

# VIKING: Deep variational inference with stochastic projections

Samuel G. Fadel, Hrittik Roy, Nicholas Krämer, Yevgen Zainchkovskyy, Stas Syrota, Alejandro Valverde Mahou, Carl Henrik Ek, Søren Hauberg



Danish  
Data Science  
Academy



# The issue with Laplace approximations

## **Laplace Approximation** (MacKay 1995)

- Place  $\mathcal{N}(\theta|\hat{\theta}, \Sigma)$  around some learned  $\hat{\theta}$

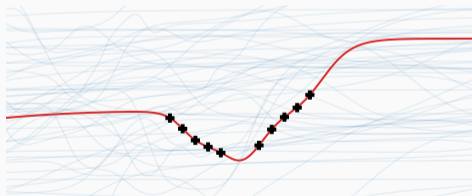
# The issue with Laplace approximations

## Laplace Approximation (MacKay 1995)

- Place  $\mathcal{N}(\theta|\hat{\theta}, \Sigma)$  around some learned  $\hat{\theta}$

— (pred.) Posterior mean

— (pred.) Posterior samples



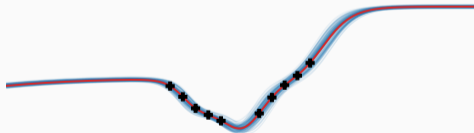
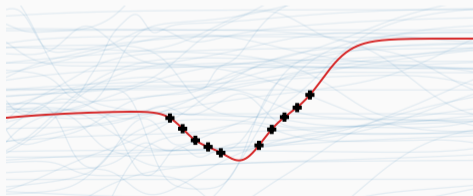
# The issue with Laplace approximations

## Laplace Approximation (MacKay 1995)

- Place  $\mathcal{N}(\theta|\hat{\theta}, \Sigma)$  around some learned  $\hat{\theta}$

— (pred.) Posterior mean

— (pred.) Posterior samples



# Reparameterisations

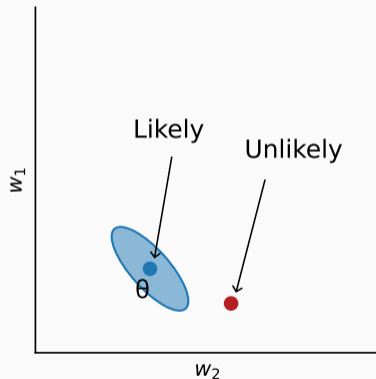
Consider

$$f(x) = w_1 \text{ReLU}(w_2 x)$$

Now reparameterise  $f$  with any  $\gamma \neq 0$  as

$$f(x) = \frac{w_1}{\gamma} \text{ReLU}(\gamma w_2 x)$$

Both  $\theta = (w_1, w_2)$  and  $(\frac{w_1}{\gamma}, \gamma w_2)$  yield identical functions!



# Reparameterisations

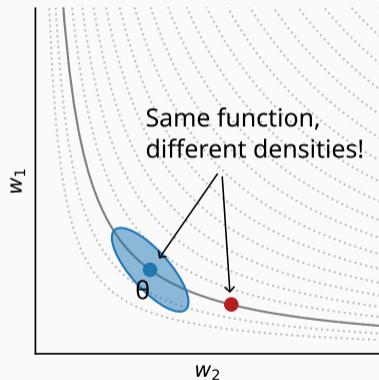
Consider

$$f(x) = w_1 \text{ReLU}(w_2 x)$$

Now reparameterise  $f$  with any  $\gamma \neq 0$  as

$$f(x) = \frac{w_1}{\gamma} \text{ReLU}(\gamma w_2 x)$$

Both  $\theta = (w_1, w_2)$  and  $(\frac{w_1}{\gamma}, \gamma w_2)$  yield **identical functions!**



# Empirical estimates of reparameterisations

The kernel (null space) of the generalised Gauss-Newton (GGN) matrix

$$\mathbf{J}_{\hat{\theta}}^{\ell \top} \mathbf{H}_{\hat{\theta}}^{\ell} \mathbf{J}_{\hat{\theta}}^{\ell} \in \mathbb{R}^{D \times D}$$

contains parameters  $\theta$  where changes to the loss are minimal<sup>1</sup>.

