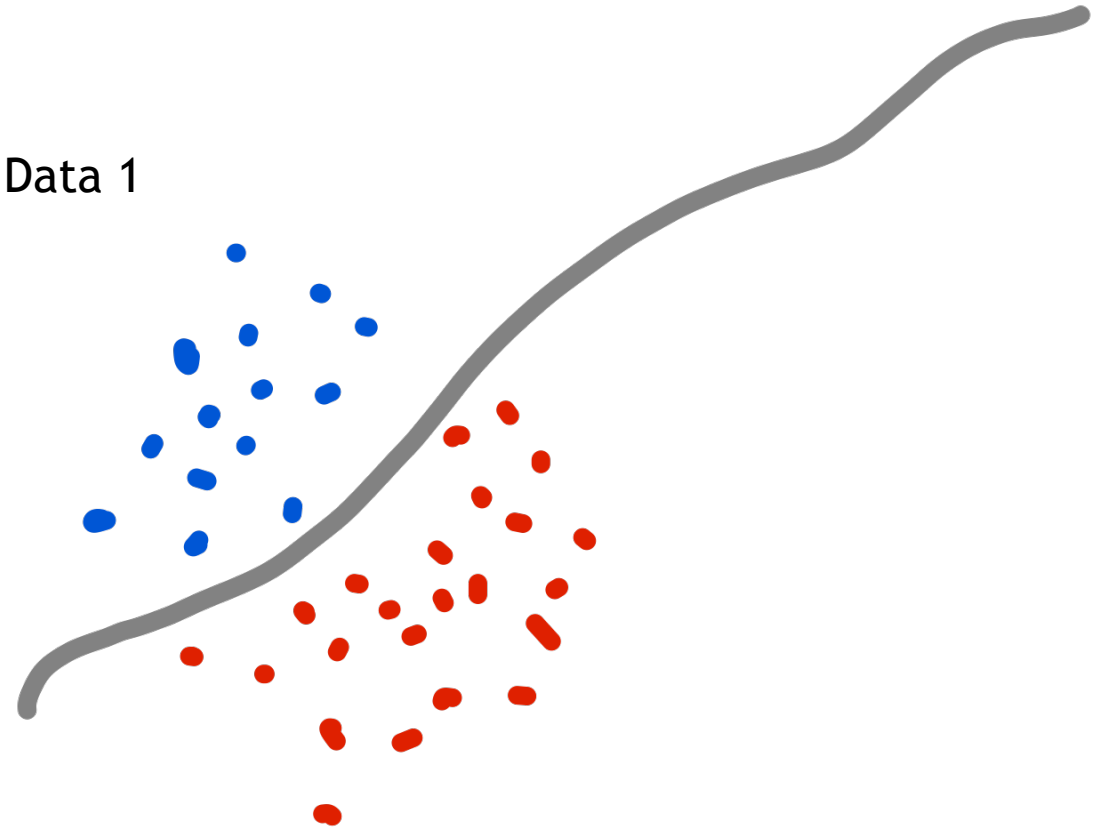
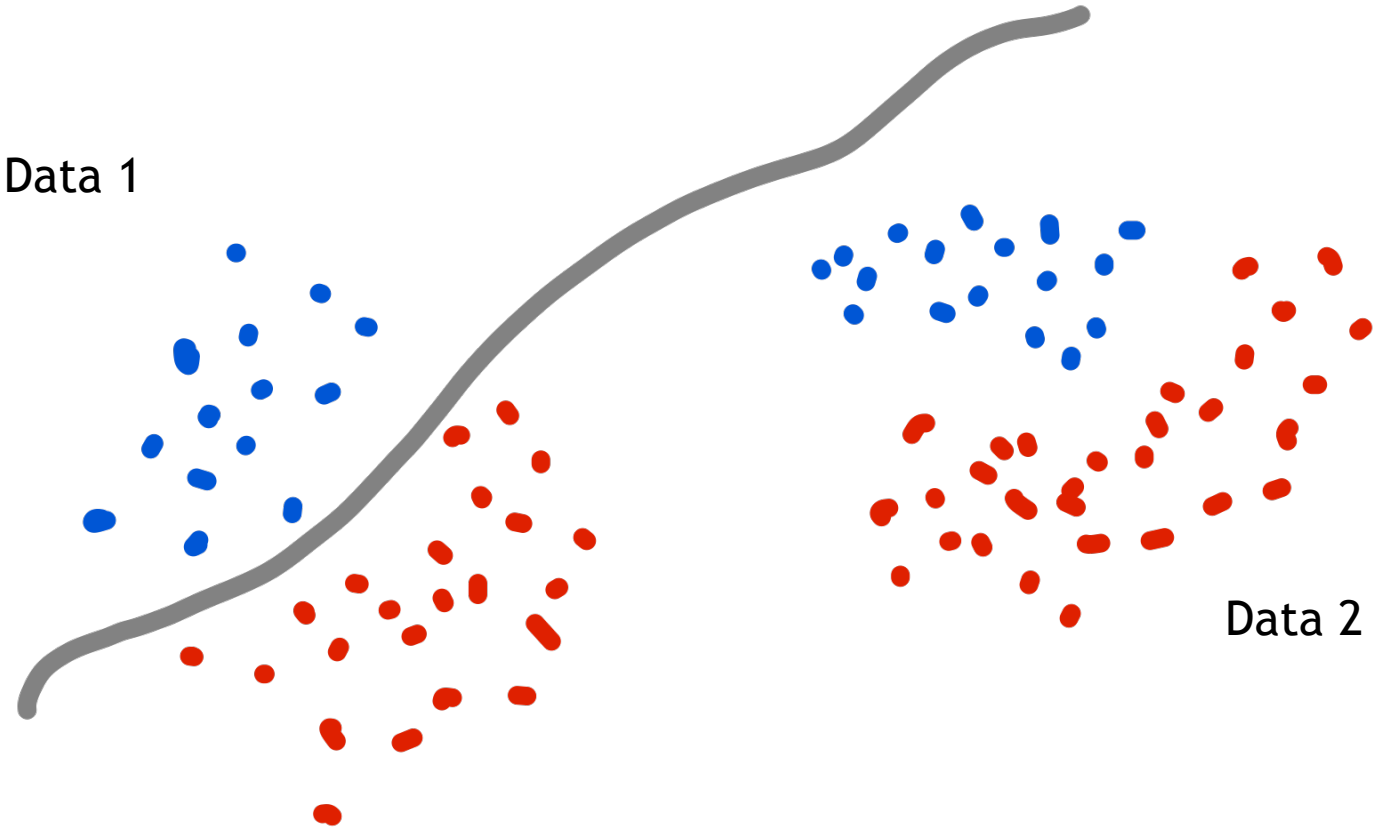
DNN2GP: From Deep Networks to Gaussian Processes

Mohammad Emtiyaz Khan
RIKEN Center for AI Project, Tokyo
http://emtiyaz.github.io

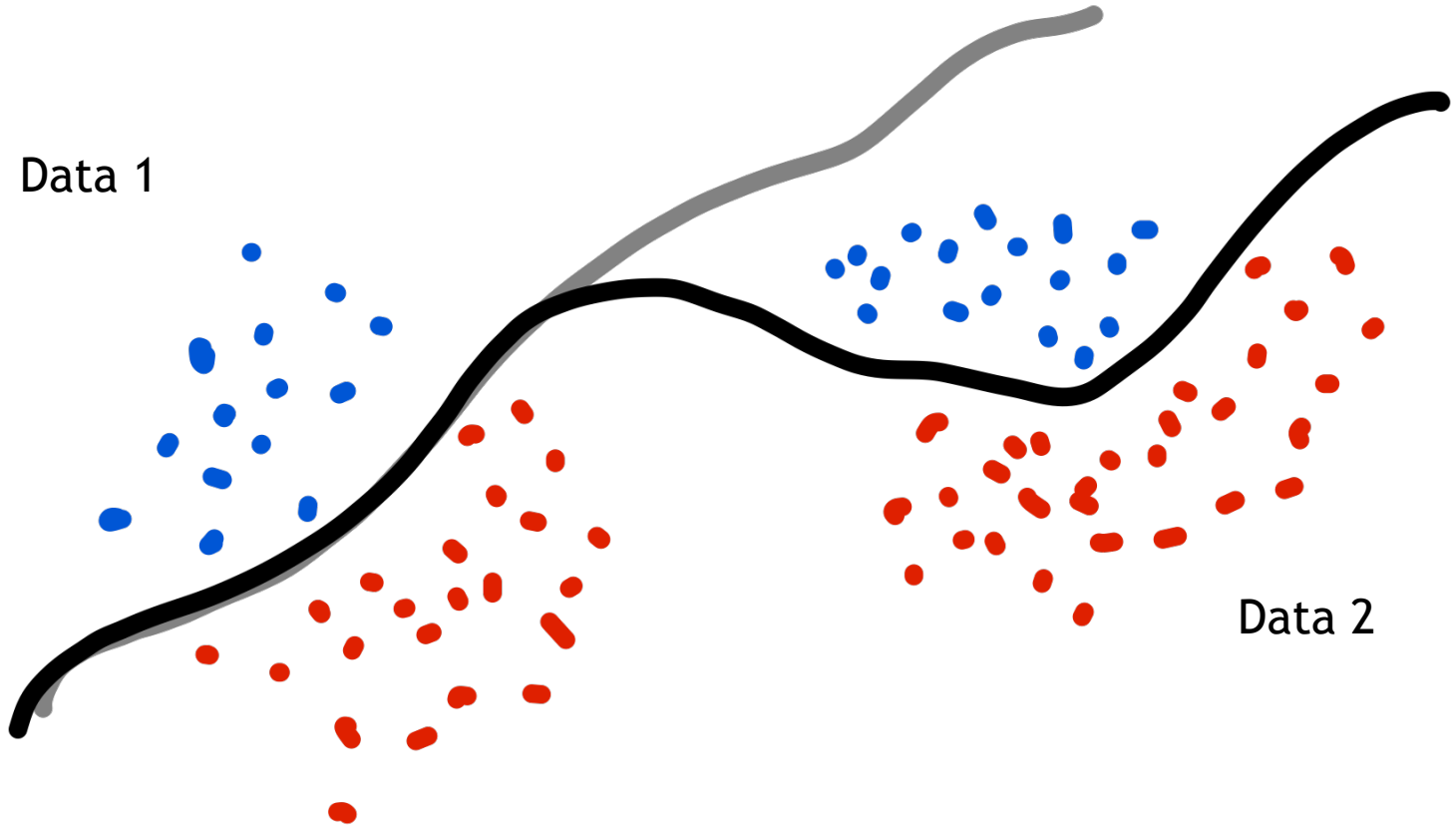
Motivation: Life-Long Learning



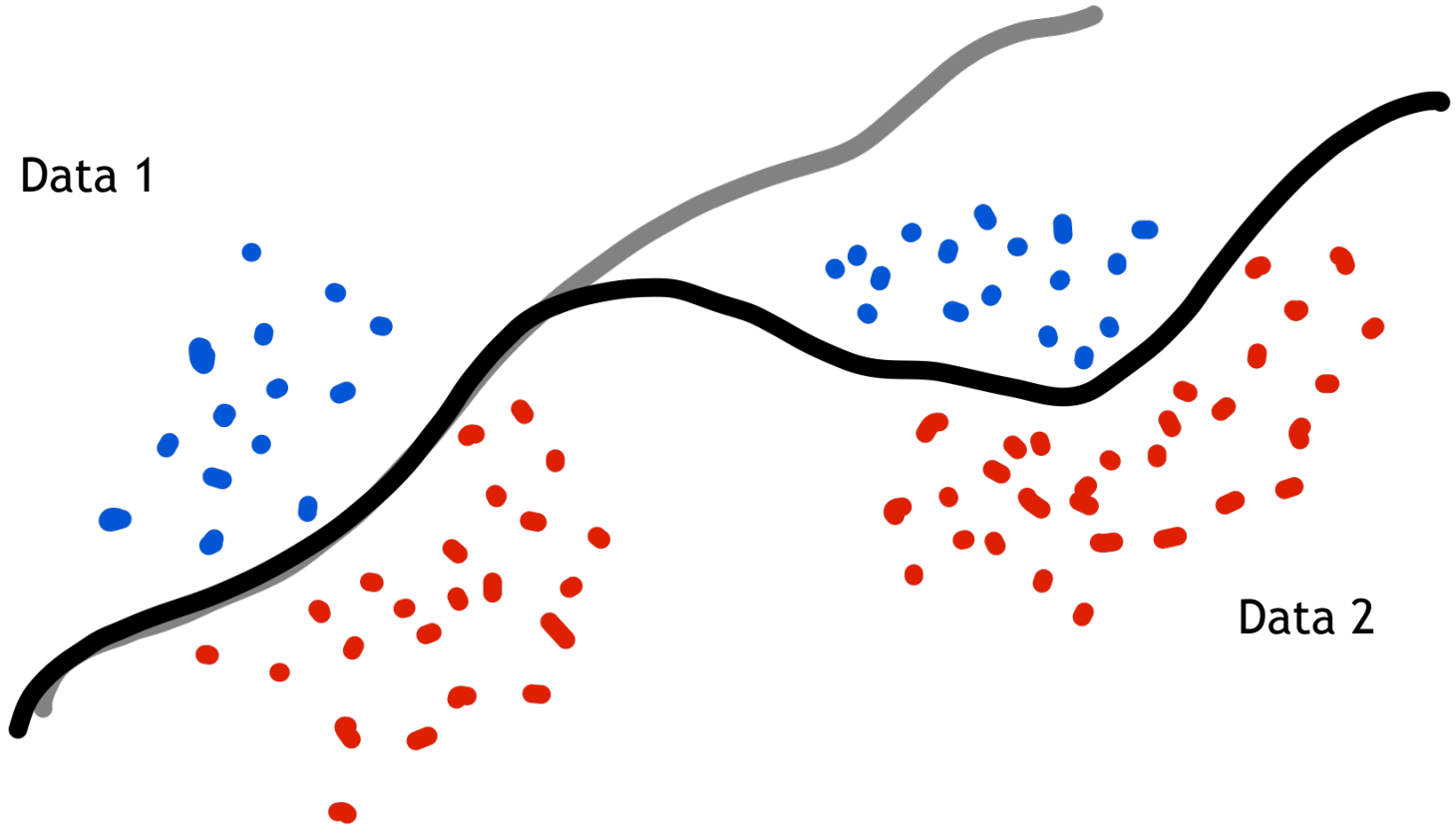
Motivation: Life-Long Learning



Motivation: Life-Long Learning



Motivation: Life-Long Learning



Change the network weights to match the network output (function) at Data 1 while classifying Data 2

