Bayesian Deep Learning via Subnetwork Inference

Erik Daxberger [1 2] **Eric Nalisnick** [* 3] **James Urquhart Allingham** [* 1] **Javier Antorán** [* 1]
José Miguel Hernández-Lobato [1 4 5]

Abstract

The Bayesian paradigm has the potential to solve core issues of deep neural networks such as poor calibration and data inefficiency. Alas, scaling Bayesian inference to large weight spaces often requires restrictive approximations. In this work, we show that it suffices to perform inference over a small subset of model weights in order to obtain accurate predictive posteriors. The other weights are kept as point estimates. This *subnetwork inference framework* enables us to use expressive, otherwise intractable, posterior approximations over such subsets. In particular, we implement *subnetwork linearized Laplace* as a simple, scalable Bayesian deep learning method: We first obtain a MAP estimate of all weights and then infer a full-covariance Gaussian posterior over a subnetwork using the linearized Laplace approximation. We propose a subnetwork selection strategy that aims to maximally preserve the model's predictive uncertainty. Empirically, our approach compares favorably to ensembles and less expressive posterior approximations over full networks.

1. Introduction

A critical shortcoming of deep neural networks (NNs) is that they tend to be poorly calibrated and overconfident in their predictions, especially when there is a shift between the train and test data distributions (Nguyen et al., 2015; Guo et al., 2017). To reliably inform decision making, NNs need to robustly quantify the *uncertainty* in their predictions (Bhatt et al., 2020). This is especially important for safety-critical applications such as healthcare or autonomous driving (Amodei et al., 2016).

Bayesian modeling (Bishop, 2006; Ghahramani, 2015) presents a principled way to capture uncertainty via the posterior distribution over model parameters. Unfortunately, exact posterior inference is intractable in NNs. Despite recent successes in the field of Bayesian deep learning (Osawa et al., 2019; Maddox et al., 2019; Dusenberry et al., 2020), existing methods invoke unrealistic assumptions to scale to NNs with large numbers of weights. This severely limits the expressiveness of the inferred posterior and thus deteriorates the quality of the induced uncertainty estimates (Ovadia et al., 2019; Fort et al., 2019; Foong et al., 2019a).

Perhaps these unrealistic inference approximations can be avoided. Due to the heavy overparameterization of NNs, their accuracy is well-preserved by a small subnetwork (Cheng et al., 2017). Moreover, doing inference over a low-dimensional subspace of the weights can result in accurate uncertainty quantification (Izmailov et al., 2019). This prompts the following question: *Can a full NN's model uncertainty be well-preserved by a small subnetwork?* In this work we demonstrate that the posterior predictive distribution of a full network *can* be well represented by that of a subnetwork. In particular, our contributions are as follows:

1. We propose *subnetwork inference*, a general framework for scalable Bayesian deep learning in which inference is performed over only a *small subset* of the NN weights, while all other weights are kept deterministic. This allows us to use *expressive posterior approximations* that are typically intractable in large NNs. We present a concrete instantiation of this framework that first fits a MAP estimate of the full NN, and then uses the linearized Laplace approximation to infer a *full-covariance Gaussian posterior* over a subnetwork (illustrated in Fig. 1).

2. We derive a subnetwork selection strategy based on the Wasserstein distance between the approximate posterior for the full network and the approximate posterior for the subnetwork. For scalability, we employ a diagonal approximation during subnetwork selection. Selecting a small subnetwork then allows us to infer weight covariances. Empirically, we find that making approximations during subnetwork selection is much less harmful to the posterior predictive than making them during inference.

3. We empirically evaluate our method on a range of bench-

[*]Equal contribution [1]University of Cambridge [2]Max Planck Institute for Intelligent Systems, Tübingen [3]University of Amsterdam [4]Microsoft Research [5]The Alan Turing Institute. Correspondence to: Erik Daxberger <ead54@cam.ac.uk>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

marks for *uncertainty calibration* and *robustness to distribution shift*. Our experiments demonstrate that expressive subnetwork inference can outperform popular Bayesian deep learning methods that do less expressive inference over the full NN as well as deep ensembles.

2. Subnetwork Posterior Approximation

Let $\mathbf{w} \in \mathbb{R}^D$ be the D-dimensional vector of all neural network weights (i.e. the concatenation and flattening of all layers' weight matrices). Bayesian neural networks (BNNs) aim to capture *model uncertainty*, i.e. uncertainty about the choice of weights $\mathbf{w}$ arising due to multiple plausible explanations of the training data $\mathcal{D}=\{\mathbf{y}, \mathbf{X}\}$. Here, $\mathbf{y} \in \mathbb{R}^O$ is the output variable (e.g. classification label) and $\mathbf{X} \in \mathbb{R}^{N\times I}$ is the feature matrix. First, a prior distribution $p(\mathbf{w})$ is specified over the BNN's weights $\mathbf{w}$. We then wish to infer their full *posterior distribution*

$$p(\mathbf{w}|\mathcal{D}) \;=\; p(\mathbf{w}|\mathbf{y}, \mathbf{X}) \;\propto\; p(\mathbf{y}|\mathbf{X}, \mathbf{w})p(\mathbf{w})\,. \quad (1)$$

Finally, predictions for new data points $\mathbf{X}^*$ are made through marginalisation of the posterior:

$$p(\mathbf{y}^*|\mathbf{X}^*, \mathcal{D}) = \int_{\mathbf{w}} p(\mathbf{y}^*|\mathbf{X}^*, \mathbf{w})p(\mathbf{w}|\mathcal{D})d\mathbf{w}\,. \quad (2)$$

This *posterior predictive* distribution translates uncertainty in weights to uncertainty in predictions. Unfortunately, due to the non-linearity of NNs, it is intractable to infer the exact posterior distribution $p(\mathbf{w}|\mathcal{D})$. It is even computationally challenging to faithfully *approximate* the posterior due to the high dimensionality of $\mathbf{w}$. Thus, crude posterior approximations such as complete factorization, i.e. $p(\mathbf{w}|\mathcal{D}) \approx \prod_{d=1}^{D} q(\mathrm{w}_d)$ where w_d is the d^{th} weight in $\mathbf{w}$, are commonly employed (Hernández-Lobato & Adams, 2015; Blundell et al., 2015; Khan et al., 2018; Osawa et al., 2019). However, it has been shown that such an approximation suffers from severe pathologies (Foong et al., 2019a;b).

In this work, we question the widespread implicit assumption that an expressive posterior approximation must include *all* D of the model weights. Instead, we try to perform inference only over a *small subset* of $S \ll D$ of the weights. The following arguments motivate this approach:

1. **Overparameterization:** Maddox et al. (2020) have shown that, in the neighborhood of local optima, there are many directions that leave the NN's predictions unchanged. Moreover, NNs can be heavily pruned without sacrificing test-set accuracy (Frankle & Carbin, 2019). This suggests that the majority of a NN's predictive power can be isolated to a small subnetwork.
2. **Inference over submodels:** Previous work[1] has provided evidence that inference can be effective even when not performed on the full parameter space. Examples include Izmailov et al. (2019) and Snoek et al. (2015) who perform inference over low-dimensional projections of the weights, and only the last layer of a NN, respectively.

[1] See Section 8 for a more thorough discussion of related work.

We therefore combine these two ideas and make the following two-step approximation of the posterior in (1):

$$p(\mathbf{w}|\mathcal{D}) \;\approx\; p(\mathbf{w}_S|\mathcal{D}) \prod_r \delta(\mathrm{w}_r - \widehat{\mathrm{w}}_r) \quad (3)$$

$$\approx\; q(\mathbf{w}_S) \prod_r \delta(\mathrm{w}_r - \widehat{\mathrm{w}}_r) \;=\; q_S(\mathbf{w})\,. \quad (4)$$

The first approximation (3) decomposes the full NN posterior $p(\mathbf{w}|\mathcal{D})$ into a posterior $p(\mathbf{w}_S|\mathcal{D})$ over the subnetwork $\mathbf{w}_S \in \mathbb{R}^S$ and Dirac delta functions $\delta(\mathrm{w}_r - \widehat{\mathrm{w}}_r)$ over the $D-S$ remaining weights w_r to keep them at fixed values $\widehat{\mathrm{w}}_r \in \mathbb{R}$. Since posterior inference over the subnetwork is still intractable, (4) further approximates $p(\mathbf{w}_S|\mathcal{D})$ by $q(\mathbf{w}_S)$. However, importantly, if the subnetwork is much smaller than the full network, we can afford to make $q(\mathbf{w}_S)$ *more expressive* than would otherwise be possible. We hypothesize that being able to capture rich dependencies across the weights within the subnetwork will provide better results than crude approximations applied to the full set of weights.

