Reparameterization invariance in approximate Bayesian inference

Hrittik Roy[†], **Marco Miani**[†]
Technical University of Denmark
{hroy, mmia}@dtu.dk

Carl Henrik Ek
University of Cambridge,
Karolinska Institutet
che29@cam.ac.uk

Philipp Hennig, Marvin Pförtner, Lukas Tatzel
University of Tübingen, Tübingen AI Center
{philipp.hennig, lukas.tatzel,
marvin.pfoertner}@uni-tuebingen.de

Søren Hauberg
Technical University of Denmark
sohau@dtu.dk

Why linearised Laplace > regular Laplace

October 2024

Section 1: Laplace Approximation in BNNs

- Neural network $f_{\mathbf{w}} : \mathbb{R}^I \to \mathbb{R}^O$ with likelihood $p(\mathbf{y}|f_{\mathbf{w}}(\mathbf{x}))$ and prior $p(\mathbf{w})$

Section 1: Laplace Approximation in BNNs

- Neural network $f_{\mathbf{w}} : \mathbb{R}^I \to \mathbb{R}^O$ with likelihood $p(\mathbf{y}|f_{\mathbf{w}}(\mathbf{x}))$ and prior $p(\mathbf{w})$

- Take second-order Taylor expansion of log-posterior around a mode $\hat{\mathbf{w}}$

$$\log p(\mathbf{x}, \mathbf{y}; \mathbf{w}) \approx \log p(\mathbf{x}, \mathbf{y}; \hat{\mathbf{w}}) - \frac{1}{2}(\mathbf{w} - \hat{\mathbf{w}})^T(-\nabla^2_{\mathbf{w}} \log p(\mathbf{x}, \mathbf{y}; \mathbf{w})|_{\hat{\mathbf{w}}})(\mathbf{w} - \hat{\mathbf{w}})$$

Section 1: Laplace Approximation in BNNs

- Neural network $f_{\mathbf{w}} : \mathbb{R}^I \rightarrow \mathbb{R}^O$ with likelihood $p(\mathbf{y}|f_{\mathbf{w}}(\mathbf{x}))$ and prior $p(\mathbf{w})$

- Take second-order Taylor expansion of log-posterior around a mode $\hat{\mathbf{w}}$

$$\log p(\mathbf{x}, \mathbf{y}; \mathbf{w}) \approx \log p(\mathbf{x}, \mathbf{y}; \hat{\mathbf{w}}) - \frac{1}{2}(\mathbf{w} - \hat{\mathbf{w}})^T (-\nabla^2_{\mathbf{w}} \log p(\mathbf{x}, \mathbf{y}; \mathbf{w})|_{\hat{\mathbf{w}}})(\mathbf{w} - \hat{\mathbf{w}})$$

- Approximate posterior as $p(\mathbf{w}|\mathbf{x}, \mathbf{y}) \approx \mathcal{N}(\mathbf{w}|\hat{\mathbf{w}}, -\mathbf{H}^{-1}_{\hat{\mathbf{w}}})$

- with Hessian matrix $\mathbf{H}_{\hat{\mathbf{w}}} = \nabla^2_{\mathbf{w}} \log p(\mathbf{x}, \mathbf{y}; \mathbf{w})|_{\hat{\mathbf{w}}}$

Linearised Laplace BNNs

- Linearise the NN at $\hat{\mathbf{w}}$

$$f_{\mathbf{w}}(\mathbf{x}) \approx f_{\hat{\mathbf{w}}}(\mathbf{x}) + \mathbf{J}_{\hat{\mathbf{w}}}(\mathbf{x})(\mathbf{w} - \hat{\mathbf{w}})$$

- using Jacobian $\mathbf{J}_{\hat{\mathbf{w}}}(\mathbf{x}) = \partial_{\mathbf{w}} f_{\mathbf{w}}(\mathbf{x})|_{\mathbf{w}=\hat{\mathbf{w}}} \in \mathbb{R}^{O \times D}$

$D = \dim(\mathbf{w})$

Linearised Laplace BNNs

- Linearise the NN at $\hat{\mathbf{w}}$

$$f_{\mathbf{w}}(\mathbf{x}) \approx f_{\hat{\mathbf{w}}}(\mathbf{x}) + \mathbf{J}_{\hat{\mathbf{w}}}(\mathbf{x})(\mathbf{w} - \hat{\mathbf{w}})$$

- using Jacobian $\mathbf{J}_{\hat{\mathbf{w}}}(\mathbf{x}) = \partial_{\mathbf{w}} f_{\mathbf{w}}(\mathbf{x})|_{\mathbf{w}=\hat{\mathbf{w}}} \in \mathbb{R}^{O \times D}$ $\quad D = \dim(\mathbf{w})$

- Linearised Laplace approximation with prior $\mathcal{N}(\mathbf{0}, \alpha^{-1}\mathbf{I})$ gives

$$q(\mathbf{w}|\mathcal{D}) = \mathcal{N}\left(\mathbf{w} \mid \hat{\mathbf{w}}, (\mathrm{GGN}_{\hat{\mathbf{w}}} + \alpha\mathbf{I})^{-1}\right)$$

Linearised Laplace BNNs

- Linearise the NN at $\hat{\mathbf{w}}$

$$f_{\mathbf{w}}(\mathbf{x}) \approx f_{\hat{\mathbf{w}}}(\mathbf{x}) + \mathbf{J}_{\hat{\mathbf{w}}}(\mathbf{x})(\mathbf{w} - \hat{\mathbf{w}})$$

- using Jacobian $\mathbf{J}_{\hat{\mathbf{w}}}(\mathbf{x}) = \partial_{\mathbf{w}} f_{\mathbf{w}}(\mathbf{x})|_{\mathbf{w}=\hat{\mathbf{w}}} \in \mathbb{R}^{O \times D}$ $\quad D = \dim(\mathbf{w})$

- Linearised Laplace approximation with prior $\mathcal{N}(\mathbf{0}, \alpha^{-1}\mathbf{I})$ gives

$$q(\mathbf{w}|\mathcal{D}) = \mathcal{N}\left(\mathbf{w} \mid \hat{\mathbf{w}}, (\mathrm{GGN}_{\hat{\mathbf{w}}} + \alpha\mathbf{I})^{-1}\right)$$

$$\mathrm{GGN}_{\hat{\mathbf{w}}} = \sum_{n=1}^{N} \mathbf{J}_{\hat{\mathbf{w}}}(\mathbf{x}_n)^\top \mathbf{H}(\mathbf{x}_n) \mathbf{J}_{\hat{\mathbf{w}}}(\mathbf{x}_n)$$

Linearised Laplace BNNs

- Linearise the NN at $\hat{\mathbf{w}}$

$$f_{\mathbf{w}}(\mathbf{x}) \approx f_{\hat{\mathbf{w}}}(\mathbf{x}) + \mathbf{J}_{\hat{\mathbf{w}}}(\mathbf{x})(\mathbf{w} - \hat{\mathbf{w}})$$

- using Jacobian $\mathbf{J}_{\hat{\mathbf{w}}}(\mathbf{x}) = \partial_{\mathbf{w}} f_{\mathbf{w}}(\mathbf{x})|_{\mathbf{w}=\hat{\mathbf{w}}} \in \mathbb{R}^{O \times D}$ $\quad D = \dim(\mathbf{w})$

- Linearised Laplace approximation with prior $\mathcal{N}(\mathbf{0}, \alpha^{-1}\mathbf{I})$ gives

$$q(\mathbf{w}|\mathcal{D}) = \mathcal{N}\left(\mathbf{w} \mid \hat{\mathbf{w}}, (\mathrm{GGN}_{\hat{\mathbf{w}}} + \alpha \mathbf{I})^{-1}\right)$$

$$\mathrm{GGN}_{\hat{\mathbf{w}}} = \sum_{n=1}^{N} \mathbf{J}_{\hat{\mathbf{w}}}(\mathbf{x}_n)^\top \mathbf{H}(\mathbf{x}_n) \mathbf{J}_{\hat{\mathbf{w}}}(\mathbf{x}_n) \qquad \mathbf{H}(\mathbf{x}) = -\partial^2_{f_{\hat{\mathbf{w}}}(\mathbf{x})} \log p(\mathbf{y}|f_{\hat{\mathbf{w}}}(\mathbf{x})) \in \mathbb{R}^{O \times O}$$

Laplace vs Linearised Laplace

- GGN is commonly used to approximate the Hessian $\mathbf{H}_{\hat{\mathbf{w}}}$
- Immer et al. (2021) argue that this choice implicitly linearises the BNN

$$q(\mathbf{w}|\mathcal{D}) \approx \mathcal{N}(\mathbf{w}|\hat{\mathbf{w}}, -\mathbf{H}_{\hat{\mathbf{w}}}^{-1})$$

$$q(\mathbf{w}|\mathcal{D}) = \mathcal{N}\left(\mathbf{w} \mid \hat{\mathbf{w}}, (\mathrm{GGN}_{\hat{\mathbf{w}}} + \alpha\mathbf{I})^{-1}\right)$$

Laplace vs Linearised Laplace

- GGN is commonly used to approximate the Hessian $\mathbf{H}_{\hat{\mathbf{w}}}$
- Immer et al. (2021) argue that this choice implicitly linearises the BNN

$$q(\mathbf{w}|\mathcal{D}) \approx \mathcal{N}(\mathbf{w}|\hat{\mathbf{w}}, -\mathbf{H}_{\hat{\mathbf{w}}}^{-1})$$

$$q(\mathbf{w}|\mathcal{D}) = \mathcal{N}\left(\mathbf{w} \mid \hat{\mathbf{w}}, (\mathrm{GGN}_{\hat{\mathbf{w}}} + \alpha\mathbf{I})^{-1}\right)$$

$$\mathbf{H}_{\hat{\mathbf{w}}} = \nabla_{\mathbf{w}}^2 \log p(\mathbf{x}, \mathbf{y}; \mathbf{w})|_{\hat{\mathbf{w}}}$$

Laplace vs Linearised Laplace

- GGN is commonly used to approximate the Hessian $\mathbf{H}_{\hat{\mathbf{w}}}$
- Immer et al. (2021) argue that this choice implicitly linearises the BNN

$$q(\mathbf{w}|\mathcal{D}) \approx \mathcal{N}(\mathbf{w}|\hat{\mathbf{w}}, -\mathbf{H}_{\hat{\mathbf{w}}}^{-1})$$

$$\mathbf{H}_{\hat{\mathbf{w}}} = \nabla^2_{\mathbf{w}} \log p(\mathbf{x}, \mathbf{y}; \mathbf{w})|_{\hat{\mathbf{w}}}$$

$$q(\mathbf{w}|\mathcal{D}) = \mathcal{N}\left(\mathbf{w} \mid \hat{\mathbf{w}}, (\mathrm{GGN}_{\hat{\mathbf{w}}} + \alpha\mathbf{I})^{-1}\right)$$

$$\mathrm{GGN}_{\hat{\mathbf{w}}} = \sum_{n=1}^{N} \mathbf{J}_{\hat{\mathbf{w}}}(\mathbf{x}_n)^\top \mathbf{H}(\mathbf{x}_n) \mathbf{J}_{\hat{\mathbf{w}}}(\mathbf{x}_n)$$

$$\mathbf{H}(\mathbf{x}) = -\partial^2_{f_{\hat{\mathbf{w}}}(\mathbf{x})} \log p(\mathbf{y}|f_{\hat{\mathbf{w}}}(\mathbf{x})) \in \mathbb{R}^{O\times O}$$

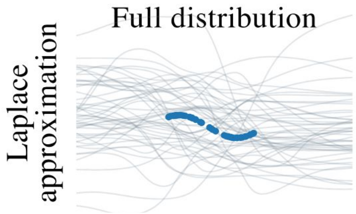
(If likelihood is gaussian, then $\mathbf{H}(\mathbf{x}) = \mathbb{I}_O$)

$$q(\mathbf{w}|\mathcal{D}) = \mathcal{N}\left(\mathbf{w} \mid \hat{\mathbf{w}}, (\mathrm{GGN}_{\hat{\mathbf{w}}} + \alpha\mathbf{I})^{-1}\right)$$

Predictive distributions

- Regular Laplace severely underfits

$$p(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D}) = \mathbb{E}_{\mathbf{w}\sim q}[p(\mathbf{y}^*|f(\mathbf{w}, \mathbf{x}^*))] \approx \frac{1}{S}\sum_{i=1}^{S} p(\mathbf{y}^*|f(\mathbf{w}_i, \mathbf{x}^*)), \quad \mathbf{w}_i \sim q$$

