The k-tied Normal Distribution: A Compact Parameterization of Gaussian Mean Field Posteriors in Bayesian Neural Networks

Jakub Swiatkowski [1+]   **Kevin Roth** [2+]   **Bastiaan S. Veeling** [3 4+]   **Linh Tran** [5+]   **Joshua V. Dillon** [4]   **Jasper Snoek** [4]
Stephan Mandt [6+]   **Tim Salimans** [4]   **Rodolphe Jenatton** [4]   **Sebastian Nowozin** [7+]

Abstract

Variational Bayesian Inference is a popular methodology for approximating posterior distributions over Bayesian neural network weights. Recent work developing this class of methods has explored ever richer parameterizations of the approximate posterior in the hope of improving performance. In contrast, here we share a curious experimental finding that suggests instead restricting the variational distribution to a more compact parameterization. For a variety of deep Bayesian neural networks trained using Gaussian mean-field variational inference, we find that the posterior standard deviations consistently exhibit strong low-rank structure after convergence. This means that by decomposing these variational parameters into a low-rank factorization, we can make our variational approximation more compact without decreasing the models' performance. Furthermore, we find that such factorized parameterizations improve the signal-to-noise ratio of stochastic gradient estimates of the variational lower bound, resulting in faster convergence.

1. Introduction

Bayesian neural networks (MacKay, 1992; Neal, 1993) are a popular class of deep learning models. The most widespread approach for training these models relies on variational inference (Peterson, 1987; Hinton & Van Camp, 1993), a training paradigm that approximates a Bayesian posterior with a simpler class of distributions by solving an optimization problem. The common wisdom is that more expressive

---
[+]Work done while at Google   [1]University of Warsaw [2]ETH Zurich [3]University of Amsterdam [4]Google Research [5]Imperial College London [6]University of California, Irvine [7]Microsoft Research. Correspondence to: Jakub Swiatkowski <jakub.swiatkowski@mimuw.edu.pl>.

distributions lead to better posterior approximations and ultimately to better model performance. This paper puts this into question and instead finds that for Bayesian neural networks, more restrictive classes of distributions, based on low-rank factorizations, can outperform the common mean-field family.

Bayesian Neural Networks explicitly represent their parameter-uncertainty by forming a *posterior distribution* over model parameters, instead of relying on a single point estimate for making predictions, as is done in traditional deep learning. For neural network weights $\mathbf{w}$, features $\mathbf{x}$ and labels $\mathbf{y}$, the posterior distribution $p(\mathbf{w}|\mathbf{x}, \mathbf{y})$ is computed using Bayes' rule, which multiplies the prior distribution $p(\mathbf{w})$ and data likelihood $p(\mathbf{y}|\mathbf{w}, \mathbf{x})$ and renormalizes. When predicting with Bayesian neural networks, we form an average over model predictions where each prediction is generated using a set of parameters that is randomly sampled from the posterior distribution. This can be viewed as a type of *ensembling*, of which various types have proven highly effective in deep learning (see e.g. Goodfellow et al., 2016, sec 7.11).

Besides offering improved predictive performance over single models, Bayesian ensembles are also more robust because ensemble members will tend to make different predictions on hard examples (Raftery et al., 2005). In addition, the diversity of the ensemble represents predictive uncertainty and can be used for out-of-domain detection or other risk-sensitive applications (Ovadia et al., 2019).

Variational inference is a popular class of methods for approximating the posterior distribution $p(\mathbf{w}|\mathbf{x}, \mathbf{y})$, since the exact Bayes' rule is often intractable to compute for models of practical interest. This class of methods specifies a distribution $q_{\boldsymbol{\theta}}(\mathbf{w})$ of given parametric or functional form as the posterior approximation, and optimizes the approximation by solving an optimization problem. In particular, we minimize the Kullback-Leibler (KL) divergence D_{KL} between the variational distribution $q_{\boldsymbol{\theta}}(\mathbf{w})$ and the true posterior dis-
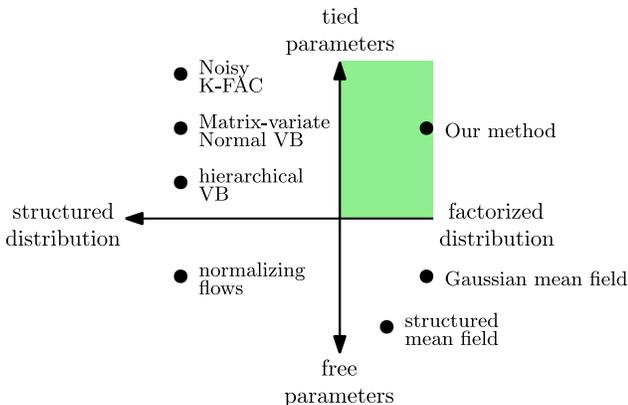
Figure 1. Approximate summarization of different variational inference methods for Bayesian deep learning. Our approach complements existing approaches by combining the mean-field assumption with a dramatic reduction in the number of parameters by weight sharing.

tribution $p(\mathbf{w}|\mathbf{x}, \mathbf{y})$, which is given by

$$
\begin{aligned}
D_{\mathrm{KL}}[q_{\boldsymbol{\theta}}(\mathbf{w})||p(\mathbf{w}|\mathbf{x}, \mathbf{y})] &= \mathbb{E}_q\left[\log \frac{q_{\boldsymbol{\theta}}(\mathbf{w})}{p(\mathbf{w}|\mathbf{x}, \mathbf{y})}\right] \\
&= \mathbb{E}_q\left[\log \frac{q_{\boldsymbol{\theta}}(\mathbf{w})}{p(\mathbf{w})p(\mathbf{y}|\mathbf{w}, \mathbf{x})/p(\mathbf{y}|\mathbf{x})}\right].
\end{aligned}
\tag{1}
$$

Here, we do not know the normalizing constant of the exact posterior $p(\mathbf{y}|\mathbf{x})$, but since this term does not depend on $\mathbf{w}$, we may ignore it for the purpose of optimizing our approximation q. We are then left with what is called the negative Evidence Lower Bound (negative ELBO):

$$
L_q = D_{\mathrm{KL}}[q_{\boldsymbol{\theta}}(\mathbf{w})||p(\mathbf{w})] - \mathbb{E}_q[\log p(\mathbf{y}|\mathbf{w}, \mathbf{x})]. \tag{2}
$$

In practice, the expectation of the log-likelihood $p(\mathbf{y}|\mathbf{w}, \mathbf{x})$ with respect to q is usually not analytically tractable and instead is estimated using Monte Carlo sampling:

$$
\begin{aligned}
\mathbb{E}_q[\log p(\mathbf{y}|\mathbf{w}, \mathbf{x})] &\approx \frac{1}{S}\sum_{s=1}^{S}\log p(\mathbf{y}|\mathbf{w}^{(s)}, \mathbf{x}), \\
\mathbf{w}^{(s)} &\sim q_{\boldsymbol{\theta}}(\mathbf{w}),
\end{aligned}
\tag{3}
$$

where the ELBO is optimized by differentiating this stochastic approximation with respect to the variational parameters $\boldsymbol{\theta}$ (Salimans et al., 2013; Kingma & Welling, 2013).

