Neural Networks 167 (2023) 309–330

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Layer adaptive node selection in Bayesian neural networks: Statistical guarantees and implementation details

Sanket Jantre [*], Shrijita Bhattacharya, Tapabrata Maiti

Department of Statistics and Probability, Michigan State University, United States of America

ARTICLE INFO

Article history:
Received 8 July 2022
Received in revised form 4 April 2023
Accepted 17 August 2023
Available online 22 August 2023

Keywords:
Node selection
Dynamic pruning
Model compression
Spike-and-slab priors
Variational inference
Contraction rates

ABSTRACT

Sparse deep neural networks have proven to be efficient for predictive model building in large-scale studies. Although several works have studied theoretical and numerical properties of sparse neural architectures, they have primarily focused on the edge selection. Sparsity through edge selection might be intuitively appealing; however, it does not necessarily reduce the structural complexity of a network. Instead pruning excessive nodes leads to a structurally sparse network with significant computational speedup during inference. To this end, we propose a Bayesian sparse solution using spike-and-slab Gaussian priors to allow for automatic node selection during training. The use of spike-and-slab prior alleviates the need of an ad-hoc thresholding rule for pruning. In addition, we adopt a variational Bayes approach to circumvent the computational challenges of traditional Markov Chain Monte Carlo (MCMC) implementation. In the context of node selection, we establish the fundamental result of variational posterior consistency together with the characterization of prior parameters. In contrast to the previous works, our theoretical development relaxes the assumptions of the equal number of nodes and uniform bounds on all network weights, thereby accommodating sparse networks with layer-dependent node structures or coefficient bounds. With a layer-wise characterization of prior inclusion probabilities, we discuss the optimal contraction rates of the variational posterior. We empirically demonstrate that our proposed approach outperforms the edge selection method in computational complexity with similar or better predictive performance. Our experimental evidence further substantiates that our theoretical work facilitates layer-wise optimal node recovery.

1. Introduction

Deep learning profoundly impacts science and society due to its impressive empirical success driven primarily by copious amounts of datasets, ever increasing computational resources, and deep neural network's (DNN) ability to learn task-specific representations. The key characteristic of deep learning is that accuracy empirically scales with the size of the model and the amount of training data. As such, large neural network models such as OpenAI GPT-3 (175 Billion) now typify the state-of-the-art across multiple domains such as natural language processing, computer vision, speech recognition etc. Nevertheless deep neural networks do have some drawbacks despite their wide ranging applications. First, this form of model scaling is exorbitantly prohibitive in terms of computational requirements, financial commitment, energy requirements etc. Second, DNNs tend to overfit leading to poor generalization in practice (Zhang et al., 2017). Finally, there are numerous scenarios where training and deploying such huge models is practically infeasible. Examples of such scenarios include federated learning, autonomous vehicles, robotics, recommendation systems where models have to be refreshed daily/hourly or in an online manner for optimal performance.

A promising direction for addressing these issues while improving the efficiency of DNNs is exploiting sparsity. From a practical perspective, it has been well-known that neural networks can be sparsified without significant loss in performance, Mozer and Smolensky (1988), and there is growing evidence that it is more so in the case of modern DNNs. Sparsity can arise naturally or be induced in multiple forms in DNNs, including input data, weights, and nodes. Weight pruning approaches perform high model compression leading to significant storage cost reduction at test-time (Frankle & Carbin, 2019; Han et al., 2016; Molchanov et al., 2017; Zhu & Gupta, 2018). However, they result in unstructured sparsity in deep neural architectures which leads to inefficient computational gains in practical setups (Wen et al., 2016). Instead, inducing group sparsity on collection of incoming weights into a given node (or node selection) reduces the dimensions of weight matrices per layer allowing for significant computational savings. To that effect, edge selection and node selection approaches are complementary with the former leading to

[*] Corresponding author.
E-mail addresses: jantresanket@gmail.com (S. Jantre), bhatta61@msu.edu (S. Bhattacharya), maiti@msu.edu (T. Maiti).

https://doi.org/10.1016/j.neunet.2023.08.029
0893-6080/© 2023 Elsevier Ltd. All rights reserved.

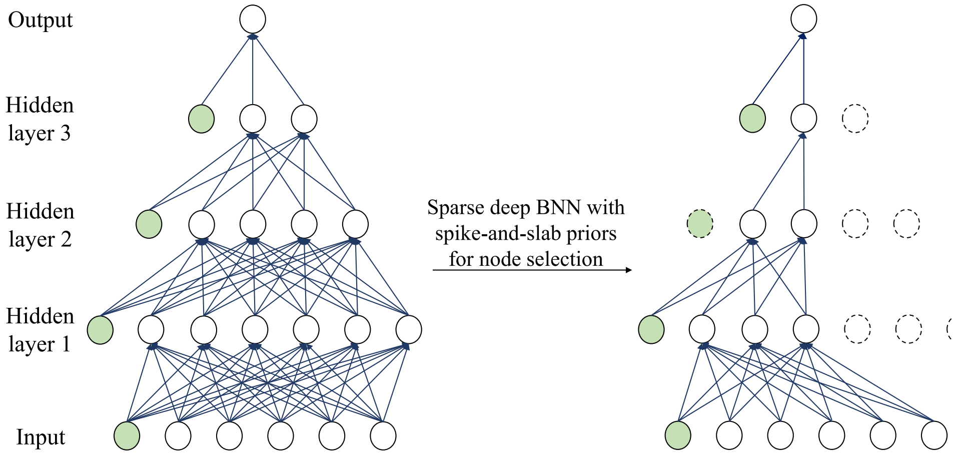

Output
Hidden layer 3
Hidden layer 2
Hidden layer 1
Input
Sparse deep BNN with spike-and-slab priors for node selection

Fig. 1. Sparse deep BNN using spike-and-slab priors achieves node selection in the given dense network on left leading to a sparse network on right.

storage reduction and the later leading to computational speedup during inference stage. Although one may argue node selection arises as a byproduct of edge selection, we clearly demonstrate that an approach which targets node selection directly leads to lower latency models (smaller number of nodes per layer) compared to an approach which achieves node selection through edge selection.

Node selection in deep neural networks has been explored under frequentist setting in Alvarez and Salzmann (2016), Scardapane et al. (2017), Wen et al. (2016) using group sparsity regularizers. On the other hand, Louizos et al. (2017), Neklyudov et al. (2017), and Ghosh et al. (2019) incorporate group sparsity via shrinkage priors in Bayesian paradigm. These group sparsity approaches specifically applied for node selection have shown significant computational speedup and lower memory footprint at inference stage. However, all of the proposed methods of neuron selection perform ad-hoc pruning requiring fine-tuned thresholding rules. Moreover, the posterior inference of network weights in Bayesian neural networks (BNN) through standard MCMC method, ex. Hamiltonian Monte Carlo (Neal, 1992), does not scale well to modern neural network architectures and large datasets used in practice. Instead computationally efficient variational inference as an alternative to MCMC (Blei et al., 2017; Jordan et al., 1999), has been explored in the context of edge selection both theoretically and numerically by Bai et al. (2020), Blundell et al. (2015), Chérief-Abdellatif (2020). On the other hand, Louizos et al. (2017) and Ghosh et al. (2019) have explored variational inference for node selection problem. In this work, we propose a Gaussian spike-and-slab prior for automatic node selection in Bayesian neural networks thereby alleviating the need of an ad-hoc thresholding rule for pruning (see a schematic Fig. 1). Further for scalability, we develop a variational Bayes algorithm for posterior inference of BNN model parameters in our proposed model and demonstrate its numerical performance through simulation and real regression and classification datasets. Finally, we provide the theoretical guarantees to our node selection method under mild restrictions on the network topology.

Related Work. A closely related work to our paper is Bai et al. (2020)'s automated edge selection model using spike-and-slab prior. There the slab distribution controls the magnitude of weights and spike allows for the exact setting of weights to 0. We introduce spike-and-slab framework for node selection in BNNs and show the key resource efficiency trade-off between node and edge selection at test-time. There are two main advantages to node selection over edge selection (1) fewer parameters to train during optimization, (2) results in structurally compact network leading to computational speedup at test-time.

On the theoretical front, sparse BNNs have been studied in the works of Polson and Ročková (2018) and Sun et al. (2021). In the context of variational inference, sparse BNNs have been studied in the recent works of Chérief-Abdellatif (2020) and Bai et al. (2020). All these works concentrate on the problem of edge selection facilitated through the use of Gaussian spike-and-slab priors. In the context of node selection, Ghosh et al. (2019) makes use of regularized horseshoe prior. The main limitations of their approach include (1) need for fine tuning of the thresholding rule for node selection, and (2) lack of a theoretical justification.

The only two works which have provided theoretical guarantees of their proposed sparse DNN methods under variational inference include those of Chérief-Abdellatif (2020) and Bai et al. (2020). Since they focus on the problem of edge selection, their theoretical developments are related to the results of Schmidt-Hieber (2020) (see the sieve construction in relation (4) in Schmidt-Hieber (2020)) and not directly extendable to our setup. Additionally, they assume certain restrictions on the network topology like (i) equal number of nodes in each layer, (ii) a known uniform bound B on all network weights, and (iii) a global sparsity parameter which may not lead to a structurally compact network. Although from a numerical standpoint, one may implicitly extend the problem of edge selection to node selection, the theoretical guarantees of node selection consistency in sparse DNNs is not immediate.

Detailed Contributions.

1. We propose a Gaussian spike-and-slab node selection model and develop a variational Bayes approach for posterior inference of the model parameters. We call our approach **SS-IG** (**S**pike-and-**S**lab **I**ndependent **G**aussian) model.
2. We derive the variational consistency using a functional space of neural networks which takes two layer dependent bounds, one which upper bounds the number of neurons in each layer and the other which upper bounds the L_1 norm of the weights incident onto each node of a layer. These layer dependent bounds allow the generalization of the theoretical results presented to guarantee the consistency of any generic shaped network structure. Further, it also guides the calculation of layer-wise prior inclusion probabilities which allow for optimal node recovery per layer in the computational experiments.
3. We measure the computational gains achieved by our approach using layer-wise node sparsities for shallow models and floating point operations in larger models. Our numerical results validate the proposed theoretical framework for

the node selection in DNN models. These empirical experiments further justify the use of layer-wise node inclusion probabilities to facilitate the optimal node recovery.

2. Nonparametric regression: deep learning approach

Consider the nonparametric regression model with p dimensional covariate $\boldsymbol{X}$.

$$Y_i = \eta_0(\boldsymbol{X}_i) + e_i, \quad i = 1, \dots, n, \tag{1}$$

where $e_i \overset{\text{i.i.d.}}{\sim} N(0, \sigma_e^2)$ (here i.i.d. denotes independent and identically distributed) and $\eta_0(\cdot) : \mathbb{R}^p \to \mathbb{R}$.

Thus, the conditional distribution of $Y|\boldsymbol{X} = \boldsymbol{x}$ under the true model is

$$f_0(y|\boldsymbol{x}) = (\sqrt{2\pi\sigma_e^2})^{-1} \exp\left(-(y - \eta_0(\boldsymbol{x}))^2/(2\sigma_e^2)\right) \tag{2}$$

where $\boldsymbol{x}$ is a feature vector from a marginal distribution $P_{\boldsymbol{X}}$ and y is the corresponding output from the conditional distribution $Y|\boldsymbol{X} = \boldsymbol{x}$.

Let $g : \mathbb{R}^p \to \mathbb{R}$ be a measurable function, then for some loss function $\mathcal{L}$, the risk of g is

$$R(g) = \int_{\mathcal{Y}\times\mathcal{X}} \mathcal{L}(Y, g(\boldsymbol{X}))dP_{\boldsymbol{X},Y}$$

where $P_{\boldsymbol{X},Y}$, the joint distribution of $(\boldsymbol{X}, Y)$ is product of $P_{\boldsymbol{X}}$ and the conditional distribution $Y|X = \boldsymbol{x}$. (see Cannings and Samworth (2017) for more details). For the squared error loss, the above risk is minimized by $g^*(\boldsymbol{x}) = \eta_0(\boldsymbol{x})$ (Friedman et al., 2009). In practice, this estimator is not useful since $\eta_0(\boldsymbol{x})$ is unknown. Thus, an estimator of $\eta_0(\boldsymbol{x})$ is obtained based on the training observations, $\mathcal{D} = \{(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_n, y_n)\}$, drawn from $P_{\boldsymbol{X},Y}$. To find the class of optimal estimators, we use DNNs as an approximation to $\eta_0(\boldsymbol{x})$.

For a $p \times 1$ input vector $\boldsymbol{x}$, consider a DNN with L hidden layers with $k_1, \dots, k_L$ as the number of nodes in the hidden layers denoted by $\eta_{\boldsymbol{\theta}}(\boldsymbol{x})$. Also,

$$\eta_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{v}_L + \boldsymbol{W}_L\psi(\boldsymbol{v}_{L-1} + \boldsymbol{W}_{L-1}\psi(\cdots\psi(\boldsymbol{v}_1 + \boldsymbol{W}_1\psi(\boldsymbol{v}_0 + \boldsymbol{W}_0\boldsymbol{x})))) \tag{3}$$

where $\boldsymbol{v}_l$ and $\boldsymbol{W}_l$, $l = 0, \dots, L$ are $k_{l+1} \times 1$ vectors and $k_{l+1} \times k_l$ matrices, respectively and ψ is the activation function. Let $\boldsymbol{\theta} = \{\overline{\boldsymbol{W}}_0, \dots, \overline{\boldsymbol{W}}_L\}$ denote all the parameters in the DNN model under consideration. Using the DNN in (3) to approximate the true function $\eta_0(\boldsymbol{x})$, the conditional distribution of $Y|\boldsymbol{X} = \boldsymbol{x}$ is

$$f_{\boldsymbol{\theta}}(y|\boldsymbol{x}) = (\sqrt{2\pi\sigma_e^2})^{-1} \exp\left(-(y - \eta_{\boldsymbol{\theta}}(\boldsymbol{x}))^2/(2\sigma_e^2)\right)$$

Thus, the likelihood function for the data $\mathcal{D}$ under the model and the truth is

$$P_{\boldsymbol{\theta}}^n = \prod_{i=1}^n f_{\boldsymbol{\theta}}(y_i|\boldsymbol{x}_i), \qquad P_0^n = \prod_{i=1}^n f_0(y_i|\boldsymbol{x}_i). \tag{4}$$

For theoretical development in the subsequent sections we shall assume $P_{\boldsymbol{X}} = U[0, 1]^p$ and $\sigma_e^2 = 1$ and ψ is any 1-Lipschitz continuous activation function.

3. Node selection with spike-and-slab prior

To allow for automatic node selection, we consider a spike-and-slab prior consisting of a Dirac spike (δ_0) at 0 and a slab distribution (Mitchell & Beauchamp, 1988). The spike part is represented by an indicator variable which is set to 0 if a node is not present in the network. The slab part comes from a Gaussian distributed random variable. To allow for the layer-wise node selection, we assume that the prior inclusion probability λ_l varies as a function of the layer index l. The symbol i.d. is used to denote independently distributed random variables.

Prior: We assume a spike-and-slab prior of the following form with z_{lj} as the indicator for the presence of jth node in the lth layer

$$\overline{\boldsymbol{w}}_{lj}|z_{lj} \overset{\text{i.d.}}{\sim} \left[(1 - z_{lj})\boldsymbol{\delta}_0 + z_{lj}N(0, \sigma_0^2\boldsymbol{I})\right], \quad z_{lj} \overset{\text{i.d.}}{\sim} \text{Ber}(\lambda_l)$$

where $l = 0, \dots, L$, $j = 1, \dots, k_{l+1}$. Also, $\overline{\boldsymbol{w}}_{lj} = (\overline{w}_{lj1}, \dots, \overline{w}_{ljk_l+1})$ is a vector of edges incident on the jth node in the lth layer. In the above formula, note $\boldsymbol{\delta}_0$ is a Dirac spike vector of dimension $k_l + 1$ with all entries zero and $\boldsymbol{I}$ is the identity matrix of dimension $k_l + 1 \times k_l + 1$. Furthermore, z_{lj} with $j = (1, \dots, k_{l+1})$ all follow Bernoulli(λ_l) to allow for common prior inclusion probability, λ_l, for each node from a given layer l. We set $\lambda_L = 1$ to ensure no node selection occurs in the output layer.

Posterior: With $\boldsymbol{z}_l = (z_{l1}, \dots, z_{lk_{l+1}})$, let $\boldsymbol{z} = (\boldsymbol{z}_1, \dots, \boldsymbol{z}_L)$ denote the vector of all indicator variables. The posterior distribution of $(\boldsymbol{\theta}, \boldsymbol{z})$ given $\mathcal{D}$ is given by

$$\pi(\boldsymbol{\theta}, \boldsymbol{z}|\mathcal{D}) = \frac{P_{\boldsymbol{\theta}}^n\pi(\boldsymbol{\theta}|\boldsymbol{z})\pi(\boldsymbol{z})}{\sum_{\boldsymbol{z}}\int P_{\boldsymbol{\theta}}^n\pi(\boldsymbol{\theta}|\boldsymbol{z})\pi(\boldsymbol{z})d\boldsymbol{\theta}} = \frac{P_{\boldsymbol{\theta}}^n\pi(\boldsymbol{\theta}|\boldsymbol{z})\pi(\boldsymbol{z})}{m(\mathcal{D})} \tag{5}$$

where $P_{\boldsymbol{\theta}}^n = \prod_{i=1}^n f_{\boldsymbol{\theta}}(y_i|\boldsymbol{x}_i)$ is the likelihood function as in (4), $\pi(\boldsymbol{z})$ is the probability mass function of $\boldsymbol{z}$ with respect to the counting measure and $\pi(\boldsymbol{\theta}|\boldsymbol{z})$ is the conditional probability density function with respect to the Lebesgue measure of $\boldsymbol{\theta}$ given $\boldsymbol{z}$. Further, $m(\mathcal{D})$ is the marginal density of the data and is free of $(\boldsymbol{\theta}, \boldsymbol{z})$.

Let $\widetilde{\pi}(\boldsymbol{\theta}) = \sum_{\boldsymbol{z}}\pi(\boldsymbol{\theta}, \boldsymbol{z})$ be the marginal prior of $\boldsymbol{\theta}$. We shall use the notation

$$\widetilde{\Pi}(\mathcal{A}) = \int_{\mathcal{A}} \widetilde{\pi}(\boldsymbol{\theta})d\boldsymbol{\theta} \tag{6}$$

to denote the probability distribution function corresponding to the density function $\widetilde{\pi}$. The marginal posterior of $\boldsymbol{\theta}$ expressed as a function of the marginal prior for $\boldsymbol{\theta}$ is

$$\widetilde{\pi}(\boldsymbol{\theta}|\mathcal{D}) = \sum_{\boldsymbol{z}}\pi(\boldsymbol{\theta}, \boldsymbol{z}|\mathcal{D}) = \frac{P_{\boldsymbol{\theta}}^n\widetilde{\pi}(\boldsymbol{\theta})}{\int P_{\boldsymbol{\theta}}^n\widetilde{\pi}(\boldsymbol{\theta})d\boldsymbol{\theta}} = \frac{P_{\boldsymbol{\theta}}^n\widetilde{\pi}(\boldsymbol{\theta})}{m(\mathcal{D})}$$

Thus, the probability distribution function corresponding to the density function $\widetilde{\pi}(|\mathcal{D})$ is then given by

$$\widetilde{\Pi}(\mathcal{A}|\mathcal{D}) = \int_{\mathcal{A}} \widetilde{\pi}(\boldsymbol{\theta}|\mathcal{D})d\boldsymbol{\theta} \tag{7}$$

Variational family: We posit the following mean field variational family $(\mathcal{Q}^{\mathbf{MF}})$ on network weights as

$$\mathcal{Q}^{\mathbf{MF}} = \Big\{ \overline{\boldsymbol{w}}_{lj}|z_{lj} \overset{\text{i.d.}}{\sim} \left[(1 - z_{lj})\boldsymbol{\delta}_0 + z_{lj}N(\boldsymbol{\mu}_{lj}, \text{diag}(\boldsymbol{\sigma}_{lj}^2))\right], \\ z_{lj} \overset{\text{i.d.}}{\sim} \text{Ber}(\gamma_{lj}) \Big\}$$

for $l = 0, \dots, L$, $j = 1, \dots, k_{l+1}$. This ensures that weight distributions follow spike-and-slab structure which allows for node sparsity through variational approximation. Further, the weight distributions conditioned on the node indicator variables are all independent of each other (hence use of the term mean field family). The variational distribution of parameters obtained post optimization will then inherently prune away redundant nodes from each layer. Also, Gaussian distribution for slab component is widely popular for approximating neural network weight distributions (Bai et al., 2020; Blundell et al., 2015; Louizos et al., 2017).

Additionally, $\boldsymbol{\mu}_{lj} = (\mu_{lj1}, \dots, \mu_{ljk_l+1})$ and $\boldsymbol{\sigma}_{lj}^2 = (\sigma_{lj1}^2, \dots, \sigma_{ljk_l+1}^2)$ denote the vectors of variational mean and standard deviation parameters of the edges incident on the jth node in the lth layer. Similarly, γ_{lj} denotes the variational inclusion probability of the jth node in the lth layer. We set $\gamma_{Lj} = 1$ to ensure no node selection occurs in the output layer.

Variational posterior: Variational posterior aims to reduce the Kullback–Leibler (KL) distance between a variational family and the true posterior (Blei & Lafferty, 2007; Hinton & Van Camp, 1993) as

$$\pi^* = \underset{q \in \mathcal{Q}^{\mathbf{MF}}}{\operatorname{argmin}} \; d_{\mathrm{KL}}(q, \pi(|\mathcal{D})) \tag{8}$$

where $d_{\mathrm{KL}}(q, \pi(|\mathcal{D}))$ denotes the KL-distance between q and $\pi(|\mathcal{D})$.

Note, the variational member q can be written as $q(\boldsymbol{\theta}, \boldsymbol{z}) = q(\boldsymbol{\theta}|\boldsymbol{z})q(\boldsymbol{z})$ where $q(\boldsymbol{z})$ is the probability mass function of $\boldsymbol{z}$ with respect to the counting measure and $q(\boldsymbol{\theta}|\boldsymbol{z})$ is the conditional density function given with respect to the Lebesgue measure of $\boldsymbol{\theta}$ given $\boldsymbol{z}$. Further,

$$\begin{aligned}
\pi^* &= \underset{q \in \mathcal{Q}^{\mathbf{MF}}}{\operatorname{argmin}} \sum_{\boldsymbol{z}} \int [\log q(\boldsymbol{\theta}, \boldsymbol{z}) - \log \pi(\boldsymbol{\theta}, \boldsymbol{z}|\mathcal{D})] q(\boldsymbol{\theta}, \boldsymbol{z}) d\boldsymbol{\theta} \\
&= \underset{q \in \mathcal{Q}^{\mathbf{MF}}}{\operatorname{argmin}} \left(\sum_{\boldsymbol{z}} \int [\log q(\boldsymbol{\theta}, \boldsymbol{z}) - \log \pi(\boldsymbol{\theta}, \boldsymbol{z}, \mathcal{D})] q(\boldsymbol{\theta}, \boldsymbol{z}) d\boldsymbol{\theta} \right. \\
&\quad \left. + \log m(\mathcal{D}) \right) \\
&= \underset{q \in \mathcal{Q}^{\mathbf{MF}}}{\operatorname{argmin}} \; [-\mathrm{ELBO}(q, \pi(|\mathcal{D}))] + \log m(\mathcal{D}) \\
&= \underset{q \in \mathcal{Q}^{\mathbf{MF}}}{\operatorname{argmax}} \; \mathrm{ELBO}(q, \pi(|\mathcal{D}))
\end{aligned} \tag{9}$$

Since $\log m(\mathcal{D})$ is free from q, it suffices to maximize the evidence lower bound (ELBO) above.

Let $\widetilde{\pi}^*(\boldsymbol{\theta}) = \sum_{\boldsymbol{z}} \pi^*(\boldsymbol{\theta}|\boldsymbol{z})\pi^*(\boldsymbol{z})$ then $\widetilde{\pi}^*$ denotes the marginal variational posterior for $\boldsymbol{\theta}$. We shall use the notation

$$\widetilde{\Pi}^*(\mathcal{A}) = \int_{\mathcal{A}} \widetilde{\pi}^*(\boldsymbol{\theta}) d\boldsymbol{\theta} \tag{10}$$

to denote the probability distribution function corresponding to the density function $\widetilde{\pi}^*$.