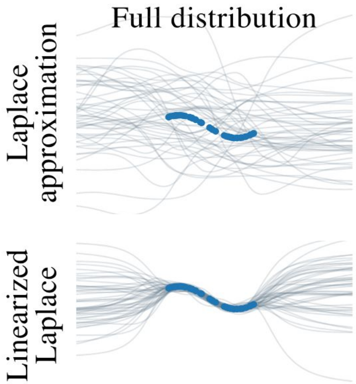


Predictive distributions

$$q(\mathbf{w}|\mathcal{D}) = \mathcal{N}\left(\mathbf{w} \mid \hat{\mathbf{w}}, (\mathrm{GGN}_{\hat{\mathbf{w}}} + \alpha\mathbf{I})^{-1}\right)$$

- Regular Laplace severely underfits

$$p(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D}) = \mathbb{E}_{\mathbf{w}\sim q}[p(\mathbf{y}^*|f(\mathbf{w}, \mathbf{x}^*))] \approx \frac{1}{S}\sum_{i=1}^{S} p(\mathbf{y}^*|f(\mathbf{w}_i, \mathbf{x}^*)), \quad \mathbf{w}_i \sim q$$



- Linearised predictions perform much better

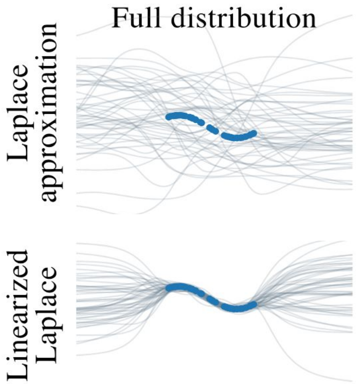
$$p(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D}) = \mathbb{E}_{\mathbf{w}\sim q}[p(\mathbf{y}^*|f_{\mathrm{lin}}^{\hat{\mathbf{w}}}(\mathbf{w}, \mathbf{x}^*))] \approx \frac{1}{S}\sum_{i=1}^{S} p(\mathbf{y}^*|f_{\mathrm{lin}}^{\hat{\mathbf{w}}}(\mathbf{w}_i, \mathbf{x}^*)), \quad \mathbf{w}_i \sim q$$

$$q(\mathbf{w}|\mathcal{D}) = \mathcal{N}\left(\mathbf{w} \mid \hat{\mathbf{w}}, (\mathrm{GGN}_{\hat{\mathbf{w}}} + \alpha\mathbf{I})^{-1}\right)$$

Predictive distributions

- Regular Laplace severely underfits

$$p(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D}) = \mathbb{E}_{\mathbf{w}\sim q}[p(\mathbf{y}^*|f(\mathbf{w}, \mathbf{x}^*))] \approx \frac{1}{S}\sum_{i=1}^{S} p(\mathbf{y}^*|f(\mathbf{w}_i, \mathbf{x}^*)), \quad \mathbf{w}_i \sim q$$



- Linearised predictions perform much better

$$p(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D}) = \mathbb{E}_{\mathbf{w}\sim q}[p(\mathbf{y}^*|f_{\mathrm{lin}}^{\hat{\mathbf{w}}}(\mathbf{w}, \mathbf{x}^*))] \approx \frac{1}{S}\sum_{i=1}^{S} p(\mathbf{y}^*|f_{\mathrm{lin}}^{\hat{\mathbf{w}}}(\mathbf{w}_i, \mathbf{x}^*)), \quad \mathbf{w}_i \sim q$$

- Why does adding another degree of approximation improve performance?

This paper: linearised laplace is better because it’s invariant to reparameterisation

- BNNs are massively overparameterised, with many points in weight-space corresponding to identical functions

This paper: linearised laplace is better because it's invariant to reparameterisation

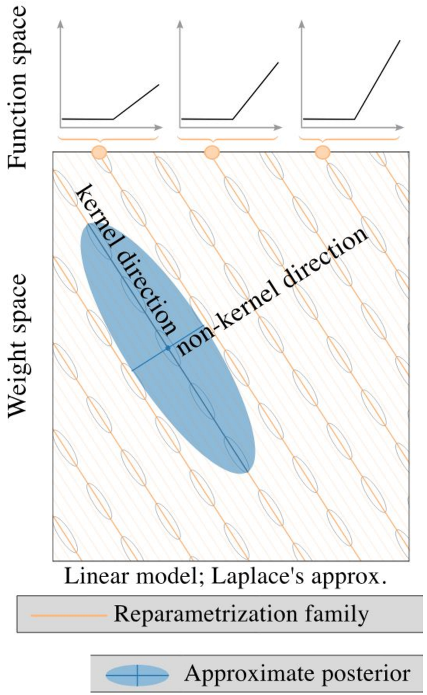
- BNNs are massively overparameterised, with many points in weight-space corresponding to identical functions

- Consider the NN $f(x) = w_1 \mathrm{ReLU}\left(w_2\, x\right);\ f : \mathbb{R} \to \mathbb{R}$
- For any $\alpha > 0$, (w_1, w_2) and $({}^{w_1}/\alpha, \alpha w_2)$ are equivalent

This paper: linearised laplace is better because it's invariant to reparameterisation

- BNNs are massively overparameterised, with many points in weight-space corresponding to identical functions

- Consider the NN $f(x) = w_1 \mathrm{ReLU}\,(w_2\, x)\,;\; f : \mathbb{R} \to \mathbb{R}$
- For any $\alpha > 0$, (w_1, w_2) and $({}^{w_1}/\alpha, \alpha w_2)$ are equivalent

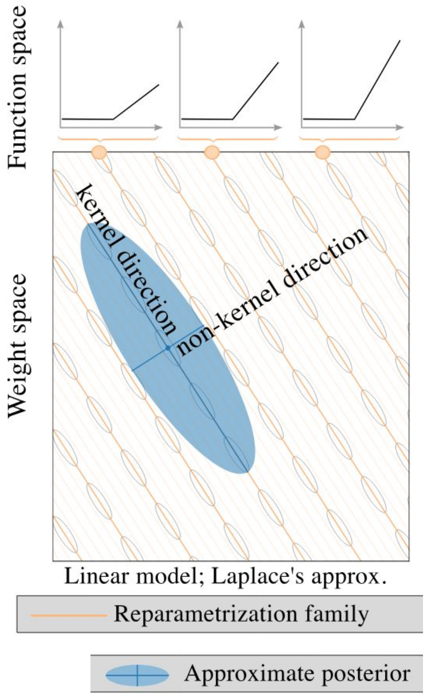


This paper: linearised laplace is better because it's invariant to reparameterisation

- BNNs are massively overparameterised, with many points in weight-space corresponding to identical functions

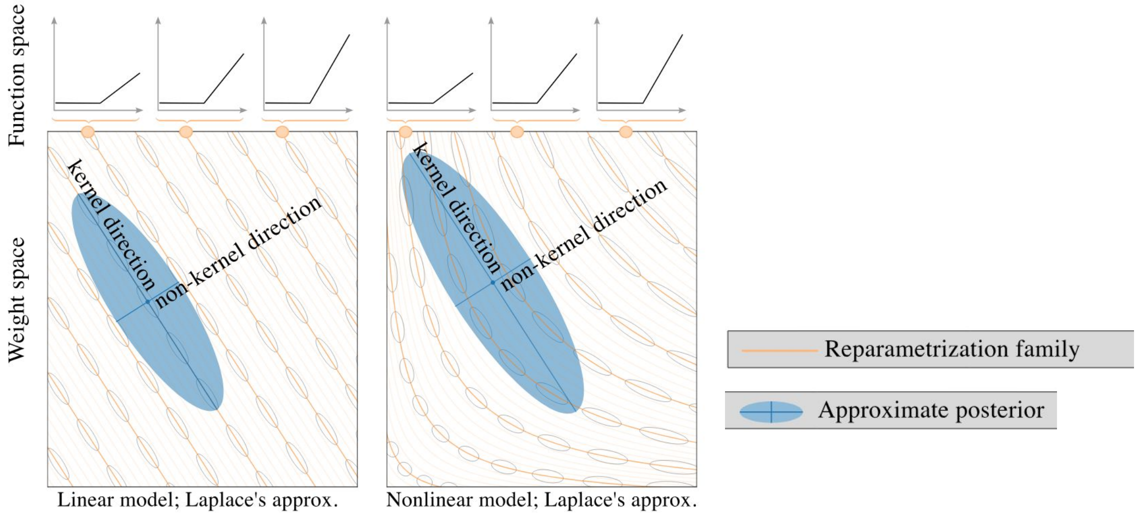
- Consider the NN $f(x) = w_1 \mathrm{ReLU}\,(w_2\, x)\,;\; f : \mathbb{R} \to \mathbb{R}$
- For any $\alpha > 0$, (w_1, w_2) and $(w_1/\alpha, \alpha w_2)$ are equivalent

- The GGN covariance naturally aligns the linearised Laplace approx. post. with the reparameterisation (kernel) direction



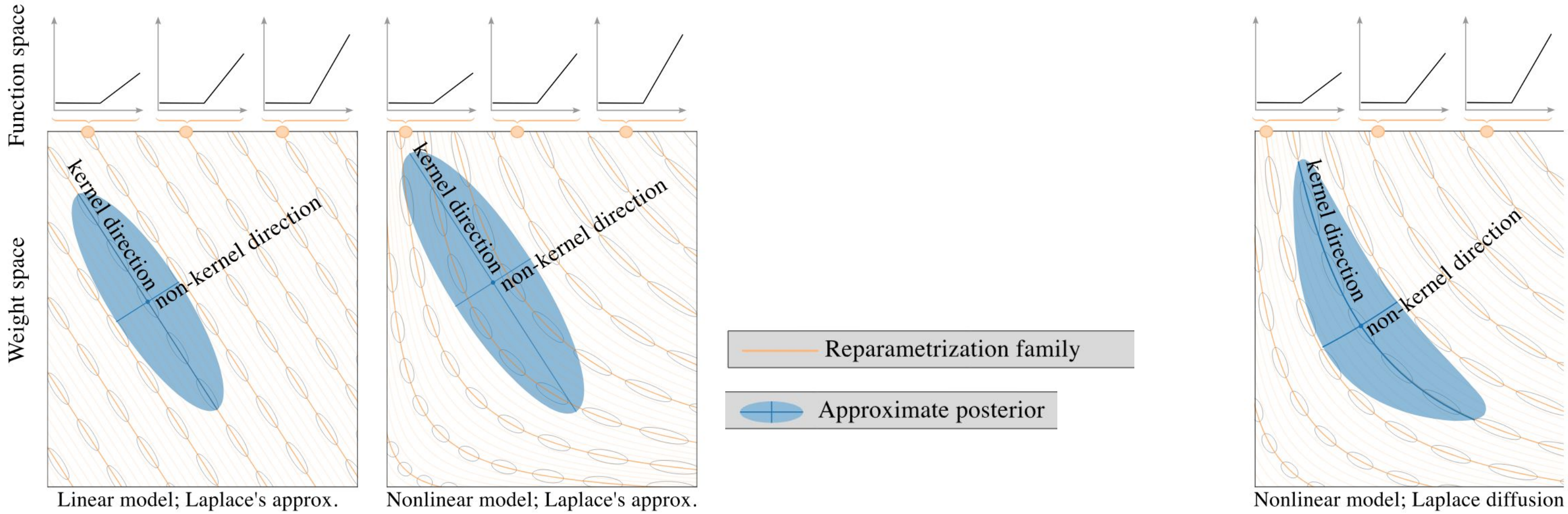
This paper: linearised laplace is better because it’s invariant to reparameterisation

- The regular Laplace approximation puts a crude pdf over the space of (nonlinear) NN functions
  - Identical functions are given different masses (this also messes up marginal likelihood estimates)



This paper: linearised laplace is better because it's invariant to reparameterisation

- The regular Laplace approximation puts a crude pdf over the space of (nonlinear) NN functions
  - Identical functions are given different masses (this also messes up marginal likelihood estimates)



Section 2: Reparameterisation of linear functions

- Consider a linear function $f(\mathbf{w}) = \mathbf{A}\mathbf{w} + \mathbf{b}$

  and a reparameterisation $g : \mathbb{R}^D \to \mathbb{R}^D$ such that $f(g(\mathbf{w})) = f(\mathbf{w})$

Section 2: Reparameterisation of linear functions

- Consider a linear function $f(\mathbf{w}) = \mathbf{A}\mathbf{w} + \mathbf{b}$

  and a reparameterisation $g : \mathbb{R}^D \to \mathbb{R}^D$ such that $f(g(\mathbf{w})) = f(\mathbf{w})$