In **Gaussian Mean Field Variational Inference** (GMFVI) (Blei et al., 2017; Blundell et al., 2015), we choose the variational approximation to be a fully factorized Gaussian distribution $q = \mathcal{N}(\boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q)$ with $w_{lij} \sim \mathcal{N}(\mu_{lij}, \sigma_{lij}^2)$, where l is a layer number, and i and j are the row and column indices in the layer's weight matrix. While Gaussian Mean-Field posteriors are considered to be one of the simplest types of variational approximations, with some known limitations (Giordano et al., 2018), they scale to comparatively large models and generally provide competitive

performance (Ovadia et al., 2019). Additionally, Farquhar et al. (2020b) have found that the Mean-Field becomes a less restrictive assumption as the depth of the network increases. However, when compared to deterministic neural networks, GMFVI doubles the number of parameters and is often harder to train due to the increased noise in stochastic gradient estimates. Furthermore, despite the theoretical advantages of GMFVI over the deterministic neural networks, GMFVI suffers from over-regularization for larger networks, which leads to underfitting and often worse predictive performance in such settings (Wenzel et al., 2020).

Beyond mean-field variational inference, recent work on approximate Bayesian inference has explored ever richer parameterizations of the approximate posterior in the hope of improving the performance of Bayesian neural networks (see Figure 1). In contrast, here we study a simpler, more compactly parameterized variational approximation. Our motivation for studying this setting is to better understand the behaviour of GMFVI with the goal to address the issues with its practical applicability. Consequently, we show that the compact approximations can also work well for a variety of models. In particular we find that:

- Converged posterior standard deviations under GMFVI consistently display strong low-rank structure. This means that by decomposing these variational parameters into a low-rank factorization, we can make our variational approximation more compact without decreasing our model's performance.
- Factorized parameterizations of posterior standard deviations improve the signal-to-noise ratio of stochastic gradient estimates, and thus not only reduce the number of parameters compared to standard GMFVI, but also can lead to faster convergence.

2. Mean Field Posterior Standard Deviations Naturally Have Low-Rank Structure

In this section we show that the converged posterior standard deviations of Bayesian neural networks trained using standard GMFVI consistently display strong low-rank structure. We also show that it is possible to compress the learned posterior standard deviation matrix using a low-rank approximation without decreasing the network's performance. We first briefly introduce the mathematical notation for our GMFVI setting and the low-rank approximation that we explore. We then provide experimental results that support the two main claims of this section.

To avoid any confusion among the readers, we would like to clarify that we use the terminology "low-rank" in a particular context. While variational inference typically makes use of low-rank decompositions to compactly represent the *dense covariance* of a Gaussian variational distribution (see

numerous references in Section 4), we investigate instead underlying low-rank structures within *the already diagonal covariance* of a Gaussian fully-factorized variational distribution. Figure 2 aims to make this even more clear by illustrating the relationship between the Gaussian fully-factorized variational distribution and its "low-rank" parameterization explored in this paper. We will make this explanation more formal in the next section.

2.1. Methodology

To introduce the notation, we consider layers that consist of a linear transformation followed by a non-linearity f,

$$\mathbf{a}_l = \mathbf{h}_l \mathbf{W}_l + \mathbf{b}_l, \qquad \mathbf{h}_{l+1} = f(\mathbf{a}_l), \qquad (4)$$

where $\mathbf{W}_l \in \mathbb{R}^{m \times n}$, $\mathbf{h}_l \in \mathbb{R}^{1 \times m}$ and $\mathbf{b}_l, \mathbf{a}_l, \mathbf{h}_{l+1} \in \mathbb{R}^{1 \times n}$. To simplify the notation in the following, we drop the subscript l such that $\mathbf{W} = \mathbf{W}_l$, $\boldsymbol{\mu}_q = \boldsymbol{\mu}_{ql}$, $\boldsymbol{\Sigma}_q = \boldsymbol{\Sigma}_{ql}$ and we focus on the kernel matrix $\mathbf{W}$ for a single layer.

In GMFVI, we model the variational posterior as

$$q(\mathbf{W}) = \mathcal{N}(\boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q) = \prod_{i=1}^{m} \prod_{j=1}^{n} q(w_{ij}),$$
$$\text{with} \quad q(w_{ij}) = \mathcal{N}(\mu_{ij}, \sigma_{ij}^2), \qquad (5)$$

where $\boldsymbol{\mu}_q \in \mathbb{R}^{mn \times 1}$ is the posterior mean vector, $\boldsymbol{\Sigma}_q \in \mathbb{R}_+^{mn \times mn}$ is the diagonal posterior covariance matrix. The weights are then usually sampled using a reparameterization trick (Kingma & Welling, 2013), i.e, for the s-th sample, we have

$$w_{ij}^{(s)} = \mu_{ij} + \sigma_{ij} \epsilon^{(s)}, \qquad \epsilon \sim \mathcal{N}(0, 1). \qquad (6)$$

In practice, we often represent the posterior standard deviation parameters σ_{ij} in the form of a matrix $\mathbf{A} \in \mathbb{R}_+^{m \times n}$. Note that we have the relationship $\boldsymbol{\Sigma}_q = \mathrm{diag}(\mathrm{vec}(\mathbf{A}^2))$ where the elementwise-squared $\mathbf{A}$ is vectorized by stacking its columns, and then expanded as a diagonal matrix into $\mathbb{R}_+^{mn \times mn}$.

In the sequel, we start by empirically studying the properties of the spectrum of matrices $\mathbf{A}$ *post-training* (after convergence), while using standard Gaussian mean-field variational distributions. Interestingly, we observe that those matrices naturally exhibit a low-rank structure (see Section 2.3 for the corresponding experiments), i.e,

$$\mathbf{A} \approx \mathbf{U}\mathbf{V}^T \qquad (7)$$

for some $\mathbf{U} \in \mathbb{R}^{m \times k}$, $\mathbf{V} \in \mathbb{R}^{n \times k}$ and k a small value (e.g., 2 or 3). This observation motivates the introduction of the following variational family, which we name k-tied Normal:

$$\boxed{\begin{aligned} k\text{-}tied\text{-}\mathcal{N}(\mathbf{W}; \boldsymbol{\mu}_q, \mathbf{U}, \mathbf{V}) = \\ \mathcal{N}(\boldsymbol{\mu}_q, \mathrm{diag}(\mathrm{vec}((\mathbf{U}\mathbf{V}^T)^2))), \end{aligned}} \qquad (8)$$

Variational family	Parameters (total)
multivariate Normal	$mn + \frac{mn\,(mn+1)}{2}$
diagonal Normal	$mn + mn$
$\mathcal{MN}$	$mn + \frac{m(m+1)}{2} + \frac{n(n+1)}{2}$
$\mathcal{MN}$(diagonal)	$mn + m + n$
k-tied Normal (ours)	$mn + k(m+n)$

Table 1. Number of variational parameters for a variational family for a matrix $\mathbf{W} \in \mathbb{R}^{m \times n}$. $\mathcal{MN}$(diagonal) is from Louizos & Welling (2016).

where the squaring of the matrix $\mathbf{U}\mathbf{V}^T$ is applied elementwise. Due to the tied parameterization of the diagonal covariance matrix, we emphasize that this variational family is *smaller*—i.e., included in—the standard Gaussian mean-field variational distribution family.