Weight-Space vs Function-Space

Weight/parameter space

- Finite dimensional
- Easy to train
  - e.g., SGD is scalable
- Difficult to regularize
  - Actual value of the weights do not matter
  - Complex relationship between network outputs and weights

Function/data space

- Infinite dimensional
- Difficult to train
  - e.g. GP reg is not
- Easy to regularize
  - Properties of the network outputs are easier to specify and interpret.

This Talk

- Connections between the weight and function spaces.
  - Cheap algorithms to train in the weight space while regularizing in the function space.
- Background
  - Linear models and GPs
  - Neural Nets and GPs (requires infinite-width nets)
- DNN2GP
  - Convert trained finite-widths nets to GPs
  - Convert the iterates of DL algorithms to GPs
- Applications to Continual Learning

Linear model and GPs

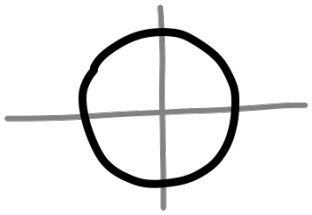
Gaussian prior on weights induces GP prior on functions

Chapter 2 in Rasmussen and Williams, Gaussian processes for Machine Learning, 2006

Linear model and GPs

Gaussian prior on weights induces GP prior on functions

$$w \sim N(0, I)$$



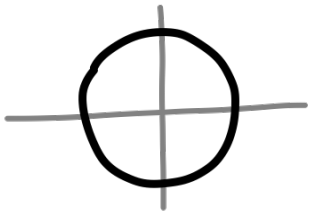
$$\text{function} \quad f(x) = \phi(x)^T w \quad \text{feature}, \ \text{weights}$$

Chapter 2 in Rasmussen and Williams, Gaussian processes for Machine Learning, 2006

Linear model and GPs

Gaussian prior on weights induces GP prior on functions

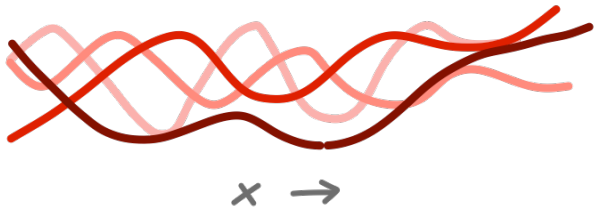
$w \sim N(0, I)$



function $f(x) = \phi(x)^T w$ feature weights

$f(x) \sim GP(0, \phi(x)^T \phi(x'))$

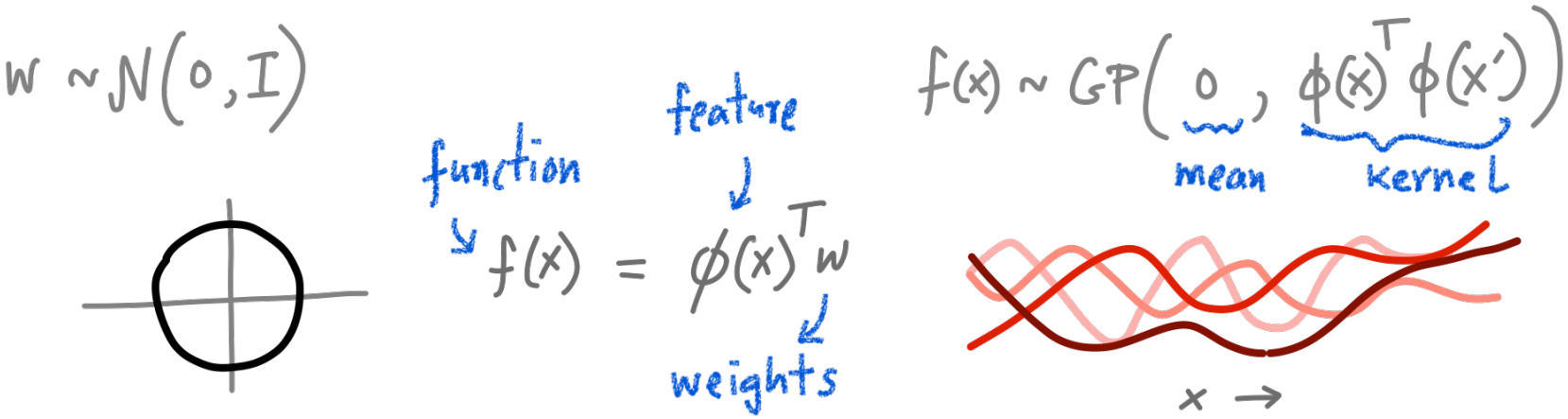
mean kernel



Chapter 2 in Rasmussen and Williams, Gaussian processes for Machine Learning, 2006

Linear model and GPs

Gaussian prior on weights induces GP prior on functions

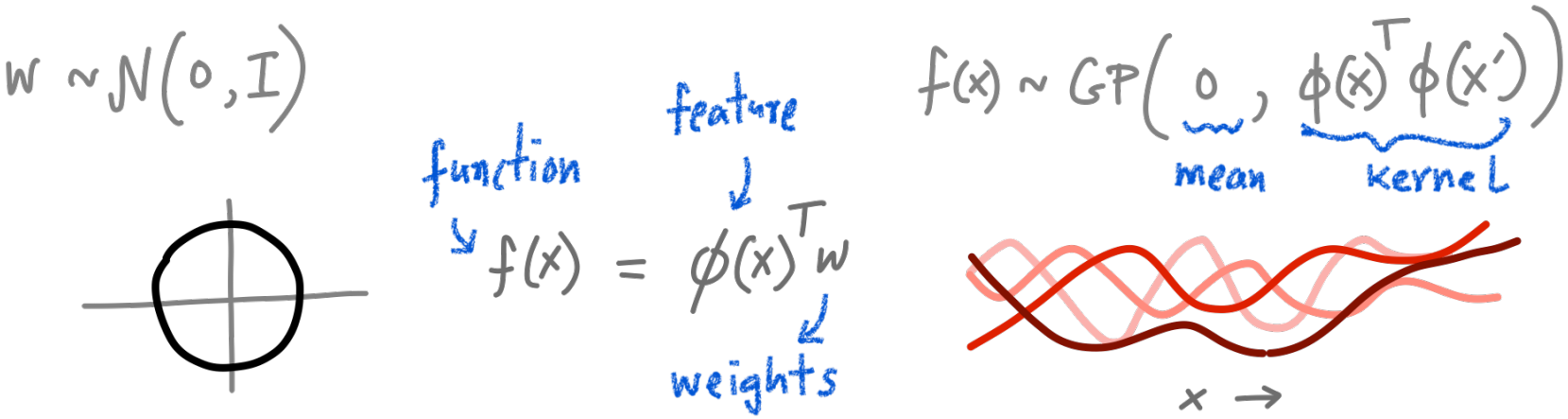


Gaussian posterior on w induces a GP posterior on f

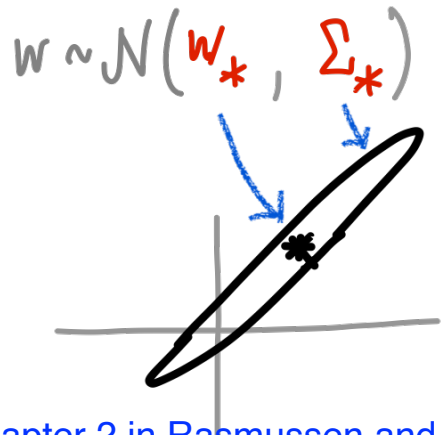
Chapter 2 in Rasmussen and Williams, Gaussian processes for Machine Learning, 2006

Linear model and GPs

Gaussian prior on weights induces GP prior on functions



Gaussian posterior on w induces a GP posterior on f

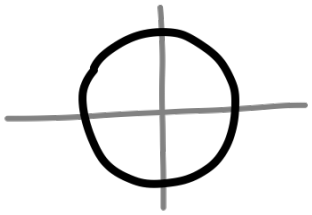


Chapter 2 in Rasmussen and Williams, Gaussian processes for Machine Learning, 2006

Linear model and GPs

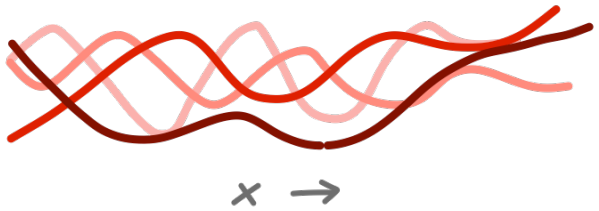
Gaussian prior on weights induces GP prior on functions

$$w \sim \mathcal{N}(0, I)$$



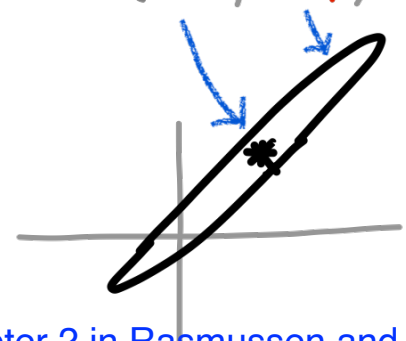
$$\text{function} \quad f(x) = \phi(x)^T w \quad \text{feature} \quad \text{weights}$$

$$f(x) \sim GP\left(\underbrace{0}_{\text{mean}}, \underbrace{\phi(x)^T \phi(x')}_{\text{kernel}}\right)$$

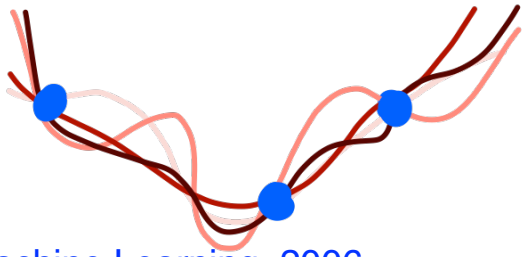


Gaussian posterior on w induces a GP posterior on f

$$w \sim \mathcal{N}(w_*, \Sigma_*)$$



$$f(x) \sim GP\left(\phi(x)^T w_*, \phi(x)^T \Sigma_* \phi(x')\right)$$



Chapter 2 in Rasmussen and Williams, Gaussian processes for Machine Learning, 2006

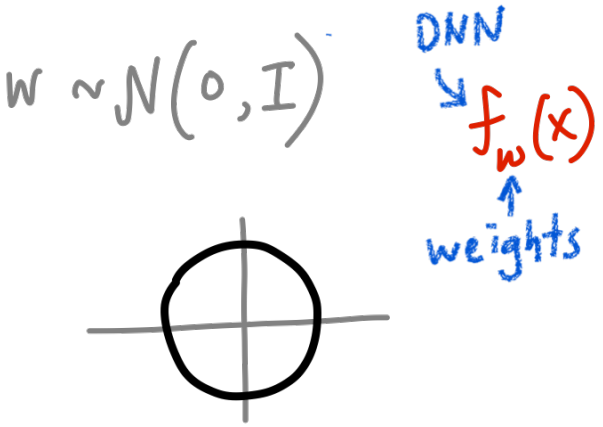
Deep Networks and GPs

Gaussian prior on weights induces GP prior on functions

Lee et al., Deep Neural Networks as Gaussian Processes, ICLR 2018 (and many more…)