- Then $\mathbf{A}(g(\mathbf{w}) - \mathbf{w}) = \mathbf{0}$

Section 2: Reparameterisation of linear functions

- Consider a linear function $f(\mathbf{w}) = \mathbf{A}\mathbf{w} + \mathbf{b}$

  and a reparameterisation $g : \mathbb{R}^D \to \mathbb{R}^D$ such that $f(g(\mathbf{w})) = f(\mathbf{w})$

- Then $\mathbf{A}(g(\mathbf{w}) - \mathbf{w}) = \mathbf{0}$
- So the direction of movement between $\mathbf{w}$ and $g(\mathbf{w})$ lies in the kernel/nullspace of $\mathbf{A}$

$$g(\mathbf{w}) - \mathbf{w} \in \ker(\mathbf{A})$$

Find the kernel of the linearised NN (and avoid it!)

$$f_{\text{lin}}^{\mathbf{w}'} : \mathbf{w}, \mathbf{x} \mapsto f_{\mathbf{w}'}(\mathbf{x}) + \mathbf{J}_{\mathbf{w}'}(\mathbf{x})(\mathbf{w} - \mathbf{w}')$$

Find the kernel of the linearised NN (and avoid it!)

$$f_{\text{lin}}^{\mathbf{w}'} : \mathbf{w}, \mathbf{x} \mapsto f_{\mathbf{w}'}(\mathbf{x}) + \mathbf{J}_{\mathbf{w}'}(\mathbf{x})(\mathbf{w} - \mathbf{w}')$$

$$\ker(\mathbf{J}_{\mathbf{w}}) = \ker(\mathrm{GGN}_{\mathbf{w}})$$

$$\mathrm{GGN}_{\mathbf{w}} = \mathbf{J}_{\mathbf{w}}^{\top}\mathbf{J}_{\mathbf{w}}$$
$$\mathrm{NTK}_{\mathbf{w}} = \mathbf{J}_{\mathbf{w}}\mathbf{J}_{\mathbf{w}}^{\top}$$

Find the kernel of the linearised NN (and avoid it!)

$$f_{\text{lin}}^{\mathbf{w}'} : \mathbf{w}, \mathbf{x} \mapsto f_{\mathbf{w}'}(\mathbf{x}) + \mathbf{J}_{\mathbf{w}'}(\mathbf{x})(\mathbf{w} - \mathbf{w}')$$

$$\ker(\mathbf{J}_{\mathbf{w}}) = \ker(\text{GGN}_{\mathbf{w}})$$

$$\text{GGN}_{\mathbf{w}} = \mathbf{J}_{\mathbf{w}}^{\top}\mathbf{J}_{\mathbf{w}}$$
$$\text{NTK}_{\mathbf{w}} = \mathbf{J}_{\mathbf{w}}\mathbf{J}_{\mathbf{w}}^{\top}$$

- Because the GGN is a self-adjoint operator (positive semi-definite matrix)

$$\text{im}(\text{GGN}_{\mathbf{w}}) \oplus \ker(\text{GGN}_{\mathbf{w}}) = \mathbb{R}^D$$

Find the kernel of the linearised NN (and avoid it!)

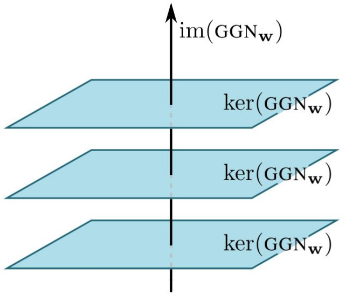
$$f_{\text{lin}}^{\mathbf{w}'} : \mathbf{w}, \mathbf{x} \mapsto f_{\mathbf{w}'}(\mathbf{x}) + \mathbf{J}_{\mathbf{w}'}(\mathbf{x})(\mathbf{w} - \mathbf{w}')$$

$$\ker(\mathbf{J}_{\mathbf{w}}) = \ker(\text{GGN}_{\mathbf{w}})$$

$$\text{GGN}_{\mathbf{w}} = \mathbf{J}_{\mathbf{w}}^{\top}\mathbf{J}_{\mathbf{w}}$$
$$\text{NTK}_{\mathbf{w}} = \mathbf{J}_{\mathbf{w}}\mathbf{J}_{\mathbf{w}}^{\top}$$

- Because the GGN is a self-adjoint operator (positive semi-definite matrix)

$$\text{im}(\text{GGN}_{\mathbf{w}}) \oplus \ker(\text{GGN}_{\mathbf{w}}) = \mathbb{R}^D$$



Find the kernel of the linearised NN (and avoid it!)

$$f_{\text{lin}}^{\mathbf{w}'} : \mathbf{w}, \mathbf{x} \mapsto f_{\mathbf{w}'}(\mathbf{x}) + \mathbf{J}_{\mathbf{w}'}(\mathbf{x})(\mathbf{w} - \mathbf{w}')$$

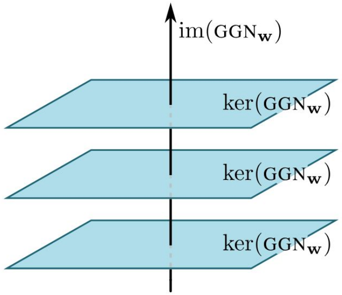
$$\ker(\mathbf{J}_{\mathbf{w}}) = \ker(\text{GGN}_{\mathbf{w}})$$

$$\text{GGN}_{\mathbf{w}} = \mathbf{J}_{\mathbf{w}}^{\top}\mathbf{J}_{\mathbf{w}}$$
$$\text{NTK}_{\mathbf{w}} = \mathbf{J}_{\mathbf{w}}\mathbf{J}_{\mathbf{w}}^{\top}$$

- Because the GGN is a self-adjoint operator (positive semi-definite matrix)

$$\text{im}(\text{GGN}_{\mathbf{w}}) \oplus \ker(\text{GGN}_{\mathbf{w}}) = \mathbb{R}^D$$

- $\text{im}(\text{GGN}_{\mathbf{w}})$ spans to the *effective parameters*
  - (ideally we'd just explore this)



Find the kernel of the linearised NN (and avoid it!)

$$f^{\mathbf{w}'}_{\text{lin}} : \mathbf{w}, \mathbf{x} \mapsto f_{\mathbf{w}'}(\mathbf{x}) + \mathbf{J}_{\mathbf{w}'}(\mathbf{x})(\mathbf{w} - \mathbf{w}')$$

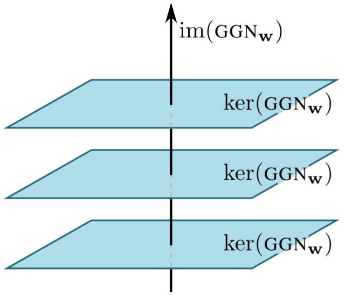
$$\ker(\mathbf{J}_{\mathbf{w}}) = \ker(\text{GGN}_{\mathbf{w}})$$

$$\text{GGN}_{\mathbf{w}} = \mathbf{J}_{\mathbf{w}}^\top \mathbf{J}_{\mathbf{w}}$$
$$\text{NTK}_{\mathbf{w}} = \mathbf{J}_{\mathbf{w}} \mathbf{J}_{\mathbf{w}}^\top$$

- Because the GGN is a self-adjoint operator (positive semi-definite matrix)

$$\text{im}(\text{GGN}_{\mathbf{w}}) \oplus \ker(\text{GGN}_{\mathbf{w}}) = \mathbb{R}^D$$

- $\text{im}(\text{GGN}_{\mathbf{w}})$ spans to the *effective parameters*
  - (ideally we'd just explore this)
- $\ker(\text{GGN}_{\mathbf{w}})$ corresponds to the directions of reparameterisation



Laplace samples can be decomposed into image and kernel contributions

- Let $\mathbf{U}^T \mathbf{\Lambda} \mathbf{U}$ be the eigendecomposition of $\mathrm{GGN}_{\hat{\mathbf{w}}}$ with $\mathbf{U_1}$ and $\mathbf{U_2}$ corresponding to non-zero and zero eigenvalues respectively

Laplace samples can be decomposed into image and kernel contributions

- Let $\mathbf{U}^T\mathbf{\Lambda}\mathbf{U}$ be the eigendecomposition of $\mathrm{GGN}_{\hat{\mathbf{w}}}$ with $\mathbf{U_1}$ and $\mathbf{U_2}$ corresponding to non-zero and zero eigenvalues respectively

$$\Sigma = \left(\left[\frac{\mathbf{U_1}}{\mathbf{U_2}} \right]^T \left[\begin{array}{c|c} \tilde{\mathbf{\Lambda}} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} \end{array} \right] \left[\frac{\mathbf{U_1}}{\mathbf{U_2}} \right] + \alpha \mathbf{I} \right)^{-1} = \mathbf{U_1^T}(\tilde{\mathbf{\Lambda}} + \alpha \mathbf{I}_k)^{-1}\mathbf{U_1} + \alpha^{-1}\mathbf{U_2^T}\mathbf{U_2}$$

Laplace samples can be decomposed into image and kernel contributions

- Let $\mathbf{U}^T \boldsymbol{\Lambda} \mathbf{U}$ be the eigendecomposition of $\mathrm{GGN}_{\hat{\mathbf{w}}}$ with $\mathbf{U_1}$ and $\mathbf{U_2}$ corresponding to non-zero and zero eigenvalues respectively

$$\Sigma = \left(\left[\frac{\mathbf{U_1}}{\mathbf{U_2}} \right]^T \left[\begin{array}{c|c} \tilde{\boldsymbol{\Lambda}} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} \end{array} \right] \left[\frac{\mathbf{U_1}}{\mathbf{U_2}} \right] + \alpha \mathbf{I} \right)^{-1} = \mathbf{U_1^T} (\tilde{\boldsymbol{\Lambda}} + \alpha \mathbf{I}_k)^{-1} \mathbf{U_1} + \alpha^{-1} \mathbf{U_2^T} \mathbf{U_2}$$

So any Laplace sample can be written $\mathbf{w} = \hat{\mathbf{w}} + \mathbf{w}_{\mathrm{im}} + \mathbf{w}_{\mathrm{ker}}$

Laplace samples can be decomposed into image and kernel contributions

- Let $\mathbf{U}^T \boldsymbol{\Lambda} \mathbf{U}$ be the eigendecomposition of $\mathrm{GGN}_{\hat{\mathbf{w}}}$ with $\mathbf{U_1}$ and $\mathbf{U_2}$ corresponding to non-zero and zero eigenvalues respectively

$$\Sigma = \left(\left[\frac{\mathbf{U_1}}{\mathbf{U_2}} \right]^T \left[\begin{array}{c|c} \tilde{\boldsymbol{\Lambda}} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} \end{array} \right] \left[\frac{\mathbf{U_1}}{\mathbf{U_2}} \right] + \alpha \mathbf{I} \right)^{-1} = \mathbf{U}_\mathbf{1}^\mathrm{T} (\tilde{\boldsymbol{\Lambda}} + \alpha \mathbf{I}_k)^{-1} \mathbf{U_1} + \alpha^{-1} \mathbf{U}_\mathbf{2}^\mathrm{T} \mathbf{U_2}$$

So any Laplace sample can be written $\mathbf{w} = \hat{\mathbf{w}} + \mathbf{w}_{\mathrm{im}} + \mathbf{w}_{\mathrm{ker}}$

(Note that all probability mass in the kernel comes from the prior)

Linearised laplace automatically ignores $\mathbf{w}_{\mathrm{ker}}$

$$f_{\mathrm{lin}}^{\hat{\mathbf{w}}}(\hat{\mathbf{w}}+\mathbf{w}_{\mathrm{ker}}+\mathbf{w}_{\mathrm{im}},\mathbf{x}) = f_{\mathrm{lin}}^{\hat{\mathbf{w}}}(\hat{\mathbf{w}}+\mathbf{w}_{\mathrm{im}},\mathbf{x})$$

Linearised laplace automatically ignores $\mathbf{w}_{\mathrm{ker}}$

$$f_{\mathrm{lin}}^{\hat{\mathbf{w}}}(\hat{\mathbf{w}} + \mathbf{w}_{\mathrm{ker}} + \mathbf{w}_{\mathrm{im}}, \mathbf{x}) = f_{\mathrm{lin}}^{\hat{\mathbf{w}}}(\hat{\mathbf{w}} + \mathbf{w}_{\mathrm{im}}, \mathbf{x})$$

- It doesn’t matter which reparameterisation gets sampled if your model is linear!

