Estimating Model Uncertainty of Neural Networks in Sparse Information Form

Jongseok Lee [1] **Matthias Humt** [1] **Jianxiang Feng** [1 2] **Rudolph Triebel** [1 2]

Abstract

We present a sparse representation of model uncertainty for Deep Neural Networks (DNNs) where the parameter posterior is approximated with an inverse formulation of the Multivariate Normal Distribution (MND), also known as the *information form*. The key insight of our work is that the information matrix, i.e. the inverse of the covariance matrix tends to be sparse in its spectrum. Therefore, dimensionality reduction techniques such as low rank approximations (LRA) can be effectively exploited. To achieve this, we develop a novel sparsification algorithm and derive a cost-effective analytical sampler. As a result, we show that the information form can be scalably applied to represent model uncertainty in DNNs. Our exhaustive theoretical analysis and empirical evaluations on various benchmarks show the competitiveness of our approach over the current methods.

1. Introduction

Whenever machine learning methods are used for safety-critical applications such as autonomous driving, it is crucial to provide a precise estimation of the failure probability of the learned predictor. Therefore, most of the current learning approaches return distributions rather than single, most-likely predictions. However, in case of DNNs, this true failure probability tends to be severely underestimated, leading to *overconfident* predictions (Guo et al., 2017). The main reason for this is that DNNs are typically trained with a principle of *maximum likelihood*, neglecting their *epistemic* or model uncertainty with the point estimates of parameters.

Imposing Gaussians on model uncertainty is arguably the most popular choice as Gaussians are for *approximate inference* what linear maps are for algebra. For example, once the *posterior distribution* is inferred, the majority of computations can be performed using the well known tools of linear algebra. For DNNs however, the space complexity of using MNDs is intractable as the covariance matrix scales quadratic to the number of parameters. Consequently, approximate inference on DNNs posterior often neglects the parameter correlations (Wu et al., 2019; Kingma et al., 2015; Graves, 2011; Herandez-Lobato & Adams, 2015) or simplifies the covariance matrix into Kronecker products of two smaller matrices (Sun et al., 2017; Louizos & Welling, 2016; Zhang et al., 2018; Park et al., 2019) regardless of the inference principles such as variational inference.

Instead, inspired by Thrun et al. (2004), we advocate to explore the dual and inverse formulation of MNDs:

$$\begin{aligned}\bar{x} &\propto \exp\left\{-\frac{1}{2}(\bar{x}-\mu)^T\Sigma^{-1}(\bar{x}-\mu)\right\}\\ &= \exp\left\{-\frac{1}{2}\bar{x}^T\Sigma^{-1}\bar{x}+\mu^T\Sigma^{-1}\bar{x}\right\}\\ &= \exp\left\{-\frac{1}{2}\bar{x}^T\boldsymbol{I}\bar{x}+\mu^{IV}\bar{x}\right\} \text{ or } \bar{x}\sim\mathcal{N}^{-1}(\mu^{IV},\boldsymbol{I})\end{aligned}$$

where the Gaussian random variable $\bar{x}$ is fully parameterized by the information vector μ^{IV} and matrix $\boldsymbol{I}$ as opposed to mean μ and covariance matrix Σ. Our major findings are that this so-called information form has important ramifications on developing scalable *Bayesian Neural Networks*. Firstly, we point out that the approximate inference for this formulation can be simplified to scalable Laplace Approximation (LA) (MacKay, 1992b; Ritter et al., 2018a), in which we improve the state-of-the-art Kronecker factored approximations of the information matrix (George et al., 2018) by correcting the diagonal variance in parameter space.

More importantly, DNNs offer a natural spectral sparsity in the information matrix (Sagun et al., 2018) as oppose to the covariance matrix. Intuitively, the information content of each parameters become weaker with increasing number of parameters and thus, sparse representations can be effectively exploited. To do so, we propose a novel low-rank representation of the given Kronecker factorization and devise a spectral sparsification algorithm that can pre-

[1]Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Wessling, Germany [2]Computer Vision Group, Technical University of Munich (TU Munich), Garching, Germany. Correspondence to: Jongseok Lee <jongseok.lee@dlr.de>.

Proceedings of the 37th International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020.

serve the Kronecker product in its eigenbasis. Based on this formulation, we further demonstrate low rank sampling computations which significantly reduces the space complexity of MND from $O(N^3)$ to $O(L^3)$ where L is the chosen low-rank dimension instead of parameter space lying in high dimensional N manifolds. Lastly, we also exhaustively perform both theoretical and empirical evaluations, yielding state-of-the-art results in both scalability and performance.

Our main contribution is a novel sparse representation for DNNs posterior that is backed up by scalable mathematical foot-works - more specifically: (i) an approximate inference that estimates model uncertainty in information form (section 2.2), (ii) a low-rank representation of Kronecker factored eigendecomposition (section 2.3), (iii) an algorithm to enable a LRA for the given representation of MNDs (algorithm 1) and (iv) derivation of a memory-wise tractable sampler (section 2.4). With our theoretical (section 2.5) and experimental results (section 4) we further showcase the state-of-the-art performance. Finally, a plug-in-and-play code is attached for enabling adoptions in practice.

2. Methodology

2.1. Background and Notation

A neural network is a parameterized function $f_\theta : \mathbb{R}^{N_1} \rightarrow \mathbb{R}^{N_l}$ where $\theta \in \mathbb{R}^{N_\theta}$ are the weights and $N_\theta = N_1 + \cdots + N_l$. This function f_θ is in fact a concatenation of l layers, where each layer $\mathfrak{i} \in \{1, ..., l\}$ computes $h_\mathfrak{i} = W_\mathfrak{i} a_{\mathfrak{i}-1}$ and $a_\mathfrak{i} = \phi(h_{\mathfrak{i}-1})$. Here, ϕ is a nonlinear function, $a_\mathfrak{i}$ are activations, $h_\mathfrak{i}$ linear pre-activations, and $W_\mathfrak{i}$ are weight matrices. The bias terms are absorbed into $W_\mathfrak{i}$ by appending 1 to each $a_\mathfrak{i}$. Thus, $\theta = \begin{bmatrix} vec(W_1)^T & \dots & vec(W_l)^T \end{bmatrix}^T$ where vec is the operator that stacks the columns of a matrix to a vector. Let $g_\mathfrak{i} = \delta h_\mathfrak{i}$, the gradient of $h_\mathfrak{i}$ w.r.t θ. Using LA the posterior is approximated with a Gaussian. The mean is then given by the MAP estimate θ_{MAP} and the covariance by the Hessian of the log-likelihood $(H + \tau I)^{-1}$ assuming a Gaussian prior with precision τ. Using loss functions such as MSE or cross entropy and piece-wise linear activation $a_\mathfrak{i}$ (e.g RELU), a good approximation of the Hessian is the Fisher information matrix (IM) $\boldsymbol{I} = \mathbb{E}\left[\delta\theta\delta\theta^T\right]$ for the backpropagated gradients $\delta\theta$ [1] and is typically scaled by the number of data points N (Martens & Grosse, 2015). IM is of size $N_\theta \times N_\theta$ resulting in too large matrix for moderately sized DNNs.

