
BAYESIAN SUBNET INFERENCE FOR DETERMINISTIC NEURAL NETWORKS

Disha Maheshwari, Ruqi Zhang
Purdue University

ABSTRACT

Deep Neural Networks (DNNs) are highly adaptable function approximators, capable of learning intricate mappings from input to output. This adaptability stems from the numerous parameters that can be tuned through gradient-based optimization. However, this same flexibility makes DNNs susceptible to overfitting, particularly when the training data is limited. Bayesian Neural Networks (BNNs) address this issue by placing a prior distribution over the network weights and computing a posterior distribution based on the training data. This method allows for the incorporation of uncertainty, leading to more robust predictions, but comes with the downside of increased training cost. For neural networks to effectively support decision-making, it is crucial to accurately quantify the uncertainty in their predictions. Unfortunately, exact posterior inference in neural networks is intractable. Due to the significant overparameterization of neural networks, their accuracy can be maintained by a smaller subnetwork. Additionally, inference over a low-dimensional subspace of the weights can lead to precise uncertainty quantification. As noted in Daxberger et al. (2021), the posterior predictive distribution of a full network can be effectively represented by that of a subnetwork. Therefore, we derive a subnetwork selection strategy using Subspace Variational Inference described in Li et al. (2024). We propose an architecture model with Bayesian subnet in a Deterministic Neural network in which we aim to learn a standard deviation subnet over a deterministic neural network. Starting from a randomly initialised low-dimensional sparse standard deviation subspace, our approach alternately optimises the sparse standard deviation subspace using a removal-and-addition strategy.

1 INTRODUCTION

Deep neural networks (DNNs) have achieved remarkable success across vision, language, and decision-making tasks due to their expressive capacity and large parameterization. However, their deterministic nature means they provide only point estimates and are prone to overfitting, particularly when training data is limited. In many safety-critical applications—such as autonomous driving, medical diagnosis, and out-of-distribution detection—quantifying predictive uncertainty is as important as achieving high accuracy.

Bayesian neural networks (BNNs) address this limitation by placing priors over weights and performing posterior inference, thereby capturing both epistemic and aleatoric uncertainty Neal (2012); Blundell et al. (2015). While BNNs provide robust uncertainty calibration and have been applied successfully in areas such as active learning and risk-sensitive decision-making, exact inference is intractable. Approximate methods such as variational inference Graves (2011); Blundell et al. (2015) and stochastic gradient MCMC Welling & Teh (2011); Chen et al. (2014) make BNNs feasible in practice but remain computationally demanding compared to deterministic deep networks.

To improve scalability, recent work has explored sparse Bayesian formulations. For example, Sparse Subspace Variational Inference Li et al. (2024) learns a posterior in a low-dimensional parameter subspace and dynamically refines it, while Hubin and Storvik Hubin & Storvik (2024) propose sparse BNNs that combine model and parameter uncertainty through scalable variational inference. These approaches demonstrate that leveraging sparsity reduces computational cost and time while preserving predictive performance and uncertainty calibration.

Another complementary line of research focuses on subnetwork Bayesian inference. Subspace methods Izmailov et al. (2019) show that posterior variability can be captured within low-dimensional trajectories defined by SGD, and Daxberger et al. (2021) propose restricting inference to carefully chosen subnetworks while fixing the rest of the parameters at their MAP estimates. Such strategies yield strong calibration and robustness while treating only a fraction of the network in a Bayesian fashion.

Finally, there is growing interest in hybrid deterministic–Bayesian models, which apply Bayesian inference selectively to parts of the network. For instance, Variational Bayesian Last Layers Harrison et al. (2024) apply Bayesian inference only to the final layer, achieving scalable uncertainty estimation with little added cost. Similar results have been reported by Zeng et al. (2018) and Osawa et al. (2019), showing that partial Bayesian treatment of layers can provide much of the benefit of full BNNs.

In this work, we introduce another method for subnetwork Bayesian inference based on combining variational inference with a gradient-based pruning strategy. Specifically, we identify and retain subnetworks by pruning weights with the smallest gradient magnitudes, under the assumption that weights with larger gradients contribute more significantly to the variational objective based on the work by Li et al. (2024).