Linearised laplace automatically ignores $\mathbf{w}_{\mathrm{ker}}$

$$f_{\mathrm{lin}}^{\hat{\mathbf{w}}}(\hat{\mathbf{w}}+\mathbf{w}_{\mathrm{ker}}+\mathbf{w}_{\mathrm{im}},\mathbf{x}) = f_{\mathrm{lin}}^{\hat{\mathbf{w}}}(\hat{\mathbf{w}}+\mathbf{w}_{\mathrm{im}},\mathbf{x})$$

- It doesn't matter which reparameterisation gets sampled if your model is linear!
- Regular Laplace doesn't ignore $\mathbf{w}_{\mathrm{ker}}$ because the kernel doesn't correspond to reparameterisations

Linearised laplace automatically ignores $\mathbf{w}_{\text{ker}}$

$$f_{\text{lin}}^{\hat{\mathbf{w}}}(\hat{\mathbf{w}}+\mathbf{w}_{\text{ker}}+\mathbf{w}_{\text{im}},\mathbf{x}) = f_{\text{lin}}^{\hat{\mathbf{w}}}(\hat{\mathbf{w}}+\mathbf{w}_{\text{im}},\mathbf{x})$$

- It doesn’t matter which reparameterisation gets sampled if your model is linear!
- Regular Laplace doesn’t ignore $\mathbf{w}_{\text{ker}}$ because the kernel doesn’t correspond to reparameterisations
  - Adds ‘incorrect’ degrees of freedom to the approx. post.

Linearised laplace automatically ignores $\mathbf{w}_{\text{ker}}$

$$f_{\text{lin}}^{\hat{\mathbf{w}}}(\hat{\mathbf{w}} + \mathbf{w}_{\text{ker}} + \mathbf{w}_{\text{im}}, \mathbf{x}) = f_{\text{lin}}^{\hat{\mathbf{w}}}(\hat{\mathbf{w}} + \mathbf{w}_{\text{im}}, \mathbf{x})$$

- It doesn't matter which reparameterisation gets sampled if your model is linear!
- Regular Laplace doesn't ignore $\mathbf{w}_{\text{ker}}$ because the kernel doesn't correspond to reparameterisations
  - Adds 'incorrect' degrees of freedom to the approx. post.

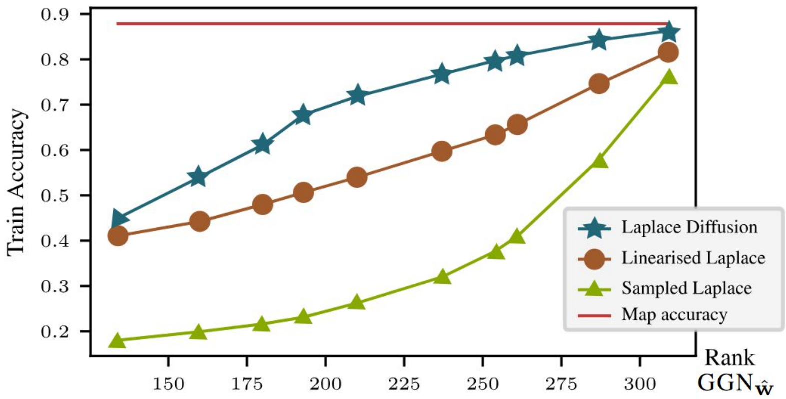
- This suggests that if $\ker(\text{GGN}_{\hat{\mathbf{w}}})$ is small (i.e. $\text{GGN}_{\hat{\mathbf{w}}}$ has a high rank) then regular Laplace should perform similarly to linearised Laplace

Linearised laplace automatically ignores $\mathbf{w}_{\text{ker}}$

$$f_{\text{lin}}^{\hat{\mathbf{w}}}(\hat{\mathbf{w}} + \mathbf{w}_{\text{ker}} + \mathbf{w}_{\text{im}}, \mathbf{x}) = f_{\text{lin}}^{\hat{\mathbf{w}}}(\hat{\mathbf{w}} + \mathbf{w}_{\text{im}}, \mathbf{x})$$

- It doesn't matter which reparameterisation gets sampled if your model is linear!
- Regular Laplace doesn't ignore $\mathbf{w}_{\text{ker}}$ because the kernel doesn't correspond to reparameterisations
  - Adds 'incorrect' degrees of freedom to the approx. post.

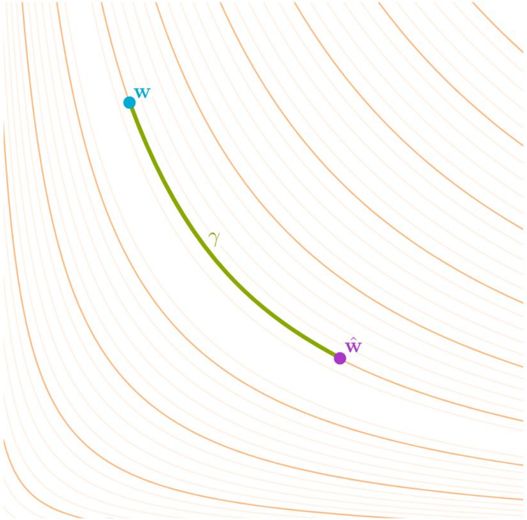
- This suggests that if $\ker(\text{GGN}_{\hat{\mathbf{w}}})$ is small (i.e. $\text{GGN}_{\hat{\mathbf{w}}}$ has a high rank) then regular Laplace should perform similarly to linearised Laplace



Small CNN on MNIST so GGN can be computed exactly – (rank increases with more data if we keep the same number of parameters)

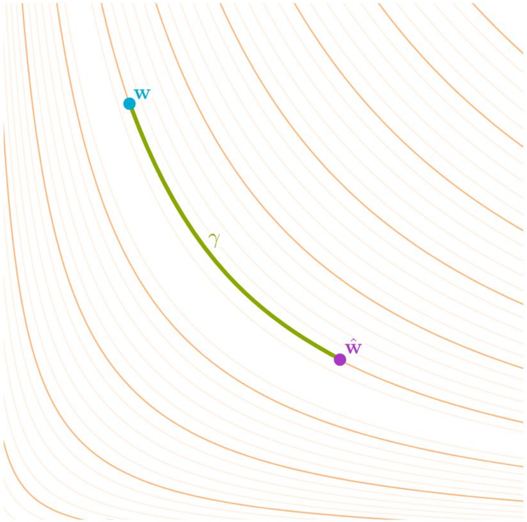
Section 3: Reparameterisation of NNs

- They do the same analysis but for nonlinear models (e.g. regular NNs)

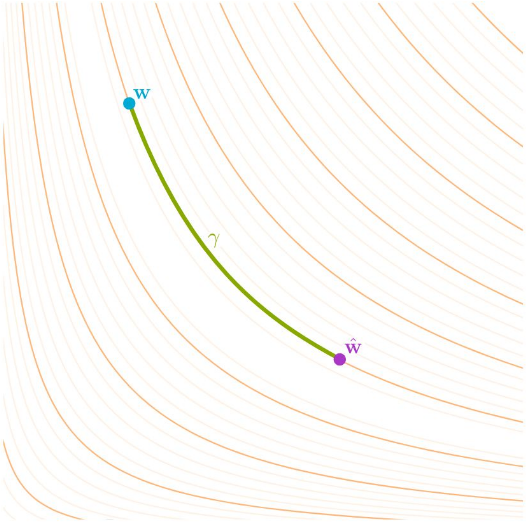


Section 3: Reparameterisation of NNs

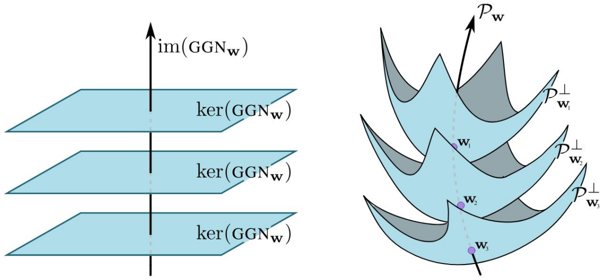
- They do the same analysis but for nonlinear models (e.g. regular NNs)
- We can similarly decompose Laplace samples into a combination of two manifolds embedded in $\mathbb{R}^D$, $(\mathcal{P}_{\mathbf{w}}, \mathfrak{m})$ and $(\mathcal{P}^{\perp}_{\mathbf{w}}, \mathfrak{m}^{\perp})$, which act like $\mathrm{im}(\mathrm{GGN}_{\mathbf{w}})$ and $\ker(\mathrm{GGN}_{\mathbf{w}})$



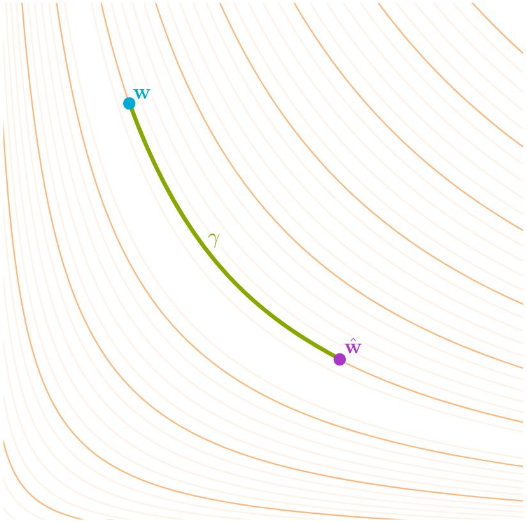
Section 3: Reparameterisation of NNs



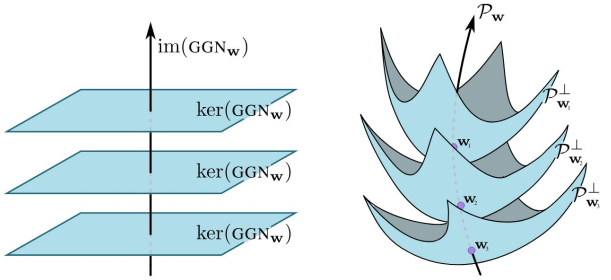
- They do the same analysis but for nonlinear models (e.g. regular NNs)
- We can similarly decompose Laplace samples into a combination of two manifolds embedded in $\mathbb{R}^D$, $(\mathcal{P}_{\mathbf{w}}, \mathfrak{m})$ and $(\mathcal{P}^{\perp}_{\mathbf{w}}, \mathfrak{m}^{\perp})$, which act like $\mathrm{im}(\mathrm{GGN}_{\mathbf{w}})$ and $\ker(\mathrm{GGN}_{\mathbf{w}})$



Section 3: Reparameterisation of NNs



- They do the same analysis but for nonlinear models (e.g. regular NNs)
- We can similarly decompose Laplace samples into a combination of two manifolds embedded in $\mathbb{R}^D$, $(\mathcal{P}_\mathbf{w}, \mathfrak{m})$ and $(\mathcal{P}_\mathbf{w}^\perp, \mathfrak{m}^\perp)$, which act like $\mathrm{im}(\mathrm{GGN}_\mathbf{w})$ and $\ker(\mathrm{GGN}_\mathbf{w})$
- Then we can ignore the reparameterisation directions by only exploring $(\mathcal{P}_\mathbf{w}, \mathfrak{m})$ (*Laplace diffusion*)



Define Effective Parameter-Space as a Quotient Group

$\mathbf{w}_1 \sim \mathbf{w}_2$ iff there exists a smooth path between $\mathbf{w}_1$ and $\mathbf{w}_2$ such that

$$f(\mathbf{w}_1, \mathbf{x}) = f(\mathbf{w}_2, \mathbf{x}) \qquad \forall \mathbf{x} \in \mathcal{D}_{\text{train}}$$

Define Effective Parameter-Space as a Quotient Group

$\mathbf{w}_1 \sim \mathbf{w}_2$ iff there exists a smooth path between $\mathbf{w}_1$ and $\mathbf{w}_2$ such that

$$f(\mathbf{w}_1, \mathbf{x}) = f(\mathbf{w}_2, \mathbf{x}) \qquad \forall \mathbf{x} \in \mathcal{D}_{\text{train}}$$

Then we define the effective parameter space as the quotient group $\mathcal{P} = \mathbb{R}^D / \sim$

(i.e. only consider parameters that give us unique functions)

Define a Reparameterisation Distance and get a (Pseudo-)Riemannian manifold

We'd like to define a distance such that $\mathrm{dist}(\mathbf{w}_1, \mathbf{w}_2) = 0 \quad \Leftrightarrow \quad \mathbf{w}_1 \sim \mathbf{w}_2$

Define a Reparameterisation Distance and get a (Pseudo-)Riemannian manifold

We'd like to define a distance such that $\mathrm{dist}(\mathbf{w}_1, \mathbf{w}_2) = 0 \quad \Leftrightarrow \quad \mathbf{w}_1 \sim \mathbf{w}_2$

$$\mathrm{dist}^2(\mathbf{w}, \mathbf{w} + \boldsymbol{\epsilon}) = \sum_{n=1}^{N} \|f(\mathbf{w}, \mathbf{x}_n) - f(\mathbf{w} + \epsilon, \mathbf{x}_n)\|^2 = \boldsymbol{\epsilon}^\top \mathrm{GGN}_{\mathbf{w}} \boldsymbol{\epsilon} + \mathcal{O}(\epsilon^3)$$

Define a Reparameterisation Distance and get a (Pseudo-)Riemannian manifold

We'd like to define a distance such that $\mathrm{dist}(\mathbf{w}_1, \mathbf{w}_2) = 0 \quad \Leftrightarrow \quad \mathbf{w}_1 \sim \mathbf{w}_2$

$$\mathrm{dist}^2(\mathbf{w}, \mathbf{w} + \boldsymbol{\epsilon}) = \sum_{n=1}^{N} \|f(\mathbf{w}, \mathbf{x}_n) - f(\mathbf{w} + \epsilon, \mathbf{x}_n)\|^2 = \boldsymbol{\epsilon}^\top \mathrm{GGN}_{\mathbf{w}} \boldsymbol{\epsilon} + \mathcal{O}(\epsilon^3)$$

So the GGN matrix infinitesimally defines an inner product, a (pseudo-)Riemannian metric!