- $\mathbf{J}_{\hat{\theta}}^{\ell}$  is a Jacobian matrix of stacked loss gradients per data point

$$\mathbf{J}_{\hat{\theta}}^{\ell} = \begin{bmatrix} \nabla_{\hat{\theta}} \ell(\hat{\theta}, \mathbf{x}_1, \mathbf{y}_1)^{\top} \\ \vdots \\ \nabla_{\hat{\theta}} \ell(\hat{\theta}, \mathbf{x}_N, \mathbf{y}_N)^{\top} \end{bmatrix}$$

---

<sup>1</sup>Miani et al. (2025, Lemma 4.3)

# VIKING

VI with kernel-image numerical Gauss-Newton

Let  $\mathbf{U}_\theta \mathbf{U}_\theta^\top$  be the projection onto the kernel of the (loss) GGN.

$$q(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta} | \hat{\boldsymbol{\theta}}, \boldsymbol{\Sigma}_{\hat{\boldsymbol{\theta}}}),$$

$$\boldsymbol{\Sigma}_\theta = \sigma_{\text{ker}}^2 \mathbf{U}_\theta \mathbf{U}_\theta^\top + \sigma_{\text{im}}^2 (\mathbb{I} - \mathbf{U}_\theta \mathbf{U}_\theta^\top)$$

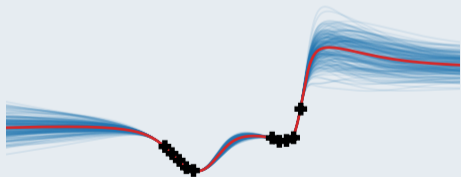
# VIKING

VI with kernel-image numerical Gauss-Newton

Let  $\mathbf{U}_\theta \mathbf{U}_\theta^\top$  be the projection onto the kernel of the (loss) GGN.

$$q(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta} | \hat{\boldsymbol{\theta}}, \boldsymbol{\Sigma}_{\hat{\boldsymbol{\theta}}}),$$

$$\boldsymbol{\Sigma}_\theta = \sigma_{\text{ker}}^2 \mathbf{U}_\theta \mathbf{U}_\theta^\top + \sigma_{\text{im}}^2 (\mathbb{I} - \mathbf{U}_\theta \mathbf{U}_\theta^\top)$$



Controls uncertainty outside training data

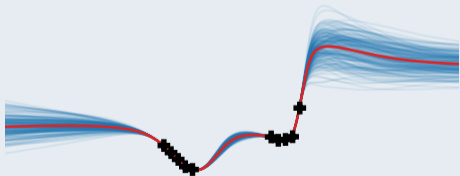
# VIKING

VI with kernel-image numerical Gauss-Newton

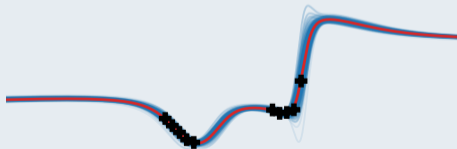
Let  $\mathbf{U}_\theta \mathbf{U}_\theta^\top$  be the projection onto the kernel of the (loss) GGN.

$$q(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta} | \hat{\boldsymbol{\theta}}, \boldsymbol{\Sigma}_{\hat{\boldsymbol{\theta}}}),$$

$$\boldsymbol{\Sigma}_\theta = \sigma_{\text{ker}}^2 \mathbf{U}_\theta \mathbf{U}_\theta^\top + \sigma_{\text{im}}^2 (\mathbb{I} - \mathbf{U}_\theta \mathbf{U}_\theta^\top)$$



Controls uncertainty outside training data



Controls uncertainty everywhere

# Kernel projections

Obtaining  $\epsilon_{\text{ker}}$  is tricky, since we need

$$\epsilon_{\text{ker}} = \mathbf{U}_{\theta} \mathbf{U}_{\theta}^{\top} \epsilon = \arg \min_{\mathbf{u}} \{ \|\mathbf{u} - \epsilon\|^2 \text{ subject to } \mathbf{J}_{\theta} \mathbf{u} = \mathbf{0} \}$$

with

- Matrix-free linear algebra for the least squares problem
- A stochastic twist on alternating projections, allowing changes to model parameters over minibatches  $t$

$$\epsilon^{(t+1)} = \mathbf{U}_{\theta}^{(t)} (\mathbf{U}_{\theta}^{(t)})^{\top} \left( \sqrt{\gamma} \epsilon^{(t)} + \sqrt{1 - \gamma} \boldsymbol{\eta}^{(t)} \right), \quad \boldsymbol{\eta}^{(t)} \sim \mathcal{N}(0, \mathbb{I}).$$