To make the computation tractable, it is first assumed that the weights across layers are uncorrelated, which corresponds to a block-diagonal form of $\boldsymbol{I}$ with blocks $\boldsymbol{I}_1, \dots, \boldsymbol{I}_l$. Then, each realisation of block $\boldsymbol{I}_i$ is represented as a Kronecker product $\delta\theta_\mathfrak{i}\delta\theta_\mathfrak{i}^T = a_{\mathfrak{i}-1}a_{\mathfrak{i}-1}^T \otimes g_\mathfrak{i} g_\mathfrak{i}^T$. Then, matrices $A_{\mathfrak{i}-1}$ and $G_\mathfrak{i}$ are assumed to be statistically independent:

$$\boldsymbol{I}_\mathfrak{i} \approx \boldsymbol{I}_{\mathfrak{i},\text{kfac}} = \mathbb{E}\left[a_{\mathfrak{i}-1}a_{\mathfrak{i}-1}^T\right] \otimes \mathbb{E}\left[g_\mathfrak{i} g_\mathfrak{i}^T\right] = A_{\mathfrak{i}-1} \otimes G_\mathfrak{i}. \quad (1)$$

We refer to Martens & Grosse (2015) for details on KFAC. Here, $A_{\mathfrak{i}-1} \in \mathbb{R}^{n_\mathfrak{i} \times n_\mathfrak{i}}$ and $G_\mathfrak{i} \in \mathbb{R}^{m_\mathfrak{i} \times m_\mathfrak{i}}$, where the number of weights is $N_\mathfrak{i} = n_\mathfrak{i} m_\mathfrak{i}$. Typically IM is scaled by the number of data points N and incorporates the prior τ. The herein presented parameter posterior omits the addition of prior precision and scaling term for simplicity. Here, N and τ are treated as hyperparameters (Ritter et al., 2018a) similar to tempering in (Wenzel et al., 2020). KFAC scales to big data sets such as ImageNet (Krizhevsky et al., 2012) with large DNNs (Ba et al., 2017) and does not require changes in the training procedure when used for LA (Ritter et al., 2018a).

2.2. Approximate Inference in Information Form

We first employ an eigenvalue correction in the Kronecker factored eigenbasis (George et al., 2018) for LA. Layer indices $\mathfrak{i}$ are omitted and explanation applies layer-wise.

Let $\boldsymbol{I} = V_\text{true}\Lambda_\text{true}V_\text{true}^T$ be the true eigendecomposition of IM per layer. From this it follows $\Lambda_\text{true} = V_\text{ture}^T \boldsymbol{I} V_\text{true} = \mathbb{E}\left[V_\text{true}^T \delta\theta\delta\theta^T V_\text{true}\right]$ and $\Lambda_{\text{true},ii} = \mathbb{E}\left[(V_\text{true}^T \delta\theta)_i^2\right]$ where $i \in \{1, \cdots, N\}$ and N is the number of parameters of this layer. Defining the eigendecomposition of A and G in (1) as $A = U_A S_A U_A^T$ and $G = U_G S_G U_G^T$, it further follows $\boldsymbol{I}_\text{kfac} \approx A \otimes G = (U_A \otimes U_G)(S_A \otimes S_G)(U_A \otimes U_G)^T$ from the properties of the Kronecker product. Now, this approximation can be improved by replacing $(S_A \otimes S_G)$ with the eigenvalues Λ_true, where V_true is approximated with $(U_A \otimes U_G)$ resulting in $\Lambda_{ii} = \mathbb{E}\left[[(U_A \otimes U_G)^T \delta\theta]_i^2\right]$. We denote this as the eigenvalue corrected, Kronecker-factored eigenbasis (EFB):

$$\boldsymbol{I}_\text{efb} = (U_A \otimes U_G)\Lambda(U_A \otimes U_G)^T \quad (2)$$

This technique has many desirable properties. Notably, $\|\boldsymbol{I} - \boldsymbol{I}_\text{efb}\|_F \leq \|\boldsymbol{I} - \boldsymbol{I}_\text{kfac}\|_F$ wrt. the Frobenius norm as the computation is more accurate by correcting the eigenvalues.

However, there is an approximation in EFB since $(U_A \otimes U_G)$ is still an approximation of the true eigenbasis V_true. Intuitively, EFB only performs a correction of the diagonal elements in the eigenbasis, but when mapping back to the parameter space this correction is again harmed by the inexact estimate of the eigenvectors. Although an exact estimation of the eigenvectors is infeasible, it is important to note that the diagonals of the exact IM $\boldsymbol{I}_{ii} = \mathbb{E}\left[\delta\theta_i^2\right]$ can be computed efficiently using back-propagation. This motivates the idea

[1] The expectation herein is defined with respect to the paramerterized density function $p_\theta(y|x)$ assuming i.i.d. samples x.

to correct the approximation further as follows:

$$\boldsymbol{I}_{\text{inf}} = (U_A \otimes U_G)\Lambda(U_A \otimes U_G)^T + D \text{ where}$$
$$D_{ii} = \mathbb{E}\left[\delta\theta_i^2\right] - \sum_{j=1}^{nm}(v_{i,j}\sqrt{\Lambda_j})^2. \tag{3}$$

In (3), we have represented $(U_A \otimes U_G)\Lambda(U_A \otimes U_G)^T_{ii}$ as $\sum_{j=1}^{nm}(v_{i,j}\sqrt{\Lambda_j})^2$ where $V = (U_A \otimes U_G) \in \mathbb{R}^{mn\times mn}$ is a Kronecker product with row elements $v_{i,j}$ (see definition 1 below). It follows from the properties of the Kronecker product that $i = m(\alpha-1)+\gamma$. The derivation is shown in supplementary materials. Note that the Kronecker products are never directly evaluated but the diagonal matrix D can be computed recursively, making it computationally feasible.

Definition 1: *For $U_A \in \mathbb{R}^{n\times n}$ and $U_G \in \mathbb{R}^{m\times m}$, the Kronecker product of $V = U_A \otimes U_G \in \mathbb{R}^{mn\times mn}$ is given by $V_{i,j} = U_{A_{\alpha,\beta}}U_{G_{\gamma,\zeta}}$, with $i = m(\alpha-1)+\gamma$ and $j = m(\beta-1)+\zeta$. $\alpha \in \{1,\cdots,n\}$ and $\beta \in \{1,\cdots,n\}$ are row and column indices of U_A. So as $\gamma \in \{1,\cdots,m\}$ and $\zeta \in \{1,\cdots,m\}$ for U_G.*

Now, the parameter posterior distribution can be represented in an information form $\mathcal{N}^{-1}$ of MND as shown below:

$$p(\theta \mid x,y) \sim \mathcal{N}(\theta_{MAP}, \boldsymbol{I}_{\text{inf}}^{-1})$$
$$= \mathcal{N}^{-1}(\theta^{IV}_{MAP}, (U_A \otimes U_G)\Lambda(U_A \otimes U_G)^T + D).$$

This shows how an information form of MND can be computed using LA, and from its graphical interpretation, keeping the diagonals of IM exact has also a consequence of obtaining information content of the parameters accurate (Paskin, 2003). We note that, similar insights have been studied for Bayesian tracking problems (Thrun et al., 2004) with wide adoptions in practice (Bailey & Durrant-Whyte, 2006; Thrun & Liu, 2005; Eustice et al., 2006).

For a full Bayesian analysis with DNNs, however, the samples of the resulting posterior are to be drawn from the information matrix instead of the covariance matrix. For this, an efficient sampling computation is proposed next.

2.3. Model Uncertainty in Sparse Information Form

Sampling from the posterior is crucial. For example, an important use-case of the parameter posterior is estimating the predictive uncertainty for test data (x^*,y^*) by a full Bayesian analysis with K_{mc} samples. This step is typically approximated with Monte-carlo integration (Gal, 2016):

$$p(y^*|x^*,x,y) \approx \frac{1}{K_{mc}}\sum_{t=1}^{K_{mc}} y^*(x^*,\theta_t^s) \text{ for } \theta_t^s \sim \mathcal{N}^{-1}(\theta^{IV}_{MAP}, \boldsymbol{I}_{\text{inf}}).$$

However, this operation is non-trivial as the sampling computation requires $O(N^3)$ complexity (the cost of inversion and finding a symmetrical factor) and for matrices that lie in a high dimensional space, it is computationally infeasible. While previous works (Martens & Grosse, 2015) showed that Kronecker products of two matrices can be exploited along a fidelity-cost trade-off, our main aim is to introduce an alternative form of the Gaussian posterior family.

