

How to Build Machines that Adapt Quickly

Mohammad **Emtiyaz** Khan

RIKEN Center for AI Project, Tokyo

<http://emtiyaz.github.io>



Human Learning at
the age of 6 months.



Converged at the
age of 12 months



Transfer
skills
at the age
of 14
months



Fail because too slow to adapt



Fail because too quick to adapt

TayTweets: Microsoft AI bot manipulated into being extreme racist upon release

Posted Fri 25 Mar 2016 at 4:38am, updated Fri 25 Mar 2016 at 9:17am



TayTweets is programmed to converse like a teenage girl who has "zero chill", according to Microsoft. (Twitter: TayTweets)

Adaptation in Machine Learning

- Even a small change may need retraining
- Huge amount of resources are required only few can afford (costly & unsustainable) [1,2, 3]
- Difficult to apply in “dynamic” settings (robotics, medicine, epidemiology, climate science, etc.)
- Our goal is to solve such challenges
- Also to reduce “magic” in deep 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>

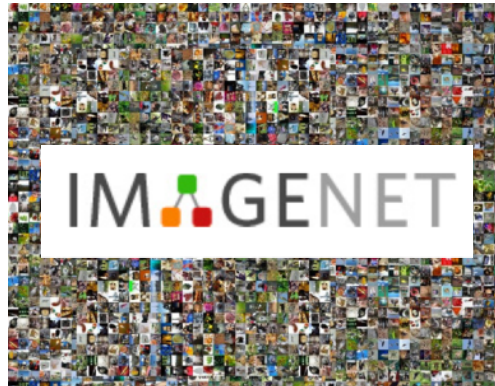
Towards Quick Adaptation

- Better **uncertainty** [1-4]
 - Bayesian Learning rule (BLR)
- Better **regularization** [5-8]
 - Knowledge-Adaptation Priors (K-priors)
- Better **memory** [9]
 - Memory Perturbation Equation (MPE)

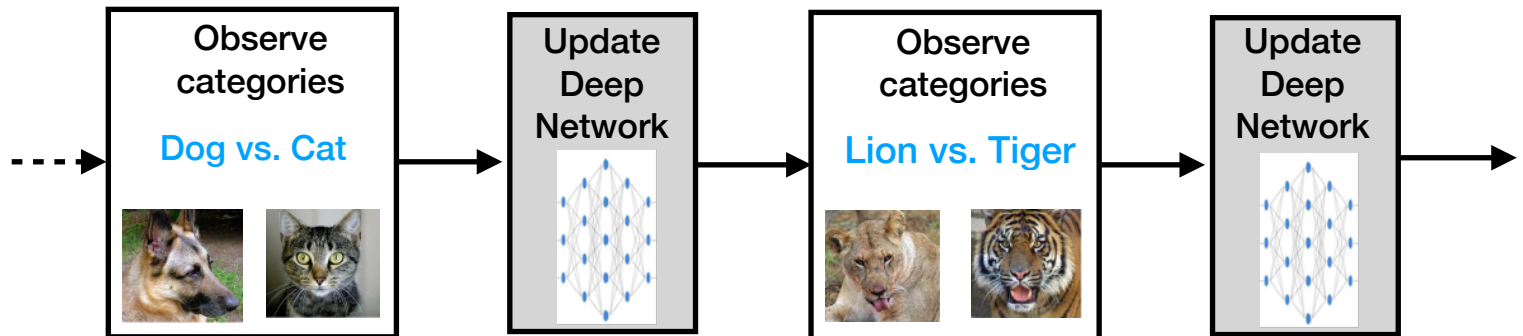
1. Khan and Rue, The Bayesian Learning Rule, JMLR (2023).
2. Khan, et al. Fast and scalable Bayesian deep learning by weight-perturbation in Adam, ICML (2018).
3. Osawa et al. Practical Deep Learning with Bayesian Principles, NeurIPS (2019).
4. Lin et al. Handling the positive-definite constraints in the BLR, ICML (2020).
5. Khan and Swaroop. Knowledge-Adaptation Priors, NeurIPS (2021)
6. Pan et al. Continual deep learning by functional regularisation of memorable past, NeurIPS (2020)
7. Daxberger et al. Improving CL by Accurate Gradient Reconstruction of the Past, TMLR (2023).
8. Daheim et al. Model merging by uncertainty-based gradient matching, arXiv 2023.
9. Nickl, Xu, Tailor, Moellenhoff, Khan, The memory-perturbation equation, NeurIPS (2023)

Example: Continual Learning

Standard
Deep
Learning



Continual Learning: past classes never revisited



Standard training leads to catastrophic forgetting.