As formally discussed in Appendix A, the matrix variate Gaussian distribution (Gupta & Nagar, 2018), referred to as $\mathcal{MN}$ and already used for variational inference by Louizos & Welling (2016) and Sun et al. (2017), is related to our k-tied Normal distribution with $k = 1$ when $\mathcal{MN}$ uses diagonal row and column covariances. Interestingly, we prove that for $k \geq 2$, our k-tied Normal distribution cannot be represented by any $\mathcal{MN}$ distribution. This illustrates the main difference of our approach from the most closely related previous work of Louizos & Welling (2016).

Notice that our diagonal covariance $\boldsymbol{\Sigma}_q$ repeatedly reuses the same elements of $\mathbf{U}$ and $\mathbf{V}$, which results in parameter sharing across different weights. The total number of the standard deviation parameters in our method is $k(m+n)$ from $\mathbf{U}$ and $\mathbf{V}$, compared to mn from $\mathbf{A}$ in the standard GMFVI parameterization. Given that in our experiments the k is very low (e.g. $k = 2$) this reduces the number of parameters from quadratic to linear in the dimensions of the layer, see Table 1. More importantly, such parameter sharing across the weights leads to higher signal-to-noise ratio during training and thus in some cases faster convergence. We demonstrate this phenomena in the next section. In the rest of this section, we show that the standard GMFVI methods already learn a low-rank structure in the posterior standard deviation matrix $\mathbf{A}$. Furthermore, we provide evidence that replacing $\mathbf{A}$ with its low-rank approximation does not degrade the predictive performance and the quality of uncertainty estimates.

2.2. Experimental setting

Before describing the experimental results, we briefly explain the key properties of the experimental setting. We analyze three types of GMFVI Bayesian neural network models:

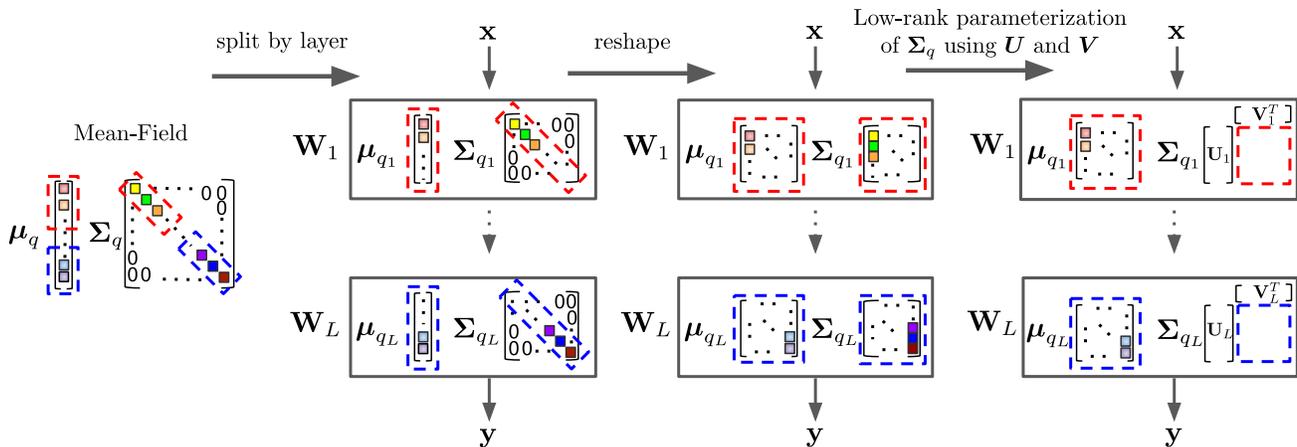- Multilayer Perceptron (MLP): a network of 3 dense

Figure 2. Illustration of the relationship between the standard Gaussian Mean-Field posterior and its "low-rank" parameterization, which we call the k-tied Normal posterior. The illustration shows the posterior parameterization for a network with L layers, where $\mathbf{x}$ and $\mathbf{y}$ are the network inputs and outputs respectively, and $\boldsymbol{\mu}_{q_1}$, $\boldsymbol{\Sigma}_{q_1}$, $\boldsymbol{\mu}_{q_L}$ and $\boldsymbol{\Sigma}_{q_L}$ are the variational parameters for the layers 1 and L respectively. The k-tied Normal distribution parameterizes the already diagonal per layer posterior covariance matrices $\boldsymbol{\Sigma}_{q_{1..L}}$ using the even more compact $\boldsymbol{U}_{1..L}$ and $\boldsymbol{V}_{1..L}$ matrices from $\mathcal{N}\big(\boldsymbol{\mu}_q, \text{diag}\big(\text{vec}\big((\mathbf{U}\mathbf{V}^T)^2\big)\big)\big)$.

layers and ReLu activations that we train on the MNIST dataset (LeCun & Cortes, 2010). We use the last 10,000 examples of the training set as a validation set.

- Convolutional Neural Network (CNN): a LeNet architecture (LeCun et al., 1998) with 2 convolutional layers and 2 dense layers that we train on the CIFAR-100 dataset (Krizhevsky et al., 2009b). We use the last 10,000 examples of the training set as a validation set.
- Long Short-Term Memory (LSTM): a model that consists of an embedding and an LSTM cell (Hochreiter & Schmidhuber, 1997), followed by a single unit dense layer. We train it on the IMDB dataset (Maas et al., 2011), in which we use the last 5,000 examples of the training set as a validation set.
- Residual Convolutional Neural Network (ResNet): a ResNet-18[1] architecture (He et al., 2016) trained on all 50,000 training examples of the CIFAR-10 dataset (Krizhevsky et al., 2009a).

In each of the four models, we use the standard mean-field Normal variational posterior and a Normal prior, for which we set a single scalar standard deviation hyper-parameter shared by all layers. Appendix B contains an ablation study result with an alternative prior. We optimize the variational posterior parameters using the Adam optimizer (Kingma & Ba, 2014). For a more comprehensive explanation of the experimental setup please refer to Appendix D. Finally, we highlight that our experiments focus primarily on the comparison across a broad range of model types rather than competing with the state-of-the-art results over the specifi-

cally used datasets. Nevertheless, we also show that our results extend to larger models with competitive performance such as the ResNet-18 model. Note that scaling GMFVI to such larger model sizes is still a challenging research problem (Osawa et al., 2019).

2.3. Main experimental observation

Our main experimental observation is that the standard GM-FVI learns posterior standard deviation matrices that have a low-rank structure across different model types (MLP, CNN, LSTM), model sizes (LeNet, ResNet-18) and layer types (dense, convolutional). To show this, we investigate the results of the SVD decomposition of posterior standard deviation matrices $\mathbf{A}$ in the four described models types. We analyze the models *post-training*, where the models are already trained until ELBO convergence using the standard GMFVI approach. While for the first three models (MLP, CNN and LSTM), we evaluate the low-rank structure only in the dense layers, for the ResNet model we consider the low-rank structure in the convolutional layers as well.