4. Posterior contraction rates

In this section, we develop the theoretical consistency of the variational posterior in (10) in context of node selection. Previous works which establish the statistical consistency of sparse deep neural networks do so only in the context of edge selection. Thereby, the works of Polson and Ročková (2018), Chérief-Abdellatif (2020) and Bai et al. (2020) use several results from the pioneer work of Schmidt-Hieber (2020). In addition to node selection consistency, we also relax certain network restrictions considered in the previous works. These restrictions include (1) equal number of nodes in each layer which restricts one from using any previous information on the number of nodes in the deep neural architecture (2) a known bound B on all the neural network weights as they essentially rely on the sieve construction in equation 3 of Schmidt-Hieber (2020) which assumes that L_∞ norm of all $\boldsymbol{\theta}$ entries is smaller than 1 (3) a global sparsity parameter s which does not always consider structurally sparse networks.

Towards the proof, firstly our sieve construction allows the number of nodes of the neural network to vary as a function of the layer. Secondly, instead of global sparsity parameter s (see the sieve construction in relation (4) of Schmidt-Hieber (2020)) we allow for layer wise sparsity vector $\boldsymbol{s}$ to account for the number of nodes in each layer. Finally, we relax the assumption of a known bound B by considering a sieve with a layer wise constraint (denoted by the vector $\boldsymbol{B}$) on the L_1 norm of the incoming edges of a node. Thus, our work extends on current literature along three directions: (1) theoretically quantifies predictive performance of Bayesian neural networks with node based pruning; (2) establishes that even without a fixed bound on network weights, one can recover the true solution by appropriate choice of the prior; (3) provides layer wise node inclusion probabilities to allow for structurally sparse solutions. The relaxation of these network structure assumptions requires us to provide the framework for node selection including appropriate sieve construction together with the derivation of the results in Schmidt-Hieber (2020) customized to our problem.

To establish the posterior contraction rates, we show that the variational posterior in (8) concentrates in shrinking Hellinger neighborhoods of the true density function P_0 with overwhelming probability. Since $\boldsymbol{X} \sim U[0, 1]^p$, thus $f_0(\boldsymbol{x}) = f_{\boldsymbol{\theta}}(\boldsymbol{x}) = 1$. This further implies $P_0 = f_0(y|\boldsymbol{x})f_0(\boldsymbol{x}) = f_0(y|\boldsymbol{x})$ and similarly $P_{\boldsymbol{\theta}} = f_{\boldsymbol{\theta}}(y|\boldsymbol{x})$. We next define the Hellinger neighborhood of the true density P_0 as

$$\mathcal{H}_\varepsilon = \{\boldsymbol{\theta} : d_{\mathrm{H}}(P_0, P_{\boldsymbol{\theta}}) < \varepsilon\}$$

where the Hellinger distance between the true density function P_0 and the model density $P_{\boldsymbol{\theta}}$ is

$$d_{\mathrm{H}}^2(P_0, P_{\boldsymbol{\theta}}) = \frac{1}{2} \int \left(\sqrt{f_{\boldsymbol{\theta}}(y|\boldsymbol{x})} - \sqrt{f_0(y|\boldsymbol{x})} \right)^2 dy d\boldsymbol{x}$$

We also define the KL neighborhood of the true density P_0 as

$$\mathcal{N}_\varepsilon = \{\boldsymbol{\theta} : d_{\mathrm{KL}}(P_0, P_{\boldsymbol{\theta}}) < \varepsilon\}$$

where the KL distance d_{KL} between the true density function P_0 and the model density $P_{\boldsymbol{\theta}}$ is

$$d_{\mathrm{KL}}(P_0, P_{\boldsymbol{\theta}}) = \int \log \frac{f_0(y|\boldsymbol{x})}{f_{\boldsymbol{\theta}}(y|\boldsymbol{x})} f_0(y|\boldsymbol{x}) dy d\boldsymbol{x}$$

Let $\boldsymbol{k} = (k_0, \ldots, k_{L+1})$ be the node vector, $\overline{\boldsymbol{W}}_l = (\boldsymbol{w}_{l1}^\top, \ldots, \boldsymbol{w}_{lk_{l+1}}^\top)^\top$ be the row representation of $\overline{\boldsymbol{W}}_l$ and $\widetilde{\boldsymbol{w}}_l = (\|\boldsymbol{w}_{l1}\|_1, \ldots, \|\boldsymbol{w}_{lk_{l+1}}\|_1)$ be the vector of L_1 norms of the rows of $\overline{\boldsymbol{W}}_l$. Next we consider layer-wise sparsity, $\boldsymbol{s} = (s_1, \ldots, s_L)$ for node selection. Similarly, we consider layer-wise norm constraints, $\boldsymbol{B} = (B_1, \ldots, B_L)$ on L_1 norms of weights including bias incident onto any given node in each layer. Based on $\boldsymbol{s}$ and $\boldsymbol{B}$, we define the following sieve of neural networks (check definition A.1).

$$\mathcal{F}(L, \boldsymbol{k}, \boldsymbol{s}, \boldsymbol{B}) = \{\eta_{\boldsymbol{\theta}} \in (3) : \|\widetilde{\boldsymbol{w}}_l\|_0 \leq s_l, \|\widetilde{\boldsymbol{w}}_l\|_\infty \leq B_l\}\,. \tag{11}$$

The construction of a sieve is one of the most important tools towards the proof of consistency in infinite-dimensional spaces. In the works of Schmidt-Hieber (2020), Polson and Ročková (2018), Chérief-Abdellatif (2020) and Bai et al. (2020), the sieve in the context of edge selection is given by

$$\mathcal{F}(L, \boldsymbol{k}, s) = \{\eta_{\boldsymbol{\theta}} \in (3) : \|\boldsymbol{\theta}\|_0 \leq s, \|\boldsymbol{\theta}\|_\infty \leq 1\}\,.$$

which works with an overall sparsity parameter s. In addition, note the L_∞ norm of all the entries in $\boldsymbol{\theta}$ is assumed to be known constant equal to 1 (see relation (4) in Schmidt-Hieber (2020) and section 4 in Polson and Ročková (2018)). Section 3 in Bai et al. (2020) does not explicitly mention the dependence of their sieve on some fixed bound B on the edges in a network, however, their derivations on covering numbers (see proof of Lemma 1.2 in the supplement of Bai et al. (2020)) borrow results from (Schmidt-Hieber, 2020) which is based on sieve with $B = 1$.

Consider any sequence ϵ_n. For Lemmas 4.1 and 4.2, we work with the sieve $\mathcal{F}(L, \boldsymbol{k}, \boldsymbol{s}, \boldsymbol{B})$ in (11) with $\boldsymbol{s} = \boldsymbol{s}^\circ$ and $\boldsymbol{B} = \boldsymbol{B}^\circ$ where $s_l^\circ + 1 = n\epsilon_n^2/(\sum_{j=0}^L u_j)$ and $\log B_l^\circ = (n\epsilon_n^2)/((L+1)\sum_{j=0}^L (s_j^\circ + 1))$ with $u_l = (L+1)^2(\log n + \log(L+1) + \log k_{l+1} + \log(k_l + 1))$. Note, s_l° and B_l° do not depend on l.

Lemma 4.1 below holds when the covering number (check definition A.2) of the functions which belong to the sieve $\mathcal{F}(L, \boldsymbol{k}, \boldsymbol{s}^\circ, \boldsymbol{B}^\circ)$ is well under control. Lemma 4.2 below states that for the same choice of the sieve, the prior gives sufficiently small probabilities on the complement space $\mathcal{F}(L, \boldsymbol{k}, \boldsymbol{s}^\circ, \boldsymbol{B}^\circ)^c$ (see the discussion under Theorem 4.4 for more details).

For the subsequent results, the symbol $\mathcal{A}^c$ will be used to denote complement of a set $\mathcal{A}$.

Lemma 4.1 (*Existence of Test Functions*). *Let $\epsilon_n \to 0$ and $n\epsilon_n^2 \to \infty$. There exists a testing function $\phi \in [0, 1]$ and constants $C_1, C_2 > 0$,*

$$\mathbb{E}_{P_0}(\phi) \le \exp\{-C_1 n\epsilon_n^2\}$$

$$\sup_{\boldsymbol{\theta} \in \mathcal{H}^c_{\epsilon_n}, \eta_{\boldsymbol{\theta}} \in \mathcal{F}(L, \boldsymbol{k}, \boldsymbol{s}^\circ, \boldsymbol{B}^\circ)} \mathbb{E}_{P_{\boldsymbol{\theta}}}(1-\phi) \le \exp\{-C_2 n d_{\mathrm{H}}^2(P_0, P_{\boldsymbol{\theta}})\}$$

where $\mathcal{H}_{\epsilon_n} = \{\boldsymbol{\theta} : d_{\mathrm{H}}(P_0, P_{\boldsymbol{\theta}}) \le \epsilon_n\}$ is the Hellinger neighborhood of radius ϵ_n.

Lemma 4.2 (*Prior mass condition.*). *Let $\epsilon_n \to 0$, $n\epsilon_n^2 \to \infty$ and $n\epsilon_n^2 / \sum_{l=0}^L u_l \to \infty$, then for $\widetilde{\Pi}$ as in (6) and some constant $C_3 > 0$,*

$$\widetilde{\Pi}(\mathcal{F}(L, \boldsymbol{k}, \boldsymbol{s}^\circ, \boldsymbol{B}^\circ)^c) \le \exp(-C_3 n\epsilon_n^2 / \sum_{l=0}^{L} u_l)$$

Whereas Lemmas 4.1 and 4.2 work with a specific choice of the sieve, the following Lemma 4.3 is developed for any generic choice of sieve indexed by $\boldsymbol{s}$ and $\boldsymbol{B}$. The final piece of the theory developed next tries to addresses two main questions (1) Can we get a sparse network solution whose layer-wise sparsity levels and L_1 norms of incident edges (including the bias) of the nodes are controlled at levels $\boldsymbol{s}$ and $\boldsymbol{B}$ respectively? (2) Does this sparse network retain the same predictive performance as the original network?

In this direction, let

$\xi = \min_{\eta_{\boldsymbol{\theta}} \in \mathcal{F}(L, \boldsymbol{k}, \boldsymbol{s}, \boldsymbol{B})} \|\eta_{\boldsymbol{\theta}} - \eta_0\|_\infty^2$

Based on the values $\boldsymbol{s}$ and $\boldsymbol{B}$, we also define

$$\vartheta_l = {B_l}^2/(k_l+1) + \sum_{m=0, m\neq l}^{L} \log B_m + L + \log k_{l+1}$$
$$+ \log(k_l+1) + \log n + \log(\sum_{m=0}^{L} u_m)$$
$$r_l = s_l(k_l+1)\vartheta_l/n \tag{12}$$

Lemma 4.3 has two sub conditions. Condition 1. requires that shrinking KL neighborhood of the true density function P_0 gets sufficiently large probability. This along with Lemmas 4.1 and 4.2 is an essential condition to guarantee the convergence of the true posterior in (5). Condition 2. is the assumption needed to control the KL distance between true posterior and variational posterior and thereby guarantees the convergence of the variational posterior in (8) (see the discussion under Theorem 4.4 for more details).

Lemma 4.3 (*Kullback–Leibler Conditions*). *Suppose $\sum_{l=0}^L r_l + \xi \to 0$ and $n(\sum_{l=0}^L r_l + \xi) \to \infty$ and the following two conditions hold for the prior $\widetilde{\Pi}$ in (6) and some $q \in \mathcal{Q}^{\mathbf{MF}}$*

1. $$\widetilde{\Pi}\left(\mathcal{N}_{\sum_{l=0}^L r_l + \xi}\right) \ge \exp(-C_4 n(\sum_{l=0}^{L} r_l + \xi))$$

2. $$d_{\mathrm{KL}}(q, \pi) + n\sum_{\boldsymbol{z}} \int d_{\mathrm{KL}}(P_0, P_{\boldsymbol{\theta}}) q(\boldsymbol{\theta}, \boldsymbol{z}) d\boldsymbol{\theta} \le C_5 n(\sum_{l=0}^{L} r_l + \xi)$$

where π is the joint prior of $(\boldsymbol{\theta}, \boldsymbol{z})$, q is the joint variational distribution of $(\boldsymbol{\theta}, \boldsymbol{z})$ and $\mathcal{N}_{\sum_{l=0}^L r_l + \xi}$ is the KL neighborhood of radius $\sum_{l=0}^L r_l + \xi$.

The following result shows that the variational posterior is consistent as long as Lemma 4.1, Lemmas 4.2 and 4.3 hold. The proof of Theorem 4.4 demonstrates how the validity of these three lemmas imply variational posterior consistency.

Theorem 4.4. *Suppose Lemma 4.3 holds and Lemmas 4.1 and 4.2 hold for $\epsilon_n = \sqrt{(\sum_{l=0}^L r_l + \xi)\sum_{l=0}^L u_l}$. Then for some slowly increasing sequence $M_n \to \infty$, $M_n \epsilon_n \to 0$ and $\widetilde{\Pi}^*$ as in (10),*

$$\widetilde{\Pi}^*(\mathcal{H}^c_{M_n\epsilon_n}) \to 0, \quad n \to \infty$$

in P_0^n probability where $\mathcal{H}^c_{M_n\epsilon_n} = \{\boldsymbol{\theta} : d_{\mathrm{H}}(P_0, P_{\boldsymbol{\theta}}) \le M_n\epsilon_n\}$ is the Hellinger neighborhood of radius $M_n\epsilon_n$.

Note, the above contraction rate depends mainly on two quantities r_l and ξ. Note r_l controls the number of nodes in the neural network. If the network is not sparse, then r_l is $k_{l+1}(k_l+1)\vartheta_l/n$ instead of $s_l(k_l+1)\vartheta_l/n$ which can in turn make the convergence of $\epsilon_n \to 0$ difficult. On the other hand, if s_l and B_l are too small, it will cause ξ to explode since a good approximation to the true function may not exist in a very sparse space.

Remark (*Rates as a Function of* n). Let $L \sim O(\log n)$, $B_l^2 \sim O(k_l+1)$ and $s_l(k_l+1) = O(n^{1-2\varrho})$, for some $\varrho > 0$, then one can work with $\epsilon_n = n^{-\varrho}\log^3(n)$ as long as $\xi = O(n^{-2\varrho}\log^2(n))$. The exact expression of ϱ is determined by the degree of smoothness of the function η_0.

Proof of Theorem 4.4.

Discussion. To further enunciate Lemmas 4.1 and 4.2 consider the quantity $\mathcal{E}_{1n} = \int_{\mathcal{H}^c_{M_n\epsilon_n}} (P^n_{\boldsymbol{\theta}}/P^n_0)\widetilde{\pi}(\boldsymbol{\theta})d\boldsymbol{\theta}$ as used in the following proof. Here, $\mathcal{E}_{1n}$ can be split into two parts

$$\mathcal{E}_{1n} = \int_{\mathcal{H}^c_{M_n\epsilon_n} \cap \mathcal{F}(L, \boldsymbol{k}, \boldsymbol{s}^\circ, \boldsymbol{B}^\circ)} (P^n_{\boldsymbol{\theta}}/P^n_0)\widetilde{\pi}(\boldsymbol{\theta})d\boldsymbol{\theta}$$
$$+ \int_{\mathcal{H}^c_{M_n\epsilon_n} \cap \mathcal{F}(L, \boldsymbol{k}, \boldsymbol{s}^\circ, \boldsymbol{B}^\circ)^c} (P^n_{\boldsymbol{\theta}}/P^n_0)\widetilde{\pi}(\boldsymbol{\theta})d\boldsymbol{\theta}$$

Whereas Lemma 4.1 provides a handle on the first term by controlling the covering number of the sieve $\mathcal{F}(L, \boldsymbol{k}, \boldsymbol{s}^\circ, \boldsymbol{B}^\circ)$, Lemma 4.2 gives a handle on the second term by controlling $\widetilde{\Pi}(\mathcal{F}(L, \boldsymbol{k}, \boldsymbol{s}^\circ, \boldsymbol{B}^\circ)^c)$ (for more details we refer to Lemma A.8 in Appendix A).

Next, consider the quantity $\mathcal{E}_{2n} = \log\int(P^n_{\boldsymbol{\theta}}/P^n_0)\widetilde{\pi}(\boldsymbol{\theta})d\boldsymbol{\theta}$ in the following proof. Lemma 4.3 part 1. provides a control on this term (see Lemma A.9 in Appendix A for more details). Finally, consider the quantity $\mathcal{E}_{3n} = d_{\mathrm{KL}}(q, \pi) + \sum_{\boldsymbol{z}}\int \log(P^n_0/P^n_{\boldsymbol{\theta}})q(\boldsymbol{\theta}, \boldsymbol{z})d\boldsymbol{\theta}$ in the following proof. Indeed Lemma 4.3 part 2. provides a control on this term (see Lemma A.10 in Appendix A for further details).

Proof. Let $\widetilde{\Pi}$ and $\widetilde{\Pi}^*$ be as in (7) and (10) respectively. Now,

$$d_{\mathrm{KL}}(\widetilde{\pi}^*, \widetilde{\pi}(|\mathcal{D})) = \int_{\mathcal{A}} \widetilde{\pi}^*(\boldsymbol{\theta}) \log \frac{\widetilde{\pi}^*(\boldsymbol{\theta})}{\widetilde{\pi}(\boldsymbol{\theta}|\mathcal{D})} d\boldsymbol{\theta}$$
$$+ \int_{\mathcal{A}^c} \widetilde{\pi}^*(\boldsymbol{\theta}) \log \frac{\widetilde{\pi}^*(\boldsymbol{\theta})}{\widetilde{\pi}(\boldsymbol{\theta}|\mathcal{D})} d\boldsymbol{\theta}$$
$$= -\widetilde{\Pi}^*(\mathcal{A}) \int_{\mathcal{A}} \frac{\widetilde{\pi}^*(\boldsymbol{\theta})}{\widetilde{\Pi}^*(\mathcal{A})} \log \frac{\widetilde{\pi}(\boldsymbol{\theta}|\mathcal{D})}{\widetilde{\pi}^*(\boldsymbol{\theta})} d\boldsymbol{\theta}$$
$$- \widetilde{\Pi}^*(\mathcal{A}^c) \int_{\mathcal{A}^c} \frac{\widetilde{\pi}^*(\boldsymbol{\theta})}{\widetilde{\Pi}^*(\mathcal{A}^c)} \log \frac{\widetilde{\pi}(\boldsymbol{\theta}|\mathcal{D})}{\widetilde{\pi}^*(\boldsymbol{\theta})} d\boldsymbol{\theta}$$

$$\geq \widetilde{\Pi}^*(\mathcal{A})\log\frac{\widetilde{\Pi}^*(\mathcal{A})}{\widetilde{\Pi}(\mathcal{A}|\mathcal{D})} + \widetilde{\Pi}^*(\mathcal{A}^c)\log\frac{\widetilde{\Pi}^*(\mathcal{A}^c)}{\widetilde{\Pi}(\mathcal{A}^c|\mathcal{D})},$$

Jensen's inequality

where the above lines hold for any set $\mathcal{A}$. Since $\widetilde{\Pi}(\mathcal{A}|\mathcal{D}) \leq 1$,

$$\begin{aligned}
&\geq \widetilde{\Pi}^*(\mathcal{A})\log\widetilde{\Pi}^*(\mathcal{A}) + \widetilde{\Pi}^*(\mathcal{A}^c)\log\widetilde{\Pi}^*(\mathcal{A}^c) - \widetilde{\Pi}^*(\mathcal{A}^c)\log\widetilde{\Pi}(\mathcal{A}^c|\mathcal{D})\\
&\geq -\widetilde{\Pi}^*(\mathcal{A}^c)\log\widetilde{\Pi}(\mathcal{A}^c|\mathcal{D}) - \log 2,\\
&\quad (\because\ x\log x + (1-x)\log(1-x) \geq -\log 2)\\
&= -\widetilde{\Pi}^*(\mathcal{A}^c)\Bigg(\underbrace{\log\int_{\mathcal{A}^c}(P_{\boldsymbol{\theta}}^n/P_0^n)\widetilde{\pi}(\boldsymbol{\theta})d\boldsymbol{\theta}}_{\mathcal{E}_{1n}} - \underbrace{\log\int(P_{\boldsymbol{\theta}}^n/P_0^n)\widetilde{\pi}(\boldsymbol{\theta})d\boldsymbol{\theta}}_{\mathcal{E}_{2n}}\Bigg)\\
&\quad - \log 2
\end{aligned}$$

The above representation is similar to the proof of Theorems 3.1 and 3.2 in Bhattacharya and Maiti (2021). For any $q \in \mathcal{Q}^{\mathbf{MF}}$,

$$\begin{aligned}
-\widetilde{\Pi}^*(\mathcal{A}^c)\mathcal{E}_{1n} &\leq d_{\mathrm{KL}}(\widetilde{\pi}^*, \widetilde{\pi}(|\mathcal{D})) - \widetilde{\Pi}^*(\mathcal{A}^c)\mathcal{E}_{2n} + \log 2\\
&\leq d_{\mathrm{KL}}(\pi^*, \pi(|\mathcal{D})) - \widetilde{\Pi}^*(\mathcal{A}^c)\mathcal{E}_{2n} + \log 2\\
&\qquad \text{by Lemma A.5}\\
&\leq d_{\mathrm{KL}}(q, \pi(|\mathcal{D})) - \widetilde{\Pi}^*(\mathcal{A}^c)\mathcal{E}_{2n} + \log 2\\
&\qquad \pi^* \text{ is the KL minimizer}\\
&\leq \underbrace{d_{\mathrm{KL}}(q, \pi) + \sum_{\boldsymbol{z}}\int \log\frac{P_0^n}{P_{\boldsymbol{\theta}}^n}q(\boldsymbol{\theta}, \boldsymbol{z})d\boldsymbol{\theta}}_{\mathcal{E}_{3n}}\\
&\quad + (1 - \widetilde{\Pi}^*(\mathcal{A}^c))\mathcal{E}_{2n} + \log 2\\
&= \mathcal{E}_{3n} + (1 - \widetilde{\Pi}^*(\mathcal{A}^c))\mathcal{E}_{2n} + \log 2 \qquad (13)
\end{aligned}$$

where the fourth inequality in the above equation follows since

$$\begin{aligned}
d_{\mathrm{KL}}(q, \pi(|\mathcal{D})) &= \sum_{\boldsymbol{z}}\int(\log q(\boldsymbol{\theta}, \boldsymbol{z}) - \log P_{\boldsymbol{\theta}}^n - \log\pi(\boldsymbol{\theta}, \boldsymbol{z})\\
&\quad + \log m(\mathcal{D}))q(\boldsymbol{\theta}, \boldsymbol{z})d\boldsymbol{\theta}\\
&= \underbrace{\sum_{\boldsymbol{z}}\int(\log q(\boldsymbol{\theta}, \boldsymbol{z}) - \log\pi(\boldsymbol{\theta}, \boldsymbol{z}))q(\boldsymbol{\theta}, \boldsymbol{z})d\boldsymbol{\theta}}_{d_{\mathrm{KL}}(q,\pi)}\\
&\quad + \sum_{\boldsymbol{z}}\int(\log P_0^n - \log P_{\boldsymbol{\theta}}^n)q(\boldsymbol{\theta}, \boldsymbol{z})d\boldsymbol{\theta}\\
&\quad + \underbrace{\log m(\mathcal{D}) - \log P_0^n}_{\mathcal{E}_{2n}}
\end{aligned}$$

where $m(\mathcal{D})$ is the marginal distribution of data as in (5).

Take $\mathcal{A} = \mathcal{H}_{M_n\epsilon_n}^c = \{\boldsymbol{\theta} : d_{\mathrm{H}}(P_0, P_{\boldsymbol{\theta}}) > M_n\epsilon_n\}$

If Lemmas 4.1 and 4.2 hold, then by Lemma A.8, it can be shown that $\mathcal{E}_{1n} \leq -nCM_n^2\epsilon_n^2/\sum u_l$ for any $M_n \to \infty$ with high probability.

If Lemma 4.3 condition 1. holds, then by Lemma A.9, $\mathcal{E}_{2n} \leq nM_n(\sum_{l=0}^L r_l + \xi)$ for any $M_n \to \infty$.

If Lemma 4.3 condition 2. hold, then by Lemma A.10, $\mathcal{E}_{3n} \leq nM_n(\sum_{l=0}^L r_l + \xi)$ for any $M_n \to \infty$.