2 PRELIMINARY

We utilize the traditional Bayesian Neural Network (BNN) approach. Given a dataset $\mathcal{X} = (X, Y)$, where X represents the independent variables and Y the dependent variables, our objective is to compute the probability $p(x|\mathcal{X})$, where $x = (x, y)$ is a new data point. Let $\theta \in \mathbb{R}^d$ be the random parameters of the BNN. The posterior predictive distribution is expressed as:

$$p(x|\mathcal{X}) = \int p(x|\mathcal{X}, \theta) p(\theta|\mathcal{X}) d\theta \approx \frac{1}{N} \sum_{i=1}^N p(x|\mathcal{X}, \theta^{(i)}), \quad \theta^{(i)} \sim p(\theta|\mathcal{X})$$

where $p(\theta|\mathcal{X})$ is the posterior distribution. In this work, we apply variational inference (VI) Jordan et al. (1999) to approximate the true posterior. Specifically, we seek a variational distribution $q_\phi(\theta)$, where ϕ are the parameters of the distribution that minimize the Kullback-Leibler (KL) divergence $\text{KL}(q_\phi(\theta)||p(\theta|\mathcal{X}))$.

In VI a mean-field assumption is typically made for both the approximate posterior q_ϕ and the prior p , so that $q_\phi(\theta) = \prod_{i=1}^d q_\phi(\theta_i)$ and $p(\theta) = \prod_{i=1}^d p(\theta_i)$. Under this assumption, the objective becomes:

$$\text{KL}(q_\phi(\theta)||p(\theta)) - \mathbb{E}_{q_\phi}[\log p(\mathcal{X}|\theta)] = \sum_{i=1}^d \text{KL}(q_\phi(\theta_i)||p(\theta_i)) - \mathbb{E}_{q_\phi}[\log p(\mathcal{X}|\theta)]$$

Since the expectation term is computationally intractable, we use Monte Carlo (MC) sampling to approximate it. During backpropagation, the reparameterization trick Kingma et al. (2015) is used to allow gradient-based optimization with respect to ϕ . These techniques and assumptions form the foundation of our method.

3 STANDARD DEVIATION SUBNET INFERENCE (SDSI)

Standard Deviation Subnet Inference (SDSI) is a variational inference method designed to approximate the posterior distribution of Bayesian neural networks while drastically reducing computational cost. The key idea is to restrict the variational approximation to a *low-dimensional subspace of the variance space*, allowing uncertainty to be modeled only for a subset of parameters while keeping the remainder deterministic.

In traditional Variational Inference (VI), the variational distribution $q_\phi(\theta)$ is parameterized by both the mean μ and variance σ , effectively doubling the number of variational parameters compared to

the model parameters θ . To construct an optimal Bayesian subnet, SDSI initializes the means μ from the weights of a pretrained deterministic network, while restricting the variational distribution of the variances $q_{\phi_\sigma}(\theta)$ to a subspace $S \subset \mathbb{R}^d$. Only a subset of parameter axes is assigned nonzero variance, while all others remain deterministic.

Formally, the optimization problem is:

$$\min_{\phi_\sigma, I} \text{KL}(q_{\phi_\sigma}(\theta) \parallel p(\theta)) - \mathbb{E}_{q_{\phi_\sigma}} [\log p(D \mid \theta)], \quad (1)$$

subject to

$$\forall i \notin I = \{n_1, \dots, n_s\}, \quad q_{\phi_\sigma}(\theta_i) = \delta(\theta_i), \quad (2)$$

where I denotes the indices of the selected parameters, $p(\theta)$ is the prior distribution, and $\delta(\cdot)$ is the Dirac delta enforcing determinism outside the subspace.

Assuming a Gaussian approximate posterior, variances σ_i^2 are learned only along the selected axes $i \in I$, while unselected axes have $\sigma_i^2 = 0$.

Equivalently, the optimization can be expressed as:

$$\min_{(\sigma^2) \in \mathbb{R}^d, \gamma \in \{0,1\}^d} \sum_{i=1}^d \gamma_i \text{KL}(q_{\phi_\sigma}(\theta_i) \parallel p(\theta_i)) - \mathbb{E}_{q_{\phi_\sigma}} [\log p(D \mid \theta)], \quad (3)$$

subject to

$$\|\gamma\|_1 = s, \quad \sigma^2 = \sigma^2 \odot \gamma, \quad (4)$$

where γ is a binary mask selecting the active variance dimensions, and \odot denotes element-wise multiplication.