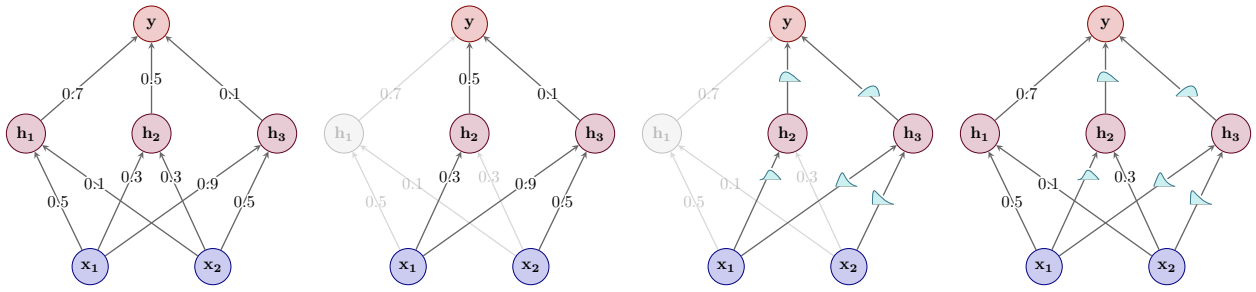
Relationship to Weight Pruning Methods. Note that the posterior approximation in (4) can be viewed as *pruning the variances* of the weights $\{\mathrm{w}_r\}_r$ to zero. This is in contrast to weight pruning methods (Cheng et al., 2017) that set the *weights* themselves to zero. I.e., weight pruning methods can be viewed as removing *weights* to preserve the predictive *mean* (i.e. to retain *accuracy* close to the full model). In contrast, subnetwork inference can be viewed as removing just the *variances* of certain weights—while keeping their means—to preserve the predictive *uncertainty* (e.g. to retain *calibration* close to the full model). Thus, they are complementary approaches. Importantly, by not pruning weights, subnetwork inference retains the *full predictive power* of the full NN to retain its predictive accuracy.

3. Background: Linearized Laplace

In this work we satisfy (4) by approximating the posterior distribution over the weights with linearized Laplace (MacKay, 1992). This is a tractable inference technique that has recently been shown to perform strongly (Foong et al., 2019b; Immer et al., 2020) and can be applied *post-hoc* to pre-trained models. We now describe it in a general setting.

We denote our NN function as $\boldsymbol{f} : \mathbb{R}^I \to \mathbb{R}^O$. We begin by defining a prior over our NN's weights, which we choose to be a fully factorised Gaussian $p(\mathbf{w}) = \mathcal{N}(\mathbf{w}; \mathbf{0}, \lambda\mathbf{I})$. We find a local optimum of the posterior, also known as a *maximum a posteriori* (MAP) setting of the weights:

$$\widehat{\mathbf{w}} = \arg\max_{\mathbf{w}} \left[\log p(\mathbf{y}|\mathbf{X}, \mathbf{w}) + \log p(\mathbf{w})\right]. \quad (5)$$



(a) Point Estimation (b) Subnetwork Selection (c) Bayesian Inference (d) Prediction

Figure 1. Schematic illustration of our proposed approach. (a) We train a neural network using standard techniques to obtain a point estimate of the weights. (b) We identify a small subset of the weights. (c) We estimate a posterior distribution over the selected subnetwork via Bayesian inference techniques. (d) We make predictions using the full network with a mix of Bayesian and deterministic weights.

The posterior is then approximated with a second order Taylor expansion around the MAP estimate:

$$\log p(\mathbf{w}|\mathcal{D}) \approx \log p(\widehat{\mathbf{w}}|\mathcal{D}) - \frac{1}{2}(\mathbf{w}-\widehat{\mathbf{w}})^\top \mathbf{H}(\mathbf{w}-\widehat{\mathbf{w}}) \quad (6)$$

where $\mathbf{H} \in \mathbb{R}^{D\times D}$ is the Hessian of the negative log-posterior density w.r.t. the network weights $\mathbf{w}$:

$$\mathbf{H} = N \cdot \mathbb{E}_{p(\mathcal{D})}\left[-\partial^2 \log p(\mathbf{y}|\mathbf{X},\mathbf{w})/\partial \mathbf{w}^2\right] + \lambda \mathbf{I}\,. \quad (7)$$

Thus, the approximate posterior takes the form of a full-covariance Gaussian with covariance matrix $\mathbf{H}^{-1}$:

$$p(\mathbf{w}|\mathcal{D}) \approx q(\mathbf{w}) = \mathcal{N}\left(\mathbf{w}; \widehat{\mathbf{w}}, \mathbf{H}^{-1}\right). \quad (8)$$

In practise, the Hessian $\mathbf{H}$ is commonly replaced with the *generalized Gauss-Newton matrix (GGN)* $\widetilde{\mathbf{H}} \in \mathbb{R}^{D\times D}$ (Martens & Sutskever, 2011; Martens, 2014; 2016)

$$\widetilde{\mathbf{H}} = \textstyle\sum_{n=1}^{N} \mathbf{J}_n^\top \mathbf{H}_n \mathbf{J}_n + \lambda \mathbf{I}\,. \quad (9)$$

Here, $\mathbf{J}_n = \partial \boldsymbol{f}(\mathbf{x}_n,\mathbf{w})/\partial \mathbf{w} \in \mathbb{R}^{O\times D}$ is the Jacobian of the model outputs $\boldsymbol{f}(\mathbf{x}_n,\mathbf{w}) \in \mathbb{R}^O$ w.r.t. $\mathbf{w}$. $\mathbf{H}_n = -\partial^2 \log p(\mathbf{y}|\boldsymbol{f}(\mathbf{x}_n,\mathbf{w}))/\partial^2 \boldsymbol{f}(\mathbf{x}_n,\mathbf{w}) \in \mathbb{R}^{O\times O}$ is the Hessian of the negative log-likelihood w.r.t. model outputs.

Interestingly, when using a Gaussian likelihood, the Gaussian with a GGN precision matrix corresponds to the *true* posterior distribution when the NN is approximated with a first-order Taylor expansion around $\widehat{\mathbf{w}}$ (Khan et al., 2019; Immer et al., 2020). The *locally linearized* function is

$$\boldsymbol{f}_{\text{lin}}(\mathbf{x},\mathbf{w}) = \boldsymbol{f}(\mathbf{x},\widehat{\mathbf{w}}) + \widehat{\mathbf{J}}(\mathbf{x})(\mathbf{w}-\widehat{\mathbf{w}}) \quad (10)$$

where $\widehat{\mathbf{J}}(\mathbf{x}) = \partial \boldsymbol{f}(\mathbf{x},\widehat{\mathbf{w}})/\partial \widehat{\mathbf{w}} \in \mathbb{R}^{O\times D}$. This turns the underlying probabilistic model from a BNN into a generalized linear model (GLM), where the Jacobian $\widehat{\mathbf{J}}(\mathbf{x})$ acts as a basis function expansion. Making predictions with the GLM $\boldsymbol{f}_{\text{lin}}$ has been found to outperform the corresponding BNN $\boldsymbol{f}$ with the GGN-Laplace posterior (Lawrence, 2001; Foong et al., 2019b; Immer et al., 2020). Additionally, the equivalence between a GLM and a linearized BNN will help us to derive a subnetwork selection strategy in Section 5.