Figure 3 shows the fraction of variance explained per each singular value k from the SVD decomposition of matrices $\mathbf{A}$ in the dense layers of the first three models. The fraction of variance explained per singular value k is calculated as $\gamma_k^2 / \sum_{i'} \gamma_{i'}^2$, where γ are the singular values. We observe that, unlike posterior means, the posterior standard deviations have most of their variance explained by the first few singular values. In particular, a rank-1 approximation of $\mathbf{A}$ explains most of its variance, while a rank-2 approximation can encompass nearly all of the remaining variance. Figure 4 further supports this claim visually by comparing

---

[1]See: https://github.com/tensorflow/probability/blob/master/tensorflow_probability/examples/cifar10_bnn.py.

the heat maps of the matrix $\mathbf{A}$ and its rank-1 and rank-2 approximations. In particular, we observe that the rank-2 approximation heat map looks visually very similar to $\mathbf{A}$, while this is not the case for the rank-1 approximation. Importantly, Figure 5 illustrates that the same low-rank structure is also present in both the dense and the convolutional layers of the larger ResNet-18 model. In the analysis of the above experiments, we use the shorthand SEM to refer to the standard error of the mean.

2.4. Low-rank approximation of mean field posterior standard deviations

Motivated by the above observations, we show that it is possible to replace the reshaped diagonal posterior standard deviation matrices $\mathbf{A}$ with their low-rank approximations without decreasing predictive performance and the quality of uncertainty estimates. Table 2 shows the performance comparison of the MLP, CNN and LSTM models with different ranks of the approximations. Figure 5 contains analogous results for the ResNet-18 model. The results show that the post-training approximations of the mean field posterior covariance with ranks higher than one achieve predictive performance close to that of the mean field posterior with no approximations for all the analyzed model types, model sizes and layer types. Furthermore, Table 3 shows that, for the ResNet-18 model, the approximations with ranks higher than one also do not decrease the quality of the uncertainty estimates compared to the mean field posterior with no approximations[2]. These observations could be used as a form of post-training network compression. Moreover, they give rise to further interesting exploration directions such as formulating posteriors that exploit such a low-rank structure. In the next section we explore this particular direction while focusing on the first three model types (MLP, CNN, LSTM).

3. The k-tied Normal Distribution: Exploiting Low-Rank Parameter-Structure in Mean Field Posteriors

In the previous section we have shown that it is possible to replace a reshaped diagonal matrix of posterior standard deviations, which is already trained using GMFVI, with its low-rank approximation without decreasing the predictive performance. In this section, we show that it is also possible to exploit this observation during training time. We achieve this by exploiting our novel variational family, the k-tied Normal distribution (see Section 2.1).

We show that using this distribution in the context of GM-

---

[2]We compute the Brier Score and the ECE using the implementations from the TensorFlow Probability (Dillon et al., 2017) Stats module: https://www.tensorflow.org/probability/api_docs/python/tfp/stats.

FVI in Bayesian neural networks allows to reduce the number of network parameters, increase the signal-to-noise ratio of the stochastic gradient estimates and speed up model convergence, while maintaining the predictive performance of the standard parameterization of the GMFVI. We start by recalling the definition of the k-tied Normal distribution:

$$k\text{-}tied\text{-}\mathcal{N}(\mathbf{W}; \boldsymbol{\mu}_q, \mathbf{U}, \mathbf{V}) = \mathcal{N}\big(\boldsymbol{\mu}_q, \mathrm{diag}\big(\mathrm{vec}\big((\mathbf{U}\mathbf{V}^T)^2\big)\big)\big)$$

where the variational parameters are comprised of $\{\boldsymbol{\mu}_q, \mathbf{U}, \mathbf{V}\}$.

3.1. Experimental setting

We now introduce the experimental setting in which we evaluate the GMFVI variational posterior parameterized by the k-tied Normal distribution. We assess the impact of the described posterior in terms of predictive performance and reduction in the number of parameters for the same first three model types (MLP, CNN, LSTM) and respective datasets (MNIST, CIFAR-100, IMDB) as we used in the previous section. Additionally, we also analyze the impact of k-tied Normal posterior on the signal-to-noise ratio of stochastic gradient estimates of the variational lower bound for the CNN model as a representative example. Overall, the experimental setup is very similar to the one introduced in the previous section. Therefore, we highlight only the key differences here.

We apply the k-tied Normal variational posterior distribution to the same layers which we analyzed in the previous section. Namely, we use the k-tied Normal variational posterior for all the three layers of the MLP model, the two dense layers of the CNN model and the LSTM cell's kernel and recurrent kernel. We initialize the parameters u_{ik} from $\mathbf{U}$ and v_{jk} from $\mathbf{V}$ of the k-tied Normal distribution so that after the outer-product operation the respective standard deviations σ_{ij} have the same mean values as we obtain in the standard GMFVI posterior parameterization. In the experiments for this section, we use *KL annealing* (Sønderby et al., 2016), where we linearly scale-up the contribution of the $D_{\mathrm{KL}}[q_{\boldsymbol{\theta}}(\mathbf{w})||p(\mathbf{w})]$ term in Equation 2 from zero to its full contribution over the course of training. Appendix C describes the impact of KL annealing on the modelled uncertainty. Furthermore, additional details on the experimental setup are available in Appendix D.