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

Bayesian Learning Rule

Better Uncertainty

Weight Regularization

Standard way to is to add a weight-regularizer [1]

$$(\theta - \theta_{\text{old}})^\top F_{\text{old}} (\theta - \theta_{\text{old}})$$

↑ Weight uncertainty

Straightforward improvement in weight-uncertainty is to use variational inference [2-4]

1. Kirkpatrick, James, et al. "Overcoming catastrophic forgetting in neural networks." *PNAS* 2017
2. Khan, et al. "Fast and scalable Bayesian deep learning by weight-perturbation in Adam." *ICML* (2018).
3. Osawa et al. "Practical Deep Learning with Bayesian Principles." *NeurIPS* (2019).
4. Lin et al. "Handling the positive-definite constraints in the BLR." *ICML* (2020).

Practical Deep Learning with Bayes

A reliable estimate of Fisher/Hessian/variance

RMSprop

$$\begin{aligned}g &\leftarrow \hat{\nabla} \ell(\theta) \\h &\leftarrow g \cdot g \\s &\leftarrow (1 - \rho)s + \rho h \\\theta &\leftarrow \theta - \alpha g / \sqrt{s}\end{aligned}$$

Bayesian Learning Rule [3]

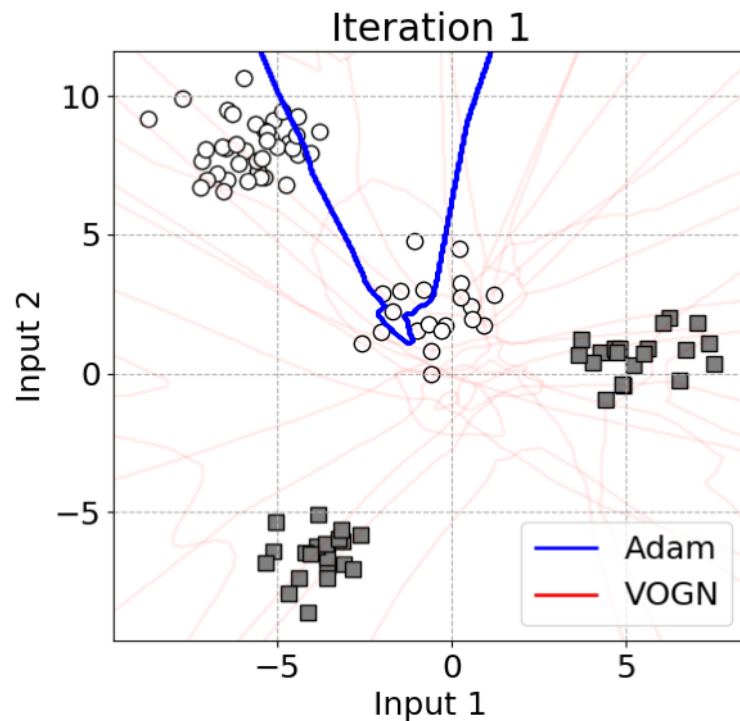
$$\begin{aligned}g &\leftarrow \hat{\nabla} \ell(\theta) \\h &\leftarrow g \cdot \sqrt{s} \cdot \epsilon \\s &\leftarrow (1 - \rho)s + \rho h + \rho^2 h^2 / (2s) \\m &\leftarrow m - \alpha g / s \\\sigma^2 &\leftarrow 1/s, \theta \leftarrow m + \epsilon \sim \mathcal{N}(0, 1/s)\end{aligned}$$

Costs are exactly the same, but the variance quality is much better!!

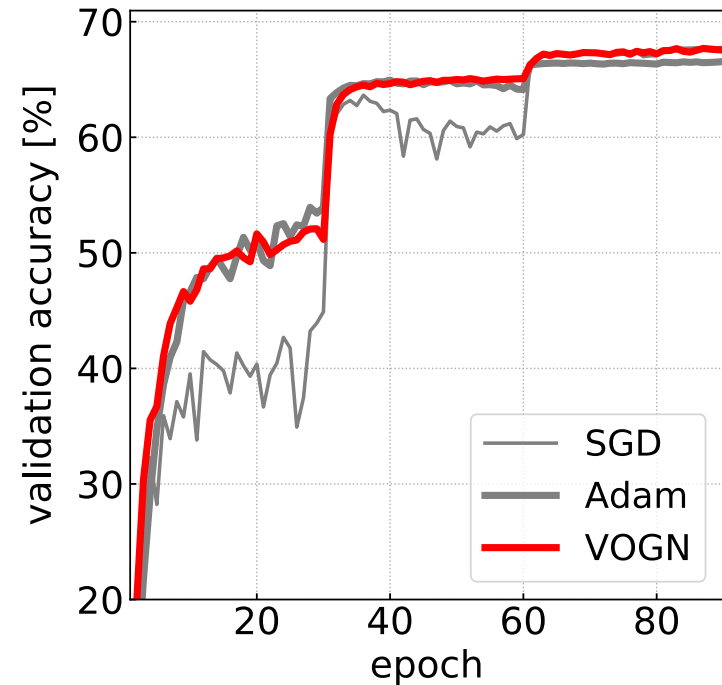
1. Khan, et al. "Fast and scalable Bayesian deep learning by weight-perturbation in Adam." *ICML* (2018).
2. Osawa et al. "Practical Deep Learning with Bayesian Principles." *NeurIPS* (2019).
3. Lin et al. "Handling the positive-definite constraints in the BLR." *ICML* (2020).

