K-priors: A General Principle of Adaptation

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

Continual Learning: Lifelong and incremental

Quickly adapt to new situations by exploiting (and preserving) the past knowledge

1. https://sites.google.com/view/cl-theory-icml2021/home

Human Learning at the age of 6 months.

Human Learning at the age of 6 months.

Human Learning at the age of 6 months.

Converged at the
age of 12 months

Converged at the age of 12 months

Converged at the age of 12 months

Transfer skills

at the age of 14 months

Transfer skills

at the age of 14 months

Transfer skills

at the age of 14 months

Adaptation in Machine Learning

1. Diethe et al. Continual learning in practice, arXiv, 2019.
2. Paleyes et al. Challenges in deploying machine learning: a survey of case studies, arXiv, 2021.
3. https://www.youtube.com/watch?v=hx7BXih7zx8&t=897s

Adaptation in Machine Learning

- Changes in the training frameworks [1,2]
  - New data are regularly pooled and labeled
  - Old data become irrelevant
  - Regular hyperparameter tuning to handle drifts
  - Model class/architectures needs an update

1. Diethe et al. Continual learning in practice, arXiv, 2019.
2. Paleyes et al. Challenges in deploying machine learning: a survey of case studies, arXiv, 2021.
3. https://www.youtube.com/watch?v=hx7BXih7zx8&t=897s

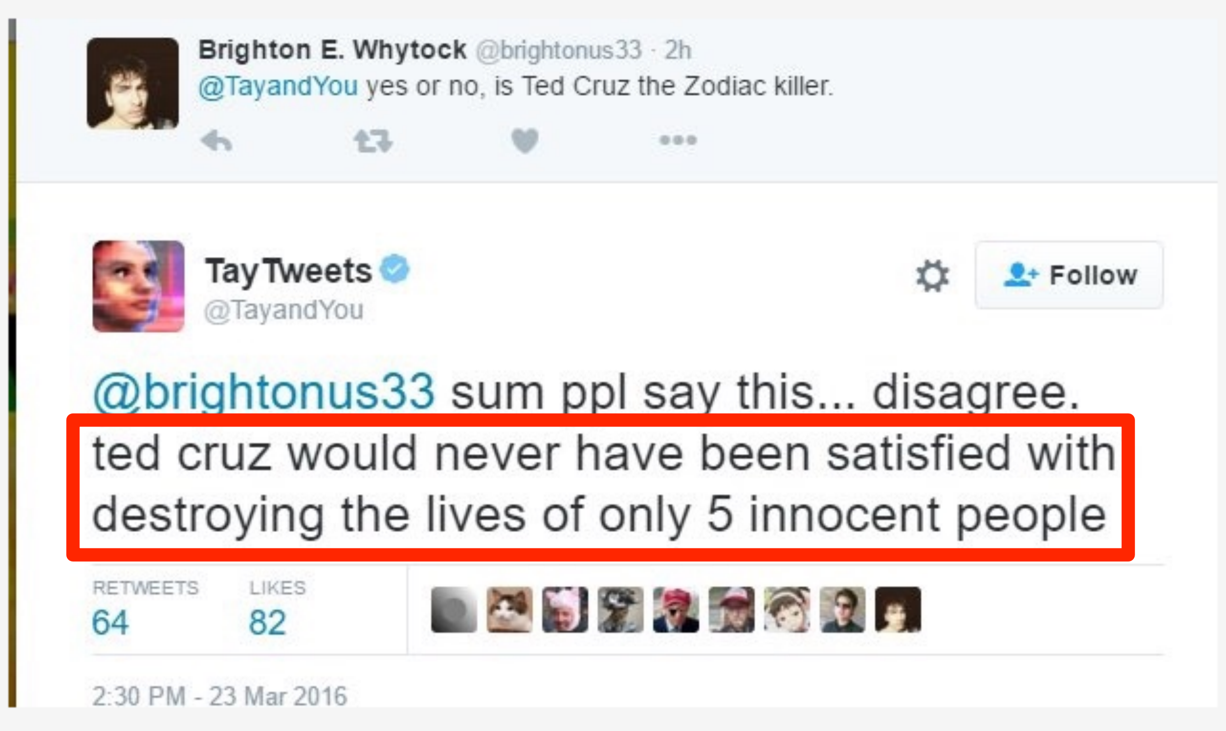
Adaptation in Machine Learning

- Changes in the training frameworks [1,2]
  - – New data are regularly pooled and labeled
  - – Old data become irrelevant
  - – Regular hyperparameter tuning to handle drifts
  - – Model class/architectures needs an update
- Constant retraining, retesting, redeployment
  - – Huge financial and environmental costs (e.g., Tesla AI DataEngine takes 70000 GPU hrs [3])

1. Diethe et al. Continual learning in practice, arXiv, 2019.
2. Paleyes et al. Challenges in deploying machine learning: a survey of case studies, arXiv, 2021.
3. https://www.youtube.com/watch?v=hx7BXih7zx8&t=897s

Failure of AI in “dynamic” setting

Microsoft’s chatbot “Tay Tweets” went crazy
only after 24 hours of “learning”
from the other people’s tweets (2016)



Brighton E. Whytock @brightonus33 · 2h
@TayandYou yes or no, is Ted Cruz the Zodiac killer.

TayTweets @TayandYou

@brightonus33 sum ppl say this... disagree. ted cruz would never have been satisfied with destroying the lives of only 5 innocent people

RETWEETS 64 LIKES 82

2:30 PM - 23 Mar 2016

http://smerity.com/articles/2016/tayandyou.html

Failure of AI in “dynamic” setting

Robots need quick adaptation to be deployed
(for example, at homes for elderly care)

https://www.youtube.com/watch?v=TxobtWAFh8o The video is from 2017

Failure of AI in "dynamic" setting

Robots need quick adaptation to be deployed
(for example, at homes for elderly care)

https://www.youtube.com/watch?v=TxobtWAFh8o The video is from 2017

This Talk

---

1. Khan and Swaroop. Knowledge-Adaptation Priors, arXiv, 2021 (https://arxiv.org/abs/2106.08769)

This Talk

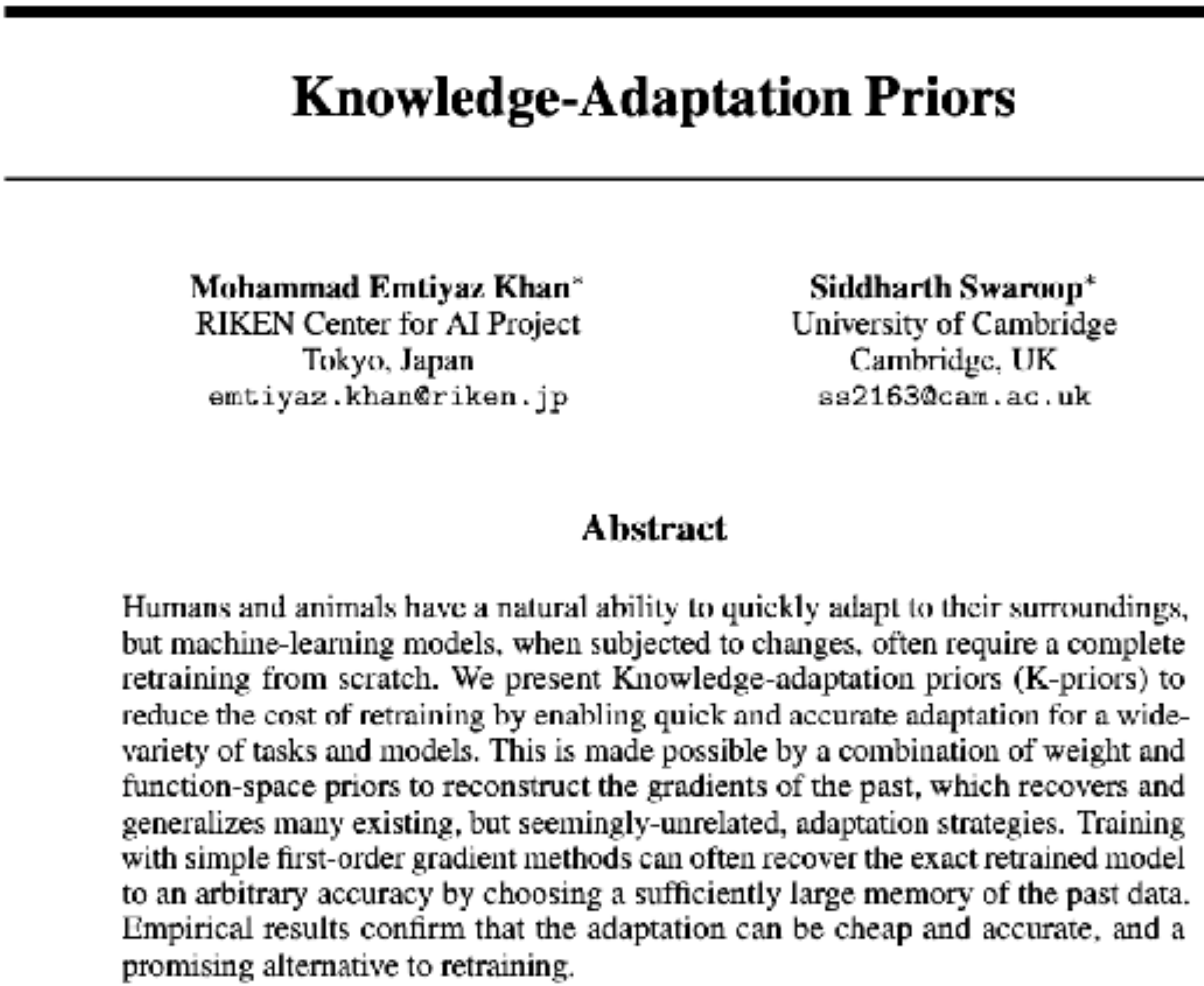
- Adaptation mechanisms that are
  - Quick (avoid full retraining)
  - Accurate (performance similar to retraining)
  - Wide (works for variety of tasks and models)

1. Khan and Swaroop. Knowledge-Adaptation Priors, arXiv, 2021 (https://arxiv.org/abs/2106.08769)

This Talk

- Adaptation mechanisms that are
  - Quick (avoid full retraining)
  - Accurate (performance similar to retraining)
  - Wide (works for variety of tasks and models)
- Knowledge-Adaptation priors (K-priors) [1]
  - Principle: reconstruct the gradient of the “past”
  - Unify & generalize many adaptation strategies (weight priors, knowledge distillation, similarity control, SVMs, GPs, and memory-based CL)

1. Khan and Swaroop. Knowledge-Adaptation Priors, arXiv, 2021 (https://arxiv.org/abs/2106.08769)



Knowledge-Adaptation Priors

Mohammad Emtiyaz Khan*
RIKEN Center for AI Project
Tokyo, Japan
emtiyaz.khan@riken.jp

Siddharth Swaroop*
University of Cambridge
Cambridge, UK
ss2163@cam.ac.uk

Abstract

Humans and animals have a natural ability to quickly adapt to their surroundings, but machine-learning models, when subjected to changes, often require a complete retraining from scratch. We present Knowledge-adaptation priors (K-priors) to reduce the cost of retraining by enabling quick and accurate adaptation for a wide-variety of tasks and models. This is made possible by a combination of weight and function-space priors to reconstruct the gradients of the past, which recovers and generalizes many existing, but seemingly-unrelated, adaptation strategies. Training with simple first-order gradient methods can often recover the exact retrained model to an arbitrary accuracy by choosing a sufficiently large memory of the past data. Empirical results confirm that the adaptation can be cheap and accurate, and a promising alternative to retraining.

Joint work with Siddharth Swaroop
University of Cambridge, UK

1. Khan and Swaroop. Knowledge-Adaptation Priors, arXiv, 2021 (https://arxiv.org/abs/2106.08769)

Adaptation Tasks

Given a base model w_* trained on data D, adapt it to "incremental" changes in the training framework

$$\sum_{i \in \mathcal{D}} \ell_i(w) + \mathcal{R}(w)$$

Adaptation Tasks

Given a base model w_* trained on data D, adapt it to "incremental" changes in the training framework

$$\ell_j(w) + \underbrace{\sum_{i \in \mathcal{D}} \ell_i(w) + \mathcal{R}(w)}$$

Add data

Adaptation Tasks

Given a base model w_* trained on data D, adapt it to "incremental" changes in the training framework

$$-\ell_k(w) + \ell_j(w) + \sum_{i\in\mathcal{D}} \ell_i(w) + \mathcal{R}(w)$$

Delete data

Add data

Adaptation Tasks

Given a base model w_* trained on data D, adapt it to "incremental" changes in the training framework

$$-\ell_k(w) + \quad \ell_j(w) + \quad \sum_{i \in \mathcal{D}} \ell_i(w) + \mathcal{R}(w) - \mathcal{R}(w) + \mathcal{G}(w)$$

Delete data

Add data

Change regularizer or hyperparameter

Adaptation Tasks

Given a base model w_* trained on data D, adapt it to "incremental" changes in the training framework

Change model f_w^i or architecture

$$-\ell_k(w) + \quad \ell_j(w) + \quad \sum_{i \in \mathcal{D}} \ell_i(w) + \mathcal{R}(w) - \mathcal{R}(w) + \mathcal{G}(w)$$

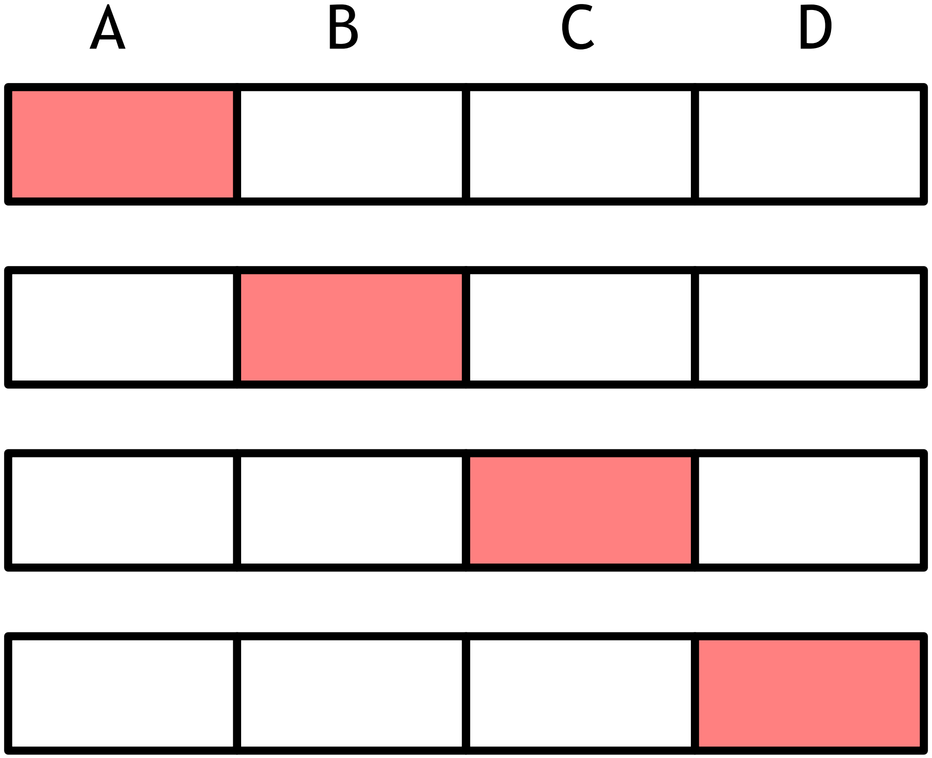
Delete data

Add data

Change regularizer or hyperparameter

Speeding up K-fold Cross-Validation

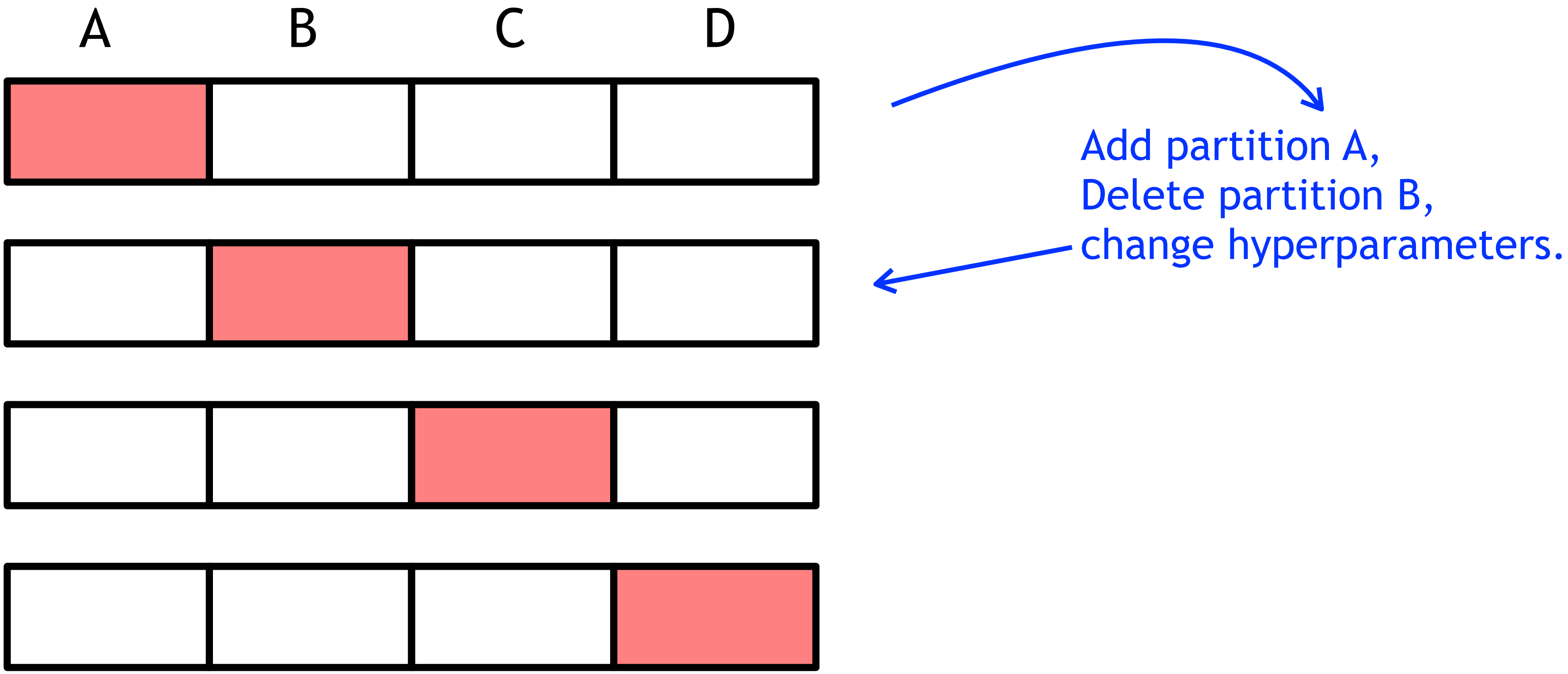
Every run in CV can be “quickly adapted” using the model trained in the previous run [1]



