Estimating Model Uncertainty of Neural Networks in Sparse Information Form

Jongseok Lee, Matthias Humt, Jianxiang Feng and Rudolph Triebel.

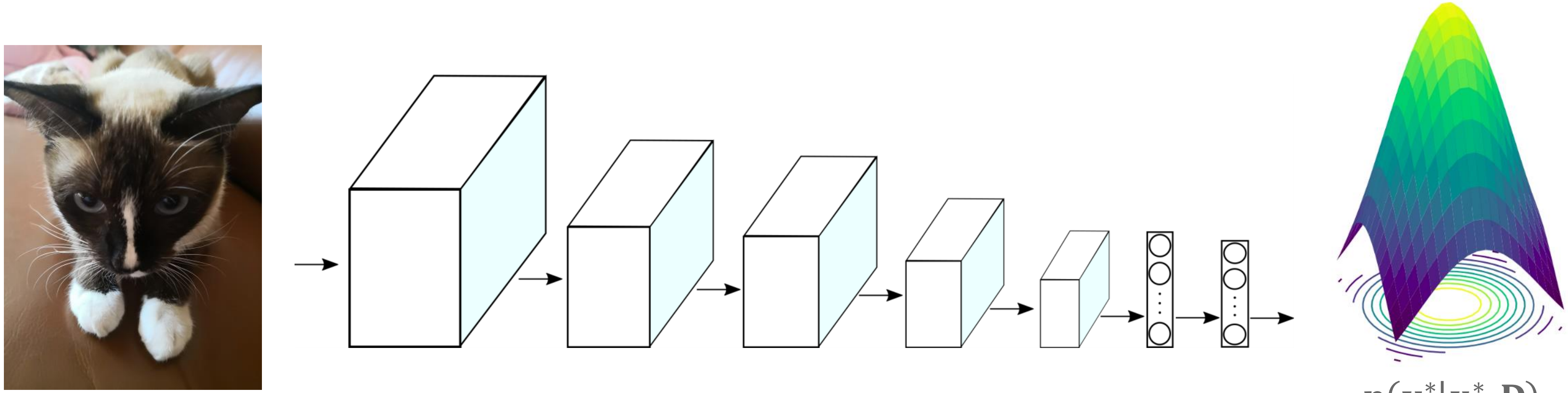
ICML 2020 Vienna.

Return distributions rather than a single, most likely prediction

[EU 2020 Autopilot – autonomous driving]

[Helmholtz ARCHES – the robot ARDEA]

Introduction to Bayesian Deep Learning



$\mathbf{p(y^*|x^*,D)}$

- **Prior:** $\mathbf{p(\theta)}$
- **Posterior:** $\mathbf{p(\theta|D) = \frac{p(y|x,\theta)p(\theta)}{p(D)}}$
- **Prediction:** $\mathbf{p(y^*|x^*,D) = \int p(y^*|x^*,\theta)p(\theta|D)d\theta}$

