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

J. Świątkowski, K. Roth, B. Veeling, L. Tran, J. Dillon, J. Snoek, S. Mandt, T. Salimans, R. Jenatton, S. Nowozin

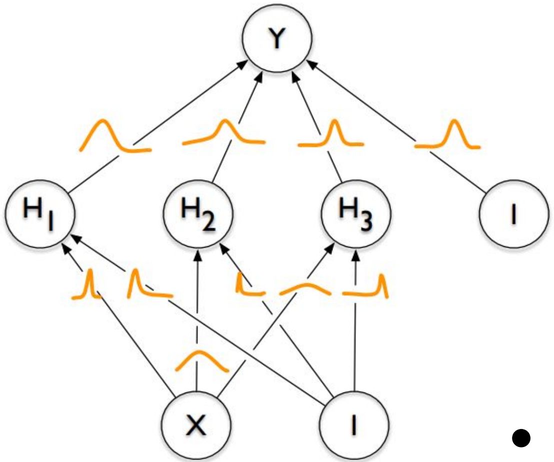









Let's bring the benefits of Bayesian inference to neural networks!



- noisy estimates of gradients
- slow convergence
- increased number of model parameters

Sources: Blundell et al. "Weight uncertainty in neural networks.", Wikipedia and Shutterstock

Preview

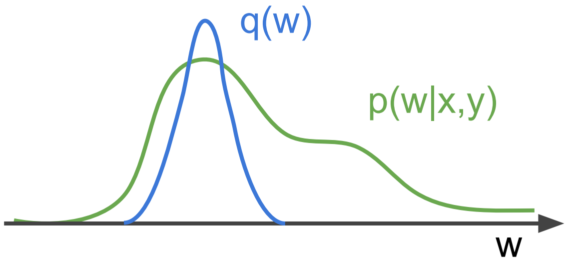
1. Gaussian Mean-Field Variational Inference (GMFVI) for Bayesian Neural Networks (BNNs).

2. Low-rank in already trained GMFVI BNNs.

3. Training a low-rank parameterization of the GMFVI BNNs.

BNN variational posterior

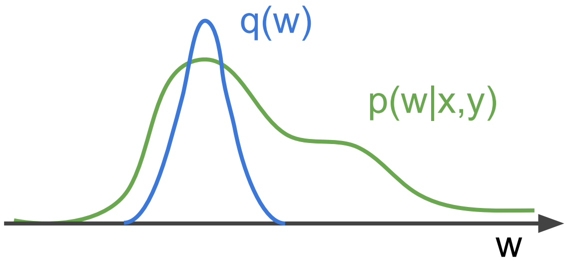
$$\theta^* = \text{argmin}_\theta D_{\text{KL}}[q_{\boldsymbol{\theta}}(\mathbf{w})||p(\mathbf{w}|\mathbf{x},\mathbf{y})]$$



- Variational inference: Cast inference as an optimization problem.

BNN variational posterior

$$\theta^* = \text{argmin}_\theta D_{\text{KL}}[q_{\boldsymbol{\theta}}(\mathbf{w})||p(\mathbf{w}|\mathbf{x},\mathbf{y})]$$



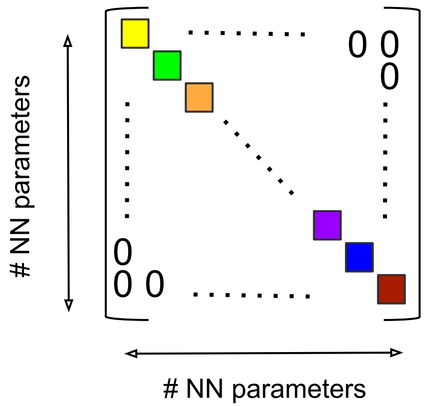
- Variational inference: Cast inference as an optimization problem.
- Key question: Which parametrization?

$$q(\mathbf{W}) = \mathcal{N}(\boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q)$$

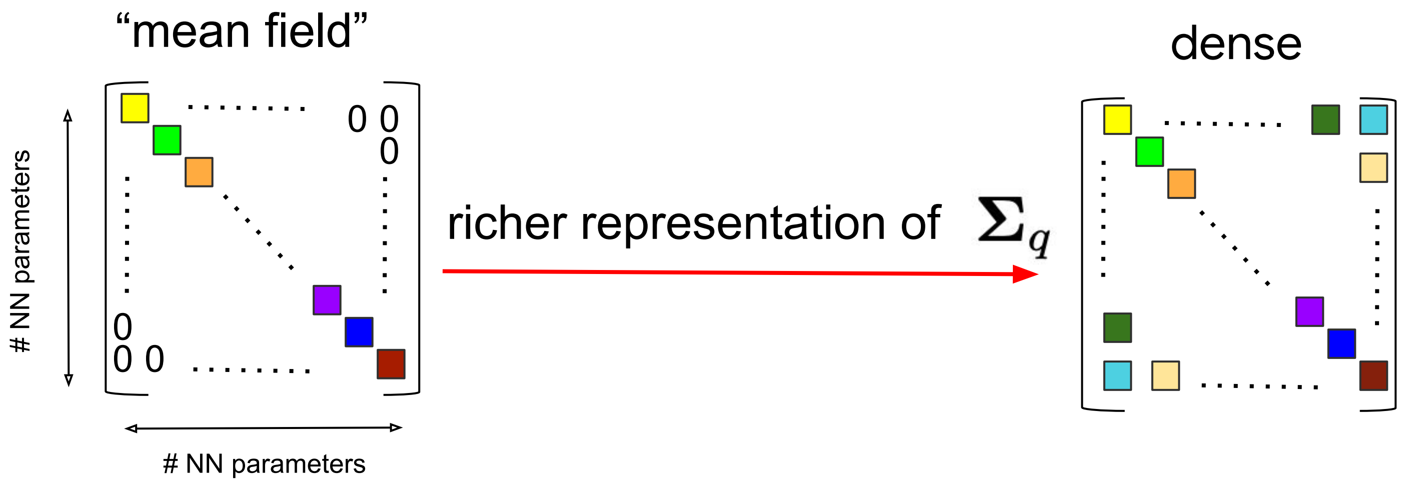
for BNNs, can be a prohibitively large object

Parametrization of variational posterior covariance

“mean field”

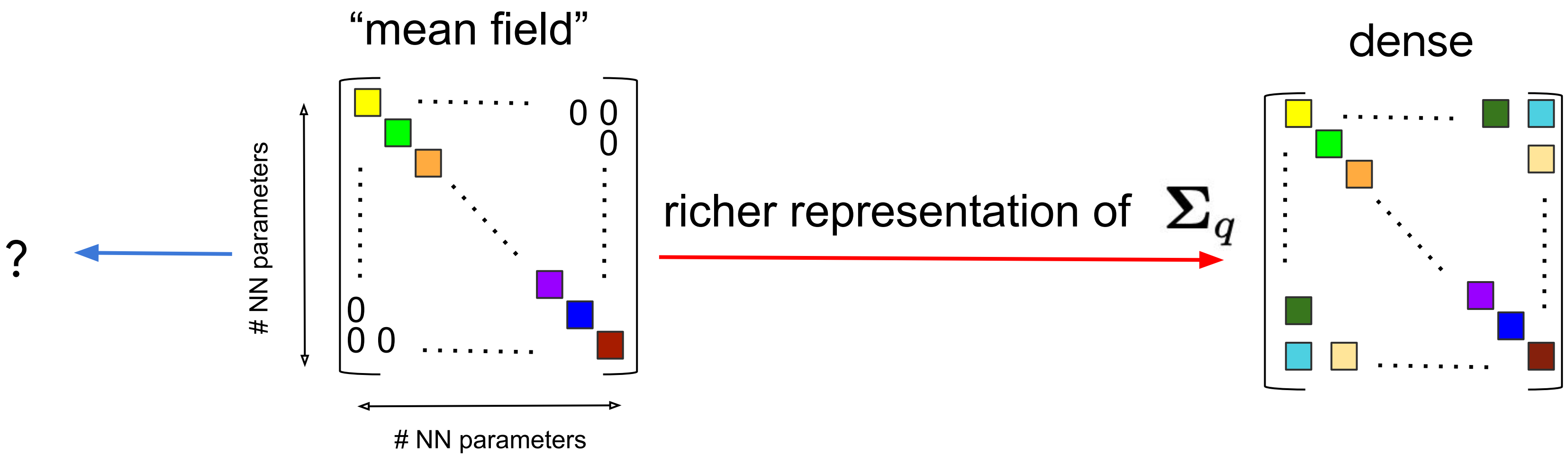


Parametrization of variational posterior covariance



- Lot of research exploring “ ”:
  - E.g., Barber & Bishop, 1998…, Zhang et al. 2017, Sun et al. 2017, Mishkin et al., 2018

Parametrization of variational posterior covariance



- Lot of research exploring " → ":
  - E.g., Barber & Bishop, 1998…, Zhang et al. 2017, Sun et al. 2017, Mishkin et al., 2018
- We investigate the opposite trend " ← ":
  - Fewer parameters to optimize
  - Less noisy gradient estimate and convergence speed-up

How? Exploit low-rank structure!