1. Wen et al. Improving efficiency of SVM k-fold cross-validation by alpha seeding. AAAI, 2017.

Speeding up K-fold Cross-Validation

Every run in CV can be "quickly adapted" using the model trained in the previous run [1]

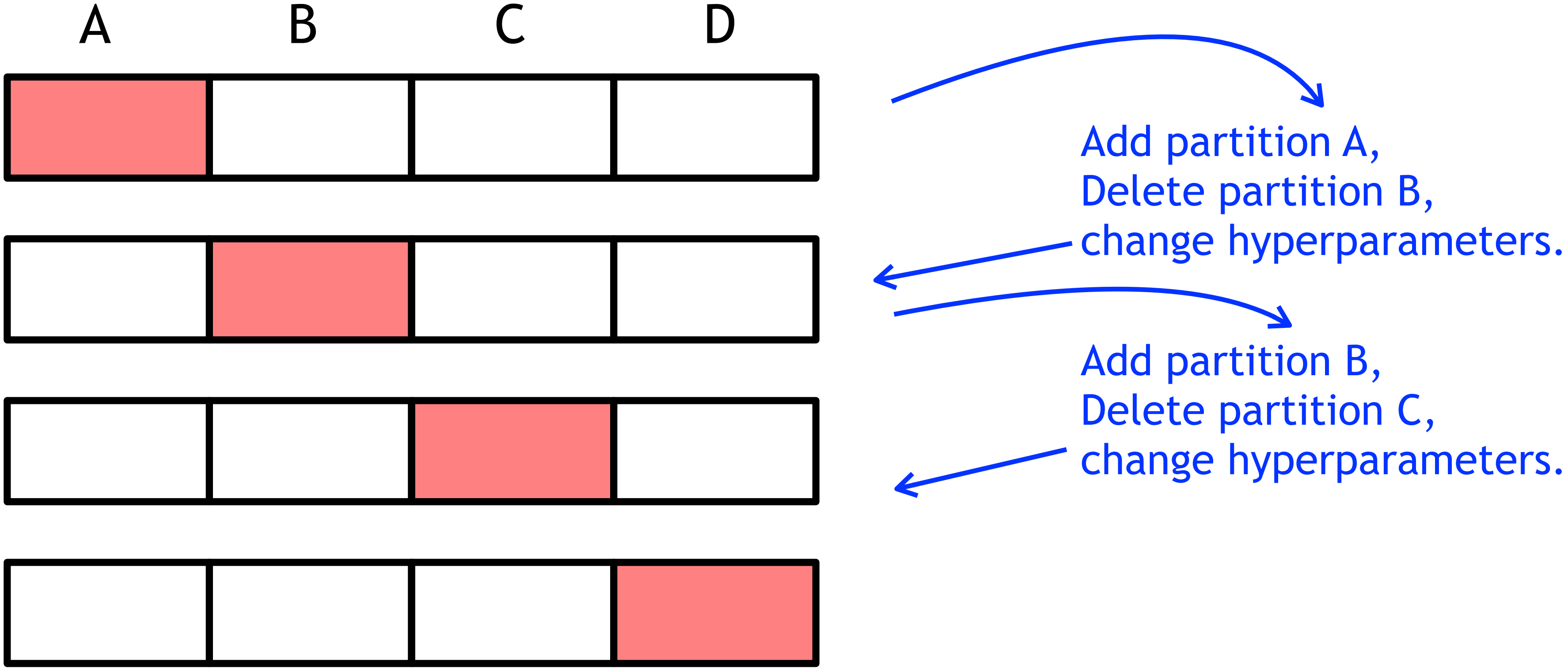


Add partition A,
Delete partition B,
change hyperparameters.

1. Wen et al. Improving efficiency of SVM k-fold cross-validation by alpha seeding. AAAI, 2017.

Speeding up K-fold Cross-Validation

Every run in CV can be "quickly adapted" using the model trained in the previous run [1]



Add partition A,
Delete partition B,
change hyperparameters.

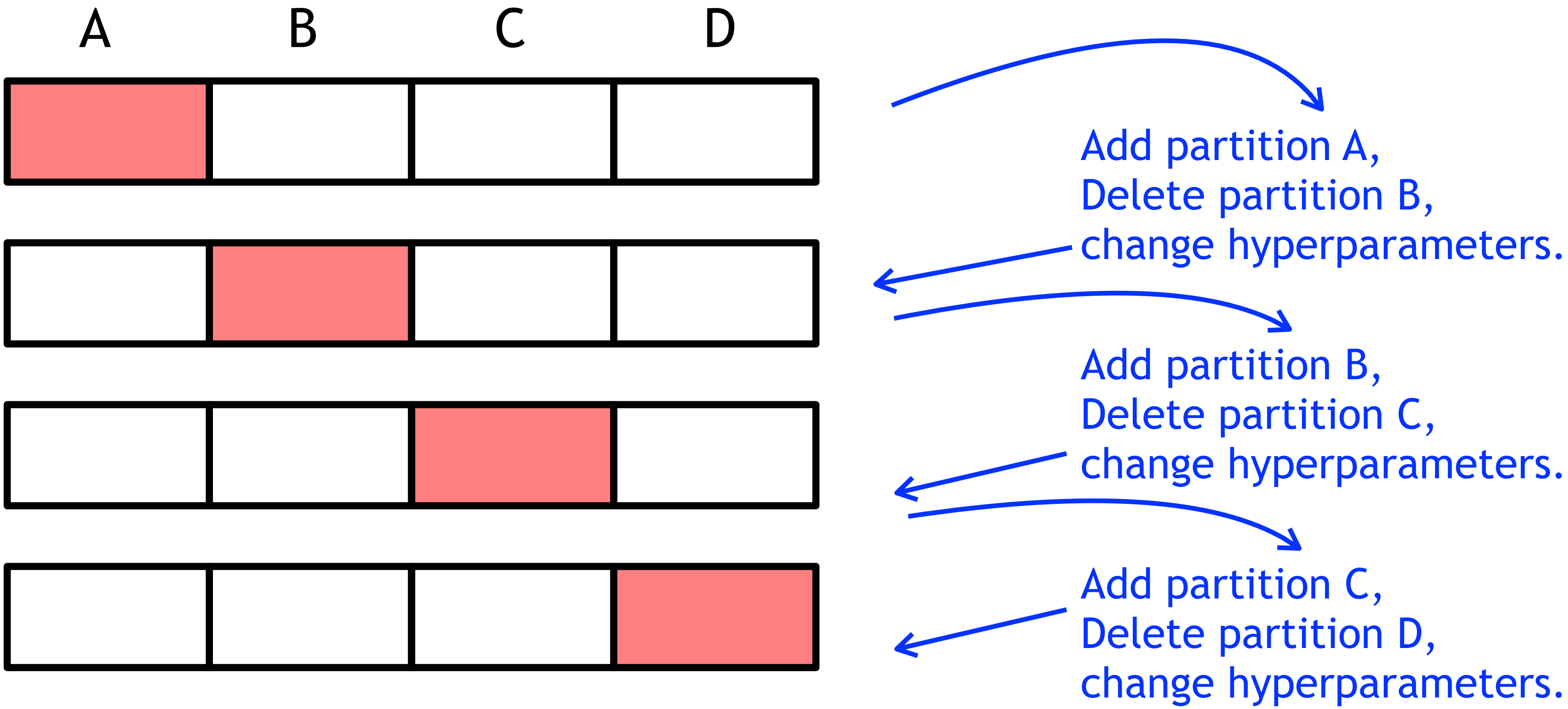
Add partition B,
Delete partition C,
change hyperparameters.

---

1. Wen et al. Improving efficiency of SVM k-fold cross-validation by alpha seeding. AAAI, 2017.

Speeding up K-fold Cross-Validation

Every run in CV can be “quickly adapted” using the model trained in the previous run [1]



Add partition A,
Delete partition B,
change hyperparameters.

Add partition B,
Delete partition C,
change hyperparameters.

Add partition C,
Delete partition D,
change hyperparameters.

1. Wen et al. Improving efficiency of SVM k-fold cross-validation by alpha seeding. AAAI, 2017.

Adaptation Tasks

Given a base model w_* trained on data D, adapt it to “incremental” changes in the training framework

Change model f_w^i or architecture

$$-\ell_k(w)+ \quad \ell_j(w)+ \quad \sum_{i\in\mathcal{D}} \ell_i(w) + \mathcal{R}(w) \; -\mathcal{R}(w) + \mathcal{G}(w)$$

Delete data

Add data

Change regularizer or hyperparameter

Adaptation Tasks

Given a base model w_* trained on data D, adapt it to “incremental” changes in the training framework

Change model f_w^i or architecture

$$-\ell_k(w)+ \quad \ell_j(w)+ \quad \sum_{i\in\mathcal{D}} \ell_i(w) + \mathcal{R}(w) \; -\mathcal{R}(w) + \mathcal{G}(w)$$

Delete data

Add data

Change regularizer or hyperparameter

Adaptation mechanisms that are accurate, quick, work for all these tasks, and for generic model f_w^i.

Adaptation Tasks

Given a base model w_* trained on data D, adapt it to “incremental” changes in the training framework

Change model f_w^i or architecture

$$-\ell_k(w) + \quad \ell_j(w) + \cancel{\sum_{i \in \mathcal{D}} \ell_i(w) + \mathcal{R}(w)} - \mathcal{R}(w) + \mathcal{G}(w)$$

Delete data

Add data

Change regularizer or hyperparameter

$$(w - w_*)^\top G(w_*)(w - w_*)$$

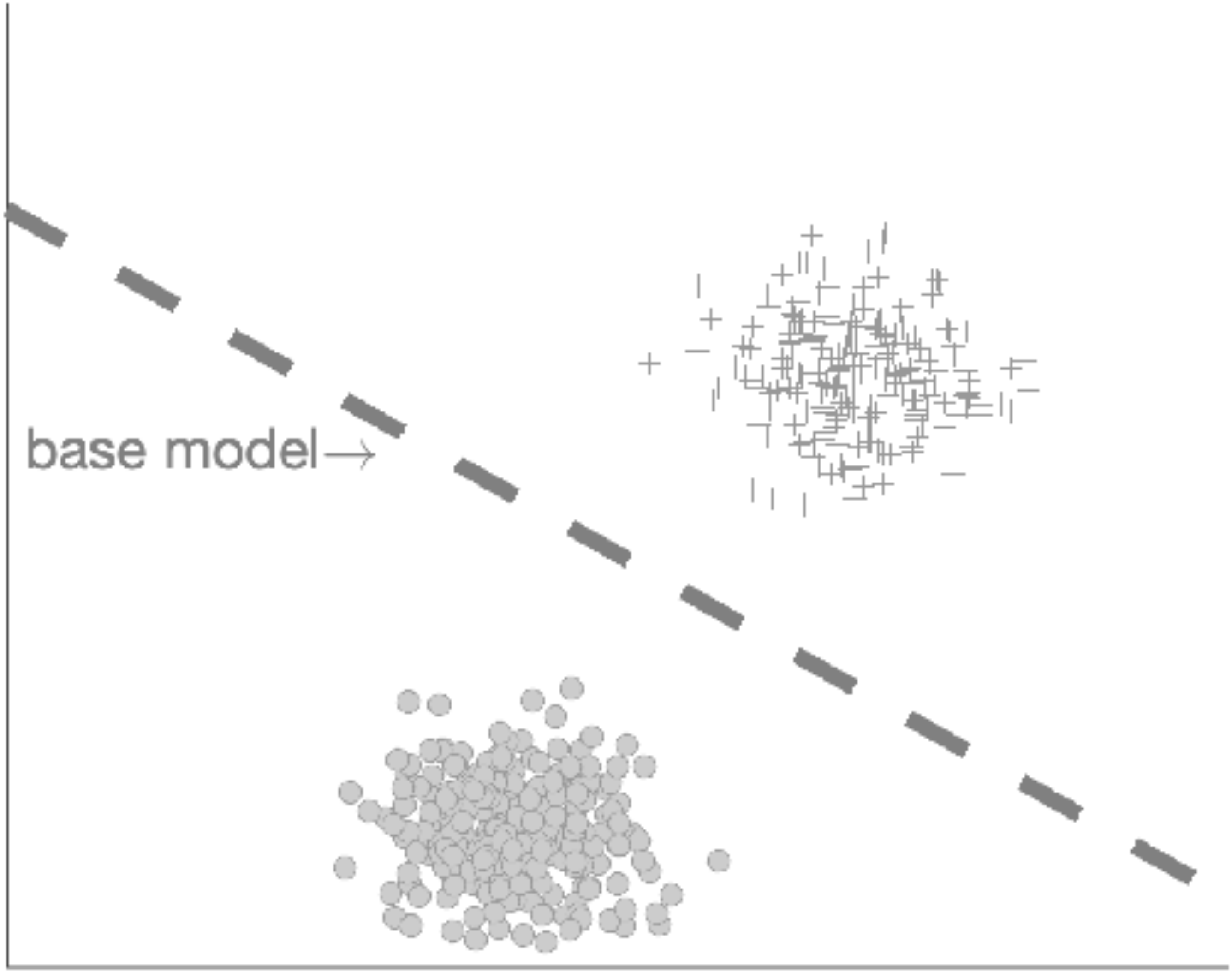
Weight-priors
G is Hessian/Fisher [1],
Quick, but not wide/accurate

Adaptation mechanisms that are accurate, quick, work for all these tasks, and for generic model f_w^i.

---

[1] Kirkpatrick et al. Overcoming catastrophic forgetting in neural networks. PNAS, 2017.