Uncertainty of Deep Nets

Variational Online Gauss-Newton (VOGN)



ImageNet Results

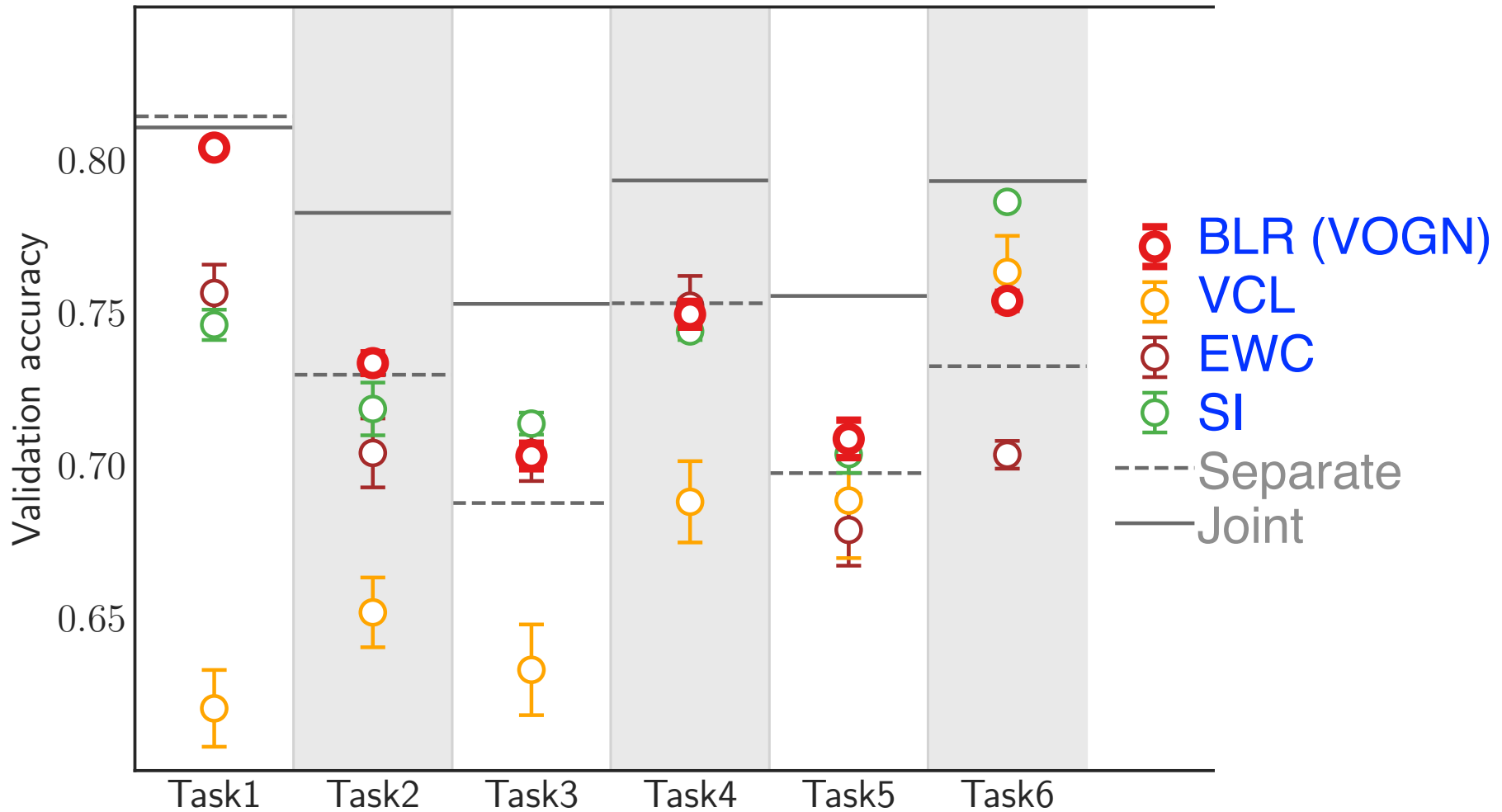


Code available at <https://github.com/team-approx-bayes/dl-with-bayes>

1. Khan, et al. "Fast and scalable Bayesian deep learning by weight-perturbation in Adam." *ICML* (2018).
2. Osawa et al. "Practical Deep Learning with Bayesian Principles." *NeurIPS* (2019).