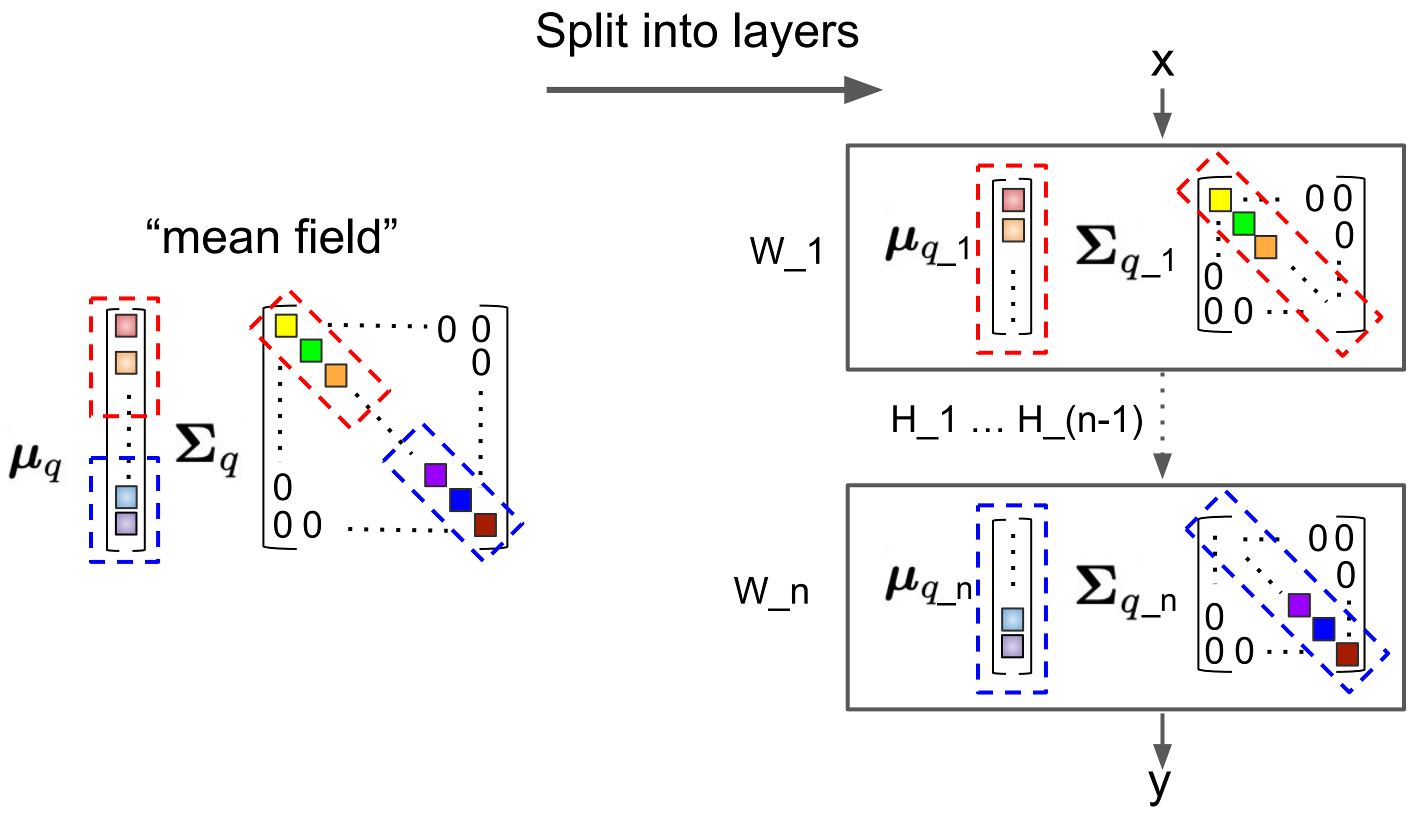
Preview

1. Gaussian Mean-Field Variational Inference (GMFVI) for Bayesian Neural Networks (BNNs).

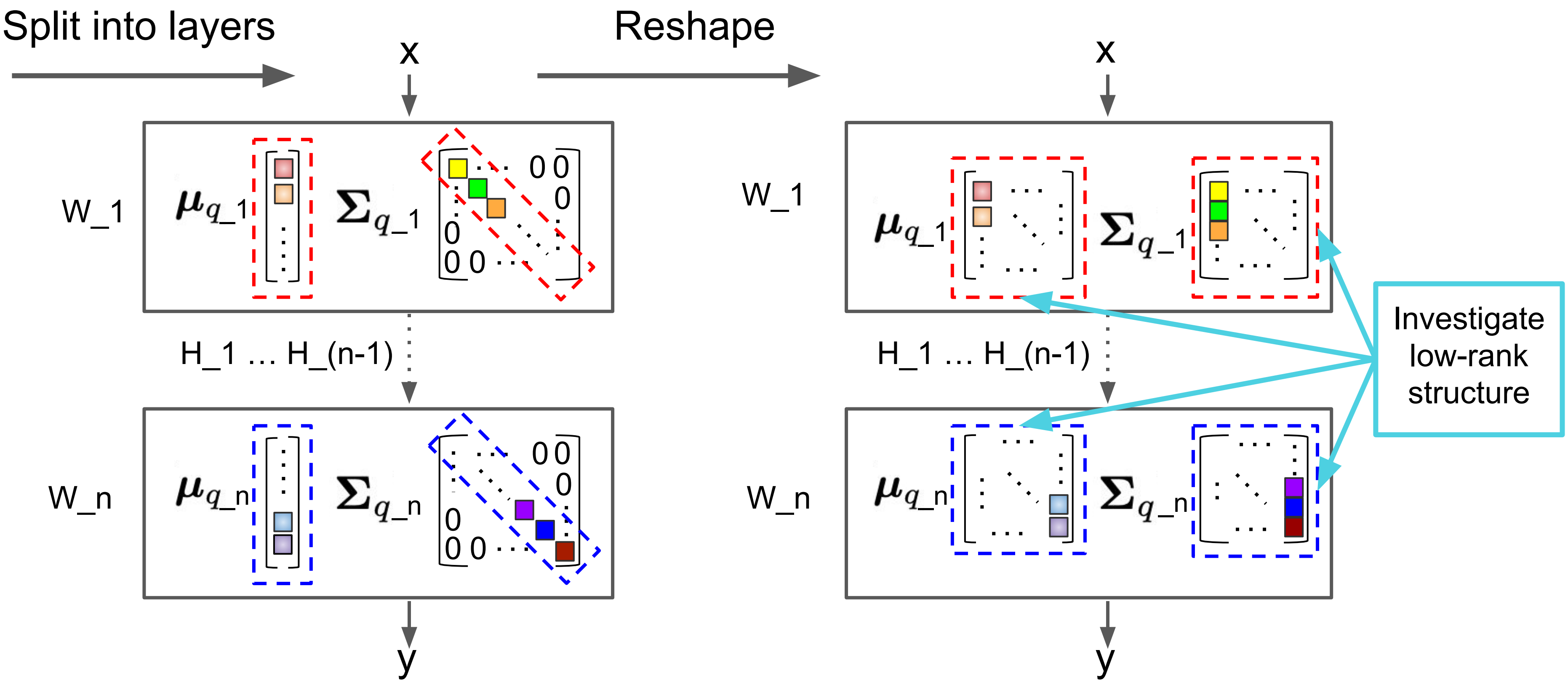
2. Low-rank in already trained GMFVI BNNs.