The resulting posterior predictive distribution is

$$p(\mathbf{y}^*|\mathbf{x}^*,\mathcal{D}) = \int p(\mathbf{y}^*|\boldsymbol{f}_{\text{lin}}(\mathbf{x}^*,\mathbf{w}))p(\mathbf{w}|\mathcal{D})\,d\mathbf{w}\,. \quad (11)$$

For regression, when using a Gaussian noise model $p(\mathbf{y}^*|\boldsymbol{f}_{\text{lin}}(\mathbf{x}^*,\mathbf{w})) = \mathcal{N}(\mathbf{y}^*; \boldsymbol{f}_{\text{lin}}(\mathbf{x}^*,\mathbf{w}), \sigma^2)$, our approximate distribution becomes exact $q(\mathbf{w}) = p(\mathbf{w}|\mathcal{D}) = \mathcal{N}(\mathbf{w}; \widehat{\mathbf{w}}, \widetilde{\mathbf{H}}^{-1})$. We obtain the closed form predictive

$$p(\mathbf{y}^*|\mathbf{x}^*,\mathcal{D}) = \mathcal{N}(\mathbf{y}^*; \boldsymbol{f}(\mathbf{x}^*,\widehat{\mathbf{w}}), \Sigma(\mathbf{x}^*) + \sigma^2\mathbf{I})\,, \quad (12)$$

where $\Sigma(\mathbf{x}^*) = \widehat{\mathbf{J}}(\mathbf{x}^*)^\top \widetilde{\mathbf{H}}^{-1} \widehat{\mathbf{J}}(\mathbf{x}^*)$. For classification with a categorical likelihood $p(\mathbf{y}^*|\boldsymbol{f}_{\text{lin}}(\mathbf{x}^*,\mathbf{w})) = \text{Cat}(\mathbf{y}^*; \phi(\boldsymbol{f}_{\text{lin}}(\mathbf{x}^*,\mathbf{w}))$, the posterior is strictly convex. This makes our Gaussian a faithful approximation. Here, $\phi(\cdot)$ refers to the softmax function. The predictive integral has no analytical solution. Instead we leverage the probit approximation (Gibbs, 1998; Bishop, 2006):

$$p(\mathbf{y}^*|\mathbf{x}^*,\mathcal{D}) \approx \text{Cat}\left(\mathbf{y}^*; \phi\left(\frac{\boldsymbol{f}(\mathbf{x}^*,\widehat{\mathbf{w}})}{\sqrt{1+\frac{\pi}{8}\text{diag}(\Sigma(\mathbf{x}^*))}}\right)\right). \quad (13)$$

These closed-form expressions are attractive since they result in the predictive mean and classification boundaries being exactly equal to those of the MAP estimated NN.

Unfortunately, storing the full $D \times D$ covariance matrix over the weight space of a modern NN (i.e. with very large D) is computationally intractable. There have been efforts to develop cheaper approximations to this object, such as only storing diagonal (Denker & LeCun, 1990) or block diagonal (Ritter et al., 2018; Immer et al., 2020) entries, but these come at the cost of reduced predictive performance.

4. Linearized Laplace Subnetwork Inference

We outline the following procedure for scaling the linearized Laplace approximation to large neural network models within the framework of subnetwork inference.

Step #1: Point Estimation, Fig. 1 (a). Train a neural network to obtain a point estimate of the weights, denoted $\widehat{\mathbf{w}}$. This can be done using stochastic gradient-based optimization methods (Goodfellow et al., 2016). Alternatively, we could make use of a pre-trained model.

Step #2: Subnetwork Selection, Fig. 1 (b). Identify a small subnetwork $\mathbf{w}_S \in \mathbb{R}^S, S \ll D$. Ideally, we would like to find the subnetwork which produces a predictive posterior 'closest' to the full-network's predictive distribution. Regrettably, reasoning in the space of functions directly is challenging (Burt et al., 2020). Instead, in Section 5, we describe a strategy that minimizes the Wasserstein distance between the sub- and full-network's weight posteriors.

Step #3: Bayesian Inference, Fig. 1 (c). Use the GGN-Laplace approximation to infer a full-covariance Gaussian posterior over the subnetwork's weights $\mathbf{w}_S \in \mathbb{R}^S$:

$$p(\mathbf{w}_S|\mathcal{D}) \approx q(\mathbf{w}_S) = \mathcal{N}(\mathbf{w}_S; \widehat{\mathbf{w}}_S, \widetilde{\mathbf{H}}_S^{-1}) \tag{14}$$

where $\widetilde{\mathbf{H}}_S \in \mathbb{R}^{S\times S}$ is the GGN w.r.t. the weights $\mathbf{w}_S$:

$$\widetilde{\mathbf{H}}_S = \textstyle\sum_{n=1}^{N} \mathbf{J}_{Sn}^\top \mathbf{H}_n \mathbf{J}_{Sn} + \lambda_S \mathbf{I}\,. \tag{15}$$

Here, $\mathbf{J}_{Sn} = \partial \boldsymbol{f}(\mathbf{x}_n, \mathbf{w}_S)/\partial \mathbf{w}_S \in \mathbb{R}^{O\times S}$ is the Jacobian w.r.t. $\mathbf{w}_S$. $\mathbf{H}_n$ is defined as in Section 2. In order to best preserve the magnitude of the predictive variance, we update our prior precision to be $\lambda_S = \lambda \cdot S/D$ (see App. C for more details). All weights not belonging to the chosen subnetwork are fixed at their MAP values. Note that this whole procedure (i.e. Steps #1-#3) is a perfectly valid mixed inference strategy: We perform full Laplace inference over the selected subnetwork and MAP inference over all remaining weights. The resulting approximate posterior (4) is

$$q_S(\mathbf{w}) \overset{(14)}{=} \mathcal{N}(\mathbf{w}_S; \widehat{\mathbf{w}}_S, \widetilde{\mathbf{H}}_S^{-1}) \textstyle\prod_r \delta(\mathrm{w}_r - \widehat{\mathrm{w}}_r)\,. \tag{16}$$

Given a sufficiently small subnetwork $\mathbf{w}_S$, it is feasible to store and invert $\widetilde{\mathbf{H}}_S$. In particular, naively storing and inverting the *full* GGN $\widetilde{\mathbf{H}}$ scales as $\mathcal{O}(D^2)$ and $\mathcal{O}(D^3)$, respectively. Using the subnetwork GGN $\widetilde{\mathbf{H}}_S$ instead reduces this burden to $\mathcal{O}(S^2)$ and $\mathcal{O}(S^3)$, respectively. In our experiments, $S \ll D$ with our subnetworks representing less that 1% of the total weights. Note that quadratic/cubic scaling in S is unavoidable if we are to capture weight correlations.

Step #4: Prediction, Fig. 1 (d). Perform a local linearization of the NN (see Section 3) while fixing $\mathbf{w}_r$ to $\widehat{\mathbf{w}}_r$:

$$\boldsymbol{f}_{\text{lin}}(\mathbf{x}, \mathbf{w}_S) = \boldsymbol{f}(\mathbf{x}, \widehat{\mathbf{w}}) + \widehat{\mathbf{J}}_S(\mathbf{x})(\mathbf{w}_S - \widehat{\mathbf{w}}_S)\,, \tag{17}$$

where $\widehat{\mathbf{J}}_S(\mathbf{x}) = \partial \boldsymbol{f}(\mathbf{x}, \widehat{\mathbf{w}}_S)/\partial \widehat{\mathbf{w}}_S \in \mathbb{R}^{O\times S}$. Following (12) and (13), the corresponding predictive distributions are

$$p(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D}) = \mathcal{N}(\mathbf{y}^*; \boldsymbol{f}(\mathbf{x}^*, \widehat{\mathbf{w}}), \Sigma_S(\mathbf{x}^*) + \sigma^2\mathbf{I}) \tag{18}$$

for regression and

$$p(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D}) \approx \text{softmax}\left(\frac{\boldsymbol{f}(\mathbf{x}^*, \widehat{\mathbf{w}})}{\sqrt{1 + \frac{\pi}{8}\text{diag}(\Sigma_S(\mathbf{x}^*))}}\right) \tag{19}$$

for classification, where $\Sigma(\mathbf{x}^*)$ in (12) and (13) is substituted with $\Sigma_S(\mathbf{x}^*) = \widehat{\mathbf{J}}_S(\mathbf{x}^*)^T \widetilde{\mathbf{H}}_S^{-1} \widehat{\mathbf{J}}_S(\mathbf{x}^*)$.