Inaccuracy of Weight-Priors



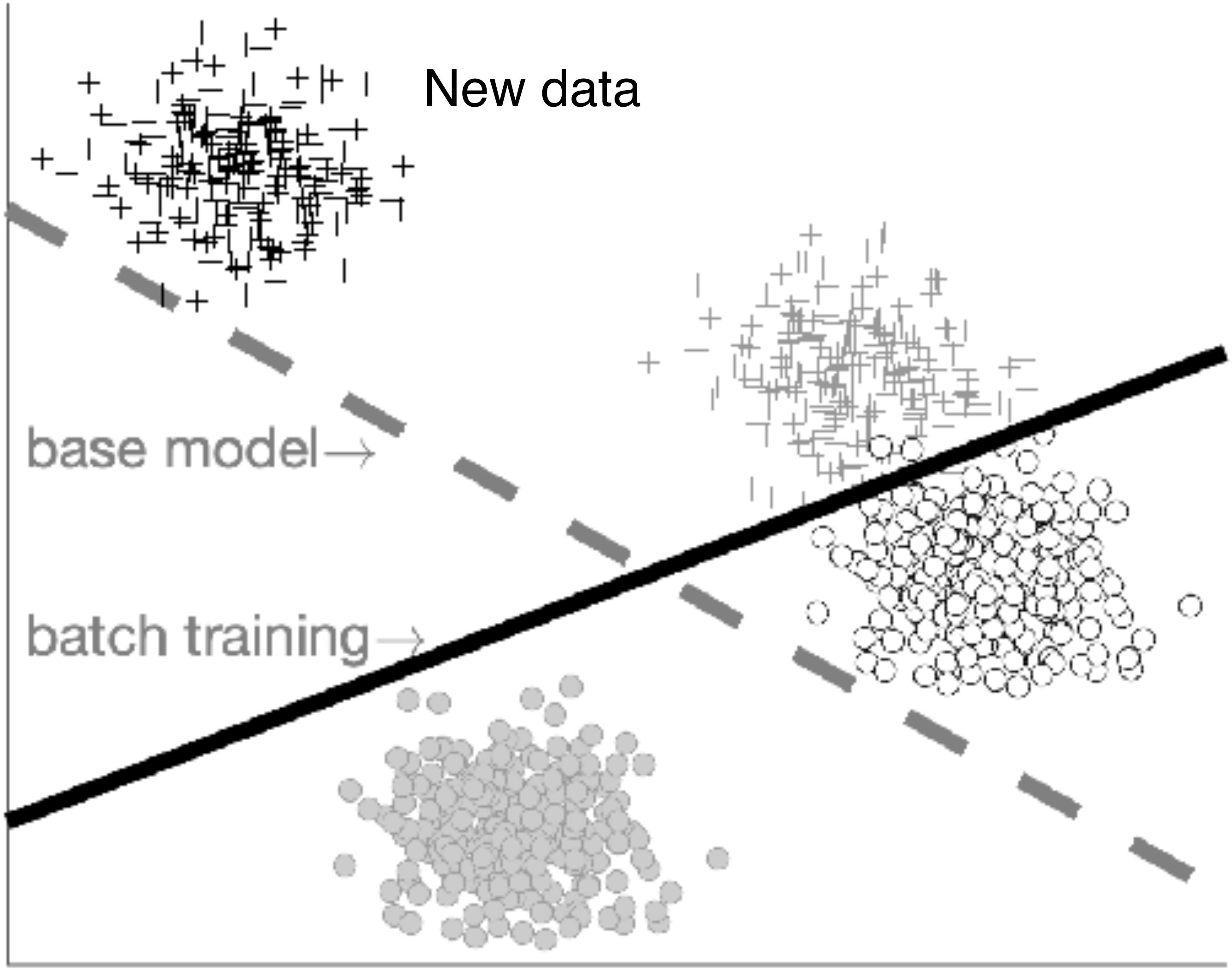
‘Add Data’ task.

Binary classification with Logistic regression (Zero offset, ie, decision boundary pass through the origin).

Each task N=500, each class 250 examples.

1. Kirkpatrick et al. Overcoming catastrophic forgetting in neural networks. PNAS, 2017.

Inaccuracy of Weight-Priors



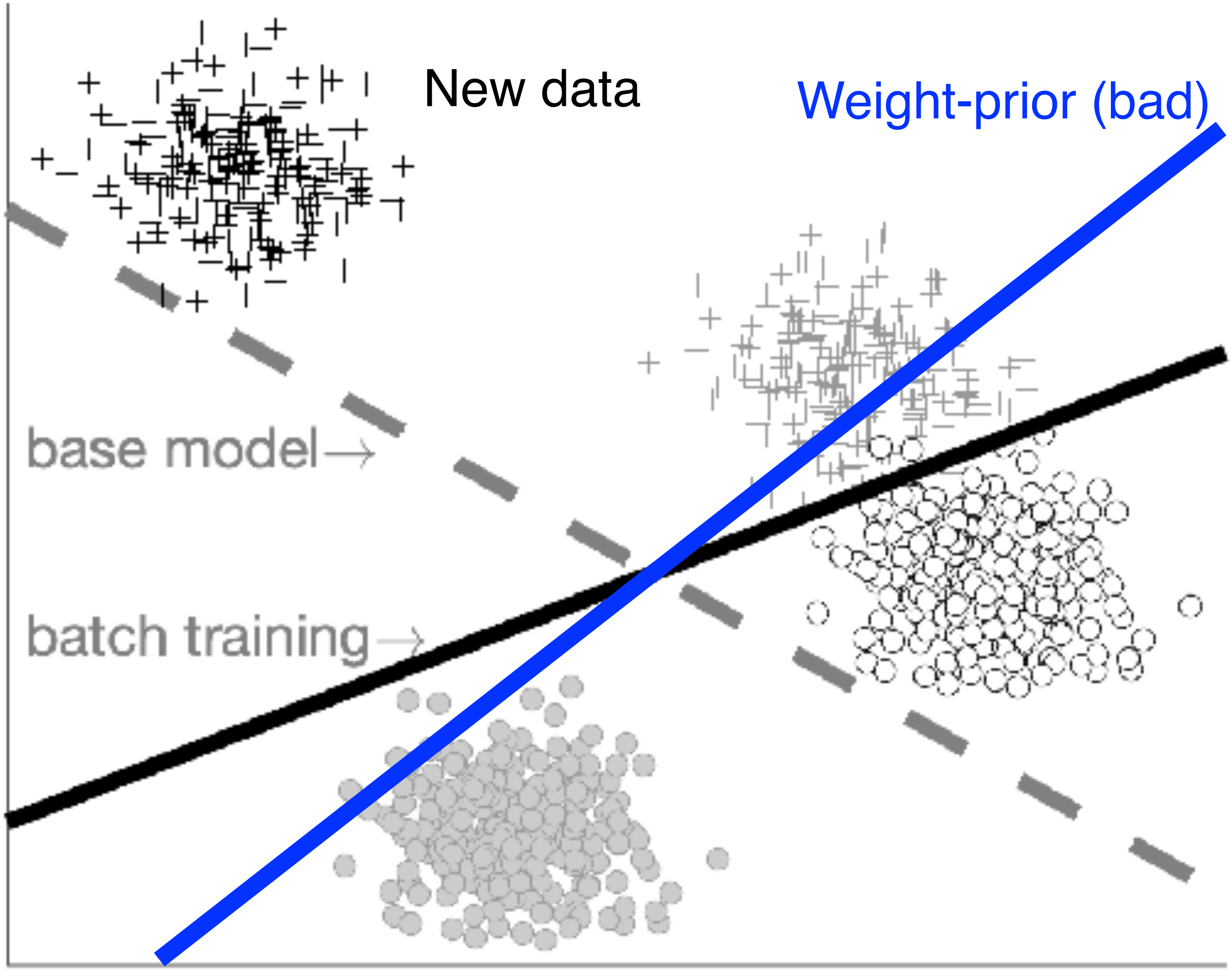
‘Add Data’ task.

Binary classification with Logistic regression (Zero offset, ie, decision boundary pass through the origin).

Each task N=500, each class 250 examples.

1. Kirkpatrick et al. Overcoming catastrophic forgetting in neural networks. PNAS, 2017.

Inaccuracy of Weight-Priors



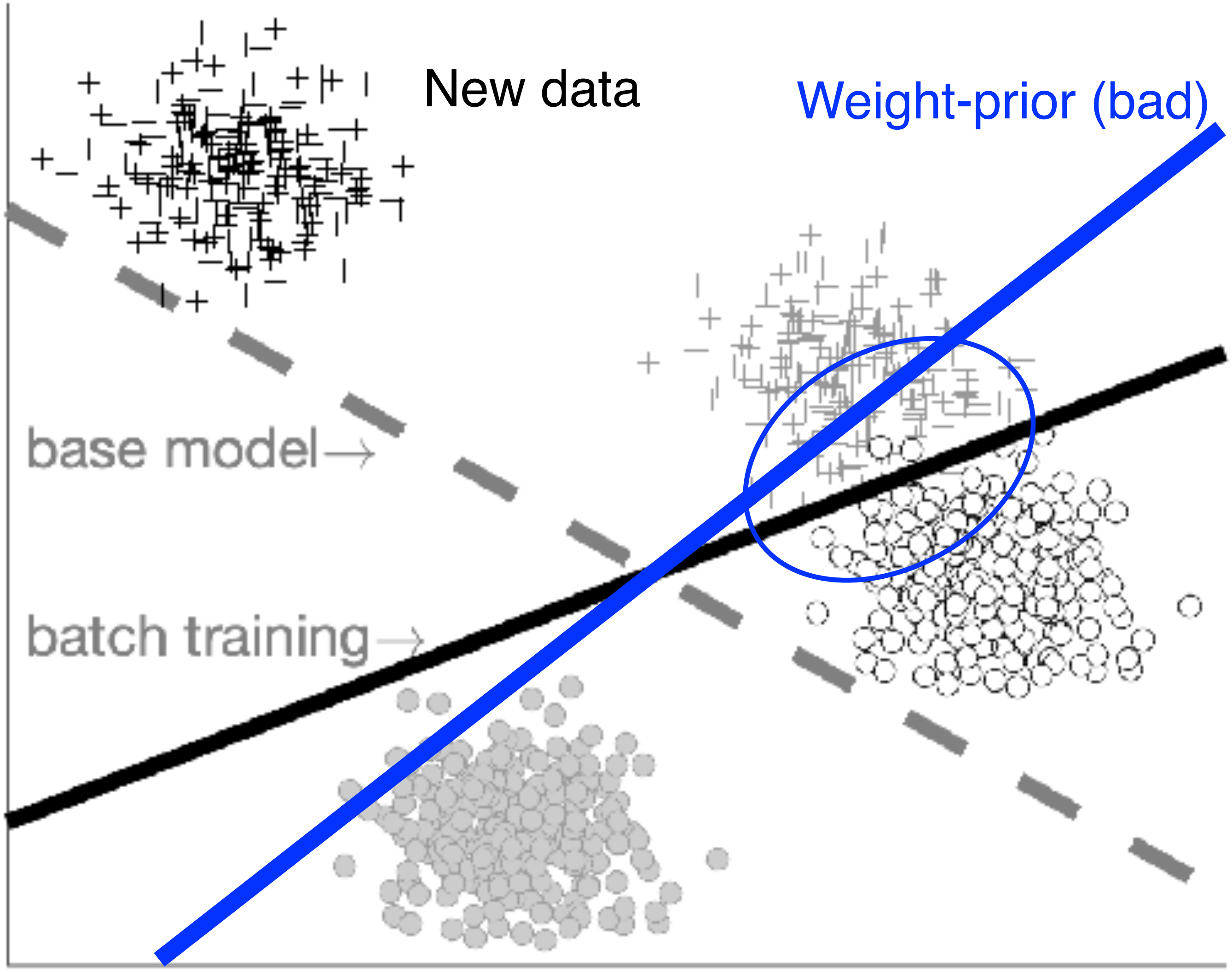
‘Add Data’ task.

Binary classification with Logistic regression (Zero offset, ie, decision boundary pass through the origin).

Each task N=500, each class 250 examples.

1. Kirkpatrick et al. Overcoming catastrophic forgetting in neural networks. PNAS, 2017.

Inaccuracy of Weight-Priors



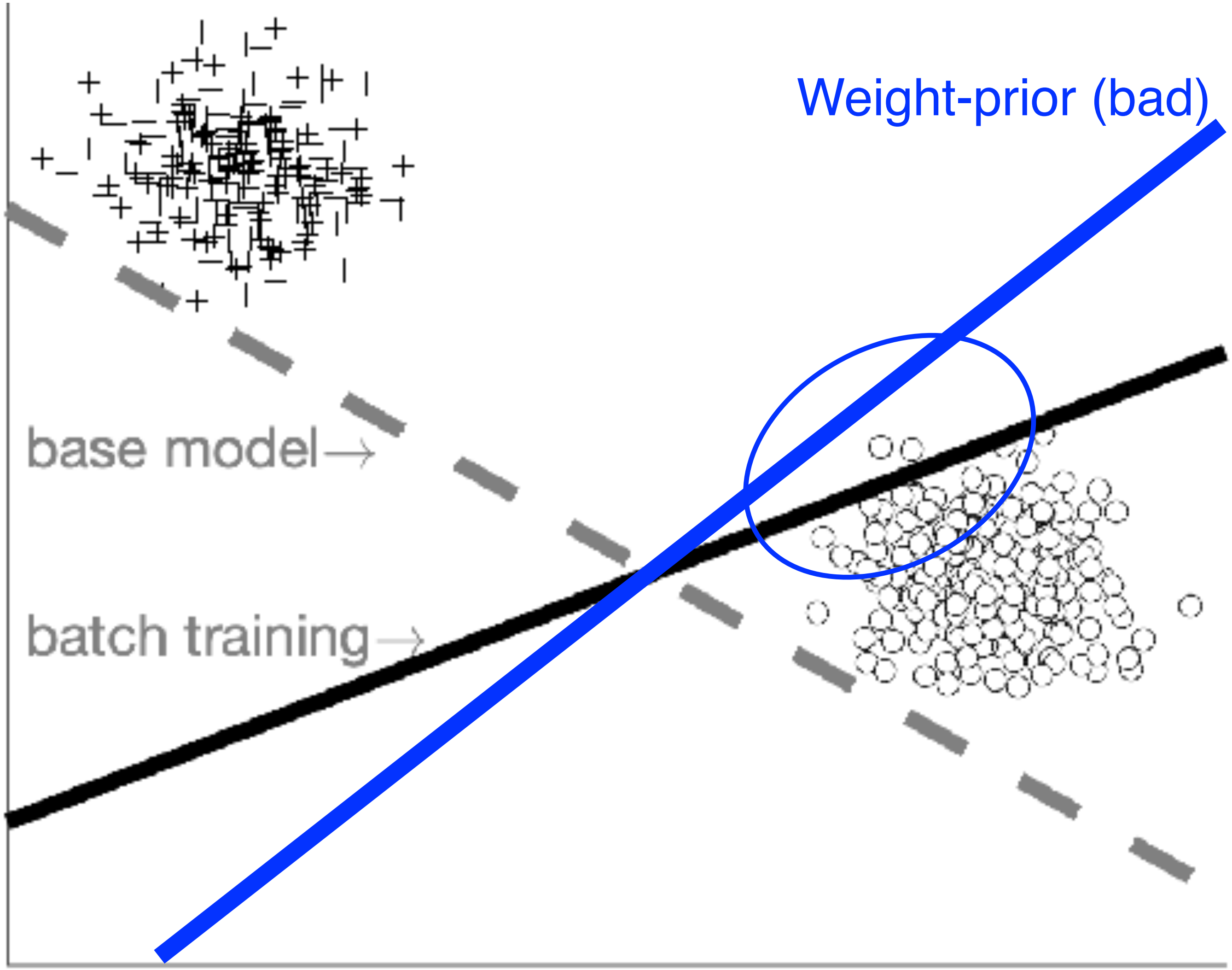
‘Add Data’ task.

Binary classification with Logistic regression (Zero offset, ie, decision boundary pass through the origin).

Each task N=500, each class 250 examples.