3.2. Experimental results

We first investigate the predictive performance of the GMFVI Bayesian neural network models trained using the k-tied Normal posterior distribution, with different levels of tying k. We compare these results to those obtained from the same models, but trained using the standard parameterization of the GMFVI. Figure 6 (left) shows that for $k \geq 2$ the k-tied Normal posterior is able to achieve the performance
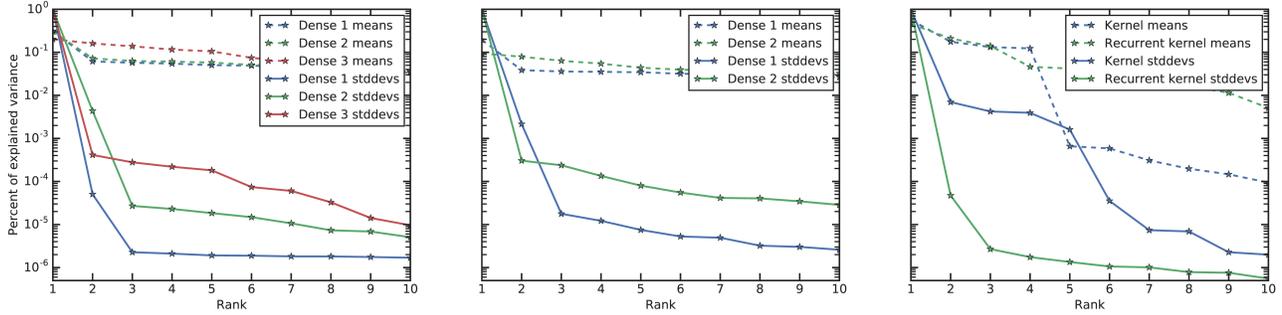
Figure 3. Fraction of variance explained per each singular value from SVD of matrices of posterior means and posterior standard deviations post-training in different dense layers of three model types trained using standard GMFVI: MLP (left), CNN (center), LSTM (right). Unlike posterior means, posterior standard deviations clearly display strong low-rank structure, with most of the variance contained in the top few singular values.
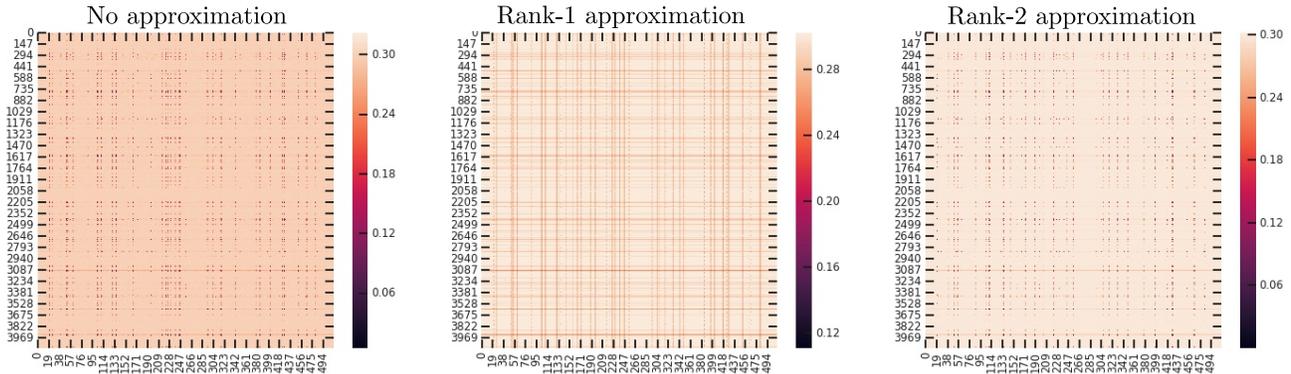

Figure 4. Post-training heat maps of the reshaped diagonal posterior standard deviation matrix for the first dense layer of a LeNet CNN trained using GMFVI on the CIFAR-100 dataset. Unlike the rank-1 approximation, the rank-2 approximation already looks visually very similar to the matrix with no approximation. This is consistent with the quantitative results from Figure 3, where the first two singular values explain most of the variance in the reshaped diagonal posterior standard deviation matrix.

competitive with the standard GMFVI posterior parameterization, while reducing the total number of model parameters. The benefits of using the k-tied Normal posterior are the most visible for models where the layers with the k-tied Normal posterior constitute a significant portion of the total number of the model parameters (e.g. the MLP model).

We further investigate the impact of the k-tied Normal posterior on the signal-to-noise ratio (SNR)[3] of stochastic gradient estimates of the variational lower bound (ELBO). In particular, we focus on the gradient SNR of the GMFVI posterior standard deviation parameters for which we perform the tying. These parameters are either u_{ik} and v_{jk} for the k-tied Normal posterior or σ_{ij} for the standard GM-FVI parameterization, all optimized in their log forms for numerical stability. Figure 6 (top right) shows that the u_{ik} and v_{jk} parameters used in the k-tied Normal posterior are

trained with significantly higher gradient SNR than the σ_{ij} parameters used in the standard GMFVI parameterization. Consequently, Figure 6 (bottom right) shows that the increased SNR from the k-tied Normal distribution translates into faster convergence for the MLP model, which uses the k-tied Normal distribution in all of its layers.

Note that the k-tied Normal posterior does not increase the training step time compared to the standard parameterization of the GMFVI, see Table 4 for the support of this claim[4]. Therefore, the k-tied Normal posterior speeds up model convergence also in terms of wall-clock time.

Figure 7 shows the convergence plots of validation negative ELBO for all the three model types. We observe that the im-

---

[3]SNR for each gradient value is calculated as $E[g_b^2]/\text{Var}[g_b]$, where g_b is the gradient value for a single parameter. The expectation E and variance Var of the gradient values g_b are calculated over a window of last 10 batches.

[4]Code to compare the training step times of the k-tied Normal and the standard GMFVI is available under: https://colab.research.google.com/drive/14pqe_VG5s49xlcXB-Jf8S9GoTFyjv4OF. The code uses the network architecture from: https://github.com/tensorflow/docs/blob/master/site/en/tutorials/keras/classification.ipynb.

	MNIST, MLP			CIFAR100, CNN			IMDB, LSTM		
Method	-ELBO ↓	NLL ↓	Accuracy ↑	-ELBO ↓	NLL ↓	Accuracy ↑	-ELBO ↓	NLL ↓	Accuracy ↑
Mean-field	$0.431_{\pm0.0057}$	$0.100_{\pm0.0034}$	$97.6_{\pm0.15}$	$3.83_{\pm0.020}$	$2.23_{\pm0.017}$	$42.1_{\pm0.49}$	$0.536_{\pm0.0058}$	$0.493_{\pm0.0057}$	$80.1_{\pm0.25}$
1-tied	$3.41_{\pm0.019}$	$0.677_{\pm0.0040}$	$93.6_{\pm0.25}$	$4.33_{\pm0.021}$	$2.30_{\pm0.016}$	$41.7_{\pm0.49}$	$0.687_{\pm0.0058}$	$0.491_{\pm0.0056}$	$80.0_{\pm0.25}$
2-tied	$0.456_{\pm0.0059}$	$0.107_{\pm0.0033}$	$97.6_{\pm0.15}$	$3.88_{\pm0.020}$	$2.24_{\pm0.017}$	$42.2_{\pm0.49}$	$0.621_{\pm0.0058}$	$0.494_{\pm0.0057}$	$80.1_{\pm0.25}$
3-tied	$0.450_{\pm0.0059}$	$0.106_{\pm0.0033}$	$97.6_{\pm0.15}$	$3.86_{\pm0.020}$	$2.24_{\pm0.017}$	$42.1_{\pm0.49}$	$0.595_{\pm0.0058}$	$0.493_{\pm0.0056}$	$80.1_{\pm0.25}$

Table 2. Impact of post-training low-rank approximation of the GMFVI-trained posterior standard deviation matrix on model's ELBO and predictive performance, for three types of models. We report mean and SEM of each metric across 100 weights samples. The low-rank approximations with ranks higher than one achieve predictive performance close to that of mean-field without the approximations.
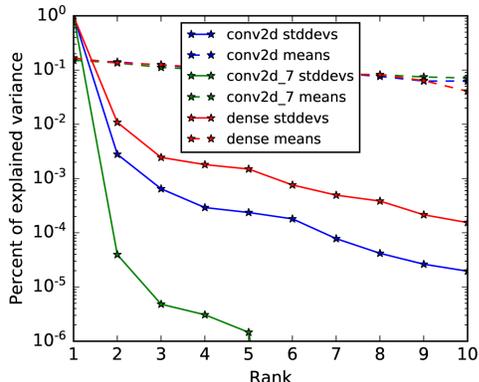
Method	-ELBO ↓	NLL ↓	Accuracy ↑
Mean-field	$122.61_{\pm0.012}$	$0.495_{\pm0.0080}$	$83.5_{\pm0.37}$
1-tied	$122.57_{\pm0.012}$	$0.658_{\pm0.0069}$	$81.7_{\pm0.39}$
2-tied	$122.77_{\pm0.012}$	$0.503_{\pm0.0080}$	$83.2_{\pm0.37}$
3-tied	$122.67_{\pm0.012}$	$0.501_{\pm0.0079}$	$83.2_{\pm0.37}$