Observing that the eigenvalues of IM tends to be close to zero for the overparameterized DNNs (Sagun et al., 2018), information form can naturally leverage dimensionality reduction in IM (in oppose to the covariance matrix). Intuitively speaking, as more and more parameters are used to explain the same set of data, information of these parameters tends to be smaller, and we can make use of this tendency.

To this end, we propose the low rank form in (4) [2] as a first step, in which we preserve the Kronecker product in eigenvectors. Here, we highlight that the proposed form differs from conventional LRA which do not preserve the Kronecker product in eigenvectors (i.e $(U_A \otimes U_G)_{1:L}$ for top L eigenvalues). Two main advantages of this representation are that it avoids the memory-wise expensive computation of evaluating the matrix $(U_a \otimes U_g)$, where U_a and U_g are sub-matrices of U_A and U_G, respectively. This formulation also results in sampling computation that is $O(L^3)$ in cost instead of $O(N^3)$, which we demonstrate later in section 2.4.

$$\boldsymbol{I}_{\text{inf}} \approx \hat{\boldsymbol{I}}_{\text{inf}} = (U_a \otimes U_g)\Lambda_{1:L}(U_a \otimes U_g)^T + D \tag{4}$$

Here, $\Lambda_{1:L} \in \mathbb{R}^{L\times L}$, $U_a \in \mathbb{R}^{m\times a}$ and $U_g \in \mathbb{R}^{n\times g}$ denote low rank form of corresponding eigenvalues and vectors (depicted in figure 1). Naturally, it follows that $L = ag$, $N = mn$ and furthermore, the persevered rank L corresponds to preserving top K and additional J eigenvalues (resulting in $L \geq K$, $L = ag = K + J$) as explained with an example.

Why can't LRA directly be used?: *Let a matrix $E = U_{1:6}\Lambda_{1:6}U_{1:6}^T \in \mathbb{R}^{6\times6}$ with $U_{1:6} = [u_1 \cdots u_6] \in \mathbb{R}^{6\times6}$ with $\{u_i\}_{i=1}^6$ the eigenvectors and $\Lambda_{1:6} = diag(\lambda_1,\cdots,\lambda_6) \in \mathbb{R}^{6\times6}$ in a descending order. In this toy example, LRA with top 3 eigenvalues results: $E_{1:3} = U_{1:3}\Lambda_{1:3}U_{1:3}^T \in \mathbb{R}^{6\times6}$. Instead, if the eigenvector matrices are expressed in Kronecker product structure, $E_{kron} = (U_{A_{1:3}} \otimes U_{G_{1:2}})\Lambda_{1:6}(U_{A_{1:3}} \otimes U_{G_{1:2}})^T \in \mathbb{R}^{6\times6}$. For LRA, it's not trivial to directly preserve the top 3 eigenvalues $\Lambda_{1:3}$ and corresponding eigenvectors $(U_{A_{1:3}} \otimes U_{G_{1:2}})_{1:3}$. Because as $(U_{A_{1:a}} \otimes U_{G_{1:g}})_{1:3} = \begin{bmatrix} u_{A_1} \otimes u_{G_1} & u_{A_1} \otimes u_{G_2} & u_{A_2} \otimes u_{G_1} \end{bmatrix}$, preserving the eigenvectors with Kronecker structure results in having to store more eigenvectors: $U_{A_{1:2}} = [u_{A_1} u_{A_2}]$ and $U_{G_{1:2}} = [u_{G_1} u_{G_2}]$. Consequently, additional eigenvalue Λ_4 needs to be saved so that $E_{kron_{1:3}} = (U_{A_{1:2}} \otimes U_{G_{1:2}})\Lambda_{1:4}(U_{A_{1:2}} \otimes U_{G_{1:2}})^T \in \mathbb{R}^{6\times6}$.*

[2] D is added after LRA which is computed similar to (3).

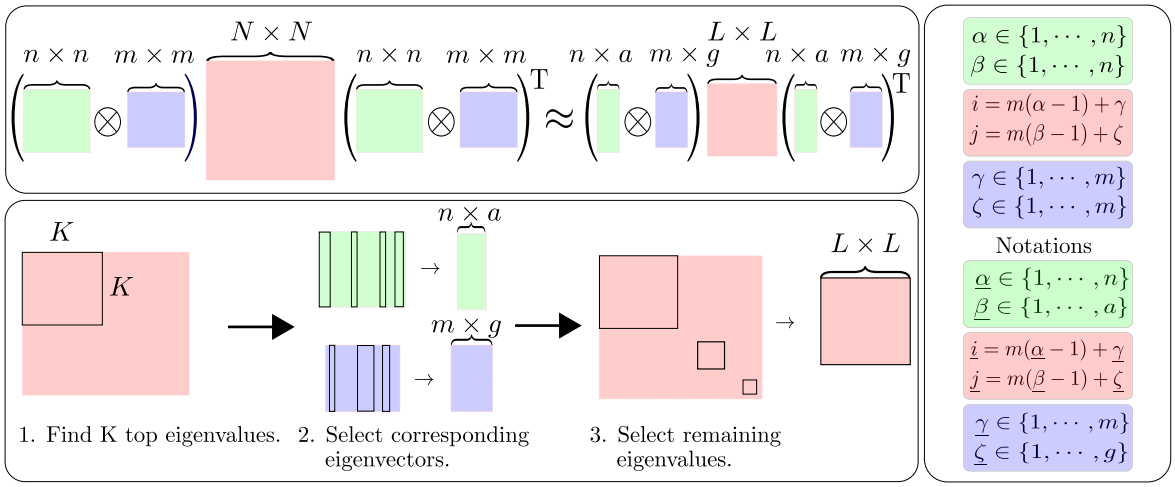

Figure 1. **Illustration of algorithm 1.** A low rank approximation on Kronecker factored eigendecomposition that preserves Kronecker structure in eigenvectors have two benefits: (a) reducing directly $(U_A \otimes U_G)_{1:L}$ is memory-wise infeasible, and (b) sampling costs are drastically reduced as demonstrated in section 2.4. Notations, low rank structure and a visualization of algorithm 1 are depicted.

Algorithm 1 Spectral sparsification

Input: Matrices U_A, U_G, Λ and Rank K.
Output: Matrices: U_a, U_g, $\Lambda_{1:L}$.
1. Find top K eigenvalues λ_i on Λ where $i \in \{1, \cdots, K\}$.
2. For all i, find $\underline{\beta} = floor(\frac{i}{m}) + 1$ and $\underline{\zeta} = i - m(\underline{\beta} - 1)$.
3. Generate sub-matrices U_a and U_g by selecting eigenvectors in U_A and U_G according to $\underline{\beta}$ and $\underline{\zeta}$.
4. Find remaining eigenvalues λ_j with $j = m(\underline{\beta} - 1) + \underline{\zeta}$.
5. Concatenate and diagonalize selected eigenvalues $\Lambda_{1:L} = diag([\lambda_i, \lambda_j])$ for all i and j.

Then, how to achieve a LRA that preserves Kronecker structures in eigenvectors? For this, we propose algorithm 1 (also illustrated in figure 1). To select the additional eigenvectors and -values correctly, we need to introduce a definition on indexing rules of Kronecker factored diagonal matrices.

Definition 2: *For diagonal matrices $S_A \in \mathbb{R}^{n \times n}$ and $S_G \in \mathbb{R}^{m \times m}$, the Kronecker product of $\Lambda = S_A \otimes S_G \in \mathbb{R}^{mn \times mn}$ is given by $\Lambda_i = s_{\alpha\beta} s_{\gamma\zeta}$, where the indices $i = m(\beta - 1) + \zeta$ with $\beta \in \{1, \cdots, m\}$ and $\zeta \in \{1, \cdots, n\}$. Then, given i and m, $\beta = floor(\frac{i}{m}) + 1$ and given β, m, and i, $\zeta = i - m(\beta - 1)$.*

Notations in algorithm 1 are also depicted in figure 1. Now we explain this computation with a toy example below.