Parameters of a neural network

Bayes Theorem

Marginalization

Main idea: represent the posterior distribution in information form

Moments

$$\Sigma = I^{-1}$$

$$\mu = I^{-1}\mu^{IV}$$

Covariance matrix and mean vector

Information Form

$$I = \Sigma^{-1}$$

$$\mu^{IV} = \Sigma^{-1}\mu$$

Information matrix and Information vector

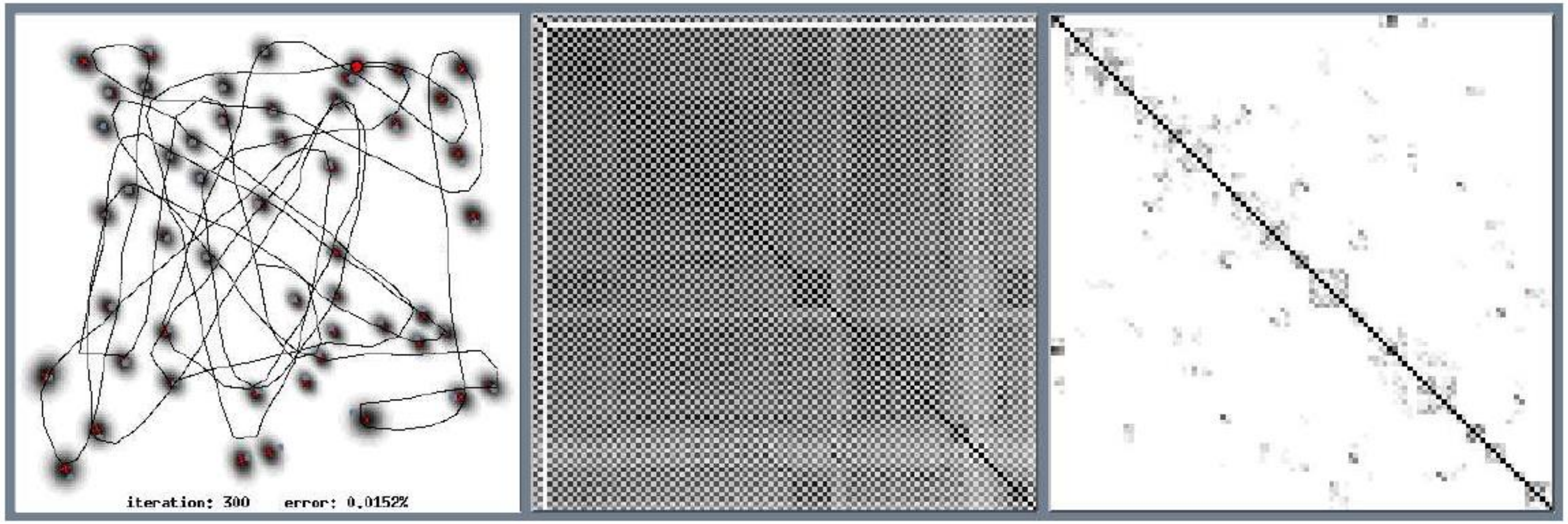
$$\theta \propto e^{-\frac{1}{2}(\theta-\mu)^{T}\Sigma^{-1}(\theta-\mu)}$$

$$= e^{-\frac{1}{2}\theta^{T}\Sigma^{-1}\theta+\mu^{T}\Sigma^{-1}\theta}$$

$$= e^{-\frac{1}{2}\theta^{T}I\theta+\mu^{IV}\theta}$$

- Two different parameterizations for Gaussian distribution
- Propose to represent the posterior distribution in the information form

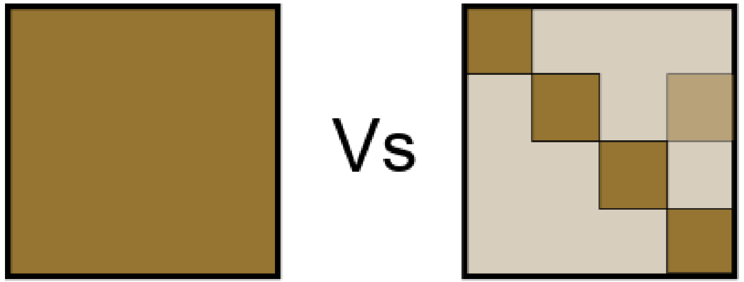
Main idea: Sparse Extended Information Filter [Thrun et al (2004)]



(a) Gaussian Bayesian tracking (b) Dense covariance (c) Sparse information matrix

Extended Kalman Filter	Sparse Extended Information Filter
• Tracks mean and covariance	• Tracks information vector and matrix
• Covariance matrix is dense	• Information matrix is sparse: - Constant time updates - Linear memory complexity