Continual Learning

CIFAR10



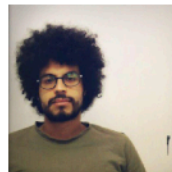
1. Osawa et al. "Practical Deep Learning with Bayesian Principles." NeurIPS (2019).

BLR variant [3] got 1st prize in NeurIPS 2021 Approximate Inference Challenge

Watch **Thomas Moellenhoff's** talk at <https://www.youtube.com/watch?v=LQInIN5EU7E>.

Mixture-of-Gaussian Posteriors with an Improved Bayesian Learning Rule

Thomas Möllenhoff¹, Yuesong Shen², Gian Maria Marconi¹
Peter Nickl¹, Mohammad Emtiyaz Khan¹



¹ Approximate Bayesian Inference Team
RIKEN Center for AI Project, Tokyo, Japan

² Computer Vision Group
Technical University of Munich, Germany

Dec 14th, 2021 — NeurIPS Workshop on Bayesian Deep Learning

1. Khan, et al. "Fast and scalable Bayesian deep learning by weight-perturbation in Adam." *ICML* (2018).
2. Osawa et al. "Practical Deep Learning with Bayesian Principles." *NeurIPS* (2019).
3. Lin et al. "Handling the positive-definite constraints in the BLR." *ICML* (2020).

Bayesian learning rule (BLR) $\lambda \leftarrow (1 - \rho)\lambda - \rho \nabla_{\mu} \mathbb{E}_q[\ell(\theta)]$

See Table 1 in
Khan and Rue, 2021

All sorts of algorithms can be derived by using two sets of approximations.

By relaxing the approximations, we get an improvement, for example, uncertainty aware deep learning optimizers

Learning Algorithm	Posterior Approx.	Natural-Gradient Approx.	Sec.
Optimization Algorithms			
Gradient Descent	Gaussian (fixed cov.)	Delta method	1.3
Newton's method	Gaussian	—"—"	1.3
Multimodal optimization <small>(New)</small>	Mixture of Gaussians	—"—"	3.2
Deep-Learning Algorithms			
Stochastic Gradient Descent	Gaussian (fixed cov.)	Delta method, stochastic approx.	4.1
RMSprop/Adam	Gaussian (diagonal cov.)	Delta method, stochastic approx., Hessian approx., square-root scaling, slow-moving scale vectors	4.2
Dropout	Mixture of Gaussians	Delta method, stochastic approx., responsibility approx.	4.3
STE	Bernoulli	Delta method, stochastic approx.	4.5
Online Gauss-Newton (OGN) <small>(New)</small>	Gaussian (diagonal cov.)	Gauss-Newton Hessian approx. in Adam & no square-root scaling	4.4
Variational OGN <small>(New)</small>	—"—"	Remove delta method from OGN	4.4
BayesBiNN <small>(New)</small>	Bernoulli	Remove delta method from STE	4.5
Approximate Bayesian Inference Algorithms			
Conjugate Bayes	Exp-family	Set learning rate $\rho_t = 1$	5.1
Laplace's method	Gaussian	Delta method	4.4
Expectation-Maximization	Exp-Family + Gaussian	Delta method for the parameters	5.2
Stochastic VI (SVI)	Exp-family (mean-field)	Stochastic approx., local $\rho_t = 1$	5.3
VMP	—"—"	$\rho_t = 1$ for all nodes	5.3
Non-Conjugate VMP	—"—"	—"—"	5.3
Non-Conjugate VI <small>(New)</small>	Mixture of Exp-family	None	5.4

1. Khan and Rue, The Bayesian Learning Rule, arXiv, <https://arxiv.org/abs/2107.04562>, 2021
2. Khan and Lin. "Conjugate-computation variational inference..." Alstats (2017).

Bayesian-SAM

An Adam-style algorithm, derived using the BLR, where variances are automatically learned.

SAM with RMSprop

$$\begin{aligned}g_1 &\leftarrow \hat{\nabla} \ell(\theta) \\ \epsilon &\leftarrow \rho \frac{g_1}{\|g_1\|} \\ g &\leftarrow \hat{\nabla} \ell(\theta + \epsilon) \\ s &\leftarrow (1 - \rho)s + \rho g^2 \\ \theta &\leftarrow \theta - \alpha(\sqrt{s} + \delta)^{-1} g\end{aligned}$$