Deep Networks and GPs

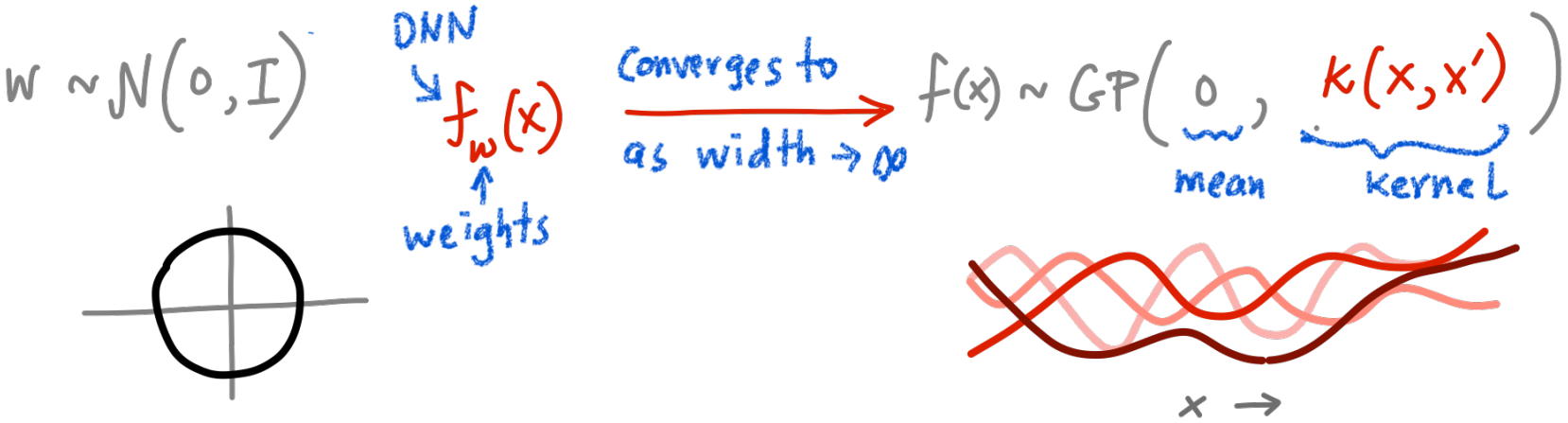
Gaussian prior on weights induces GP prior on functions



Lee et al., Deep Neural Networks as Gaussian Processes, ICLR 2018 (and many more…)

Deep Networks and GPs

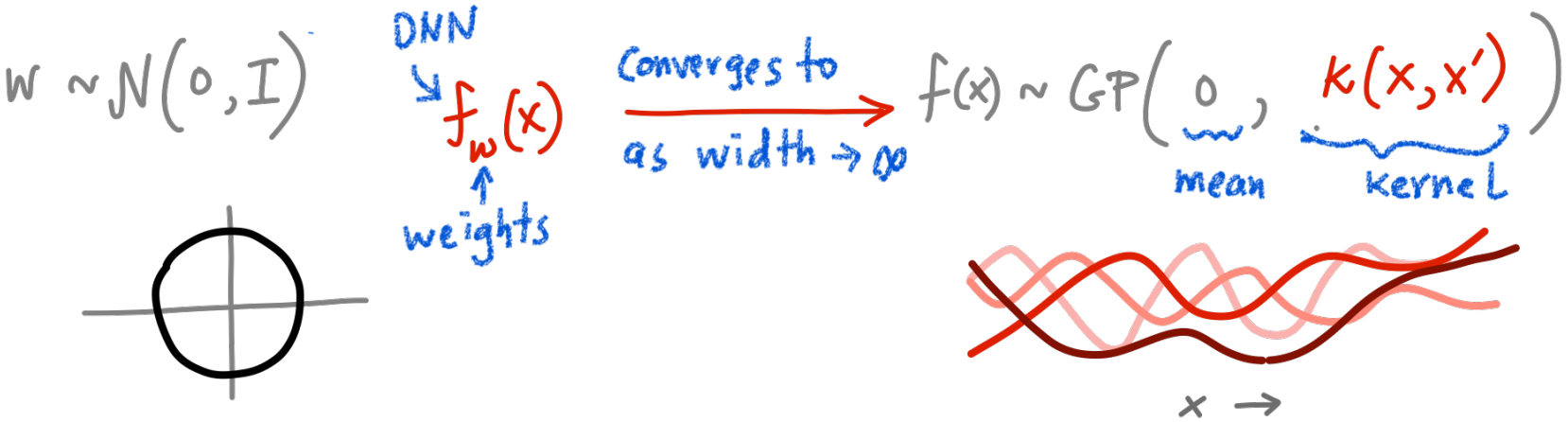
Gaussian prior on weights induces GP prior on functions



Lee et al., Deep Neural Networks as Gaussian Processes, ICLR 2018 (and many more…)

Deep Networks and GPs

Gaussian prior on weights induces GP prior on functions

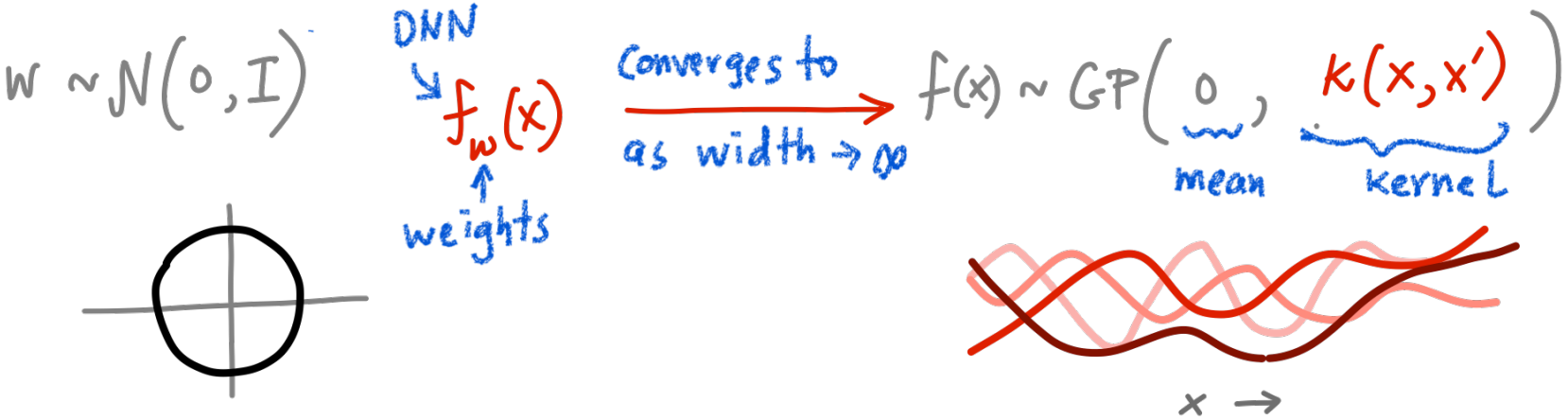


Q: Does this hold at finite width? And for posteriors?

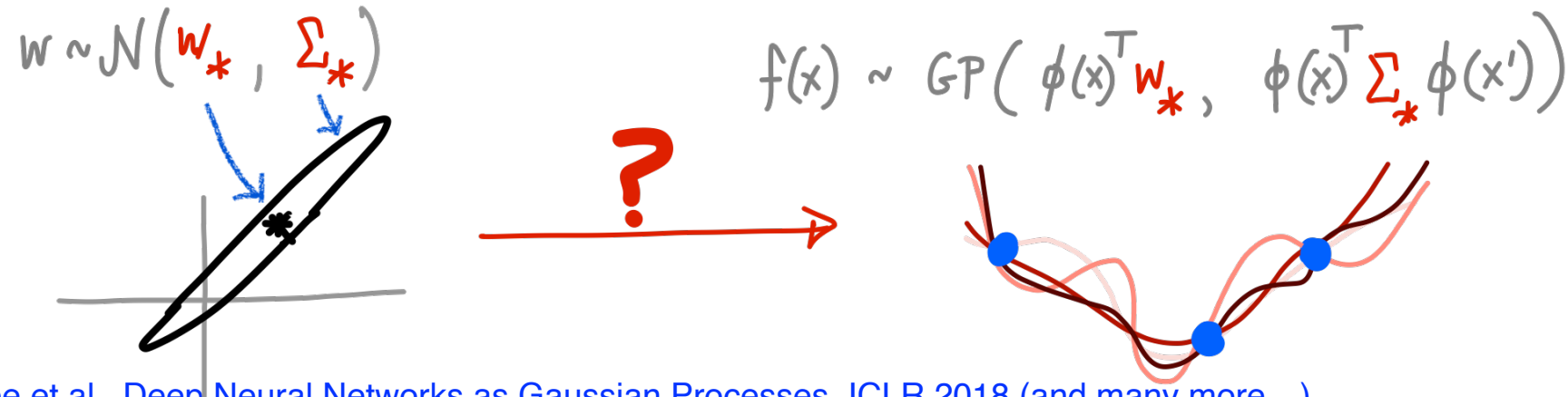
Lee et al., Deep Neural Networks as Gaussian Processes, ICLR 2018 (and many more…)

Deep Networks and GPs

Gaussian prior on weights induces GP prior on functions



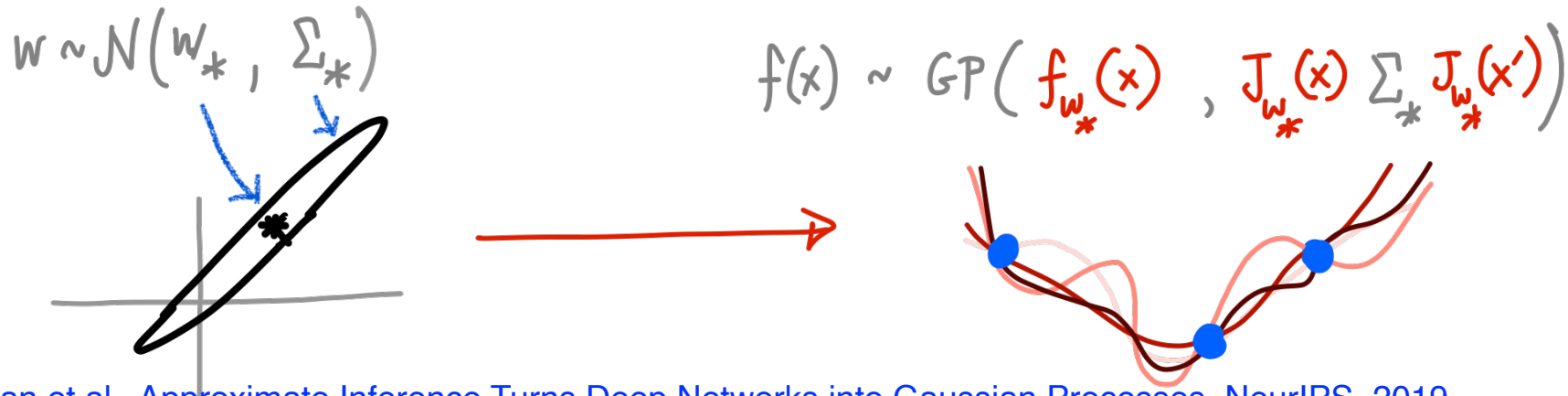
Q: Does this hold at finite width? And for posteriors?