Main idea: represent the posterior distribution in information form

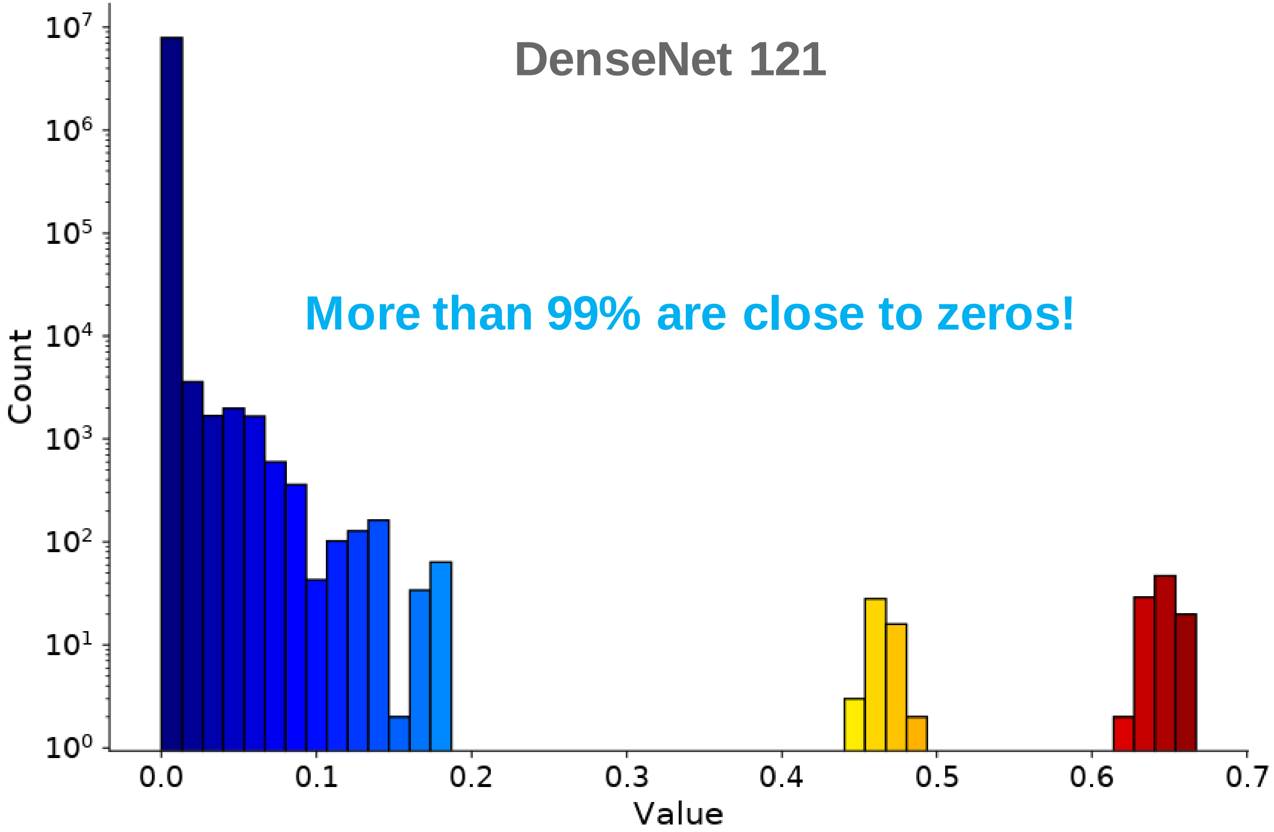


Covariance

Spectrally sparse
[Sagun et al 2018]

Sparse Information Form for Bayesian Deep Learning

- Inference using scalable Laplace Approximation
  - Theoretic guarantee on accuracy
- Information matrix is spectrally sparse:
  - Reduced space complexity
  - Competitive performance



[Thomas Bayes]

[Geoffrey Hinton]

[Sebastian Thrun]

A sparse representation for deep neural networks posterior distribution, and its scalable realization!

Table of contents

A sparse representation for deep neural networks posterior distribution, and its scalable realization!

Approximate Inference in Information Form

- **Approximate inference using Laplace Approximation:**

$$p(\theta|D) \sim \mathcal{N}\left(\theta_{map}, H^{-1}\right) \quad \text{or} \quad \sim \mathcal{N}^{-1}\left(\theta_{map}{}^{IV}, H\right)*$$

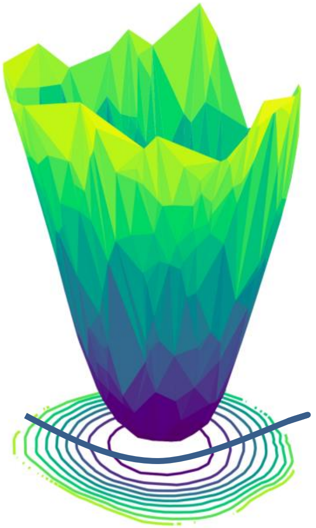
- **Employ EFB [George et al 2018] to estimate the Hessian H:**