Algorithm 1 for a toy example: *To explain, the same toy example can be revisited. Firstly, we aim to preserve the top 3 eigenvalues, $i \in \{1, 2, 3\}$ which are indices of eigenvalues $\Lambda_{1:3}$ (step 1). Then, $\underline{\beta} \in \{1, 2\}$ and $\underline{\zeta} \in \{1, 2\}$ can be computed using definition 2 (step 2). This relation holds as Λ is computed from $S_A \otimes S_G$, and thus, U_A and U_G are their corresponding eigenvectors respectively. Then we produce $U_{A_{1:2}}$ and $U_{G_{1:2}}$ according to $\underline{\beta}$ and $\underline{\zeta}$ (step 3). Again, in order to fulfill the Kronecker product operation, we need to find the eigenvalues $\underline{j} \in \{1, 2, 3, 4\}$, and preserve $\Lambda_{1:4}$ (step 4&5). This results in saving top 3 and additional 1 eigenvalues. Algorithm 1 provides the generalization of these steps and even if eigendecomposition does not come with a descending order, the same logic trivially applies.*

Next, we describe the last step of the sampling derivation.

2.4. Low Rank Sampling Computations

Consider drawing samples $\theta_t^s \in \mathbb{R}^{mn}$ from the representation:

$$\theta_t^s \sim \mathcal{N}^{-1}(\theta_{\text{MAP}}^{IV}, (U_a \otimes U_g)\Lambda_{1:L}(U_a \otimes U_g)^T + D). \tag{5}$$

Typically, drawing such samples θ_t^s requires finding a symmetrical factor of the covariance matrix (e.g. Chloesky decomposition) which is cubic in cost $O(N^3)$ (here $N = mn$). Furthermore, in our representation, it requires first an inversion of IM and then the computation of a symmetrical factor which overall constitutes two operations of cost $O(N^3)$. Clearly, if N lies in a high dimension sampling becomes infeasible. Therefore, we need a sampling computation that performs these operations in the dimensions of low rank L.

Let us define $X^l \in \mathbb{R}^{mn}$ as the samples from a standard MND. Then, the samples θ_t^s can be computed analytically as,

$$\begin{aligned} &\theta_t^s = \theta_{\text{MAP}} + F^c X^l \text{ where} \\ &F^c = D^{-\frac{1}{2}}(I_{nm} - D^{-\frac{1}{2}}(U_a \otimes U_g)\Lambda_{1:L}^{\frac{1}{2}} \\ &\underbrace{(C^{-1} + V_s^T V_s)^{-1}}_{\text{cost: } O(L^3) \ll O(N^3)} \Lambda_{1:L}^{\frac{1}{2}}(U_a \otimes U_g)^T D^{-\frac{1}{2}}). \end{aligned} \tag{6}$$

Firstly, the symmetrical factor $F^c \in \mathbb{R}^{mn \times mn}$ in (6) is a function of matrices that are feasible to store as they are diagonal or small Kronecker factored matrices. Furthermore,

$$V_s = D^{-\frac{1}{2}}(U_a \otimes U_g)\Lambda_{1:L}^{\frac{1}{2}} \text{ and } C = A_c^{-T}(B_c - I_L)A_c^{-1}$$

with A_c and B_c being the Cholesky decomposed matrices of $V_s^T V_s \in \mathbb{R}^{L \times L}$ and $V_s^T V_s + I_L \in \mathbb{R}^{L \times L}$ such that:

$$A_c A_c^T = V_s^T V_s \text{ and } B_c B_c^T = V_s^T V_s + I_L.$$

Consequently, the matrices in (6) are defined as $C \in \mathbb{R}^{L \times L}$, $(C^{-1} + V_s^T V_s) \in \mathbb{R}^{L \times L}$ and identity matrix $I_L \in \mathbb{R}^{L \times L}$. In this way, the two operations namely Cholesky decomposition and inversion that are cubic in cost $O(N^3)$ are reduced to the low rank dimension L with complexity $O(L^3)$. The complete derivation is in supplementary materials where we further show how the Kronecker structure in eigendecomposition can be exploited to compute $F^c X^l$ using the *vec* trick.

2.5. Theoretical Guarantees

We outline theoretical guarantees of our approach below. Note that these properties hold regardless of data and model.

Lemma 1: *Let $\boldsymbol{I}$ be the real information matrix, and let $\boldsymbol{I}_{inf}$ and $\boldsymbol{I}_{efb}$ be the INF and EFB estimates of it respectively. It is guaranteed to have $\left\|\boldsymbol{I} - \boldsymbol{I}_{efb}\right\|_F \geq \left\|\boldsymbol{I} - \boldsymbol{I}_{inf}\right\|_F$.*

Corollary 1: *Let $\boldsymbol{I}_{kfac}$ and $\boldsymbol{I}_{inf}$ be KFAC and our estimates of real information matrix $\boldsymbol{I}$ respectively. Then, it is guaranteed to have $\left\|\boldsymbol{I} - \boldsymbol{I}_{kfac}\right\|_F \geq \left\|\boldsymbol{I} - \boldsymbol{I}_{inf}\right\|_F$.*

Lemma 2: *Let $\boldsymbol{I}$ be the real information matrix, and let $\hat{\boldsymbol{I}}_{inf}$, $\boldsymbol{I}_{efb}$ and $\boldsymbol{I}_{kfac}$ be the low rank INF, EFB and KFAC estimates of it respectively. Then, it is guaranteed to have $\left\| diag(\boldsymbol{I}) - diag(\boldsymbol{I}_{efb})\right\|_F \geq \left\| diag(\boldsymbol{I}) - diag(\hat{\boldsymbol{I}}_{inf})\right\|_F = 0$ and $\left\| diag(\boldsymbol{I}) - diag(\boldsymbol{I}_{kfac})\right\|_F \geq \left\| diag(\boldsymbol{I}) - diag(\hat{\boldsymbol{I}}_{inf})\right\|_F = 0$. Furthermore, if the eigenvalues of $\hat{\boldsymbol{I}}_{inf}$ contains all non-zero eigenvalues of $\boldsymbol{I}_{inf}$, it follows: $\left\|\boldsymbol{I} - \boldsymbol{I}_{efb}\right\|_F \geq \left\|\boldsymbol{I} - \hat{\boldsymbol{I}}_{inf}\right\|_F$.*

Proofs along with a further remarks and theoretical analysis on (i) error bounds of the proposed LRA and (ii) validity of the posterior can be found in supplementary materials.

3. Related Works

Sparse Information Filters: Thrun et al. (2004) proposed the extended sparse information filter (SEIF), which is a dual form of the extended Kalman filter. A key property of SEIF is that all update equations can be executed in constant time, which is achieved by relying on the information form and its sparsity. Our work brings the key ideas of SEIF in the context of approximate Bayesian inference for DNNs.

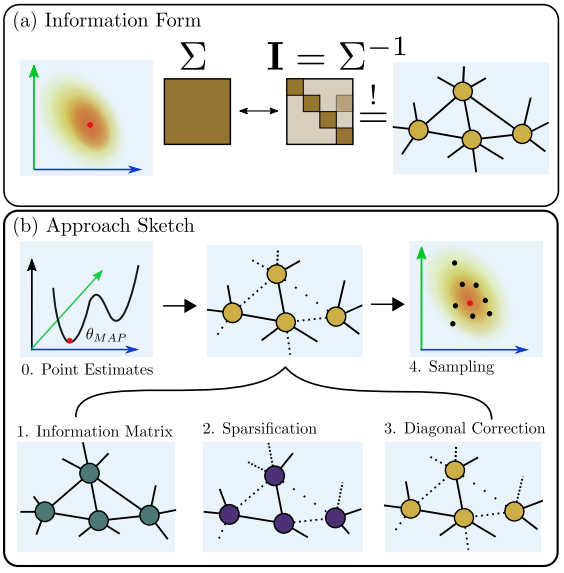

Figure 2. **Illustration of the main idea and the pipeline sketch**. We demonstrate that approximate Bayesian inference can work with the inverse covariance matrix - the information matrix. From its graphical interpretation (Paskin, 2003), the diagonal elements represent the information content of the node while its off-diagonal elements represent the link between the nodes. Our approach is designed with this insight where we sparsify the weak links while keeping the information content of the node accurate.