3. Training a low-rank parameterization of the GMFVI BNNs.

Low-rank structure in mean-field variational posteriors

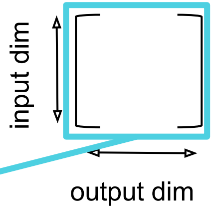


Low-rank structure in mean-field variational posteriors

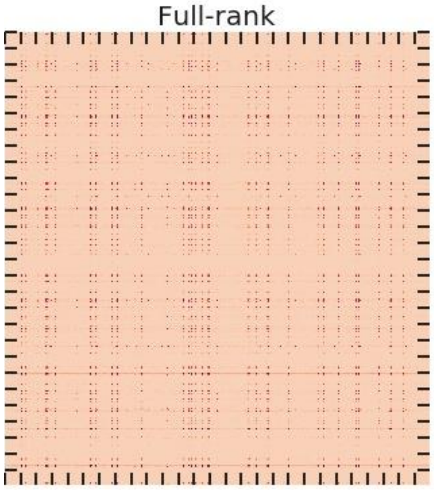


Low-rank structure in mean-field variational posteriors

Reshaped diagonal $\boldsymbol{\Sigma}_q$:

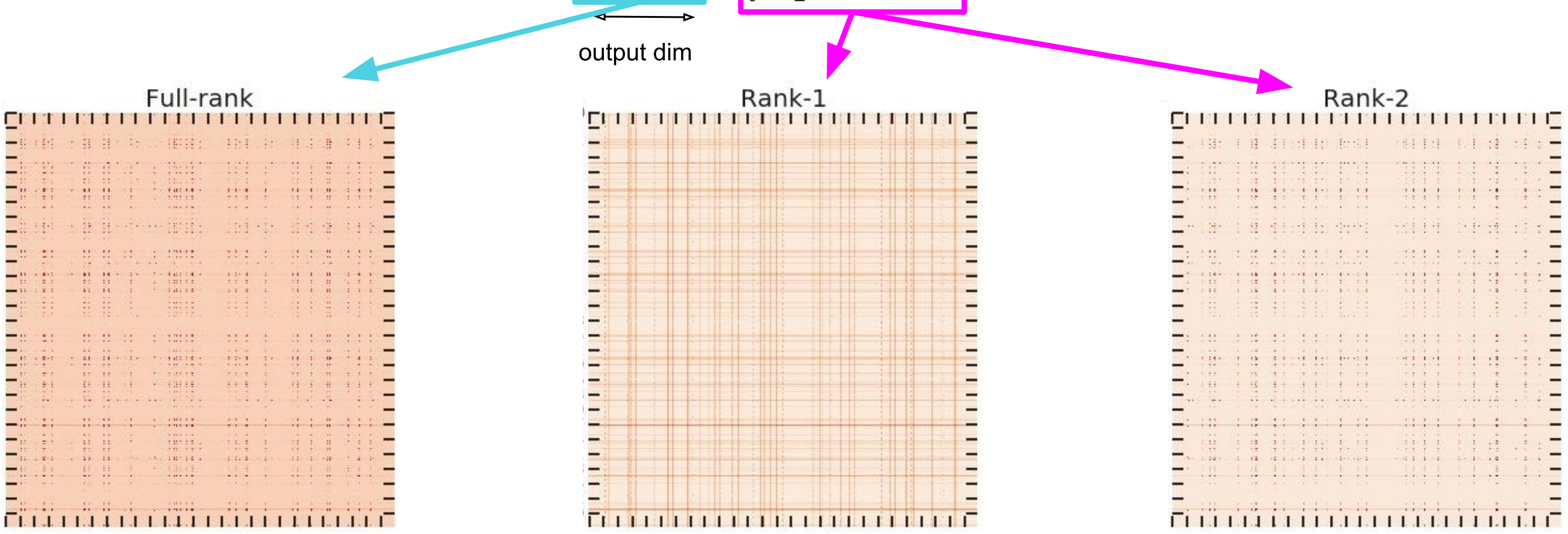


CNN, LeNet, CIFAR100, Dense 1



Low-rank structure in mean-field variational posteriors

Reshaped diagonal $\boldsymbol{\Sigma}_q$: input dim $\approx \sum_{i'=1}^{k} \gamma_{i'} \mathbf{u}_{i'} \mathbf{v}_{i'}^T$ CNN, LeNet, CIFAR100, Dense 1



Low-rank structure in mean-field variational posteriors

Fraction of explained variance of both $\boldsymbol{\mu}_q$ and diagonal $\boldsymbol{\Sigma}_q$:

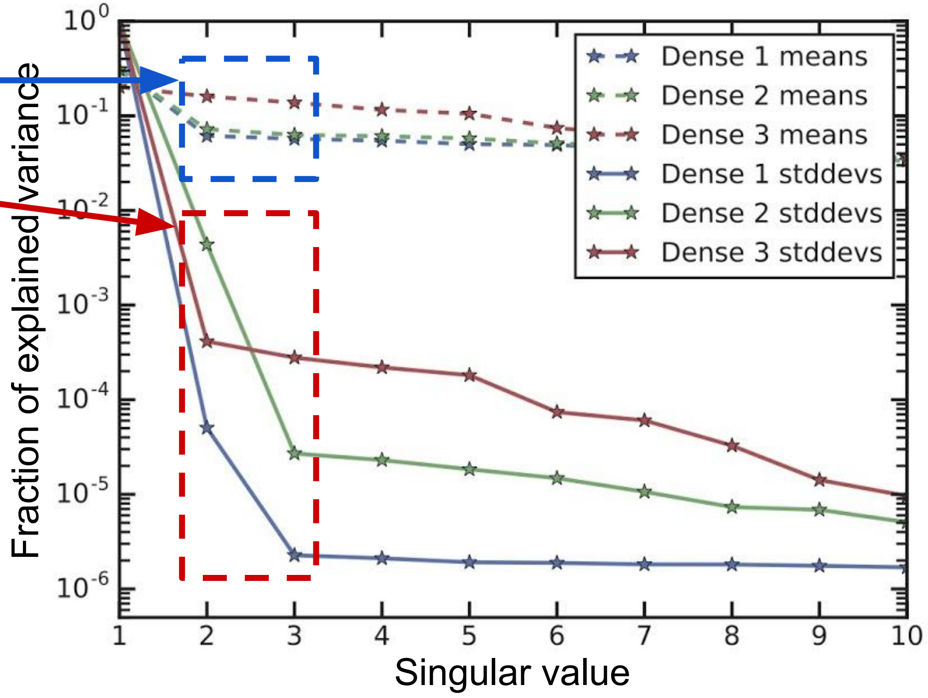
$$\gamma_k^2 / \sum_{i'} \gamma_{i'}^2$$