3.1 VARIANCE SPACE REMOVAL STRATEGY

In Standard Deviation Subnet Inference (SDSI), the variance subspace is initialized with a low-dimensional basis. Parameters with low absolute *Signal-to-Noise Ratio (SNR)*, defined as

$$\text{SNR}_{q_\phi}(\theta_i) = \frac{|\mu_i|}{\sigma_i} \quad (5)$$

where μ_i and σ_i are the mean and standard deviation of the i -th parameter, are pruned from the active subspace. This ensures that only parameters contributing significantly to predictive uncertainty remain active. The idea of pruning based on SNR metrics is inspired by Li et al. (2024) and prior Bayesian pruning works.

3.2 VARIANCE SPACE ADDITION STRATEGY

After pruning, we reintroduce some of the removed parameters back into the variance subspace γ . We select important parameters based on *absolute gradient magnitude*. Li et al. (2024)

For a stochastic batch x with batch size B , and loss $f_{\theta, \gamma}(x)$ where θ is sampled from the variational posterior, the selection criterion is:

$$\mathbb{E}_{q_\phi} \left[\frac{1}{B} \sum_{i=1}^B \nabla_{\theta_i} f_{\theta, \gamma}(x_i) \right] \quad (6)$$

To compute this expectation, we use a *one-step Monte Carlo approximation* over both θ and x :

$$\frac{1}{B} \sum_{i=1}^B \nabla_{\theta_i} f_{\theta, \gamma}(x_i), \quad \theta \sim q_\phi(\theta), \quad x \sim \text{batch}. \quad (7)$$

Parameters with the largest absolute gradient are re-added to the subspace

4 ALGORITHM

This section illustrates the algorithm implemented.

Variable	Description
θ	Random variables representing the weights of the Bayesian Neural Network (BNN).
$p(\theta)$	Prior distribution of the weights
$q_{\phi_{\sigma}}(\theta)$	Variational distribution used to approximate the posterior, parameterized by ϕ_{σ} .
ϕ	Parameters of the variational distribution, representing the mean and variance of the weights in BNNs.
s	Target sparsity level, defining how many parameters should be non-zero in the model.
d	Dimensionality of the parameter space.
S	Subspace where the variational parameters are optimized, defined by a subset of axes in \mathbb{R}^d .
γ	Binary vector indicating which parameters are active (1) or inactive (0) in the sparse subspace.
μ, σ	Mean and variance of the Gaussian distribution for each weight in the variational distribution.
M, T	Number of gradient descent steps and total steps in the optimization algorithm.
P	Pretrained weights from a Deep Neural Network

Table 1: Description of Variables Used in the Algorithm

Algorithm 1 Global Standard Deviation Subnet Inference

Require: A BNN $\theta \in \mathbb{R}^d$ with prior $p(\theta)$, posterior distribution, $q_{\phi}(\theta)$, target sparsity s/d , replacement rate $\{r_t\}$, inner update steps M , total steps T , pretrained deterministic weights P

- 1: Randomly initialize (σ^0, γ^0) , set $\gamma^{-1} = \gamma^0$
- 2: Set $\mu = P$
- 3: Set $\sigma^0 = 0$ if $\gamma^0 = 0$
- 4: **for** $t = 0, \dots, T$ **do**
- 5: **Update** σ .
- 6: $\sigma^{t,0} = \text{Initialize}(t, \sigma^t, \gamma^t, \gamma^{t-1})$
- 7: **for** $m = 0, \dots, M - 1$ **do**
- 8: Obtain $\sigma^{t,m+1}$ using the gradient
- 9: **end for**
- 10: $\sigma^{t+1} = \sigma^{t,M}$.
- 11: **Update** γ .
- 12: $\gamma_{\text{remove}}^t = \text{Removal}(\gamma^t, \sigma^{t+1}, r_t)$
- 13: $\gamma^{t+1} = \text{Addition}(\gamma_{\text{remove}}^t, \sigma^{t+1}, r_t)$
- 14: **end for**

The subspace addition strategy is guided by the absolute magnitude of the gradients, such that weights corresponding to higher gradient magnitudes are preferentially selected in the subspace. The base parameters μ remain fixed during this process, leading the algorithm to implicitly prioritize directions in parameter space that contribute more significantly to the loss gradient.

Empirically, the subspace distribution exhibits minimal change between the initial and final update steps. The distribution change is largely driven by the gradient of the Kullback–Leibler (KL) divergence term in eqn 8 between the trainable posterior and the prior, which dominates the optimization dynamics in the added subspace. This entire process is repeated for just a few epochs.