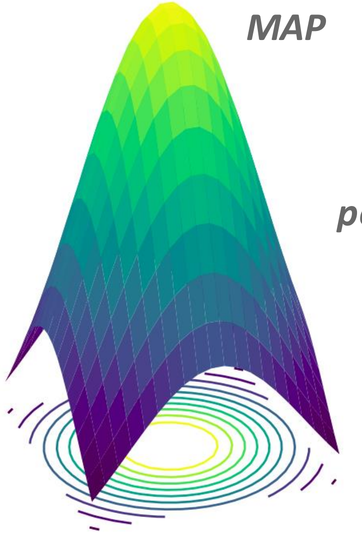
$$H \approx I_{efb} = (U_A \otimes U_G)\Lambda(U_A \otimes U_G)^T$$ **EFB Fisher Information matrix**

U_A **is an eigenvector of** $A = \mathbb{E}[aa^T]$ **Forward pass**

U_G **is an eigenvector of** $G = \mathbb{E}[gg^T]$ **Backward pass**

$$\Lambda_{ii} = \mathbb{E}[\left((U_A \otimes U_G)^T \delta\theta\right)_i^2]$$ **Eigenvalue updates**





***We omit the prior term for the simplicity of the presentation**

Approximate Inference in Information Form

- **Diagonal elements of true Information matrix is known and easy to compute!**

$\mathbf{I} = \mathbb{E}[\delta\theta\delta\theta^{T}]$ **by definition, and** $\mathbf{I}_{ii} = \mathbb{E}[\delta\theta_i^2] \ \forall i$

Not true for the covariance matrix

- **Resulting Kronecker-factored Eigen-decomposition plus diagonal structured information matrix:**

$\mathbf{I}_{inf} = (\mathbf{U}_A \otimes \mathbf{U}_G)\Lambda(\mathbf{U}_A \otimes \mathbf{U}_G)^{T} + \mathbf{D}^*$

Exact on the diagonals

- **This step brings a theoretical guarantee on improvements:**

Lemma 1: theoretical guarantees regardless of the chosen data-set and architecture
Let I be the real information matrix, and let $\mathbf{I}_{inf}$ and $\mathbf{I}_{efb}$ be the INF and EFB estimates of it respectively.
Then, it is guaranteed to have $\|\mathbf{I} - \mathbf{I}_{efb}\|_F \geq \|\mathbf{I} - \mathbf{I}_{inf}\|_F$

*** we add this term after sparsification, which will be discussed next**

Low Rank Sampling Computations

- Computing the predictive uncertainty requires samples from the posterior

$$p(y^*|x^*, D) = \int p(y^*|x^*, \theta)p(\theta|D)d\theta$$

Samples from the information form

$$\approx \frac{1}{T}\sum_{t=1}^{T} y^*(x^*, \theta_t^s) \quad \text{for} \quad \theta_t^s \sim \mathcal{N}^{-1}\left(\theta_{map}{}^{IV}, I_{inf}\right)$$

Monte-carlo integration

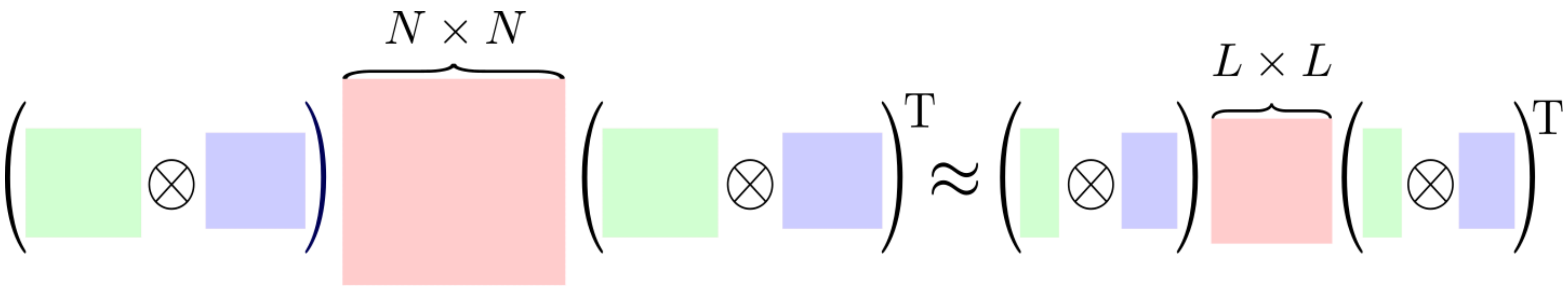
- A naive approach is not sufficient if there are many parameters (e.g. millions)