Lee et al., Deep Neural Networks as Gaussian Processes, ICLR 2018 (and many more…)

DNN2GP for regression

Using DNN2GP, we can convert a trained network into GP



Khan et al., Approximate Inference Turns Deep Networks into Gaussian Processes, NeurIPS, 2019

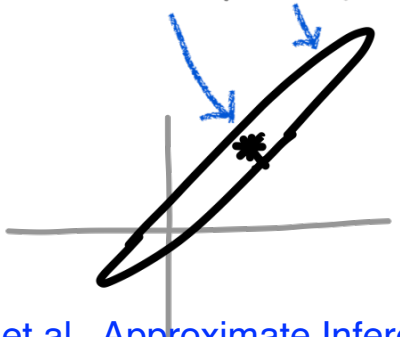
DNN2GP for regression

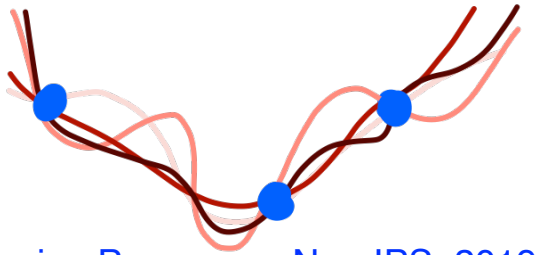
Using DNN2GP, we can convert a trained network into GP

$$w_* = \arg\min_w \sum_{i=1}^{N} \underbrace{\left(y_i - f_w(x_i)\right)^2}_{\text{squared loss}} + \underbrace{\delta\, w^T w}_{L_2 \text{ prior}}$$

$$w \sim \mathcal{N}(w_*, \Sigma_*)$$

$$f(x) \sim GP\left(f_{w_*}(x),\ J_{w_*}(x)\, \Sigma_*\, J_{w_*}(x')\right)$$





Khan et al., Approximate Inference Turns Deep Networks into Gaussian Processes, NeurIPS, 2019

DNN2GP for regression

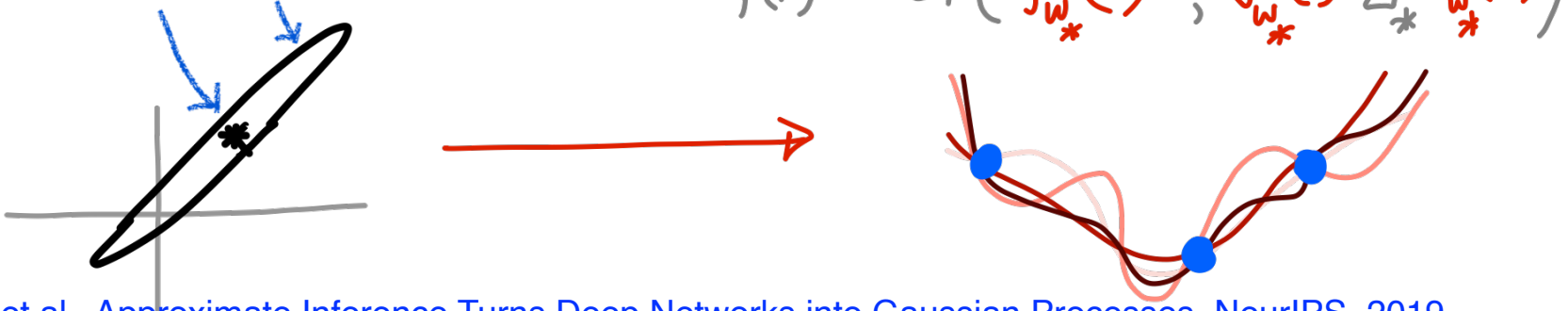
Using DNN2GP, we can convert a trained network into GP

$$w_* = \arg\min_w \sum_{i=1}^{N} \underbrace{\left(y_i - f_w(x_i)\right)^2}_{\text{squared loss}} + \underbrace{\delta\, w^T w}_{L_2 \text{ prior}}$$

$$\Sigma_*^{-1} = \sum_{i=1}^{N} \nabla_w f_{w_*}(x_i)\, \underbrace{\nabla_w f_{w_*}(x_i)^T}_{J_{w_*}(x_i)^T \leftarrow \text{Jacobian}} + \delta I \quad \leftarrow \text{Gauss-Newton Curvature}$$

$$w \sim \mathcal{N}(w_*, \Sigma_*)$$

$$f(x) \sim GP\left(f_{w_*}(x),\ J_{w_*}(x)\, \Sigma_*\, J_{w_*}(x')\right)$$



Khan et al., Approximate Inference Turns Deep Networks into Gaussian Processes, NeurIPS, 2019

DNN2GP Generalization

This generalizes to twice differentiable loss and priors

$$w_* = \arg\min_w \sum_{i=1}^{N} \underbrace{\ell(y_i, \sigma(f_w(x_i)))}_{\text{Diff. loss}} + \underbrace{\delta R(w)}_{\text{Convex prior}}$$

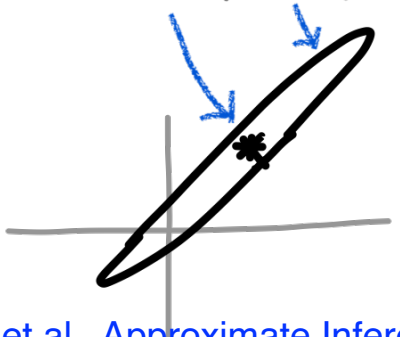
(Link function: σ)

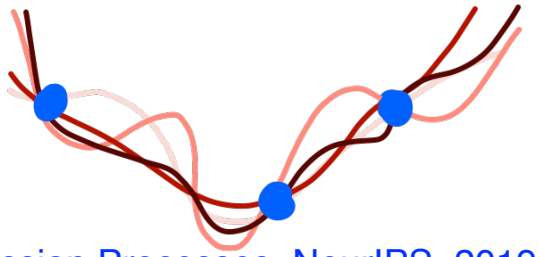
$$\Sigma_*^{-1} = \sum_{i=1}^{N} \nabla_w f_{w_*}(x_i) \underbrace{\nabla^2_{ff} \ell(\,)}_{\Lambda_{w_*}(x)} \nabla_w f_{w_*}(x_i)^T + \delta \nabla^2_{ww} R(w)$$

← Generalized Gauss-Newton Curvature

$$w \sim \mathcal{N}(w_*, \Sigma_*)$$

$$f(x) \sim \mathrm{GP}\left(\sigma\left(f_{w_*}(x)\right), \lambda^{(x)}_{w_*} J_{w_*}(x)\, \Sigma_*\, J_{w_*}(x') \lambda^{(x)}_{w_*}\right)$$





Khan et al., Approximate Inference Turns Deep Networks into Gaussian Processes, NeurIPS, 2019

Deep Learning as GP inference

Iterations of algorithms too can be written as GP inference

$$w_{t+1} \leftarrow w_t - \rho \underbrace{(S_t + \delta I)^{-1}}_{\Sigma_t} \left[\sum_{i \in M} g_i\right] \leftarrow \text{minibatch gradient}$$

$$\text{Scale matrix} \rightarrow S_{t+1} \leftarrow (1-\beta) S_t + \sum_{i \in M} J_{w_t}(x_i) J_{w_t}(x_i)^T \leftarrow \text{minibatch GN curvature}$$

Khan et al., Approximate Inference Turns Deep Networks into Gaussian Processes, NeurIPS, 2019

Deep Learning as GP inference

Iterations of algorithms too can be written as GP inference

$$w_{t+1} \leftarrow w_t - \rho \underbrace{(S_t + \delta I)^{-1}}_{\Sigma_t} \left[\sum_{i \in M} g_i\right] \leftarrow \text{minibatch gradient}$$

$$\text{Scale matrix} \rightarrow \quad S_{t+1} \leftarrow (1-\rho) S_t + \sum_{i \in M} J_{w_t}(x_i) J_{w_t}(x_i)^T \leftarrow \text{minibatch GN curvature}$$

Training in w space induces a sequence in f space

Khan et al., Approximate Inference Turns Deep Networks into Gaussian Processes, NeurIPS, 2019

Deep Learning as GP inference

Iterations of algorithms too can be written as GP inference

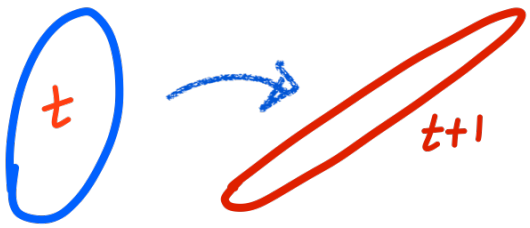
$$w_{t+1} \leftarrow w_t - \rho \underbrace{(S_t + \delta I)^{-1}}_{\Sigma_t} \left[\sum_{i \in M} g_i\right] \leftarrow \text{minibatch gradient}$$

$$\text{Scale matrix} \rightarrow \quad S_{t+1} \leftarrow (1-\rho) S_t + \sum_{i \in M} J_{w_t}(x_i) J_{w_t}(x_i)^T \leftarrow \text{minibatch GN curvature}$$

Training in w space induces a sequence in f space

$\mathcal{N}(w_t, \Sigma_t)$

$\mathcal{N}(w_{t+1}, \Sigma_{t+1})$



Khan et al., Approximate Inference Turns Deep Networks into Gaussian Processes, NeurIPS, 2019

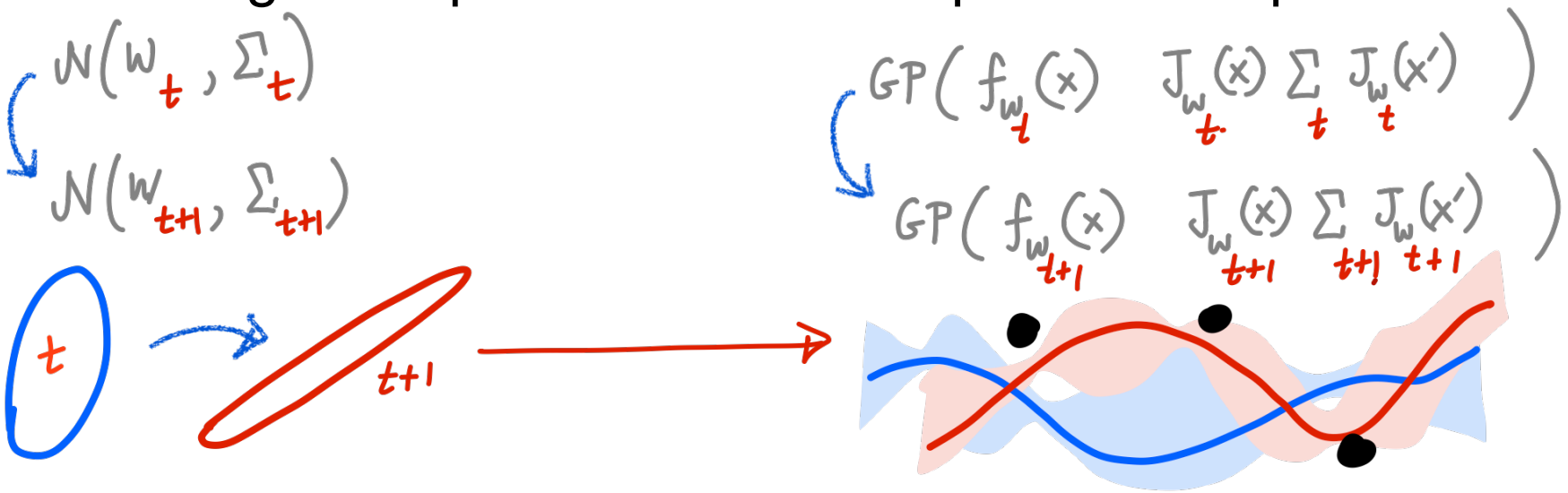
Deep Learning as GP inference

Iterations of algorithms too can be written as GP inference

$$w_{t+1} \leftarrow w_t - \rho \underbrace{(S_t + \delta I)^{-1}}_{\Sigma_t} \left[\sum_{i \in M} g_i\right] \leftarrow \text{minibatch gradient}$$

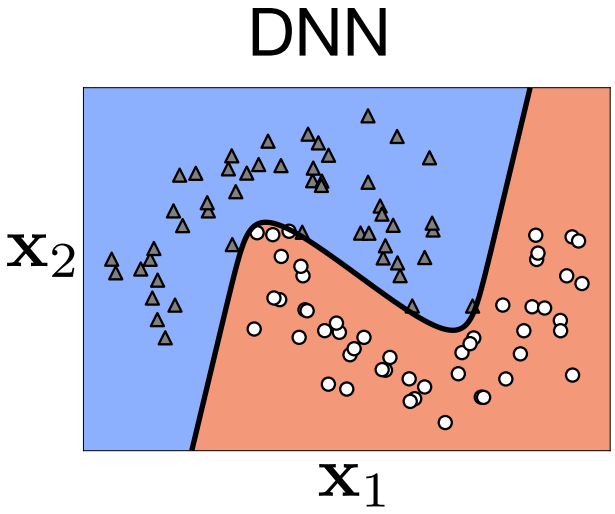
$$\text{Scale matrix} \rightarrow S_{t+1} \leftarrow (1-\rho) S_t + \sum_{i \in M} J_{w_t}(x_i) J_{w_t}(x_i)^T \leftarrow \text{minibatch GN curvature}$$

Training in w space induces a sequence in f space



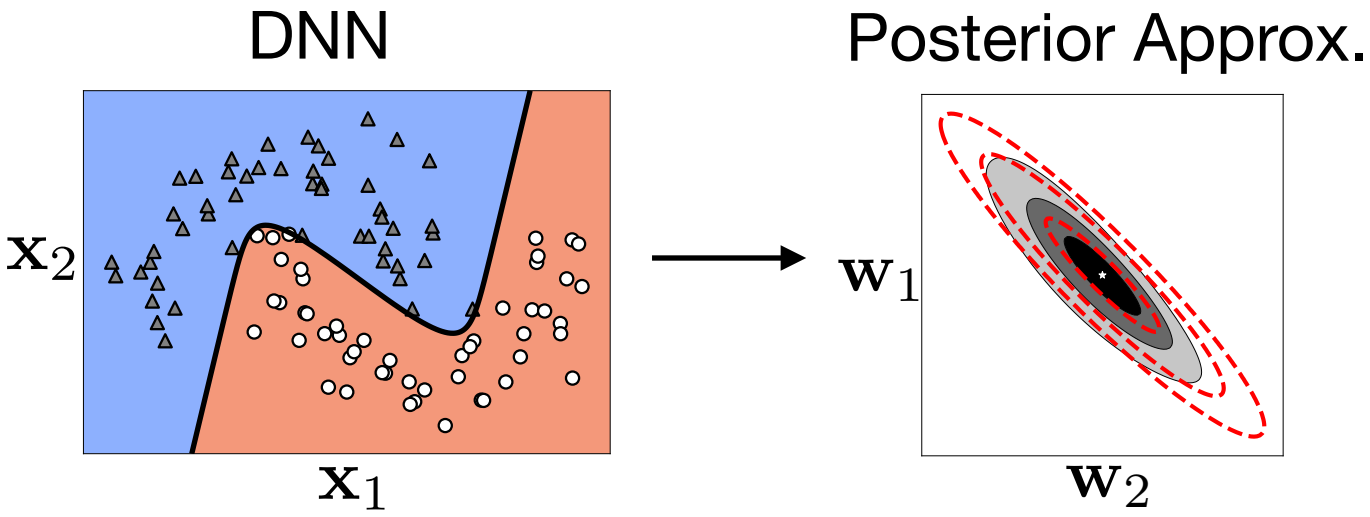
Khan et al., Approximate Inference Turns Deep Networks into Gaussian Processes, NeurIPS, 2019