Therefore, by (13), we get

$$\begin{aligned}
\frac{nCM_n^2\epsilon_n^2}{\sum u_l}\widetilde{\Pi}^*\left(\mathcal{H}_{M_n\epsilon_n}^c\right) &\leq nM_n(\sum_{l=0}^L r_l + \xi) + nM_n(\sum_{l=0}^L r_l + \xi) + \log 2\\
&\leq nM_n(\sum_{l=0}^L r_l + \xi) + nM_n(\sum_{l=0}^L r_l + \xi)\\
&\quad + M_n(\sum_{l=0}^L r_l + \xi)
\end{aligned}$$

$$\implies \widetilde{\Pi}^*\left(\mathcal{H}_{M_n\epsilon_n}^c\right) \leq \frac{3M_n(\sum_{l=0}^L r_l + \xi)\sum u_l}{C_1 M_n^2\epsilon_n^2}$$

Taking $\epsilon_n = \sqrt{\sum_{l=0}^L (r_l + \xi)\sum u_l}$ and noting $M_n \to \infty$, the proof follows. □

We next give conditions on the prior probabilities λ_l and σ_0 to guarantee that Lemmas 4.1–4.3 hold. This in turn implies the conditions of Theorem 4.4 hold and variational posterior is consistent.

Corollary 4.5. *Let $\sigma_0^2 = 1$, $-\log\lambda_l = \log(k_{l+1}) + C_l(k_l+1)\vartheta_l$, then conditions of Theorem 4.4 hold and $\widetilde{\Pi}^*$ as in (10) satisfies*

$$\widetilde{\Pi}^*(\mathcal{H}_{M_n\epsilon_n}^c) \to 0, \quad n \to \infty$$

in P_0^n probability where and $\mathcal{H}_{M_n\epsilon_n} = \{\boldsymbol{\theta} : d_{\mathrm{H}}(P_0, P_{\boldsymbol{\theta}}) \leq M_n\epsilon_n\}$ is the Hellinger neighborhood of radius $M_n\epsilon_n$.

The proof of the corollary has been provided in Appendix A.

In the preceding corollary, note that our expression of prior inclusion probability varies as a function of l thereby providing a handle on layer-wise sparsity. Indeed, using these expressions in numerical studies further substantiates the theoretical framework developed in this section.

Remark (*Optimal Contraction*)**.** For a fixed choice of $\boldsymbol{k}$, the optimal contraction rate is achieved at $\boldsymbol{s}^\star, \boldsymbol{B}^\star = \underset{\boldsymbol{s},\boldsymbol{B}}{\operatorname{argmin}}(\sum r_l + \xi)$. Thus, $\boldsymbol{s}^\star$ and $\boldsymbol{B}^\star$ are the optimal values of $\boldsymbol{s}$ and $\boldsymbol{B}$ which give the best sparse network with minimal loss in the true accuracy. The corresponding probability expressions in Corollary 4.5 can be accordingly modified by setting $\boldsymbol{s} = \boldsymbol{s}^\star$ and $\boldsymbol{B} = \boldsymbol{B}^\star$ in the expressions of ϑ_l and r_l in (12).

5. Implementation details

Evidence Lower Bound. The ELBO presented in (9) is given by $\mathcal{L} = -E_q[\log P_{\boldsymbol{\theta}}^n] + d_{\mathrm{KL}}(q, \pi)$ which is further simplified as

$$\begin{aligned}
&-E_q[\log P_{\boldsymbol{\theta}}^n] + d_{\mathrm{KL}}(q, \pi)\\
&= -\mathbb{E}_{q(\boldsymbol{\theta}|\boldsymbol{z})q(\boldsymbol{z})}[\log P_{\boldsymbol{\theta}}^n] + d_{\mathrm{KL}}\left(q(\boldsymbol{\theta}|\boldsymbol{z})q(\boldsymbol{z}), \pi(\boldsymbol{\theta}|\boldsymbol{z})\pi(\boldsymbol{z})\right)\\
&= -\mathbb{E}_{q(\boldsymbol{\theta}|\boldsymbol{z})q(\boldsymbol{z})}[\log P_{\boldsymbol{\theta}}^n] + \sum_{l,j} d_{\mathrm{KL}}(q(z_{lj})||\pi(z_{lj}))\\
&\quad + \sum_{l,j}\Big[q(\boldsymbol{z}_{lj} = 1)d_{\mathrm{KL}}(q(\overline{\boldsymbol{w}}_{lj}|z_{lj} = 1)||\pi(\overline{\boldsymbol{w}}_{lj}|z_{lj} = 1))\\
&\quad + q(\boldsymbol{z}_{lj} = 0)d_{\mathrm{KL}}(q(\overline{\boldsymbol{w}}_{lj}|z_{lj} = 0)||\pi(\overline{\boldsymbol{w}}_{lj}|z_{lj} = 0))\Big]\\
&= -\mathbb{E}_{q(\boldsymbol{\theta}|\boldsymbol{z})q(\boldsymbol{z})}[\log P_{\boldsymbol{\theta}}^n] + \sum_{l,j} d_{\mathrm{KL}}(q(z_{lj})||\pi(z_{lj}))\\
&\quad + \sum_{l,j} q(\boldsymbol{z}_{lj} = 1)d_{\mathrm{KL}}(q(\overline{\boldsymbol{w}}_{lj}|z_{lj} = 1)||\pi(\overline{\boldsymbol{w}}_{lj}|z_{lj} = 1))\\
&= -\mathbb{E}_{q(\boldsymbol{\theta}|\boldsymbol{z})q(\boldsymbol{z})}[\log P_{\boldsymbol{\theta}}^n] + \sum_{l,j} d_{\mathrm{KL}}(q(z_{lj})||\pi(z_{lj}))\\
&\quad + \sum_{l,j} q(\boldsymbol{z}_{lj} = 1)d_{\mathrm{KL}}(N(\boldsymbol{\mu}_{lj}, \operatorname{diag}(\boldsymbol{\sigma}_{lj}^2))||N(0, \sigma_0^2\boldsymbol{I}))
\end{aligned}$$

The KL of discrete variables appearing in the above expression creates a challenge in practical implementation. Jang et al. (2017), Maddison et al. (2017) proposed to replace discrete random variable with its continuous relaxation. Specifically, the continuous relaxation approximation is achieved through Gumbel-softmax (GS) distribution, that is $q(z_{lj}) \sim \mathrm{Ber}(\gamma_{lj})$ is approximated

Algorithm 1 Variational inference in SS-IG Bayesian neural networks

```
Inputs: training dataset, network architecture, and optimizer tuning parameters.
Model inputs: prior parameters for $\boldsymbol{\theta}, \boldsymbol{z}$.
Variational inputs: number of Monte Carlo samples $S$.
Output: Variational parameter estimates of network weights and sparsity.
Method: Set initial values of variational parameters.
repeat
    Generate $S$ samples from $\boldsymbol{\zeta}_{lj} \sim N(0, \boldsymbol{I})$ and $u_{lj} \sim U(0,1)$
    Generate $S$ samples for $(z_{lj}, \tilde{z}_{lj})$ using $u_{lj}$
    Use $\boldsymbol{\mu}_{lj}, \boldsymbol{\sigma}_{lj}, \boldsymbol{\zeta}_{lj}$ and $z_{lj}$ to compute loss (ELBO) in forward pass
    Use $\boldsymbol{\mu}_{lj}, \boldsymbol{\sigma}_{lj}, \boldsymbol{\zeta}_{lj}$ and $\tilde{z}_{lj}$ to compute gradient of loss in backward pass
    Update the variational parameters with gradient of loss using stochastic gradient descent algorithm (e.g. Adam (Kingma & Ba, 2015))
until change in ELBO $< \epsilon$
```

by $q(\tilde{z}_{lj}) \sim \text{GS}(\gamma_{lj}, \tau)$, where

$$\tilde{z}_{lj} = (1 + \exp(-\eta_{lj}/\tau))^{-1},$$
$$\eta_{lj} = \log(\gamma_{lj}/(1-\gamma_{lj})) + \log(u_{lj}/(1-u_{lj})), \quad u_{lj} \sim U(0,1)$$

where τ is the temperature. We set $\tau = 0.5$ for this paper (also see section 5 in Bai et al. (2020)). $\tilde{z}_{lj}$ is used in the backward pass for easier gradient calculation, while z_{lj} will be used for selecting nodes in the forward pass. We use non-centered parameterization for the Gaussian slab variational approximation where $N(\boldsymbol{\mu}_{lj}, \text{diag}(\boldsymbol{\sigma}^2_{lj}))$ is reparameterized as $\boldsymbol{\mu}_{lj} + \boldsymbol{\sigma}_{lj} \odot \boldsymbol{\zeta}_{lj}$ for $\boldsymbol{\zeta}_{lj} \sim N(0, \boldsymbol{I})$, where $\odot$ denotes the entry-wise (Hadamard) product.

6. Numerical experiments

In this section, we present several numerical experiments to demonstrate the performance of our spike-and-slab independent Gaussian (SS-IG) Bayesian neural networks which we implement in PyTorch (Paszke et al., 2019). Further, to evaluate the efficacy of the variational inference we benchmark our model on synthetic as well as real datasets. Our numerical investigation justifies the use of proposed choices of prior hyperparameters specifically layer-wise prior inclusion probabilities, which in turn substantiates the significance of our theoretical developments. With fully Bayesian treatment, we are also able to quantify the uncertainties for the parameter estimates and variational inference helps to scale our model to large network architectures as well as complex datasets.

We compare our sparse model with a node selection technique: horseshoe BNN (HS-BNN) (Ghosh et al., 2019) and an edge selection technique: spike-and-slab BNN (SV-BNN) (Bai et al., 2020) in the second simulation study and UCI regression dataset examples. We use optimal choices of prior parameters and fine tuning parameters provided by the authors of HS-BNN and SV-BNN in their respective models. Further we compare our model against dense variational BNN model (VBNN) (Blundell et al., 2015) in all of the experiments. Since it has no sparse structure, it serves as a baseline allowing to check whether sparsity compromises accuracy. In all the experiments, we fix $\sigma_0^2 = 1$ and $\sigma_e^2 = 1$. For our model, the choices of layer-wise λ_l follow from Corollary 4.5: $\lambda_l = (1/k_{l+1})\exp(-C_l(k_l+1)\vartheta_l)$. We take C_l values in the negative order of 10 such that prior inclusion probabilities do not fall below 10^{-50} otherwise λ_l values close to 0 might prune away all the nodes from a layer (check appendix B for more discussion). The remaining tuning parameter details such as learning rate, minibatch size, and initial parameter choice are provided in the appendix B. The prediction accuracy is calculated using variational Bayes posterior mean estimator with 30 Monte Carlo samples in testing phase.

Node sparsity estimates. In our experiments, we provide node sparsity estimates for each hidden layer separately. For all models, the node sparsity in a given hidden layer is the ratio of number of neurons with at least one nonzero incoming edge over the original number of neurons present in that layer before training. The layer-wise node sparsity estimates give clear picture of the structural compactness of the trained model during test time. The structurally compact trained model has lower latency during inference stage.

6.1. Simulation study - I

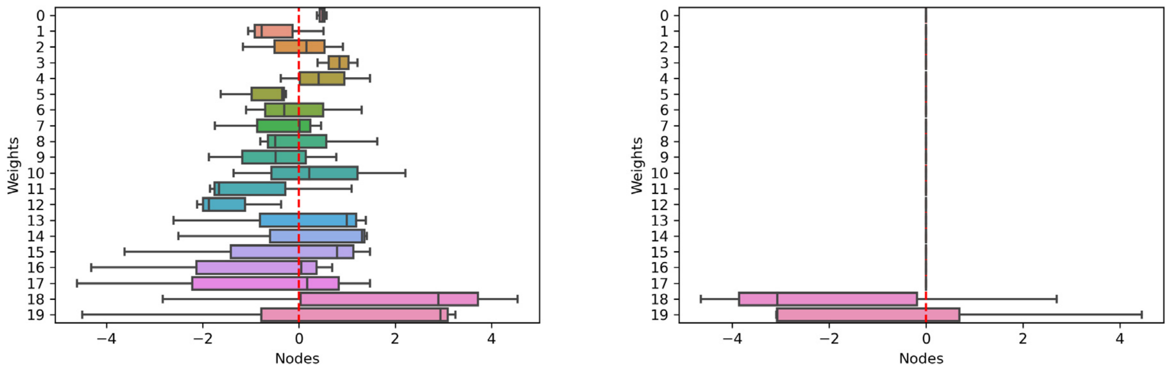
We consider a two dimensional regression problem where the true response y_0 is generated by sampling X from $U([-1,1]^2)$ and feeding it to a deep neural network with known parameters. We add a random Gaussian noise with $\sigma = 5\%\sqrt{Var(y_0)}$ to y_0 to get noisy outputs y. We create the dataset using a shallow neural network consisting of 2 inputs, one hidden layer with 2 nodes and 1 output (2-2-1 network). We train our SS-IG model and VBNN model using a single hidden layer network with 20 neurons in the hidden layer and administer sigmoid activation. Each model is trained till convergence. We found that both models give competitive predictive performance while fitting the given data. In Fig. 2 we plot the magnitudes of the incoming weights into the hidden layer nodes using boxplots. Our model with the help of spike and slab prior is able to prune away redundant nodes not required for fitting the model. Since VBNN is densely connected, it shows all the nodes being active in its final model. From this experiment, it is clear that neural networks can be pruned leading to more compact models at inference stage without compromising the accuracy. We also performed the same experiment with a wider neural network consisting of 100 nodes in the single hidden layer and provide the results in Appendix B. There again we show that our model can easily recover very sparse solution with competitive predictive performance.

6.2. Simulation study - II

We consider a nonlinear regression example where we generate the data from the following model:

$$y = \frac{7x_2}{1+x_1^2} + \sin(x_3x_4) + 2x_5 + \varepsilon,$$

where $\varepsilon \sim N(0,1)$. Further all the covariates are i.i.d. $N(0,1)$ and independent of ε. We generated 3000 data entries to create the training data for the experiment. Additional 1000 observations were generated for testing. We modeled this data using 2-hidden layer neural network which consists of 20 neurons per hidden layer. Sigmoid activation function is administered for each model used for comparative analysis. Table 1 provides the RMSEs on train and test dataset as well as layer-wise node sparsity estimates for SS-IG, SV-BNN, HS-BNN, and VBNN models. Our model is extremely well at pruning redundant nodes which leads to the most compact model compared to the other sparse models: SV-BNN and HS-BNN. Moreover it exhibits lower root mean squared error (RMSE) values on test data among the sparse models while showing similar predictive performance compared to the densely connected VBNN. This experiment further underscores the major benefit of our proposed approach to generate very compact models which could reduce computational times and memory usage at inference stage.



(a) VBNN

(b) SS-IG

Fig. 2. Node-wise weight magnitudes recovered by VBNN and proposed SS-IG model in the synthetic regression data generated using 2-2-1 network. The boxplots show the distribution of incoming weights into a given hidden layer node.

Table 1
Performance of the proposed SS-IG, SV-BNN, HS-BNN, and VBNN models in simulation study II. Each model was trained for 10k epochs with learning rate 5×10^{-3}. Mean and S.D. of RMSE values and median sparsity estimates were calculated from last 1000 epochs (with jump of 10 giving us sample of 100). The sparsity estimates are given as a tuple of 2 values representing layer-1 and layer-2 node sparsities.

Model	Train RMSE	Test RMSE	Sparsity estimate
SS-IG	1.2087 ± 0.0490	1.1947 ± 0.0587	(0.35, 0.05)
SV-BNN	1.2897 ± 0.0323	1.2760 ± 0.0363	(0.45, 0.35)
HS-BNN	1.2580 ± 0.0305	1.2436 ± 0.0394	(1.00, 1.00)
VBNN	1.1661 ± 0.0335	1.1614 ± 0.0349	NA

6.3. UCI regression datasets

We apply our model to traditional UCI regression datasets (Dua & Graff, 2017) and contrast our performance against SV-BNN, HS-BNN, and VBNN models. We follow the protocol proposed by Hernandez-Lobato and Adams (2015) and train a single layer neural network with sigmoid activations. For smaller datasets - *Concrete, Wine, Power Plant, Kin8nm*, we take 50 nodes in the hidden layer, while for larger datasets - *Protein, Year*, we take 100 nodes in the hidden layer. We spilt data randomly while maintaining 9:1 train–test ratio in each case and for smaller datasets we repeat this technique 20 times. In *Protein* data we perform 5 repetitions while in *Year* data we use a single random split (more details in Appendix B). For the comparative analysis, we benchmark against SV-BNN, HS-BNN and VBNN. Moreover, VBNN test RMSEs serve as baseline in each dataset. Table 2 summarizes our results including the sparsity estimate representing hidden layer-1 node sparsity (since there is only one hidden layer in the networks considered).

We achieve lower RMSEs compared to SV-BNN and HS-BNN in *Power Plant, Kin8 nm,* and *Year* datasets and in other cases we achieve comparable RMSE values. In all the datasets, our predictive performance is close to the dense baseline of VBNN. We provide node sparsity estimates in our SS-IG and SV-BNN models. HS-BNN was not able to achieve sparse structure which is consistent with the results provided in the appendix of Ghosh et al. (2019). In contrast to HS-BNN, our model sparsifies the model during training without requiring ad-hoc thresholding rule for pruning. Table 2 demonstrates that our model uniformly achieves better sparsity than SV-BNN. In particular, *Concrete* and *Wine* datasets show the high compressive ability of our model over SV-BNN leading to very compact models for inference.

6.4. Image classification datasets

Here, we benchmark the empirical performance of our proposed SS-IG method on network architectures and image classification datasets used in practice.

Baselines. We compare our model against VBNN model which serves as a dense baseline to gauge the trade-off between predictive performance and sparsity. Moreover, to highlight the complementary behavior in memory and computational efficiency of node selection compared to edge selection achieved via Bayesian spike-and-slab prior framework, we compare our model against the edge selection model, SV-BNN.

Network architectures. We consider 2 neural network model architectures: (i) multi-layer perceptron (MLP), and (ii) Lenet-Caffe. In MLP model, we take 2 hidden layers with 400 neurons in each layer. Output layer has 10 neurons since there are 10 classes in both datasets. Next, Lenet-Caffe model has 2 convolutional layers with 20 and 50 feature maps respectively with filter size 5×5 for both layers. In SS-IG model, for convolution layers, we prune output channels (similar to neurons in linear layers) using our spike-and-slab prior where each output channel is assigned a Bernoulli variable to collectively prune parameters incident on that channel. On the other hand for SV-BNN model, each weight in the convolution layer is assigned a spike-and-slab prior which prunes weights similar to fully connected layers. We apply 2×2 max pooling layer after each convolution layer. The flattened feature layer after second convolution layer has size $4 * 4 * 50 = 800$ serving as input to the fully connected block, where there are 2 hidden layers with 800 and 500 neurons respectively. The output layer has 10 neurons.

Datasets. We apply each network architecture on 2 image classification datasets: (i) MNIST: dataset of 60,000 small square 28×28 pixel grayscale images of handwritten single digits between 0 and 9, and (ii) Fashion-MNIST: dataset of 60,000 small square 28×28 pixel grayscale images of items of 10 types of clothing. We preprocess the images in the MNIST data by dividing their pixel values by 126. In Fashion-MNIST data, we horizontally flip images at random with probability of 0.5.

Metrics. We quantify the predictive performance using the accuracy of the test data (MNIST and Fashion-MNIST). Besides the test accuracy, we evaluate our model against SV-BNN using the metrics that relate to the model compression and computational complexity. First the *compression ratio* is the ratio of number of nonzero weights in the compressed network versus the dense model and is an indicator of storage cost at test-time. Next,

Table 2
Results on UCI regression datasets.

Dataset	$n(k_0)$	Test RMSE				Sparsity estimate	
		SS-IG	SV-BNN	HS-BNN	VBNN	SS-IG	SV-BNN
Concrete	1030 (8)	7.92 ± 0.68	8.22 ± 0.70	5.34 ± 0.53	7.34 ± 0.62	0.42 ± 0.06	0.98 ± 0.02
Wine	1599 (11)	0.66 ± 0.05	0.65 ± 0.05	0.66 ± 0.05	0.64 ± 0.05	0.18 ± 0.05	0.87 ± 0.04
Power Plant	9568 (4)	4.28 ± 0.20	4.32 ± 0.19	4.34 ± 0.18	4.27 ± 0.17	0.18 ± 0.03	0.24 ± 0.03
Kin8 nm	8192 (8)	0.09 ± 0.00	0.11 ± 0.01	0.10 ± 0.00	0.09 ± 0.00	0.43 ± 0.04	0.47 ± 0.04
Protein	45730 (9)	4.85 ± 0.05	4.93 ± 0.06	4.59 ± 0.02	4.78 ± 0.06	0.81 ± 0.03	0.93 ± 0.03
Year	515345 (90)	8.68 ± NA	8.78 ± NA	9.33 ± NA	8.67 ± NA	0.71 ± NA	0.78 ± NA

we present layer-wise node sparsities in MLP experiments to highlight the computational speedups at test-time. In Lenet-Caffe experiments, we provide the *floating point operations (FLOPs) ratio* which is the ratio of number of FLOPs required to predict y from x during test time in the compressed network versus its dense counterpart. We have detailed the FLOPs calculation in neural networks in Appendix B.

Nonlinear activation. We use swish activations (Elfwing et al., 2018; Ramachandran et al., 2017) instead of ReLUs in our proposed SS-IG model to avoid the dying neuron problem where ReLU neurons become inactive and only output 0 for any input (Lu et al., 2020). Specifically in large scale datasets turning off a node with more than 100 incoming edges adversely impacts the training process of ReLU networks. Smoother activation functions such as sigmoid, tanh, swish etc help alleviate this problem. We choose swish since it has the best performance. For VBNN and SV-BNN, we use ReLU activations as recommended by their authors.

MLP experiments

The results of MLP network experiments on MNIST and Fashion-MNIST are presented in Fig. 3. We provide test data accuracy, model compression ratio, and layer-wise node sparsities in each experiment.

In MLP-MNIST experiment (Figs. 3(a)–3(d)), we observe that VBNN and SS-IG models only require ∼400 epochs to achieve stable predictive performance (Fig. 3(a)). In contrast, SV-BNN slightly degrades after 600 epochs and takes longer to achieve convergence in layer-wise node sparsities compared to our approach (Figs. 3(c) and 3(d)). Moreover, for SS-IG model, we observe that as we start to learn sparse network our model shows peak test accuracy when most of the nodes are present in the model and it starts to drop as we learn sparser network and ultimately the test accuracy stabilizes when the node sparsities converge. Furthermore, SV-BNN has better model compression ratio (Fig. 3(b)) in this experiment at the expense of lower predictive performance. Our method is prunes off ∼80% of first hidden layer nodes and ∼90% of second hidden layer nodes at the expense of ∼2% accuracy loss due to sparsification compared to the dense VBNN.

In MLP-Fashion-MNIST experiment (Figs. 3(e)–3(h)), we observe that VBNN model takes ∼200 epochs and our model takes ∼600 epochs for convergence. SV-BNN model takes longer to achieve convergence in layer-wise node sparsities (Figs. 3(g) and 3(h)). We also observe the complementary behavior of our model and SV-BNN in memory and computational efficiency where our model achieves better layer-wise node sparsities and SV-BNN has better model compression ratio (Fig. 3(f)) with both models having similar predictive performance (Fig. 3(e)). Furthermore, our method prunes off ∼90% of first hidden layer nodes and ∼92% of second hidden layer nodes at the expense of ∼3% accuracy loss due to sparsification compared to the densely connected VBNN.

Lenet-Caffe experiments

The results of more complex Lenet-Caffe network experiments on MNIST and Fashion-MNIST are presented in Fig. 4. We provide test data accuracy, model compression ratio, and FLOPs ratio in each experiment over 1200 epochs. Here, FLOPs ratio serves as a collective indicator of layer-wise node sparsities since FLOPs are directly related to how many neurons or channels are remaining in linear or convolution layers respectively.