Low-rank structure in mean-field variational posteriors

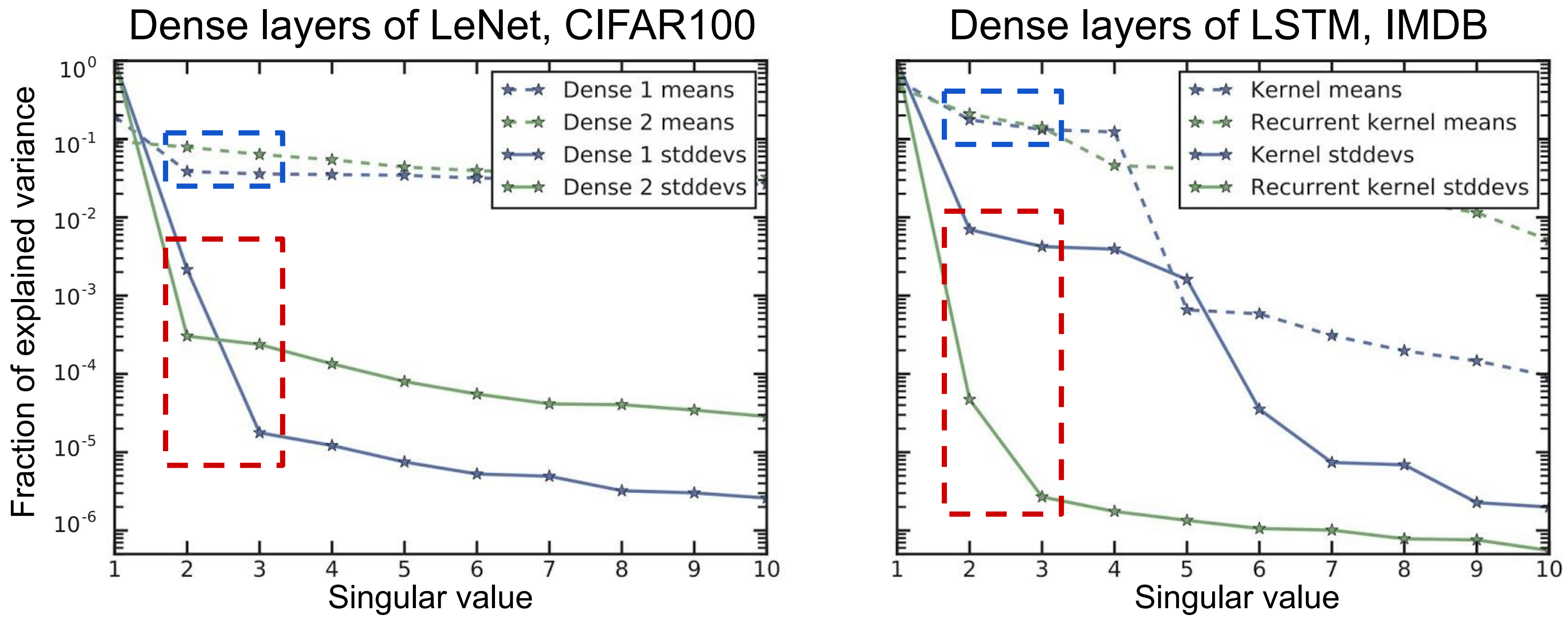
Fraction of explained variance for both $\boldsymbol{\mu}_q$ and diagonal $\boldsymbol{\Sigma}_q$:

$$\gamma_k^2 / \sum_{i'} \gamma_{i'}^2$$

Dense layers of MLP, MNIST

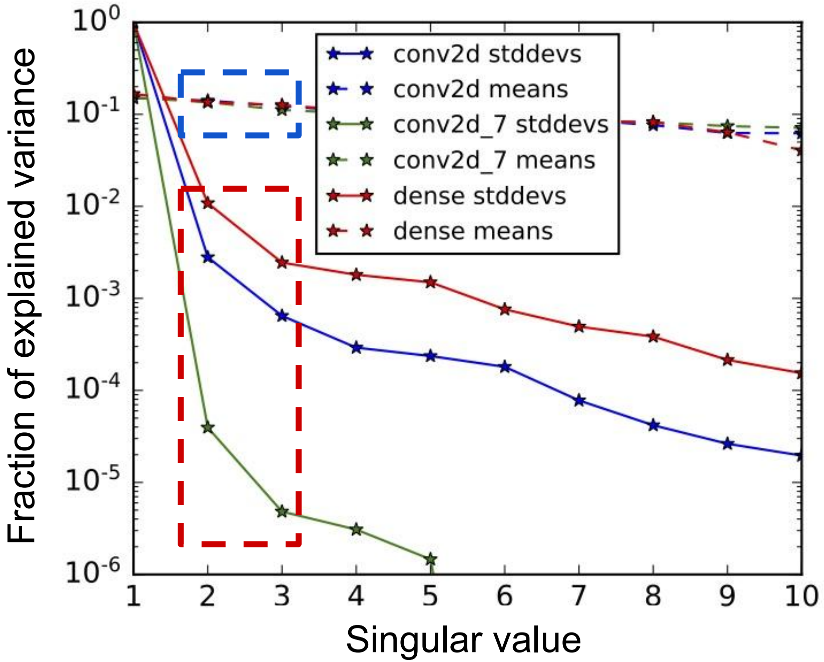


Low-rank structure in mean-field variational posteriors



Low-rank structure in mean-field variational posteriors

Dense and conv layers of ResNet-18, CIFAR10



Post-training low-rank approximation

Rank-k approximation of diagonal $\boldsymbol{\Sigma}_q$:

MLP, MNIST

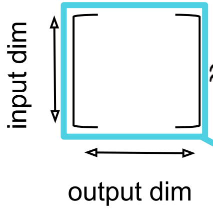
input dim

$$\begin{bmatrix} & & \\ & & \\ & & \end{bmatrix} \approx \sum_{i'=1}^{k} \gamma_{i'} \mathbf{u}_{i'} \mathbf{v}_{i'}^T$$

output dim

Post-training low-rank approximation

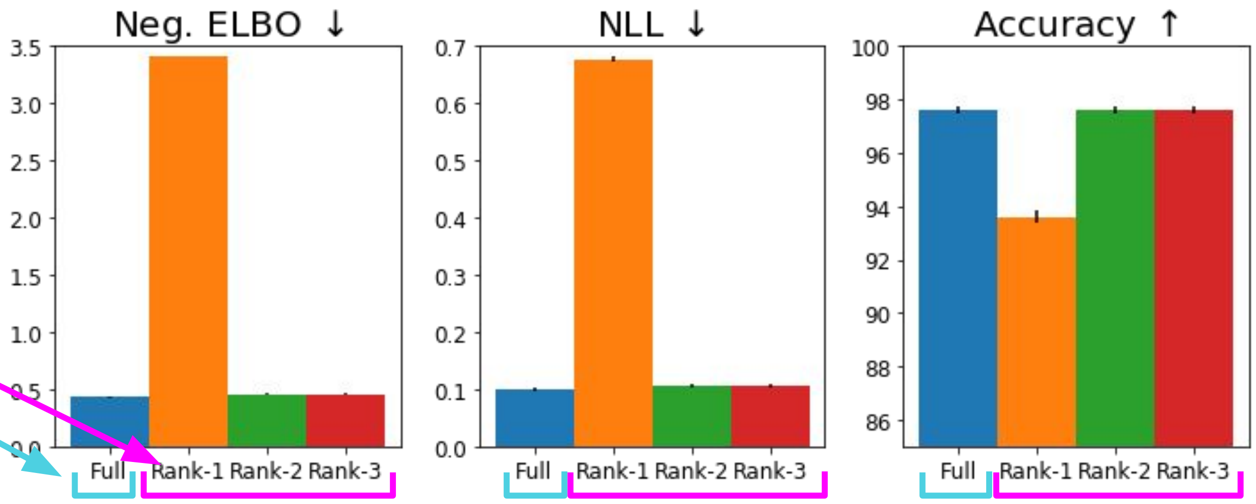
Rank-k approximation of diagonal $\boldsymbol{\Sigma}_q$:



$$\approx \sum_{i'=1}^{k} \gamma_{i'} \mathbf{u}_{i'} \mathbf{v}_{i'}^T$$

No approximation / mean-field / diagonal $\boldsymbol{\Sigma}_q$

MLP, MNIST



Post-training low-rank approximation

Dense layers of CNN (LeNet, CIFAR100) and LSTM (IMDB):