# How does it perform?

		Accuracy <sup>↑</sup>	NLL <sup>↓</sup>	ECE <sup>↓</sup>	MCE <sup>↓</sup>
MNIST	MAP	0.986 ± 0.001	0.070 ± 0.005	0.247 ± 0.011	0.861 ± 0.045
	VIKING (ours)	<b>0.991 ± 0.001</b>	0.055 ± 0.003	0.096 ± 0.004	0.690 ± 0.102
	Miani et al. (2025)	0.949 ± 0.000	1.225 ± 0.099	0.666 ± 0.007	0.894 ± 0.011
	IVON	0.989 ± 0.001	<b>0.043 ± 0.002</b>	<b>0.077 ± 0.005</b>	<b>0.651 ± 0.042</b>
	SWAG	0.982 ± 0.000	0.064 ± 0.006	0.788 ± 0.005	0.906 ± 0.013
	Last Layer LA	0.975 ± 0.002	0.090 ± 0.005	0.784 ± 0.007	0.887 ± 0.008
Fashion MNIST	MAP	0.883 ± 0.002	0.410 ± 0.010	0.153 ± 0.008	<b>0.590 ± 0.141</b>
	VIKING (ours)	<b>0.900 ± 0.001</b>	0.332 ± 0.003	0.075 ± 0.002	0.611 ± 0.160
	Miani et al. (2025)	0.871 ± 0.006	1.529 ± 0.371	0.617 ± 0.025	0.901 ± 0.013
	IVON	0.897 ± 0.004	0.335 ± 0.011	<b>0.073 ± 0.005</b>	0.683 ± 0.024
	SWAG	0.898 ± 0.001	<b>0.327 ± 0.001</b>	0.725 ± 0.003	0.907 ± 0.003
	Last Layer LA	0.896 ± 0.002	0.339 ± 0.011	0.727 ± 0.004	0.902 ± 0.004
SVHN	MAP	0.947 ± 0.004	0.201 ± 0.014	0.055 ± 0.010	0.608 ± 0.228
	VIKING (ours)	<b>0.960 ± 0.001</b>	<b>0.177 ± 0.002</b>	<b>0.028 ± 0.002</b>	<b>0.308 ± 0.024</b>
	Miani et al. (2025)	0.949 ± 0.003	0.191 ± 0.007	0.734 ± 0.017	0.880 ± 0.012
	IVON	0.943 ± 0.002	0.302 ± 0.016	0.082 ± 0.004	0.492 ± 0.248
	SWAG	0.947 ± 0.004	0.217 ± 0.014	0.745 ± 0.007	0.874 ± 0.003
	Last Layer LA	0.946 ± 0.001	0.197 ± 0.009	0.740 ± 0.007	0.899 ± 0.009
CIFAR-10	MAP	0.824 ± 0.012	0.536 ± 0.055	0.075 ± 0.012	0.619 ± 0.243
	VIKING (ours)	0.877 ± 0.004	0.407 ± 0.010	<b>0.041 ± 0.004</b>	<b>0.331 ± 0.094</b>
	Miani et al. (2025)	0.855 ± 0.002	2.643 ± 0.205	0.559 ± 0.006	0.802 ± 0.005
	IVON	0.835 ± 0.017	0.817 ± 0.075	0.086 ± 0.014	0.436 ± 0.244
	SWAG	0.865 ± 0.029	0.445 ± 0.063	0.694 ± 0.018	0.881 ± 0.005
	Last Layer LA	<b>0.894 ± 0.001</b>	<b>0.406 ± 0.005</b>	0.704 ± 0.000	0.880 ± 0.007