5. Subnetwork Selection

Ideally, we would like to choose a subnetwork such that the induced predictive posterior distribution is as close as possible to the predictive posterior provided by inference over the full network (11). This discrepancy between stochastic processes is often quantified through the functional Kullback-Leibler (KL) divergence (Sun et al., 2019; Burt et al., 2020):

$$\sup_{n\in\mathbb{N}, \mathbf{X}^*\in\mathcal{X}^n} D_{\text{KL}}(p_S(\mathbf{y}^*|\mathbf{X}^*, \mathcal{D}) \,||\, p(\mathbf{y}^*|\mathbf{X}^*, \mathcal{D})), \tag{20}$$

where p_S denotes the subnetwork predictive posterior and $\mathcal{X}^n$ denotes a finite measurement set of n elements. Regrettably, reasoning directly in function space is a difficult task (Nalisnick & Smyth, 2018; Pearce et al., 2019; Sun et al., 2019; Antorán et al., 2020; Nalisnick et al., 2020; Burt et al., 2020). Instead we focus our attention on weight space.

In weight space, our aim is to minimise the discrepancy between the exact posterior over the full network (1) and the subnetwork approximate posterior (4). This provides two challenges. Firstly, computing the exact posterior distribution remains intractable. Secondly, common discrepancies, like the KL divergence or the Hellinger distance, are not well defined for the Dirac delta distributions found in (4).

To solve the first issue, we again resort to local linearization, introduced in Section 3. The true posterior for the linearized model is Gaussian or approximately Gaussian[2]:

$$p(\mathbf{w}|\mathcal{D}) \simeq \mathcal{N}(\mathbf{w}; \widehat{\mathbf{w}}, \widetilde{\mathbf{H}}^{-1})\,. \tag{21}$$

We solve the second issue by choosing the squared 2-Wasserstein distance, which is well defined for distributions with disjoint support. For the case of a full covariance Gaussian (21) and a product of a full covariance Gaussian with Dirac deltas (16), this metric takes the following form:

$$\begin{aligned} &W_2(p(\mathbf{w}|\mathcal{D}), q_S(\mathbf{w}))^2 \qquad (22)\\ &= \text{Tr}\left(\widetilde{\mathbf{H}}^{-1} + \widetilde{\mathbf{H}}_{S+}^{-1} - 2\left(\widetilde{\mathbf{H}}_{S+}^{-1/2}\widetilde{\mathbf{H}}^{-1}\widetilde{\mathbf{H}}_{S+}^{-1/2}\right)^{1/2}\right)\,, \end{aligned}$$

[2]When not making predictions with the linearized model, the Gaussian posterior would represent a crude approximation.

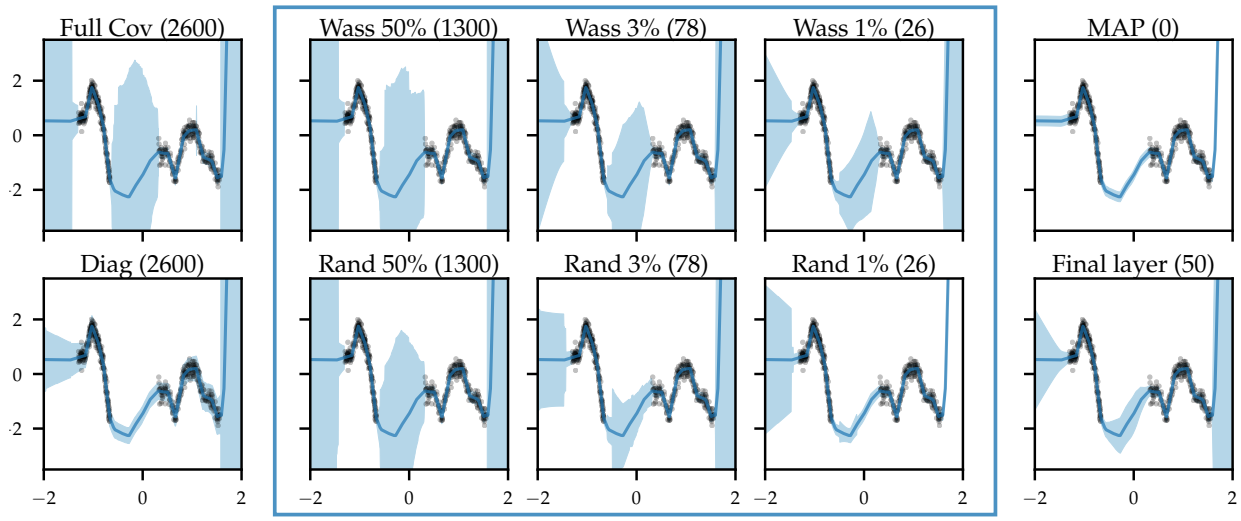


Figure 2. Predictive distributions (mean ± std) for 1D regression. The numbers in parentheses denote the number of parameters over which inference was done (out of 2600 in total). The blue box highlights subnetwork inference using Wasserstein (top) and random (bottom) subnetwork selection. Wasserstein subnetwork inference maintains richer predictive uncertainties at smaller parameter counts.

where the covariance matrix $\widetilde{\mathbf{H}}^{-1}_{S+}$ is equal to $\widetilde{\mathbf{H}}^{-1}_{S}$ padded with zeros at the positions corresponding to w_r, matching the shape of $\widetilde{\mathbf{H}}^{-1}$. See App. B for details.

Finding the subset $\mathbf{w}_S \in \mathbb{R}^S$ of size S that minimizes (22) would be combinatorially difficult, as the contribution of each weight depends on every other weight. To address this issue, we make an independence assumption among weights, resulting in the simplified objective

$$W_2(p(\mathbf{w}|\mathcal{D}), q_S(\mathbf{w}))^2 \approx \sum_{d=1}^{D} \sigma_d^2(1 - m_d) \,, \tag{23}$$

where σ_d^2 is the *marginal variance* of the d^{th} weight, and $m_d = 1$ if $\mathrm{w}_d \in \mathbf{w}_S$ and 0 otherwise (see App. B). The objective (23) is trivially minimized by a subnetwork containing the S weights with highest variances. This is related to common magnitude-based weight pruning methods (Cheng et al., 2017). The main difference is that our selection strategy involves weight *variances* rather than *magnitudes* as we target predictive uncertainty rather than accuracy.

In practice, even computing the marginal variances (i.e. the diagonal of $\widetilde{\mathbf{H}}^{-1}$) is intractable, as it requires storing and inverting the GGN $\widetilde{\mathbf{H}}$. However, we can approximate posterior marginal variances with the diagonal Laplace approximation $\mathrm{diag}(\widetilde{\mathbf{H}}^{-1}) \approx \mathrm{diag}(\widetilde{\mathbf{H}})^{-1}$ (Denker & LeCun, 1990; Kirkpatrick et al., 2017), diagonal SWAG (Maddox et al., 2019), or even mean-field variational inference (Blundell et al., 2015; Osawa et al., 2019). In this work we rely on the former two, as the the latter involves larger overhead.

It may seem that we have resorted to the poorly performing diagonal assumptions that we sought to avoid in the first place (Ovadia et al., 2019; Foong et al., 2019a; Ashukha et al., 2020). However, there is a key difference. We make the diagonal assumption during *subnetwork selection* rather than *inference*; we do full covariance inference over $\mathbf{w}_S$. In Section 6, we provide evidence that making a diagonal assumption during subnetwork selection is reasonable by showing that 1) it is substantially less harmful to predictive performance than making the same assumption during inference, and 2) it outperforms random subnetwork selection.

6. Experiments

We empirically assess the effectiveness of subnetwork inference compared to methods that do less expressive inference over the full network as well as state-of-the-art methods for uncertainty quantification in deep learning. We consider three benchmark settings: 1) small-scale toy regression, 2) medium-scale tabular regression, and 3) image classification with ResNet-18. Further experimental results and setup details are presented in App. A and App. D, respectively.