Outline of the derivation



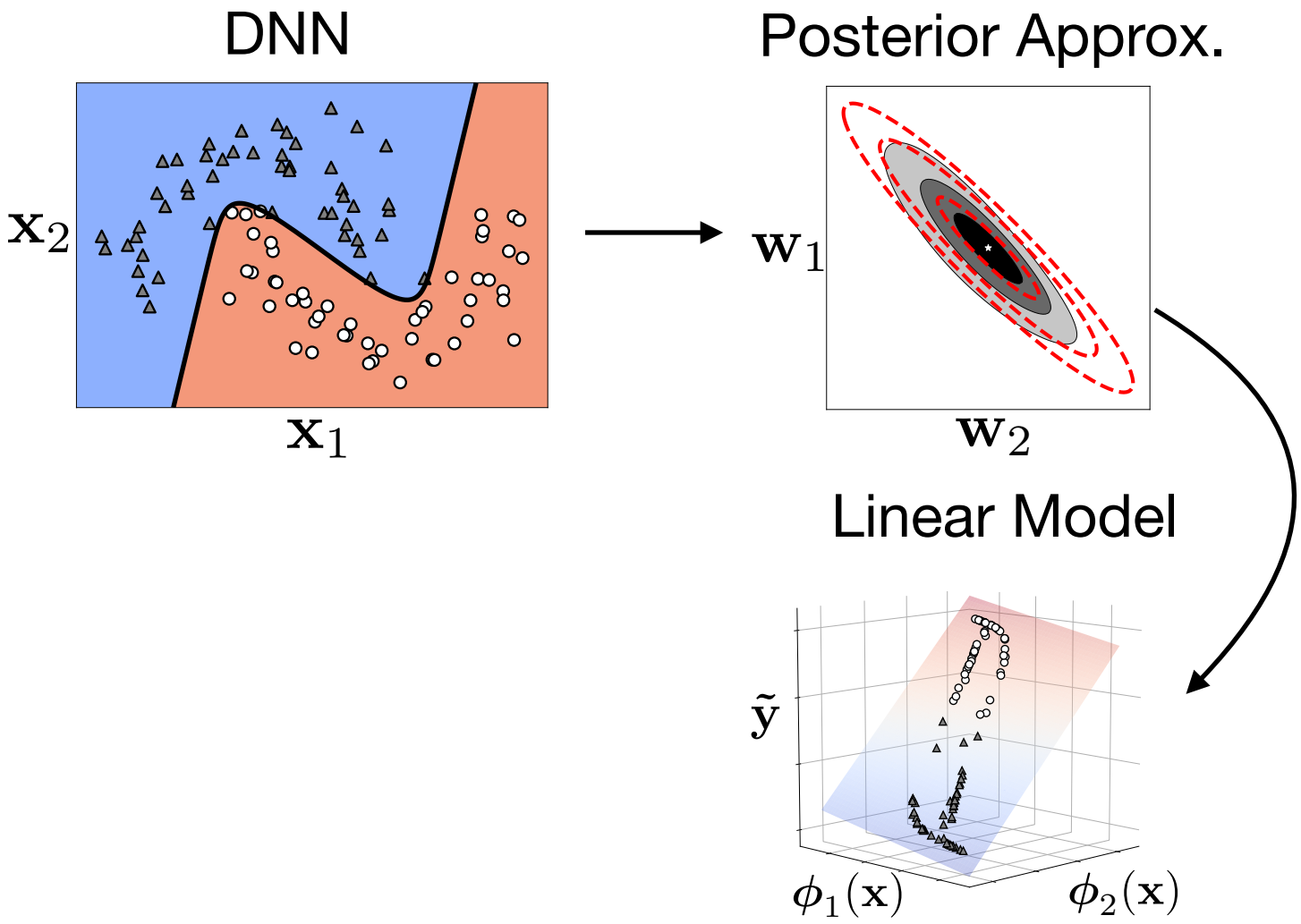
Khan et al., Approximate Inference Turns Deep Networks into Gaussian Processes, NeurIPS, 2019

Outline of the derivation



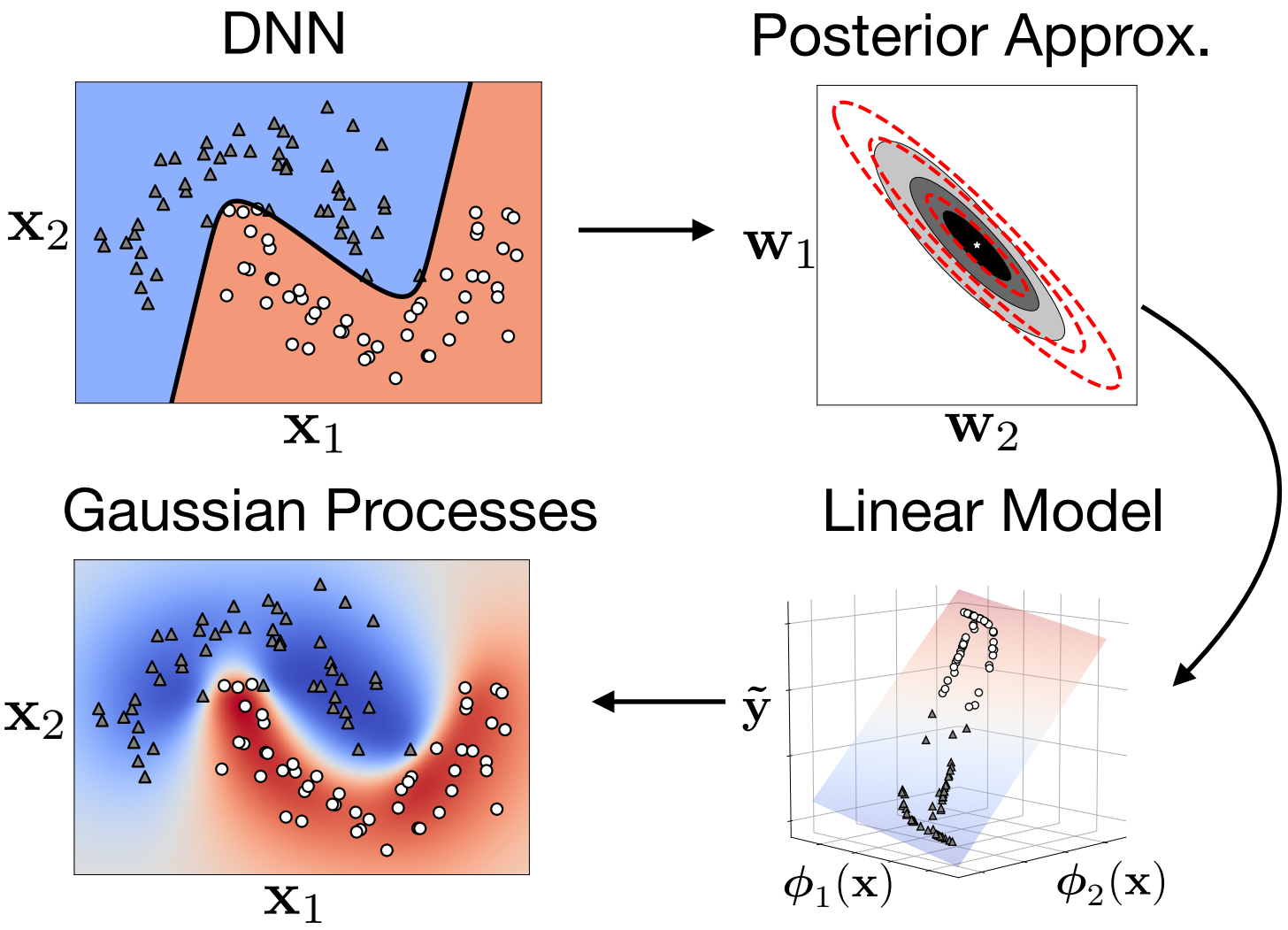
Khan et al., Approximate Inference Turns Deep Networks into Gaussian Processes, NeurIPS, 2019

Outline of the derivation



Khan et al., Approximate Inference Turns Deep Networks into Gaussian Processes, NeurIPS, 2019

Outline of the derivation



Khan et al., Approximate Inference Turns Deep Networks into Gaussian Processes, NeurIPS, 2019

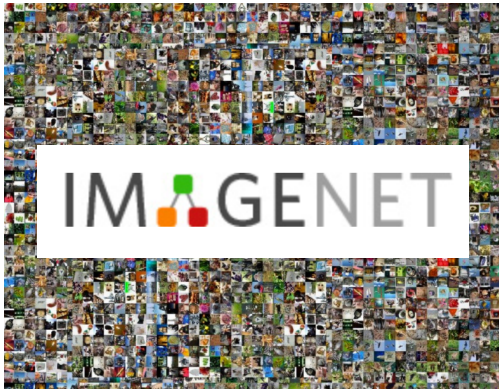
Application to Continual Learning

Standard Deep Learning

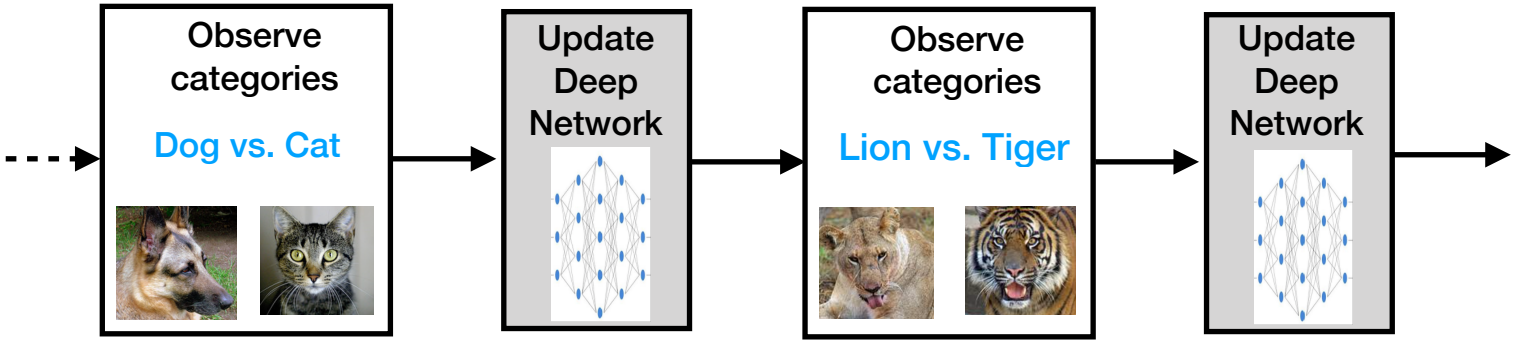
Kirkpatrick, James, et al. "Overcoming catastrophic forgetting in neural networks." *Proceedings of the national academy of sciences* 114.13 (2017): 3521-3526.