	CNN			LSTM		
Rank	-ELBO ↓	NLL ↓	Accuracy ↑	-ELBO ↓	NLL ↓	Accuracy ↑
Full	3.83$\pm$0.020	2.23$\pm$0.017	42.1$\pm$0.49	0.536$\pm$0.0058	0.493$\pm$0.0057	80.1$\pm$0.25
1	4.33$\pm$0.021	2.30$\pm$0.016	41.7$\pm$0.49	0.687$\pm$0.0058	0.491$\pm$0.0056	80.0$\pm$0.25
2	3.88$\pm$0.020	2.24$\pm$0.017	42.2$\pm$0.49	0.621$\pm$0.0058	0.494$\pm$0.0057	80.1$\pm$0.25
3	3.86$\pm$0.020	2.24$\pm$0.017	42.1$\pm$0.49	0.595$\pm$0.0058	0.493$\pm$0.0056	80.1$\pm$0.25

Dense and convolutional layers of a ResNet-18 (CIFAR10):

Rank	-ELBO ↓	NLL ↓	Accuracy ↑
Full	122.61$\pm$0.012	0.495$\pm$0.0080	83.5$\pm$0.37
1	122.57$\pm$0.012	0.658$\pm$0.0069	81.7$\pm$0.39
2	122.77$\pm$0.012	0.503$\pm$0.0080	83.2$\pm$0.37
3	122.67$\pm$0.012	0.501$\pm$0.0079	83.2$\pm$0.37

Generality of the low-rank structure finding

Low-rank structure in posterior standard deviations holds for:

- Different model types:
  - MLP
  - CNN
  - LSTM
- Different mode sizes:
  - Small 3 layer MLP
  - Large ResNet-18
- Different layer types:
  - Dense
  - Convolutional

Suggests generality of the low-rank structure finding

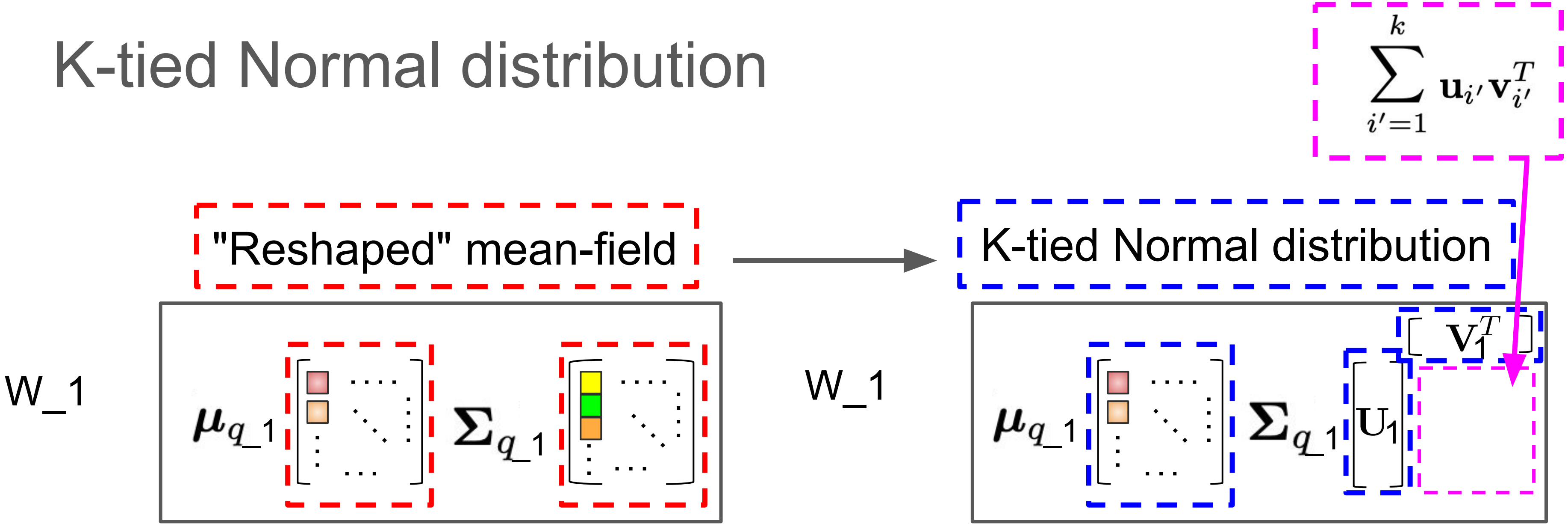
Preview

1. Gaussian Mean-Field Variational Inference (GMFVI) for Bayesian Neural Networks (BNNs).

2. Low-rank in already trained GMFVI BNNs.

3. Training a low-rank parameterization of the GMFVI BNNs.

K-tied Normal distribution

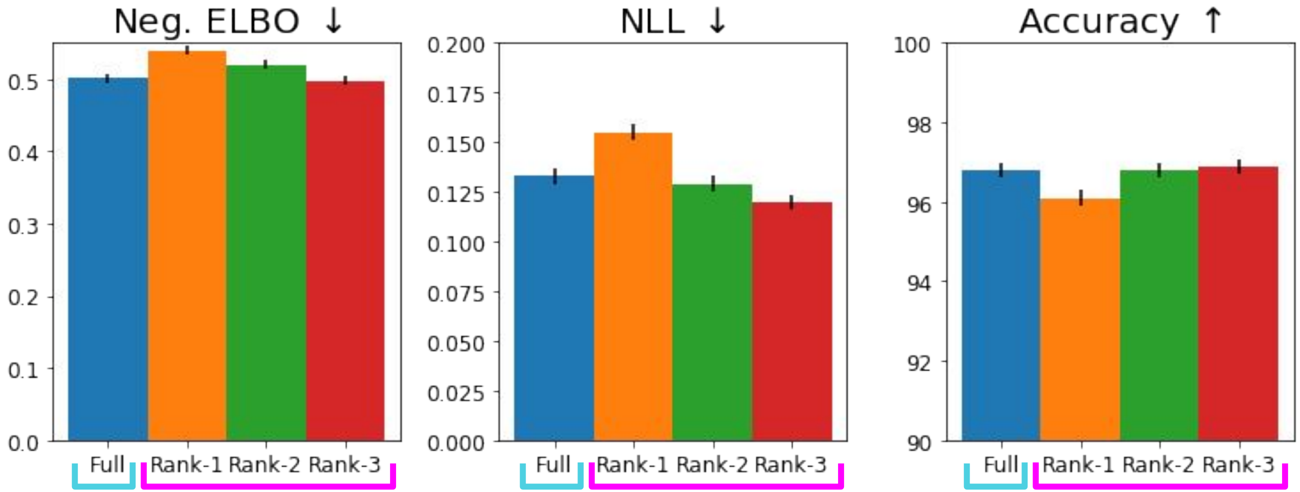


W_1 Parameters: $\boldsymbol{\mu}_{q\_1}$, $\boldsymbol{\Sigma}_{q\_1}$

W_1 Parameters: $\boldsymbol{\mu}_{q\_1}$, $\mathbf{U}_1$, $\mathbf{V}_1$

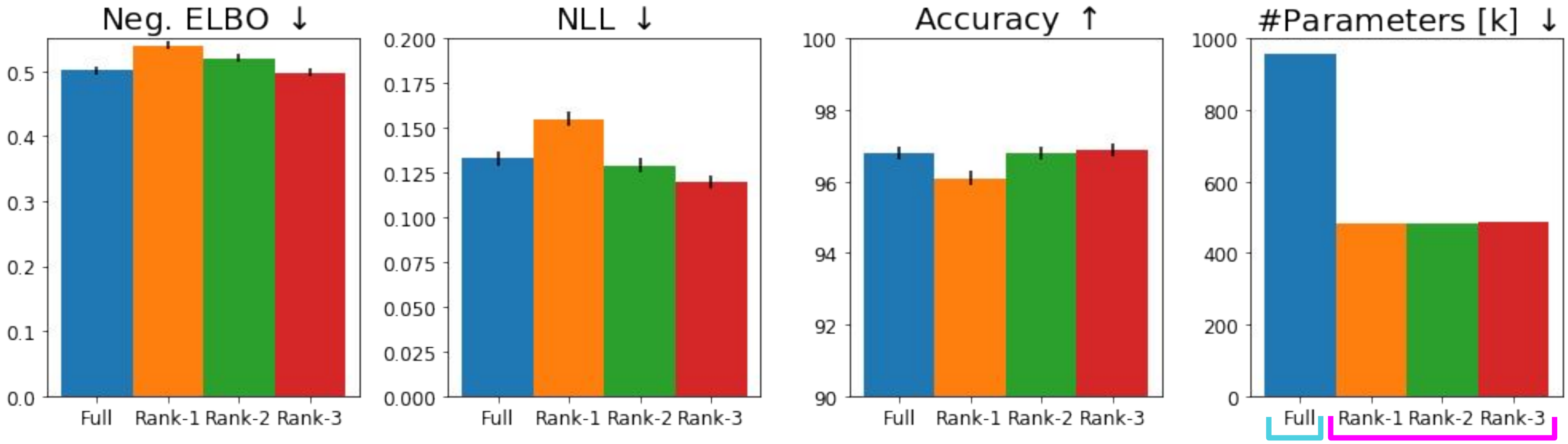
K-tied Normal distribution - training performance