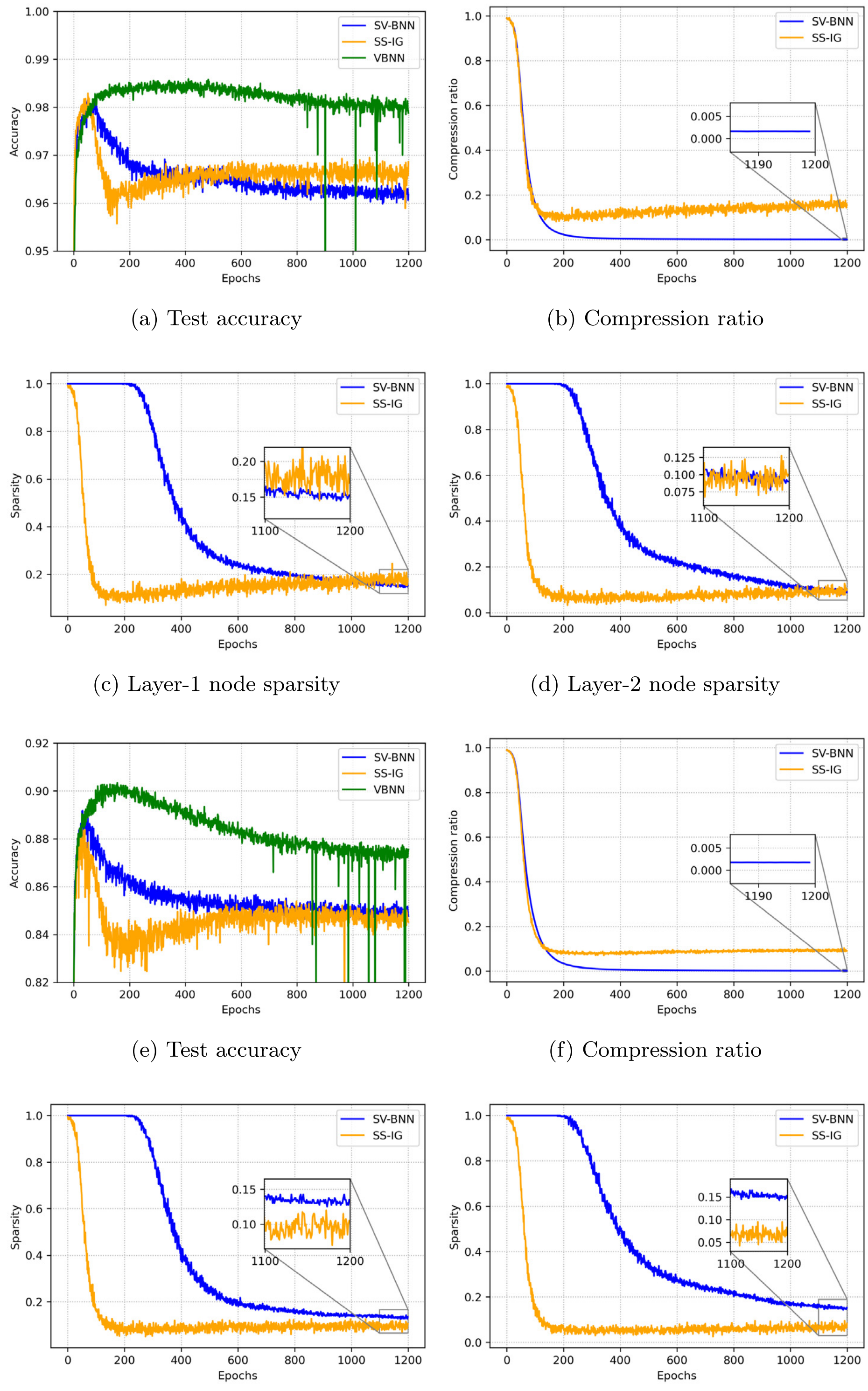
In Lenet-Caffe-MNIST experiment (Figs. 4(a)–4(c)), we observe that our model has better predictive accuracy than SV-BNN (Fig. 4(a)). Moreover, we achieve 10% more reduction in Flops (Fig. 4(c))) compared to SV-BNN whereas SV-BNN achieves better model compression than our approach (Fig. 4(b)). In particular, we prune out more output channels in two convolution layers and nodes in two fully connected layers leading to lower FLOPs at inference compared to SV-BNN. We only include FLOPs ratio for brevity. Lastly, our method is able to reduce the FLOPs of the model during inference at test-time by 90% at the expense of ∼0.5% accuracy loss due to sparsification compared to the densely connected VBNN.

In Lenet-Caffe-Fashion-MNIST experiment (Figs. 4(d)–4(f)), we observe that both SS-IG and SV-BNN have similar test accuracies at convergence (Fig. 4(d)). However, our model has 40% less FLOPs (Fig. 4(f)) during inference stage compared to SV-BNN which again achieves better model compression (Fig. 4(e)). In comparison to SV-BNN, we observe fewer output channels in two convolution layers and nodes in two fully connected layers leading to lower FLOPs at inference. However, we only present FLOPs ratio for brevity. This highlights the complementary nature of our method of node selection that leads to a structurally sparse model with significantly lower (almost 5 times) FLOPs compared to weight pruning approach, SV-BNN, which induces unstructured sparsity in the pruned network leading to significant model compression with low storage cost. Lastly, our method leads to a sparse model with only 8% of the FLOPs as compared to VBNN at the expense of ∼3% accuracy loss underscoring the trade-off between predictive accuracy and sparsity.

7. Conclusion and discussion

Deep learning has been harnessed by big industrial corporations in recent years to improve their products. However, as deep learning models are pushed into smaller and smaller embedded devices, such as smart cameras recognizing visitors at your front door, designing resource-efficient neural networks for real-time, on-device inference is of practical importance. Our work addresses this computational bottleneck by compressing neural networks by inducing structured sparsity during training. The estimation of posterior allows us to quantify uncertainties around the parameter estimates which can be vital in medical diagnostics.

In this paper, we have proposed sparse deep Bayesian neural networks using spike-and-slab priors for optimal node recovery. Our method incorporates layer-wise prior inclusion probabilities and recovers underlying structurally sparse model effectively. Our theoretical developments highlight the conditions required for



(a) Test accuracy

(b) Compression ratio

(c) Layer-1 node sparsity

(d) Layer-2 node sparsity

(e) Test accuracy

(f) Compression ratio

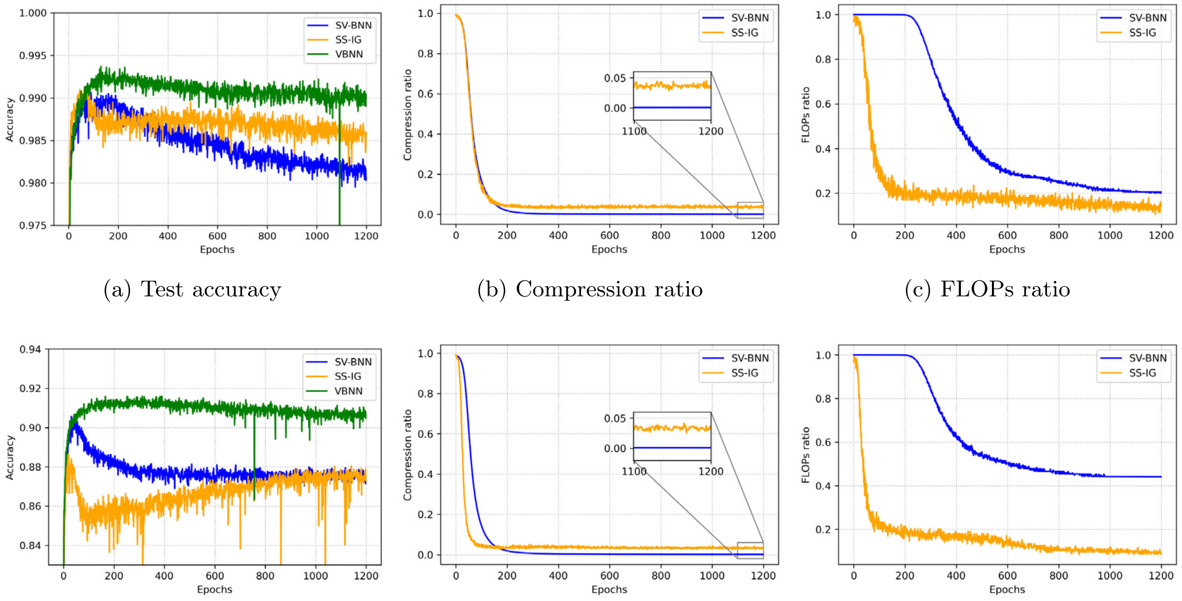
(g) Layer-1 node sparsity

(h) Layer-2 node sparsity

Fig. 3. MLP architecture experiment results. First two rows (a)–(d) represent the MLP on MNIST experiment results. Bottom two rows (e)–(h) represent the MLP on Fashion-MNIST experiment results.

the posterior consistency of the variational posterior to hold. With the layer-wise characterization of prior inclusion probabilities, we show that the proposed sparse BNN approximations can achieve predictive performance comparable to dense networks. Our results relax the constraints of equal number of nodes and uniform bounds on weights thereby achieving optimal node recovery on a more generic neural network structure. The closeness of a true function to the topology induced by layer-wise node distribution depends on the degree of smoothness of the true underlying function. In this work, this has not been studied in depth and forms a promising direction for future work.

We have developed variational posterior consistency in our model under MLP network assumption. One can extend this theoretical derivation to CNN by (see Section 3.4.1 in Gal (2016)).



(a) Test accuracy (b) Compression ratio (c) FLOPs ratio

(d) Test accuracy (e) Compression ratio (f) FLOPs ratio

Fig. 4. Lenet-Caffe architecture experiment results. Top row (a)–(c) represent the Lenet-Caffe on MNIST experiment results. Bottom row (d)–(f) represent the Lenet-Caffe on Fashion-MNIST experiment results.

In fact, each convolutional operation can be taken as a special case of linear mapping with a Toeplitz weight matrix. Thereby, the corresponding weight matrix of the fully connected layer is a large matrix that is mostly zero except for certain blocks (due to local connectivity) where the weights in many of the blocks are equal (due to parameter sharing). To generalize the theory developed for MLPs to CNNs, one will need an adaptation of the sieve construction in (11) for the case of convolutional neural networks together with a rederivation of the Kullback–Leibler neighborhoods of the true density function by modifying the expressions in (12). We leave this development for future work.

Note, in contrast to previous works, our work assumes a spike-and-slab prior on the entire vector of incoming weights and bias onto a node. We underscore the fact that node selection has complementary behavior with edge selection approaches as established by our empirical experiments. Node selection offers significant computational speedup whereas edge selection achieves significant model compression at test-time. The demonstration of the efficacy of our node selection approach opens the avenue for the exploration of sophisticated group sparsity priors for node selection. Our detailed experiments show the subnetwork selection ability of our method which underscores the notion that deep neural networks can be heavily pruned without losing predictive performance. The experiment with convolution neural network (Lenet-Caffe), where we induce structural sparsity via channel pruning in convolution layers, highlights the generalizability of our approach from mere multi-layer perceptron to complex deep learning models. Although our method performs model reduction while maintaining predictive power, further improvements may be obtained by choosing the number of layers in a data-driven fashion and can be a part of future work.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

The authors thank the editor, the action editor, and anonymous referees for their insightful comments and suggestions that significantly improved the quality of the work. The work is partially supported by the grants NSF-1924724, NSF-1952856, NSF-2124605, and an agreement between Michigan State University and Argonne National Laboratory, IF-60562.

Appendix A. Proofs of theoretical results

A.1. Definitions

Definition A.1 (*Sieve*)**.** Consider a sequence of function classes $\mathcal{F}_1 \subseteq \mathcal{F}_2 \subseteq \cdots \subseteq \mathcal{F}_n \subseteq \mathcal{F}_{n+1} \subseteq \cdots \subseteq \mathcal{F}$, where $\forall f \in \mathcal{F}$, $\exists f_n \in \mathcal{F}_n$ s.t. $d(f, f_n) \to 0$ as $n \to \infty$ where $d(.,.)$ is some pseudo-metric on $\mathcal{F}$. More precisely, $\cup_{n=1}^{\infty}\mathcal{F}_n$ is dense in $\mathcal{F}$. $\mathcal{F}_n$ is called a sieve space of $\mathcal{F}$ with respect to the pseudo-metric $d(.,.)$, and the sequence $\{f_n\}$ is called a sieve (Grenander, 1981).

Definition A.2 (*Covering Number*)**.** Let $(V, \|.\|)$ be a normed space, and $\mathcal{F} \subset V$. $\{V_1, \ldots, V_N\}$ is an ε-covering of $\mathcal{F}$ if $\mathcal{F} \subset \cup_{i=1}^{N} B(V_i, \varepsilon)$, or equivalently, $\forall\ \varrho \in \mathcal{F}$, $\exists\ i$ such that $\|\varrho - V_i\| < \varepsilon$. The covering number of $\mathcal{F}$ denoted by $N(\varepsilon, \mathcal{F}, \|.\|) = \min\{n : \exists\, \varepsilon -$ covering over $\mathcal{F}$ of size $n\}$ (Pollard, 1991).

A.2. General lemmas

Lemma A.3. *Let g_1 and g_2 be any two density functions. Then*

$$E_{g_1}(|\log(g_1/g_2)|) \leq d_{\mathrm{KL}}(g_1, g_2) + 2/e$$

Proof. Refer to Lemma 4 in Lee (2000). □

Lemma A.4. *For any $K > 0$, let $\boldsymbol{a}, \boldsymbol{a^0} \in [0,1]^K$ such that $\sum_{k=1}^K a_k = \sum_{k=1}^K a_k^0 = 1$, then the KL divergence between mixture densities $\sum_{k=1}^K a_k g_k$ and $\sum_{k=1}^K a_k^0 g_k^0$ is bounded as*

$$d_{\mathrm{KL}}\left(\sum_{k=1}^K a_k^0 g_k^0, \sum_{k=1}^K a_k g_k\right) \leq d_{\mathrm{KL}}(\boldsymbol{a^0}, \boldsymbol{a}) + \sum_{k=1}^K a_k^0 d_{\mathrm{KL}}(g_k^0, g_k)$$

Proof. Refer to Lemma 6.1 in Chérief-Abdellatif and Alquier (2018). □

Lemma A.5.

$$d_{\mathrm{KL}}(\widetilde{\pi}^*, \widetilde{\pi}(|\mathcal{D})) \leq d_{\mathrm{KL}}(\pi^*, \pi(|\mathcal{D}))$$

Proof. Using Lemma A.4 with $\boldsymbol{a}^0 = \pi^*(\boldsymbol{z})$, $\boldsymbol{a} = \pi(\boldsymbol{z}|\mathcal{D})$, $g^0 = \pi^*(\boldsymbol{\theta}|\boldsymbol{z})$ and $g = \pi(\boldsymbol{\theta}|\boldsymbol{z}, \mathcal{D})$, we get

$$\begin{aligned} d_{\mathrm{KL}}(\widetilde{\pi}^*, \widetilde{\pi}(|\mathcal{D})) &= d_{\mathrm{KL}}(\sum_{\boldsymbol{z}} \pi^*(\boldsymbol{\theta}|\boldsymbol{z})\pi^*(\boldsymbol{z}), \sum_{\boldsymbol{z}} \pi(\boldsymbol{\theta}|\boldsymbol{z}, \mathcal{D})\pi(\boldsymbol{z}|\mathcal{D})) \\ &\leq d_{\mathrm{KL}}(\pi^*(\boldsymbol{z}), \pi(\boldsymbol{z}|\mathcal{D})) \\ &\quad + \sum_{\boldsymbol{z}} d_{\mathrm{KL}}(\pi^*(\boldsymbol{\theta}|\boldsymbol{z}), \pi(\boldsymbol{\theta}|\boldsymbol{z}, \mathcal{D}))\pi^*(\boldsymbol{z}) \\ &= d_{\mathrm{KL}}(\pi^*(\boldsymbol{\theta}, \boldsymbol{z}), \pi(\boldsymbol{\theta}, \boldsymbol{z}|\mathcal{D})) = d_{\mathrm{KL}}(\pi^*, \pi(|\mathcal{D})) \quad \square \end{aligned}$$

Lemma A.6. *For any 1-Lipschitz continuous activation function ψ such that $\psi(x) \leq x\ \forall x \geq 0$,*

$$N(\delta, \mathcal{F}(L, \boldsymbol{k}, \boldsymbol{s}, \boldsymbol{B}), \|.\|_\infty) \leq \sum_{s_L^* \leq s_L} \cdots \sum_{s_0^* \leq s_0} \left[\prod_{l=0}^L \left(\frac{B_l}{\delta B_l/(2(L+1)(\prod_{j=0}^L B_j))} k_{l+1}\right)^{s_l}\right]$$

where N denotes the covering number.

Proof. Given a neural network

$$\begin{aligned} \eta(\boldsymbol{x}) = \boldsymbol{v}_L + \boldsymbol{W}_L\psi(\boldsymbol{v}_{L-1} + \boldsymbol{W}_{L-1}\psi(\boldsymbol{v}_{L-2} \\ + \boldsymbol{W}_{L-2}\psi(\cdots\psi(\boldsymbol{v}_1 + \boldsymbol{W}_1\psi(\boldsymbol{v}_0 + \boldsymbol{W}_0\boldsymbol{x}))))) \end{aligned}$$

for $l \in \{1, \ldots, L\}$, we define $A_l^+\eta : [0,1]^p \to \mathbb{R}^{k_l}$,

$$\begin{aligned} A_l^+\eta(\boldsymbol{x}) = \psi(\boldsymbol{v}_{l-1} + \boldsymbol{W}_{l-1}\psi(\boldsymbol{v}_{l-2} \\ + \boldsymbol{W}_{l-2}\psi(\cdots\psi(\boldsymbol{v}_1 + \boldsymbol{W}_1\psi(\boldsymbol{v}_0 + \boldsymbol{W}_0\boldsymbol{x}))))) \end{aligned}$$

and $A_l^-\eta : \mathbb{R}^{k_{l-1}} \to \mathbb{R}^{k_{L+1}}$,

$$\begin{aligned} A_l^-\eta(\boldsymbol{y}) = \boldsymbol{v}_L + \boldsymbol{W}_L\psi(\boldsymbol{v}_{L-1} \\ + \boldsymbol{W}_{L-1}\psi(\cdots\psi(\boldsymbol{v}_l + \boldsymbol{W}_l\psi(\boldsymbol{v}_{l-1} + \boldsymbol{W}_{l-1}\boldsymbol{y})))) \end{aligned}$$

The above framework is also used in the proof of lemma 5 in Schmidt-Hieber (2020). Next, set $A_0^+\eta(\boldsymbol{x}) = A_{L+2}^-\eta(\boldsymbol{x}) = \boldsymbol{x}$ and further note that for $\eta \in \mathcal{F}(L, \boldsymbol{k})$, $|A_l^+\eta(\boldsymbol{x})|_\infty \leq \prod_{j=0}^{l-1} B_j$ where $\boldsymbol{k} = (p, k_1, \ldots, k_L, k_{L+1})$ and $k_{L+1} = 1$. Next, we derive upper bound on Lipschitz constant of $A_l^-\eta$.

$$\begin{aligned} &|\boldsymbol{W}_L A_L^+\eta(\boldsymbol{x}_1) - \boldsymbol{W}_L A_L^+\eta(\boldsymbol{x}_2)|_\infty \\ &\quad = |A_l^-\eta(A_{l-1}^+\eta(\boldsymbol{x}_1)) - A_l^-\eta(A_{l-1}^+\eta(\boldsymbol{x}_2))|_\infty \end{aligned} \tag{14}$$

l.h.s. is bounded above by $\prod_{j=0}^L B_j$ and r.h.s consists of composition of Lipschitz functions $A_l^-\eta$ and $A_{l-1}^+\eta$ with C_1 and C_2 being corresponding Lipschitz constants. So we can bound r.h.s. by,

$$|A_l^-\eta(A_{l-1}^+\eta(\boldsymbol{x}_1)) - A_l^-\eta(A_{l-1}^+\eta(\boldsymbol{x}_2))|_\infty \leq C_1C_2\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_\infty \quad \forall \boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{R}^p$$

If we choose $\boldsymbol{x}_1 = \boldsymbol{x} \in [0,1]^p$ and $\boldsymbol{x}_2 = \boldsymbol{0}$ then,

$$|A_l^-\eta(A_{l-1}^+\eta(\boldsymbol{x})) - A_l^-\eta(A_{l-1}^+\eta(\boldsymbol{0}))|_\infty \leq C_1C_2 \quad \forall \boldsymbol{x} \in [0,1]^p$$

Since C_2 is Lipschitz constant for $A_{l-1}^+\eta$ and we know that $|A_{l-1}^+\eta|_\infty \leq \prod_{j=0}^{l-2} B_j$. So we get $C_2 \leq 2\prod_{j=0}^{l-2} B_j$. We use this in above expression,

$$|A_l^-\eta(A_{l-1}^+\eta(\boldsymbol{x})) - A_l^-\eta(A_{l-1}^+\eta(\boldsymbol{0}))|_\infty \leq 2C_1\prod_{j=0}^{l-2} B_j \quad \forall \boldsymbol{x} \in [0,1]^p \tag{15}$$

Next we know that l.h.s. of (15) can be bounded above by $2\prod_{j=0}^L B_j$ because of (14). So we get bound on Lipschitz constant of $A_l^-\eta$,

$$2C_1\prod_{j=0}^{l-2} B_j \leq 2\prod_{j=0}^L B_j \implies C_1 \leq \prod_{j=l-1}^L B_j$$

Let $\eta, \eta^* \in \mathcal{F}(L, \boldsymbol{k}, \boldsymbol{s}, \boldsymbol{B})$ be two neural networks with $\overline{\boldsymbol{W}}_l = (\boldsymbol{v}_l, \boldsymbol{W}_l)$ and $\overline{\boldsymbol{W}}_l^* = (\boldsymbol{v}_l^*, \boldsymbol{W}_l^*)$ respectively. Here, we define $\overline{\boldsymbol{\delta}}_l$ using the L_1 norms of the rows of $\overline{\boldsymbol{D}}_l = \overline{\boldsymbol{W}}_l - \overline{\boldsymbol{W}}_l^*$ as follows

$$\overline{\boldsymbol{D}}_l = (\overline{\boldsymbol{d}}_{l1}^\top, \ldots, \overline{\boldsymbol{d}}_{lk_{l+1}}^\top)^\top \qquad \overline{\boldsymbol{\delta}}_l = (\|\overline{\boldsymbol{d}}_{l1}\|_1, \ldots, \|\overline{\boldsymbol{d}}_{lk_{l+1}}\|_1)$$

We choose η, η^* such that $\|\overline{\boldsymbol{\delta}}_l\|_\infty \leq \zeta B_l$. This also means that all parameters in each layer of these two networks are at most ζB_l distance away from each other. Then, we can bound the absolute difference between these two neural networks by,

$$\begin{aligned} &|\eta(\boldsymbol{x}) - \eta^*(\boldsymbol{x})| \\ &\leq \sum_{l=1}^{L+1} |A_{l+1}^-\eta(\psi(\boldsymbol{v}_{l-1} + \boldsymbol{W}_{l-1}A_{l-1}^+\eta^*(\boldsymbol{x}))) \\ &\quad - A_{l+1}^-\eta(\psi(\boldsymbol{v}_{l-1}^* + \boldsymbol{W}_{l-1}^*A_{l-1}^+\eta^*(\boldsymbol{x})))| \\ &\leq \sum_{l=1}^{L+1}\left(\prod_{j=l}^L B_j\right) ||\psi(\boldsymbol{v}_{l-1} + \boldsymbol{W}_{l-1}A_{l-1}^+\eta^*(\boldsymbol{x})) \\ &\quad - \psi(\boldsymbol{v}_{l-1}^* + \boldsymbol{W}_{l-1}^*A_{l-1}^+\eta^*(\boldsymbol{x}))||_\infty \\ &\leq \sum_{l=1}^{L+1}\left(\prod_{j=l}^L B_j\right) \|\boldsymbol{v}_{l-1} - \boldsymbol{v}_{l-1}^* + (\boldsymbol{W}_{l-1} - \boldsymbol{W}_{l-1}^*)A_{l-1}^+\eta^*(\boldsymbol{x})\|_\infty \\ &\leq \sum_{l=1}^{L+1}\left(\prod_{j=l}^L B_j\right) \|\overline{\boldsymbol{\delta}}_{l-1}\|_\infty\|A_{l-1}^+\eta^*(\boldsymbol{x})\|_\infty \\ &\leq \sum_{l=1}^{L+1}\left(\prod_{j=l}^L B_j\right) \zeta B_{l-1}\prod_{j=0}^{l-2} B_j = \zeta(L+1)\left(\prod_{j=0}^L B_j\right) \end{aligned} \tag{16}$$

Recall that we have at most k_l number of nodes in each layer and there are $\binom{k_{l+1}}{s_l} \leq k_{l+1}^{s_l}$ combinations of nodes to choose s_l active nodes in the given layer. Since supremum norm of L_1 norms of the rows of $\overline{\boldsymbol{W}}_l$ is bounded above by B_l in our family of neural networks $\mathcal{F}(L, \boldsymbol{k}, \boldsymbol{s}, \boldsymbol{B})$ so we can discretize these L_1 norms with grid size $\delta B_l/(2(L+1)(\prod_{j=0}^L B_j))$ and obtain upper bound on covering number as follows

$$\begin{aligned} &N(\delta, \mathcal{F}(L, \boldsymbol{k}, \boldsymbol{s}, \boldsymbol{B}), \|.\|_\infty) \\ &\quad \leq \sum_{s_L^* \leq s_L} \cdots \sum_{s_0^* \leq s_0} \left[\prod_{l=0}^L \left(\frac{B_l}{\delta B_l/(2(L+1)(\prod_{j=0}^L B_j))} k_{l+1}\right)^{s_l}\right] \\ &\quad \leq \prod_{l=0}^L \left(2\delta^{-1}(L+1)\left(\prod_{j=0}^L B_j\right) k_{l+1}\right)^{(s_l+1)} \qquad \square \end{aligned} \tag{17}$$