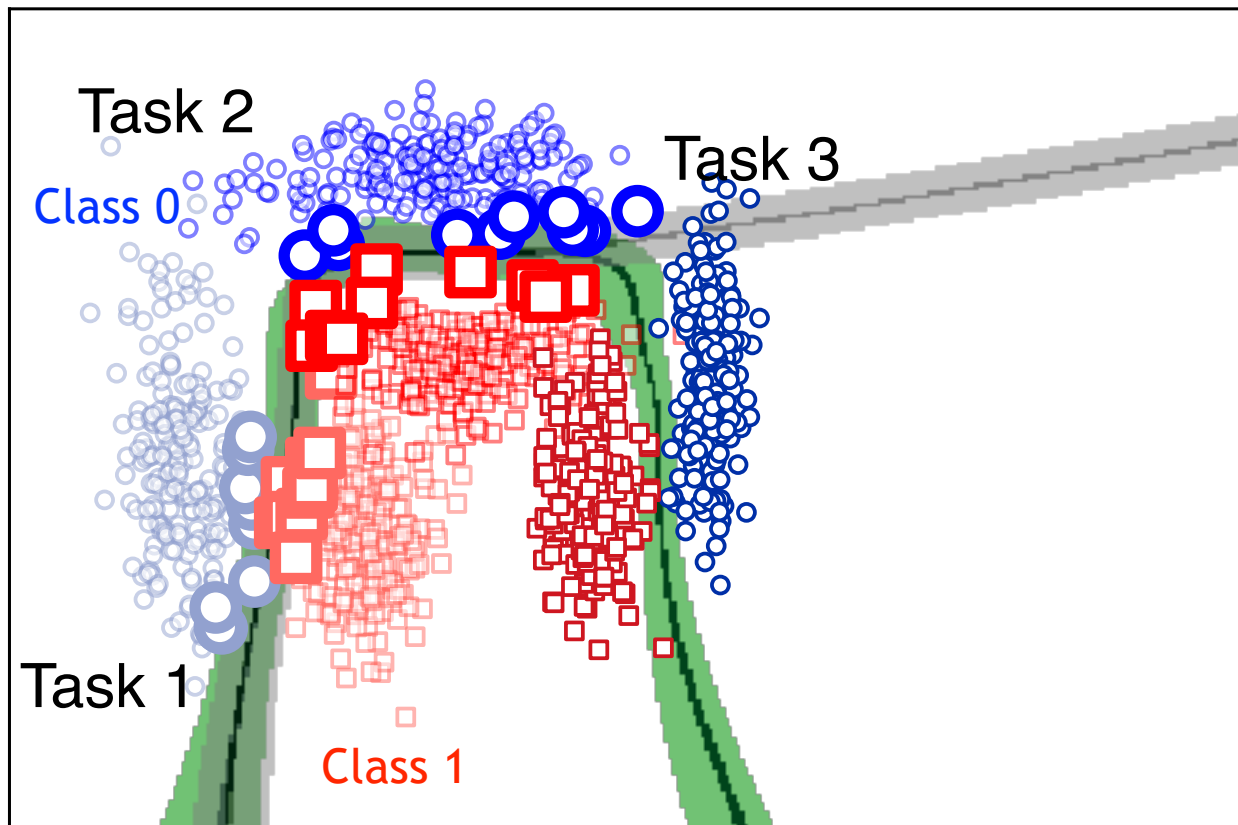
SAM with BLR

$$\begin{aligned}g_1 &\leftarrow \hat{\nabla} \ell(\theta) \\ \epsilon &\leftarrow \frac{\rho'}{s} g_1 \\ g &\leftarrow \hat{\nabla} \ell(\theta + \epsilon) \\ s &\leftarrow (1 - \rho)s + \rho \sqrt{s} |g_1| \\ \theta &\leftarrow \theta - \alpha(s + \gamma)^{-1} g \\ \sigma^2 &\leftarrow (s + \gamma)^{-1}, \quad \theta \leftarrow m + \epsilon' \sigma\end{aligned}$$