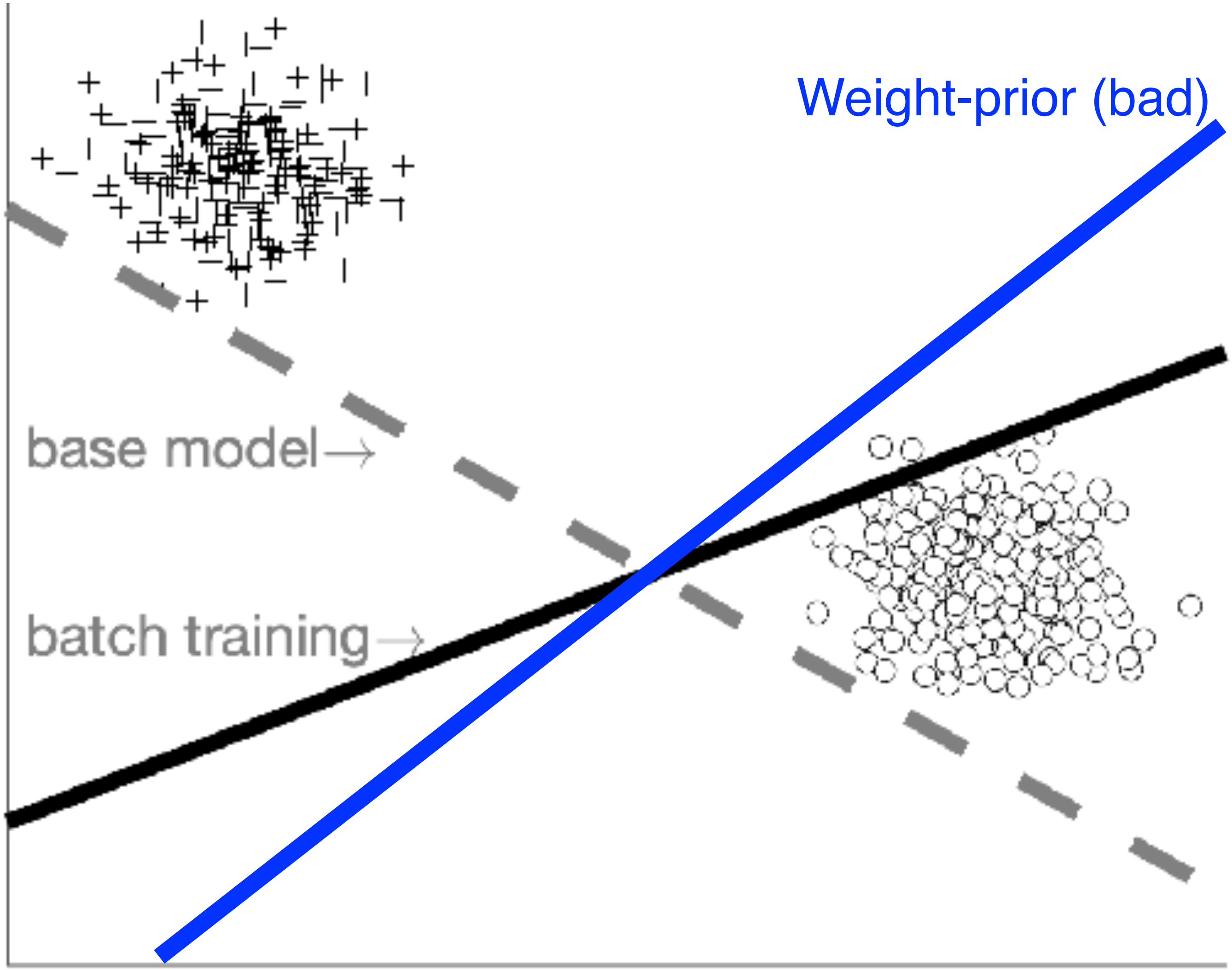
1. Kirkpatrick et al. Overcoming catastrophic forgetting in neural networks. PNAS, 2017.

Knowledge-Adaptation Priors



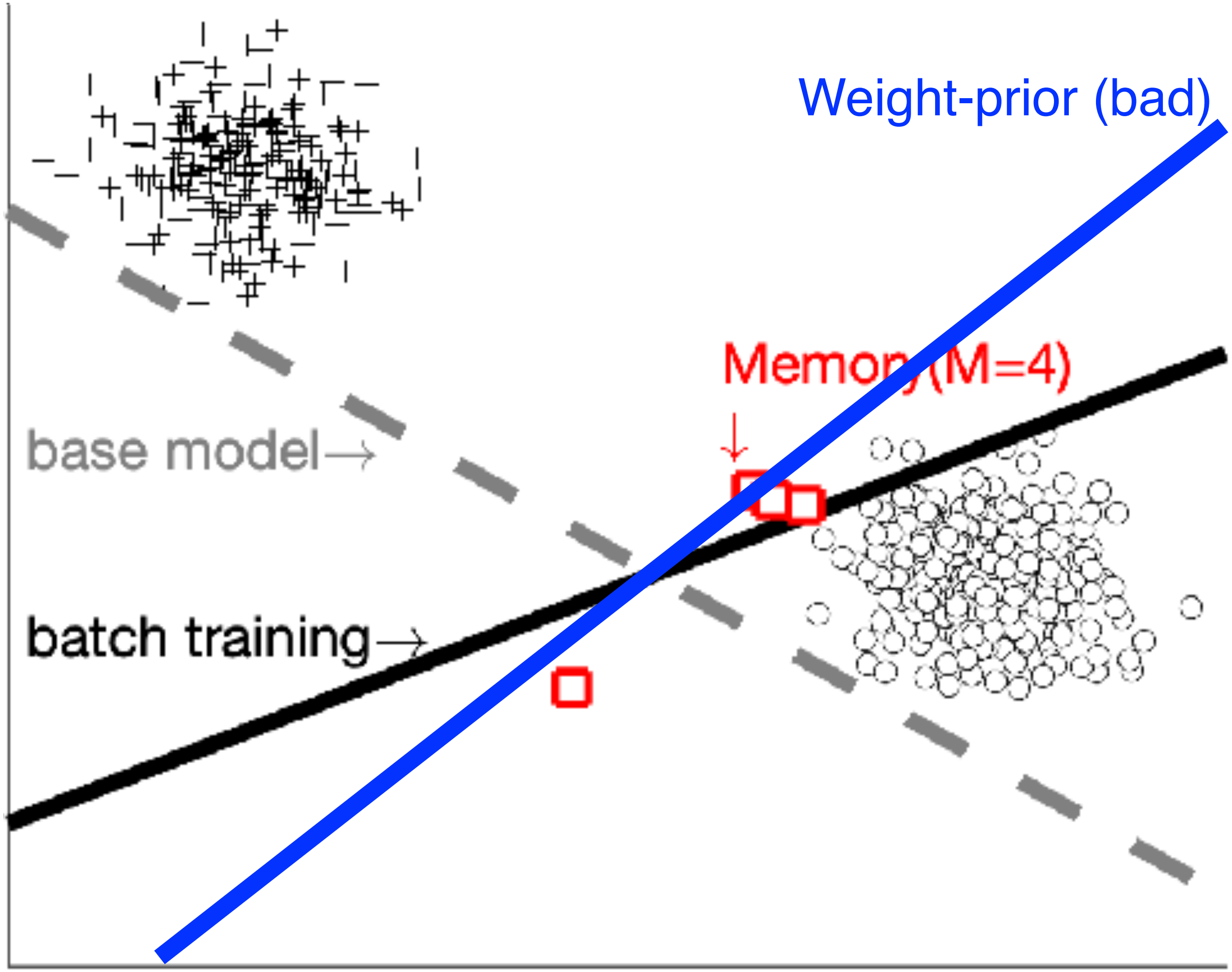
K-priors use past-memory $\mathscr{M}$ (size M) in addition to the base model.

Knowledge-Adaptation Priors



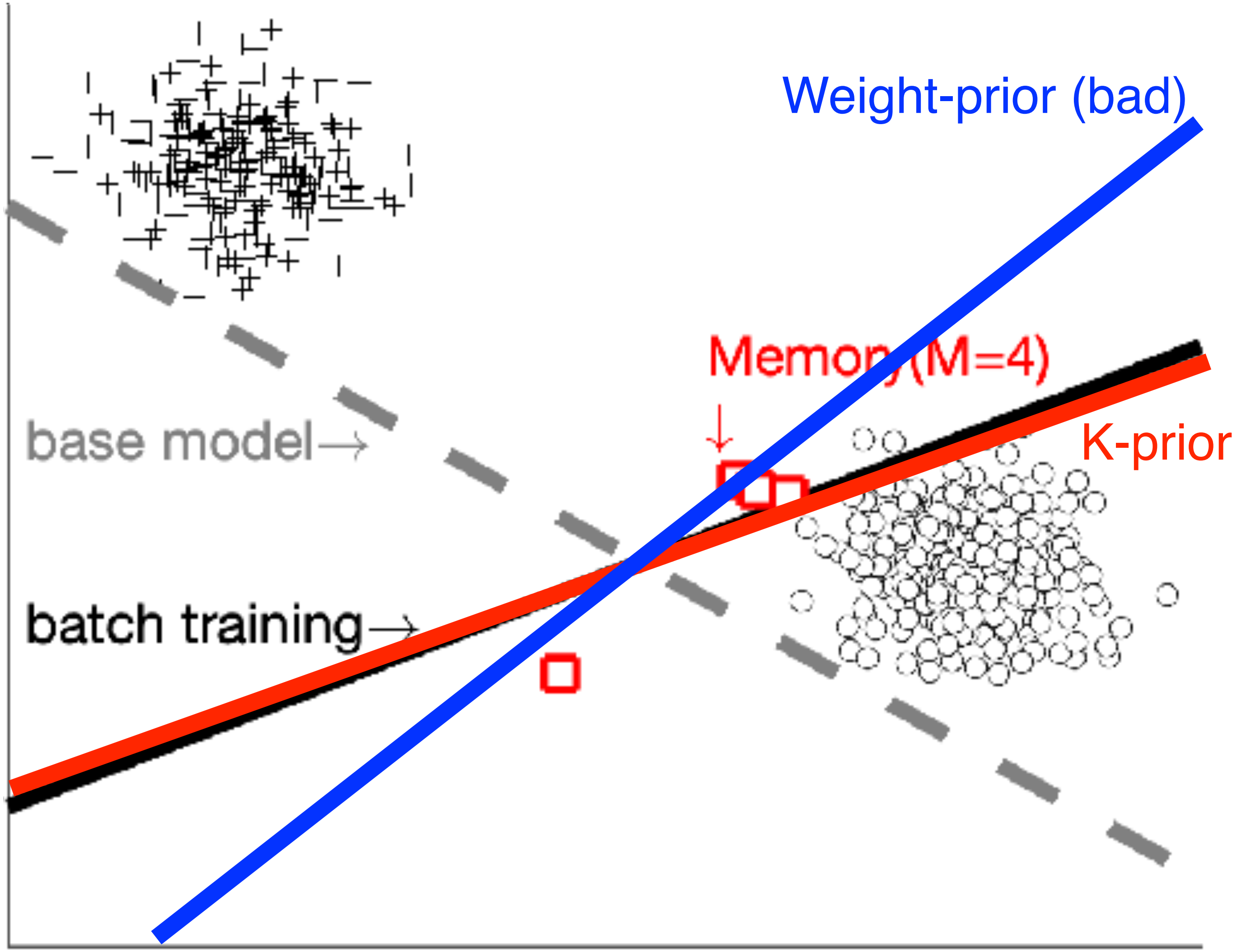
K-priors use past-memory $\mathscr{M}$ (size M) in addition to the base model.

Knowledge-Adaptation Priors



K-priors use past-memory $\mathscr{M}$ (size M) in addition to the base model.

Knowledge-Adaptation Priors



K-priors use past-memory $\mathscr{M}$ (size M) in addition to the base model.

A General Principle of Adaptation

K-priors $K(w; w_*, \mathscr{M})$ use w_* and $\mathscr{M}$

$$-\ell_k(w) + \quad \ell_j(w) + \quad \sum_{i \in \mathcal{D}} \ell_i(w) + \mathcal{R}(w) - \mathcal{R}(w) + \mathcal{G}(w)$$

A General Principle of Adaptation

K-priors $K(w; w_*, \mathscr{M})$ use w_* and $\mathscr{M}$

$$-\ell_k(w)+ \quad \ell_j(w)+ \underset{K(w; w_*, \mathscr{M})}{\cancel{\sum_{i\in\mathcal{D}} \ell_i(w) + \mathcal{R}(w)}} -\mathcal{R}(w)+\mathcal{G}(w)$$

A General Principle of Adaptation

K-priors $K(w; w_*, \mathscr{M})$ use w_* and $\mathscr{M}$

$$-\ell_k(w) + \quad \ell_j(w) + \underset{K(w; w_*, \mathscr{M})}{\cancel{\sum_{i \in \mathcal{D}} \ell_i(w) + \mathcal{R}(w)}} - \mathcal{R}(w) + \mathcal{G}(w)$$

The principle is to choose $K(w)$ and memory $\mathscr{M}$ s.t. the "gradient of the past" is faithfully reconstructed.

$$\nabla K(w) \approx \nabla \Big[\sum_{i \in \mathcal{D}} \ell_i(w) + \mathcal{R}(w) \Big]$$

K-prior Construction

Combine weight and function-space divergences

Weight-space Function-space

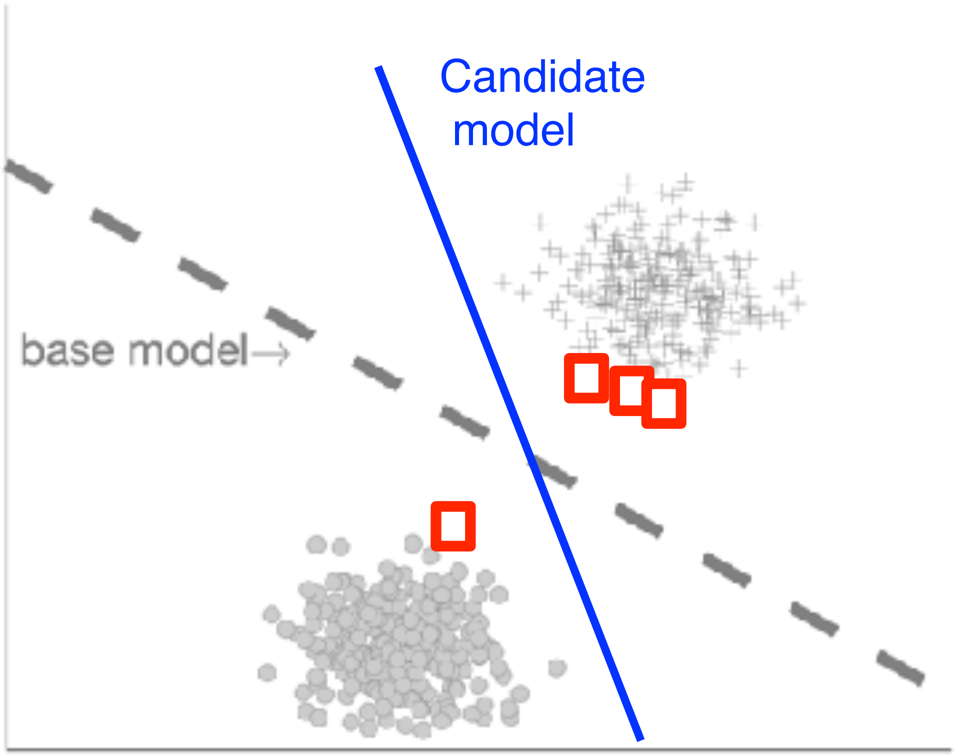
$$\mathcal{K}(w) = \tau \mathbb{D}_w(w \| w_*) + \mathbb{D}_f(\mathbf{f}(w) \| \mathbf{f}(w_*))$$

K-prior Construction

Combine weight and function-space divergences

Weight-space Function-space

$$\mathcal{K}(w) = \tau \mathbb{D}_w(w\|w_*) + \mathbb{D}_f(\mathbf{f}(w)\|\mathbf{f}(w_*))$$

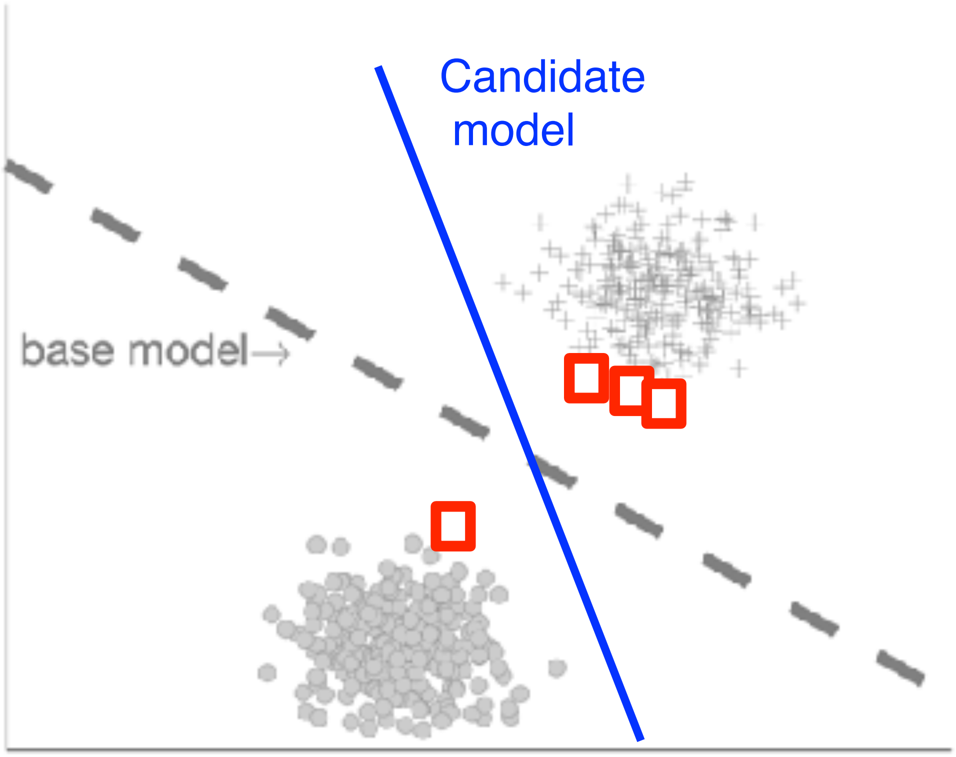


$$\begin{bmatrix} f_w^1 \\ f_w^2 \\ f_w^3 \\ f_w^4 \end{bmatrix} \quad \begin{bmatrix} f_{w_*}^1 \\ f_{w_*}^2 \\ f_{w_*}^3 \\ f_{w_*}^4 \end{bmatrix}$$