MLP, MNIST



K-tied Normal distribution - training performance

MLP, MNIST



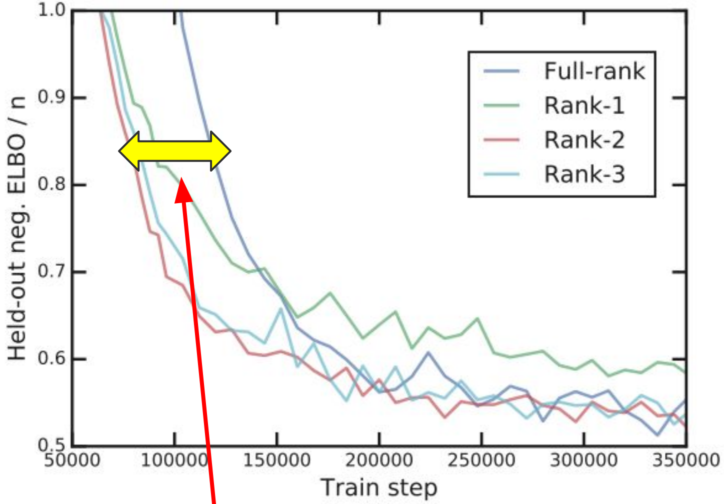
K-tied Normal increases gradient SNR

$$\mathrm{SNR} = E[g_b^2]/\mathrm{Var}[g_b]$$

Rank k	MNIST, MLP Dense 2, SNR at step 1000	5000	9000
full	4.13$\pm$0.027	4.45$\pm$0.091	3.21$\pm$0.035
1	5840$\pm$190	158$\pm$3.8	5.3$\pm$0.20
2	7500$\pm$240	140$\pm$11	4.3$\pm$0.26
3	7000$\pm$270	117$\pm$1.7	4.1$\pm$0.20

K-tied Normal speeds up convergence

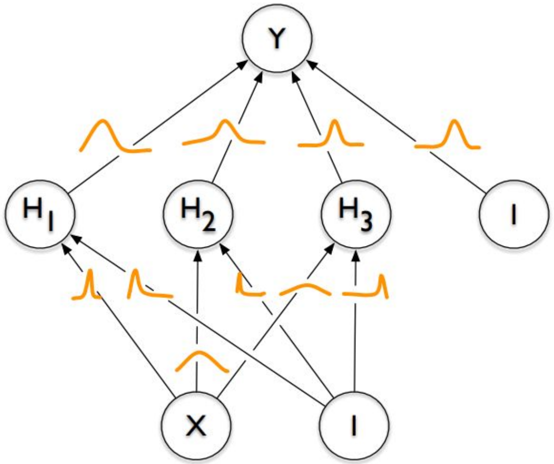
MLP (MNIST)



Increased convergence speed

Rank k	MNIST, MLP, -ELBO at step 1000	5000	9000
full	$42.16_{\pm 0.070}$	$26.52_{\pm 0.016}$	$15.39_{\pm 0.016}$
1	$43.11_{\pm 0.039}$	$14.85_{\pm 0.017}$	$2.06_{\pm 0.027}$
2	$42.74_{\pm 0.090}$	$13.97_{\pm 0.023}$	$1.82_{\pm 0.017}$
3	$42.63_{\pm 0.068}$	$13.61_{\pm 0.020}$	$1.80_{\pm 0.031}$

Let's bring the benefits of Bayesian inference to neural networks



Sources: Blundell et al. "Weight uncertainty in neural networks.", Wikipedia and [Shutterstock]

Thank you!

Review

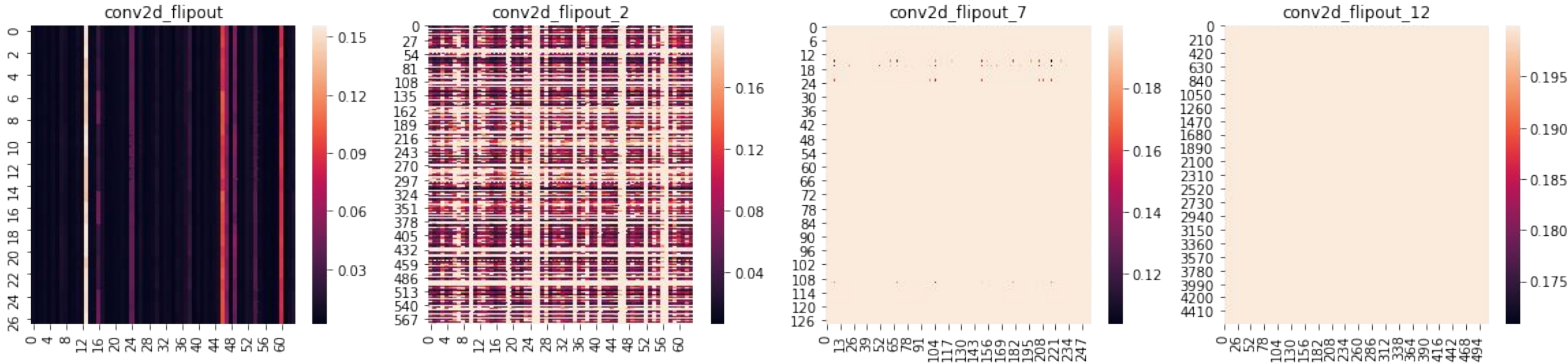
1. Gaussian Mean-Field Variational Inference (GMFVI) for Bayesian Neural Networks (BNNs).