Application to Continual Learning

Standard Deep Learning

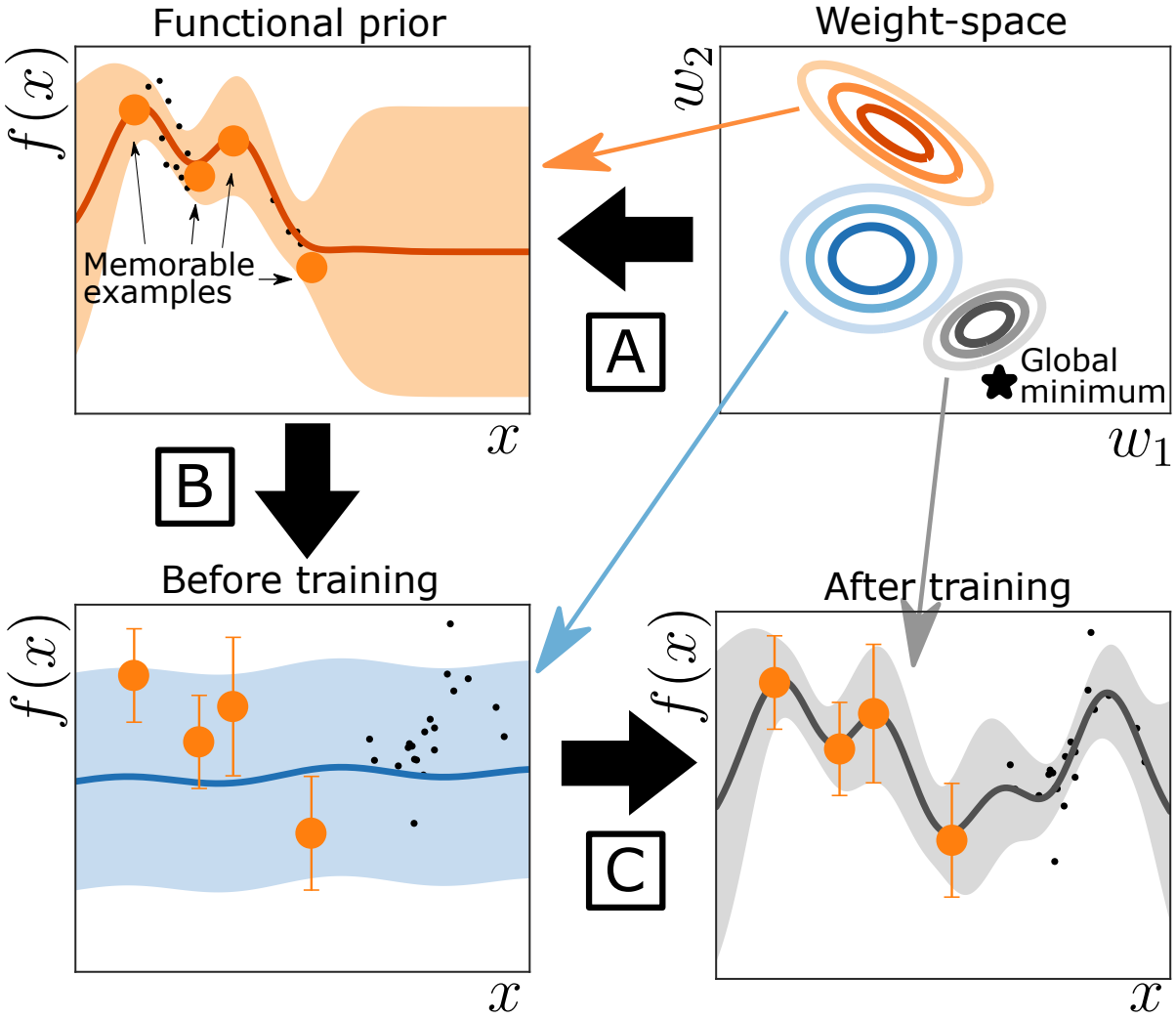


Continual Learning: past classes never revisited



Kirkpatrick, James, et al. "Overcoming catastrophic forgetting in neural networks." *Proceedings of the national academy of sciences* 114.13 (2017): 3521-3526.

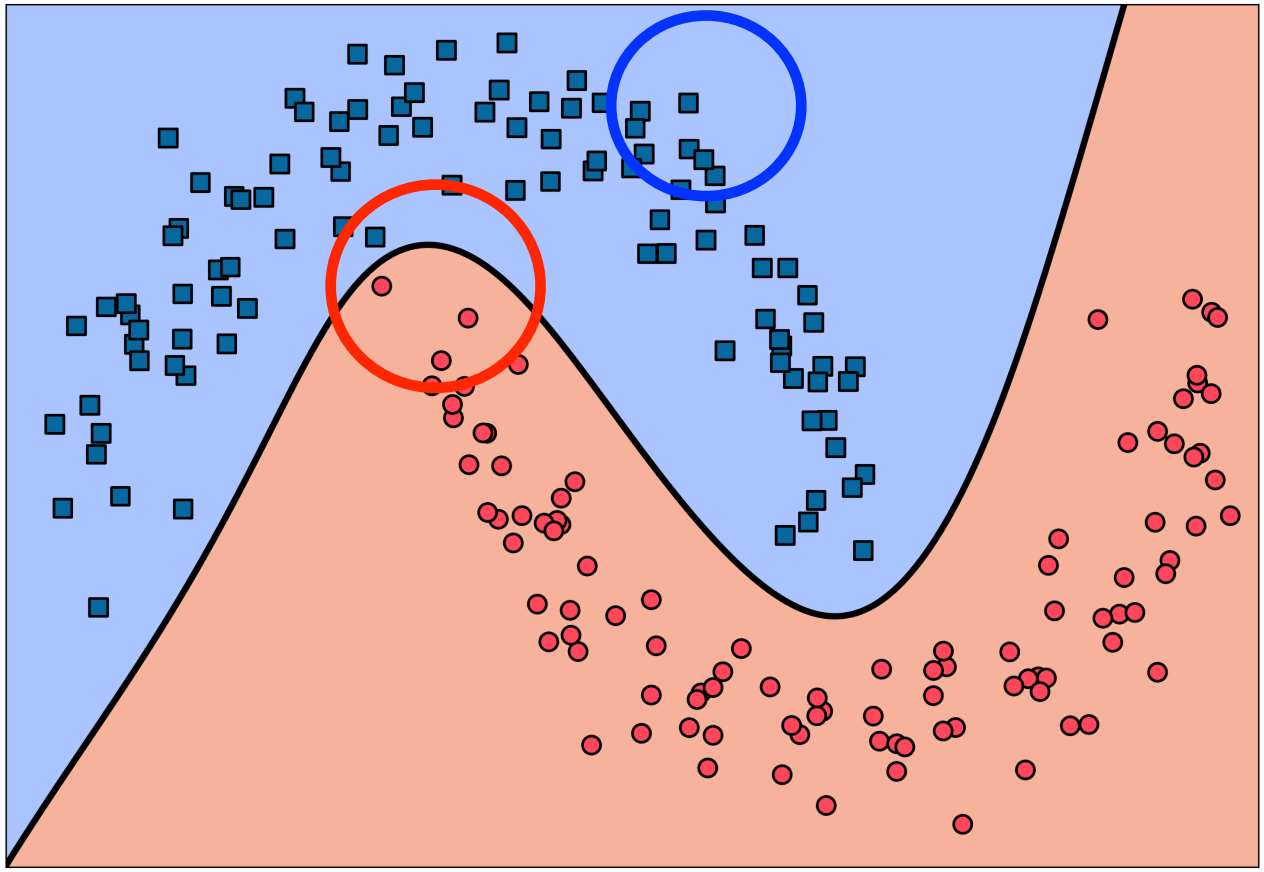
FROMP in Three Steps



Pan et al. Continual Deep Learning by Functional Regularisation of Memorable Past, ArXiv 2020

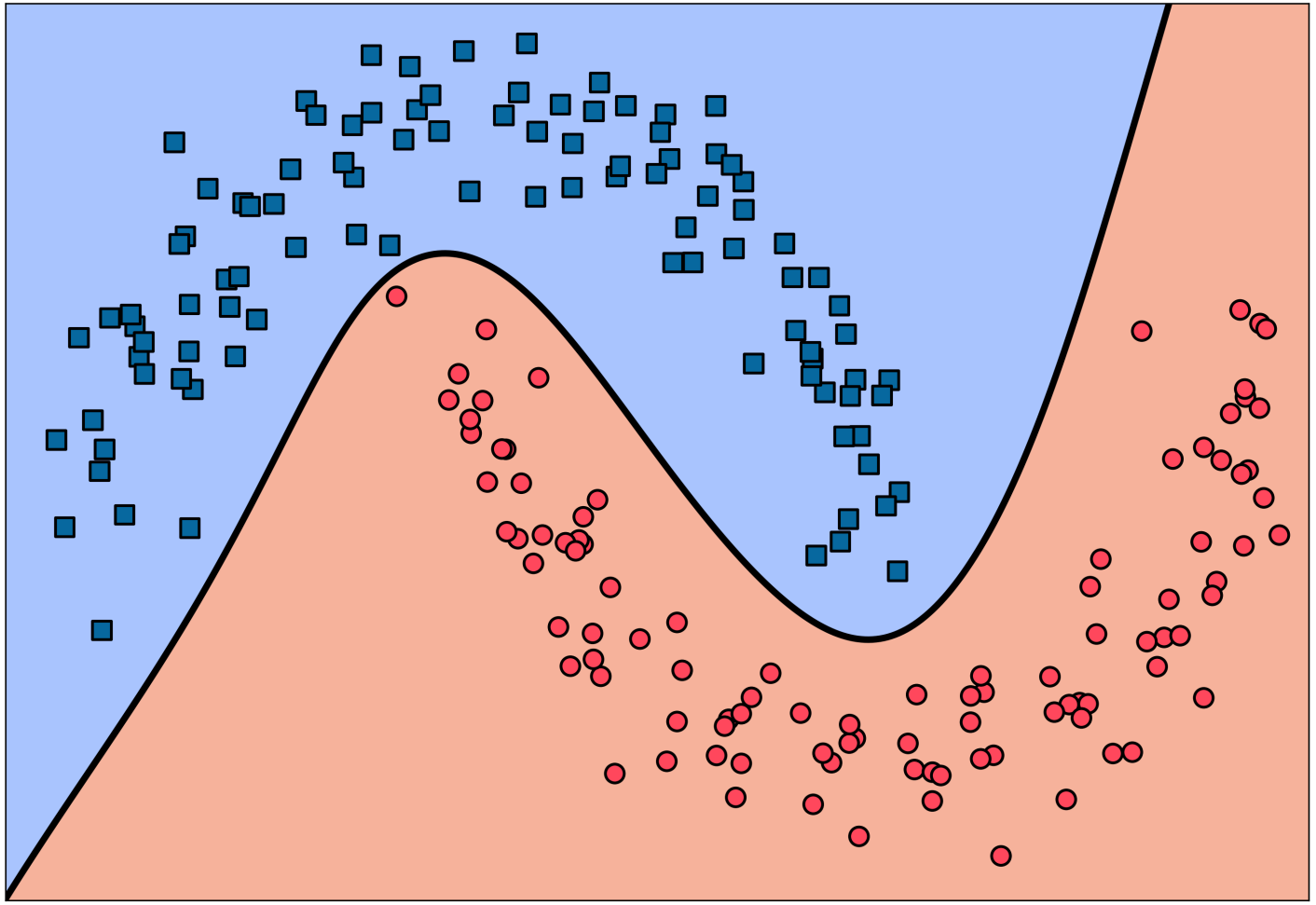
Memorable Past

Which examples are most relevant for the classifier? Red circle vs Blue circle.



Model view vs Data view

DNN2GP provides a measure of relevance

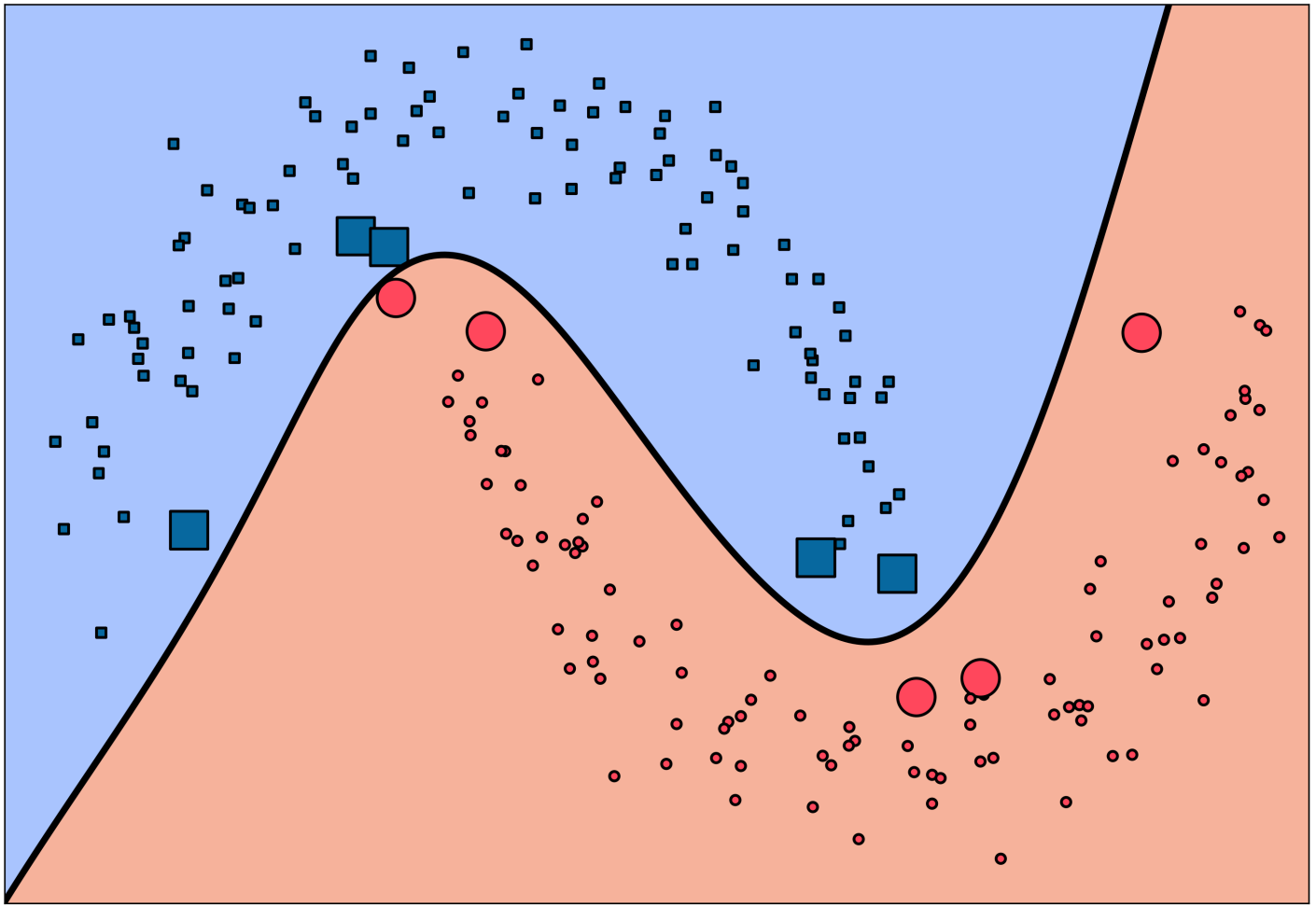


Model view

(By Roman Bachmann)