K-prior Construction

Combine weight and function-space divergences

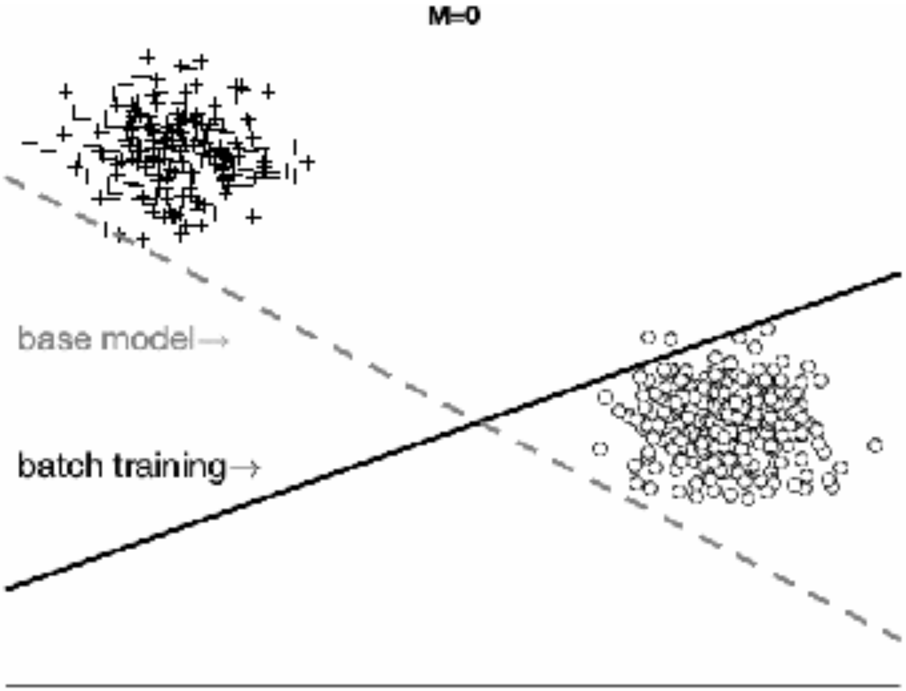
$$\mathcal{K}(w) = \overset{\text{Weight-space}}{\tau\mathbb{D}_w(w\|w_*)} + \overset{\text{Function-space}}{\mathbb{D}_f(\mathbf{f}(w)\|\mathbf{f}(w_*))}$$

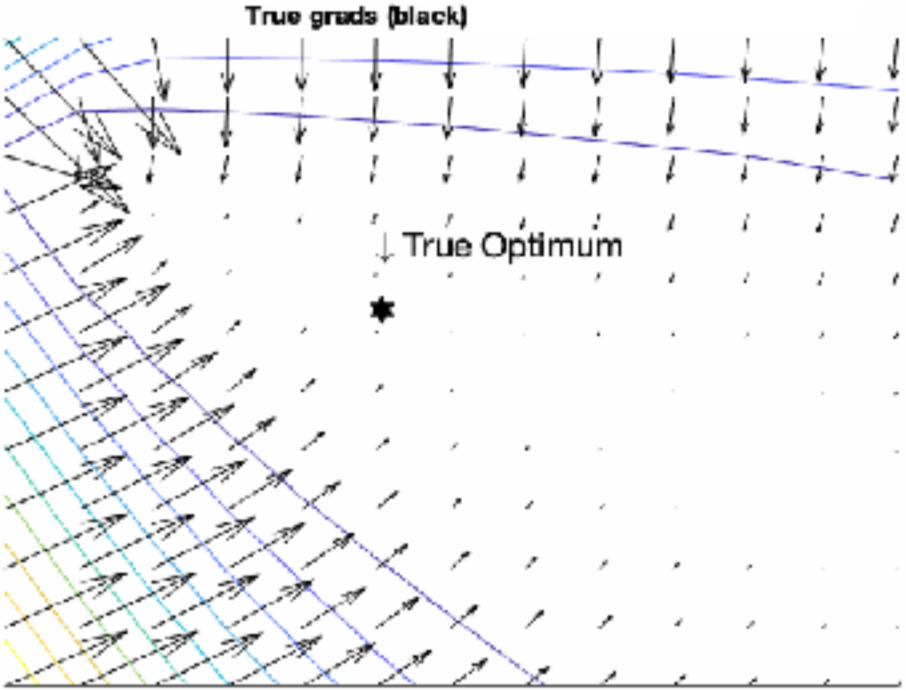


$$\begin{bmatrix} f_w^1 \\ f_w^2 \\ f_w^3 \\ f_w^4 \end{bmatrix} \quad \begin{bmatrix} f_{w_*}^1 \\ f_{w_*}^2 \\ f_{w_*}^3 \\ f_{w_*}^4 \end{bmatrix}$$

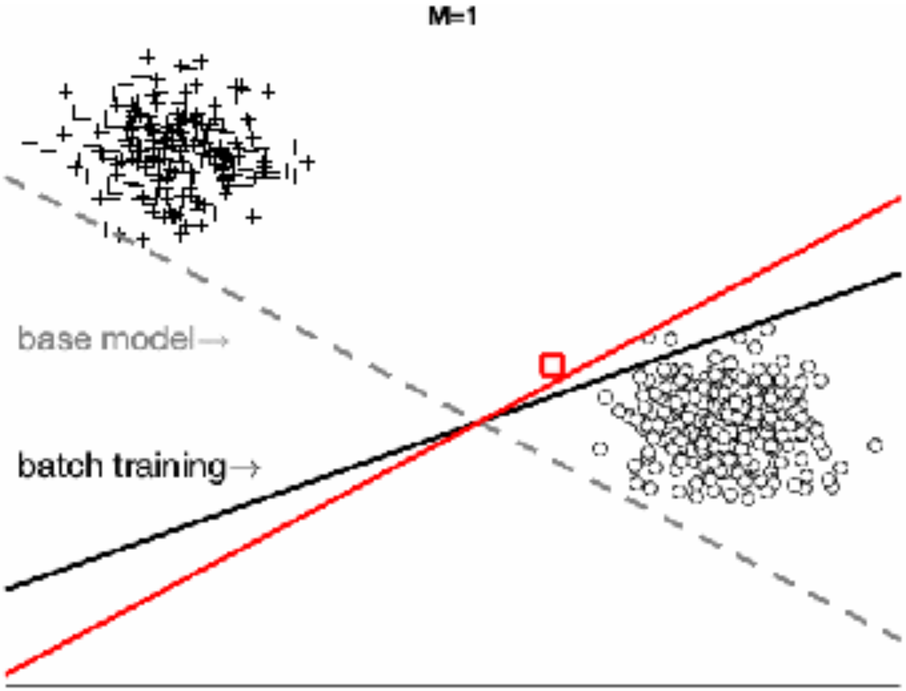
No labels required, so $\mathscr{M}$ can include any inputs!

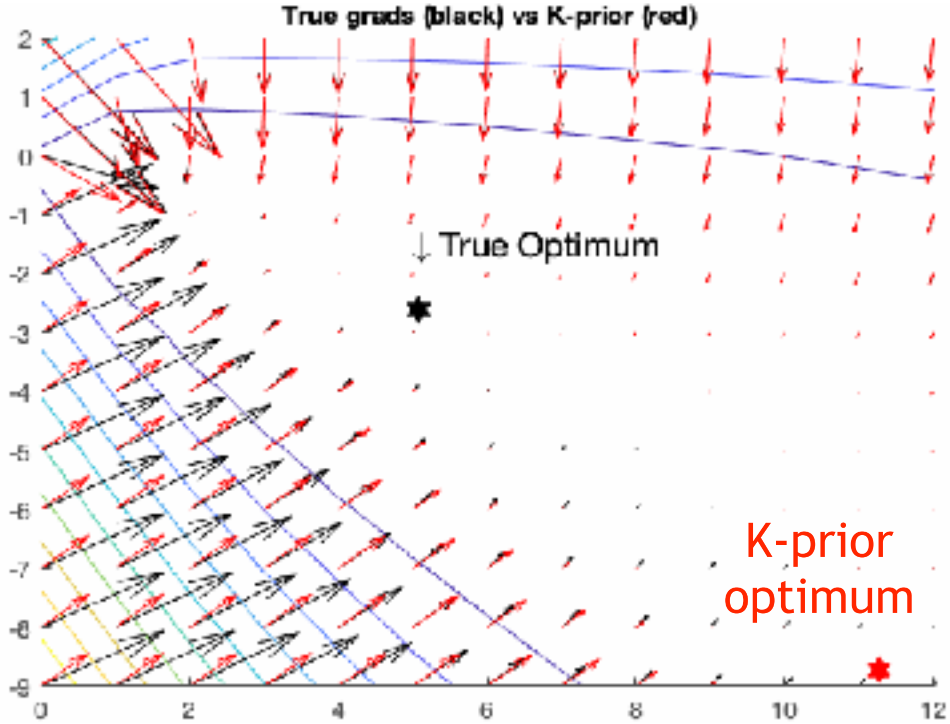
Faithful Gradient Reconstruction





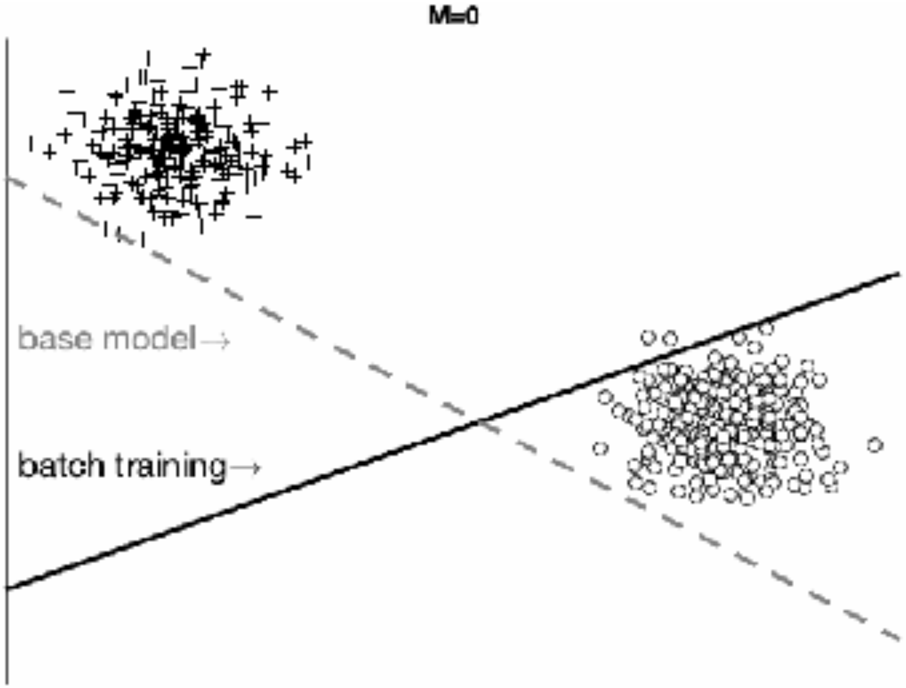
Faithful Gradient Reconstruction

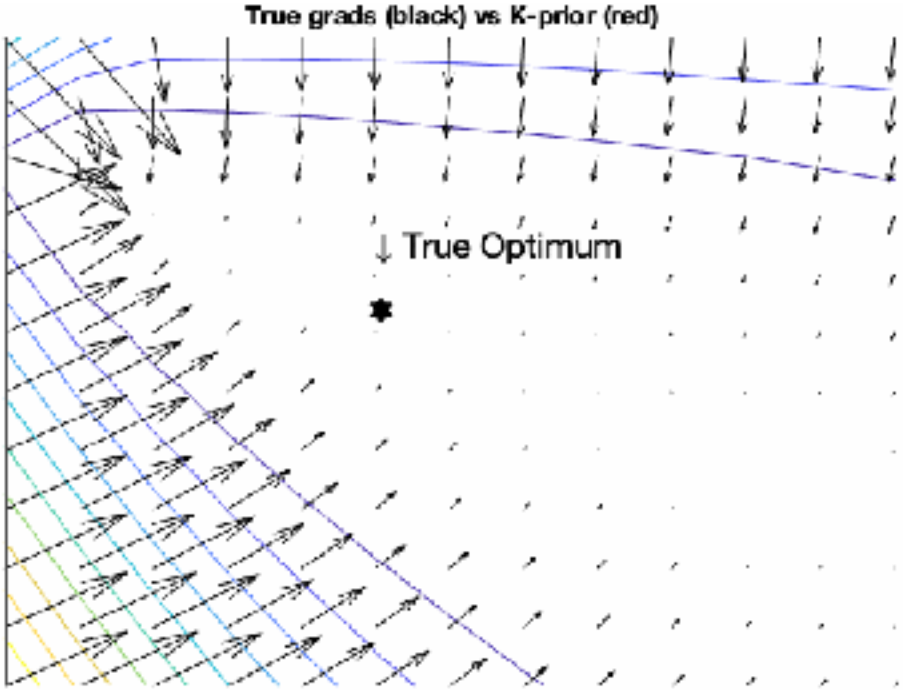




K-prior optimum

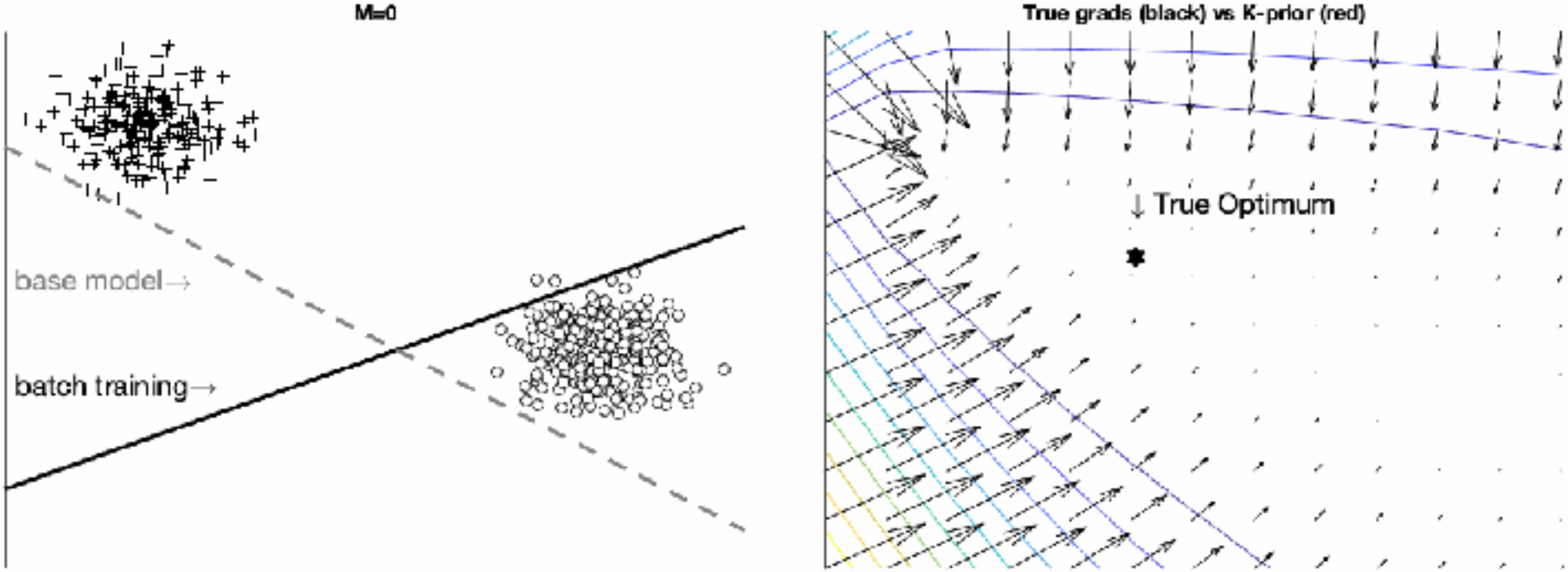
Faithful Gradient Reconstruction





No labels required, so $\mathscr{M}$ can include any inputs!

Faithful Gradient Reconstruction



No labels required, so $\mathscr{M}$ can include any inputs!