Figure 5. Unlike posterior means, the posterior standard deviations of both dense and convolutional layers in the ResNet-18 model trained using standard GMFVI display strong low-rank structure post-training and can be approximated without loss in predictive metrics. Top: Fraction of variance explained per each singular value of the matrices of converged posterior means and standard deviations. Bottom: Impact of post-training low-rank approximation of the posterior standard deviation matrices on the model's performance. We report mean and SEM of each metric across 100 weights samples.

pact of the k-tied Normal posterior on convergence depends on the model type. As shown in Figure 6 (bottom right), the impact on the MLP model is strong and consistent with the k-tied Normal posterior increasing convergence speed compared to the standard GMFVI parameterization. For the LSTM model we also observe a similar speed-up. However, for the CNN model the impact of the k-Normal posterior on the ELBO convergence is much smaller. We hypothesize that this is due to the fact that we use the k-tied Normal posterior for all the layers trained using GMFVI in the MLP and the LSTM models, while in the CNN model we use the k-tied Normal posterior only for some of the GMFVI trained layers. More precisely, in the CNN model we use the k-tied Normal posterior only for the two dense layers, while the two convolutional layers are trained using the standard parameterization of the GMFVI.

Method	Brier Score ↓	NLL ↓	ECE ↓
Mean-field	$-0.761_{\pm0.0039}$	$0.495_{\pm0.0080}$	0.0477
1-tied	$-0.695_{\pm0.0034}$	$0.658_{\pm0.0069}$	0.1642
2-tied	$-0.758_{\pm0.0038}$	$0.503_{\pm0.0080}$	0.0540
3-tied	$-0.758_{\pm0.0038}$	$0.501_{\pm0.0079}$	0.0541

Table 3. Quality of predictive uncertainty estimates for the ResNet-18 model on the CIFAR10 dataset without and with post-training low-rank approximations of the GMFVI posterior standard deviation matrices in all the layers of the model. The approximations with ranks $k \geq 2$ match the quality of the predictive uncertainty estimates from the mean-field posteriors without the approximations. The quality of the predictive uncertainty estimates is measured by the negative log-likelihood (NLL), the Brier Score and the ECE (with 15 bins). For the NLL and the Brier Score metrics we report mean and SEM across 100 weights samples.

Training method	Train step time [ms] ↓
Point estimate	$2.00_{\pm0.0064}$
Standard GMFVI	$7.17_{\pm0.014}$
2-tied Normal GMFVI	$6.14_{\pm0.018}$

Table 4. Training step evaluation times for a simple model architecture with two dense layers for different training methods. We report mean and SEM of evaluation times across a single training run in the Google Colab environment linked in the footnote. The k-tied Normal posterior with $k = 2$ does not increase the train step evaluation times compared to the standard parameterization of the GMFVI posterior. We expect this to hold more generally because the biggest additional operation per step when using the k-tied Normal posterior is the $\mathbf{U}\mathbf{V}^T$ multiplication to materialize the matrix of posterior standard deviations $\mathbf{A}$, where $\mathbf{U} \in \mathbb{R}^{m \times k}$, $\mathbf{V} \in \mathbb{R}^{n \times k}$ and k is a small value (e.g., 2 or 3). The time complexity of this operations is $\mathcal{O}(kmn)$, which is usually negligible compared to the time complexity of data-weight matrix multiplication $\mathcal{O}(bmn)$, where b is the batch size.

Model & Dataset	Method	-ELBO ↓	NLL ↓	Accuracy ↑	#Par. [k]↓
MNIST, MLP	Mean-field	$0.501_{\pm0.0061}$	$0.133_{\pm0.0040}$	$96.8_{\pm0.18}$	957
MNIST, MLP	1-tied	$0.539_{\pm0.0063}$	$0.155_{\pm0.0043}$	$96.1_{\pm0.19}$	482
MNIST, MLP	2-tied	$0.520_{\pm0.0063}$	$0.129_{\pm0.0039}$	$96.8_{\pm0.18}$	484
MNIST, MLP	3-tied	$0.497_{\pm0.0060}$	$0.120_{\pm0.0038}$	$96.9_{\pm0.18}$	486
CIFAR100, CNN	Mean-field	$3.72_{\pm0.018}$	$2.16_{\pm0.016}$	$43.9_{\pm0.50}$	4,405
CIFAR100, CNN	1-tied	$3.65_{\pm0.017}$	$2.12_{\pm0.015}$	$45.5_{\pm0.50}$	2,262
CIFAR100, CNN	2-tied	$3.76_{\pm0.019}$	$2.15_{\pm0.016}$	$44.3_{\pm0.50}$	2,268
CIFAR100, CNN	3-tied	$3.73_{\pm0.018}$	$2.13_{\pm0.016}$	$44.3_{\pm0.50}$	2,273
IMDB, LSTM	Mean-field	$0.538_{\pm0.0054}$	$0.478_{\pm0.0052}$	$79.5_{\pm0.26}$	2,823
IMDB, LSTM	1-tied	$0.592_{\pm0.0041}$	$0.512_{\pm0.0040}$	$77.6_{\pm0.26}$	2,693
IMDB, LSTM	2-tied	$0.560_{\pm0.0042}$	$0.484_{\pm0.0041}$	$78.2_{\pm0.26}$	2,694
IMDB, LSTM	3-tied	$0.550_{\pm0.0051}$	$0.491_{\pm0.0050}$	$78.8_{\pm0.26}$	2,695

Method	MNIST, MLP Dense 2, SNR at step		
	1000	5000	9000
Mean-field	$4.13_{\pm0.027}$	$4.45_{\pm0.091}$	$3.21_{\pm0.035}$
1-tied	$5840_{\pm190}$	$158_{\pm3.8}$	$5.3_{\pm0.20}$
2-tied	$7500_{\pm240}$	$140_{\pm11}$	$4.3_{\pm0.26}$
3-tied	$7000_{\pm270}$	$117_{\pm1.7}$	$4.1_{\pm0.20}$

Method	MNIST, MLP, -ELBO at step		
	1000	5000	9000
Mean-field	$42.16_{\pm0.070}$	$26.52_{\pm0.016}$	$15.39_{\pm0.016}$
1-tied	$43.11_{\pm0.039}$	$14.85_{\pm0.017}$	$2.06_{\pm0.027}$
2-tied	$42.74_{\pm0.090}$	$13.97_{\pm0.023}$	$1.82_{\pm0.017}$
3-tied	$42.63_{\pm0.068}$	$13.61_{\pm0.020}$	$1.80_{\pm0.031}$