2. Low-rank in already trained GMFVI BNNs.

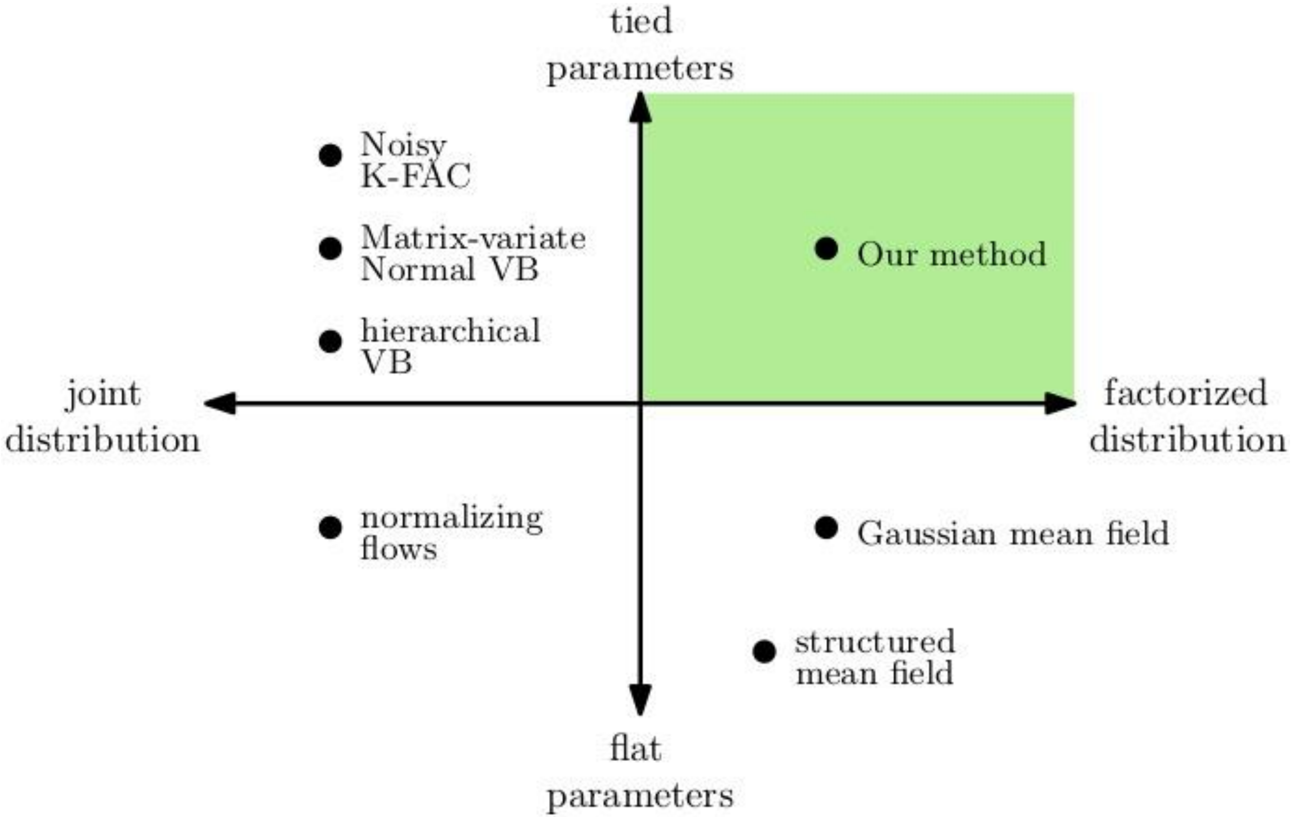
3. Training a low-rank parameterization of the GMFVI BNNs.

Heatmaps of conv posterior stddevs

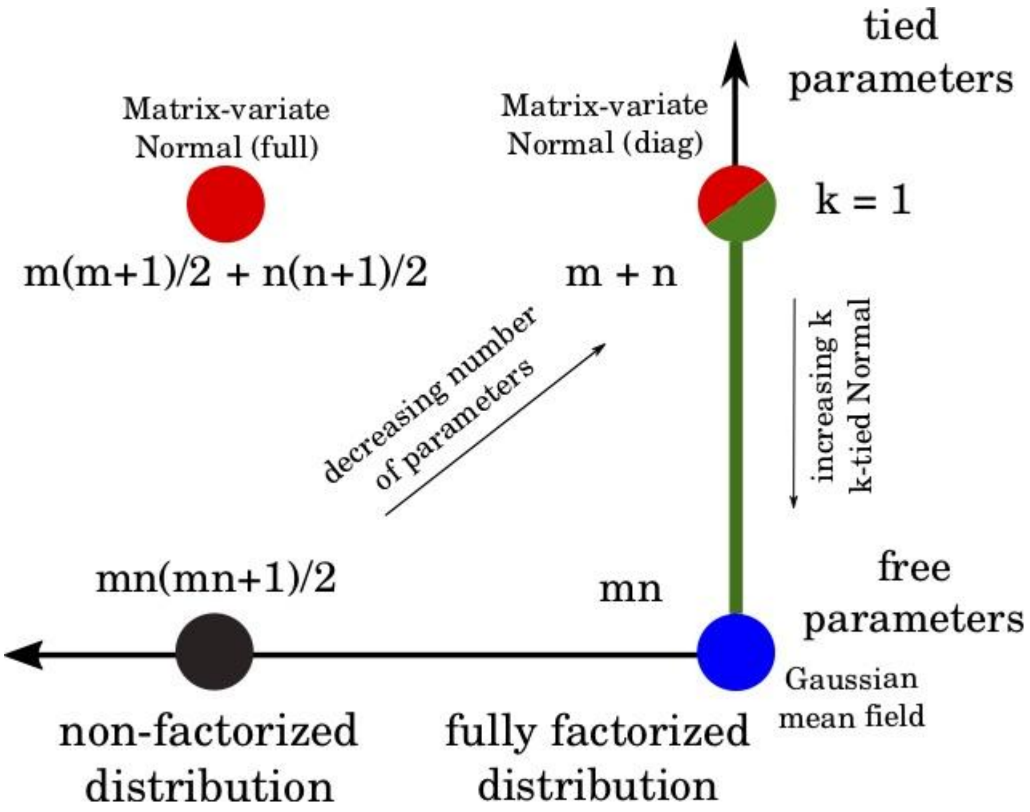
Reshaping conv layers e.g.: [3, 3, 10, 20] -> [3 * 3 * 10, 20].