Approximation of the Hessian: The Hessian of DNNs is prohibitively too large as its size is quadratic to the number of parameters. For this problem, an efficient approximation is a layer-wise Kronecker factorization (Martens & Grosse, 2015; Botev et al., 2017) with demonstrably impressive scalability (Ba et al., 2017). In a recent extension by George et al. (2018) the eigenvalues of the Kronecker factored matrices are re-scaled so that the diagonal variance in its eigenbasis is exact. The work demonstrates a provable method of achieving improved performance. We heavily build upon these for Bayesian DNNs, as well-built software infrastructures such as (Dangel et al., 2020) already exists.

Laplace Approximation: Instead of methods rooted in variational inference (Hinton & van Camp, 1993) and sampling (Neal, 1996), we utilize LA (MacKay, 1992b) for the inference principle. Recently, diagonal (Becker & Lecun, 1989) and Kronecker-factored approximations (Botev et al., 2017) to the Hessian have been applied to LA by Ritter et al. (2018a). The authors have further proposed to use LA in continual learning (Ritter et al., 2018b), and demonstrate competitive results by significantly outperforming its benchmarks (Kirkpatrick et al., 2017; Zenke et al., 2017). Building upon Ritter et al. (2018a) for approximate inference, we propose to use more expressive posterior distribution than matrix normal distribution. Concurrently, Kristiadi et al.

(2020) provides a formal statement on how approximate inference such as LA can mitigate overconfident behavior of DNNs with ReLU for a binary classification case.

In the context of variational inference, SLANG (Mishkin et al., 2018) share similar spirit to ours in using a low-rank plus diagonal form of covariance where the authors show the benefits of low-rank approximation in detail. Yet, SLANG is different to ours as it does not explore Kronecker structures. SWA (Maddox et al., 2018), SWAG (Maddox et al., 2019) and subspace inference (Izmailov et al., 2019) have also demonstrated strong results by exploring the insights on loss landscape of DNNs. We also acknowledge there exists alternatives paradigms. Some examples are the post-hoc calibrations (Guo et al., 2017; Wenger et al., 2020), ensembles (Lakshminarayanan et al., 2017) and combining Bayesian Neural Networks with probabilistic graphical models such as Conditional Random Fields (Feng et al., 2019).

Dimensionality Reduction: A vast literature is available for dimensionality reduction beyond principal component analysis (Wold et al., 1987) and singular value decomposition (Golub & Reinsch, 1971; Van Der Maaten et al., 2009). To our knowledge though, dimensionality reduction in Kronecker factored eigendecomposition has not been studied.

4. Experimental Results

We perform an empirical study with regression, classification and active learning tasks. The chosen datasets are toy regression, UCI (Dua & Graff, 2017), MNIST (Lecun et al., 1998), CIFAR10 (Krizhevsky, 2009) and ImageNet (Krizhevsky et al., 2012) datasets. In total, 10 baseline with default 3 LA-based approaches (Diag, KFAC and EFB) are compared, and we also study the effects of varying LRA.

Our main aim is to introduce the sparse information form of MND in the context of Bayesian Deep Learning, and thus, the experiments are designed to show the insight that spectral sparsification does not induce significant approximation errors while reducing the space complexity of using an expressive and structured posterior distribution. More importantly, we demonstrate that our method scales to datasets of ImageNet size and large architectures, and push the method further to compete with existing approaches. Implementation details can be found in supplementary materials.

4.1. Small Scale Experiments

Firstly, evaluations on toy regression and UCI datasets are presented. Due to the small scale of the set-up, these experiments have advantages that we can not only evaluate the quality of uncertainty estimation, but also directly compare various approximations to the Hessian with LRA. The later empirically validates our theoretic claims and further connects the qualities of uncertainty estimates and the approximates of the Hessian. For the toy regression problem, we consider a single-layered network with 7 units. We have used 100 uniformly distributed points $x \sim U(-4,4)$ and samples $y \sim N(x^3, 3^2)$. On UCI datasets we follow Herandez-Lobato & Adams (2015) in which each dataset are split into 20 sets. A single layered network with 50 units are used with an exception of protein, where we have used 100 units.

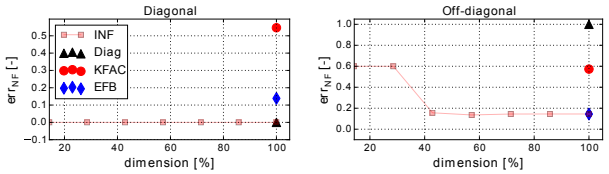

Figure 3. **Effects of Low Rank Approximation in Frobenius norm of error.** This measure is normalized. Lower the better. EFB, Diag, KFAC and INF are compared in terms of diagonal and off-diagonal errors to exact block diagonal information matrix.

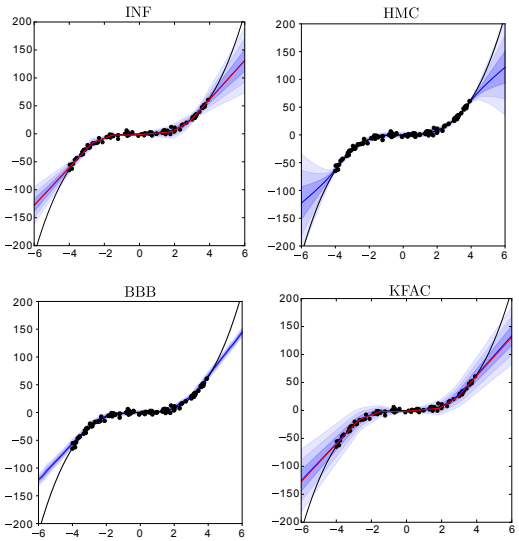

Figure 4. **Predictive uncertainty.** The black dots and the black lines are data points (x, y). The red and blue lines show predictions of the deterministic network and the mean output respectively. Upto three standard deviations are shown with blue shades.

We initially perform a direct evaluation of computed IM with a measure on normalized Frobenius norm of error err_{NF} w.r.t the block-wise exact IM. Note that this is only possible for the small network architectures. The results are shown in figure 3 and table 1 for toy regression and UCI datasets respectively. Across all the experiments, we find that INF is exact on diagonal elements regardless of the ranks while KFAC and EFB induces significant errors. On off-diagonals, INF with full rank performs similar to EFB while decreasing the ranks of INF tend to increase the errors. Interestingly, INF with only 5% of the ranks, often outperforms KFAC

Table 1. **Evaluating the accuracy of information matrix on UCI datasets.** The Frobenius norm of errors for diagonal and off-diagonal approximations w.r.t the exact block diagonal information matrix are depicted. Reported values are normalized. Here, INF (5%) indicates that 95% of the ranks are thrown away, and Nr refers to the number of data points for each dataset. Lower the better.