Define a Reparameterisation Distance and get a (Pseudo-)Riemannian manifold

We'd like to define a distance such that $\operatorname{dist}(\mathbf{w}_1, \mathbf{w}_2) = 0 \quad \Leftrightarrow \quad \mathbf{w}_1 \sim \mathbf{w}_2$

$$\operatorname{dist}^2(\mathbf{w}, \mathbf{w} + \boldsymbol{\epsilon}) = \sum_{n=1}^{N} \|f(\mathbf{w}, \mathbf{x}_n) - f(\mathbf{w} + \epsilon, \mathbf{x}_n)\|^2 = \boldsymbol{\epsilon}^\top \mathrm{GGN}_{\mathbf{w}} \boldsymbol{\epsilon} + \mathcal{O}(\epsilon^3)$$

So the GGN matrix infinitesimally defines an inner product, a (pseudo-)Riemannian metric!

NOTE:

1. GGN is rank-deficient (hence pseudo-riemannian)

Define a Reparameterisation Distance and get a (Pseudo-)Riemannian manifold

We'd like to define a distance such that $\mathrm{dist}(\mathbf{w}_1, \mathbf{w}_2) = 0 \quad \Leftrightarrow \quad \mathbf{w}_1 \sim \mathbf{w}_2$

$$\mathrm{dist}^2(\mathbf{w}, \mathbf{w} + \boldsymbol{\epsilon}) = \sum_{n=1}^{N} \|f(\mathbf{w}, \mathbf{x}_n) - f(\mathbf{w} + \epsilon, \mathbf{x}_n)\|^2 = \boldsymbol{\epsilon}^\top \mathrm{GGN}_{\mathbf{w}} \boldsymbol{\epsilon} + \mathcal{O}(\epsilon^3)$$

So the GGN matrix infinitesimally defines an inner product, a (pseudo-)Riemannian metric!

NOTE:

1. GGN is rank-deficient (hence pseudo-riemannian)
2. $\mathbf{w}_1 \sim \mathbf{w}_2$ doesn't then necessarily mean $\mathrm{GGN}_{\mathbf{w}_1} = \mathrm{GGN}_{\mathbf{w}_2}$ (pseudo-metric might be different based on your Laplace centre/mode (... but infinitesimally that doesn't matter too much))

Equivalence of the manifold and the quotient group

Proposition 4.4. *There exists a bijection between* $(\mathbb{R}^D, \mathrm{GGN}_{\mathbf{w}})$ *and* $\mathcal{P}$.

Equivalence of the manifold and the quotient group

Proposition 4.4. *There exists a bijection between* $(\mathbb{R}^D, \mathrm{GGN}_{\mathbf{w}})$ *and* $\mathcal{P}$.

(Basically, using the tangent space of the GGN gives us a path along the homotopies in the equivalence classes.)

Equivalence of the manifold and the quotient group

Proposition 4.4. *There exists a bijection between* $(\mathbb{R}^D, \mathrm{GGN}_{\mathbf{w}})$ *and* $\mathcal{P}$.

(Basically, using the tangent space of the GGN gives us a path along the homotopies in the equivalence classes.)

- Why do we care? Answer: Riemannian diffusion.

Equivalence of the manifold and the quotient group

Proposition 4.4. *There exists a bijection between* $(\mathbb{R}^D, \mathrm{GGN}_{\mathbf{w}})$ *and* $\mathcal{P}$.

(Basically, using the tangent space of the GGN gives us a path along the homotopies in the equivalence classes.)

- Why do we care? Answer: Riemannian diffusion.
- We can run a Markov chain to give us samples from the Riemannian manifold $(\mathrm{M}, \mathbf{G})$

Equivalence of the manifold and the quotient group

Proposition 4.4. *There exists a bijection between* $(\mathbb{R}^D, \mathrm{GGN}_{\mathbf{w}})$ *and* $\mathcal{P}$.

(Basically, using the tangent space of the GGN gives us a path along the homotopies in the equivalence classes.)

- Why do we care? Answer: Riemannian diffusion.
- We can run a Markov chain to give us samples from the Riemannian manifold $(\mathrm{M}, \mathbf{G})$

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \sqrt{2h_t}\mathbf{G}(\mathbf{w}_t)^{-\frac{1}{2}}\boldsymbol{\epsilon}, \text{ where } \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

Equivalence of the manifold and the quotient group

Proposition 4.4. *There exists a bijection between* $(\mathbb{R}^D, \mathrm{GGN}_{\mathbf{w}})$ *and* $\mathcal{P}$.

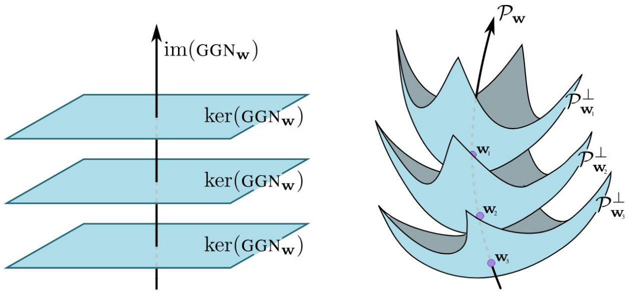
(Basically, using the tangent space of the GGN gives us a path along the homotopies in the equivalence classes.)

- Why do we care? Answer: Riemannian diffusion.
- We can run a Markov chain to give us samples from the Riemannian manifold $(\mathrm{M}, \mathbf{G})$

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \sqrt{2h_t}\mathbf{G}(\mathbf{w}_t)^{-\frac{1}{2}}\boldsymbol{\epsilon}, \text{ where } \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

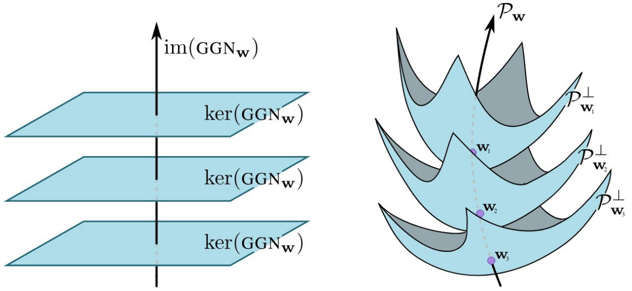
Problem: $(\mathbb{R}^D, \mathrm{GGN}_{\mathbf{w}})$ is only pseudo-Riemannian (GGN is rank-deficient)

Define two Riemannian manifolds which act as $\mathrm{im}(\mathrm{GGN}_{\mathbf{w}})$ and $\ker(\mathrm{GGN}_{\mathbf{w}})$



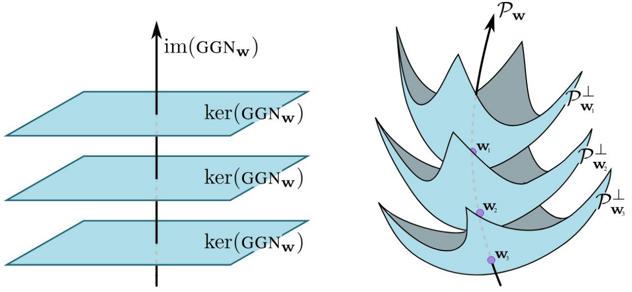
Define two Riemannian manifolds which act as $\mathrm{im}(\mathrm{GGN}_{\mathbf{w}})$ and $\ker(\mathrm{GGN}_{\mathbf{w}})$

1. $(\mathcal{P}_{\mathbf{w}}, \mathfrak{m})$ where $\mathfrak{m} = \mathrm{GGN}_{\mathbf{w}}^{+}$



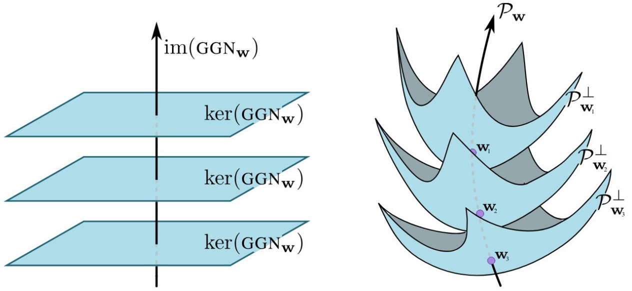
Define two Riemannian manifolds which act as $\mathrm{im}(\mathrm{GGN}_{\mathbf{w}})$ and $\ker(\mathrm{GGN}_{\mathbf{w}})$

1. $(\mathcal{P}_{\mathbf{w}}, \mathfrak{m})$ where $\mathfrak{m} = \mathrm{GGN}_{\mathbf{w}}^{+}$
   - Never intersects the same equivalence class more than once (at least locally)



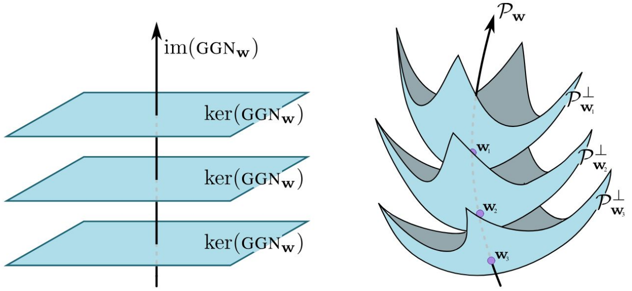
Define two Riemannian manifolds which act as $\mathrm{im}(\mathrm{GGN}_{\mathbf{w}})$ and $\ker(\mathrm{GGN}_{\mathbf{w}})$

1. $(\mathcal{P}_{\mathbf{w}}, \mathfrak{m})$ where $\mathfrak{m} = \mathrm{GGN}_{\mathbf{w}}^{+}$
   - Never intersects the same equivalence class more than once (at least locally)
   - Like the effective parameter manifold (but actually Riemannian)



Define two Riemannian manifolds which act as $\mathrm{im}(\mathrm{GGN}_{\mathbf{w}})$ and $\ker(\mathrm{GGN}_{\mathbf{w}})$

1. $(\mathcal{P}_{\mathbf{w}}, \mathfrak{m})$ where $\mathfrak{m} = \mathrm{GGN}_{\mathbf{w}}^{+}$
   - Never intersects the same equivalence class more than once (at least locally)
   - Like the effective parameter manifold (but actually Riemannian)

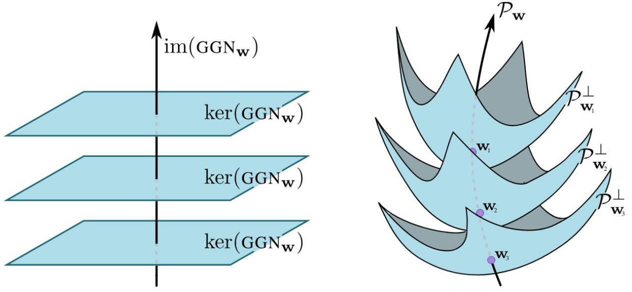


2. $(\mathcal{P}^{\perp}_{\mathbf{w}}, \mathfrak{m}^{\perp})$ where $\mathfrak{m}^{\perp} = \alpha \mathbf{I}$ for $\alpha > 0$

Define two Riemannian manifolds which act as $\mathrm{im}(\mathrm{GGN}_{\mathbf{w}})$ and $\ker(\mathrm{GGN}_{\mathbf{w}})$

1. $(\mathcal{P}_{\mathbf{w}}, \mathfrak{m})$ where $\mathfrak{m} = \mathrm{GGN}_{\mathbf{w}}^{+}$
    - Never intersects the same equivalence class more than once (at least locally)
    - Like the effective parameter manifold (but actually Riemannian)