1. Evaluate the matrix: $I_{inf} = (U_A \otimes U_G)\Lambda(U_A \otimes U_G)^T + D$ — O(N2): infeasible
2. Perform Cholesky decomposition: $I_{inf}^{-1} = F_c F_c^T$ — O(N3): infeasible
3. Draw samples from the distribution: $\theta_t^s = \theta_{MAP} + F_c X^l$ with X^l the samples of a standard Gaussian

DLR

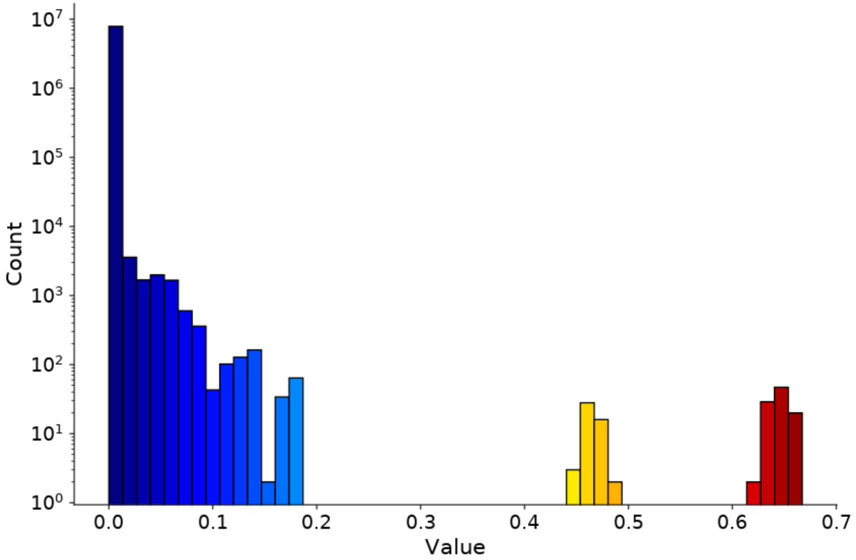
Low Rank Sampling Computations

- **Step 1: low rank approximation while saving Kronecker products in eigenvectors:**



$$(\mathbf{U_A} \otimes \mathbf{U_G})\boldsymbol{\Lambda}(\mathbf{U_A} \otimes \mathbf{U_G})^{\mathbf{T}} \approx (\mathbf{U_a} \otimes \mathbf{U_g})\boldsymbol{\Lambda}_{\mathbf{1:L}}(\mathbf{U_a} \otimes \mathbf{U_g})^{\mathbf{T}}$$

Differs from $(\mathbf{U_A} \otimes \mathbf{U_G})_{\mathbf{1:L}}\boldsymbol{\Lambda}_{\mathbf{1:1}}(\mathbf{U_A} \otimes \mathbf{U_G})^{\mathbf{T}}$



- **Step 2: samples with much lower cost and insignificant errors!**

$$\boldsymbol{\theta}_{\mathbf{t}}^{\mathbf{s}} = \boldsymbol{\theta}_{\mathbf{MAP}} + \mathbf{F_c X^l}$$