Dataset		Diagonals				Off-diagonals			
	Nr	**KFAC**	**EFB**	**INF**	**INF (5%)**	**KFAC**	**EFB**	**INF**	**INF (5%)**
Boston	506	0.238±0.019	0.296±0.015	**0.000±0.000**	**0.000±0.000**	0.672±0.021	**0.524±0.006**	**0.524±0.006**	*0.608±0.009*
Concrete	1030	0.185±0.020	0.253±0.008	**0.000±0.000**	**0.000±0.000**	0.632±0.018	**0.506±0.008**	**0.506±0.008**	0.639±0.008
Energy	768	0.138±0.035	0.335±0.029	**0.000±0.000**	**0.000±0.000**	0.646±0.012	**0.504±0.006**	**0.504±0.006**	*0.619±0.012*
Kin8nm	8192	0.077±0.008	0.256±0.020	**0.000±0.000**	**0.000±0.000**	0.594±0.005	**0.526±0.005**	**0.526±0.005**	0.670±0.003
Naval	11934	0.235±0.024	0.224±0.026	**0.000±0.000**	**0.000±0.000**	0.716±0.029	**0.465±0.003**	**0.465±0.003**	*0.480±0.003*
Power	9568	0.113±0.012	0.252 ±0.011	**0.000±0.000**	**0.000±0.000**	0.681±0.006	**0.492±0.008**	**0.492±0.008**	*0.570±0.009*
Protein	45730	0.323±0.067	0.332±0.043	**0.000±0.000**	**0.000±0.000**	0.779±0.040	**0.541±0.021**	**0.541±0.021**	*0.548±0.019*
Wine	1599	0.221±0.021	0.287±0.022	**0.000±0.000**	**0.000±0.000**	0.638±0.006	**0.535±0.009**	**0.535±0.009**	0.685±0.006
Yacht	308	0.104±0.007	0.201±0.019	**0.000±0.000**	**0.000±0.000**	0.653±0.009	**0.516±0.007**	**0.516±0.007**	0.699±0.007

by a significant margin (e.g. Naval, Power, Protein). These observations are expected by the design of our approach and highlights the benefits of the information form. As the exact diagonals of IM are known and simple to compute (as oppose to the covariance matrix), we can design methods with theoretical guarantees on approximation quality of IM. Moreover, as the spectrum of IM tends to be sparse, LRA can be effectively exploited without inducing significant errors. For UCI experiments, we further study the varying effects of LRA in supplementary materials.

Visualization of predictive uncertainty is shown in figure 4 for the toy regression. Here, HMC (Neal, 1996) acts as ground truth while we compare our approach to KFAC and Bayes-by-backprop or BBB (Blundell et al., 2015). The hyperparameter sets for KFAC is chosen similar to (Ritter et al., 2018a) while INF did not require the tuning of hyperparameters (after ensuring that IM is non-degenerate similar to MacKay (1992b)). All the methods show higher uncertainty in the regimes far away from training data where BBB showing the most difference to HMC. Furthermore, KFAC predicts rather high uncertainty even within the regions that are covered by the training data. INF produces the most comparable fit to HMC with accurate uncertainty estimates. In supplementary materials, further comparison studies can be found. Furthermore, we also report the results of UCI experiments where we compare the reliability of uncertainty estimates using the test log-likelihood as a measure, and demonstrate competitive results as well as limitations.

4.2. Active Learning

We next show that our method can also perform a downstream task such as active learning. In this scenario, we further study the effects of LRA. For this purpose, we choose 3 UCI datasets (boston housing, wine and energy) closely following Hernández-Lobato & Adams (2015). In details the model is a neural network with a single layer of 10 hidden units. We employ the same criterion as MacKay (1992a) which linearizes the propagation of variance of weights to the output. Besides comparing with a baseline: random

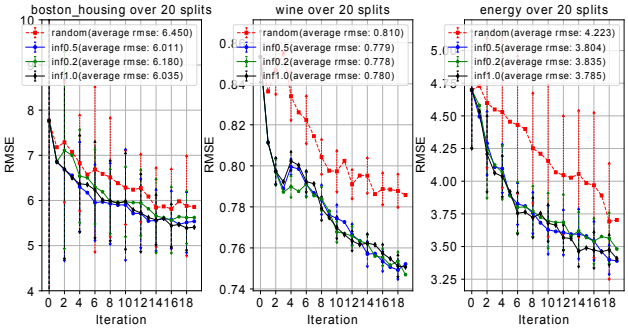

Figure 5. **Active learning with INF.** Average test RMSE over 20 splits and their standard errors in the active learning experiments with Boston Housing, Wine and Energy datasets. The average RMSE along 20 iterations of each curve is inside the bracket.

selection strategy, we also compare 3 different number of ranks (20%, 50%, 100%) to verify the influence of LRA.

As shown in figure 5, uncertainty estimates of INF enable the model to learn more quickly and lead to statistically significant improvements when compared to the random selection strategy. Further, even after cutting the ranks significantly, the performance can be maintained. This can be clearly seen on both Boston Housing and Energy dataset: the mean RMSE of lower percentage version decreased, while their standard deviation mostly overlap. On Wine, there is nearly no decrease in performance with lower ranks. To summarize, our experiments show that INF can perform active learning, and LRA does not jeopardize the task.

4.3. Classification Tasks

Next, we evaluate predictive uncertainty on classification tasks where the proposed LRA is strictly necessary. To this end, we choose the classification tasks with known and unknown classes, e.g. a network is not only trained and evaluated on MNIST but also tested on notMNIST. Note that under such tests, any probabilistic methods should report their evaluations on both known and unknown classes with the same hyperparameter settings. This is because Bayesian

Table 2. **Results of classification experiments.** Accuracy and ECE are evaluated on in-domain distribution (MNIST and CIFAR10) whereas entropy is evaluated on out-of-distribution (notMNIST and SHVN). Lower the better for ECE. Higher the better for entropy.

Experiment	Measure	NN	Diag	KFAC	MC-dropout	Ensemble	EFB	INF
MNIST vs notMNIST	*Accuracy*	0.993	0.9935	0.9929	0.9929	**0.9937**	0.9929	0.9927
	ECE	0.395	0.0075	0.0078	0.0105	0.0635	0.012	**0.0069**
	Entropy	0.055±0.133	0.555 ± 0.196	0.599 ± 0.199	0.562 ± 0.19	0.596 ± 0.133	0.618 ± 0.185	**0.635 ± 0.19**
CIFAR10 vs SHVN	*Accuracy*	0.8606	**0.8659**	0.8572	N/A	0.8651	0.8638	0.8646
	ECE	0.0819	0.0358	0.0351	N/A	0.0809	0.0343	**0.0084**
	Entropy	0.245 ± 0.215	0.4129 ± 0.197	0.408 ± 0.197	N/A	0.370 ± 0.192	0.417 ± 0.196	**0.4338 ± 0.18**

Neural Networks can be always highly uncertain, which may seem to work well for out-of-distribution (OOD) detection tasks but overestimates the uncertainty, even for the correctly classified samples within the training data distribution. For evaluating predictive uncertainty on known classes, Expectation Calibration Error (ECE Guo et al. (2017)) has been used. Normalized entropy is reported for evaluating predictive uncertainty on unknown classes. LeNet with ReLU and a L2 coefficient of 1e-8 has been chosen for MNIST dataset, which constitutes of 2 convolution layers followed by 2 fully connected layers. The networks are intentionally trained to over-fit or over-confident, so that we can observe the effects of capturing model uncertainty. For CIFAR10, we choose VGG like architecture with 2 convolution layers followed by 3 fully connected layers. We used batch normalization instead of dropout layers.

The results are reported in table 2 where we also compared to MC-dropout (Gal, 2016) and deep ensemble (Lakshminarayanan et al., 2017), which are widely used baselines in practice. For CIFAR10, we omitted MC-dropout as additionally inserting dropout layers would result in a different network and thus, the direct comparisons would not be difficult. For LA-based methods, we have reported the best results after searching 300 hyperparameters each. We find this evaluation protocol to be crucial, as LA-based methods are sensitive to these regularizing hyperparameters. Importantly, these results demonstrate that when projected to different success criteria, no inference methods largely win uniformly. Yet these experiments also show empirical evidence that our method works in principle and compares well to the current state-of-the-art. Estimating the layer-wise parameter posterior distribution in a sparse information form of MND, and demonstrating a low rank sampling computations, we show an alternative approach of designing scalable, high performance and practical inference framework.