- Not SotA in all cases
- Consistency is key

# How does it perform?

		Accuracy <sup>↑</sup>	NLL <sup>↓</sup>	ECE <sup>↓</sup>	MCE <sup>↓</sup>
MNIST	MAP	0.986 ± 0.001	0.070 ± 0.005	0.247 ± 0.011	0.861 ± 0.045
	VIKING (ours)	<b>0.991 ± 0.001</b>	0.055 ± 0.003	0.096 ± 0.004	0.690 ± 0.102
	Miani et al. (2025)	0.949 ± 0.000	1.225 ± 0.099	0.666 ± 0.007	0.894 ± 0.011
	IVON	0.989 ± 0.001	<b>0.043 ± 0.002</b>	<b>0.077 ± 0.005</b>	<b>0.651 ± 0.042</b>
	SWAG	0.982 ± 0.000	0.064 ± 0.006	0.788 ± 0.005	0.906 ± 0.013
	Last Layer LA	0.975 ± 0.002	0.090 ± 0.005	0.784 ± 0.007	0.887 ± 0.008
Fashion MNIST	MAP	0.883 ± 0.002	0.410 ± 0.010	0.153 ± 0.008	<b>0.590 ± 0.141</b>
	VIKING (ours)	<b>0.900 ± 0.001</b>	0.332 ± 0.003	0.075 ± 0.002	0.611 ± 0.160
	Miani et al. (2025)	0.871 ± 0.006	1.529 ± 0.371	0.617 ± 0.025	0.901 ± 0.013
	IVON	0.897 ± 0.004	0.335 ± 0.011	<b>0.073 ± 0.005</b>	0.683 ± 0.024
	SWAG	0.898 ± 0.001	<b>0.327 ± 0.001</b>	0.725 ± 0.003	0.907 ± 0.003
	Last Layer LA	0.896 ± 0.002	0.339 ± 0.011	0.727 ± 0.004	0.902 ± 0.004
SVHN	MAP	0.947 ± 0.004	0.201 ± 0.014	0.055 ± 0.010	0.608 ± 0.228
	VIKING (ours)	<b>0.960 ± 0.001</b>	<b>0.177 ± 0.002</b>	<b>0.028 ± 0.002</b>	<b>0.308 ± 0.024</b>
	Miani et al. (2025)	0.949 ± 0.003	0.191 ± 0.007	0.734 ± 0.017	0.880 ± 0.012
	IVON	0.943 ± 0.002	0.302 ± 0.016	0.082 ± 0.004	0.492 ± 0.248
	SWAG	0.947 ± 0.004	0.217 ± 0.014	0.745 ± 0.007	0.874 ± 0.003
	Last Layer LA	0.946 ± 0.001	0.197 ± 0.009	0.740 ± 0.007	0.899 ± 0.009
CIFAR-10	MAP	0.824 ± 0.012	0.536 ± 0.055	0.075 ± 0.012	0.619 ± 0.243
	VIKING (ours)	0.877 ± 0.004	0.407 ± 0.010	<b>0.041 ± 0.004</b>	<b>0.331 ± 0.094</b>
	Miani et al. (2025)	0.855 ± 0.002	2.643 ± 0.205	0.559 ± 0.006	0.802 ± 0.005
	IVON	0.835 ± 0.017	0.817 ± 0.075	0.086 ± 0.014	0.436 ± 0.244
	SWAG	0.865 ± 0.029	0.445 ± 0.063	0.694 ± 0.018	0.881 ± 0.005
	Last Layer LA	<b>0.894 ± 0.001</b>	<b>0.406 ± 0.005</b>	0.704 ± 0.000	0.880 ± 0.007

- Not SotA in all cases
- Consistency is key

# It scales to larger models!

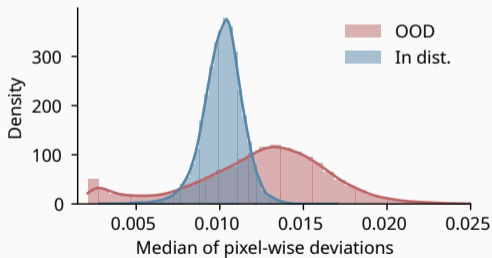
	Accuracy <sup>↑</sup>	NLL <sup>↓</sup>	ECE <sup>↓</sup>	MCE <sup>↓</sup>
MAP	0.852 ± 0.002	0.481 ± 0.009	0.084 ± 0.010	0.717 ± 0.082
VIKING (ours)	<b>0.887</b> ± <b>0.003</b>	<b>0.403</b> ± <b>0.010</b>	0.077 ± 0.001	0.612 ± 0.162
IVON	0.876 ± 0.023	0.656 ± 0.136	<b>0.069</b> ± <b>0.011</b>	<b>0.464</b> ± <b>0.230</b>

ResNet34 (21.7M params.) trained on Imagenette

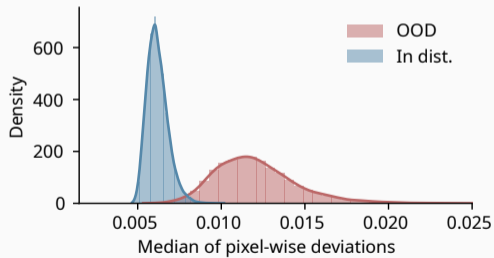
# OOD detection and generative modelling

Evaluated on  $\{\theta^{(s)} | \theta^{(s)} \sim q(\theta)\}_{s=1}^{32}$  from VAEs trained on CelebA

## IVON (Shen et al. 2024)



## VIKING (ours)



Thank you for your attention!



Paper



Code

# References I

- MacKay, David JC (1995).** “Probable networks and plausible predictions-a review of practical Bayesian methods for supervised neural networks”. In: *Network: computation in neural systems* 6.3, p. 469.
- Miani, Marco, Hritik Roy, and Søren Hauberg (2025).** “Bayes without Underfitting: Fully Correlated Deep Learning Posteriors via Alternating Projections”. In: *The 28th International Conference on Artificial Intelligence and Statistics*.

## References II

Shen, Yuesong, Nico Daheim, Bai Cong, Peter Nickl, Gian Maria Marconi, Clement Bazan, Rio Yokota, Iryna Gurevych, Daniel Cremers, Mohammad Emtiyaz Khan, et al. (2024). “Variational learning is effective for large deep networks”. In: *arXiv preprint arXiv:2402.17641*.