Exact Gradient Reconstruction

Consider logistic regression $f_w^i = \phi_i^\top w$

$$\bar{l}(w) = \sum_{i \in \mathcal{D}} \ell(y_i, \sigma(f_w^i)) + \delta \|w\|^2$$

Exact Gradient Reconstruction

Consider logistic regression $f_w^i = \phi_i^\top w$

$$\bar{l}(w) = \sum_{i \in \mathcal{D}} \ell(y_i, \sigma(f_w^i)) + \delta \|w\|^2$$

Function-space

Weight-space

$$\mathcal{K}(w) = \sum_{i \in \mathcal{D}} \ell(\sigma(f_{w_*}^i), \sigma(f_w^i)) + \delta \|w - w_*\|^2$$

Exact Gradient Reconstruction

Consider logistic regression $f_w^i = \phi_i^\top w$

$$\bar{l}(w) = \sum_{i \in \mathcal{D}} \ell(y_i, \sigma(f_w^i)) + \delta \|w\|^2$$

Function-space

Weight-space

$$\mathcal{K}(w) = \sum_{i \in \mathcal{D}} \ell(\sigma(f_{w_*}^i), \sigma(f_w^i)) + \delta \|w - w_*\|^2$$

Memory = all past data

Exact Gradient Reconstruction

Consider logistic regression $f_w^i = \phi_i^\top w$

$$\bar{l}(w) = \sum_{i \in \mathcal{D}} \ell(y_i, \sigma(f_w^i)) + \delta \|w\|^2$$

Function-space

Weight-space

$$\mathcal{K}(w) = \sum_{i \in \mathcal{D}} \ell(\sigma(f_{w_*}^i), \sigma(f_w^i)) + \delta \|w - w_*\|^2$$

Memory = all past data

The K-prior recovers the exact gradients!

Exact Gradient Reconstruction

Consider logistic regression $f_w^i = \phi_i^\top w$

$$\bar{l}(w) = \sum_{i \in \mathcal{D}} \ell(y_i, \sigma(f_w^i)) + \delta \|w\|^2$$

Function-space Weight-space

$$\mathcal{K}(w) = \sum_{i \in \mathcal{D}} \ell(\sigma(f_{w_*}^i), \sigma(f_w^i)) + \delta \|w - w_*\|^2$$

Memory = all past data

The K-prior recovers the exact gradients!

$$\nabla \mathcal{K}(w) = \sum_{i \in \mathcal{D}} \phi_i (\sigma(f_w^i) - \sigma(f_{w_*}^i)) + \delta (w - w_*)$$

Exact Gradient Reconstruction

Consider logistic regression $f_w^i = \phi_i^\top w$

$$\bar{l}(w) = \sum_{i \in \mathcal{D}} \ell(y_i, \sigma(f_w^i)) + \delta \|w\|^2$$

Function-space

Weight-space

$$\mathcal{K}(w) = \sum_{i \in \mathcal{D}} \ell(\sigma(f_{w_*}^i), \sigma(f_w^i)) + \delta \|w - w_*\|^2$$

Memory = all past data

The K-prior recovers the exact gradients!

$$\nabla \mathcal{K}(w) = \sum_{i \in \mathcal{D}} \phi_i(\sigma(f_w^i) \underset{-y_i + y_i}{-} \sigma(f_{w_*}^i)) + \delta(w - w_*)$$

Exact Gradient Reconstruction

Consider logistic regression $f_w^i = \phi_i^\top w$

$$\bar{l}(w) = \sum_{i \in \mathcal{D}} \ell(y_i, \sigma(f_w^i)) + \delta \|w\|^2$$

Function-space Weight-space

$$\mathcal{K}(w) = \sum_{i \in \mathcal{D}} \ell(\sigma(f_{w_*}^i), \sigma(f_w^i)) + \delta \|w - w_*\|^2$$

Memory = all past data

The K-prior recovers the exact gradients!

$$\nabla \mathcal{K}(w) = \sum_{i \in \mathcal{D}} \phi_i(\sigma(f_w^i) \underset{-y_i + y_i}{-} \sigma(f_{w_*}^i)) + \delta(w - w_*)$$

$$= \sum_{i \in \mathcal{D}} \phi_i(\sigma(f_w^i) - y_i) + \delta w - \sum_{i \in \mathcal{D}} \phi_i(\sigma(f_{w_*}^i) - y_i) - \delta w_*$$

Exact Gradient Reconstruction

Consider logistic regression $f_w^i = \phi_i^\top w$

$$\bar{l}(w) = \sum_{i \in \mathcal{D}} \ell(y_i, \sigma(f_w^i)) + \delta \|w\|^2$$

Function-space Weight-space

$$\mathcal{K}(w) = \sum_{i \in \mathcal{D}} \ell(\sigma(f_{w_*}^i), \sigma(f_w^i)) + \delta \|w - w_*\|^2$$

Memory = all past data

The K-prior recovers the exact gradients!

$$\nabla \mathcal{K}(w) = \sum_{i \in \mathcal{D}} \phi_i(\sigma(f_w^i) \underset{-y_i + y_i}{-} \sigma(f_{w_*}^i)) + \delta(w - w_*)$$

$$= \sum_{i \in \mathcal{D}} \phi_i(\sigma(f_w^i) - y_i) + \delta w \boxed{- \sum_{i \in \mathcal{D}} \phi_i(\sigma(f_{w_*}^i) - y_i) - \delta w_*}$$

$$\nabla \bar{l}(w_*) = 0$$

Exact Gradient Reconstruction

Consider logistic regression $f_w^i = \phi_i^\top w$

$$\bar{l}(w) = \sum_{i\in\mathcal{D}} \ell(y_i, \sigma(f_w^i)) + \delta\|w\|^2$$

Function-space

Weight-space

$$\mathcal{K}(w) = \sum_{i\in\mathcal{D}} \ell(\sigma(f_{w_*}^i), \sigma(f_w^i)) + \delta\|w - w_*\|^2$$

Memory = all past data

The K-prior recovers the exact gradients!

$$\nabla\mathcal{K}(w) = \sum_{i\in\mathcal{D}} \phi_i(\sigma(f_w^i) \underset{-y_i+y_i}{-} \sigma(f_{w_*}^i)) + \delta(w - w_*)$$

$$= \sum_{i\in\mathcal{D}} \phi_i(\sigma(f_w^i) - y_i) + \delta w \boxed{\cancel{-\sum_{i\in\mathcal{D}} \phi_i(\sigma(f_{w_*}^i) - y_i) - \delta w_*}}$$

$$\nabla\bar{l}(w_*) = 0$$

Exact Gradient Reconstruction

Consider logistic regression $f_w^i = \phi_i^\top w$

$$\bar{l}(w) = \sum_{i\in\mathcal{D}} \ell(y_i, \sigma(f_w^i)) + \delta\|w\|^2$$

Function-space

Weight-space

$$\mathcal{K}(w) = \sum_{i\in\mathcal{D}} \ell(\sigma(f_{w_*}^i), \sigma(f_w^i)) + \delta\|w - w_*\|^2$$

Memory = all past data

The K-prior recovers the exact gradients!

$$\nabla\mathcal{K}(w) = \sum_{i\in\mathcal{D}} \phi_i(\sigma(f_w^i) \underset{-y_i + y_i}{-} \sigma(f_{w_*}^i)) + \delta(w - w_*)$$

$$= \boxed{\sum_{i\in\mathcal{D}} \phi_i(\sigma(f_w^i) - y_i) + \delta w} \; \boxed{\cancel{\sum_{i\in\mathcal{D}} \phi_i(\sigma(f_{w_*}^i) - y_i) - \delta w_*}}$$

$\nabla\bar{l}(w)$ $\nabla\bar{l}(w_*) = 0$

How to Choose Memory?

Memory should contain points where the (unknown) future and past models disagree the most

$$\nabla \bar{l}(w) - \nabla K(w) = \sum_{i \in \mathcal{D} \setminus \mathcal{M}} \phi_i (\sigma(f_w^i) - \sigma(f_{w_*}^i))$$

Prediction disagreement

How to Choose Memory?

Memory should contain points where the (unknown) future and past models disagree the most

$$\nabla \bar{l}(w) - \nabla K(w) = \sum_{i \in \mathcal{D} \backslash \mathcal{M}} \phi_i \left(\sigma(f_w^i) - \sigma(f_{w_*}^i)\right)$$

Prediction disagreement

$$\approx \underbrace{\left[\sum_{i \in \mathcal{D} \backslash \mathcal{M}} \phi_i \underbrace{\sigma'(f_{w_*}^i)}_{\text{2nd derivative of the loss}} \phi_i^\top \right]}_{\substack{\text{Generalized Gauss-Newton (GGN)} \\ \text{Independent of w}}} (w - w_*)$$

How to Choose Memory?

Memory should contain points where the (unknown) future and past models disagree the most

$$\nabla \bar{l}(w) - \nabla K(w) = \sum_{i \in \mathcal{D} \setminus \mathcal{M}} \phi_i(\sigma(f_w^i) - \sigma(f_{w_*}^i))$$

Prediction disagreement

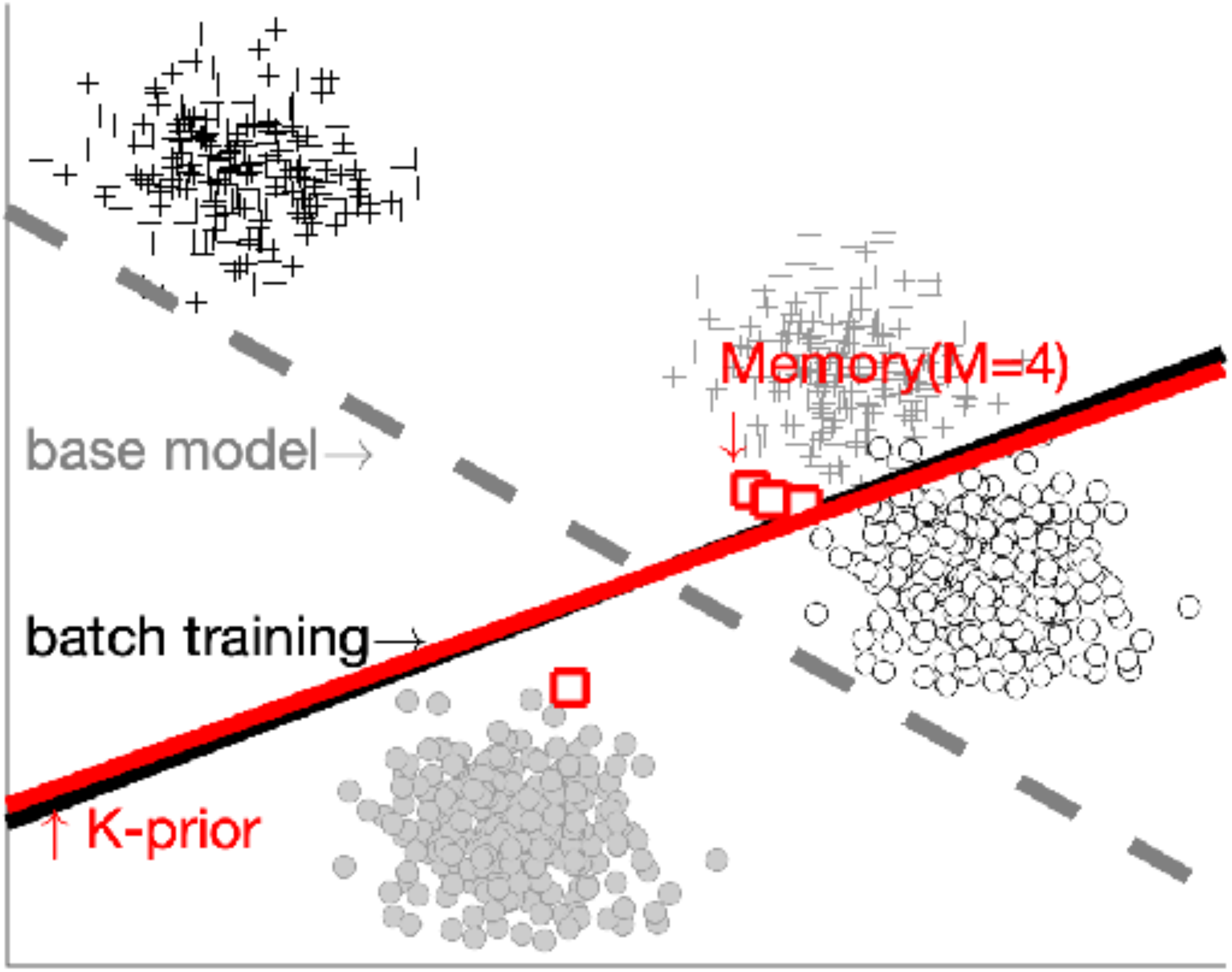
$$\approx \underbrace{\left[\sum_{i \in \mathcal{D} \setminus \mathcal{M}} \phi_i \underbrace{\sigma'(f_{w_*}^i)}_{\text{2nd derivative of the loss}} \phi_i^\top \right]}_{\substack{\text{Generalized Gauss-Newton (GGN)} \\ \text{Independent of w}}} (w - w_*)$$

Pick points to minimize the GGN approximations. We can use any low-rank approximation. We pick top-M $\sigma'(f_{w_*}^i)$ which is called memorable past [1].

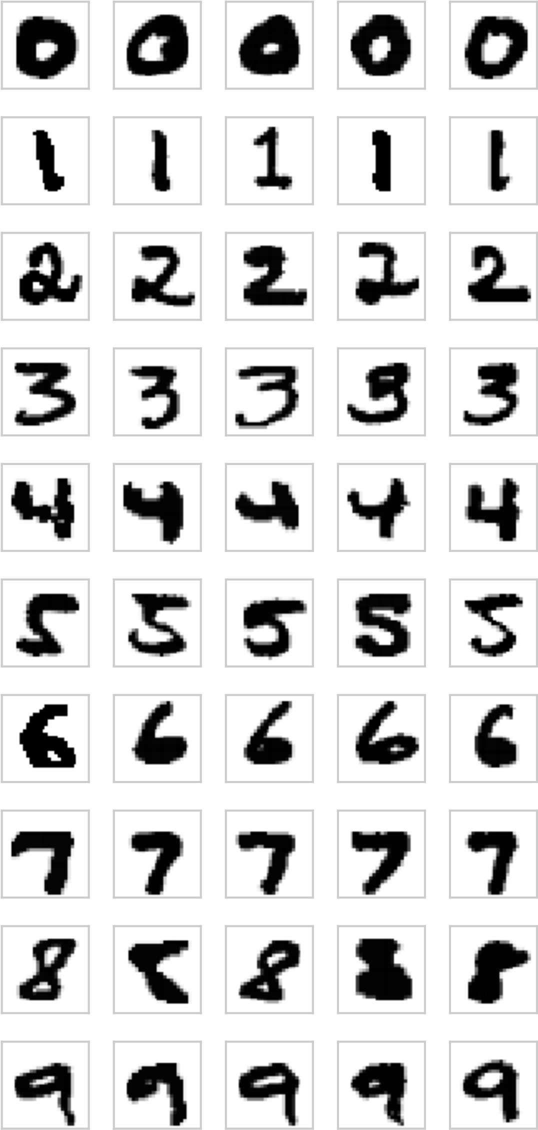
---

[1] Pan et al. Continual deep learning by functional regularisation of memorable past. NeurIPS, 2020.