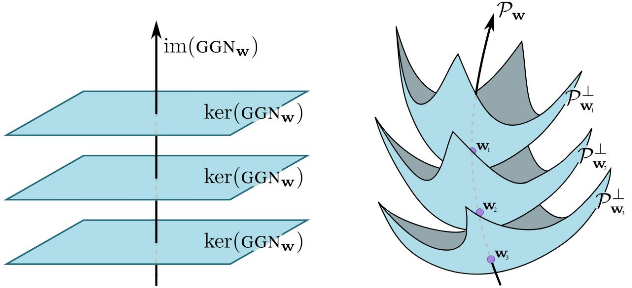
2. $(\mathcal{P}_{\mathbf{w}}^{\perp}, \mathfrak{m}^{\perp})$ where $\mathfrak{m}^{\perp} = \alpha \mathbf{I}$ for $\alpha > 0$
    - Entirely contained in the same equivalence class $\mathcal{P}_{\mathbf{w}}^{\perp} \subseteq [\mathbf{w}]$



Define two Riemannian manifolds which act as $\mathrm{im}(\mathrm{GGN}_{\mathbf{w}})$ and $\ker(\mathrm{GGN}_{\mathbf{w}})$

1. $(\mathcal{P}_{\mathbf{w}}, \mathfrak{m})$ where $\mathfrak{m} = \mathrm{GGN}_{\mathbf{w}}^{+}$
    - Never intersects the same equivalence class more than once (at least locally)
    - Like the effective parameter manifold (but actually Riemannian)

2. $(\mathcal{P}_{\mathbf{w}}^{\perp}, \mathfrak{m}^{\perp})$ where $\mathfrak{m}^{\perp} = \alpha\mathbf{I}$ for $\alpha > 0$
    - Entirely contained in the same equivalence class $\mathcal{P}_{\mathbf{w}}^{\perp} \subseteq [\mathbf{w}]$
    - So contains only functions which are identical on the training set

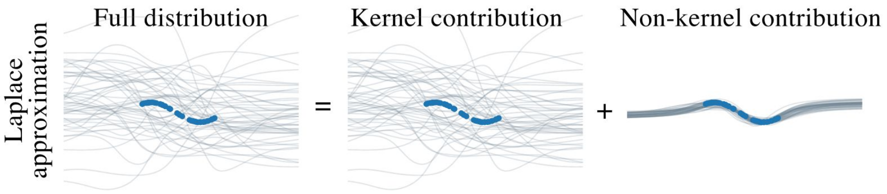


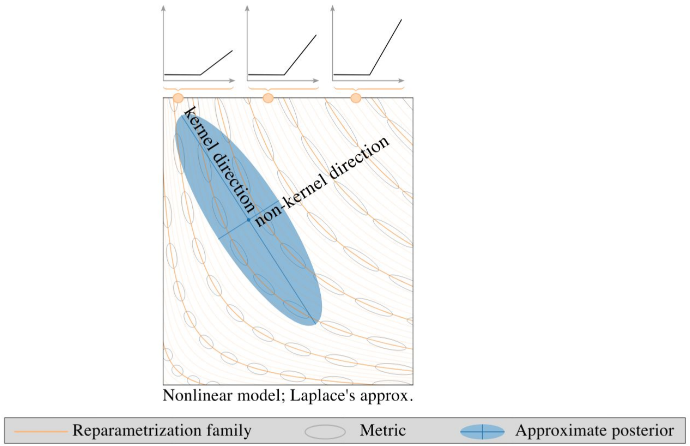
Section 4: Diffusion on various manifolds

0. Effective-parameter is only pseudo-Riemannian $(\mathbb{R}^D, \mathrm{GGN}_{\mathbf{w}})$

Section 4: Diffusion on various manifolds

0. Effective-parameter is only pseudo-Riemannian $(\mathbb{R}^D, \mathbf{GGN_w})$
1. Laplace Approximation $(\mathbb{R}^D, \mathbf{GGN_w} + \alpha\mathbf{I})$

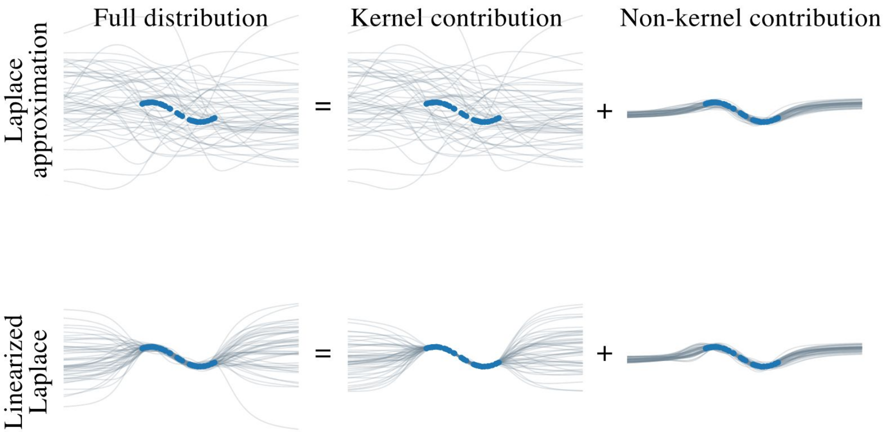


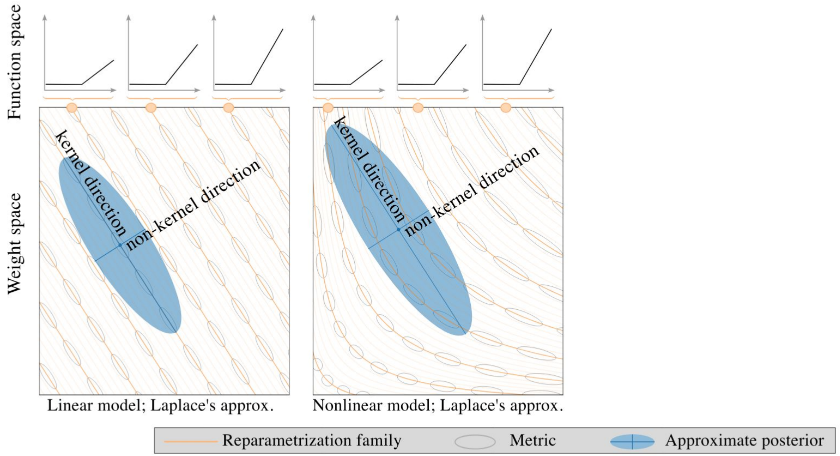


Nonlinear model; Laplace's approx.

Section 4: Diffusion on various manifolds

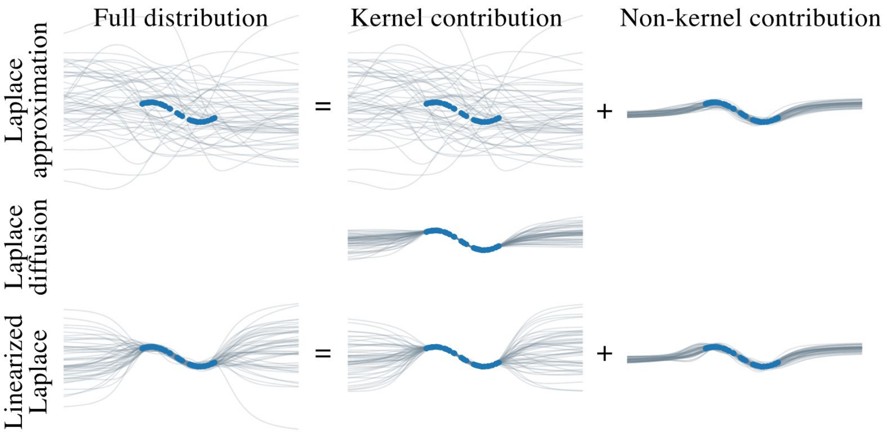
0. Effective-parameter is only pseudo-Riemannian $(\mathbb{R}^D, \mathbf{GGN_w})$
1. Laplace Approximation $(\mathbb{R}^D, \mathbf{GGN_w} + \alpha \mathbf{I})$

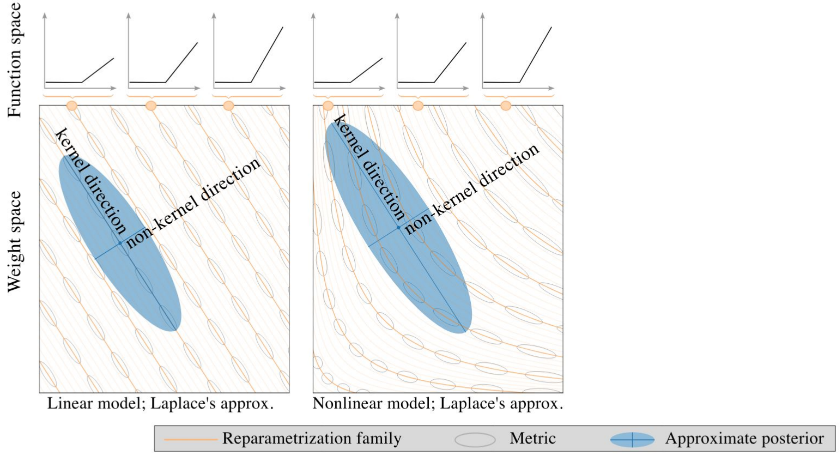




Section 4: Diffusion on various manifolds

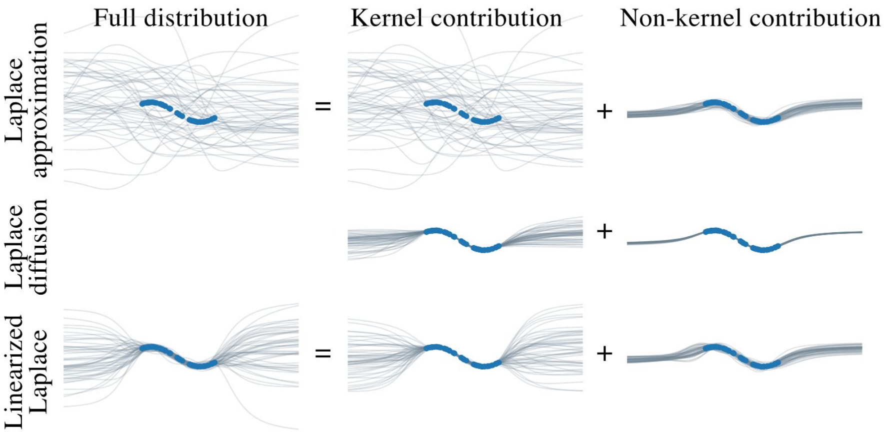
0. Effective-parameter is only pseudo-Riemannian $(\mathbb{R}^D, \mathbf{GGN_w})$
1. Laplace Approximation $(\mathbb{R}^D, \mathbf{GGN_w} + \alpha\mathbf{I})$
2. Kernel manifold ("never underfits") $(\mathcal{P}^{\perp}_{\mathbf{w}}, \alpha\mathbf{I})$

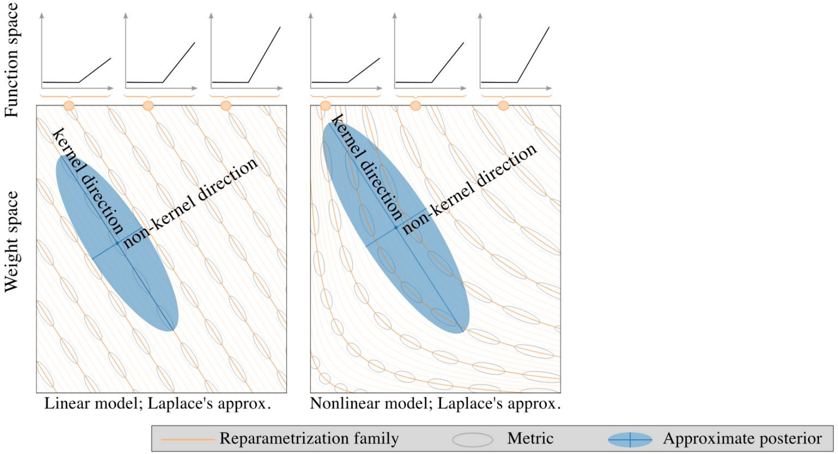




Section 4: Diffusion on various manifolds

0. Effective-parameter is only pseudo-Riemannian $(\mathbb{R}^D, \mathrm{GGN}_{\mathbf{w}})$
1. Laplace Approximation $(\mathbb{R}^D, \mathbf{GGN}_{\mathbf{w}} + \alpha\mathbf{I})$
2. Kernel manifold ("never underfits") $(\mathcal{P}_{\mathbf{w}}^{\perp}, \alpha\mathbf{I})$
3. Non-kernel-parameter manifold, i.e. "Laplace Diffusion" $(\mathcal{P}_{\mathbf{w}}, \mathrm{GGN}_{\mathbf{w}}^{+})$