Figure 6. Left: impact of the k-tied Normal posterior on test ELBO, test predictive performance and number of model parameters. We report the test metrics on the test splits of the respective datasets as a mean and SEM across 100 weights samples after training each of the models for $\approx$300 epochs. The k-tied Normal distribution with rank $k \geq 2$ allows to train models with smaller number of parameters without decreasing the predictive performance. Top right: mean gradient SNR in the log posterior standard deviation parameters of the Dense 2 layer of the MNIST MLP model at increasing training steps for different ranks of tying k. The k-tied Normal distribution significantly increases the SNR for these parameters. We observe a similar increase in the SNR from tying in all the layers that use the k-tied Normal posterior. Bottom right: negative ELBO on the MNIST validation dataset at increasing training steps for different ranks of tying k. The higher SNR from the k-tied Normal posterior translates into the increased convergence speed for the MLP model. We report mean and SEM across 3 training runs with different random seeds in both the top right and the bottom right table.
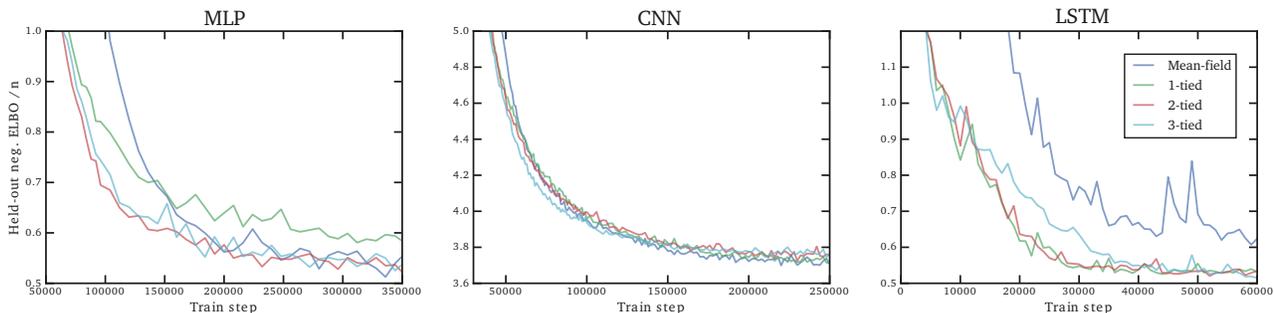

Figure 7. Convergence of negative ELBO (lower is better) for the three model types on their respective validation datasets when training with mean-field and k-tied variational posteriors. The k-tied Normal posteriors result in faster initial convergence for the MLP and LSTM models. For the CNN models the speed-up is not as significant when using the k-tied Normal posterior only for the two dense layers of the LeNet architecture.

4. Related Work

The application of variational inference to neural networks dates back at least to Peterson (1987) and Hinton & Van Camp (1993). Many developments[5] have followed those seminal research efforts, in particular regarding (1) the expressiveness of the variational posterior distribution and (2) the way the variational parameters themselves can be structured to lead to compact, easier-to-learn and scalable formulations. We organize the discussion of this section around those two aspects, with a specific focus on the Gaussian case.

Full Gaussian posterior. Because of their substantial memory and computational cost, Gaussian variational distributions with full covariance matrices have been primarily

---

[5]We refer the interested readers to Zhang et al. (2018) for a recent review of variational inference.

applied to (generalized) linear models and shallow neural networks (Jaakkola & Jordan, 1997; Barber & Bishop, 1998; Marlin et al., 2011; Titsias & Lázaro-Gredilla, 2014; Miller et al., 2017; Ong et al., 2018).

To represent the dense covariance matrix efficiently in terms of variational parameters, several schemes have been proposed, including the sum of low-rank plus diagonal matrices (Barber & Bishop, 1998; Seeger, 2000; Miller et al., 2017; Zhang et al., 2017; Ong et al., 2018), the Cholesky decomposition (Challis & Barber, 2011) or by operating instead on the precision matrix (Tan & Nott, 2018; Mishkin et al., 2018).

Gaussian posterior with block-structured covariances. In the context of Bayesian neural networks, the layers represent a natural structure to be exploited by the covariance matrix. When assuming independence across layers, the resulting covariance matrix exhibits a *block-diagonal struc-*

ture that has been shown to be a well-performing simplification of the dense setting (Sun et al., 2017; Zhang et al., 2017), with both memory and computational benefits.

Within each layer, the corresponding diagonal block of the covariance matrix can be represented by a Kronecker product of two smaller matrices (Louizos & Welling, 2016; Sun et al., 2017), possibly with a parameterization based on rotation matrices (Sun et al., 2017). Finally, using similar techniques, Zhang et al. (2017) proposed to use a block tridiagonal structure that better approximates the behavior of a dense covariance.

Fully factorized mean-field Gaussian posterior. A fully factorized Gaussian variational distribution constitutes the simplest option for variational inference. The resulting covariance matrix is diagonal and all underlying parameters are assumed to be independent. While the mean-field assumption is known to have some limitations—e.g., underestimated variance of the posterior distribution (Turner & Sahani, 2011) and robustness issues (Giordano et al., 2018)—it leads to scalable formulations, with already competitive performance, as for instance illustrated by the recent uncertainty quantification benchmark of Ovadia et al. (2019).

Because of its simplicity and scalability, the fully-factorized Gaussian variational distribution has been widely used for Bayesian neural networks (Graves, 2011; Ranganath et al., 2014; Blundell et al., 2015; Hernández-Lobato & Adams, 2015; Zhang et al., 2017; Khan et al., 2018).

Our approach can be seen as an attempt to further reduce the number of parameters of the (already) diagonal covariance matrix. Closest to our approach is the work of Louizos & Welling (2016). Their matrix variate Gaussian distribution instantiated with the Kronecker product of the diagonal row- and column-covariance matrices leads to a rank-1 tying of the posterior variances. In contrast, we explore tying strategies beyond the rank-1 case, which we show to lead to better performance (both in terms of ELBO and predictive metrics). Importantly, we further prove that tying strategies with a rank greater than one cannot be represented in a matrix variate Gaussian distribution, thus clearly departing from Louizos & Welling (2016) (see Appendix A for details).

Our approach can be also interpreted as a form of *hierarchical* variational inference from Ranganath et al. (2016). In this interpretation, the prior on the variational parameters corresponds to a Dirac distribution, non-zero only when a pre-specified low-rank tying relationship holds. More recently, Karaletsos et al. (2018) proposed a hierarchical structure which also couples network weights, but achieves this by introducing representations of network units as latent variables.

Our reduction in the number parameters through tying decreases the variance of the stochastic gradient estimates of the ELBO objective for the posterior standard deviation parameters. Alterantive methods for the variance reduction propose to either change the noise sampling scheme (Kingma et al., 2015; Wen et al., 2018; Farquhar et al., 2020a) or to determinize the variational posterior approximation (Wu et al., 2019). Those methods can be combined with our method, but some of them are not valid in cases when our method is applicable (e.g. the method from Kingma et al. (2015) is not valid for convolutional layers).