Lemma A.7. *Let* $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}\in\mathcal{F}(L,\boldsymbol{k},\boldsymbol{s},\boldsymbol{B})} \|\eta_{\boldsymbol{\theta}} - \eta_0\|_\infty^2$ *and* $\widetilde{W}_l = \sup_i \|\overline{\boldsymbol{w}}_{li} - \overline{\boldsymbol{w}}_{li}^*\|_1$*, then for any density* $q = \prod_{j=0}^L q(\theta_j)$*,*

$$\begin{aligned}
&\int \|\eta_{\boldsymbol{\theta}} - \eta_{\boldsymbol{\theta}^*}\|_2^2 q(\boldsymbol{\theta})d\boldsymbol{\theta} \\
&\le \sum_{j=0}^{L} c_{j-1}^2 \int \widetilde{W}_j^2 q_j(\theta_j)d\theta_j \prod_{m=j+1}^{L} \int (\widetilde{W}_m + B_m)^2 q(\boldsymbol{\theta})d\boldsymbol{\theta} \\
&\quad + 2\sum_{j=0}^{L}\sum_{j'=0}^{j-1} c_{j-1}c_{j'-1} \int \widetilde{W}_j(\widetilde{W}_j + B_j)q_j(\theta_j)d\theta_j \\
&\quad \times \prod_{m=j+1}^{L} \int (\widetilde{W}_m + B_m)^2 q(\boldsymbol{\theta})d\boldsymbol{\theta} \\
&\quad\quad \times \int \widetilde{W}_{j'} q_{j'}(\theta_{j'})d\theta_{j'} \prod_{m=j'+1}^{j-1} \int (\widetilde{W}_m + B_m) q(\boldsymbol{\theta})d\boldsymbol{\theta}
\end{aligned} \tag{18}$$

where $c_{j-1} \le \prod_{m=0}^{j-1} B_m$*.*

Proof. Let $\eta_{\boldsymbol{\theta}}^l$ be the partial networks defined as

$$\begin{cases}
\eta_{\boldsymbol{\theta}}^0(\boldsymbol{x}) := \psi(\boldsymbol{W}_0\boldsymbol{x} + \boldsymbol{v}_0), \\
\eta_{\boldsymbol{\theta}}^l(\boldsymbol{x}) := \psi(\boldsymbol{W}_l\eta_{\boldsymbol{\theta}}^{l-1}(\boldsymbol{x}) + \boldsymbol{v}_l), \\
\eta_{\boldsymbol{\theta}}^L(\boldsymbol{x}) := \boldsymbol{W}_L\eta_{\boldsymbol{\theta}}^{L-1}(\boldsymbol{x}) + \boldsymbol{v}_L.
\end{cases}$$

Similar to the proof of theorem 2 in Chérief-Abdellatif (2020), define

$$\varphi_l(\boldsymbol{\theta}) = \sup_{x\in[0,1]^p} \sup_{1\le i\le k_{l+1}} |\eta_{\boldsymbol{\theta}}^l(\boldsymbol{x})_i - \eta_{\boldsymbol{\theta}^*}^l(\boldsymbol{x})_i|.$$

We next show by induction

$$\varphi_l(\boldsymbol{\theta}) \le \sum_{j=0}^{l} \widetilde{W}_j c_{j-1} R_{j+1}^l$$

where we define $c_l = \max(\sup_{x\in[0,1]^p}\sup_{1\le i\le k_{l+1}} |\eta_{\boldsymbol{\theta}^*}^l(\boldsymbol{x})_i|, 1)$, $c_0 = 1$, $R_{j+1}^l = \prod_{m=j+1}^l (\widetilde{W}_m + B_m)$.

Claim: $c_l \le B_l c_{l-1}$. Note

$$\begin{aligned}
c_l &\le \sup_{x\in[0,1]^p} \sup_{1\le i\le k_{l+1}} (|\boldsymbol{w}_{li}^{*\top}\eta_{\boldsymbol{\theta}^*}^{l-1}(\boldsymbol{x})| + |v_{li}|) \\
&\le \sup_{x\in[0,1]^p} \sup_{1\le i\le k_{l+1}} (\sum_{j=1}^{k_l} |w_{lij}^*||\eta_{\boldsymbol{\theta}^*}^{l-1}(\boldsymbol{x})_j| + |v_{li}|) \\
&\le \sup_{1\le i\le k_{l+1}} (c_{l-1}\sum_{j=1}^{k_l} |w_{lij}^*| + c_{l-1}|v_{li}|) \\
&\le c_{l-1}\sup_{1\le i\le k_{l+1}} \|\overline{\boldsymbol{w}}_{li}^*\|_1 = B_l c_{l-1}
\end{aligned}$$

where the above result holds since $\sup_i \|\overline{\boldsymbol{w}}_{li}^*\|_1 \le B_l$. Next,

$$\begin{aligned}
&\varphi_l(\boldsymbol{\theta}) \\
&\le \sup_{x\in[0,1]^p} \sup_{1\le i\le k_{l+1}} (\sum_{j=1}^{k_l} |w_{lij}\eta_{\boldsymbol{\theta}}^{l-1}(\boldsymbol{x})_j - w_{lij}^*\eta_{\boldsymbol{\theta}^*}^{l-1}(\boldsymbol{x})_j| + |v_{li} - v_{li}^*|) \\
&\le \sup_{x\in[0,1]^p} \sup_{1\le i\le k_{l+1}} (\sum_{j=1}^{k_l} |w_{lij}\eta_{\boldsymbol{\theta}}^{l-1}(\boldsymbol{x})_j - w_{lij}^*\eta_{\boldsymbol{\theta}}^{l-1}(\boldsymbol{x})_j| \\
&\qquad + |w_{lij}^*\eta_{\boldsymbol{\theta}}^{l-1}(\boldsymbol{x})_j - w_{lij}^*\eta_{\boldsymbol{\theta}^*}^{l-1}(\boldsymbol{x})_j| + |v_{li} - v_{li}^*|) \\
&\le \sup_{x\in[0,1]^p} \sup_{1\le i\le k_{l+1}} (\sum_{j=1}^{k_l} |w_{lij} - w_{lij}^*||\eta_{\boldsymbol{\theta}}^{l-1}(\boldsymbol{x})_j| \\
&\qquad + \sum_{j=1}^{k_l} |w_{lij}^*||\eta_{\boldsymbol{\theta}}^{l-1}(\boldsymbol{x})_j - \eta_{\boldsymbol{\theta}^*}^{l-1}(\boldsymbol{x})_j| + |v_{li} - v_{li}^*|) \\
&\le \sup_{x\in[0,1]^p} \sup_{1\le i\le k_{l+1}} (\sum_{j=1}^{k_l} |w_{lij} - w_{lij}^*||\eta_{\boldsymbol{\theta}}^{l-1}(\boldsymbol{x})_j - \eta_{\boldsymbol{\theta}^*}^{l-1}(\boldsymbol{x})_j| \\
&\qquad + \sum_{j=1}^{k_l} |w_{lij} - w_{lij}^*||\eta_{\boldsymbol{\theta}^*}^{l-1}(\boldsymbol{x})_j| + |v_{li} - v_{li}^*|) \\
&\qquad + \varphi_{l-1}(\boldsymbol{\theta})B_l \\
&\le \widetilde{W}_l(\varphi_{l-1}(\boldsymbol{\theta}) + c_{l-1}) + \varphi_{l-1}(\boldsymbol{\theta})B_l = \varphi_{l-1}(\boldsymbol{\theta})(\widetilde{W}_l + B_l) + c_{l-1}\widetilde{W}_l
\end{aligned}$$

Now applying recursion we get

$$\begin{aligned}
\varphi_l(\boldsymbol{\theta}) &\le (\varphi_{l-2}(\boldsymbol{\theta})(\widetilde{W}_{l-1} + B_{l-1}) + c_{l-2}\widetilde{W}_{l-1})(\widetilde{W}_l + B_l) + c_{l-1}\widetilde{W}_l \\
&= \varphi_{l-2}(\boldsymbol{\theta})(\widetilde{W}_l + B_l)(\widetilde{W}_{l-1} + B_{l-1}) \\
&\quad + c_{l-2}\widetilde{W}_{l-1}(\widetilde{W}_l + B_l) + c_{l-1}\widetilde{W}_l
\end{aligned}$$

Repeating this we get

$$\begin{aligned}
\varphi_l(\boldsymbol{\theta}) &\le \varphi_0(\boldsymbol{\theta})\prod_{j=1}^{l}(\widetilde{W}_j + B_j) + \sum_{j=1}^{l} c_{j-1}\widetilde{W}_j \prod_{u=j+1}^{l}(\widetilde{W}_j + B_j) \\
&= \widetilde{W}_0\prod_{j=1}^{l}(\widetilde{W}_j + B_j) + \sum_{j=1}^{l} B_1\cdots B_{j-1}\widetilde{W}_j \prod_{u=j+1}^{l}(\widetilde{W}_j + B_j) \\
&= \sum_{j=0}^{l} B_1\cdots B_{j-1}\widetilde{W}_j \prod_{u=j+1}^{l}(\widetilde{W}_j + B_j) = \sum_{j=0}^{l}\widetilde{W}_j c_{j-1}R_{j+1}^l
\end{aligned}$$

$$\begin{aligned}
&\int \|\eta_{\boldsymbol{\theta}} - \eta_{\boldsymbol{\theta}^*}\|_2^2 q(\boldsymbol{\theta})d\boldsymbol{\theta} \le \int \|\eta_{\boldsymbol{\theta}} - \eta_{\boldsymbol{\theta}^*}\|_\infty^2 q(\boldsymbol{\theta})d\boldsymbol{\theta} = \int \varphi_L^2(\boldsymbol{\theta})q(\boldsymbol{\theta})d\boldsymbol{\theta} \\
&= \int (\sum_{j=0}^{L}\widetilde{W}_j c_{j-1}R_{j+1}^L)^2 q(\boldsymbol{\theta})d\boldsymbol{\theta} \\
&= \sum_{j=0}^{L} c_{j-1}^2 \int \widetilde{W}_j^2 (R_{j+1}^L)^2 q(\boldsymbol{\theta})d\boldsymbol{\theta} \\
&\quad + 2\sum_{j=0}^{L}\sum_{j'=0}^{j-1} c_{j-1}c_{j'-1}\int \widetilde{W}_j\widetilde{W}_{j'}R_{j+1}^L R_{j'+1}^L q(\boldsymbol{\theta})d\boldsymbol{\theta} \\
&= \sum_{j=0}^{L} c_{j-1}^2 \int \widetilde{W}_j^2 \left(\prod_{m=j+1}^{L}(\widetilde{W}_m + B_m)\right)^2 q(\boldsymbol{\theta})d\boldsymbol{\theta} \\
&\quad + 2\sum_{j=0}^{L}\sum_{j'=0}^{j-1} c_{j-1}c_{j'-1}\int \widetilde{W}_j\widetilde{W}_{j'}\prod_{m=j+1}^{L}(\widetilde{W}_m + B_m) \\
&\quad \times \prod_{m=j'+1}^{L}(\widetilde{W}_m + B_m)q(\boldsymbol{\theta})d\boldsymbol{\theta}
\end{aligned}$$

The proof follows by noting $q(\boldsymbol{\theta}) = \prod_{j=0}^{L} q(\theta_j)$. □

Lemma A.8. *Suppose Lemmas 4.1 and 4.2 in the main paper hold, with dominating probability*

$$\log \int_{\mathcal{H}_{\epsilon_n}^c} \frac{P_{\boldsymbol{\theta}}^n}{P_0^n}\pi(\boldsymbol{\theta})d\boldsymbol{\theta} \le -\frac{Cn\epsilon_n^2}{\sum u_l}$$

Proof. Let $\mathcal{F}_n = \mathcal{F}(L, \boldsymbol{k}, \boldsymbol{s}^\circ, \boldsymbol{B}^\circ)$, $s_l^\circ + 1 = n\epsilon_n^2/\sum_{j=0}^{L} u_j$, $\log B_l^\circ = n\epsilon_n^2/((L+1)\sum_{j=0}^{L}(s_j^\circ + 1))$ and $\mathcal{H}_{\epsilon_n} = \{\boldsymbol{\theta} : d_{\mathrm{H}}(P_0, P_{\boldsymbol{\theta}}) < \epsilon_n\}$ is the

Hellinger neighborhood of size ϵ_n

$$\begin{aligned}\int_{\mathcal{H}^c_{\epsilon_n}} \frac{P^n_{\boldsymbol{\theta}}}{P^n_0}\widetilde{\pi}(\boldsymbol{\theta})d\boldsymbol{\theta} &\le \int_{\mathcal{H}^c_{\epsilon_n}\cap\mathcal{F}_n} \frac{P^n_{\boldsymbol{\theta}}}{P^n_0}\widetilde{\pi}(\boldsymbol{\theta})d\boldsymbol{\theta} + \int_{\mathcal{F}^c_n} \frac{P^n_{\boldsymbol{\theta}}}{P^n_0}\widetilde{\pi}(\boldsymbol{\theta})d\boldsymbol{\theta}\\ &\le \int_{\mathcal{H}^c_{\epsilon_n}\cap\mathcal{F}_n} \frac{P^n_{\boldsymbol{\theta}}}{P^n_0}\widetilde{\pi}(\boldsymbol{\theta})d\boldsymbol{\theta} + \exp\left(-\frac{(C_0/2)n\epsilon_n^2}{\sum u_l}\right)\end{aligned}$$

where the last inequality follows from Lemma 4.2 because by Markov's inequality

$$\begin{aligned}&\mathbb{P}_{P^n_0}\left(\int_{\mathcal{F}^c_n} \frac{P^n_{\boldsymbol{\theta}}}{P^n_0}\widetilde{\pi}(\boldsymbol{\theta})d\boldsymbol{\theta} > \exp\left(-\frac{(C_0/2)n\epsilon_n^2}{\sum u_l}\right)\right)\\ &\le \exp\left(\frac{(C_0/2)n\epsilon_n^2}{\sum u_l}\right)\mathbb{E}_{P^n_0}\left(\int_{\mathcal{F}^c_n} \frac{P^n_{\boldsymbol{\theta}}}{P^n_0}\widetilde{\pi}(\boldsymbol{\theta})d\boldsymbol{\theta}\right)\\ &\le \exp\left(\frac{(C_0/2)n\epsilon_n^2}{\sum u_l}\right)\widetilde{\Pi}(\mathcal{F}^c_n) = \exp\left(-\frac{(C_0/2)n\epsilon_n^2}{\sum u_l}\right) \to 0\end{aligned}$$

Further,

$$\begin{aligned}\int_{\mathcal{H}^c_{\epsilon_n}\cap\mathcal{F}_n} \frac{P^n_{\boldsymbol{\theta}}}{P^n_0}\widetilde{\pi}(\boldsymbol{\theta})d\boldsymbol{\theta} &\le \underbrace{\int_{\mathcal{H}^c_{\epsilon_n}\cap\mathcal{F}_n} \phi\frac{P^n_{\boldsymbol{\theta}}}{P^n_0}\widetilde{\pi}(\boldsymbol{\theta})d\boldsymbol{\theta}}_{T_1}\\ &\quad + \underbrace{\int_{\mathcal{H}^c_{\epsilon_n}\cap\mathcal{F}_n} (1-\phi)\frac{P^n_{\boldsymbol{\theta}}}{P^n_0}\widetilde{\pi}(\boldsymbol{\theta})d\boldsymbol{\theta}}_{T_2}\end{aligned}$$

Next, borrowing steps from proof of theorem 3.1 in Pati et al. (2018), we have $\mathbb{E}_{P^n_0}(\phi) \le \exp(-C_1 n\epsilon_n^2)$, thus for any $C_1' < C_1$, $\phi \le \exp(-C_1' n\epsilon_n^2)$ with probability at least $1-\exp(-(C_1-C_1')n\epsilon_n^2)$. Thus,

$$T_1 \le \exp(-C_1' n\epsilon_n^2)T_1 + T_2$$

which implies with dominating probability $T_1 \le T_2$. Thus, it only remains to show $T_2 \le \exp(-C_2'(n\epsilon_n^2)/(\sum u_l))$ for some $C_2' > 0$. This is true since

$$\begin{aligned}\mathbb{P}_{P^n_0}\left(T_2 > e^{-\frac{C_2 n\epsilon_n^2}{\sum u_l}}\right) &\le e^{C_2\frac{n\epsilon_n^2}{\sum u_l}}\mathbb{E}_{P^n_0}(T_2)\\ &\le e^{\frac{C_2 n\epsilon_n^2}{\sum u_l}}\int_{\mathcal{H}^c_{\epsilon_n}\cap\mathcal{F}_n}\mathbb{E}_{P_{\boldsymbol{\theta}}}(1-\phi)\widetilde{\pi}(\boldsymbol{\theta})d\boldsymbol{\theta}\\ &\le e^{\frac{C_2 n\epsilon_n^2}{\sum u_l}}\int_{\mathcal{H}^c_{\epsilon_n}\cap\mathcal{F}_n} e^{-C_2 n d^2_{\mathrm{H}}(P_0,P_{\boldsymbol{\theta}})}\widetilde{\pi}(\boldsymbol{\theta})d\boldsymbol{\theta}\\ &\le e^{\frac{C_2 n\epsilon_n^2}{\sum u_l}}e^{-C_2 n\epsilon_n^2}\int_{\mathcal{H}^c_{\epsilon_n}\cap\mathcal{F}_n}\widetilde{\pi}(\boldsymbol{\theta})d\boldsymbol{\theta}\\ &\le \exp(-C_2' n\epsilon_n^2/\sum u_l)\end{aligned}$$

Therefore, for sufficiently large n and $C = \min(C_0/2, C_2')/2$

$$\begin{aligned}\int_{\mathcal{H}^c_{\epsilon_n}} \frac{P^n_{\boldsymbol{\theta}}}{P^n_0}\widetilde{\pi}(\boldsymbol{\theta})d\boldsymbol{\theta} &\le 2\exp(-C_2' n\epsilon_n^2/\sum u_l)\\ &\quad + \exp(-(C_0/2)n\epsilon_n^2/\sum u_l)\\ &\le \exp(-Cn\epsilon_n^2/\sum u_l) \quad \square\end{aligned}$$

Lemma A.9. *Suppose Lemma 4.3 part 1. in the main paper holds, then for any $M_n \to \infty$, with dominating probability,*

$$\log\int\frac{P^n_0}{P^n_{\boldsymbol{\theta}}}\widetilde{\pi}(\boldsymbol{\theta})d\boldsymbol{\theta} \le nM_n(\sum r_l + \xi)$$

Proof. By Markov's inequality,

$$\begin{aligned}&\mathbb{P}_{P^n_0}\left(\log\int\frac{P^n_0}{P^n_{\boldsymbol{\theta}}}\widetilde{\pi}(\boldsymbol{\theta}) \ge nM_n(\sum r_l + \xi)\right)\\ &\le \frac{1}{nM_n(\sum r_l+\xi)}\mathbb{E}_{P^n_0}\left|\log\int\frac{P^n_{\boldsymbol{\theta}}}{P^n_0}\widetilde{\pi}(\boldsymbol{\theta})d\boldsymbol{\theta}\right|\\ &= \frac{1}{nM_n(\sum r_l+\xi)}\int\left|\log\int\frac{P^n_{\boldsymbol{\theta}}}{P^n_0}\widetilde{\pi}(\boldsymbol{\theta})d\boldsymbol{\theta}\right|P^n_0 d\mu\\ &\le \frac{1}{nM_n(\sum r_l+\xi)}\left(d_{\mathrm{KL}}(P^n_0, L^*) + \frac{2}{e}\right)\end{aligned}$$

where $L^* = \int P^n_{\boldsymbol{\theta}}\widetilde{\pi}(\boldsymbol{\theta})d\boldsymbol{\theta}$ and the last inequality follows from Lemma A.3.

$$\begin{aligned}d_{\mathrm{KL}}(P^n_0, L^*) &= \mathbb{E}_{P^n_0}\left(\log\frac{P^n_0}{\int P^n_{\boldsymbol{\theta}}\widetilde{\pi}(\boldsymbol{\theta})d\boldsymbol{\theta}}\right)\\ &\le \mathbb{E}_{P^n_0}\left(\log\frac{P^n_0}{\int_{\mathcal{N}_{\sum r_l+\xi}} P^n_{\boldsymbol{\theta}}\widetilde{\pi}(\boldsymbol{\theta})d\boldsymbol{\theta}}\right)\\ &\le \int_{\mathcal{N}_{\sum r_l+\xi}}\widetilde{\pi}(\boldsymbol{\theta})d\boldsymbol{\theta}\\ &\quad + \int_{\mathcal{N}_{\sum r_l+\xi}} d_{\mathrm{KL}}(P^n_0, P^n_{\boldsymbol{\theta}})\widetilde{\pi}(\boldsymbol{\theta})d\boldsymbol{\theta} \qquad \text{Jensen's inequality}\\ &\le -\log e^{-nC(\sum r_l+\xi)} + n(\sum r_l+\xi)\\ &= n(C+1)(\sum r_l+\xi)\end{aligned}$$

where the last inequality follows from Lemma 4.3 part 1. in the main paper. The proof follows by noting $C/M_n \to 0$. □

Lemma A.10. *Suppose Lemma 4.3 part 2. in the main paper holds, then for any $M_n \to \infty$, with dominating probability,*

$$d_{\mathrm{KL}}(q,\pi) + \sum_{\boldsymbol{z}}\int\log\frac{P^n_0}{P^n_{\boldsymbol{\theta}}}q(\boldsymbol{\theta},\boldsymbol{z})d\boldsymbol{\theta} \le nM_n(\sum r_l+\xi)$$

Proof. By Markov's inequality we have

$$\begin{aligned}&\mathbb{P}_{P^n_0}\left(d_{\mathrm{KL}}(q,\pi) + \sum_{\boldsymbol{z}}\int q(\boldsymbol{\theta},\boldsymbol{z})\log\frac{P^n_0}{P^n_{\boldsymbol{\theta}}}d\boldsymbol{\theta} > nM_n(\sum r_l+\xi)\right)\\ &\le \frac{1}{nM_n(\sum r_l+\xi)}\left(d_{\mathrm{KL}}(q,\pi) + \mathbb{E}_{P^n_0}\left|\sum_{\boldsymbol{z}}\int q(\boldsymbol{\theta},\boldsymbol{z})\log\frac{P^n_0}{P^n_{\boldsymbol{\theta}}}d\boldsymbol{\theta}\right|\right)\\ &\le \frac{1}{nM_n(\sum r_l+\xi)}\left(d_{\mathrm{KL}}(q,\pi) + \mathbb{E}_{P^n_0}\left(\sum_{\boldsymbol{z}}\int q(\boldsymbol{\theta},\boldsymbol{z})\left|\log\frac{P^n_{\boldsymbol{\theta}}}{P^n_0}\right|d\boldsymbol{\theta}\right)\right)\\ &= \frac{1}{nM_n(\sum r_l+\xi)}\left(d_{\mathrm{KL}}(q,\pi) + \sum_{\boldsymbol{z}}\int q(\boldsymbol{\theta},\boldsymbol{z})\int\left|\log\frac{P^n_0}{P^n_{\boldsymbol{\theta}}}\right|P^n_0 d\mu d\boldsymbol{\theta}\right)\end{aligned}$$

By Lemma A.3, we get

$$\begin{aligned}&\le \frac{1}{nM_n(\sum r_l+\xi)}\left(d_{\mathrm{KL}}(q,\pi) + \sum_{\boldsymbol{z}}\int q(\boldsymbol{\theta},\boldsymbol{z})\left(d_{\mathrm{KL}}(P^n_0,P^n_{\boldsymbol{\theta}}) + \frac{2}{e}\right)d\boldsymbol{\theta}\right)\\ &= \frac{1}{nM_n(\sum r_l+\xi)}\left(d_{\mathrm{KL}}(q,\pi) + n\sum_{\boldsymbol{z}}\int q(\boldsymbol{\theta},\boldsymbol{z})d_{\mathrm{KL}}(P_0,P_{\boldsymbol{\theta}})d\boldsymbol{\theta} + \frac{2}{e}\right)\\ &= \frac{C}{nM_n(\sum r_l+\xi)}\left(n(\sum r_l+\xi) + (2/e)\right) \to 0\end{aligned}$$

where the last line in the above holds due to Lemma 4.3 part 2. in the main paper. □

A.3. Proof of lemmas and corollary in the main paper

Proof of Lemma 4.1. Take $s_l^\circ + 1 = (n\epsilon_n^2)/(\sum_{j=0}^L u_j)$ and $\log B_l^\circ = (n\epsilon_n^2)/((L+1)\sum_{j=0}^L(s_j^\circ+1))$.