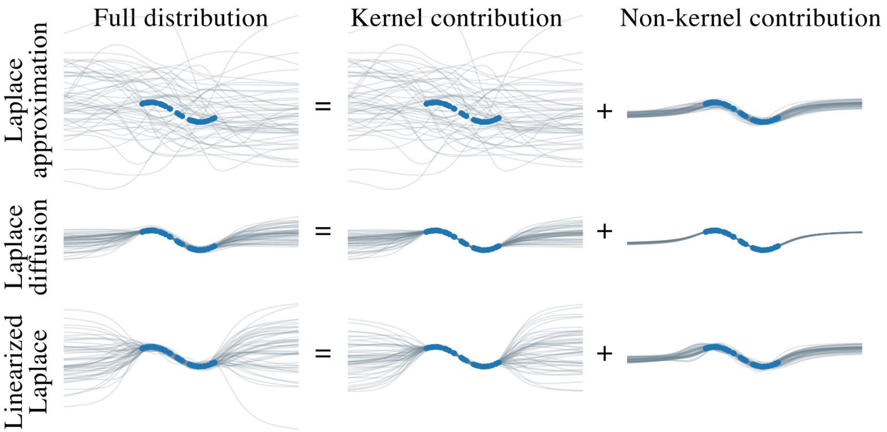


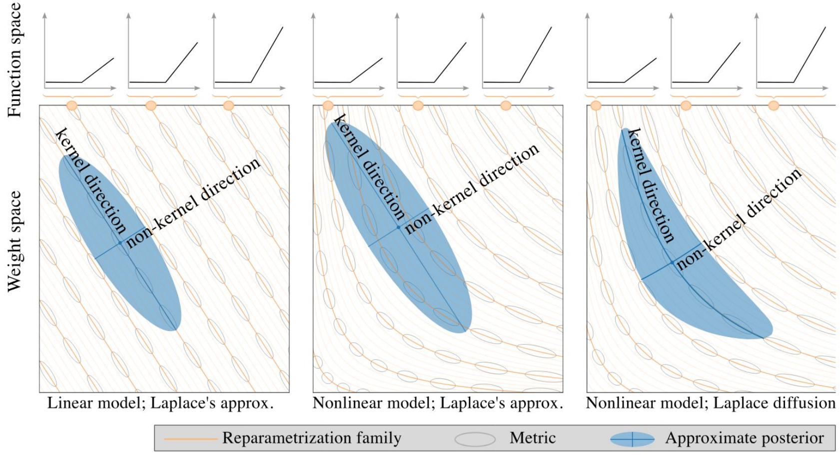


Section 4: Diffusion on various manifolds

0. Effective-parameter is only pseudo-Riemannian $(\mathbb{R}^D, \mathrm{GGN}_{\mathbf{w}})$
1. Laplace Approximation $(\mathbb{R}^D, \mathbf{GGN}_{\mathbf{w}} + \alpha\mathbf{I})$
2. Kernel manifold ("never underfits") $(\mathcal{P}_{\mathbf{w}}^{\perp}, \alpha\mathbf{I})$
3. Non-kernel-parameter manifold, i.e. "Laplace Diffusion" $(\mathcal{P}_{\mathbf{w}}, \mathrm{GGN}_{\mathbf{w}}^{+})$

4. Alternate between the two





Results

Table 1: In-distribution performance across methods trained on MNIST, FMNIST and CIFAR-10.

		Conf. (↑)	NLL (↓)	Acc. (↑)	Brier (↓)	ECE (↓)	MCE (↓)
MNIST	Laplace Diffusion (ours)	**0.988±0.001**	**0.042±0.007**	**0.987±0.002**	**0.022±0.003**	**0.137±0.019**	**0.775±0.043**
	Sampled Laplace	0.589±0.008	3.812±0.284	0.146±0.032	1.176±0.046	0.443±0.026	0.985±0.002
	Linearised Laplace	0.968±0.004	0.306±0.041	0.926±0.008	0.117±0.012	0.251±0.034	0.855±0.041
FMNIST	Laplace Diffusion (ours)	**0.900±0.001**	**0.001±0.000**	**0.906±0.007**	**0.141±0.006**	**0.108±0.015**	**0.729±0.092**
	Sampled Laplace	0.618±0.021	4.507±0.000	0.098±0.010	1.295±0.014	0.518±0.013	0.986±0.001
	Linearised Laplace	0.897±0.003	0.423±0.000	0.862±0.005	0.207±0.006	0.147±0.017	0.756±0.048
CIFAR-10	Laplace Diffusion (ours)	**0.952±0.007**	**0.345±0.062**	**0.905±0.007**	**0.155±0.019**	0.259±0.008	0.870±0.021
	Sampled Laplace	0.843±0.004	0.997±0.222	0.717±0.049	0.422±0.081	**0.221±0.047**	0.804±0.080
	Linearised Laplace	0.951±0.007	0.614±0.020	0.863±0.001	0.222±0.002	0.337±0.022	**0.789±0.035**

Results

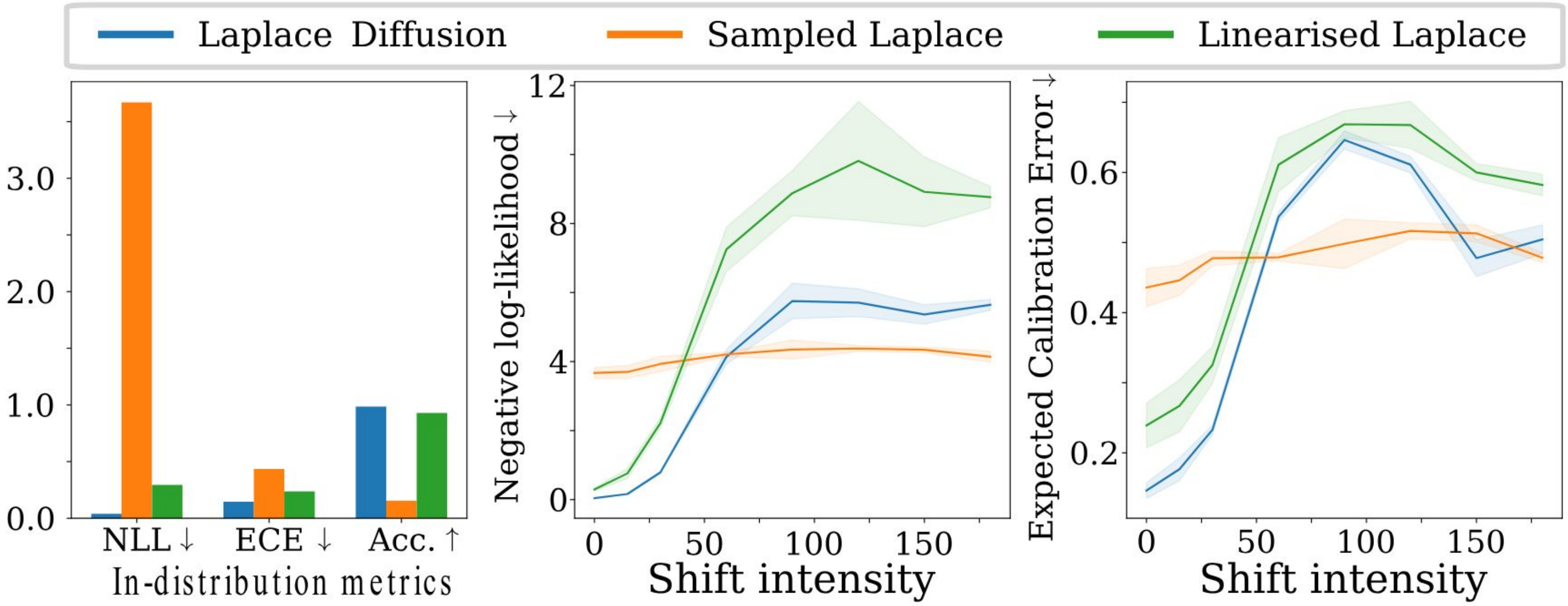
Table 1: In-distribution performance across methods trained on MNIST, FMNIST and CIFAR-10.

		Conf. (↑)	NLL (↓)	Acc. (↑)	Brier (↓)	ECE (↓)	MCE (↓)
MNIST	Laplace Diffusion (ours)	**0.988±0.001**	**0.042±0.007**	**0.987±0.002**	**0.022±0.003**	**0.137±0.019**	**0.775±0.043**
	Sampled Laplace	0.589±0.008	3.812±0.284	0.146±0.032	1.176±0.046	0.443±0.026	0.985±0.002
	Linearised Laplace	0.968±0.004	0.306±0.041	0.926±0.008	0.117±0.012	0.251±0.034	0.855±0.041
FMNIST	Laplace Diffusion (ours)	**0.900±0.001**	**0.001±0.000**	**0.906±0.007**	**0.141±0.006**	**0.108±0.015**	**0.729±0.092**
	Sampled Laplace	0.618±0.021	4.507±0.000	0.098±0.010	1.295±0.014	0.518±0.013	0.986±0.001
	Linearised Laplace	0.897±0.003	0.423±0.000	0.862±0.005	0.207±0.006	0.147±0.017	0.756±0.048
CIFAR-10	Laplace Diffusion (ours)	**0.952±0.007**	**0.345±0.062**	**0.905±0.007**	**0.155±0.019**	0.259±0.008	0.870±0.021
	Sampled Laplace	0.843±0.004	0.997±0.222	0.717±0.049	0.422±0.081	**0.221±0.047**	0.804±0.080
	Linearised Laplace	0.951±0.007	0.614±0.020	0.863±0.001	0.222±0.002	0.337±0.022	**0.789±0.035**

Table 2: Out-of-distribution AUROC (↑) performance for MNIST, FMNIST and CIFAR-10.

Trained on	MNIST			FMNIST			CIFAR-10	
Tested on	FMNIST	EMNIST	KMNIST	MNIST	EMNIST	KMNIST	CIFAR-100	SVHN
Laplace Diffusion (ours)	**0.909±0.033**	**0.625±0.018**	**0.929±0.008**	**0.759±0.045**	**0.741±0.010**	**0.749±0.023**	**0.851±0.002**	**0.862±0.010**
Sampled Laplace	0.500±0.026	0.494±0.006	0.482±0.013	0.495±0.037	0.503±0.036	0.493±0.033	0.687±0.033	0.599±0.038
Linearised Laplace	0.758±0.070	0.602±0.027	0.790±0.018	0.625±0.050	0.628±0.013	0.624±0.020	0.837±0.006	0.854±0.024

Rotated MNIST (shift = rotation angle)



Summary

- Linearised Laplace works better than full Laplace because it implicitly ignores the reparameterisation issues of BNNs

Summary

- Linearised Laplace works better than full Laplace because it implicitly ignores the reparameterisation issues of BNNs
- Their new method (*Laplace diffusion*) also tries to remove those reparameterisation issues (but is significantly more complicated)

Summary

- Linearised Laplace works better than full Laplace because it implicitly ignores the reparameterisation issues of BNNs

- Their new method (*Laplace diffusion*) also tries to remove those reparameterisation issues (but is significantly more complicated)

- In practice, we often use an approximation of the GGN (e.g. KFAC), which would break the motivation behind their new method