Knowledge-Adaptation Prior

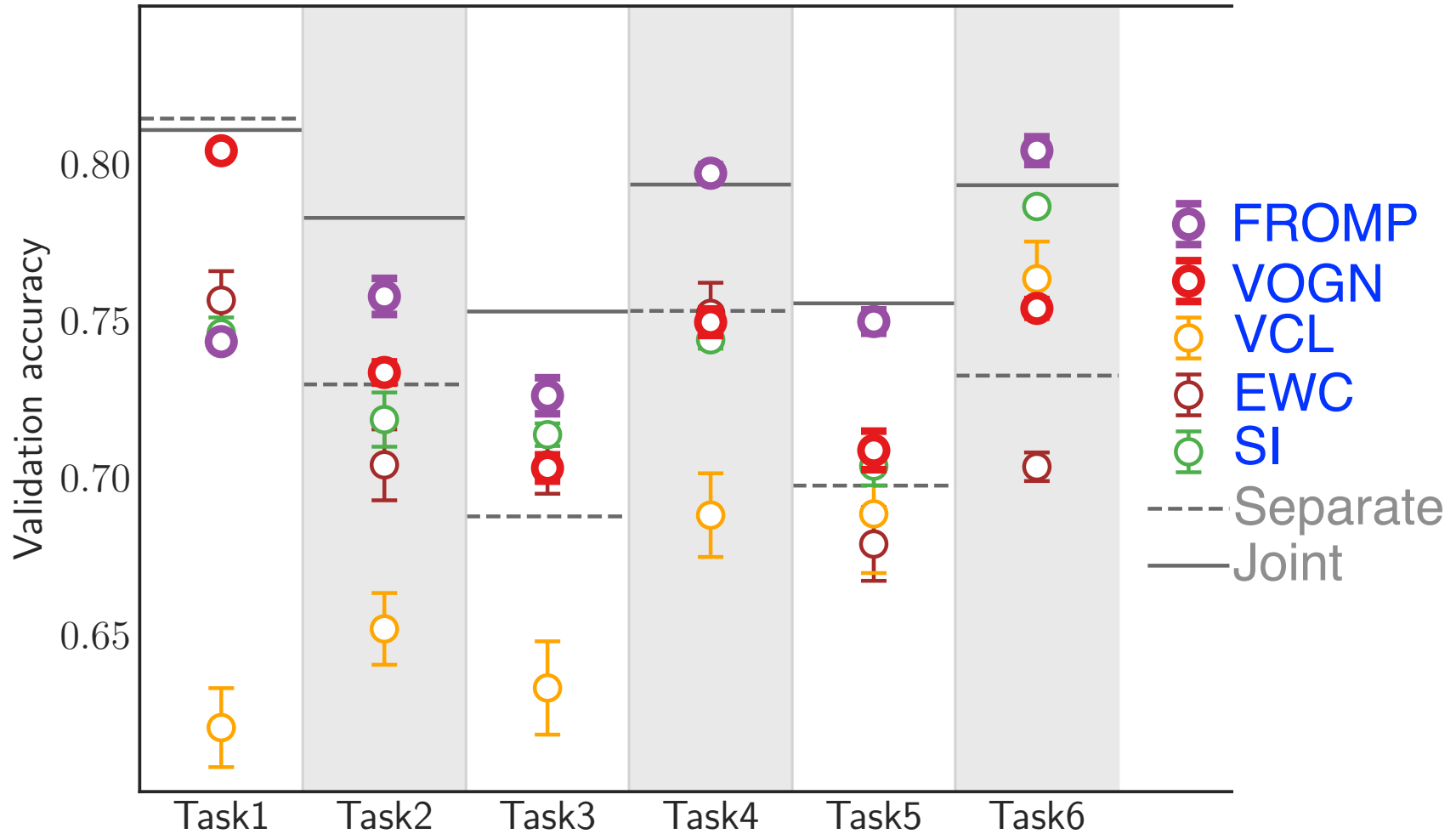
Better Regularization

Function Regularization of Memorable Examples [2]



1. Khan et al. Approximate Inference Turns Deep Networks into Gaussian Process, NeurIPS, 2019
2. Pan et al. Continual Deep Learning by Functional Regularisation of Memorable Past, NeurIPS, 2020

Improvements over EWC and VOGN



Functional Regularization of Memorable Past (FROMP)

Weight-regularizer (EWC) [1]

$$(\theta - \theta_{\text{old}})^\top \underset{\substack{\uparrow \\ \text{Weight uncertainty}}}{F_{\text{old}}} (\theta - \theta_{\text{old}})$$