We know from Lemma 2 of Ghosal and van der Vaart (2007) that, there exists a function $\varphi \in [0, 1]$, such that

$$\mathbb{E}_{P_0}(\varphi) \le \exp\{-nd_{\mathrm{H}}^2(P_{\boldsymbol{\theta}_1}, P_0)/2\}$$
$$\mathbb{E}_{P_{\boldsymbol{\theta}}}(1-\varphi) \le \exp\{-nd_{\mathrm{H}}^2(P_{\boldsymbol{\theta}_1}, P_0)/2\}$$

for all $P_{\boldsymbol{\theta}} \in \mathcal{F}(L, \boldsymbol{k}, \boldsymbol{s}^\circ, \boldsymbol{B}^\circ)$ satisfying $d_{\mathrm{H}}(P_{\boldsymbol{\theta}}, P_{\boldsymbol{\theta}_1}) \le d_{\mathrm{H}}(P_0, P_{\boldsymbol{\theta}_1})/18$.

Let $H = N(\epsilon_n/19, \mathcal{F}(L, \boldsymbol{k}, \boldsymbol{s}^\circ, \boldsymbol{B}^\circ), d_{\mathrm{H}}(.,.))$ denote the covering number of $\mathcal{F}(L, \boldsymbol{k}, \boldsymbol{s}^\circ, \boldsymbol{B}^\circ)$, i.e., there exist H Hellinger balls of radius $\epsilon_n/19$, that entirely cover $\mathcal{F}(L, \boldsymbol{k}, \boldsymbol{s}^\circ, \boldsymbol{B}^\circ)$. For any $\boldsymbol{\theta} \in \mathcal{F}(L, \boldsymbol{k}, \boldsymbol{s}^\circ, \boldsymbol{B}^\circ)$ w.l.o.g we assume $P_{\boldsymbol{\theta}}$ belongs to the Hellinger ball centered at $P_{\boldsymbol{\theta}_h}$ and if $d_{\mathrm{H}}(P_{\boldsymbol{\theta}}, P_0) > \epsilon_n$, then we must have that $d_{\mathrm{H}}(P_0, P_{\boldsymbol{\theta}_h}) > (18/19)\epsilon_n$ and there exists a testing function φ_h, such that

$$\begin{aligned}\mathbb{E}_{P_0}(\varphi_h) &\le \exp\{-nd_{\mathrm{H}}^2(P_{\boldsymbol{\theta}_h}, P_0)/2\}\\ &\le \exp\{-((18^2/19^2)/2)n\epsilon_n^2\}\\ \mathbb{E}_{P_{\boldsymbol{\theta}}}(1-\varphi_h) &\le \exp\{-nd_{\mathrm{H}}^2(P_{\boldsymbol{\theta}_h}, P_0)/2\}\\ &\le \exp\{-n(d_{\mathrm{H}}(P_0, P_{\boldsymbol{\theta}}) - \epsilon_n/19)^2/2\}\\ &\le \exp\{-((18^2/19^2)/2)nd_{\mathrm{H}}^2(P_0, P_{\boldsymbol{\theta}})\}.\end{aligned}$$

Next we define $\phi = \max_{h=1,\dots,H} \varphi_h$. Then we must have

$$\begin{aligned}\mathbb{E}_{P_0}(\phi) &\le \sum_h \mathbb{E}_{P_0}(\varphi_h) \le H\exp\{-((18^2/19^2)/2)n\epsilon_n^2\}\\ &\le \exp\{-((18^2/19^2)/2)n\epsilon_n^2 - \log H\}\end{aligned}$$

Using Lemma A.6 with $\boldsymbol{s} = \boldsymbol{s}^\circ$ and $\boldsymbol{B} = \boldsymbol{B}^\circ$, we get

$$\begin{aligned}&\log H = \log N(\epsilon_n/19, \mathcal{F}(L, \boldsymbol{k}, \boldsymbol{s}^\circ, \boldsymbol{B}^\circ), d_{\mathrm{H}}(.,.))\\ &\le \log N(\sqrt{8}\sigma_e^2\epsilon_n/19, \mathcal{F}(L, \boldsymbol{k}, \boldsymbol{s}^\circ, \boldsymbol{B}^\circ), \|.\|_\infty)\\ &\le \log\left[\prod_{l=0}^{L}\left(\frac{38}{\sqrt{8}\sigma_e^2\epsilon_n}(L+1)\left(\prod_{j=0}^{L}B_j^\circ\right)k_{l+1}\right)^{(s_l^\circ+1)}\right]\\ &= \sum_{l=0}^{L}(s_l^\circ+1)\log\left(\frac{38}{\sqrt{8}\sigma_e^2\epsilon_n}(L+1)\left(\prod_{j=0}^{L}B_j^\circ\right)k_{l+1}\right)\\ &\le C\left[\sum_{l=0}^{L}(s_l^\circ+1)\left(\log\frac{1}{\epsilon_n} + \log(L+1) + \sum_{j=0}^{L}\log B_j^\circ + \log k_{l+1}\right)\right]\\ &\le C\sum_{l=0}^{L}(s_l^\circ+1)(\log n + \log(L+1) + \sum_{j=0}^{L}\log B_j^\circ + \log k_{l+1})\\ &\le C\sum_{l=0}^{L}(s_l^\circ+1)(\log n + \log(L+1)\\ &\quad + \sum_{j=0}^{L}\log B_j^\circ + \log k_{l+1} + \log(k_l+1)) \le Cn\epsilon_n^2\end{aligned}$$

where, C in each step is different which tends to absorb the extra constants in it. First inequality holds due to the following

$$d_{\mathrm{H}}^2(P_{\boldsymbol{\theta}}, P_0) \le 1 - \exp\left\{-\frac{1}{8\sigma_e^2}\|\eta_0 - \eta_{\boldsymbol{\theta}}\|_\infty^2\right\}$$

and $\epsilon_n = o(1)$, the second inequality is due to (17), and fourth inequality is due to $s_l^\circ\log(1/\epsilon_n) \asymp s_l^\circ\log n$. Therefore,

$$\mathbb{E}_{P_0}(\phi) \le \sum_h \mathbb{E}_{P_0}(\varphi_h) = \exp\{-C_1n\epsilon_n^2\}$$

for some $C_1 = (18^2/19^2)/2 - 1/4$. On the other hand, for any $\boldsymbol{\theta}$, such that $d_{\mathrm{H}}(P_{\boldsymbol{\theta}}, P_0) \ge \epsilon_n$, say $P_{\boldsymbol{\theta}}$ belongs to the hth Hellinger ball, then we have

$$\mathbb{E}_{P_{\boldsymbol{\theta}}}(1-\phi) \le \mathbb{E}_{P_{\boldsymbol{\theta}}}(1-\varphi_h) \le \exp\{-C_2nd_{\mathrm{H}}^2(P_0, P_{\boldsymbol{\theta}})\}$$

where $C_2 = (18^2/19^2)/2$. This concludes the proof. □

Proof of Lemma 4.2.

$$\begin{aligned}\textit{Assumption}: \quad & s_l^\circ + 1 = (n\epsilon_n^2)/(\sum_{j=0}^{L}u_j),\ \lambda_lk_{l+1}/s_l^\circ \to 0,\\ & \sum u_l\log L = o(n\epsilon_n^2)\end{aligned} \tag{19}$$

$$\begin{aligned}&\widetilde{\Pi}(\mathcal{F}(L, \boldsymbol{k}, \boldsymbol{s}^\circ, \boldsymbol{B}^\circ)^c)\\ &\le \widetilde{\Pi}\left(\bigcup_{l=0}^{L}\{\|\widetilde{\boldsymbol{w}}_l\|_0 > s_l^\circ\}\right) + \widetilde{\Pi}\left(\bigcup_{l=0}^{L}\{\|\widetilde{\boldsymbol{w}}_l\|_\infty > B_l^\circ\}\right)\\ &\le \sum_{l=0}^{L}\widetilde{\Pi}(\|\widetilde{\boldsymbol{w}}_l\|_0 > s_l^\circ) + \sum_{l=0}^{L}\widetilde{\Pi}(\|\widetilde{\boldsymbol{w}}_l\|_\infty > B_l^\circ)\\ &= \sum_{l=0}^{L}\sum_{\boldsymbol{z}}\Pi(\|\widetilde{\boldsymbol{w}}_l\|_0 > s_l^\circ|\boldsymbol{z})\pi(\boldsymbol{z}) + \sum_{l=0}^{L}\sum_{\boldsymbol{z}}\Pi(\|\widetilde{\boldsymbol{w}}_l\|_\infty > B_l^\circ|\boldsymbol{z})\pi(\boldsymbol{z})\\ &\le \sum_{l=0}^{L}\mathbb{P}\left(\sum_{i=1}^{k_{l+1}}z_{li} > s_l^\circ\right) + \sum_{l=0}^{L}\mathbb{P}\left(\sup_{i=1,\dots,k_{l+1}}\|\overline{\boldsymbol{w}}_{li}\|_1 > B_l^\circ\middle|\boldsymbol{z}\right)\end{aligned}$$

where $\widetilde{\boldsymbol{w}}_l = (\|\overline{\boldsymbol{w}}_{l1}\|_1, \dots, \|\overline{\boldsymbol{w}}_{lk_{l+1}}\|_1)^T$ and the last inequality holds since $\Pi(\|\widetilde{\boldsymbol{w}}_l\|_0 > s_l^\circ|\boldsymbol{z}) \le 1$, $\Pi(\|\widetilde{\boldsymbol{w}}_l\|_0 > s_l^\circ|\boldsymbol{z}) = 1$ iff $\sum z_{li} > \boldsymbol{s}_l^\circ$ and $\pi(\boldsymbol{z}) \le 1$. We will now break the proof in two parts as follows.

Part 1.

$$\sum_{l=0}^{L}\mathbb{P}\left(\sum_{i=1}^{k_{l+1}}z_{li} > s_l^\circ\right) = \sum_{l=0}^{L}\mathbb{P}\left(\sum_{i=1}^{k_{l+1}}z_{li} - k_{l+1}\lambda_l > s_l^\circ - k_{l+1}\lambda_l\right)$$

By Bernstein inequality

$$\begin{aligned}&\le \sum_{l=0}^{L}\exp\left(\frac{-1/2(s_l^\circ - k_{l+1}\lambda_l)^2}{k_{l+1}\lambda_l(1-\lambda_l) + 1/3(s_l^\circ - k_{l+1}\lambda_l)}\right)\\ &\le \sum_{l=0}^{L}\exp\left(\frac{-1/2(s_l^\circ - k_{l+1}\lambda_l)^2}{k_{l+1}\lambda_l + 1/3(s_l^\circ - k_{l+1}\lambda_l)}\right)\\ &= \sum_{l=0}^{L}\exp\left(\frac{-s_l^\circ/2(1 - k_{l+1}\lambda_l/s_l^\circ)^2}{1/3(1 + 2k_{l+1}\lambda_l/s_l^\circ)}\right) \to \sum_{l=0}^{L}\exp\left(-\frac{3s_l^\circ}{2}\right)\\ &\qquad \text{since } \frac{k_{l+1}\lambda_l}{s_l^\circ} \to 0 \text{ by (19)}\\ &= \sum_{l=0}^{L}\exp\left(-\frac{3n\epsilon_n^2}{4\sum u_l} + \frac{3}{2}\right) \le 5(L+1)\exp\left(-\frac{n\epsilon_n^2}{2\sum u_l}\right)\\ &\le \exp\left(-\frac{n\epsilon_n^2}{4\sum u_l}\right)\end{aligned}$$

since $\sum u_l\log(5(L+1)) \sim \sum u_l\log L = o(n\epsilon_n^2)$ by (19).

Part 2.

$$\begin{aligned}&\sum_{l=0}^{L}\mathbb{P}\left(\sup_{i=1,\dots,k_{l+1}}\|\boldsymbol{w}_{li}\|_1 > B_l^\circ\middle|\boldsymbol{z}\right)\\ &\le \sum_{l=0}^{L}\sum_{i=1}^{k_{l+1}}\mathbb{P}\left(\|\boldsymbol{w}_{li}\|_1 > B_l^\circ\middle|\boldsymbol{z}\right)\\ &\le \sum_{l=0}^{L}\sum_{i=1}^{k_{l+1}}\mathbb{P}\left(\|\boldsymbol{w}_{li}\|_\infty > \frac{B_l^\circ}{k_l+1}\middle|\boldsymbol{z}\right)\end{aligned}$$

$$
\begin{aligned}
&\le \sum_{l=0}^{L}\sum_{i=1}^{k_{l+1}}\sum_{j=1}^{k_l+1}\mathbb{P}\left(|w_{lij}| > \frac{B_l^\circ}{k_l+1}\middle|\boldsymbol{z}\right)\\
&\le 2\sum_{l=0}^{L}\sum_{i=1}^{k_{l+1}}\sum_{j=1}^{k_l+1}\exp\left(-\frac{B_l^{\circ 2}}{(k_l+1)^2}\right)
\end{aligned}
$$

By concentration inequality

$$
\begin{aligned}
&= 2\sum_{l=0}^{L}\sum_{i=1}^{k_{l+1}}\sum_{j=1}^{k_l+1}\exp\Bigg(-\exp\Bigg(\frac{2n\epsilon_n^2}{(L+1)\sum_{j'=0}^{L}(s_{j'}^\circ+1)}\\
&\quad - 2\log(k_l+1)\Bigg)\Bigg)\\
&\le \sum_{l=0}^{L}\sum_{i=1}^{k_{l+1}}\sum_{j=1}^{k_l+1}\frac{1}{(L+1)k_{l+1}(k_l+1)}\exp(-n\epsilon_n^2) = \exp(-n\epsilon_n^2)
\end{aligned}
$$

where the third inequality holds since $|w_{lij}|$ given $\boldsymbol{z}$ is bound above by a $|N(0,\sigma_0^2)|$ random variable. The above proof holds as long as

$$
\begin{aligned}
&\exp\left(\frac{2n\epsilon_n^2}{(L+1)\sum_{j'=0}^{L}(s_{j'}^\circ+1)} - 2\log(k_l+1)\right)\\
&\quad \ge n\epsilon_n^2 + \log(L+1) + \log k_{l+1} + \log(k_l+1) + \log 2
\end{aligned}
$$

Taking log on both sides we get

$$
\begin{aligned}
&\left(\frac{n\epsilon_n^2}{(L+1)\sum_{j'=0}^{L}(s_{j'}^\circ+1)} - \log(k_l+1)\right)\\
&\quad \ge \frac{1}{2}\log(n\epsilon_n^2 + \log(L+1) + \log k_{l+1} + \log(k_l+1) + \log 2)
\end{aligned}
$$

This is true since $\sum_{j'=0}^{L}(s_{j'}^\circ+1) = (L+1)n\epsilon_n^2/\sum u_l$ is bounded above by

$$
\frac{n\epsilon_n^2}{(L+1)(\log(k_l+1) + \frac{1}{2}\log(n\epsilon_n^2 + \log(L+1) + \log k_{l+1} + \log(k_l+1) + \log 2))}
$$

□

Proof of Lemma 4.3 part 1.

$$
\begin{aligned}
\textit{Assumption :}\quad & -\log\lambda_l = O\{(k_l+1)\vartheta_l\},\\
& -\log(1-\lambda_l) = O\{(s_l/k_{l+1})(k_l+1)\vartheta_l\} \qquad (20)
\end{aligned}
$$

$$
\begin{aligned}
d_{\mathrm{KL}}(P_0,P_{\boldsymbol{\theta}}) &= \int_{\boldsymbol{x}\in[0,1]^p}\int_{y\in R}\left(\log\frac{P_0(y,\boldsymbol{x})}{P_{\boldsymbol{\theta}}(y,\boldsymbol{x})}\right)P_0(y,\boldsymbol{x})dyd\boldsymbol{x}\\
P_0(y,\boldsymbol{x}) &= \frac{1}{\sqrt{2\pi\sigma_e^2}}\exp\left(-\frac{(y-\eta_0(\boldsymbol{x}))^2}{2\sigma_e^2}\right)\\
P_{\boldsymbol{\theta}}(y,\boldsymbol{x}) &= \frac{1}{\sqrt{2\pi\sigma_e^2}}\exp\left(-\frac{(y-\eta_{\boldsymbol{\theta}}(\boldsymbol{x}))^2}{2\sigma_e^2}\right)
\end{aligned}
$$

So we get,

$$
\begin{aligned}
&d_{\mathrm{KL}}(P_0,P_{\boldsymbol{\theta}})\\
&= \int_{\boldsymbol{x}\in[0,1]^p}\int_{y\in\mathbb{R}}\log\left(\exp\left[-\frac{(y-\eta_0(\boldsymbol{x}))^2}{2\sigma_e^2}+\frac{(y-\eta_{\boldsymbol{\theta}}(\boldsymbol{x}))^2}{2\sigma_e^2}\right]\right)P_0(y,\boldsymbol{x})dyd\boldsymbol{x}\\
&= \int_{\boldsymbol{x}\in[0,1]^p}\int_{y\in\mathbb{R}}\frac{2y(\eta_0(\boldsymbol{x})-\eta_{\boldsymbol{\theta}}(\boldsymbol{x}))-(\eta_0^2(\boldsymbol{x})-\eta_{\boldsymbol{\theta}}^2(\boldsymbol{x}))}{2\sigma_e^2}P_0(y,\boldsymbol{x})dyd\boldsymbol{x}\\
&= \int_{\boldsymbol{x}\in[0,1]^p}\frac{2\eta_0^2(\boldsymbol{x})-2\eta_0(\boldsymbol{x})\eta_{\boldsymbol{\theta}}(\boldsymbol{x})-\eta_0^2(\boldsymbol{x})+\eta_{\boldsymbol{\theta}}^2(\boldsymbol{x})}{2\sigma_e^2}d\boldsymbol{x}\\
&= \int_{\boldsymbol{x}\in[0,1]^p}\frac{(\eta_0(\boldsymbol{x})-\eta_{\boldsymbol{\theta}}(\boldsymbol{x}))^2}{2}d\boldsymbol{x} = \frac{1}{2}\|\eta_0-\eta_{\boldsymbol{\theta}}\|_2^2 \qquad (21)
\end{aligned}
$$

where, $\sigma_e^2 = 1$ can be chosen w.l.o.g. Next, let $\eta_{\boldsymbol{\theta}^*}(\boldsymbol{x})$ be $\boldsymbol{\theta}^*$ satisfying $\arg\min_{\eta_{\boldsymbol{\theta}}\in\mathcal{F}(L,\boldsymbol{k},\boldsymbol{s},\boldsymbol{B})}\|\eta_{\boldsymbol{\theta}}-\eta_0\|_\infty^2$. Then,

$$
\|\eta_{\boldsymbol{\theta}^*}-\eta_0\|_1 \le \|\eta_{\boldsymbol{\theta}^*}-\eta_0\|_\infty = \sqrt{\xi} \qquad (22)
$$

Here, we redefine $\overline{\boldsymbol{\delta}}_l$ by considering the L_1 norms of the rows of $\overline{\boldsymbol{D}}_l = \overline{\boldsymbol{W}}_l - \overline{\boldsymbol{W}}_l^*$ as follows

$$
\overline{\boldsymbol{D}}_l = (\overline{\boldsymbol{d}}_{l1}^\top,\ldots,\overline{\boldsymbol{d}}_{lk_{l+1}}^\top)^\top \qquad \overline{\boldsymbol{\delta}}_l = (\|\overline{\boldsymbol{d}}_{l1}\|_1,\ldots,\|\overline{\boldsymbol{d}}_{lk_{l+1}}\|_1)
$$

Next we define a neighborhood $\mathcal{M}_{\sqrt{\sum r_l}}$ as follows:

$$
\begin{aligned}
\mathcal{M}_{\sqrt{\sum r_l}} = \Bigg\{\boldsymbol{\theta} : \|\overline{\boldsymbol{d}}_{li}\|_1 \le \frac{\sqrt{\sum r_l B_l}}{(L+1)(\prod_{j=0}^{L}B_j)},\\
i\in\mathcal{S}_l,\ \|\overline{\boldsymbol{d}}_{li}\|_1 = 0,\ i\in\mathcal{S}_l^c,\ l=0,\ldots,L\Bigg\}
\end{aligned}
$$

where $\mathcal{S}_l^c$ is the set where $\|\overline{\boldsymbol{w}}_{li}^*\|_1 = 0$, $l = 0,\ldots,L$. Then, for every $\boldsymbol{\theta}\in\mathcal{M}_{\sqrt{\sum r_l}}$ using (16), we have

$$
\|\eta_{\boldsymbol{\theta}}-\eta_{\boldsymbol{\theta}^*}\|_1 \le \sqrt{\sum r_l} \qquad (23)
$$

Combining (22) and (23), we get for $\boldsymbol{\theta}\in\mathcal{M}_{\sqrt{\sum r_l}}$, $\|\eta_{\boldsymbol{\theta}}-\eta_0\|_1 \le \sqrt{\sum r_l}+\sqrt{\xi}$. So we get,

$$
d_{\mathrm{KL}}(P_0,P_{\boldsymbol{\theta}}) \le \frac{(\sqrt{\sum r_l}+\sqrt{\xi})^2}{2} \le \sum r_l + \xi
$$

Since $\boldsymbol{\theta}\in\mathcal{N}_{\sum r_l+\xi}$ for every $\boldsymbol{\theta}\in\mathcal{M}_{\sqrt{\sum r_l}}$; therefore,

$$
\int_{\boldsymbol{\theta}\in\mathcal{N}_{\sum r_l+\xi}}\widetilde{\pi}(\boldsymbol{\theta})d\boldsymbol{\theta} \ge \int_{\boldsymbol{\theta}\in\mathcal{M}_{\sqrt{\sum r_l}}}\widetilde{\pi}(\boldsymbol{\theta})d\boldsymbol{\theta}
$$