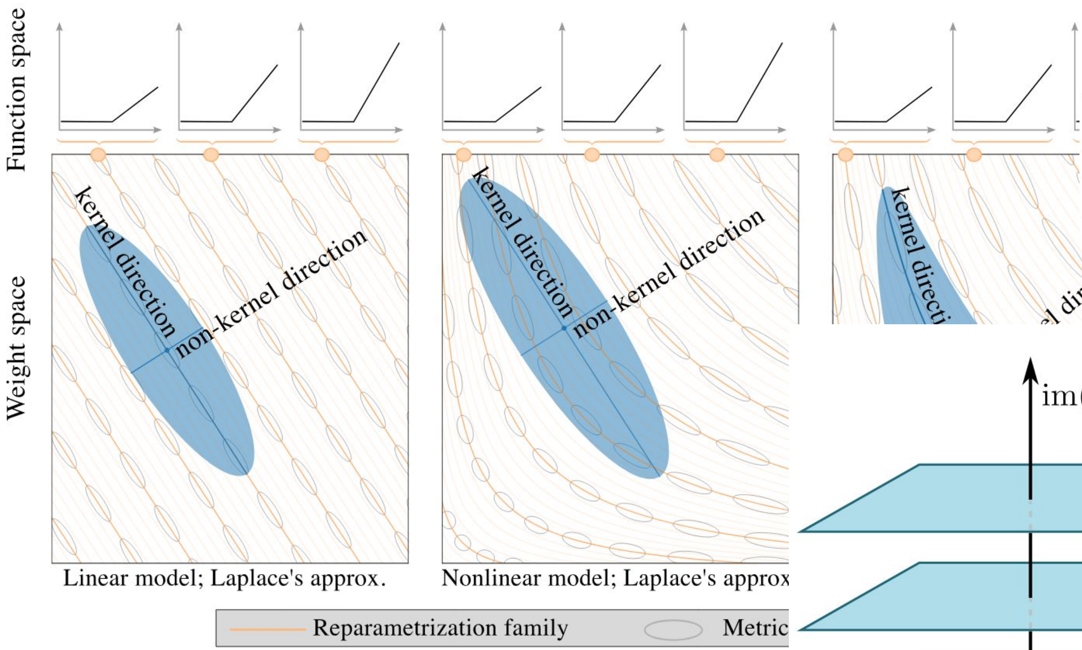
Links

Paper: https://arxiv.org/pdf/2406.03334

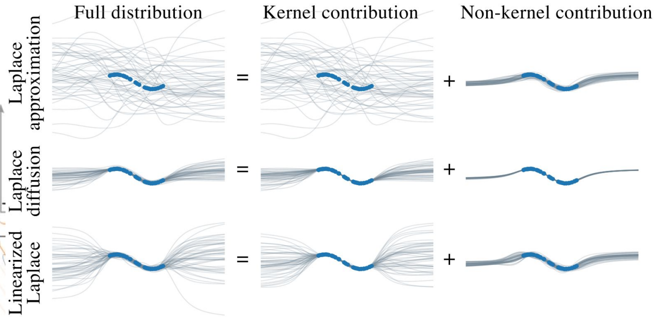
Slides from a talk by one of the authors:
https://www2.compute.dtu.dk/~sohau//talks/2024_MTNS/

Immer et al. (2021): “ Improving predictions of Bayesian neural nets via local linearization,” Alexander Immer, Maciej Korzepa, Matthias Bauer. Proceedings of The 24th International Conference on Artificial Intelligence and Statistics, PMLR 130:703-711, 2021. https://proceedings.mlr.press/v130/immer21a.html

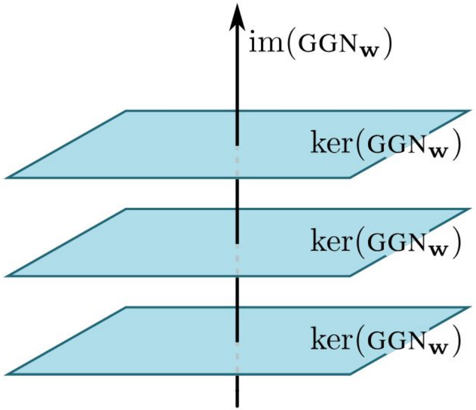
pics



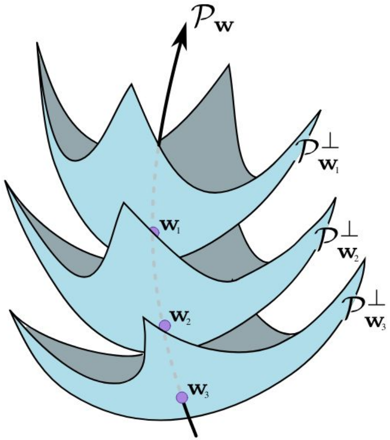
Function space
Weight space
kernel direction
non-kernel direction
kernel direction
non-kernel direction
kernel directi
l direction
Linear model; Laplace's approx.
Nonlinear model; Laplace's approx
Reparametrization family
Metric



Full distribution
Kernel contribution
Non-kernel contribution
Laplace approximation
Laplace diffusion
Linearized Laplace
=
+

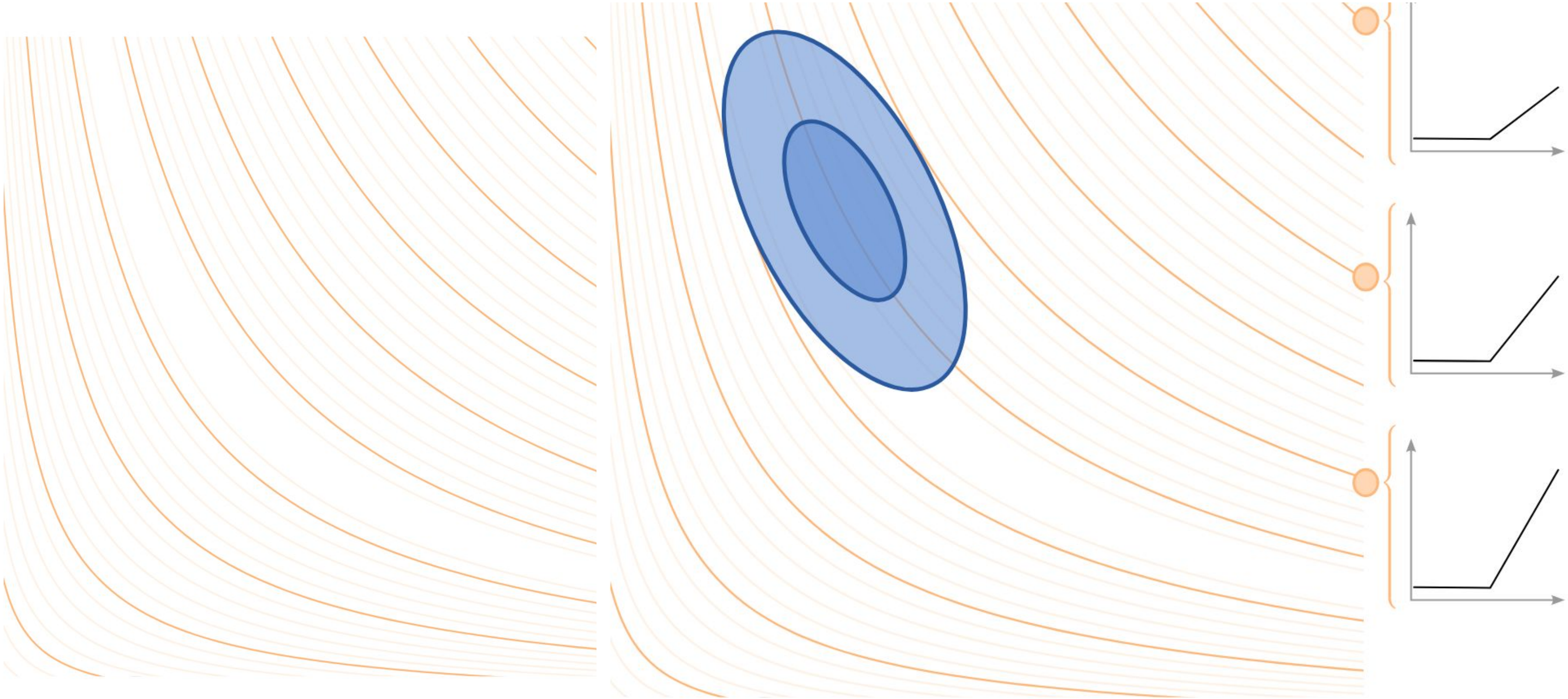


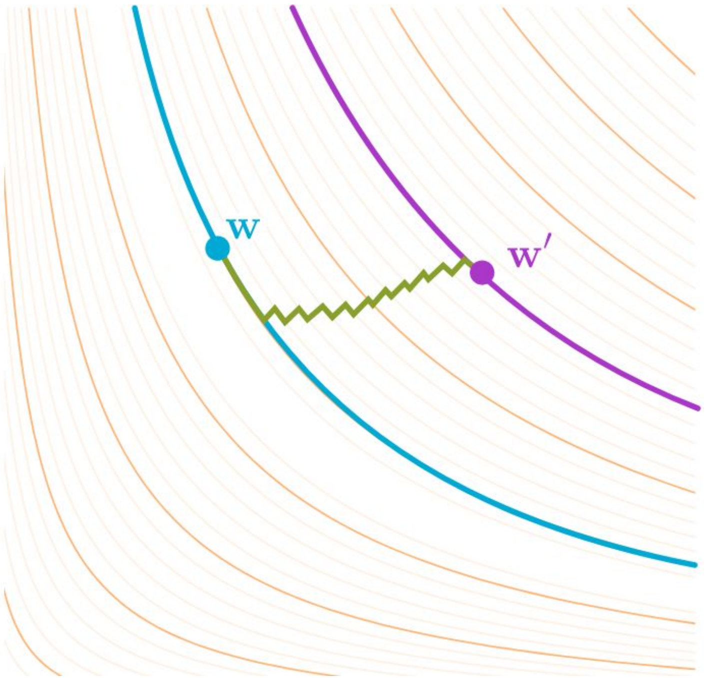
$\mathrm{im}(\mathrm{GGN}_{\mathbf{w}})$
$\ker(\mathrm{GGN}_{\mathbf{w}})$
$\ker(\mathrm{GGN}_{\mathbf{w}})$
$\ker(\mathrm{GGN}_{\mathbf{w}})$

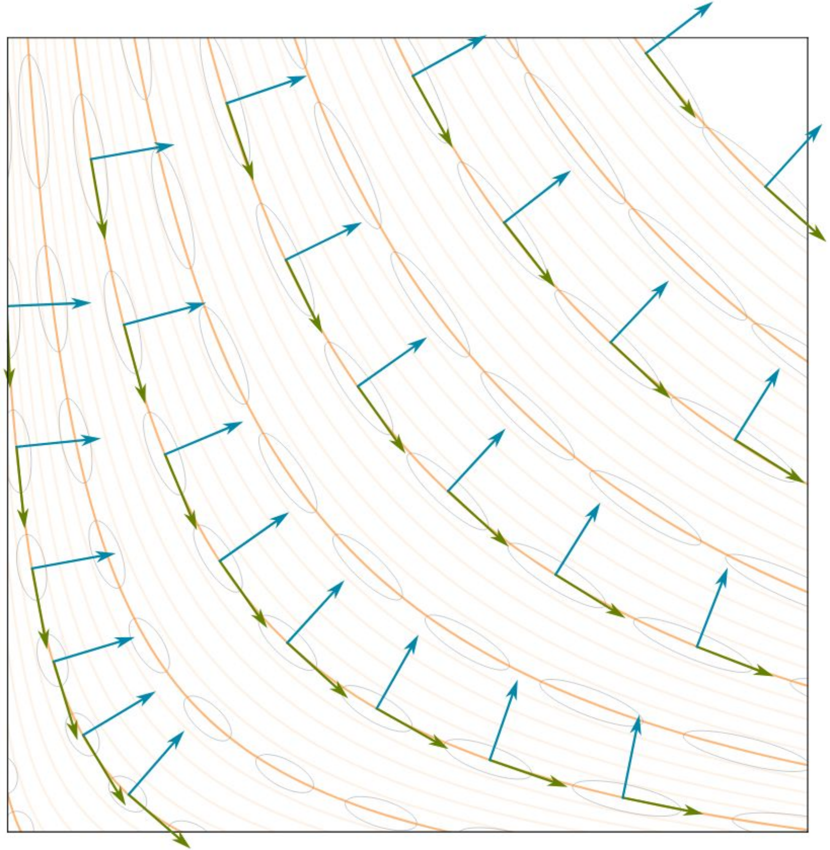


$\mathcal{P}_{\mathbf{w}}$
$\mathcal{P}^{\perp}_{\mathbf{w}_1}$
$\mathcal{P}^{\perp}_{\mathbf{w}_2}$
$\mathcal{P}^{\perp}_{\mathbf{w}_3}$
$\mathbf{w}_1$
$\mathbf{w}_2$
$\mathbf{w}_3$

From slides




w
w′



Section 4: They do diffusion

$$(\mathrm{M}, \mathbf{G})$$

$$\mathrm{d}\mathbf{w} = \sqrt{2\tau}\mathbf{G}(\mathbf{w})^{-\frac{1}{2}}\mathrm{d}W + \tau\Gamma\mathrm{d}t \qquad \text{where} \quad \Gamma_i(\mathbf{w}) = \sum_{j=1}^{D} \frac{\partial}{\partial \mathbf{w}_j}(\mathbf{G}(\mathbf{w})^{-1})_{i}$$

$$\mathbf{w}_{t+1} \;=\; \mathbf{w}_t + \sqrt{2h_t}\mathbf{G}(\mathbf{w}_t)^{-\frac{1}{2}}\boldsymbol{\epsilon}, \text{ where } \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$