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \mathcal{L}_{\text{KL}} \quad (8)$$

Furthermore, the newly introduced standard deviation parameters are initialized with small random values of the order of 10^{-6} , ensuring minimal initial influence and promoting stable convergence during early training iterations.

5 EXPERIMENTS AND RESULTS

We evaluate our proposed Global Standard Deviation Subnet Inference (SDSI) against several established baselines, including Sparse Bayesian Neural Networks (BNNs), standard BNNs, deterministic neural networks (DNNs), Variational Bayesian Last Layer (VBLL) methods, and Laplace Subnetwork Inference. All experiments are conducted on the CIFAR-100 dataset using a ResNet-18 backbone, with additional robustness tests on corrupted CIFAR-100 and out-of-distribution (OOD) detection benchmarks.

5.1 CIFAR-100 CLASSIFICATION

On clean CIFAR-100, standard BNNs achieve the highest accuracy of 77.88%, but at a significant parameter and computational cost. Sparse BNNs, while more efficient, drop to 75.66%. Our Global SDSI achieves 76.36%, surpassing Sparse BNNs, DNNs, and both VBLL variants, while also offering the lowest expected calibration error (ECE = 0.0014). This highlights the ability of SDSI to maintain strong accuracy with superior calibration under a reduced parameter cost. Results are summarized in table 2

Models	accuracy	loss	ece
Sparse BNN (10%)	75.66	0.9832	0.00159
Standard BNN	77.88	0.9032	0.00346
DNN	76.12	0.9405	0.00180
VBLL (Full Training)	75.34	1.631	0.0059
VBLL (Post Training)	76.23	0.967	0.0017
Laplace Subnetwork Inference	72.25	1.078	0.0054
Global SDSI (10%)	76.36	0.9655	0.0014

Table 2: Performance comparison of different models on cifar100 dataset

5.2 CORRUPTED CIFAR-100 ROBUSTNESS

When evaluated on the corrupted CIFAR-100 dataset, which introduces distributional shifts, Global SDSI achieves the highest accuracy of 47.98%. This result outperforms standard BNNs (46.53%) and DNNs (46.43%), while maintaining calibration quality comparable to the best baselines. These results demonstrate that Global SDSI preserves robustness even under challenging corrupted data settings. Results are summarized in table 3

Models	accuracy	loss	ece
Sparse BNN (10%)	43.88	2.769	0.0232
Standard BNN	46.53	2.433	0.0192
DNN	46.43	2.485	0.0217
VBLL (Full Training)	47.8	2.96	0.0125
VBLL (Post Training)	47.67	2.49	0.0216
Laplace Subnetwork Inference	42.84	2.563	0.0197
Global SDSI (10%)	47.98	2.49	0.0216

Table 3: Performance comparison of different models on corrupted cifar100 dataset

5.3 OUT-OF-DISTRIBUTION DETECTION

For OOD detection, we tested on SVHN and CiFAR10. Laplace Subnetwork Inference performs strongly on SVHN (AUROC = 0.8486) but not so well with CiFAR10 dataset, while VBLL (post-training) achieves strong performance for CiFAR10 outlier detection (AUROC = 0.7969, AUPR = 0.7493). Global SDSI achieves nearly identical OOD detection scores (AUROC = 0.7964, AUPR = 0.7494) and performs strongly on SVHN OOD dataset as well (AUROC = 0.8281, AUPR = 0.901), showing that sparsifying the variational subspace over deterministic networks can retain uncertainty quantification quality close or better than other baseline methods. Results are summarized in table 4

Models	AUROC SVHN	AUPR SVHN	AUROC CiFAR10	AUPR CiFAR10
Sparse BNN (10%)	0.8339	0.9069	0.5624	0.5509
Standard BNN	0.8312	0.9031	0.5496	0.5429
DNN	0.8299	0.9019	0.5459	0.5332
VBLL (Full Training)	0.8036	0.8865	0.533	0.5227
VBLL (Post Training)	0.8269	0.9002	0.7969	0.7493
Laplace Subnetwork Inference	0.8486	0.9125	0.5472	0.5385
Global SDSI (10%)	0.8281	0.9010	0.7964	0.7494

Table 4: AUC - ROC values

5.4 BASELINES

5.4.1 FULL TRAINING VBLL HARRISON ET AL. (2024)

We jointly optimize the last layer variational posterior (σ_{last_layer}) together with the MAP estimation (θ) of the features.

