida, S. (2019). Robust learning with jacobian regularization. *arXiv preprint arXiv:1908.02729*.

Hornik, K., Stinchcombe, M., and White, H. (1989a). Multilayer feedforward networks are universal approximators. *Neural Network*, 2(5):359–366.

Hornik, K., Stinchcombe, M., and White, H. (1989b). Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366.

Hubin, A., Storvik, G., and Frommlet, F. (2018). Deep bayesian regression models. *arXiv preprint arXiv:1806.02160*.

Immer, A., Bauer, M., Fortuin, V., Rätsch, G., and Emtiyaz, K. M. (2021). Scalable marginal likelihood estimation for model selection in deep learning. In *International Conference on Machine Learning*, pages 4563–4573. PMLR.

Izmailov, P., Vikram, S., Hoffman, M. D., and Wilson, A. G. G. (2021). What are bayesian neural network posteriors really like? In *International conference on machine learning*, pages 4629–4640. PMLR.

Javid, K., Handley, W., Hobson, M., and Lasenby, A. (2020). Compromise-free bayesian neural networks. *arXiv preprint arXiv:2004.12211*.

Kamnitsas, K., Ledig, C., Newcombe, V. F., Simpson, J. P., Kane, A. D., Menon, D. K., Rueckert, D., and Glocker, B. (2017). Efficient multi-scale 3d cnn with fully connected crf for accurate brain lesion segmentation. *Medical image analysis*, 36:61–78.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90.

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

Lampinen, J. and Vehtari, A. (2001). Bayesian approach for neural networks—review and case studies. *Neural networks*, 14(3):257–274.

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

Leibig, C., Allken, V., Ayhan, M. S., Berens, P., and Wahl, S. (2017). Leveraging uncertainty information from deep neural networks for disease detection. *Scientific reports*, 7(1):17816.

Louizos, C., Ullrich, K., and Welling, M. (2017). Bayesian compression for deep learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 3290–3300, Red Hook, NY, USA. Curran Associates Inc.

MacKay, D. J. (1995). Probable networks and plausible predictions-a review of practical bayesian methods for supervised neural networks. *Network: computation in neural systems*, 6(3):469.

Molchanov, D., Ashukha, A., and Vetrov, D. (2017). Variational dropout sparsifies deep neural networks. In *International Conference on Machine Learning*, pages 2498–2507. PMLR.

Mullachery, V., Khera, A., and Husain, A. (2018). Bayesian neural networks. *arXiv preprint arXiv:1801.07710*.

Neal, R. M. (1992). Bayesian training of backpropagation networks by the hybrid monte carlo method. Technical report, Citeseer.

Neal, R. M. (1996). *Bayesian Learning for Neural Networks*. Springer-Verlag New York, Inc.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Pollard, D. (1990). Empirical processes: Theory and applications. In *NSF-CBMS Regional Conference Series in Probability and Statistics*, pages i–86. JSTOR.

Quinonero-Candela, J., Rasmussen, C. E., Sinz, F., Bousquet, O., and Schölkopf, B. (2005). Evaluating predictive uncertainty challenge. In *Machine Learning Challenges Workshop*, pages 1–27. Springer.

Ranganath, R., Gerrish, S., and Blei, D. (2014). Black box variational inference. In *Artificial intelligence and statistics*, pages 814–822. PMLR.

Robbins, H. E. (1992). An empirical bayes approach to statistics. In *Breakthroughs in statistics*, pages 388–394. Springer.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088):533–536.

Simard, P. Y., Steinkraus, D., Platt, J. C., et al. (2003). Best practices for convolutional neural networks applied to visual document analysis. In *Icdar*, volume 3. Edinburgh.

Springenberg, J. T., Klein, A., Falkner, A., and Hutter, F. (2016). Bayesian optimization with robust bayesian neural networks. In *NeurIPS*.

Sun, S., Chen, C., and Carin, L. (2017). Learning structured weight uncertainty in bayesian neural networks. In *Artificial Intelligence and Statistics*, pages 1283–1292. PMLR.

Tomczak, M., Swaroop, S., Foong, A., and Turner, R. (2021). Collapsed variational bounds for bayesian neural networks. *Advances in Neural Information Processing Systems*, 34:25412–25426.

Vaart, A. W. and Wellner, J. A. (1996). Weak convergence. In *Weak convergence and empirical processes*, pages 16–28. Springer.

Welling, M. and Teh, Y. W. (2011). Bayesian learning via stochastic gradient langevin dynamics. In *ICML*.

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

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

Wong, W. H. and Shen, X. (1995). Probability inequalities for likelihood ratios and convergence rates of sieve mles. *The Annals of Statistics*, pages 339–362.

Worrall, D. E., Wilson, C. M., and Brostow, G. J. (2016). Automated retinopathy of prematurity case detection with convolutional neural networks. In *International Workshop on Deep Learning in Medical Image Analysis*, pages 68–76. Springer.

Zhang, G., Sun, S., Duvenaud, D., and Grosse, R. (2018). Noisy natural gradient as variational inference. In *International Conference on Machine Learning*, pages 5852–5861. PMLR.

Zhou, X., Jiao, Y., Liu, J., and Huang, J. (2022). A deep generative approach to conditional sampling. *Journal of the American Statistical Association*, pages 1–12.