Let $\delta_n = (\sqrt{\sum r_l}B_l)/((L+1)(\prod_{j=0}^{L}B_j))$ and $A = \{\overline{\boldsymbol{w}}_{li} : \quad \|\overline{\boldsymbol{w}}_{li} - \overline{\boldsymbol{w}}_{li}^*\|_1 \le \delta_n\}$

$$
\begin{aligned}
&\widetilde{\Pi}\left(\mathcal{M}_{\sqrt{\sum r_l}}\right)\\
&= \sum_{\boldsymbol{z}}\Pi\left(\mathcal{M}_{\sqrt{\sum r_l}}\middle|\boldsymbol{z}\right)\pi(\boldsymbol{z})\\
&\ge \sum_{\{\boldsymbol{z}:z_{li}=1,i\in\mathcal{S}_l,z_{li}=0,i\in\mathcal{S}_l^c,l=0,\ldots,L\}}\Pi\left(\mathcal{M}_{\sqrt{\sum r_l}}\middle|\boldsymbol{z}\right)\pi(\boldsymbol{z})\\
&= \prod_{l=0}^{L}(1-\lambda_l)^{k_{l+1}-s_l}\lambda_l^{s_l}\prod_{i\in S_l}\mathbb{E}(\mathbb{1}_{\{\overline{\boldsymbol{w}}_{li}\in A\}}|z_{li}=1)\\
&\ge \prod_{l=0}^{L}(1-\lambda_l)^{k_{l+1}-s_l}\lambda_l^{s_l}\prod_{i\in\mathcal{S}_l}\int_{\overline{\boldsymbol{w}}_{li}\in A}\left(\frac{1}{2\pi}\right)^{\frac{k_l+1}{2}}\prod_{j=1}^{k_l+1}\exp\left(-\frac{\overline{w}_{lij}^2}{2}\right)d\overline{w}_{lij}\\
&\ge \prod_{l=0}^{L}(1-\lambda_l)^{k_{l+1}-s_l}\lambda_l^{s_l}\prod_{i\in\mathcal{S}_l}\left(\frac{1}{2\pi}\right)^{\frac{k_l+1}{2}}\\
&\quad\times\prod_{j=1}^{k_l+1}\int_{\overline{w}_{lij}^*-\frac{\delta_n}{k_l+1}}^{\overline{w}_{lij}^*+\frac{\delta_n}{k_l+1}}\exp\left(-\frac{\overline{w}_{lij}^2}{2}\right)d\overline{w}_{lij}\\
&= \prod_{l=0}^{L}(1-\lambda_l)^{k_{l+1}-s_l}\lambda_l^{s_l}\prod_{i\in\mathcal{S}_l}\left(\frac{1}{2\pi}\right)^{\frac{k_l+1}{2}}\prod_{j=1}^{k_l+1}\frac{2\delta_n}{k_l+1}\exp\left(-\frac{\widehat{w}_{lij}^2}{2}\right)
\end{aligned}
$$

where the third equality follows since $\mathbb{E}(\mathbb{1}_{\{\overline{\boldsymbol{w}}_{li}\in A\}}|z_{li}=0)=1$ since $\|\overline{\boldsymbol{w}}^*_{li}\|_1=0$, for $i\in\mathcal{S}_l^c$. The last equality is by mean value theorem, $\widehat{w}_{lij}\in[\overline{w}^*_{lij}-\delta_n/(k_l+1),\overline{w}^*_{lij}+\delta_n/(k_l+1)]$, thus

$$
\begin{aligned}
&=\prod_{l=0}^{L}(1-\lambda_l)^{k_{l+1}-s_l}\lambda_l^{s_l}\prod_{i\in\mathcal{S}_l}\exp\Bigg(\frac{k_l+1}{2}\log\frac{1}{2\pi}\\
&\quad+(k_l+1)\log\frac{2\delta_n}{k_l+1}-\sum_{j=1}^{k_l+1}\frac{\widehat{w}^2_{lij}}{2}\Bigg)\\
&=\exp\Bigg[-\sum_{l=0}^{L}\Bigg\{s_l\log\left(\frac{1}{\lambda_l}\right)+(k_{l+1}-s_l)\log\left(\frac{1}{1-\lambda_l}\right)\\
&\qquad+\sum_{i\in\mathcal{S}_l}\Bigg(-\frac{k_l+1}{2}\log\frac{1}{2\pi}\\
&\qquad-(k_l+1)\log\frac{2\delta_n}{k_l+1}+\sum_{j=1}^{k_l+1}\frac{\widehat{w}^2_{lij}}{2}\Bigg)\Bigg\}\Bigg]\\
&=\exp\Bigg[-\sum_{l=0}^{L}\Bigg\{s_l\log\left(\frac{1}{\lambda_l}\right)+(k_{l+1}-s_l)\log\left(\frac{1}{1-\lambda_l}\right)\\
&\qquad-\frac{s_l(k_l+1)}{2}\log\frac{1}{2\pi}-s_l(k_l+1)\log\frac{2\delta_n}{k_l+1}\\
&\qquad+\sum_{i\in\mathcal{S}_l}\sum_{j=1}^{k_l+1}\frac{\widehat{w}^2_{lij}}{2}\Bigg\}\Bigg]
\end{aligned}\tag{24}
$$

Now,

$$
\begin{aligned}
&\sum_{l=0}^{L}\sum_{i\in\mathcal{S}_l}\sum_{j=1}^{k_l+1}\frac{\widehat{w}^2_{lij}}{2}\\
&\le\frac{1}{2}\sum_{l=0}^{L}\sum_{i\in\mathcal{S}_l}\sum_{j=1}^{k_l+1}\max((\overline{w}^*_{lij}-\delta_n/(k_l+1))^2,(\overline{w}^*_{lij}+\delta_n/(k_l+1))^2)\\
&\le\sum_{l=0}^{L}\sum_{i\in\mathcal{S}_l}\sum_{j=1}^{k_l+1}(\overline{w}^{*2}_{lij}+\delta_n^2/(k_l+1)^2)\\
&\le\sum_{l=0}^{L}\sum_{i\in\mathcal{S}_l}\|\overline{\boldsymbol{w}}^*_{li}\|_1^2+\sum_{l=0}^{L}\sum_{i\in\mathcal{S}_l}\delta_n^2/(k_l+1)\\
&\le\sum_{l=0}^{L}s_l(B_l^2+1)\le n\sum r_l\le n\left(\sum r_l+\xi\right)
\end{aligned}\tag{25}
$$

where the above line uses $\delta_n\to 0$. Finally

$$
\begin{aligned}
&\sum_{l=0}^{L}\Bigg(s_l\log\left(\frac{1}{\lambda_l}\right)+(k_{l+1}-s_l)\log\left(\frac{1}{1-\lambda_l}\right)\\
&\quad-\frac{s_l(k_l+1)}{2}\log\frac{1}{2\pi}-s_l(k_l+1)\log\frac{2\delta_n}{k_l+1}\Bigg)\\
&\le\sum_{l=0}^{L}\Bigg(Cnr_l+\frac{s_l(k_l+1)}{2}\Bigg\{2\log(k_l+1)+2\log(L+1)\\
&\quad+2\sum_{m=0,m\neq l}^{L}\log B_m-\log\sum r_l\Bigg\}\Bigg)\\
&\le Cn\sum r_l\le Cn\left(\sum r_l+\xi\right)
\end{aligned}\tag{26}
$$

where the first inequality follows from and expanding δ_n. The last inequality follows since $n\sum r_l\to\infty$ which implies $-\log\sum r_l=O(\log n)$. Combining (25) and (26) and replacing (24), the proof follows. □

Proof of Lemma 4.3 part 2.

$$
\begin{aligned}
\textit{Assumption}:\quad&-\log\lambda_l=O\{(k_l+1)\vartheta_l\},\\
&-\log(1-\lambda_l)=O\{(s_l/k_{l+1})(k_l+1)\vartheta_l\}
\end{aligned}
$$

Suppose there exists $q\in\mathcal{Q}^{\mathbf{MF}}$ such that

$$
\begin{aligned}
&d_{\mathrm{KL}}(q,\pi)\le C_1n\sum r_l,\\
&\sum_{\boldsymbol{z}}\int_{\Theta}\|\eta_{\boldsymbol{\theta}}-\eta_{\boldsymbol{\theta}^*}\|_2^2\,q(\boldsymbol{\theta},\boldsymbol{z})d\boldsymbol{\theta}\le\sum r_l.
\end{aligned}\tag{27}
$$

Recall $\boldsymbol{\theta}^*=\arg\min_{\boldsymbol{\theta}\in\boldsymbol{\theta}(L,p,\boldsymbol{s},\boldsymbol{B})}\|\eta_{\boldsymbol{\theta}}-\eta_0\|_\infty^2$. By relation (21),

$$
\begin{aligned}
\sum_{\boldsymbol{z}}\int nd_{\mathrm{KL}}(P_0,P_{\boldsymbol{\theta}})q(\boldsymbol{\theta},\boldsymbol{z})d\boldsymbol{\theta}&=\sum_{\boldsymbol{z}}\frac{n}{2}\int\|\eta_0-\eta_{\boldsymbol{\theta}}\|_2^2q(\boldsymbol{\theta},\boldsymbol{z})d\boldsymbol{\theta}\\
&\le\frac{n}{2}\sum_{\boldsymbol{z}}\int\|\eta_{\boldsymbol{\theta}^*}-\eta_{\boldsymbol{\theta}}\|_2^2q(\boldsymbol{\theta},\boldsymbol{z})d\boldsymbol{\theta}\\
&\quad+\frac{n}{2}\|\eta_{\boldsymbol{\theta}^*}-\eta_0\|_\infty^2\\
&\le Cn(\sum r_l+\xi)
\end{aligned}
$$

where the above relation is due to (27) which will complete the proof.

We next construct $q\in\mathcal{Q}^{\mathbf{MF}}$ as

$$
\begin{aligned}
&\overline{w}_{lij}|z_{li}\sim z_{li}\mathcal{N}(\overline{w}^*_{lij},\sigma_l^2)+(1-z_{li})\delta_0,\\
&\quad z_{li}\sim\mathrm{Bern}(\gamma^*_{li})\qquad\gamma^*_{li}=\mathbb{1}(\|\boldsymbol{w}^*_{li}\|_1\neq 0)
\end{aligned}
$$

where $\sigma_l^2=\frac{s_l}{8n(L+1)}(4^{L-l}(k_l+1)\log(k_{l+1}2^{k_l+1})\prod_{m=0,m\neq l}^{L}B_m^2)^{-1}$.

We next consider the relation (18) in Lemma A.7.

We upper bound the expectation of the supremum of L_1 norm of multivariate Gaussian variables:

$$
\begin{aligned}
\int\widetilde{W}_lq(\boldsymbol{\theta},\boldsymbol{z})d\boldsymbol{\theta}&\le\int\sup_i\|\overline{\boldsymbol{w}}_{li}-\overline{\boldsymbol{w}}^*_{li}\|_1q(\boldsymbol{\theta}|\boldsymbol{z})d\boldsymbol{\theta}\\
&\le\int\sup_i\|\overline{\boldsymbol{w}}_{li}-\overline{\boldsymbol{w}}^*_{li}\|_1q(\boldsymbol{\theta}|\boldsymbol{z}=\mathbf{1})d\boldsymbol{\theta}
\end{aligned}
$$

since $q(\boldsymbol{z})\le 1$. If $z_{li}=1$, then $\|\overline{\boldsymbol{w}}_{li}-\overline{\boldsymbol{w}}^*_{li}\|_1=0$, thus the above integral is maximized at $\boldsymbol{z}=\mathbf{1}$ where $\boldsymbol{z}=\mathbf{1}$ indicates all neurons are present in the network. In this case, all w_{lij} are nothing but independent Gaussian random variables. In this direction we make use of concentration inequalities similar to the proof of theorem 2 in Chérief-Abdellatif (2020). Let, $Y=\sup_i\|\overline{\boldsymbol{w}}_{li}-\overline{\boldsymbol{w}}^*_{li}\|_1$.

$$
\begin{aligned}
\exp(t\mathbb{E}Y)\le\mathbb{E}(\exp(tY))&=\mathbb{E}[\sup_i\exp(t\|\overline{\boldsymbol{w}}_{li}-\overline{\boldsymbol{w}}^*_{li}\|_1)]\\
&\le\sum_{i=1}^{k_{l+1}}\mathbb{E}[\exp(t\sum_{j=1}^{k_l+1}|\overline{w}_{lij}-\overline{w}^*_{lij}|)]\\
&=\sum_{i=1}^{k_{l+1}}\prod_{j=1}^{k_l+1}\mathbb{E}[\exp(t|\overline{w}_{lij}-\overline{w}^*_{lij}|)]\\
&=\sum_{i=1}^{k_{l+1}}\prod_{j=1}^{k_l+1}2\exp\left[\frac{\sigma_l^2t^2}{2}\right]\Phi(\sigma_lt)\\
&\le k_{l+1}2^{k_l+1}\exp\left[(k_l+1)\frac{\sigma_l^2t^2}{2}\right]
\end{aligned}
$$

Thus, $\mathbb{E}Y \le (\log(k_{l+1}2^{k_l+1}) + (k_l+1)\sigma_l^2 t^2/2)/t$. Let $t = (1/\sigma_l)\sqrt{(2/(k_l+1))\log(k_{l+1}2^{k_l+1})}$,

$$\mathbb{E}Y \le \sigma_l\sqrt{\frac{k_l+1}{2}}\left[\sqrt{\log(k_{l+1}2^{k_l+1})} + \sqrt{\log(k_{l+1}2^{k_l+1})}\right]$$
$$= \sqrt{2\sigma_l^2(k_l+1)\log(k_{l+1}2^{k_l+1})} \le \sqrt{4\sigma_l^2(k_l+1)\log(k_{l+1}2^{k_l+1})}$$

Similarly,

$$\int \widetilde{W}_l^2 q(\boldsymbol{\theta},\boldsymbol{z})d\boldsymbol{\theta} = \int \sup_i(\|\overline{\boldsymbol{w}}_{li} - \overline{\boldsymbol{w}}_{li}^*\|_1)^2 q(\boldsymbol{\theta},\boldsymbol{z})d\boldsymbol{\theta}$$
$$\le \int \sup_i(\|\overline{\boldsymbol{w}}_{li} - \overline{\boldsymbol{w}}_{li}^*\|_1)^2 q(\boldsymbol{\theta}|\boldsymbol{z}=\boldsymbol{1})$$

Let, $Y' = \sup_i(\|\overline{\boldsymbol{w}}_{li} - \overline{\boldsymbol{w}}_{li}^*\|_1)^2$.

$$\exp(t\mathbb{E}Y') \le \mathbb{E}(\exp(tY')) = \mathbb{E}[\sup_i \exp(t(\|\overline{\boldsymbol{w}}_{li} - \overline{\boldsymbol{w}}_{li}^*\|_1)^2)]$$
$$\le \sum_{i=1}^{k_{l+1}} \mathbb{E}[\exp(t(\sum_{j=1}^{k_l+1}|\overline{w}_{lij} - \overline{w}_{lij}^*|)^2)]$$
$$\le \sum_{i=1}^{k_{l+1}} \mathbb{E}[\exp(t(k_l+1)\sum_{j=1}^{k_l+1}(\overline{w}_{lij} - \overline{w}_{lij}^*)^2)]$$
$$= \sum_{i=1}^{k_{l+1}}\prod_{j=1}^{k_l+1} \mathbb{E}[\exp(t(k_l+1)(\overline{w}_{lij} - \overline{w}_{lij}^*)^2)]$$
$$= \sum_{i=1}^{k_{l+1}}\prod_{j=1}^{k_l+1}\left(\frac{1}{1-2t(k_l+1)\sigma_l^2}\right)^{\frac{1}{2}}$$
$$\le k_{l+1}\left(\frac{1}{1-2t(k_l+1)\sigma_l^2}\right)^{\frac{k_l+1}{2}}$$

Thus, $\mathbb{E}Y' \le (\log k_{l+1} - ((k_l+1)/2)\log(1-2t(k_l+1)\sigma_l^2))/t$. Let $t = 1/(4\sigma_l^2(k_l+1))$,

$$\mathbb{E}Y' \le 4\sigma_l^2(k_l+1)\left[\log k_{l+1} + \left(\frac{k_l+1}{2}\right)\log 2\right]$$
$$= 4\sigma_l^2(k_l+1)\log(k_{l+1}2^{\frac{k_l+1}{2}})$$
$$\le 4\sigma_l^2(k_l+1)\log(k_{l+1}2^{k_l+1})$$

Next we also get,

$$\int(\widetilde{W}_l + B_l)q(\boldsymbol{\theta},\boldsymbol{z})d\boldsymbol{\theta} = \int \widetilde{W}_l q(\boldsymbol{\theta},\boldsymbol{z})d\boldsymbol{\theta} + B_l$$
$$\le \sqrt{4\sigma_l^2(k_l+1)\log(k_{l+1}2^{k_l+1})} + B_l \le 2B_l$$

$$\int(\widetilde{W}_l + B_l)^2 q(\boldsymbol{\theta},\boldsymbol{z})d\boldsymbol{\theta} = \int \widetilde{W}_l^2 q(\boldsymbol{\theta},\boldsymbol{z})d\boldsymbol{\theta} + 2B_l\int \widetilde{W}_l q(\boldsymbol{\theta},\boldsymbol{z})d\boldsymbol{\theta} + B_l^2$$
$$\le 4\sigma_l^2(k_l+1)\log(k_{l+1}2^{k_l+1})$$
$$+\ 2B_l\sqrt{4\sigma_l^2(k_l+1)\log(k_{l+1}2^{k_l+1})} + B_l^2 \le 4B_l^2$$

$$\int \widetilde{W}_l(\widetilde{W}_l + B_l)q(\boldsymbol{\theta},\boldsymbol{z})d\boldsymbol{\theta} = \int \widetilde{W}_l^2 q(\boldsymbol{\theta},\boldsymbol{z})d\boldsymbol{\theta} + B_l\int \widetilde{W}_l q(\boldsymbol{\theta},\boldsymbol{z})d\boldsymbol{\theta}$$
$$\le 4\sigma_l^2(k_l+1)\log(k_{l+1}2^{k_l+1}) + B_l\sqrt{4\sigma_l^2(k_l+1)\log(k_{l+1}2^{k_l+1})}$$
$$\le \sqrt{4\sigma_l^2(k_l+1)\log(k_{l+1}2^{k_l+1})}\left(\sqrt{4\sigma_l^2(k_l+1)\log(k_{l+1}2^{k_l+1})} + B_l\right)$$
$$\le 2B_l\sqrt{4\sigma_l^2(k_l+1)\log(k_{l+1}2^{k_l+1})}$$

since $\sqrt{4\sigma_l^2(k_l+1)\log(k_{l+1}2^{k_l+1})}$ is bounded above by

$$\sqrt{\frac{4s_l}{8n(L+1)}\left(4^{L-l}(k_l+1)\log(k_{l+1}2^{k_l+1})\prod_{m=0,m\neq l}^{L} B_m^2\right)^{-1}(k_l+1)\log(k_{l+1}2^{k_l+1})}$$
$$= B_l\sqrt{\frac{s_l}{2n(L+1)}\left(4^{L-l}\prod_{m=0}^{L}B_m^2\right)^{-1}} \le B_l,$$

The quantity in square root <1 for large n.

Let $b_j = (k_j+1)\log(k_{j+1}2^{k_j+1})$. From relation (18), we get

$$\int \|\eta_{\boldsymbol{\theta}} - \eta_{\boldsymbol{\theta}^*}\|_2^2 q(\boldsymbol{\theta},\boldsymbol{z})d\boldsymbol{\theta} \le \sum_{j=0}^{L} c_{j-1}^2(4\sigma_j^2 b_j)\left(\prod_{m=j+1}^{L} 4B_m^2\right)$$
$$+\ 2\sum_{j=0}^{L}\sum_{j'=0}^{j-1} c_{j-1}c_{j'-1}2B_j\sqrt{4\sigma_j^2 b_j}\left(\prod_{m=j+1}^{L}4B_m^2\right)$$
$$\times\ \sqrt{4\sigma_{j'}^2 b_{j'}}\left(\prod_{m=j'+1}^{j-1}2B_m\right)$$
$$= 4\sum_{j=0}^{L}4^{L-j}\sigma_j^2 b_j\left(\prod_{m=0}^{j-1}B_m^2\right)\left(\prod_{m=j+1}^{L}B_m^2\right)$$
$$+\ 8\sum_{j=0}^{L}\sum_{j'=0}^{j-1}\left(\prod_{m=0}^{j-1}B_m\right)\left(\prod_{m=0}^{j-1}B_m\right)2B_j\left(\prod_{m=j+1}^{L}4B_m^2\right)$$
$$\times\ \left(\prod_{m=j'+1}^{j-1}2B_m\right)\sqrt{\sigma_j^2 b_j}\sqrt{\sigma_{j'}^2 b_{j'}}$$
$$= 4\sum_{j=0}^{L}2^{2L-2j}\sigma_j^2 b_j\prod_{m=0,m\neq j}^{L}B_m^2$$
$$+\ 8\sum_{j=0}^{L}\sum_{j'=0}^{j-1}4^{L-j}2^{j-j'}\left(\prod_{m=0}^{j-1}B_m\right)\left(\prod_{m=0}^{j-1}B_m\right)$$
$$\times\ \left(\prod_{m=j+1}^{L}B_m\right)\left(\prod_{m=j'+1}^{L}B_m\right)\sqrt{\sigma_j^2 b_j}\sqrt{\sigma_{j'}^2 b_{j'}}$$
$$= 4\sum_{j=0}^{L}2^{2L-2j}\sigma_j^2 b_j\left(\prod_{m=0,m\neq j}^{L}B_m^2\right)$$
$$+\ 8\sum_{j=0}^{L}\sum_{j'=0}^{j-1}2^{L-j}2^{L-j'}\left(\prod_{m=0,m\neq j}^{L}B_m\right)\left(\prod_{m=0,m\neq j'}^{L}B_m\right)\sqrt{\sigma_j^2 b_j}\sqrt{\sigma_{j'}^2 b_{j'}}$$
$$= 4\left(\sum_{j=0}^{L}2^{L-j}\sqrt{\sigma_j^2 b_j}\left(\prod_{m=0,m\neq j}^{L}B_m\right)\right)^2 = 4\left(\sum_{j=0}^{L}\sqrt{\frac{s_j}{8n(L+1)}}\right)^2$$
$$= \frac{1}{2n(L+1)}\left(\sum_{j=0}^{L}\sqrt{s_j}\right)^2 \le \frac{\sum_{j=0}^{L}s_j}{2n} \le \sum_{j=0}^{L}r_l$$

This concludes the proof of (27). Next,

$$d_{\mathrm{KL}}(q,\pi) \le \log\frac{1}{\pi(\boldsymbol{z})}$$
$$+\ \mathbb{1}(\boldsymbol{z}=\boldsymbol{\gamma}^*)d_{\mathrm{KL}}\Bigg(\Bigg\{\prod_{l=0}^{L-1}\prod_{i=1}^{k_{l+1}}\prod_{j=1}^{k_l+1}\Big\{\gamma_{li}^*\mathcal{N}(\overline{w}_{lij}^*,\sigma_l^2) + (1-\gamma_{li}^*)\delta_0\Big\}$$
$$\prod_{j=1}^{k_L+1}\mathcal{N}(\overline{w}_{Lj}^*,\sigma_L^2)\Bigg\},\Bigg\{\prod_{l=0}^{L-1}\prod_{i=1}^{k_{l+1}}\prod_{j=1}^{k_l+1}\Bigg\{z_{li}\mathcal{N}(0,\sigma_0^2)$$

$$
+ (1 - z_{li})\delta_0 \Big\} \prod_{j=1}^{k_L+1} \mathcal{N}(0, \sigma_0^2) \Big\} \Big)
$$

$$
= \log \frac{1}{\prod_{l=0}^{L-1} \lambda_l^{s_l} (1-\lambda_l)^{k_{l+1}-s_l}}
$$

$$
+ \sum_{l=0}^{L-1} \sum_{i=1}^{k_{l+1}} \sum_{j=1}^{k_l+1} d_{\mathrm{KL}} \Big(\gamma_{li}^* \mathcal{N}(\overline{w}_{lij}^*, \sigma_l^2) + (1-\gamma_{li}^*)\delta_0,
$$

$$
\gamma_{li}^* \mathcal{N}(0, \sigma_0^2) + (1-\gamma_{li}^*)\delta_0 \Big) + \sum_{j=1}^{k_L+1} d_{\mathrm{KL}} \Big(\mathcal{N}(\overline{w}_{Lj}^*, \sigma_L^2), \mathcal{N}(0, \sigma_0^2) \Big)
$$

$$
= \sum_{l=0}^{L-1} \left(s_l \log \frac{1}{\lambda_l} + (k_{l+1} - s_l) \log \frac{1}{1-\lambda_l} \right)
$$

$$
+ \sum_{l=0}^{L-1} \sum_{i=1}^{k_{l+1}} \sum_{j=1}^{k_l+1} \gamma_{li}^* \left\{ \frac{1}{2} \log \frac{\sigma_0^2}{\sigma_l^2} + \frac{\sigma_l^2 + \overline{w}_{lij}^{*\,2}}{2\sigma_0^2} - \frac{1}{2} \right\}
$$

$$
+ \sum_{j=1}^{k_L+1} \left\{ \frac{1}{2} \log \frac{\sigma_0^2}{\sigma_L^2} + \frac{\sigma_L^2 + \overline{w}_{Lj}^{*\,2}}{2\sigma_0^2} - \frac{1}{2} \right\}
$$

$$
\leq \sum_{l=0}^{L-1} Cnr_l + \sum_{l=0}^{L-1} \frac{s_l k_l + s_l}{2} \left[\frac{\sigma_l^2}{\sigma_0^2} + \frac{B_l^2}{\sigma_0^2 (k_l+1)} - 1 + \log \frac{\sigma_0^2}{\sigma_l^2} \right]
$$

$$
+ \frac{k_L + 1}{2} \left[\frac{\sigma_L^2}{\sigma_0^2} + \frac{B_L^2}{\sigma_0^2 (k_L+1)} - 1 + \log \frac{\sigma_0^2}{\sigma_L^2} \right]
$$

where the first inequality follows from Lemma A.4. The inequality in the above line uses $\sum_{j=1}^{k_l+1} \overline{w}_{lij}^{*\,2} \leq B_l^2$ and similar to the proof of Lemma 4.1 in Bai et al. (2020) uses .