6.1. How does Subnetwork Inference preserve Posterior Predictive Uncertainty?

We first assess how the predictive distribution of a full-covariance Gaussian posterior over a selected subnetwork qualitatively compares to that obtained from 1) a full-covariance Gaussian over the *full* network (*Full Cov*), 2) a *factorised* Gaussian posterior over the full network (*Diag*), 3) a full-covariance Gaussian over only the (*Final layer*) of the network (Snoek et al., 2015), and 4) a point estimate (*MAP*). For subnetwork inference, we consider both Wasserstein (*Wass*) (as described in Section 5) and uniform

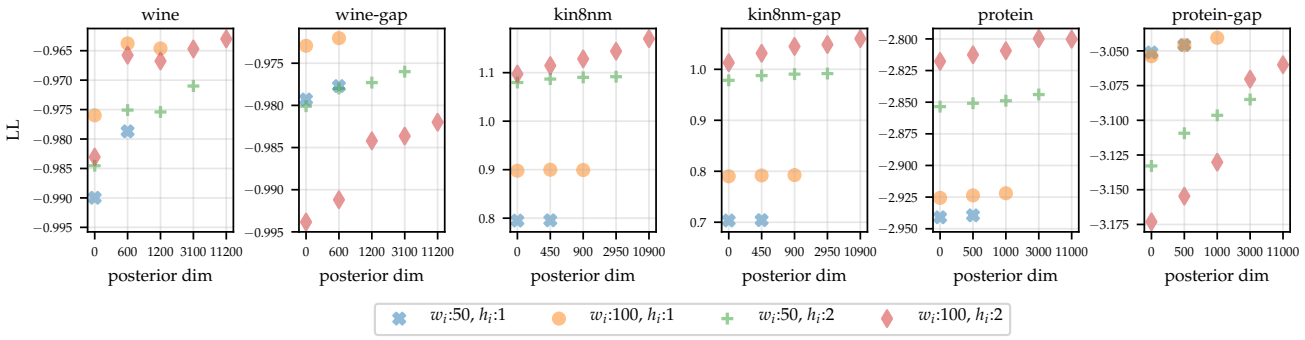


Figure 3. Mean test log-likelihood values obtained on UCI datasets across all splits. Different markers indicate models with different numbers of weights. The horizontal axis indicates the number of weights over which full covariance inference is performed. 0 corresponds to MAP parameter estimation, and the rightmost setting for each marker corresponds to full network inference.

random subnetwork selection (*Rand*) to obtain subnetworks that comprise of only 50%, 3% and 1% of the model parameters. For this toy example, it is tractable to compute exact posterior marginal variances to guide subnetwork selection.

Our NN consists of 2 ReLU hidden layers with 50 hidden units each. We employ a homoscedastic Gaussian likelihood function where the noise variance is optimised with maximum likelihood. We use GGN-Laplace inference over network weights (not biases) in combination with the linearized predictive distribution in (18). Thus, all approaches considered share their predictive mean, allowing better comparison of their uncertainty estimates. We set the full network prior precision to $\lambda = 3$ (a value which we find to work well empirically) and set $\lambda_S = \lambda \cdot {}^{S}/_{D}$.

We use a synthetic 1D regression task with two separated clusters of inputs (Antorán et al., 2020), allowing us to probe for 'in-between' uncertainty (Foong et al., 2019b). Results are shown in Fig. 2. Subnetwork inference preserves more of the uncertainty of full network inference than diagonal Gaussian or final layer inference while doing inference over fewer weights. By capturing weight correlations, subnetwork inference retains uncertainty in between clusters of data. This is true for both random and Wasserstein subnetwork selection. However, the latter preserves more uncertainty with smaller subnetworks. Finally, the strong superiority to diagonal Laplace shows that making a diagonal assumption for subnetwork selection but then using a full-covariance Gaussian for inference (as we do) performs significantly better than making a diagonal assumption for the inferred posterior directly (cf. Section 5). These results suggest that **expressive inference over a carefully selected subnetwork retains more predictive uncertainty than crude approximations over the full network**.

6.2. Subnetwork Inference in Large Models vs Full Inference over Small Models

Secondly, we study how subnetwork inference in larger NNs compares to full network inference in smaller ones. We explore this by considering 4 fully connected NNs of increasing size. These have numbers of hidden layers $h_d{=}\{1, 2\}$ and hidden layer widths $w_d{=}\{50, 100\}$. For a dataset with input dimension i_d, the number of weights is given by $D{=}(i_d{+}1)w_d{+}(h_d{-}1)w_d^2$. Our 2 hidden layer, 100 hidden unit NNs have a weight count of the order 10^4. Full covariance inference in these NNs borders the limit of computational tractability on commercial hardware. We first obtain a MAP estimate of each NN's weights and our homoscedastic likelihood function's noise variance. We then perform full network GGN-Laplace inference for each NN. We also use our proposed Wassertein rule to prune every NN's weight variances such that the number of variances that remain matches the size of every smaller NN under consideration. We employ the diagonal Laplace approximation to cheaply estimate posterior marginal variances for subnetwork selection. We employ the linearization in (12) and (18) to compute predictive distributions. Consequently, NNs with the same number of weights make the same mean predictions. Increasing the number of weight variances considered will thus only increase predictive uncertainty.

We employ 3 tabular datasets of increasing size (input dimensionality, n. points): wine (11, 1439), kin8nm (8, 7373) and protein (9, 41157). We consider their standard train-test splits (Hernández-Lobato & Adams, 2015) and their gap variants (Foong et al., 2019b), designed to test for out-of-distribution uncertainty. Details are provided in App. D.4. For each split, we set aside 15% of the train data as a validation set. We use these for early stopping when finding MAP estimates and for selecting the weights' prior precision. We keep other hyperparameters fixed across all models and datasets. Results are shown in Fig. 3.

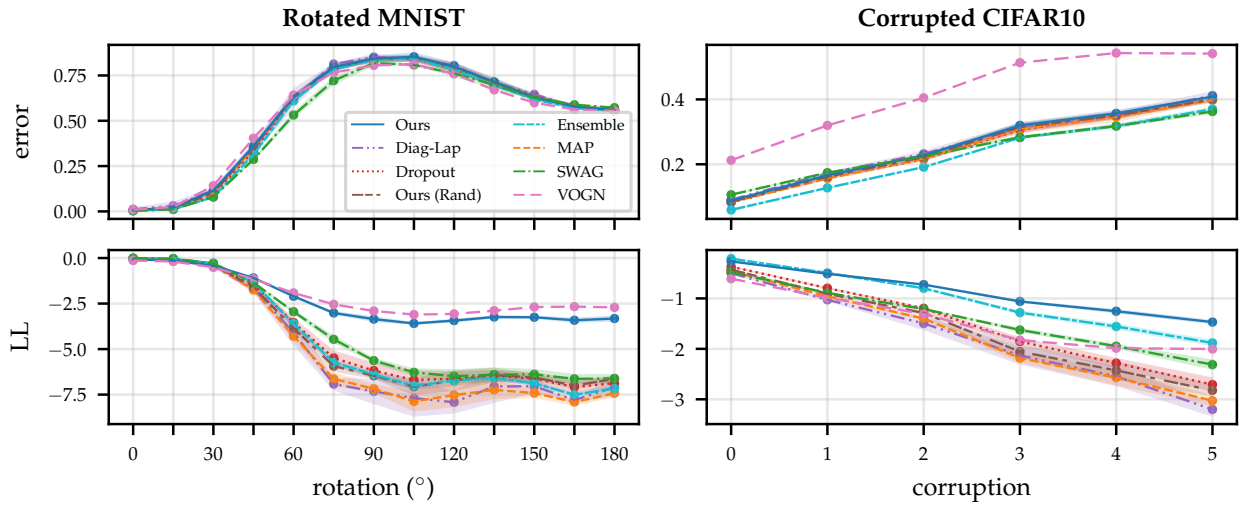


Figure 4. Results on the rotated MNIST (left) and the corrupted CIFAR (right) benchmarks, showing the mean ± std of the error (top) and log-likelihood (bottom) across three different seeds. Subnetwork inference retains better uncertainty calibration and robustness to distribution shift than point-estimated networks and other Bayesian deep learning approaches. See App. A for ECE and Brier score results.