4.4. Large Scale Experiments

To show the scalability of our method, we conduct an extensive experimental evaluation on the ImageNet dataset, using 5 DNNs. This result alone is a key benefit of estimating model uncertainty of DNNs in sparse information form since approximate Bayesian inference only involves the computations of the information matrix in a training-free

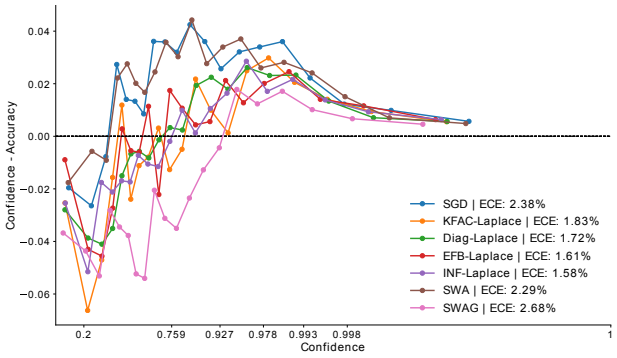

Figure 6. **Reliability diagram of ResNet18 on ImageNet** Showing a calibration comparison of a deterministic forward pass (SGD) against Diag, KFAC, EKFAC, INF as well as SWA and SWAG.

manner and the method boosts relaxed assumptions about the model when compared to MC-dropout (e.g. no specific regularizer is required). We also emphasize that the proposed diagonal correction requires only back-propagated gradients. EFB also uses the same gradients in their update step, and the whole chain takes around 8 hours on ImageNet with 1 NVIDIA Volta GPU. This means that one can store the exact diagonals of the Fisher during EFB computations and simply add a correction term without involving any data. Thus, the added computational overhead due to the diagonal correction is negligible in practice. Similar to section 4.3, we evaluate the both calibration and OOD detection performances. SWAG and SWA are the chosen baselines, as Maddox et al. (2019) demonstrated that these methods scale to the ImageNet dataset. Moreover, to achieve comparability for large-scale settings, we scaled the other LA-based methods up so that they are applicable to ImageNet. This was not available in the original papers, and it constitutes an advance in evaluating LA-based methods for realistic scenarios. More importantly, we perform an extensive hyperparameter search over 100 randomly selected configurations for each LA-based methods. We believe that such protocols are required for fair evaluations of LA-based methods.

To directly compare the calibration across different methods, a variant of the reliability diagram (Maddox et al., 2019) is used (figure 6 for ResNet 18). We present results for all other

Table 3. **Network Space Complexity Comparison:** The total number of information matrix parameters and its size in MB are reported for ResNet and DenseNet variants. Lower the better. Here, we also check if the methods take into account the weight correlations (corr).

	Diag			KFAC			EFB			INF		
Model	#Parameters	Size	Corr	#Parameters	Size	Corr	#Parameters	Size	Corr	#Parameters	Size	Corr
ResNet18	11,679,912	*44.6*	X	95,013,546	362.4	✓	106,693,458	407.0	✓	12,317,373	**47.0**	✓
ResNet50	25,503,912	*97.3*	X	153,851,562	586.9	✓	179,355,474	684.2	✓	27,614,896	**105.3**	✓
ResNet152	60,041,384	*229.0*	X	389,519,018	1485.9	✓	449,560,402	1714.9	✓	65,558,402	**250.1**	✓
DenseNet121	7,895,208	*30.1*	X	103,094,954	393.3	✓	110,990,162	423.4	✓	9,711,081	**37.0**	✓
DenseNet161	28,461,064	*108.6*	X	379,105,514	1446.2	✓	407,566,578	1554.7	✓	32,329,191	**123.3**	✓

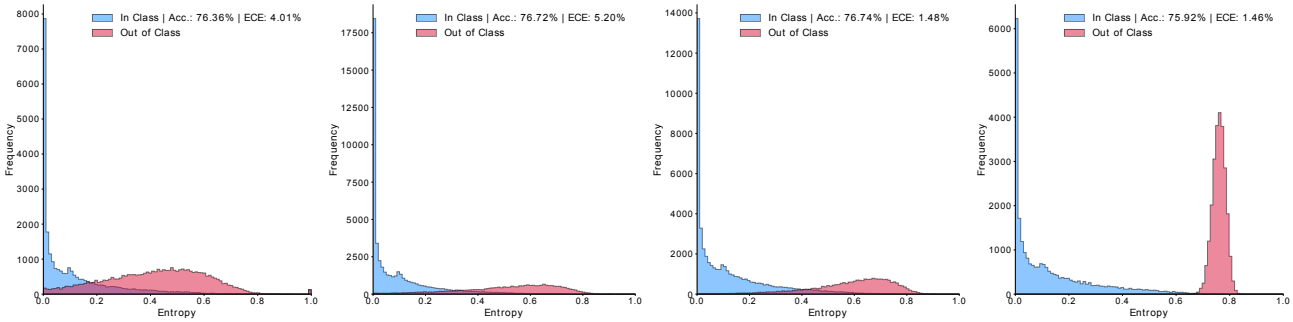

Figure 7. **Entropy histograms ResNet50:** From left to right: SGD, SWA, SWAG and INF. While SGD, SWA and SWAG fail to separate in- and out-of-domain data, INF is able to almost completely differentiate between known and unknown data.

architectures in supplementary materials. Here, we observe that all LA variants significantly outperform SGD, SWA and SWAG. Results of OOD detection tasks are reported in figure 7 where we also show the predictive uncertainty of the in-domain data for checking the under-confident behavior (Grimmett et al., 2015; Mund et al., 2015; Grimmett et al., 2013). We use artistic impressions and paintings of landscapes and objects as OOD data. Again, SGD, SWAG, and SWA are in turn significantly outperformed by INF, which clearly separates in-distribution and OOD data. These results show the competitiveness of our approach for the real-world applications. However, we also find that all the LA-based methods almost identically yield strong results, clearly separating the in-distribution and OOD data.

While our method scales to ImageNet, we do not find that improvements in terms of Frobenius norm of error translates to performance in uncertainty estimation. This is the effects of regularizing hyperparameters (τ and N) which is a limitation of LA based approaches. We find a counter-intuitive result that Diag LA performs similar to ours and KFAC. It is therefore a strong alternative in practice where its complexity is superior than others (see table 3). Yet, ours, as we use rank 100 in the experiments, are significantly superior in space complexity when compared to EFB and KFAC, while modeling the weight correlations. Therefore, we find the main use-case of INF: the applications such as aerial systems (Lee et al., 2018; 2020) and medical devices (Petrou et al., 2018a;b) which cannot afford much more memory due to the limited on-board computations, and requires structured form of model uncertainty, unlike Diag. Last but not least, the mathematical tools we develop and the idea of working on the inverse space of MND can also be useful in the context of variational inference as an example.

5. Conclusion

This work introduces the sparse information form as an alternative Gaussian posterior family for which, we presented novel mathematical tools such as a sparsification algorithm for the Kronecker factored eigendecomposition, and demonstrated how to efficiently sample from the resulting distribution. Our experiments show that our approach yields accurate estimates of the information matrix with theoretical guarantees, compares well to the current methods for the task of uncertainty estimation, scales to large scale data-sets while reducing space complexity, and can also be utilized for downstream tasks such as active learning.

Acknowledgements

We thank the anonymous reviewers and area chairs for their time and thoughtful comments. The authors also acknowledge the support of Helmholtz Association, the project ARCHES (contract number ZT-0033) and the EU-project AUTOPILOT (contract number 731993). Jianxiang Feng is supported by the Munich School for Data Science (MUDS) and Rudolph Triebel is a member of MUDS.

References

Ba, J., Grosse, R. B., and Martens, J. Distributed second-order optimization using kronecker-factored approximations. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

Bailey, T. and Durrant-Whyte, H. Simultaneous localization and mapping (slam): Part ii. *IEEE robotics & automation magazine*, 13(3):108–117, 2006.