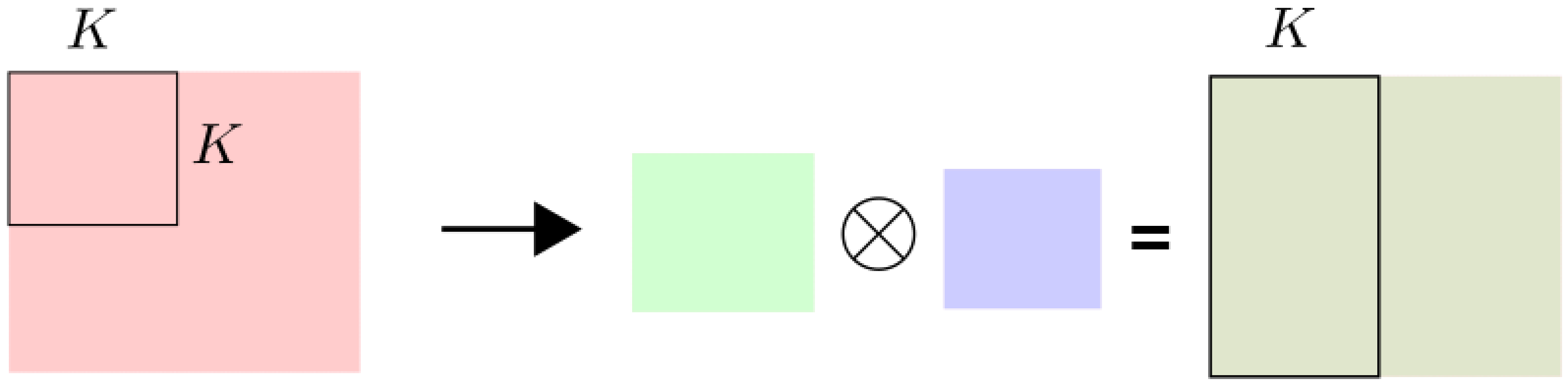
$$\mathbf{F_c} = \mathbf{D}^{-\frac{1}{2}}\left(\mathbf{I_{nm}} - \mathbf{D}^{-\frac{1}{2}}(\mathbf{U_a} \otimes \mathbf{U_g})\boldsymbol{\Lambda}_{\mathbf{1:L}}^{\frac{1}{2}}\left(\mathbf{C}^{-1} + \mathbf{V_s^T V_s}\right)^{-1}\boldsymbol{\Lambda}_{\mathbf{1:L}}^{\frac{1}{2}}(\mathbf{U_a} \otimes \mathbf{U_g})^{\mathbf{T}}\mathbf{D}^{-\frac{1}{2}}\right)$$

Sparsification algorithm

- **How to perform low rank approximation on the Kronecker-factored eigendecomposition?**

$$(\mathbf{U_A} \otimes \mathbf{U_G})\mathbf{\Lambda}(\mathbf{U_A} \otimes \mathbf{U_G})^{\mathbf{T}} \approx \left(\mathbf{U_a} \otimes \mathbf{U_g}\right)\mathbf{\Lambda_{1:L}}\left(\mathbf{U_a} \otimes \mathbf{U_g}\right)^{\mathbf{T}}$$

- **Conventional low rank approximation such as singular value decomposition:**

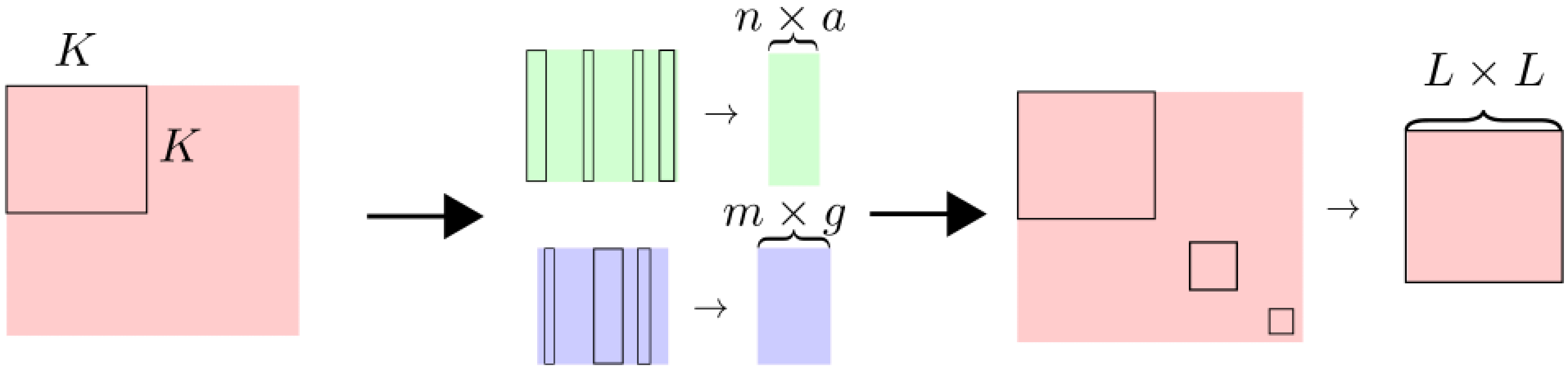


1. Find K top eigenvalues. 2. Select corresponding eigenvectors.

1. Select the top L eigenvalues and then: $\mathbf{\Lambda} \approx \mathbf{\Lambda_{1:L}}$

2. Using the indices of L eigenvalues, $\mathbf{V} = (\mathbf{U_A} \otimes \mathbf{U_G})$ **and** $\mathbf{V} \approx \mathbf{V_{1:L}}$ **Cannot preserve Kronecker structure!**