Let $\sigma_0^2 = 1$ and it could be easily derived that $\sigma_l^2 \leq 1$.

$$
d_{\mathrm{KL}}(q, \pi) \leq \sum_{l=0}^{L-1} Cnr_l + \sum_{l=0}^{L-1} \frac{s_l}{2}(k_l+1) \left[\frac{B_l^2}{k_l+1} - \log \sigma_l^2 \right]
$$

$$
+ \frac{(k_L+1)}{2} \left[\frac{B_L^2}{k_L+1} - \log \sigma_L^2 \right]
$$

$$
= \sum_{l=0}^{L-1} Cnr_l + \sum_{l=0}^{L-1} \frac{s_l}{2}(k_l+1) \Bigg[\frac{B_l^2}{k_l+1}
$$

$$
- \log \left(\frac{s_l}{8n(L+1)} \left[4^{L-l} b_l \prod_{m=0, m \neq l}^{L} B_m^2 \right]^{-1} \right) \Bigg]
$$

$$
+ \frac{(k_L+1)}{2} \left[\frac{B_L^2}{k_L+1} - \log \left(\frac{1}{8n(L+1)} \left[b_L \prod_{m=0, m \neq L}^{L} B_m^2 \right]^{-1} \right) \right]
$$

$$
= \sum_{l=0}^{L-1} Cnr_l + \sum_{l=0}^{L} \frac{s_l}{2}(k_l+1) \Bigg[\frac{B_l^2}{k_l+1}
$$

$$
- \log \left(\frac{s_l}{8n(L+1)} \left[4^{L-l} b_l \prod_{m=0, m \neq l}^{L} B_m^2 \right]^{-1} \right) \Bigg]
$$

$$
= \sum_{l=0}^{L-1} Cnr_l + \sum_{l=0}^{L} \frac{s_l}{2} B_l^2 + \sum_{l=0}^{L} \frac{s_l}{2}(k_l+1) \log \left(\frac{8n(L+1)}{s_l} \right)
$$

$$
+ \sum_{l=0}^{L} s_l (k_l+1)(L-l) \log 2 + \sum_{l=0}^{L} \frac{s_l}{2}(k_l+1) \log(k_l+1)
$$

$$
+ \sum_{l=0}^{L} \frac{s_l}{2}(k_l+1) \log \left(\log(k_{l+1} 2^{k_l+1}) \right)
$$

$$
+ \sum_{l=0}^{L} s_l (k_l+1) \left(\sum_{m=0, m \neq l}^{L} \log B_m \right)
$$

$$
\leq \sum_{l=0}^{L-1} Cnr_l + \sum_{l=0}^{L} \frac{s_l}{2} B_l^2 + \sum_{l=0}^{L} \frac{s_l}{2}(k_l+1) \log \left(\frac{8n(L+1)}{s_l} \right)
$$

$$
+ L \sum_{l=0}^{L} s_l (k_l+1)
$$

$$
+ \sum_{l=0}^{L} \frac{s_l}{2}(k_l+1)(\log(k_l+1) + \log(k_{l+1} + k_l + 1))
$$

$$
+ \sum_{l=0}^{L} s_l (k_l+1) \left(\sum_{m=0, m \neq l}^{L} \log B_m \right)
$$

$$
\leq \sum_{l=0}^{L-1} Cnr_l + \sum_{l=0}^{L} \frac{s_l}{2} B_l^2 + \sum_{l=0}^{L} \frac{s_l}{2}(k_l+1) \log \left(\frac{8n(L+1)}{s_l} \right)
$$

$$
+ L \sum_{l=0}^{L} s_l (k_l+1)
$$

$$
+ \sum_{l=0}^{L} s_l (k_l+1) \log(k_{l+1} + k_l + 1)
$$

$$
+ \sum_{l=0}^{L} s_l (k_l+1) \left(\sum_{m=0, m \neq l}^{L} \log B_m \right)
$$

$$
\leq \sum_{l=0}^{L-1} Cnr_l + \sum_{l=0}^{L} s_l (k_l+1) \Bigg[\frac{B_l^2}{2(k_l+1)} + \left(\sum_{m=0, m \neq l}^{L} \log B_m \right)
$$

$$
+ L + \log(k_{l+1} + k_l + 1)
$$

$$
+ \frac{1}{2} \log \left(\frac{8n(L+1)}{s_l} \right) \Bigg]
$$

$$
\leq \sum_{l=0}^{L-1} (C + C') nr_l + C' nr_L
$$

$$
+ \sum_{l=0}^{L} s_l (k_l+1) \Bigg[\frac{B_l^2}{k_l+1} + \left(\sum_{m=0, m \neq l}^{L} \log B_m \right)
$$

$$
+ L + \log(k_{l+1} + k_l + 1) + \log \left(\frac{n}{s_l} \right) \Bigg]
$$

$$
\leq \sum_{l=0}^{L-1} (C + C') nr_l + C' nr_L + \sum_{l=0}^{L} s_l (k_l+1) \vartheta_l \leq C_1 n \sum_{l=0}^{L} r_l
$$

This concludes the proof of (27). □

Proof of Corollary 4.5. The proof is a direct consequence of Theorem 4.4 in the main paper as long as assumptions of Lemmas 4.2 and 4.3 parts 1 and 2 hold when $\sigma_0^2 = 1, -\log \lambda_l = \log(k_{l+1}) + C_l (k_l+1) \vartheta_l$ and $\epsilon_n = \sqrt{(\sum_{l=0}^{L} r_l + \xi) \sum_{l=0}^{L} u_l}$. This what we show next.

Verifying assumption (19) under Proof of Lemma 4.2: Note, $\sum u_l = O(\epsilon_n^2)$, thus

$$
\sum u_l \log L = o(n\epsilon_n^2) \iff \log L = o(n(\sum r_l + \xi))
$$

which is indeed true since $\log L = o(L^2)$ and $L^2 \leq n \sum r_l$. We will show that $(k_{l+1}\lambda_l)/s_l^\circ \to 0$. With $\lambda_l = (1/k_{l+1}) \exp(-C_l(k_l+1)\vartheta_l)$,

$$\begin{aligned}\frac{k_{l+1}\lambda_l}{s_l^\circ} &\leq \frac{\sum u_l \exp(-C(k_l+1)\vartheta_l)}{n\epsilon_n^2}\\ &= \frac{\exp(-C(k_l+1)\vartheta_l + \log \sum u_l)}{n\epsilon_n^2}\\ &\leq \frac{\exp(-C(k_l+1)\vartheta_l + \vartheta_l)}{n\epsilon_n^2} \to 0\end{aligned}$$

where the above relation holds since $\log \sum u_l \leq \vartheta_l$, $\vartheta_l \to \infty$, $k_l \to \infty$ and $n\epsilon_n^2 \to \infty$.

Verifying assumption under Proof of Lemma 4.3 part 1. and part 2. Note,

$$-\log \lambda_l = \log(k_{l+1}) + C_l(k_l+1)\vartheta_l \leq \vartheta_l + C_l(k_l+1)\vartheta_l = O\{(k_l+1)\vartheta_l\}$$

And then,

$$1 - \lambda_l = 1 - \exp(-C_l\vartheta_l(k_l+1))/k_{l+1}$$

$$-\log(1-\lambda_l) \sim \exp(-C_l\vartheta_l(k_l+1))/k_{l+1} = O\{(k_l+1)s_l\vartheta_l/k_{l+1}\}$$

since $\exp(-C_l\vartheta_l(k_l+1)) \to 0$ and $(k_l+1)s_l\vartheta_l \to \infty$. □

Appendix B. Additional numerical experiments details

B.1. FLOPs calculation

We only count multiply operation for floating point operations (FLOPs) similar to Zhao et al. (2019). In 2D convolution layer, we assume convolution is implemented as a sliding window and that the nonlinearity function is computed for free. Then, for a 2D convolutional layer (given bias is present) we get FLOPs as:

$$\text{FLOPs} = (C_{in,pruned}K_wK_h + 1)O_wO_hC_{out,pruned}$$

where, $C_{in,pruned}$, $C_{out,pruned}$ are the number of input channels and output channels after pruning. Channels are pruned if all the parameters associated with that channel in convolution mapping are zero. K_w and K_h are the kernel width and height respectively. Finally, O_w, O_h are output width and height where $O_w = (I_w + 2 \times P_w - D_w \times (K_w - 1) - 1)/S_w + 1$ and $O_h = (I_h + 2 \times P_h - D_h \times (K_h - 1) - 1)/S_h + 1$. Here, I_w, I_h are input, P_w, P_h are padding, D_w, D_h are dilation, S_w, S_h are stride widths and heights respectively.

For fully connected (linear) layers (with bias) we get FLOPs as:

$$\text{FLOPs} = (I_{pruned} + 1)O_{pruned}$$

where, I_{pruned} is the number of pruned input neurons and O_{pruned} is the number of pruned output neurons.

B.2. Variational parameters initialization

We initialize the γ_{lj}'s at a value close to 1 for all of our experiments. This ensures that at epoch 0, we have a fully connected deep neural network. This also warrants that most of the weights do not get pruned off at a very early stage of training which might lead to bad performance. The variational parameters $\mu_{ljj'}$ are initialized using $U(-0.6, 0.6)$ for simulation and UCI regression examples whereas for classification Kaiming uniform initialization (He et al., 2015) is used. Moreover, $\sigma_{ljj'}$ are reparameterized using softplus function: $\sigma_{ljj'} = \log(1 + \exp(\rho_{ljj'}))$ and $\rho_{ljj'}$ are initialized using a constant value of -6. This keeps initial values of $\sigma_{ljj'}$ close to 0 ensuring that the initial values of network weights stay close to Kaiming uniform initialization.

B.3. Hyperparameters for training

We keep MC sample size (S) to be 1 during training. We choose learning rate of 3×10^{-3}, batch size of 400, and 10000 epochs in the 20 neurons case of simulation study-I. We use learning rate of 10^{-3}, batch size of 400, and 20000 epochs in the 100 neurons case of simulation study-I. Next, we use learning rate of 5×10^{-3}, full batch, and 10000 epochs for simulation study-II. In UCI regression datasets, we choose batch size = 128 and run 500 epochs for *Concrete, Wine, Power Plant*, 800 epochs for *Kin8nm*. For *Protein* and *Year* datasets, we choose batch size of 256 and run 100 epochs. For all the UCI regression datasets we keep learning rate of 10^{-3}. The Adam algorithm (Kingma & Ba, 2015) is chosen for optimization of model parameters.

In image classification datasets, for SS-IG model, we use 10^{-3} learning rate and minibatch size of 1024 in all experiments except in Lenet-Caffe on Fashion-MNIST experiment where we use 2×10^{-3} learning rate and 1024 minibatch size. For SV-BNN model, we take 10^{-3} learning rate and 1024 minibatch size in all experiments after extensive hyperparameter search. For VBNN model, we take learning rate of 10^{-4} and minibatch size of 128 according to Blundell et al. (2015). We train each model for 1200 epochs using Adam optimizer in all the image classification experiments provided in main paper.

B.4. Fine tuning of constant in prior inclusion probability expression

Recall the layer-wise prior inclusion probabilities: $\lambda_l = (1/k_{l+1}) \exp(-C_l(k_l+1)\vartheta_l)$ from Corollary 4.5. In our numerical experiments, we use this expression to choose an optimal value of λ_l in each layer of a given network. The λ_l varies as we vary our constant C_l and we next describe how is C_l chosen. The influence of C_l is mainly due to the $k_l + 1$ term and ${B_l}^2/(k_l+1)$ from ϑ_l term. We ensure that each incoming weight and bias onto the node from layer $l+1$ is bounded by 1 which leads us to choose B_l to be k_l+1. So the leading term from $(k_l+1)\vartheta_l$ is (k_l+1) and C_l has to be chosen such that we avoid making exponential term from λ_l expression close to 0. In our experiments we choose C_l values in the negative order of 10 such that prior inclusion probabilities do not fall below 10^{-50}. If we instead choose a λ_l value very close to 0 then we might prune off all the nodes in each layer or might make the training unstable which is not ideal. Overall the aforementioned strategy of choosing C_l constant values ensure reasonable values for the λ_l in each layer.

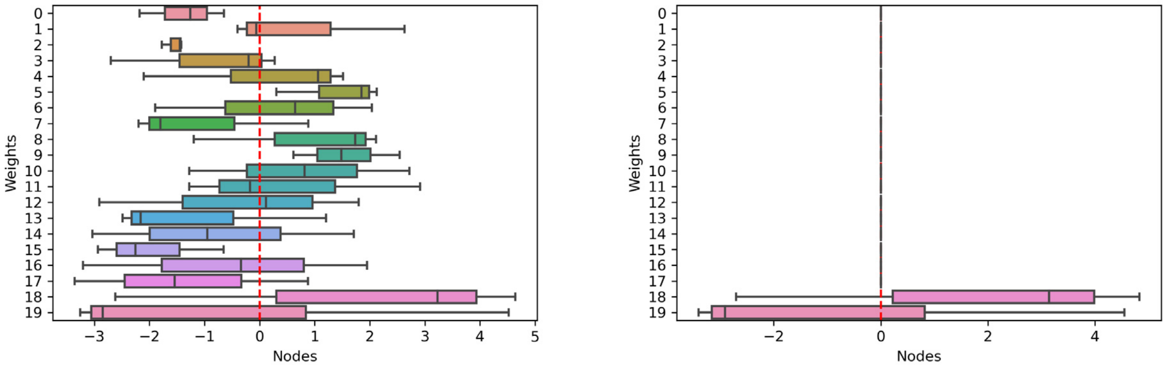
B.5. Simulation study I: extra details

First we provide the network parameters used to generate the data for this simulation experiment. The edge weights in the underlying 2-2-1 network are as follows: $\boldsymbol{W}_0 = \{w_{011} = 10, w_{012} = 15, w_{021} = -15, w_{022} = 10\}$; $\boldsymbol{W}_1 = \{w_{111} = -3, w_{121} = 3\}$ and $\boldsymbol{v}_0 = \{v_{01} = -5, v_{02} = 5\}$; $\boldsymbol{v}_1 = \{v_{11} = 4\}$.

Below we provide additional results demonstrating the model selection ability of our SS-IG approach in a wider network consisting of 100 nodes in the single hidden layer structure considered in the simulation study-I from main paper (see Fig. 5).

B.6. Effect of hidden layer widths

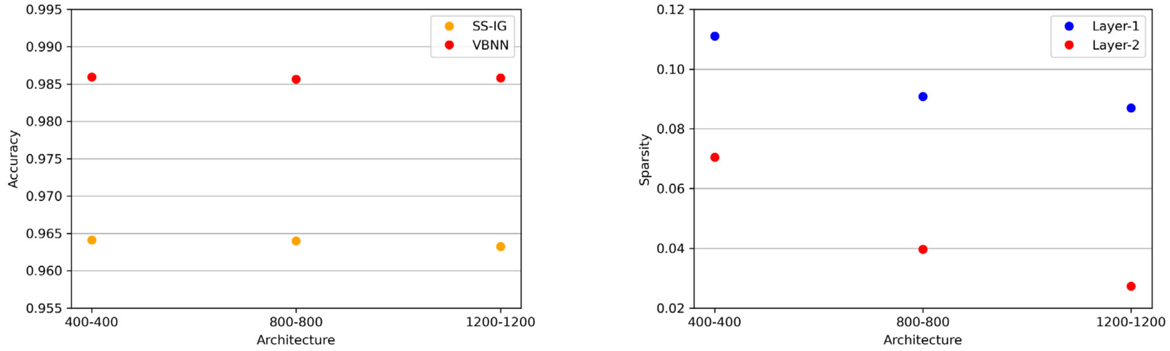
Here, we explore 2-hidden layer neural networks with varying widths. For our SS-IG model we use 10^{-3} learning rate and minibatch size of 1024 while for VBNN model, we take learning rate of 10^{-4} and minibatch size of 128 according to Blundell et al. (2015). We train both the models for 400 epochs using Adam optimizer.



(a) VBNN

(b) SS-IG

Fig. 5. Node-wise weight magnitudes recovered by VBNN and proposed SS-IG model in the synthetic regression data generated using 2-2-1 network. The boxplots show the distribution of incoming weights into a given hidden layer node. Only the 20 nodes with the largest edge weights are displayed.



(a) Prediction accuracy per architecture

(b) Layer-wise sparsity per architecture

Fig. 6. MNIST experiment results for varying hidden layer widths.

Fig. 6 summarizes the results. We have provided results for 3 different architectures which have 400, 800, and 1200 nodes each in their 2-hidden layers. In Fig. 6(a), we find that across the architectures both SS-IG and VBNN models have similar predictive performance. Further, our method is able to prune off more than 88% of first hidden layer nodes and more than 92% of second hidden layer nodes (Fig. 6(b)) at the expense of 2% accuracy loss due to sparsification compared to the densely connected VBNN. We also observe that as model capacity increases the sparsity percentage per layer decreases. This suggests that, each architecture is trying to reach a sparse network of comparable size.

References

Alvarez, J. M., & Salzmann, M. (2016). Learning the number of neurons in deep networks. In *Proceedings of the 30th Advances in neural information processing systems*. Barcelona, Spain.

Bai, J., Song, Q., & Cheng, G. (2020). Efficient variational inference for sparse deep learning with theoretical guarantee. In *Proceedings of the 34th Advances in neural information processing systems* (pp. 466–476). Vancouver, Canada.

Bhattacharya, S., & Maiti, T. (2021). Statistical foundation of Variational Bayes neural networks. *Neural Networks*, *137*, 151–173.

Blei, D. M., Kucukelbir, A., & McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, [ISSN: 1537-274X] *112*(518), 859–877.

Blei, D. M., & 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., & Wierstra, D. (2015). Weight uncertainty in neural network. In *Proceedings of machine learning research, vol. 37* (pp. 1613–1622). PMLR.

Cannings, T. I., & Samworth, R. J. (2017). Random-projection ensemble classification. *Journal of the Royal Statistical Society. Series B. Statistical Methodology*, *79*(4), 959–1035.

Chérief-Abdellatif, B.-E. (2020). Convergence rates of variational inference in sparse deep learning. In *Proceedings of the 37th International conference on machine learning, vol. 119* (pp. 1831–1842). Vienna, Austria.

Chérief-Abdellatif, B.-E., & Alquier, P. (2018). Consistency of variational Bayes inference for estimation and model selection in mixtures. *Electronic Journal of Statistics*, *12*(2), 2995–3035. http://dx.doi.org/10.1214/18-EJS1475.

Dua, D., & Graff, C. (2017). UCI machine learning repository. http://archive.ics.uci.edu/ml.

Elfwing, S., Uchibe, E., & Doya, K. (2018). Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, [ISSN: 0893-6080] *107*, 3–11. http://dx.doi.org/10.1016/j.neunet.2017.12.012, Special issue on deep reinforcement learning.

Frankle, J., & Carbin, M. (2019). The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International conference on learning representations*. New Orleans, USA: URL https://openreview.net/forum?id=rJl-b3RcF7.

Friedman, J., Hastie, T., & Tibshirani, R. (2009). *Springer series in statistics.*, *The elements of statistical learning*. Springer, New York.

Gal, Y. (2016). *Uncertainty in deep learning* Ph.D. thesis.

Ghosal, S., & van der Vaart, A. W. (2007). Convergence rates of posterior distributions for noniid observations. *The Annals of Statistics*, *35*(1), 192–223.

Ghosh, S., Yao, J., & Doshi-Velez, F. (2019). Model selection in Bayesian neural networks via horseshoe priors. *Journal of Machine Learning Research*, *20*, 1–46.

Grenander, U. (1981). *Abstract inference*. New York: Wiley.

Han, S., Mao, H., & Dally, W. J. (2016). Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. In *4th International conference on learning representations*. San Juan, Puerto Rico.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *IEEE International conference on computer vision* (pp. 1026–1034). http://dx.doi.org/10.1109/ICCV.2015.123.

Hernandez-Lobato, J. M., & Adams, R. (2015). Probabilistic backpropagation for scalable learning of Bayesian neural networks. In *Proceedings of the 32nd International conference on machine learning* (pp. 1861–1869). Lille, France.

Hinton, G. E., & 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* (pp. 5–13). Santa Cruz, USA.

Jang, E., Gu, S., & Poole, B. (2017). Categorical reparameterization with gumbel-softmax. In *5th International conference on learning representations*. Toulon, France.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., & Sau, L. K. (1999). An introduction to variational methods for graphical models. *Machine Learning*, *37*, 183–233.

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In *3rd International conference on learning representations*. San Diego, USA.

Lee, H. K. H. (2000). Consistency of posterior distributions for neural networks. *Neural Networks*, *13*, 629–642.

Louizos, C., Ullrich, K., & Welling, M. (2017). Bayesian compression for deep learning. In *Proceedings of the 30th Advances in neural information processing systems* (pp. 3288–3298). Long Beach, CA, USA.

Lu, L., Shin, Y., Su, Y., & Em Karniadakis, G. (2020). Dying ReLU and initialization: Theory and numerical examples. *Communications in Computational Physics*, *28*(5), 1671–1706. http://dx.doi.org/10.4208/cicp.OA-2020-0165.

Maddison, C. J., Mnih, A., & Teh, Y. W. (2017). The concrete distribution: A continuous relaxation of discrete random variables. In *5th International conference on learning representations*. Toulon, France.

Mitchell, T. J., & Beauchamp, J. J. (1988). Bayesian variable selection in linear regression. *Journal of the American Statistical Association*, *83*, (404), 1023–1032.

Molchanov, D., Ashukha, A., & Vetrov, D. (2017). Variational dropout sparsifies deep neural networks. In *Proceedings of the 34th international conference on machine learning, vol. 70* (pp. 2498–2507). Sydney, NSW, Australia.

Mozer, M. C., & Smolensky, P. (1988). Skeletonization: A technique for trimming the fat from a network via relevance assessment. In *Advances in neural information processing systems, vol. 1* (pp. 107–115). Denver, USA.

Neal, R. (1992). Bayesian learning via stochastic dynamics. In *Proceedings of the 5th Advances in neural information processing systems, vol. 5*.

Neklyudov, K., Molchanov, D., Ashukha, A., & Vetrov, D. P. (2017). Structured Bayesian pruning via log-normal multiplicative noise. In *Proceedings of the 30th Advances in neural information processing systems* (pp. 6775–6784). Long Beach, CA, USA.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., ... Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems, vol. 32* (pp. 8024–8035). Curran Associates, Inc..

Pati, D., Bhattacharya, A., & Yang, Y. (2018). On statistical optimality of variational Bayes. In A. Storkey, & F. Perez-Cruz (Eds.), *Proceedings of Machine Learning Research*: *vol. 84*, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics* (pp. 1579–1588). PMLR, URL http://proceedings.mlr.press/v84/pati18a.html.

Pollard, D. (1991). Bracketing methods in statistics and econometrics. In W. A. Barnett, J. Powell, & G. E. Tauchen (Eds.), *Nonparametric and semiparametric methods in econometrics and statistics: proceedings of the fifth international symposium in econometric theory and econometrics* (pp. 337–355). Cambridge, UK: Cambridge University Press.

Polson, N., & Ročková, V. (2018). Posterior concentration for sparse deep learning. In *32nd Conference on advances in neural information processing systems* (pp. 930–941). Montréal, Canada.

Ramachandran, P., Zoph, B., & Le, Q. V. (2017). Searching for activation functions. CoRR abs/1710.05941 arXiv:1710.05941.

Scardapane, S., Comminiello, D., Hussain, A., & Uncini, A. (2017). Group sparse regularization for deep neural networks. *Neurocomputing*, *241*, 81–89.

Schmidt-Hieber, J. (2020). Nonparametric regression using deep neural networks with ReLU activation function. *The Annals of Statistics*, *48*(4), 1875–1897.

Sun, Y., Song, Q., & Liang, F. (2021). Consistent sparse deep learning: Theory and computation. *Journal of the American Statistical Association*, (ja), 1–42.

Wen, W., Wu, C., Wang, Y., Chen, Y., & Li, H. (2016). Learning structured sparsity in deep neural networks. In *Proceedings of the 29th Advances in neural information processing systems*. Barcelona, Spain.

Zhang, C., Bengio, S., Hardt, M., Recht, B., & Vinyals, O. (2017). Understanding deep learning requires rethinking generalization. In *5th International conference on learning representations*. Toulon, France.

Zhao, C., Ni, B., Zhang, J., Zhao, Q., Zhang, W., & Tian, Q. (2019). Variational convolutional neural network pruning. In *2019 IEEE/CVF Conference on computer vision and pattern recognition* (pp. 2775–2784). http://dx.doi.org/10.1109/CVPR.2019.00289.

Zhu, M., & Gupta, S. (2018). To prune, or not to prune: Exploring the efficacy of pruning for model compression. In *6th International conference on learning representations (ICLR 2018), Workshop Track Proceedings*. Vancouver, Canada: OpenReview.net, URL https://openreview.net/forum?id=Sy1iIDkPM.