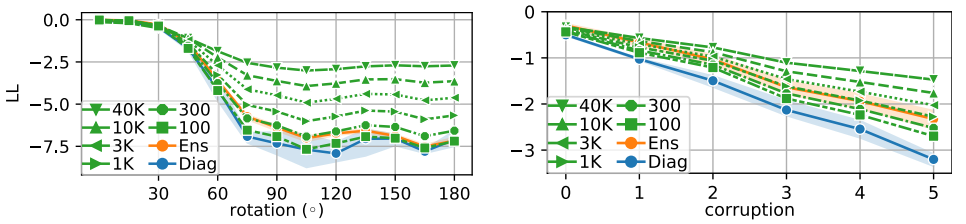
We present mean test log-likelihood (LL) values, as these take into account both accuracy and uncertainty. Larger ($w_d = 100, h_d = 2$) models tend to perform best when combined with full network inference, although Wine-gap and Protein-gap are exceptions. Interestingly, these larger models are still best when we perform inference over subnetworks of the size of smaller models. We conjecture this is due to an abundance of degenerate directions (i.e. weights) in the weight posterior NN models (Maddox et al., 2020). Full network inference in small models captures information about both useful and non-useful weights. In larger models, our subnetwork selection strategy allows us to dedicate a larger proportion of our resources to modelling informative weight variances and covariances. In 3 out of 6 datasets, we find abrupt increases in LL as we increase the number of weights over which we perform inference, followed by a plateau. Such plateaus might be explained by most of the informative weight variances having already been accounted for. Considering that the cost of computing the GGN dominates that of NN training, these results suggest that, **given the same amount of compute, it is better to perform subnetwork inference in larger models than full network inference in small ones.**

6.3. Image Classification under Distribution Shift

We now assess the robustness of large convolutional neural networks with subnetwork inference to distribution shift on image classification tasks compared to the following baselines: point-estimated networks (MAP), Bayesian deep learning methods that do less expressive inference over the full network: MC Dropout (Gal & Ghahramani, 2016), diagonal Laplace, VOGN (Osawa et al., 2019) (all of which assume factorisation of the weight posterior), and SWAG (Maddox et al., 2019) (which assumes a diagonal plus low-rank posterior). We also benchmark deep ensembles (Lakshminarayanan et al., 2017). The latter is considered state-of-the-art for uncertainty quantification in deep learning (Ovadia et al., 2019; Ashukha et al., 2020). We use ensembles of 5 NNs, as suggested by (Ovadia et al., 2019), and 16 samples for MC Dropout, diagonal Laplace and SWAG. We use a Dropout probability of 0.1 and a prior precision of $\lambda = 4 \times 10^4$ for diagonal Laplace, found via grid search. We apply all approaches to ResNet-18 (He et al., 2016), which is composed of an input convolutional block, 8 residual blocks and a linear layer, for a total of 11,168,000 parameters.

For subnetwork inference, we compute the linearized predictive distribution in (19). We use Wasserstein subnetwork selection to retain only 0.38% of the weights, yielding a subnetwork with only 42,438 weights. This is the largest subnetwork for which we can tractably compute a full covariance matrix. Its size is $42,438^2 \times 4$ Bytes ≈ 7.2 GB. We use diagonal SWAG (Maddox et al., 2019) to estimate the marginal weight variances needed for subnetwork selection. We tried diagonal Laplace but found that the selected weights where those where the Jacobian of the NN evaluated at the train points was always zero (i.e. dead ReLUs). The posterior variance of these weights is large as it matches the prior. However, these weights have little effect on the NN function. SWAG does not suffer from this problem as it disregards weights with zero training gradients. We use a prior precision of $\lambda = 500$, found via grid search.

To assess to importance of principled subnetwork selection, we also consider the baseline where we select the subnetwork uniformly at random (called *Ours (Rand)*). We per-



(a) Rotated MNIST (b) Corrupted CIFAR10

Subnet Size	Memory
11.2M (100%)	500**TB**
40K (0.36%)	6.4**GB**
1K (0.01%)	4.0**MB**
100 (0.001%)	40**KB**

(c) Memory Footprints

Figure 5. Log-likelihoods of our method with subnetwork sizes between 100-40K using ResNet-18 on rotated MNIST (left) and corrupted CIFAR10 (middle), vs. **Ens**embles and **Diag**onal Laplace, and respective covariance matrix memory footprints (right). For all subnetwork sizes, we use the same hyperparameters as in Section 6.3 (i.e. no individual tuning per size). Performance degrades smoothly with subnetwork size, but our method retains strong calibration even with very small subnetworks (requiring only marginal extra memory).

form the following two experiments, with results in Fig. 4.

Rotated MNIST: Following (Ovadia et al., 2019; Antorán et al., 2020), we train all methods on MNIST and evaluate their predictive distributions on increasingly rotated digits. While all methods perform well on the original MNIST test set, their accuracy degrades quickly for rotations larger than 30 degrees. In terms of LL, ensembles perform best out of our baselines. Subnetwork inference obtains significantly larger LL values than almost all baselines, including ensembles. The only exception is VOGN, which achieves slightly better performance. It was also observed in (Ovadia et al., 2019) that mean-field variational inference (which VOGN is an instance of) is very strong on MNIST, but its performance deteriorates on larger datasets. Subnetwork inference makes accurate predictions in-distribution while assigning higher uncertainty than the baselines to out-of-distribution points.

Corrupted CIFAR: Again following (Ovadia et al., 2019; Antorán et al., 2020), we train on CIFAR10 and evaluate on data subject to 16 different corruptions with 5 levels of intensity each (Hendrycks & Dietterich, 2019). Our approach matches a MAP estimated network in terms of predictive error as local linearization makes their predictions the same. Ensembles and SWAG are the most accurate. Even so, subnetwork inference differentiates itself by being the least overconfident, outperforming all baselines in terms of log-likelihood at all corruption levels. Here, VOGN performs rather badly; while this might appear to contrast its strong performance on the MNIST benchmark, the behaviour that mean-field VI performs well on MNIST but poorly on larger datasets was also observed in (Ovadia et al., 2019).

Furthermore, on both benchmarks, we find that randomly selecting the subnetwork performs substantially worse than using our more sophisticated Wasserstein subnetwork selection strategy. This highlights the importance of the way the subnetwork is selected. Overall, these results suggest that **subnetwork inference results in better uncertainty calibration and robustness to distribution shift than other popular uncertainty quantification approaches**.

What about smaller subnetworks? One might wonder if a subnetwork of $\sim$40K weights is actually necessary. In Fig. 5, we show that one can also retain strong calibration with significantly smaller subnetworks. Full covariance inference in a ResNet-18 would require storing $\sim 11.2\text{M}^2$ params ($\sim$500TB). Subnet inference reduces the cost (on top of MAP) to as little as 1K^2 params ($\sim$4.0MB) while remaining competitive with deep ensembles. This suggests that subnetwork inference can allow otherwise intractable inference methods to be applied to even larger NNs.

7. Scope and Limitations

Jacobian computation in multi-output models remains challenging. With reverse mode automatic differentiation used in most deep learning frameworks, it requires as many backward passes as there are model outputs. This prevents using linearized Laplace in settings like semantic segmentation (Liu et al., 2019) or classification with large numbers of classes (Deng et al., 2009). Note that this issue applies to the linearized Laplace method and that other inference methods, without this limitation, could be used in our framework.

The choice of prior precision λ determines the performance of the Laplace approximation to a large degree. Our proposed scheme to update λ for subnetworks relies on having a sensible parameter setting for the full network. Since inference in the full network is often intractable, currently the best approach for choosing λ is cross validation using the subnetwork approximation directly.

The space requirements for the Hessian limit the maximum number of subnetwork weights. For example, storing a Hessian for 40K weights requires around 6.4GB of memory. For very large models, like modern transformers, tractable subnetworks would represent a vanishingly small proportion of the weights. While we demonstrated that strong performance does not necessarily require large subnetworks (see Fig. 5), finding better subnetwork selection strategies remains a key direction for future research.