5.4.2 POST TRAINING VBLL HARRISON ET AL. (2024)

A two-step procedure is used, the feature weights are trained by an arbitrary training procedure (e.g., standard neural network) and the last layer posterior (σ_{last_layer}) is trained with frozen features (θ).

5.4.3 LAPLACE SUBNETWORK INFERENCE DAXBERGER ET AL. (2021)

Laplace Subnetwork Inference fits a Gaussian posterior (via a linearized Laplace approximation) over a selected subnetwork of weights, while keeping the rest fixed at their MAP estimate. The implementation presented in Daxberger et al. (2021) was used to test the idea.

5.5 GLOBAL SUBNET STANDARD DEVIATION INFERENCE (10%)

The model was trained for 40 epochs. The first layer which contributes 0.015% to the total weights was kept 100% dense to give better results.

Starting Distribution

$$\text{nonzero percentages for posterior layers} = \begin{bmatrix} 100.00\% & 10.00\% & 10.00\% & 10.00\% & 10.00\% & 10.00\% & 10.01\% & 10.00\% \\ 10.00\% & 10.00\% & 10.00\% & 10.00\% & 10.00\% & 10.00\% & 10.00\% & 10.00\% \\ 10.00\% & 10.00\% & 10.00\% & 10.00\% & 10.00\% & 10.00\% & 10.00\% & 10.00\% \end{bmatrix}$$

$$\text{total nonzero params} = \frac{12333040}{22420864} (55.01\%)$$

$$\text{total conv nonzero params} = \frac{12276720}{22318464} (55.01\%)$$

Final Distribution

$$\text{nonzero percentages for posterior layers} = \begin{bmatrix} 100.00\% & 95.64\% & 98.97\% & 98.73\% & 98.91\% & 98.15\% & 95.08\% & 100\% \\ 96.99\% & 97.77\% & 93.05\% & 31.18\% & 11.49\% & 0.98\% & 0.04\% & 0.01\% \\ 0.00\% & 0.00\% & 0.01\% & 0.00\% & 0.00\% & 0.00\% & 0.00\% & 0.00\% \end{bmatrix}$$

$$\text{total nonzero params} = \frac{12333040}{22420864} (55.01\%)$$

$$\text{total conv nonzero params} = \frac{12281840}{22318464} (55.03\%)$$

The breakdown of the parameter statistics is as follows. The mean parameters, denoted by μ , constitute 50% of the total weights. The associated variational parameters, denoted by σ , have a sparsity level of 90%, i.e., only 10% of the variational weights are non-zero.

The overall density of non-zero weights is approximately 55%, which can be explained as:

$$\text{Total non-zero density} = 50\% (\mu) + 10\% \times 50\% (\sigma) = 55\%$$

The only non-convolutional layer in the network is the final layer, which is a fully connected (linear) layer. Figure 1 gives a better picture of the subspace distribution in the model.

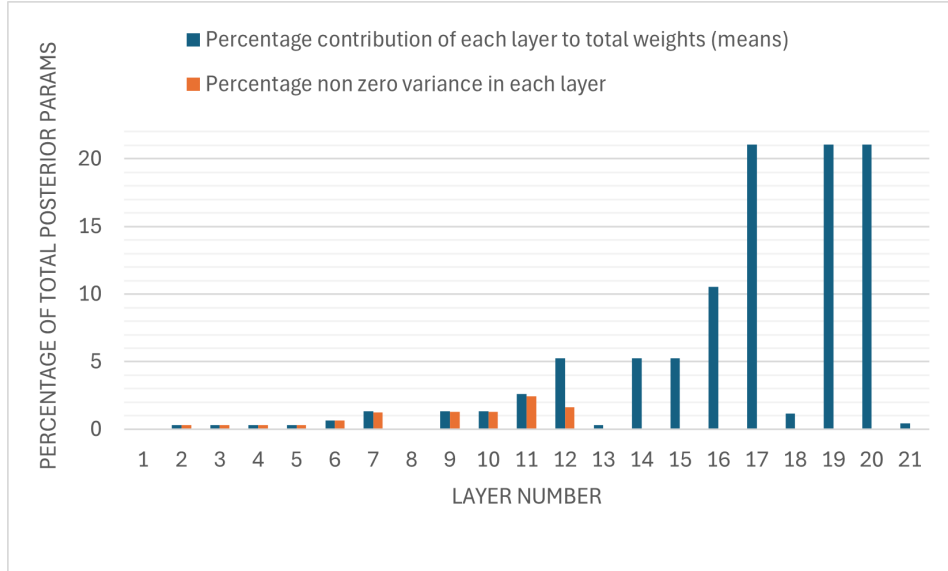


Figure 1: Percentage of nonzero posterior parameters per layer with respect to total posterior params