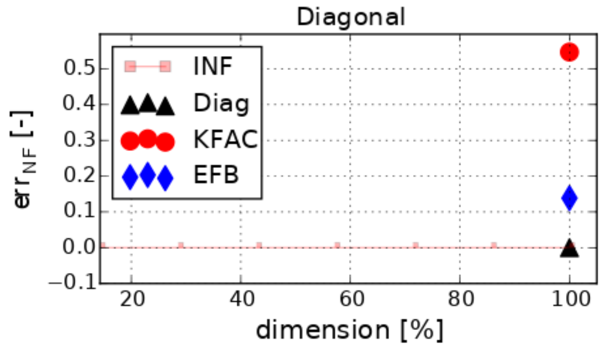
Sparsification algorithm

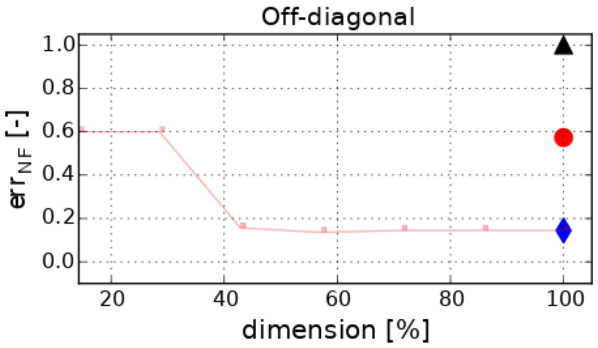


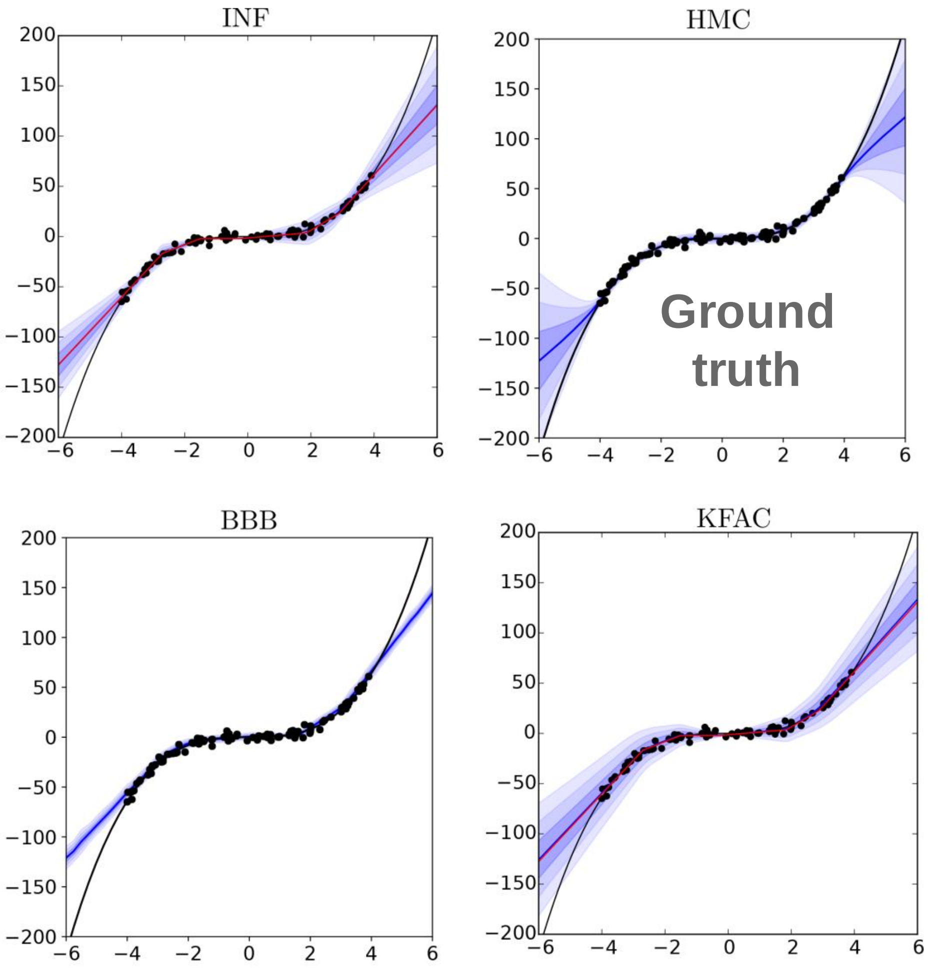
1. Find K top eigenvalues.
2. Select corresponding eigenvectors.
3. Select remaining eigenvalues.