8. Related Work

Bayesian Deep Learning. There have significant efforts to characterise the posterior distribution over NN weights $p(\mathbf{w}|\mathcal{D})$. To this day, Hamiltonian Monte Carlo (Neal, 1995) remains the golden standard for approximate inference in BNNs. Although asymptotically unbiased, sampling based approaches are difficult to scale to the large datasets (Betancourt, 2015). As a result, approaches which find the best surrogate posterior among an approximating family (most often Gaussians) have gained popularity. The first of these was the Laplace approximation, introduced by MacKay (1992), who also proposed approximating the predictive posterior with that of the linearised model (Khan et al., 2019; Immer et al., 2020). The popularisation of larger NN models has made surrogate distributions that capture correlations between weights computationally intractable. Thus, most modern methods make use of the mean field assumption (Blundell et al., 2015; Hernández-Lobato & Adams, 2015; Gal & Ghahramani, 2016; Mishkin et al., 2018; Osawa et al., 2019). This comes at the cost of limited expressivity (Foong et al., 2019a) and empirical under-performance (Ovadia et al., 2019; Antorán et al., 2020). We note that, Farquhar et al. (2020) argue that in deeper networks the mean-field assumption should not be restrictive. Our empirical results seem to contradict this proposition. We find that scaling up approximations that *do* consider weight correlations (e.g. MacKay (1992); Louizos & Welling (2016); Maddox et al. (2019); Ritter et al. (2018)) by lowering the dimensionality of the weight space outperforms diagonal approximations. We conclude that more research is warranted in this area.

Neural Linear Methods. These represent a generalised linear model in which the basis functions are defined by the $l-1$ first layers of a NN. That is, neural linear methods perform inference over only the last layer of a NN, while keeping all other layers fixed (Snoek et al., 2015; Riquelme et al., 2018; Ovadia et al., 2019; Ober & Rasmussen, 2019; Pinsler et al., 2019; Kristiadi et al., 2020). They can also be viewed as a special case of subnetwork inference, in which the subnetwork is simply defined to be the last NN layer.

Inference over Subspaces. The subfield of NN pruning aims to increase the computational efficiency of NNs by identifying the smallest subset of weights which are required to make accurate predictions; see e.g. (Frankle & Carbin, 2019; Wang et al., 2020). Our work differs in that it retains all NN weights but aims to find a small subset over which to perform probabilistic reasoning. More closely related work to ours is that of (Izmailov et al., 2019), who propose to perform inference over a low-dimensional subspace of weights; e.g. one constructed from the principal components of the SGD trajectory. Moreover, several recent approaches use low-rank parameterizations of approximate posteriors in the context of variational inference (Rossi et al., 2019; Swiatkowski et al., 2020; Dusenberry et al., 2020). This could also be viewed as doing inference over an implicit subspace of weight space. In contrast, we propose a technique to find subsets of weights which are relevant to predictive uncertainty, i.e., we identify axis aligned subspaces.

9. Conclusion

Our work has three main findings: 1) modelling weight correlations in NNs is crucial to obtaining reliable predictive posteriors, 2) given these correlations, unimodal approximations of the posterior can be competitive with approximations that assign mass to multiple modes (e.g. deep ensembles), 3) inference does not need to be performed over all the weights in order to obtain reliable predictive posteriors.

We use these insights to develop a framework for scaling Bayesian inference to NNs with a large number of weights. We approximate the posterior over a subset of the weights while keeping all others deterministic. Computational cost is decoupled from the total number of weights, allowing us to conveniently trade it off with the quality of approximation. This allows us to use more expressive posterior approximations, such as full-covariance Gaussian distributions.

Linearized Laplace subnetwork inference can be applied post-hoc to any pre-trained model, making it particularly attractive for practical use. Our empirical analysis suggests that this method 1) is more expressive and retains more uncertainty than crude approximations over the full network, 2) allows us to employ larger NNs, which fit a broader range of functions, without sacrificing the quality of our uncertainty estimates, and 3) is competitive with state-of-the-art uncertainty quantification methods, like deep ensembles.

We are excited to investigate combining subnetwork inference with different approximate inference methods, develop better subnetwork selection strategies and further explore the properties of subnetworks on the predictive distribution.

Acknowledgments

We thank Matthias Bauer, Alexander Immer, Andrew Y. K. Foong and Robert Pinsler for helpful discussions. ED acknowledges funding from the EPSRC and Qualcomm. JA acknowledges support from Microsoft Research, through its PhD Scholarship Programme, and from the EPSRC. JUA acknowledges funding from the EPSRC and the Michael E. Fisher Studentship in Machine Learning. This work has been performed using resources provided by the Cambridge Tier-2 system operated by the University of Cambridge Research Computing Service (http://www.hpc.cam.ac.uk) funded by EPSRC Tier-2 capital grant EP/P020259/1.

References

Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., and Mané, D. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

Antorán, J., Allingham, J. U., and Hernández-Lobato, J. M. Depth uncertainty in neural networks, 2020.

Ashukha, A., Lyzhov, A., Molchanov, D., and Vetrov, D. P. Pitfalls of in-domain uncertainty estimation and ensembling in deep learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

Betancourt, M. The fundamental incompatibility of scalable hamiltonian monte carlo and naive data subsampling. volume 37 of *Proceedings of Machine Learning Research*, pp. 533–540, Lille, France, 07–09 Jul 2015. PMLR. URL http://proceedings.mlr.press/v37/betancourt15.html.

Bhatt, U., Zhang, Y., Antorán, J., Liao, Q. V., Sattigeri, P., Fogliato, R., Melançon, G. G., Krishnan, R., Stanley, J., Tickoo, O., et al. Uncertainty as a form of transparency: Measuring, communicating, and using uncertainty. *arXiv preprint arXiv:2011.07586*, 2020.

Bishop, C. M. *Pattern recognition and machine learning*. springer, 2006.

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. Weight Uncertainty in Neural Networks. In *Proceedings of The 32nd International Conference on Machine Learning (ICML)*, pp. 1613–1622, 2015.

Burt, D. R., Ober, S. W., Garriga-Alonso, A., and van der Wilk, M. Understanding variational inference in function-space, 2020.

Cheng, Y., Wang, D., Zhou, P., and Zhang, T. A survey of model compression and acceleration for deep neural networks. *arXiv preprint arXiv:1710.09282*, 2017.

Deng, J., Dong, W., Socher, R., Li, L., Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Denker, J. S. and LeCun, Y. Transforming neural-net output levels to probability distributions. In *Proceedings of the 3rd International Conference on Neural Information Processing Systems*, NIPS'90, pp. 853–859, San Francisco, CA, USA, 1990. Morgan Kaufmann Publishers Inc. ISBN 1558601848.

Dua, D. and Graff, C. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Dusenberry, M. W., Jerfel, G., Wen, Y., Ma, Y.-a., Snoek, J., Heller, K., Lakshminarayanan, B., and Tran, D. Efficient and scalable bayesian neural nets with rank-1 factors. *arXiv preprint arXiv:2005.07186*, 2020.

Farquhar, S., Smith, L., and Gal, Y. Liberty or depth: Deep bayesian neural nets do not need complex weight posterior approximations. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Filos, A., Farquhar, S., Gomez, A. N., Rudner, T. G., Kenton, Z., Smith, L., Alizadeh, M., de Kroon, A., and Gal, Y. Benchmarking bayesian deep learning with diabetic retinopathy diagnosis. *Preprint*, 2019.

Foong, A. Y., Burt, D. R., Li, Y., and Turner, R. E. On the expressiveness of approximate inference in bayesian neural networks. *arXiv*, pp. arXiv–1909, 2019a.

Foong, A. Y., Li, Y., Hernández-Lobato, J. M., and Turner, R. E. In-between uncertainty in bayesian neural networks. *ICML Workshop on Uncertainty and Robustness in Deep Learning*, 2019b.

Fort, S., Hu, H., and Lakshminarayanan, B. Deep ensembles: A loss landscape perspective. *arXiv preprint arXiv:1912.02757*, 2019.

Frankle, J. and Carbin, M. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2019.