Model view vs Data view

DNN2GP provides a measure of relevance

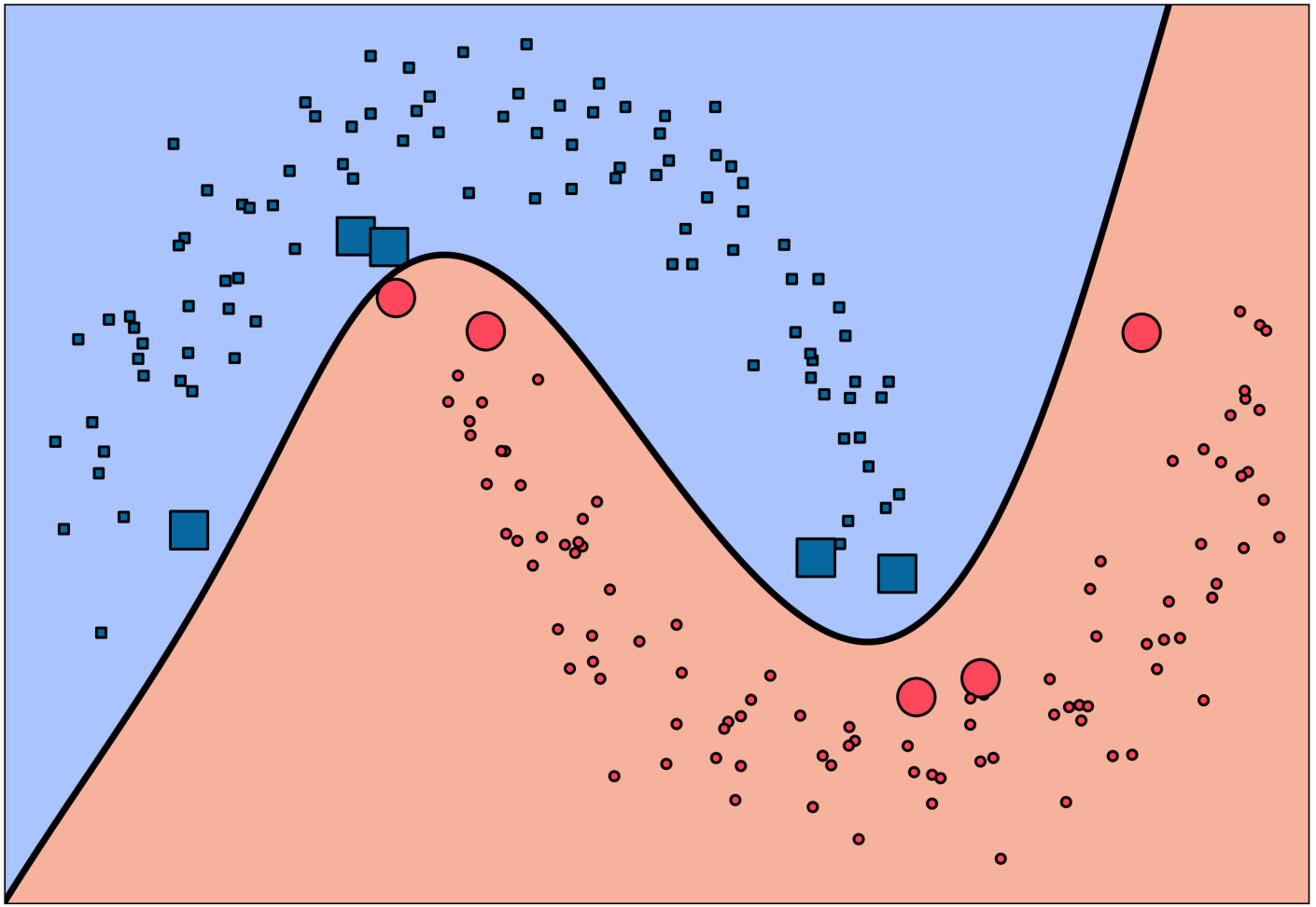


Data view

(By Roman Bachmann)

Model view vs Data view

DNN2GP provides a measure of relevance



Data view

Sort

$\pi_i(x)$

$= \nabla^2_{ff} \ell(y_i, f_0(x_i))$

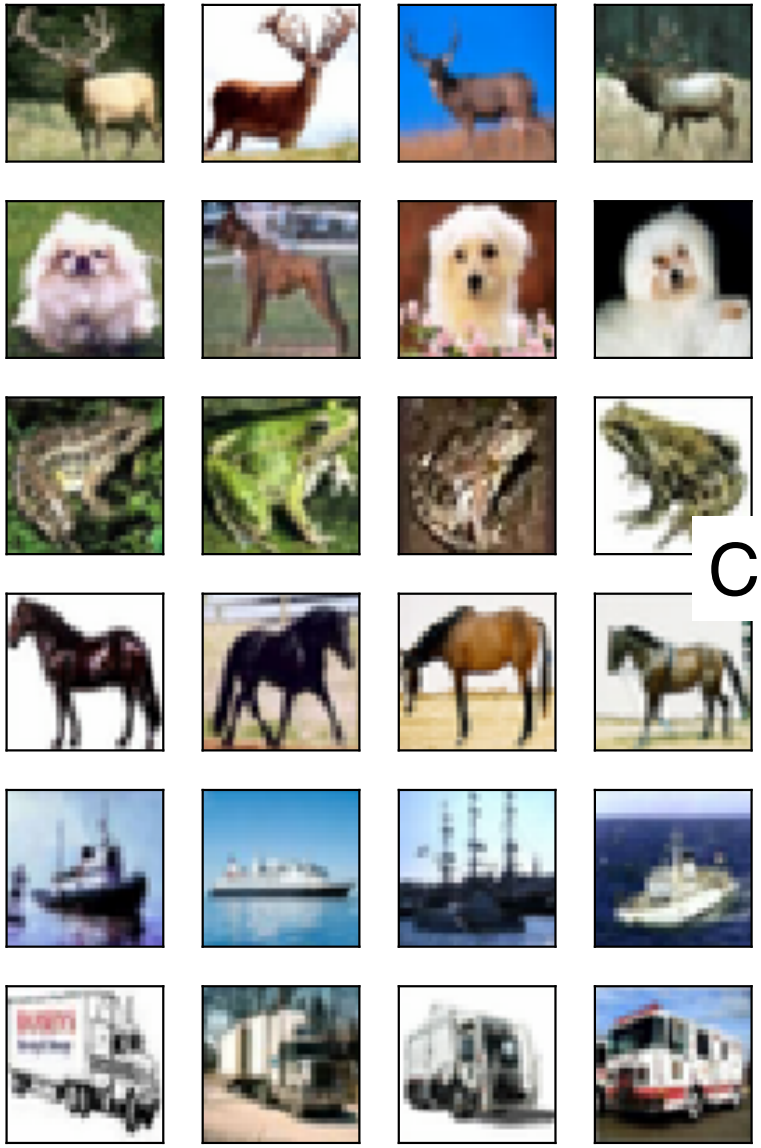
(By Roman Bachmann)