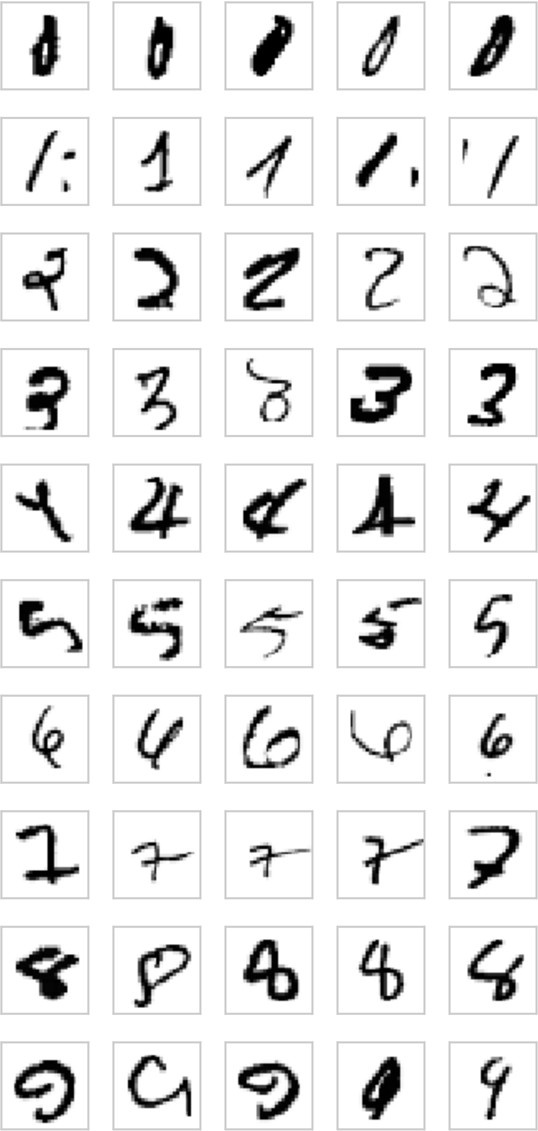
Memorable Past: Example



Least Memorable



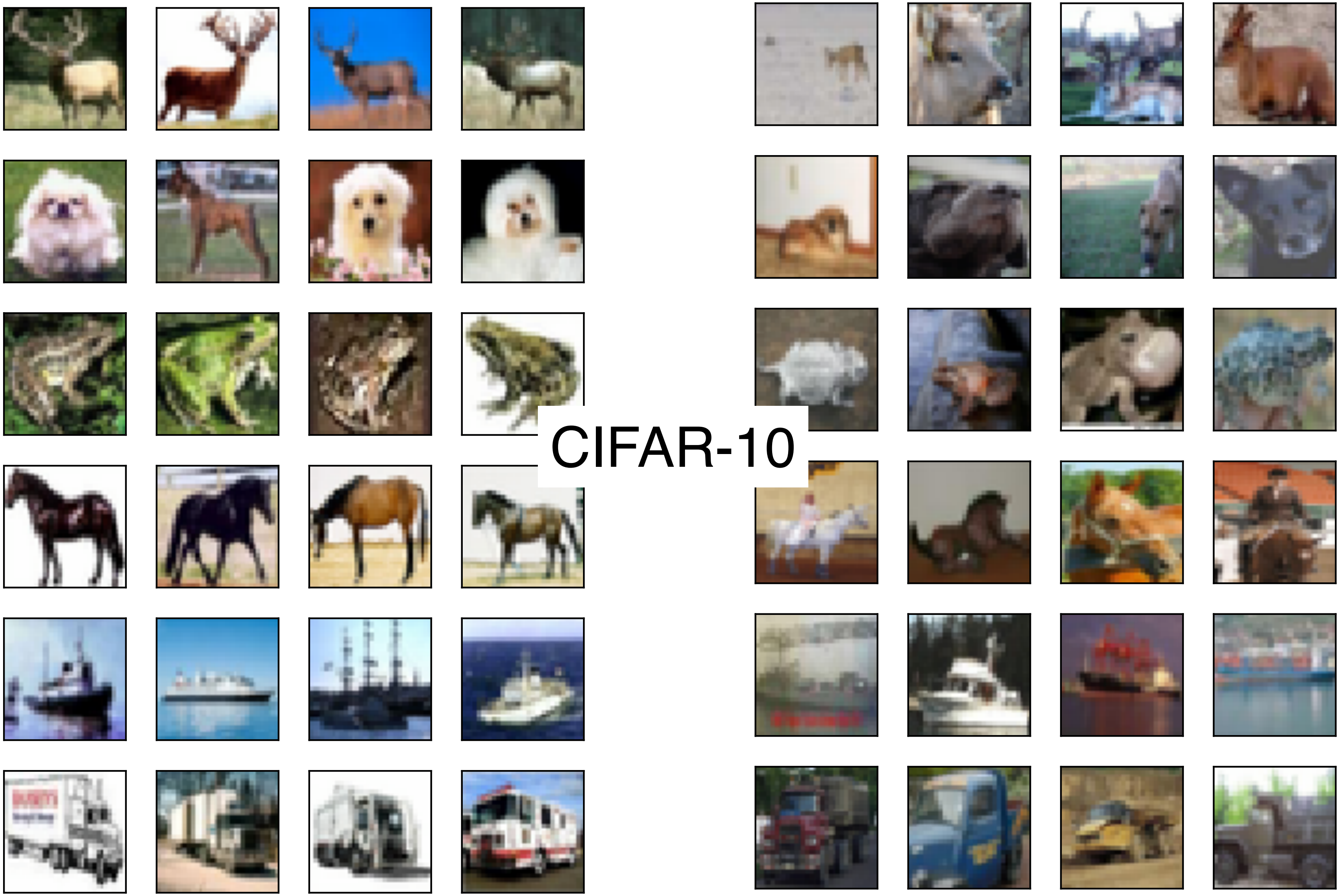
Most Memorable



MNIST

1. Pan et al. Continual deep learning by functional regularisation of memorable past. NeurIPS, 2020.

Least Memorable Most Memorable



1. Pan et al. Continual deep learning by functional regularisation of memorable past. NeurIPS, 2020.

Existing Work

K-priors unify many seemingly unrelated existing work, and provide speed-accuracy trade-off

	Wide
Weight priors [1]	✗
SVMs [2]	✗
Knowledge Distillation [3]	✗
Learning under privileged info [4]	✗
Gaussian Process [5]	✗
Memory-based CL [6]	✗

1. Kirkpatrick et al. Overcoming catastrophic forgetting in neural networks. PNAS, 2017.
2. Cauwenberghs and Poggio. Incremental and decremental SVM learning. NeurIPS, 2001.
3. Hinton et al. Distilling the knowledge in a neural network, arXiv, 2015.
4. Vapnik and Izmailov. Learning using privileged information: similarity control and …. JMLR, 2015.
5. Csató and Opper. Sparse on-line Gaussian processes. Neural computation, 2002.
6. Pan et al. Continual deep learning by functional regularisation of memorable past. NeurIPS, 2020.

Existing Work

K-priors unify many seemingly unrelated existing work, and provide speed-accuracy trade-off

	Wide
Weight priors [1]	✗
SVMs [2]	✗
Knowledge Distillation [3]	✗
Learning under privileged info [4]	✗
Gaussian Process [5]	✗
Memory-based CL [6]	✗
K-priors	✓

1. Kirkpatrick et al. Overcoming catastrophic forgetting in neural networks. PNAS, 2017.
2. Cauwenberghs and Poggio. Incremental and decremental SVM learning. NeurIPS, 2001.
3. Hinton et al. Distilling the knowledge in a neural network, arXiv, 2015.
4. Vapnik and Izmailov. Learning using privileged information: similarity control and …. JMLR, 2015.
5. Csató and Opper. Sparse on-line Gaussian processes. Neural computation, 2002.
6. Pan et al. Continual deep learning by functional regularisation of memorable past. NeurIPS, 2020.

Existing Work

K-priors unify many seemingly unrelated existing work, and provide speed-accuracy trade-off

	Wide	Accurate	Quick
Weight priors [1]	✗	✗	✓
SVMs [2]	✗	✓	✗
Knowledge Distillation [3]	✗	✓	✗
Learning under privileged info [4]	✗	✓	✗
Gaussian Process [5]	✗	✓	✗
Memory-based CL [6]	✗	✓	✓
K-priors	✓		

} Require storing all past data

1. Kirkpatrick et al. Overcoming catastrophic forgetting in neural networks. PNAS, 2017.
2. Cauwenberghs and Poggio. Incremental and decremental SVM learning. NeurIPS, 2001.
3. Hinton et al. Distilling the knowledge in a neural network, arXiv, 2015.
4. Vapnik and Izmailov. Learning using privileged information: similarity control and …. JMLR, 2015.
5. Csató and Opper. Sparse on-line Gaussian processes. Neural computation, 2002.
6. Pan et al. Continual deep learning by functional regularisation of memorable past. NeurIPS, 2020.

Existing Work

K-priors unify many seemingly unrelated existing work, and provide speed-accuracy trade-off

	Wide	Accurate	Quick
Weight priors [1]	✗	✗	✓
SVMs [2]	✗	✓	✗
Knowledge Distillation [3]	✗	✓	✗
Learning under privileged info [4]	✗	✓	✗
Gaussian Process [5]	✗	✓	✗
Memory-based CL [6]	✗	✓	✓
K-priors	✓	✓	✓

}
Require storing all past data

1. Kirkpatrick et al. Overcoming catastrophic forgetting in neural networks. PNAS, 2017.
2. Cauwenberghs and Poggio. Incremental and decremental SVM learning. NeurIPS, 2001.
3. Hinton et al. Distilling the knowledge in a neural network, arXiv, 2015.
4. Vapnik and Izmailov. Learning using privileged information: similarity control and …. JMLR, 2015.
5. Csató and Opper. Sparse on-line Gaussian processes. Neural computation, 2002.
6. Pan et al. Continual deep learning by functional regularisation of memorable past. NeurIPS, 2020.

Knowledge Distillation (KD)

K-priors with no weight-div and temperature set to 1, gives us KD. Gradients are not exact now.

$$\nabla K(w) = \sum_{i \in \mathcal{D}} \nabla f_w^i (\sigma(f_w^i) - y_i) - \sum_{i \in \mathcal{D}} \nabla f_w^i \underset{\uparrow}{r}{}_{w_*}^i$$

Residuals $f_{w_*}^i - y_i$

1. Hinton et al. Distilling the knowledge in a neural network, arXiv, 2015.
2. Vapnik and Izmailov. Learning using privileged information: similarity control and …. JMLR, 2015.

Knowledge Distillation (KD)

K-priors with no weight-div and temperature set to 1, gives us KD. Gradients are not exact now.

$$\nabla K(w) = \sum_{i \in \mathcal{D}} \nabla f_w^i (\sigma(f_w^i) - y_i) - \sum_{i \in \mathcal{D}} \nabla f_w^i r_{w_*}^i$$

Residuals $f_{w_*}^i - y_i$

“Avoid past mistakes of the teacher”. Very similar to using “slack” in SVM [2] to improve student’s learning.

1. Hinton et al. Distilling the knowledge in a neural network, arXiv, 2015.
2. Vapnik and Izmailov. Learning using privileged information: similarity control and …. JMLR, 2015.

Knowledge Distillation (KD)

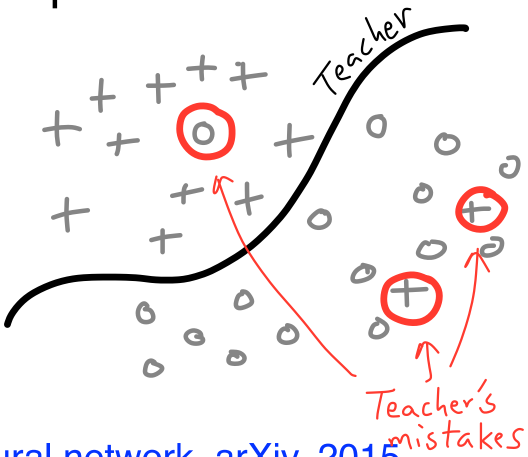
K-priors with no weight-div and temperature set to 1, gives us KD. Gradients are not exact now.

$$\nabla K(w) = \sum_{i \in \mathcal{D}} \nabla f_w^i (\sigma(f_w^i) - y_i) - \sum_{i \in \mathcal{D}} \nabla f_w^i r_{w_*}^i$$

Residuals $f_{w_*}^i - y_i$

"Avoid past mistakes of the teacher". Very similar to using "slack" in SVM [2] to improve student's learning.

Teacher's mistakes provided to the student



1. Hinton et al. Distilling the knowledge in a neural network, arXiv, 2015.
2. Vapnik and Izmailov. Learning using privileged information: similarity control and …. JMLR, 2015.

Knowledge Distillation (KD)

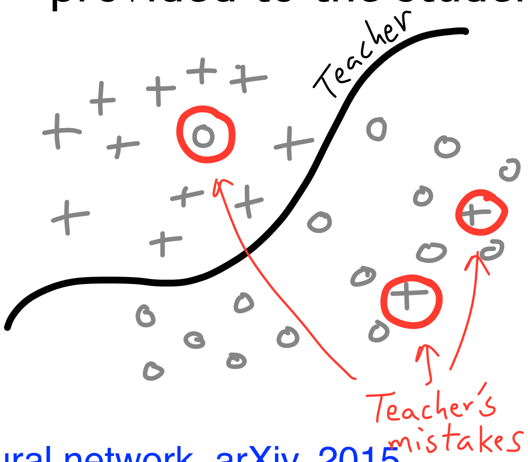
K-priors with no weight-div and temperature set to 1, gives us KD. Gradients are not exact now.

$$\nabla K(w) = \sum_{i\in\mathcal{D}} \nabla f^i_w(\sigma(f^i_w) - y_i) - \sum_{i\in\mathcal{D}} \nabla f^i_w r^i_{w_*}$$

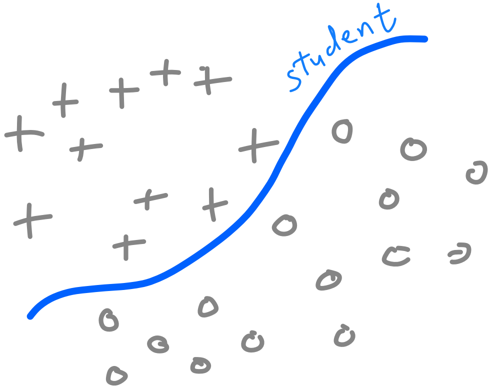
Residuals $f^i_{w_*} - y_i$

"Avoid past mistakes of the teacher". Very similar to using "slack" in SVM [2] to improve student's learning.

Teacher's mistakes provided to the student



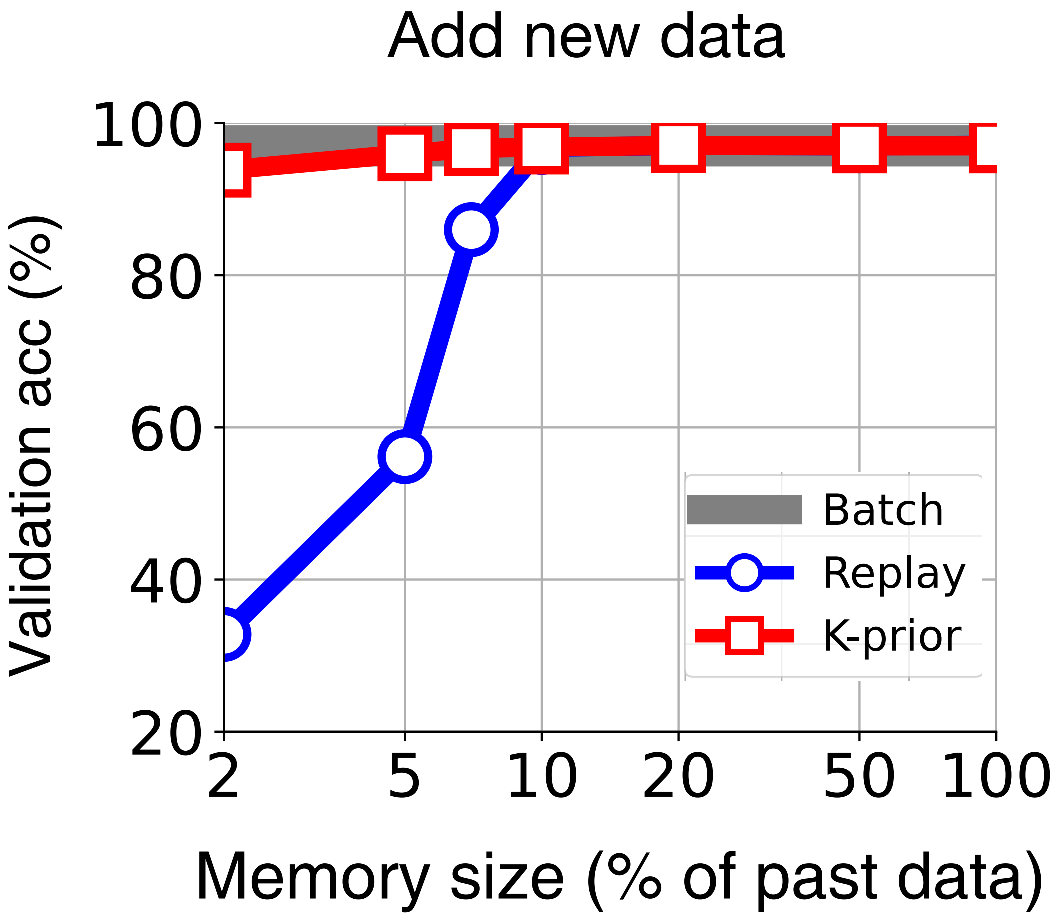
Student solves a simpler problem



1. Hinton et al. Distilling the knowledge in a neural network, arXiv, 2015.
2. Vapnik and Izmailov. Learning using privileged information: similarity control and …. JMLR, 2015.