Becker, S. and Lecun, Y. Improving the convergence of back-propagation learning with second-order methods. In Touretzky, D., Hinton, G., and Sejnowski, T. (eds.), *Proceedings of the 1988 Connectionist Models Summer School, San Mateo*, pp. 29–37. Morgan Kaufmann, 1989.

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. Weight uncertainty in neural networks. In *Proceedings of the 32Nd International Conference on Machine Learning - Volume 37*, ICML'15, pp. 1613–1622. JMLR.org, 2015.

Botev, A., Ritter, H., and Barber, D. Practical Gauss-Newton optimisation for deep learning. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 557–565, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

Dangel, F., Kunstner, F., and Hennig, P. Backpack: Packing more into backprop. In *International Conference on Learning Representations*, 2020.

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

Eustice, R. M., Singh, H., and Leonard, J. J. Exactly sparse delayed-state filters for view-based slam. *IEEE Transactions on Robotics*, 22(6):1100–1114, 2006.

Feng, J., Durner, M., Marton, Z.-C., Balint-Benczedi, F., and Triebel, R. Introspective robot perception using smoothed predictions from bayesian neural networks. In *International Symposium on Robotic Research (ISRR)*, 2019.

Gal, Y. *Uncertainty in Deep Learning*. PhD thesis, University of Cambridge, 2016.

George, T., Laurent, C., Bouthillier, X., Ballas, N., and Vincent, P. Fast approximate natural gradient descent in a kronecker factored eigenbasis. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.*, pp. 9573–9583, 2018.

Golub, G. H. and Reinsch, C. Singular value decomposition and least squares solutions. In *Linear Algebra*, pp. 134–151. Springer, 1971.

Graves, A. Practical variational inference for neural networks. In Shawe-Taylor, J., Zemel, R. S., Bartlett, P. L., Pereira, F., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 24*, pp. 2348–2356. Curran Associates, Inc., 2011.

Grimmett, H., Paul, R., Triebel, R., and Posner, I. Knowing when we don't know: Introspective classification for mission-critical decision making. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

Grimmett, H., Triebel, R., Paul, R., and Posner, I. Introspective classification for robot perception. *The International Journal of Robotics Research (IJRR)*, 2015.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1321–1330, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

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

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. E. and van Camp, D. Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the Sixth Annual Conference on Computational Learning Theory*, COLT '93, pp. 5–13, New York, NY, USA, 1993. ACM. ISBN 0-89791-611-5.

Izmailov, P., Maddox, W., Kirichenko, P., Garipov, T., Vetrov, D. P., and Wilson, A. G. Subspace inference for bayesian deep learning. In *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI 2019, Tel Aviv, Israel, July 22-25, 2019*, pp. 435, 2019.

Kingma, D. P., Salimans, T., and Welling, M. Variational Dropout and the Local Reparameterization Trick. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 28*, pp. 2575–2583. Curran Associates, Inc., 2015.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N. C., 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 of the United States of America*, 114 13:3521–3526, 2017.

Kristiadi, A., Hein, M., and Hennig, P. Being bayesian, even just a bit, fixes overconfidence in relu networks. In *Proceedings of the 37th International Conference on Machine Learning*, Proceedings of Machine Learning Research, 2020.

Krizhevsky, A. Learning multiple layers of features from tiny images. Technical report, Canadian Institute for Advanced Research, 2009.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States.*, pp. 1106–1114, 2012.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pp. 6402–6413, 2017.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pp. 2278–2324, 1998.

Lee, J., Muskardin, T., Pacz, C. R., Oettershagen, P., Stastny, T., Sa, I., Siegwart, R., and Kondak, K. Towards autonomous stratospheric flight: A generic global system identification framework for fixed-wing platforms. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6233–6240. IEEE, 2018.

Lee, J., Balachandran, R., Sarkisov, Y. S., De Stefano, M., Coelho, A., Shinde, K., Kim, M. J., Triebel, R., and Kondak, K. Visual-inertial telepresence for aerial manipulation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

Louizos, C. and Welling, M. Structured and efficient variational deep learning with matrix gaussian posteriors. In Balcan, M. F. and Weinberger, K. Q. (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1708–1716, New York, New York, USA, 20–22 Jun 2016. PMLR.

MacKay, D. J. Information-based objective functions for active data selection. *Neural computation*, 4(4):590–604, 1992a.

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

Maddox, W. J., Garipov, T., Izmailov, P., Vetrov, D., and Wilson, A. G. Fast uncertainty estimates and bayesian model averaging of dnns. In *Uncertainty in Deep Learning Workshop at UAI*, 2018.

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.

Martens, J. and Grosse, R. B. Optimizing neural networks with kronecker-factored approximate curvature. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pp. 2408–2417, 2015.

Mishkin, A., Kunstner, F., Nielsen, D., Schmidt, M. W., and Khan, M. E. SLANG: fast structured covariance approximations for bayesian deep learning with natural gradient. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montreal, Canada.*, pp. 6248–6258, 2018.

Mund, D., Triebel, R., and Cremers, D. Active online confidence boosting for efficient object classification. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

Neal, R. M. *Bayesian Learning for Neural Networks*. Springer-Verlag, Berlin, Heidelberg, 1996. ISBN 0387947248.

Park, Y., Kim, C., and Kim, G. Variational Laplace autoencoders. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5032–5041, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

Paskin, M. A. Thin junction tree filters for simultaneous localization and mapping. In *in Int. Joint Conf. on Artificial Intelligence*. Citeseer, 2003.

Petrou, A., Kuster, D., Lee, J., Meboldt, M., and Daners, M. S. Comparison of flow estimators for rotary blood pumps: An in vitro and in vivo study. *Annals of biomedical engineering*, 46(12):2123–2134, 2018a.

Petrou, A., Lee, J., Dual, S., Ochsner, G., Meboldt, M., and Schmid Daners, M. Standardized comparison of selected physiological controllers for rotary blood pumps: in vitro study. *Artificial organs*, 42(3):E29–E42, 2018b.

Ritter, H., Botev, A., and Barber, D. A scalable laplace approximation for neural networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018a.

Ritter, H., Botev, A., and Barber, D. Online structured laplace approximations for overcoming catastrophic forgetting. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.*, pp. 3742–3752, 2018b.

Sagun, L., Evci, U., Güney, V. U., Dauphin, Y., and Bottou, L. Empirical analysis of the hessian of over-parametrized neural networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Workshop Track Proceedings*, 2018.

Sun, S., Chen, C., and Carin, L. Learning Structured Weight Uncertainty in Bayesian Neural Networks. In Singh, A. and Zhu, J. (eds.), *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1283–1292, Fort Lauderdale, FL, USA, 20–22 Apr 2017. PMLR.

Thrun, S. and Liu, Y. Multi-robot slam with sparse extended information filers. In *Robotics Research. The Eleventh International Symposium*, pp. 254–266. Springer, 2005.

Thrun, S., Liu, Y., Koller, D., Ng, A. Y., Ghahramani, Z., and Durrant-Whyte, H. Simultaneous localization and mapping with sparse extended information filters. *The international journal of robotics research*, 23(7-8):693–716, 2004.

Van Der Maaten, L., Postma, E., and Van den Herik, J. Dimensionality reduction: a comparative. *Journal of Machine Learning Research*, 10(66-71):13, 2009.

Wenger, J., Kjellström, H., and Triebel, R. Non-parametric calibration for classification. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*, Proceedings of Machine Learning Research, 2020.

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.

Wold, S., Esbensen, K., and Geladi, P. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

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

Zenke, F., Poole, B., and Ganguli, S. Continual learning through synaptic intelligence. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3987–3995, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

Zhang, G., Sun, S., Duvenaud, D. K., and Grosse, R. B. Noisy natural gradient as variational inference. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, pp. 5847–5856, 2018.