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

Ghahramani, Z. Probabilistic machine learning and artificial intelligence. *Nature*, 521(7553):452–459, 2015.

Gibbs, M. N. *Bayesian Gaussian processes for regression and classification*. PhD thesis, Citeseer, 1998.

Givens, C. R., Shortt, R. M., et al. A class of wasserstein metrics for probability distributions. *The Michigan Mathematical Journal*, 31(2):231–240, 1984.

Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. *Deep learning*, volume 1. MIT Press, 2016.

Goyal, P., Dollár, P., Girshick, R. B., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., and He, K. Accurate, large minibatch SGD: Training ImageNet in 1 hour. *CoRR*, abs/1706.02677, 2017.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1321–1330. JMLR. org, 2017.

He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pp. 1026–1034. IEEE Computer Society, 2015.

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

Hendrycks, D. and Dietterich, T. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

Hernández-Lobato, J. M. and Adams, R. Probabilistic backpropagation for scalable learning of bayesian neural networks. In *International Conference on Machine Learning*, pp. 1861–1869, 2015.

Immer, A., Korzepa, M., and Bauer, M. Improving predictions of bayesian neural networks via local linearization, 2020.

Izmailov, P., Maddox, W. J., Kirichenko, P., Garipov, T., Vetrov, D., and Wilson, A. G. Subspace inference for bayesian deep learning. In *35th Conference on Uncertainty in Artificial Intelligence, UAI 2019*, 2019.

Khan, M. E., Nielsen, D., Tangkaratt, V., Lin, W., Gal, Y., and Srivastava, A. Fast and scalable bayesian deep learning by weight-perturbation in adam. *arXiv preprint arXiv:1806.04854*, 2018.

Khan, M. E. E., Immer, A., Abedi, E., and Korzepa, M. Approximate inference turns deep networks into gaussian processes. In *Advances in neural information processing systems*, pp. 3094–3104, 2019.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D., and Hadsell, R. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. doi: 10.1073/pnas.1611835114.

Kristiadi, A., Hein, M., and Hennig, P. Being bayesian, even just a bit, fixes overconfidence in relu networks. *arXiv preprint arXiv:2002.10118*, 2020.

Krizhevsky, A. and Hinton, G. Learning Multiple Layers of Features from Tiny Images. Technical report, University of Toronto, 2009.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pp. 6402–6413, 2017.

Lawrence, N. D. *Variational inference in probabilistic models*. PhD thesis, University of Cambridge, 2001.

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

Liu, X., Deng, Z., and Yang, Y. Recent progress in semantic image segmentation. 52(2), 2019. ISSN 0269-2821. doi: 10.1007/s10462-018-9641-3. URL https://doi.org/10.1007/s10462-018-9641-3.

Lobacheva, E., Chirkova, N., Kodryan, M., and Vetrov, D. P. On power laws in deep ensembles. *Advances in Neural Information Processing Systems*, 33, 2020.

Louizos, C. and Welling, M. Structured and efficient variational deep learning with matrix gaussian posteriors. In *International Conference on Machine Learning*, pp. 1708–1716, 2016.

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

Maddox, W. J., Izmailov, P., Garipov, T., Vetrov, D. P., and Wilson, A. G. A simple baseline for bayesian uncertainty in deep learning. In *Advances in Neural Information Processing Systems*, pp. 13132–13143, 2019.

Maddox, W. J., Benton, G., and Wilson, A. G. Rethinking parameter counting in deep models: Effective dimensionality revisited. *arXiv preprint arXiv:2003.02139*, 2020.

Martens, J. New insights and perspectives on the natural gradient method. *arXiv preprint arXiv:1412.1193*, 2014.

Martens, J. *Second-order optimization for neural networks*. University of Toronto (Canada), 2016.

Martens, J. and Sutskever, I. Learning recurrent neural networks with hessian-free optimization. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 1033–1040. Citeseer, 2011.

Mishkin, A., Kunstner, F., Nielsen, D., Schmidt, M., and Khan, M. E. Slang: Fast structured covariance approximations for bayesian deep learning with natural gradient. In *Advances in Neural Information Processing Systems*, pp. 6245–6255, 2018.

Nalisnick, E., Matsukawa, A., Whye Teh, Y., Gorur, D., and Lakshminarayanan, B. Do Deep Generative Models Know What They Don't Know? In *International Conference on Learning Representations (ICLR)*, 2019.

Nalisnick, E. T. and Smyth, P. Learning priors for invariance. In Storkey, A. J. and Pérez-Cruz, F. (eds.), *International Conference on Artificial Intelligence and Statistics, AISTATS 2018, 9-11 April 2018, Playa Blanca, Lanzarote, Canary Islands, Spain*, volume 84 of *Proceedings of Machine Learning Research*, pp. 366–375. PMLR, 2018. URL http://proceedings.mlr.press/v84/nalisnick18a.html.

Nalisnick, E. T., Gordon, J., and Hernández-Lobato, J. M. Predictive complexity priors. *CoRR*, abs/2006.10801, 2020. URL https://arxiv.org/abs/2006.10801.

Neal, R. M. *Bayesian Learning for Neural Networks*. PhD thesis, CAN, 1995. AAINN02676.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading Digits in Natural Images with Unsupervised Feature Learning. In *NeurIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

Nguyen, A., Yosinski, J., and Clune, J. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 427–436, 2015.

Ober, S. W. and Rasmussen, C. E. Benchmarking the neural linear model for regression. *arXiv preprint arXiv:1912.08416*, 2019.

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

Ovadia, Y., Fertig, E., Lakshminarayanan, B., Nowozin, S., Sculley, D., Dillon, J., Ren, J., Nado, Z., and Snoek, J. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems*, pp. 13969–13980, 2019.

Pearce, T., Tsuchida, R., Zaki, M., Brintrup, A., and Neely, A. Expressive priors in bayesian neural networks: Kernel combinations and periodic functions. In Globerson, A. and Silva, R. (eds.), *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI 2019, Tel Aviv, Israel, July 22-25, 2019*, volume 115 of *Proceedings of Machine Learning Research*, pp. 134–144. AUAI Press, 2019. URL http://proceedings.mlr.press/v115/pearce20a.html.

Pinsler, R., Gordon, J., Nalisnick, E., and Hernández-Lobato, J. M. Bayesian batch active learning as sparse subset approximation. In *Advances in Neural Information Processing Systems*, pp. 6359–6370, 2019.

Riquelme, C., Tucker, G., and Snoek, J. Deep bayesian bandits showdown: An empirical comparison of bayesian deep networks for thompson sampling. In *International Conference on Learning Representations*, 2018.

Ritter, H., Botev, A., and Barber, D. A scalable laplace approximation for neural networks. In *International Conference on Learning Representations*, 2018.

Rossi, S., Marmin, S., and Filippone, M. Walsh-hadamard variational inference for bayesian deep learning. *arXiv preprint arXiv:1905.11248*, 2019.

Snoek, J., Rippel, O., Swersky, K., Kiros, R., Satish, N., Sundaram, N., Patwary, M., Prabhat, M., and Adams, R. Scalable bayesian optimization using deep neural networks. In *International conference on machine learning*, pp. 2171–2180, 2015.

Sun, S., Zhang, G., Shi, J., and Grosse, R. Functional variational bayesian neural networks. *arXiv preprint arXiv:1903.05779*, 2019.

Swiatkowski, J., Roth, K., Veeling, B. S., Tran, L., Dillon, J. V., Mandt, S., Snoek, J., Salimans, T., Jenatton, R., and Nowozin, S. The k-tied normal distribution: A compact parameterization of gaussian mean field posteriors in bayesian neural networks. *arXiv preprint arXiv:2002.02655*, 2020.

Wang, C., Zhang, G., and Grosse, R. Picking winning tickets before training by preserving gradient flow. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SkgsACVKPH.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. 2017.

Zagoruyko, S. and Komodakis, N. Wide residual networks. In Wilson, R. C., Hancock, E. R., and Smith, W. A. P. (eds.), *Proceedings of the British Machine Vision Conference 2016, BMVC 2016, York, UK, September 19-22, 2016*. BMVA Press, 2016.