6 CONCLUSION

In this work, we introduce **Global Subnet Standard Deviation Inference (SDSI)**, a variational inference method that constrains posterior variances to a sparse, low-dimensional subspace. Across CIFAR-100 experiments (both clean and corrupted), SDSI consistently outperforms Sparse BNNs, VBLL methods, and Laplace Subnetworks, while performing competitively with full BNNs at a fraction of the computational cost.

The proposed subspace addition strategy ensures that high-gradient directions are retained, which improves both accuracy and calibration under clean and corrupted datasets. Furthermore, SDSI achieves competitive OOD detection performance, validating its reliability for uncertainty-aware applications. Importantly, the model converges within just 40 training epochs, making it highly efficient.

In summary, Global SDSI offers a favorable trade-off between scalability, predictive performance, and uncertainty estimation. By optimizing only the sparse variance subspace rather than the full parameter space, SDSI substantially reduces computational overhead, providing a practical and efficient approach to Bayesian deep learning.

ACKNOWLEDGMENTS

We sincerely thank Professor Ruqi Zhang and Junbo Li for their invaluable guidance, support, and insightful feedback throughout this work.

REFERENCES

- Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks, 2015. URL <https://arxiv.org/abs/1505.05424>.
- Tianqi Chen, Emily Fox, and Carlos Guestrin. Stochastic gradient hamiltonian monte carlo. In Eric P. Xing and Tony Jebara (eds.), *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pp. 1683–1691, Beijing, China, 22–24 Jun 2014. PMLR. URL <https://proceedings.mlr.press/v32/chen14.html>.
- Erik Daxberger, Eric Nalisnick, James U Allingham, Javier Antorán, and José Miguel Hernández-Lobato. Bayesian deep learning via subnetwork inference. In *International Conference on Machine Learning*, pp. 2510–2521. PMLR, 2021.
- Alex Graves. Practical variational inference for neural networks. *Advances in neural information processing systems*, 24, 2011.
- James Harrison, John Willes, and Jasper Snoek. Variational bayesian last layers, 2024. URL <https://arxiv.org/abs/2404.11599>.
- Aliaksandr Hubin and Geir Storvik. Sparse bayesian neural networks: Bridging model and parameter uncertainty through scalable variational inference. *Mathematics*, 12(6), 2024. ISSN 2227-7390. doi: 10.3390/math12060788. URL <https://www.mdpi.com/2227-7390/12/6/788>.
- Pavel Izmailov, Wesley J. Maddox, Polina Kirichenko, Timur Garipov, Dmitry P. Vetrov, and Andrew Gordon Wilson. Subspace inference for bayesian deep learning. *CoRR*, abs/1907.07504, 2019. URL <http://arxiv.org/abs/1907.07504>.
- Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37:183–233, 1999.
- Diederik P. Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick, 2015. URL <https://arxiv.org/abs/1506.02557>.
- Junbo Li, Zichen Miao, Qiang Qiu, and Ruqi Zhang. Training bayesian neural networks with sparse subspace variational inference. *arXiv preprint arXiv:2402.11025*, 2024.
- Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.
- Kazuki Osawa, Siddharth Swaroop, Anirudh Jain, Runa Eschenhagen, Richard E. Turner, Rio Yokota, and Mohammad Emtiyaz Khan. Practical deep learning with bayesian principles, 2019. URL <https://arxiv.org/abs/1906.02506>.
- Max Welling and Yee Whye Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML’11, pp. 681–688, Madison, WI, USA, 2011. Omnipress. ISBN 9781450306195.
- Jiaming Zeng, Adam Lesnikowski, and Jose M. Alvarez. The relevance of bayesian layer positioning to model uncertainty in deep bayesian active learning, 2018. URL <https://arxiv.org/abs/1811.12535>.

A APPENDIX

You may include other additional sections here.