Least Relevant

MNIST

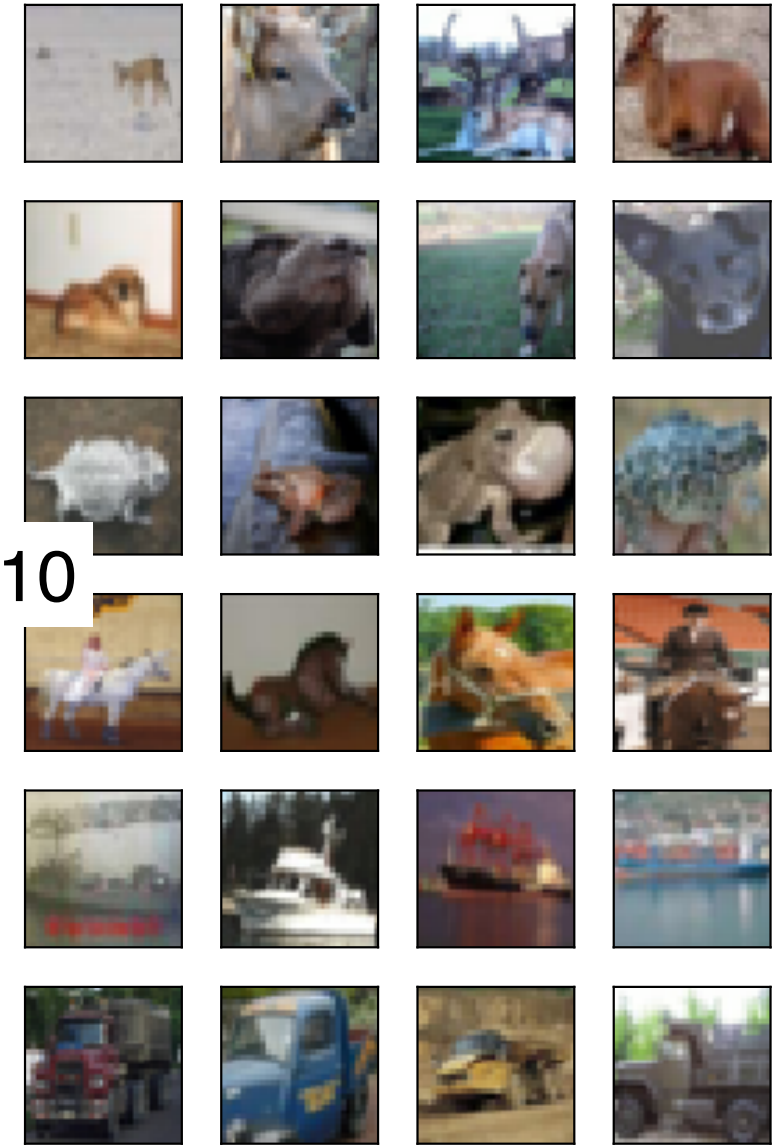
Most Relevant

Least Relevant

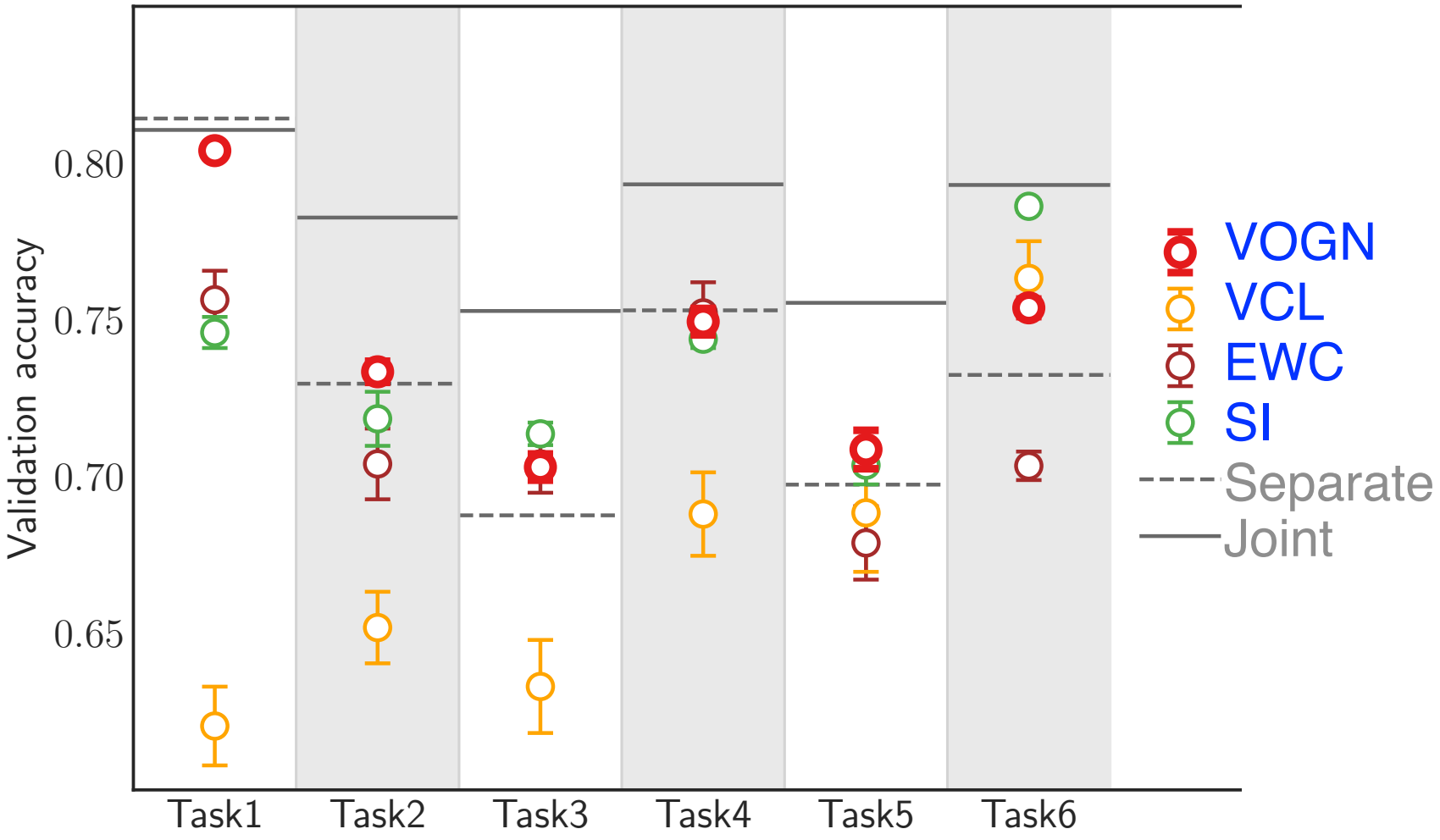


CIFAR-10

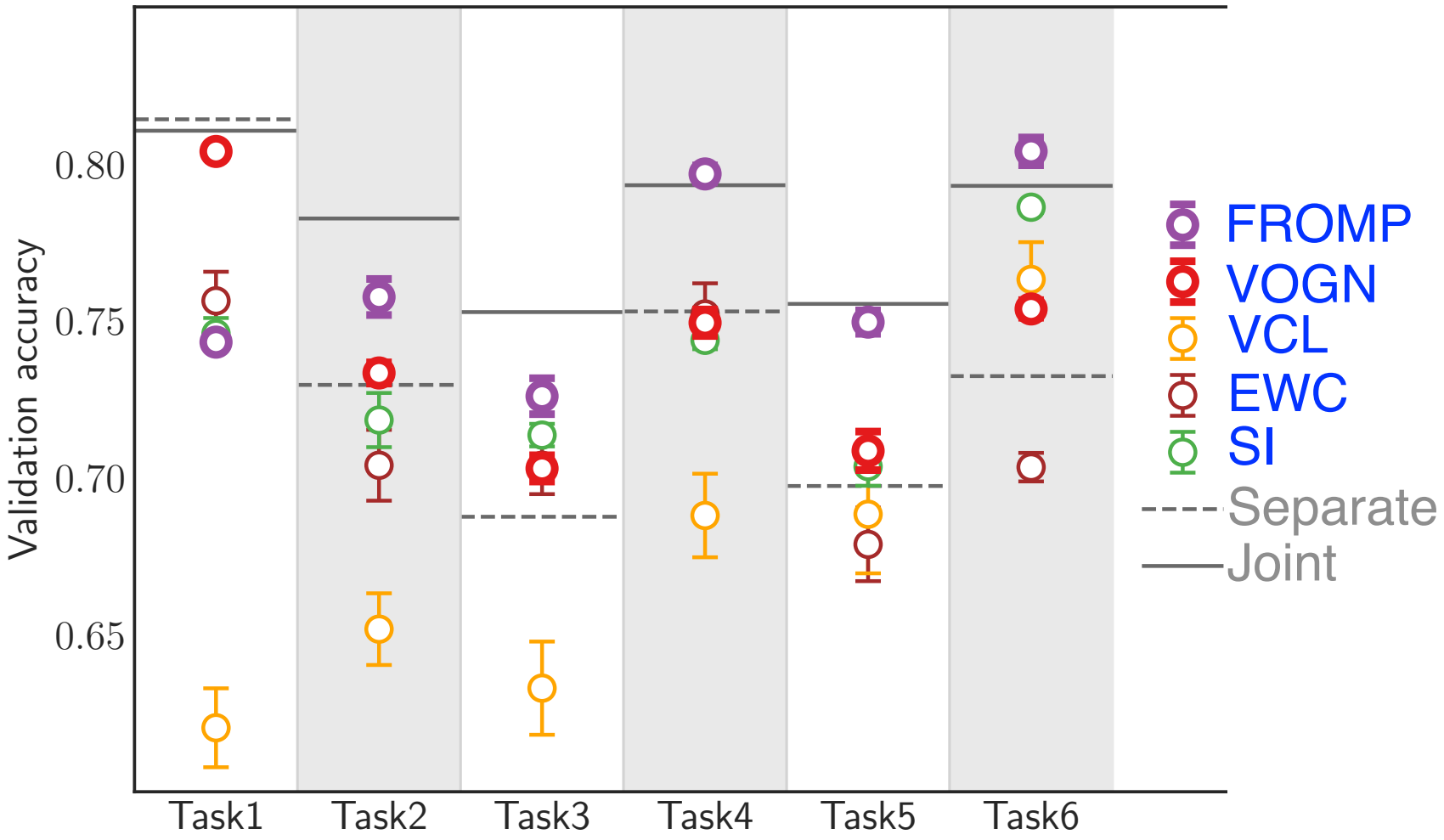
Most Relevant



FROMP improves over EWC!



FROMP improves over EWC!

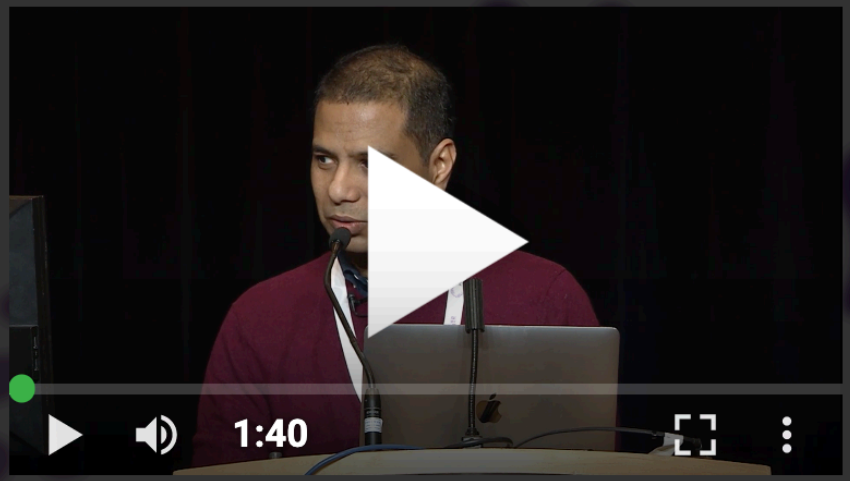


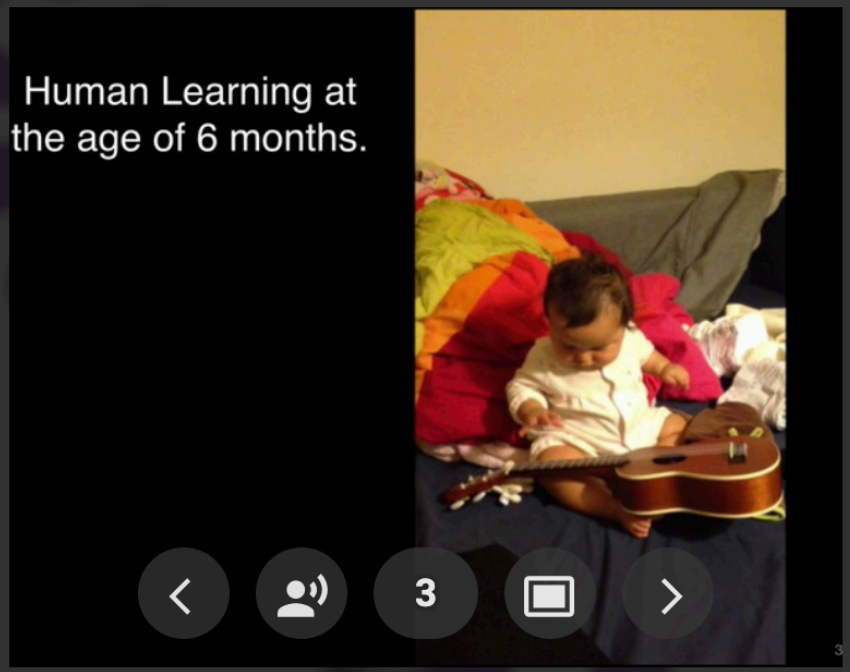
Pan et al. Continual Deep Learning by Functional Regularisation of Memorable Past, ArXiv 2020

(Some) Regularization-based Continual Learning Methods

- Elastic-weight consolidation (EWC) [1]
  - Based on a diagonal Laplace approximation
  - [2] considers structured Laplace
- Synaptic Intelligence (SI) [3]
- Variational Continual learning (VCL) [4]
  - Based on variational inference
- With better approximations, we expect accuracy to improve, but unfortunately we don't see this!

1. Kirkpatrick, James, et al. "Overcoming catastrophic forgetting in neural networks." *PNAS* (2017).
2. Ritter et al. "Online structured laplace … for overcoming catastrophic forgetting." *NeurIPs*. 2018.
3. Zenke et al. "Continual learning through synaptic intelligence." *ICML,* 2017.
4. Nguyen et al. "Variational continual learning." *arXiv preprint arXiv:1710.10628* (2017).





Deep Learning with Bayesian Principles

by **Mohammad Emtiyaz Khan** · Dec 9, 2019 ·

NeurIPS 2019 Tutorial

#NeurIPS 2019 Follow

Views **151 807** Presentations **263** Followers **200**

Latest | Popular | ···

From System 1 Deep Learning to System 2 Deep Learning

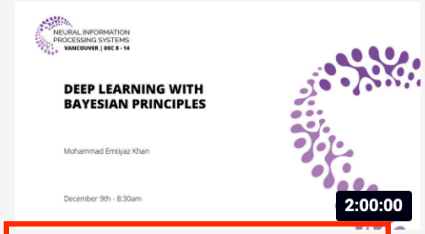
by Yoshua Bengio

17,953 views · Dec 11, 2019

NeurIPS Workshop on Machine Learning for Creativity and Design...

by Aaron Hertzmann, Adam Roberts, ...

9,654 views · Dec 14, 2019



Deep Learning with Bayesian Principles

by Mohammad Emtiyaz Khan

8,084 views · Dec 9, 2019

Efficient Processing of Deep Neural Network: from Algorithms to...

by Vivienne Sze

7,163 views · Dec 9, 2019

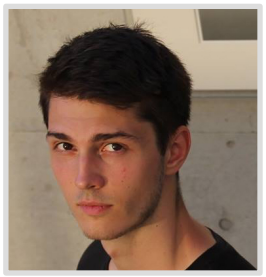
Acknowledgements

Slides, papers, & code are at emtiyaz.github.io

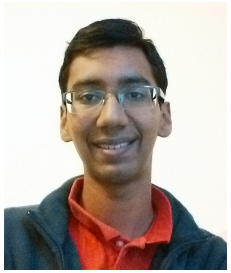
Kazuki Osawa (Tokyo Tech)

Rio Yokota (Tokyo Tech)

PingBo Pan (UT Sydney)



Runa Eschenhagen (Intern from University of Osnabruck)

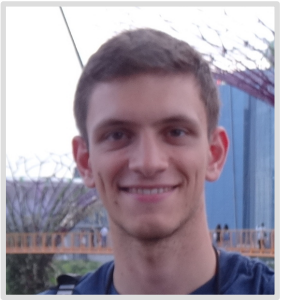


Siddharth Swaroop (University of Cambridge)

Rich Turner (University of Cambridge)

Alexander Immer (Intern from EPFL)

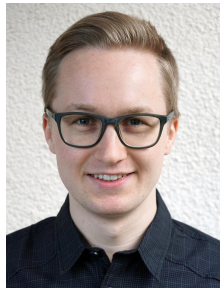
Ehsan Abedi (Intern from EPFL)



Maciej Korzepa (Intern from DTU)

Pierre Alquier (RIKEN AIP)

Xiangming Meng (RIKEN-AIP)



Roman Bachmann (Intern from EPFL)

Approximate Bayesian Inference Team