We close this related work section by mentioning the existence of other strategies to produce more flexible approximate posteriors, e.g., normalizing flows (Rezende & Mohamed, 2015) and extensions thereof (Louizos & Welling, 2017).

5. Conclusion

In this work we have shown that Bayesian Neural Networks trained with standard Gaussian Mean-Field Variational Inference learn posterior standard deviation matrices that can be approximated with little information loss by low-rank SVD decompositions. This suggests that richer parameterizations of the variational posterior may not always be needed, and that compact parameterizations can also work well. We used this insight to propose a simple, yet effective variational posterior parameterization, which speeds up training and reduces the number of variational parameters without degrading predictive performance on a range of model types.

In future work, we hope to scale up variational inference with compactly parameterized approximate posteriors to much larger models and more complex problems. For mean-field variational inference to work well in that setting several challenges will likely need to be addressed (Wenzel et al., 2020); improving the signal-to-noise ratio of ELBO gradients using our compact variational parameterizations may provide a piece of the puzzle.

References

Barber, D. and Bishop, C. M. Ensemble learning for multilayer networks. In *Advances in neural information processing systems*, pp. 395–401, 1998.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra,

D. Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*, 2015.

Challis, E. and Barber, D. Concave gaussian variational approximations for inference in large-scale bayesian linear models. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 199–207, 2011.

Dillon, J. V., Langmore, I., Tran, D., Brevdo, E., Vasudevan, S., Moore, D., Patton, B., Alemi, A., Hoffman, M., and Saurous, R. A. Tensorflow distributions. *arXiv preprint arXiv:1711.10604*, 2017.

Farquhar, S., Osborne, M., and Gal, Y. Radial Bayesian neural networks: Beyond discrete support in large-scale bayesian deep learning. *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*, 2020a.

Farquhar, S., Smith, L., and Gal, Y. Try Depth instead of weight correlations: Mean-field is a less restrictive assumption for variational inference in deep networks. *Bayesian Deep Learning Workshop at NeurIPS*, 2020b.

Giordano, R., Broderick, T., and Jordan, M. I. Covariances, robustness and variational bayes. *The Journal of Machine Learning Research*, 19(1):1981–2029, 2018.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep learning*. MIT press, 2016.

Graves, A. Practical variational inference for neural networks. In *Advances in neural information processing systems*, pp. 2348–2356, 2011.

Gupta, A. K. and Nagar, D. K. *Matrix variate distributions*. Chapman and Hall/CRC, 2018.

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.

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.

Hinton, G. and Van Camp, D. Keeping neural networks simple by minimizing the description length of the weights. In *in Proc. of the 6th Ann. ACM Conf. on Computational Learning Theory*. Citeseer, 1993.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Jaakkola, T. and Jordan, M. A variational approach to bayesian logistic regression models and their extensions. In *Sixth International Workshop on Artificial Intelligence and Statistics*, volume 82, pp. 4, 1997.

Karaletsos, T., Dayan, P., and Ghahramani, Z. Probabilistic meta-representations of neural networks. *arXiv preprint arXiv:1810.00555*, 2018.

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.

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

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

Kingma, D. P., Salimans, T., and Welling, M. Variational dropout and the local reparameterization trick. In *Advances in Neural Information Processing Systems*, pp. 2575–2583, 2015.

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

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

LeCun, Y. and Cortes, C. MNIST handwritten digit database. 2010. URL http://yann.lecun.com/exdb/mnist/.

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

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.

Louizos, C. and Welling, M. Multiplicative normalizing flows for variational bayesian neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 2218–2227. JMLR. org, 2017.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*, pp. 142–150. Association for Computational Linguistics, 2011.

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

Marlin, B. M., Khan, M. E., and Murphy, K. P. Piecewise bounds for estimating bernoulli-logistic latent gaussian models. In *Proceedings of the International Conference on Machine Learning*, pp. 633–640, 2011.

Miller, A. C., Foti, N. J., and Adams, R. P. Variational boosting: Iteratively refining posterior approximations. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 2420–2429. JMLR. org, 2017.

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.

Neal, R. M. Bayesian learning via stochastic dynamics. In *Advances in neural information processing systems*, pp. 475–482, 1993.

Ong, V. M.-H., Nott, D. J., and Smith, M. S. Gaussian variational approximation with a factor covariance structure. *Journal of Computational and Graphical Statistics*, 27 (3):465–478, 2018.

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., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J. V., Lakshminarayanan, B., and Snoek, J. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *arXiv preprint arXiv:1906.02530*, 2019.

Peterson, C. A mean field theory learning algorithm for neural networks. *Complex systems*, 1:995–1019, 1987.

Raftery, A. E., Gneiting, T., Balabdaoui, F., and Polakowski, M. Using bayesian model averaging to calibrate forecast ensembles. *Monthly weather review*, 133(5):1155–1174, 2005.

Ranganath, R., Gerrish, S., and Blei, D. Black box variational inference. In *Artificial Intelligence and Statistics*, pp. 814–822, 2014.

Ranganath, R., Tran, D., and Blei, D. Hierarchical variational models. In *International Conference on Machine Learning*, pp. 324–333, 2016.

Rezende, D. and Mohamed, S. Variational inference with normalizing flows. In *International Conference on Machine Learning*, pp. 1530–1538, 2015.

Salimans, T., Knowles, D. A., et al. Fixed-form variational posterior approximation through stochastic linear regression. *Bayesian Analysis*, 8(4):837–882, 2013.

Seeger, M. Bayesian model selection for support vector machines, gaussian processes and other kernel classifiers. In *Advances in neural information processing systems*, pp. 603–609, 2000.

Sønderby, C. K., Raiko, T., Maaløe, L., Sønderby, S. K., and Winther, O. How to train deep variational autoencoders and probabilistic ladder networks. In *33rd International Conference on Machine Learning (ICML 2016)*, 2016.

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

Tan, L. S. and Nott, D. J. Gaussian variational approximation with sparse precision matrices. *Statistics and Computing*, 28(2):259–275, 2018.

Titsias, M. and Lázaro-Gredilla, M. Doubly stochastic variational bayes for non-conjugate inference. In *International conference on machine learning*, pp. 1971–1979, 2014.

Turner, R. and Sahani, M. *Two problems with variational expectation maximisation for time-series models*, pp. 109–130. Cambridge University Press, 2011.

Wen, Y., Vicol, P., Ba, J., Tran, D., and Grosse, R. Flipout: Efficient pseudo-independent weight perturbations on mini-batches. *arXiv preprint arXiv:1803.04386*, 2018.

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

Wu, A., Nowozin, S., Meeds, E., Turner, R. E., Hernández-Lobato, J. M., and Gaunt, A. L. Deterministic variational inference for robust bayesian neural networks. In *International Conference on Learning Representations (ICLR 2019)*, 2019.

Zhang, C., Butepage, J., Kjellstrom, H., and Mandt, S. Advances in variational inference. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

Zhang, G., Sun, S., Duvenaud, D., and Grosse, R. Noisy natural gradient as variational inference. *arXiv preprint arXiv:1712.02390*, 2017.