Experiments on toy regression

- **A single layer neural network trained on a synthetic data**
- **Errors ↑ Predictive Uncertainty ↑**
- **Errors ↓ Predictive Uncertainty ↓**
- **Improved estimates of predictive uncertainty and the information matrix**







Experiments on MNIST and CIFAR10*

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**

- **Convolutional Neural Network trained MNIST and CIFAR10 datasets**
- **Calibration performance for in-domain (MNIST and CIFAR10)**
- **Normalized entropy for out-domain datasets (notMNIST and SHVN)**

***More experiments on small-scale data such as active learning on UCI can be found in the paper**

Experiments on ImageNet

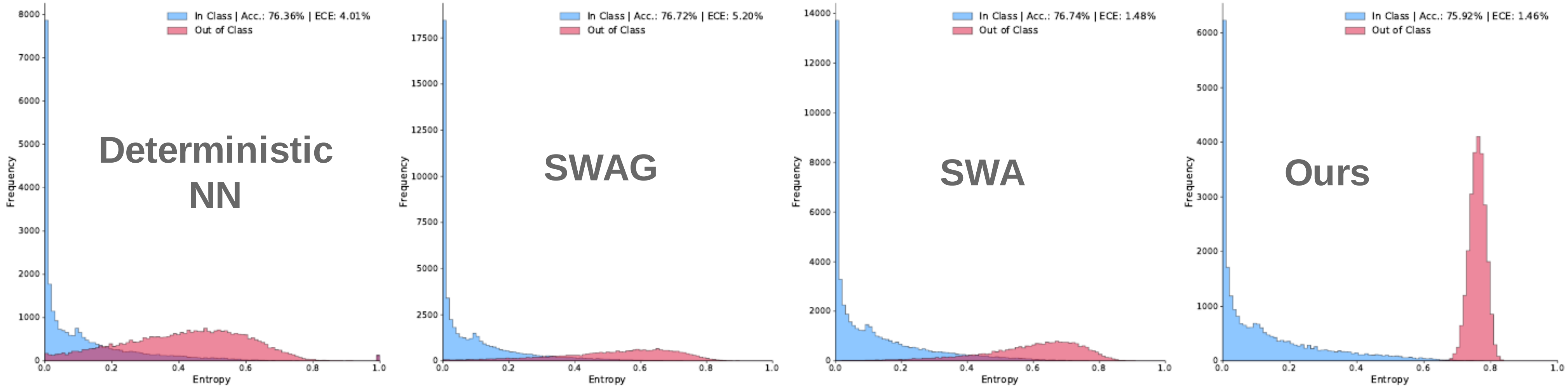


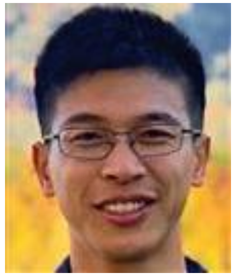
Table 3. **Network Space Complexity Comparison:** The total number of information matrix parameters and its size in MB are reported for ResNet and DenseNet variants. Lower the better. Here, we also check if the methods take into account the weight correlations (corr).

	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**	✓

Main contributions

- A novel sparse representation of the posterior distribution for deep neural networks
- Mathematical tools from approximate inference, low rank approximation to sampling computations
- Main msg: information form of Gaussian can bring certain benefits for Bayesian neural networks

Matthias Humt



Jianxiang Feng

Rudolph Triebel