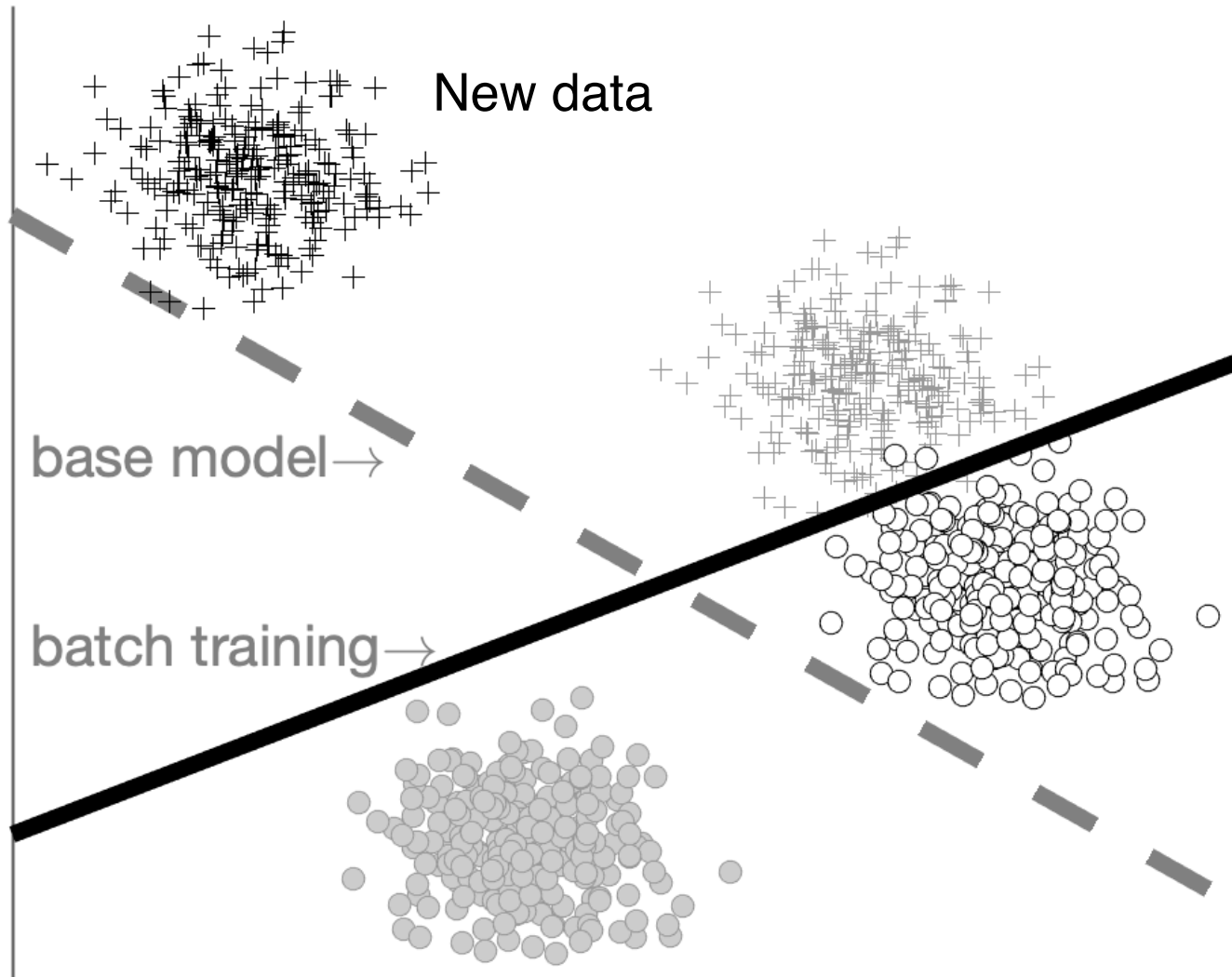
Functional regularizer (FROMP) [2]

$$[\sigma(\mathbf{f}(\theta)) - \sigma(\mathbf{f}_{\text{old}})]^\top \underset{\substack{\uparrow \\ \text{Uncertainty}}}{K_{\text{old}}^{-1}} [\sigma(\mathbf{f}(\theta)) - \underset{\substack{\uparrow \\ \text{Predictions}}}{\sigma(\mathbf{f}_{\text{old}})}]$$

Why does this work? It is a way to replay past gradients, which leads to the idea of **K-priors**.

1. Kirkpatrick, James, et al. "Overcoming catastrophic forgetting in neural networks." *PNAS* 2017
2. Pan et al. Continual Deep Learning by Functional Regularisation of Memorable Past, NeurIPS, 2020