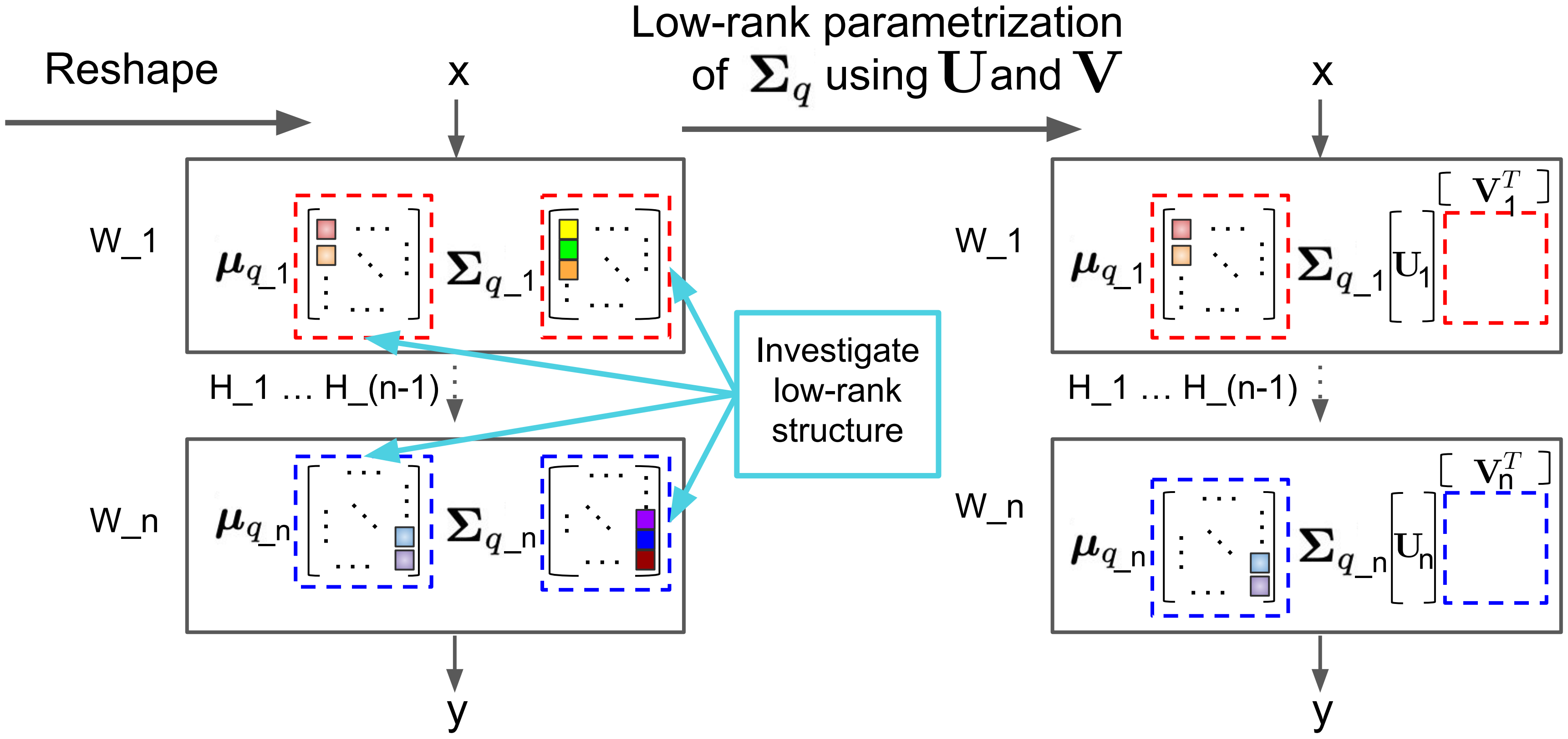
Relation to other work



Relation to Matrix-Variate Normal

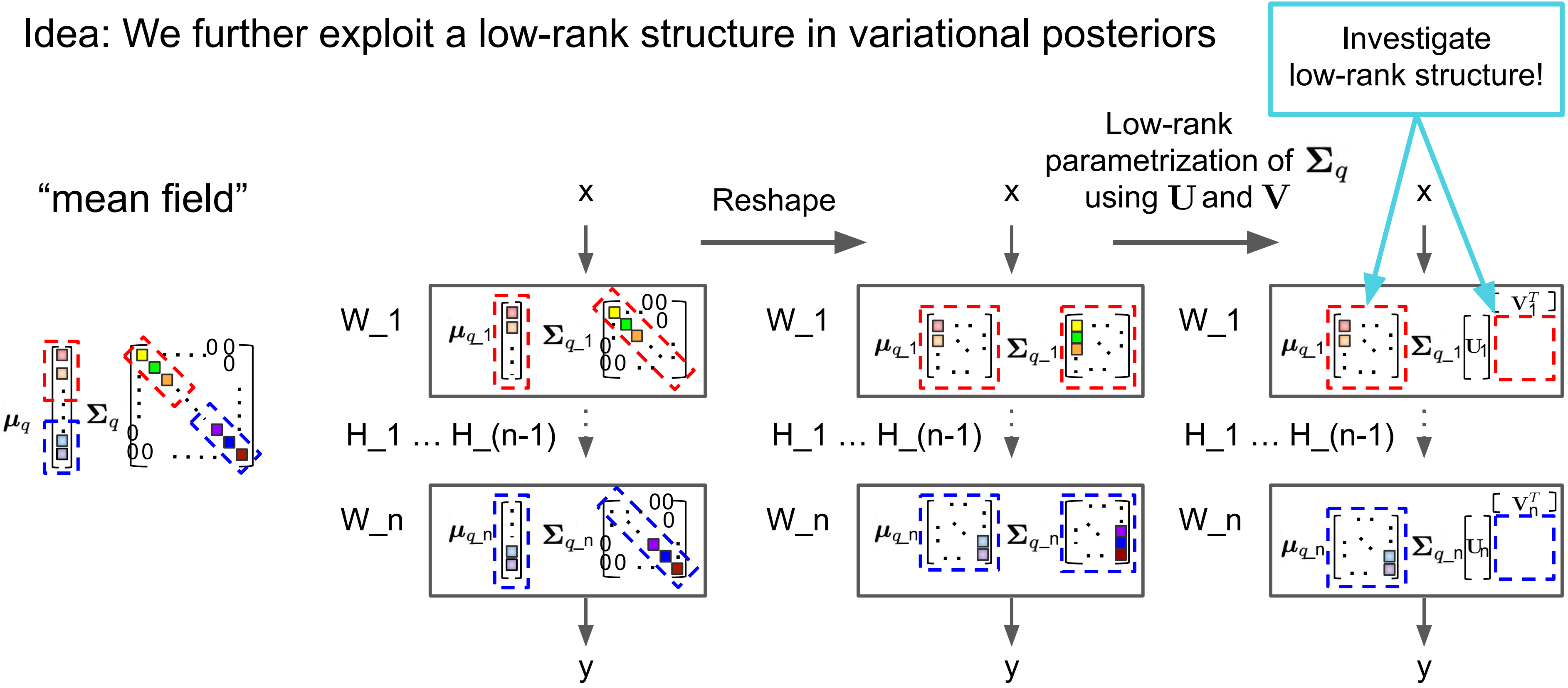


Low-rank structure in mean-field variational posteriors



Low-rank structure in mean-field variational posteriors

Idea: We further exploit a low-rank structure in variational posteriors



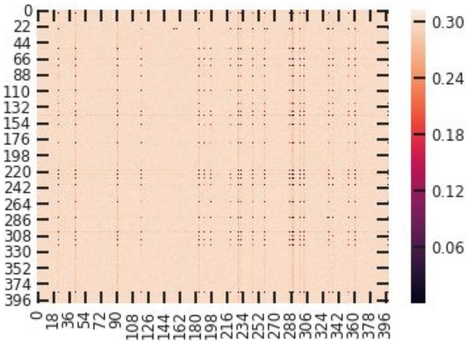
We investigate per layer GMFVI posterior matrices

$$\mathbf{a}_l = \mathbf{h}_l \mathbf{W}_l + \mathbf{b}_l, \qquad \mathbf{h}_{l+1} = f(\mathbf{a}_l), \qquad \mathbf{W}_l \in \mathbb{R}^{m \times n}$$

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

$$\boldsymbol{\mu}_q = \mathrm{vec}\big(\mathbf{M}\big), \quad \mathbf{M} \in \mathbb{R}^{m \times n}$$

$$\boldsymbol{\Sigma}_q = \mathrm{diag}(\mathrm{vec}(\mathbf{A}^2)) \quad \mathbf{A} \in \mathbb{R}_{+}^{m \times n} \longrightarrow$$



We prose a k-tied Normal variational posterior that exploits the low-rank structure

$$q(\mathbf{W}) = \mathcal{N}(\boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q)$$

$$\boldsymbol{\Sigma}_q \;=\; \mathrm{diag}(\mathrm{vec}(\mathbf{A}^2)) \qquad \mathbf{A} \in \mathbb{R}_+^{m \times n}$$

$$\mathbf{A} \approx \mathbf{U}\mathbf{V}^T$$

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

K-tied Normal posterior reduces the number of parameters without reducing performance

Model & Dataset	Rank k	-ELBO ↓	NLL ↓	Accuracy ↑	#Par. [k] ↓
MNIST, MLP	full	$0.501_{\pm 0.0061}$	$0.133_{\pm 0.0040}$	$96.8_{\pm 0.18}$	957
MNIST, MLP	1	$0.539_{\pm 0.0063}$	$0.155_{\pm 0.0043}$	$96.1_{\pm 0.19}$	482
MNIST, MLP	2	$0.520_{\pm 0.0063}$	$0.129_{\pm 0.0039}$	$96.8_{\pm 0.18}$	484
MNIST, MLP	3	$0.497_{\pm 0.0060}$	$0.120_{\pm 0.0038}$	$96.9_{\pm 0.18}$	486
CIFAR100, CNN	full	$3.72_{\pm 0.018}$	$2.16_{\pm 0.016}$	$43.9_{\pm 0.50}$	4,405
CIFAR100, CNN	1	$3.65_{\pm 0.017}$	$2.12_{\pm 0.015}$	$45.5_{\pm 0.50}$	2,262
CIFAR100, CNN	2	$3.76_{\pm 0.019}$	$2.15_{\pm 0.016}$	$44.3_{\pm 0.50}$	2,268
CIFAR100, CNN	3	$3.73_{\pm 0.018}$	$2.13_{\pm 0.016}$	$44.3_{\pm 0.50}$	2,273
IMDB, LSTM	full	$0.538_{\pm 0.0054}$	$0.478_{\pm 0.0052}$	$79.5_{\pm 0.26}$	2,823
IMDB, LSTM	1	$0.592_{\pm 0.0041}$	$0.512_{\pm 0.0040}$	$77.6_{\pm 0.26}$	2,693
IMDB, LSTM	2	$0.560_{\pm 0.0042}$	$0.484_{\pm 0.0041}$	$78.2_{\pm 0.26}$	2,694
IMDB, LSTM	3	$0.550_{\pm 0.0051}$	$0.491_{\pm 0.0050}$	$78.8_{\pm 0.26}$	2,695

K-tied Normal speeds up convergence