Results

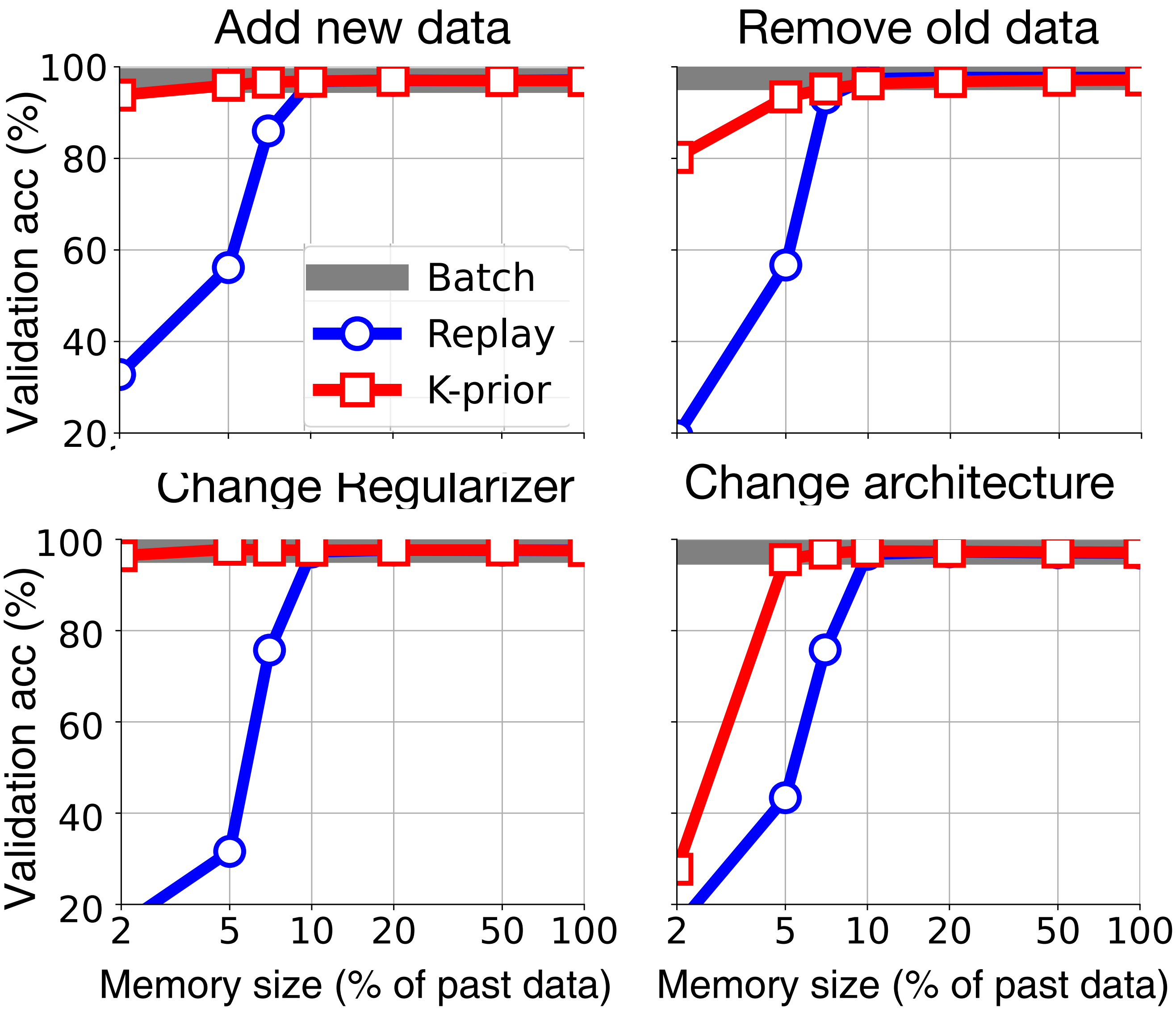
K-priors need < 2% of past data to match “batch”.



The results are on USPS binary classification with Neural nets.

“Replay” uses the same memory but with true outputs.

Results



K-priors only need about 2-5% of the past data to match retraining on full batch.

The results are on USPS binary classification with Neural nets.

Future Directions

Future Directions

- The general principle of adaptation in K-priors is to faithfully reconstruct “past gradients”

Future Directions

- The general principle of adaptation in K-priors is to faithfully reconstruct “past gradients”
- This is an instance of a more general Bayesian principle to reconstruct “past natural parameters” of the posterior approx.

Future Directions

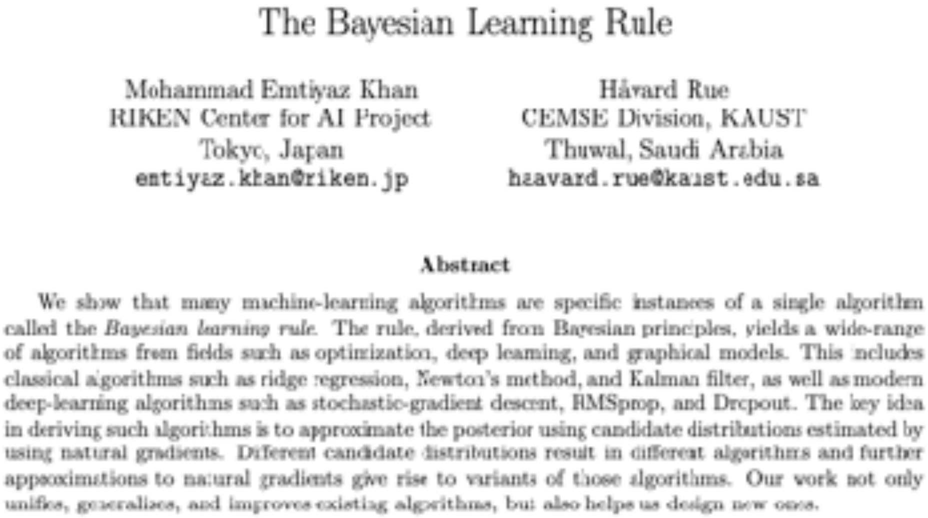
- The general principle of adaptation in K-priors is to faithfully reconstruct “past gradients”
- This is an instance of a more general Bayesian principle to reconstruct “past natural parameters” of the posterior approx.
  - K-prior is a first-order approx. (Gaussian with unknown mean)

Future Directions

- The general principle of adaptation in K-priors is to faithfully reconstruct “past gradients”
- This is an instance of a more general Bayesian principle to reconstruct “past natural parameters” of the posterior approx.
  - K-prior is a first-order approx. (Gaussian with unknown mean)
  - Extend with posteriors with higher-order sufficient statistics (Gaussian with unknown covariance)

Future Directions

- The general principle of adaptation in K-priors is to faithfully reconstruct "past gradients"
- This is an instance of a more general Bayesian principle to reconstruct "past natural parameters" of the posterior approx.
  - K-prior is a first-order approx. (Gaussian with unknown mean)
  - Extend with posteriors with higher-order sufficient statistics (Gaussian with unknown covariance)



The Bayesian Learning Rule

Mohammad Emtiyaz Khan
RIKEN Center for AI Project
Tokyo, Japan
`emtiyaz.khan@riken.jp`

Håvard Rue
CEMSE Division, KAUST
Thuwal, Saudi Arabia
`haavard.rue@kaust.edu.sa`

Abstract

We show that many machine-learning algorithms are specific instances of a single algorithm called the *Bayesian learning rule*. The rule, derived from Bayesian principles, yields a wide-range of algorithms from fields such as optimization, deep learning, and graphical models. This includes classical algorithms such as ridge regression, Newton's method, and Kalman filter, as well as modern deep-learning algorithms such as stochastic-gradient descent, RMSprop, and Dropout. The key idea in deriving such algorithms is to approximate the posterior using candidate distributions estimated by using natural gradients. Different candidate distributions result in different algorithms and further approximations to natural gradients give rise to variants of those algorithms. Our work not only unifies, generalizes, and improves existing algorithms, but also helps us design new ones.

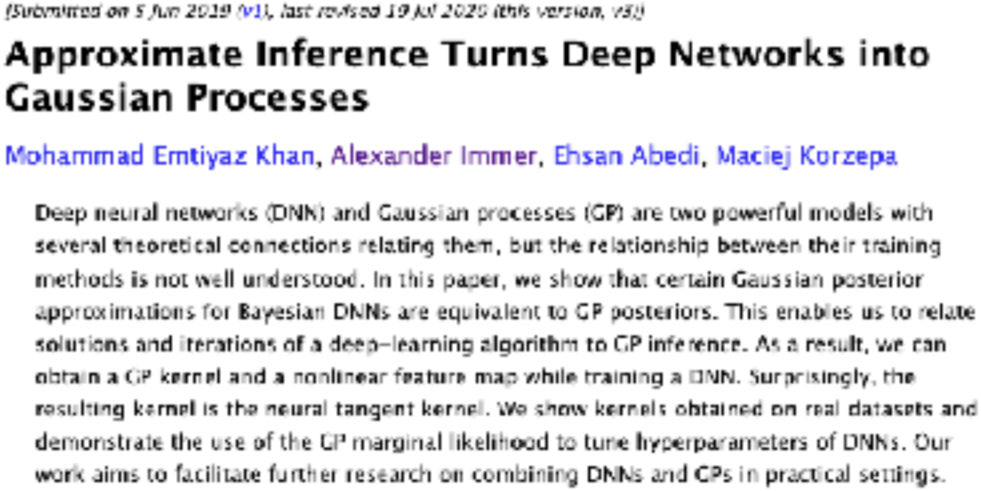
Future Directions

Future Directions

- Another challenge is what to store and how much memory to allocate
  - Inherent trade-off between speed and accuracy
  - We “have” to reasonable assumptions about the future
  - The “dual space” of the “divergence” plays a key role

Future Directions

- Another challenge is what to store and how much memory to allocate
  - Inherent trade-off between speed and accuracy
  - We “have” to reasonable assumptions about the future
  - The “dual space” of the “divergence” plays a key role
- We are developing “dual representations” are used for Knowledge representation, transfer, and collection
  - A new paper on “memorable past” coming soon

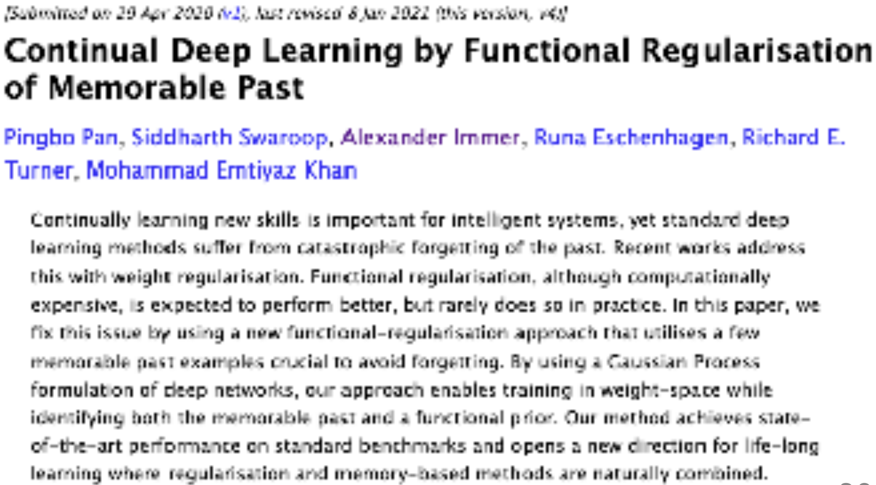


[Submitted on 5 Jun 2019 (v1), last revised 19 Jul 2020 (this version, v3)]

Approximate Inference Turns Deep Networks into Gaussian Processes

Mohammad Emtiyaz Khan, Alexander Immer, Ehsan Abedi, Maciej Korzepa

Deep neural networks (DNN) and Gaussian processes (GP) are two powerful models with several theoretical connections relating them, but the relationship between their training methods is not well understood. In this paper, we show that certain Gaussian posterior approximations for Bayesian DNNs are equivalent to GP posteriors. This enables us to relate solutions and iterations of a deep-learning algorithm to GP inference. As a result, we can obtain a GP kernel and a nonlinear feature map while training a DNN. Surprisingly, the resulting kernel is the neural tangent kernel. We show kernels obtained on real datasets and demonstrate the use of the GP marginal likelihood to tune hyperparameters of DNNs. Our work aims to facilitate further research on combining DNNs and GPs in practical settings.



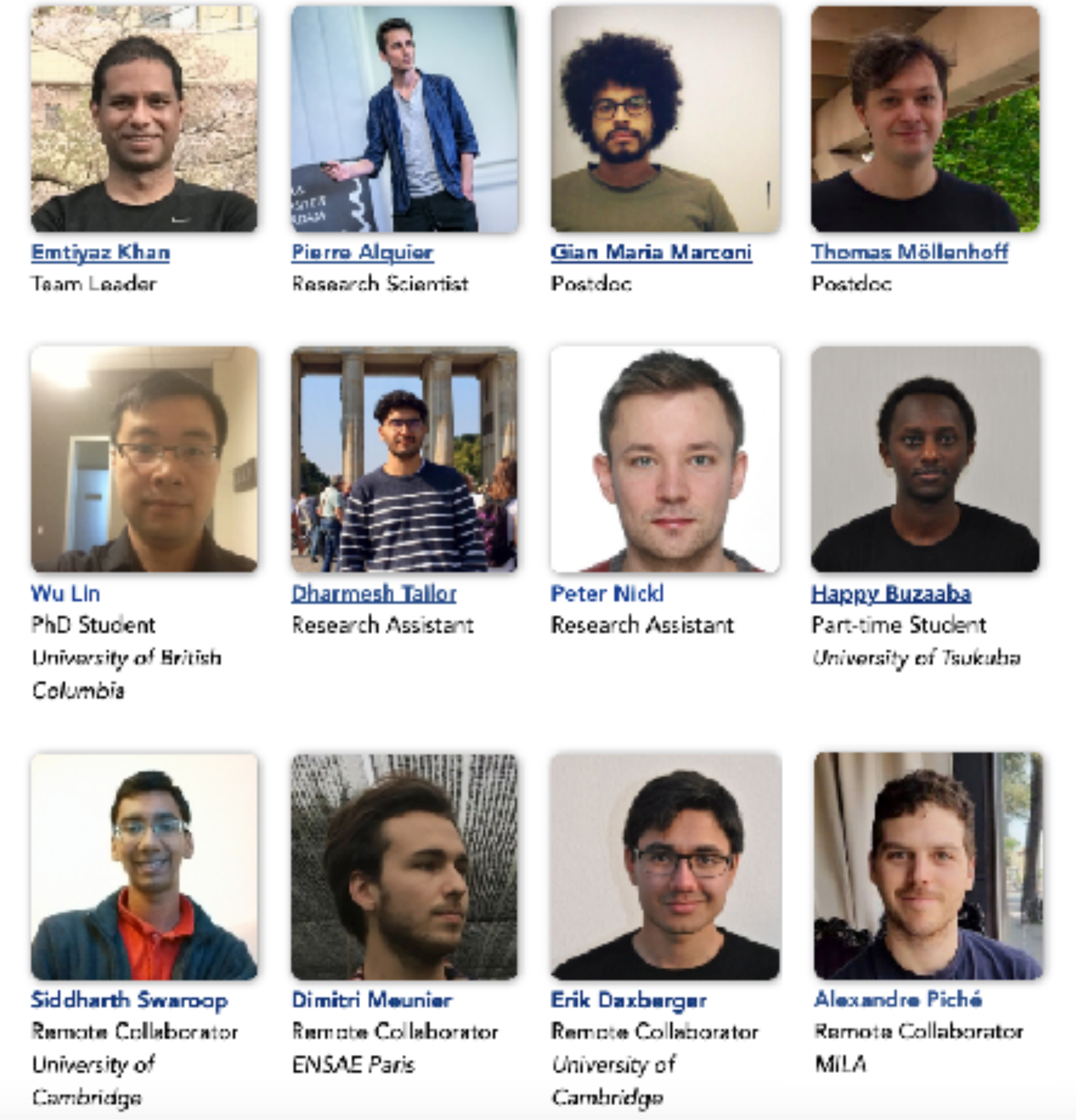
[Submitted on 29 Apr 2020 (v1), last revised 8 Jan 2021 (this version, v4)]

Continual Deep Learning by Functional Regularisation of Memorable Past

Pingbo Pan, Siddharth Swaroop, Alexander Immer, Runa Eschenhagen, Richard E. Turner, Mohammad Emtiyaz Khan

Continually learning new skills is important for intelligent systems, yet standard deep learning methods suffer from catastrophic forgetting of the past. Recent works address this with weight regularisation. Functional regularisation, although computationally expensive, is expected to perform better, but rarely does so in practice. In this paper, we fix this issue by using a new functional-regularisation approach that utilises a few memorable past examples crucial to avoid forgetting. By using a Gaussian Process formulation of deep networks, our approach enables training in weight-space while identifying both the memorable past and a functional prior. Our method achieves state-of-the-art performance on standard benchmarks and opens a new direction for life-long learning where regularisation and memory-based methods are naturally combined.

Approximate Bayesian Inference Team



Emtiyaz Khan
Team Leader

Pierre Alquier
Research Scientist

Gian Maria Marconi
Postdoc

Thomas Möllenhoff
Postdoc

Wu Lin
PhD Student
University of British Columbia

Dharmesh Tailor
Research Assistant

Peter Nickl
Research Assistant

Happy Buzaaba
Part-time Student
University of Tsukuba

Siddharth Swaroop
Remote Collaborator
University of Cambridge

Dimitri Meunier
Remote Collaborator
ENSAE Paris

Erik Daxberger
Remote Collaborator
University of Cambridge

Alexandre Piché
Remote Collaborator
MILA

https://team-approx-bayes.github.io/