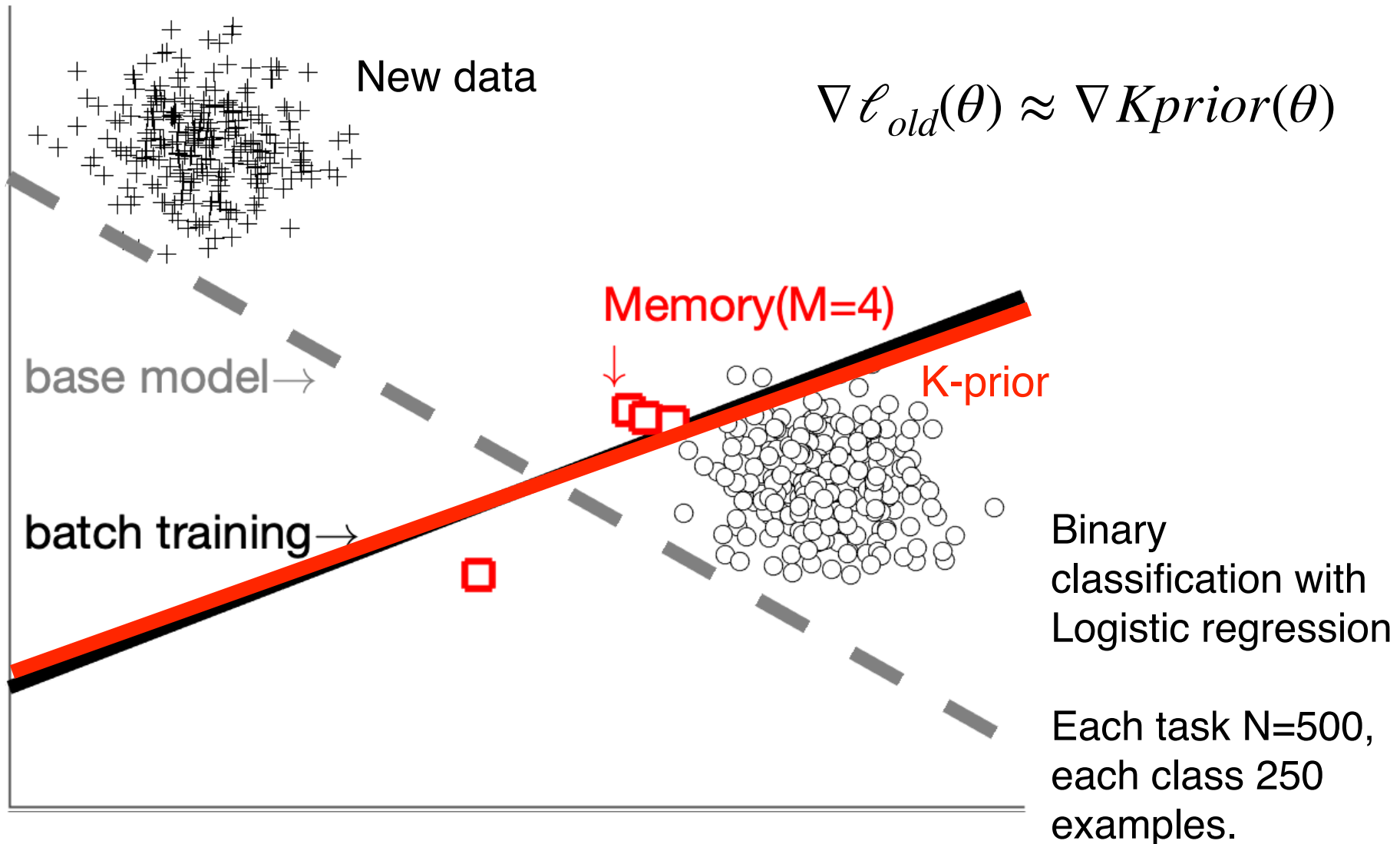
Intuition behind K-priors



Binary
classification with
Logistic regression

Each task $N=500$,
each class 250
examples.

Intuition behind K-priors



Easy to see in Linear Regression

Weight-space Function-space

$$\arg \min_{\theta} \ell_{old} = \|\theta\|^2 + \|y - X\theta\|^2 \qquad F_{old} = I + X^T X$$

$$(\theta - \theta_{old})^T F_{old} (\theta - \theta_{old}) = (\theta - \theta_{old})^T (I + X^T X) (\theta - \theta_{old})$$

Entirely in weight-space (EWC) [1]

$$= \|\theta - \theta_{old}\|^2 + \|X\theta - X\theta_{old}\|^2$$

Weight-space Function-space

Knowledge-adaptation prior [3]

$$= (X\theta - X\theta_{old})^T K^{-1} (X\theta - X\theta_{old})$$

Entirely in function-space (FROMP) [2]

In linear regression, they are equivalent and are all ways to reconstruct the old problem (or its gradients)

1. Kirkpatrick, James, et al. "Overcoming catastrophic forgetting in neural networks." *PNAS* 2017
2. Pan et al. Continual Deep Learning by Functional Regularisation of Memorable Past, NeurIPS, 2020
3. Khan and Swaroop. Knowledge-Adaptation Priors, NeurIPS, 2021

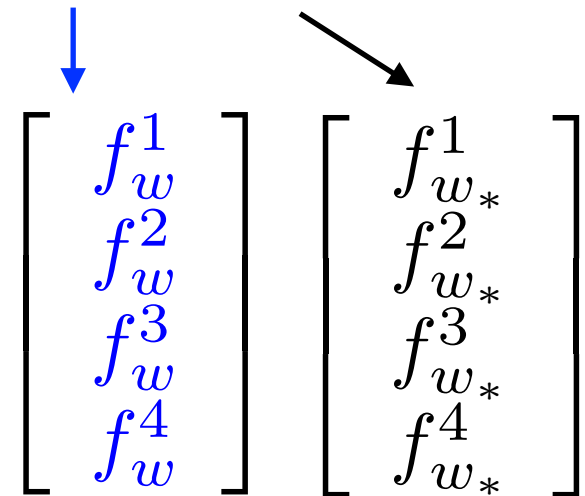
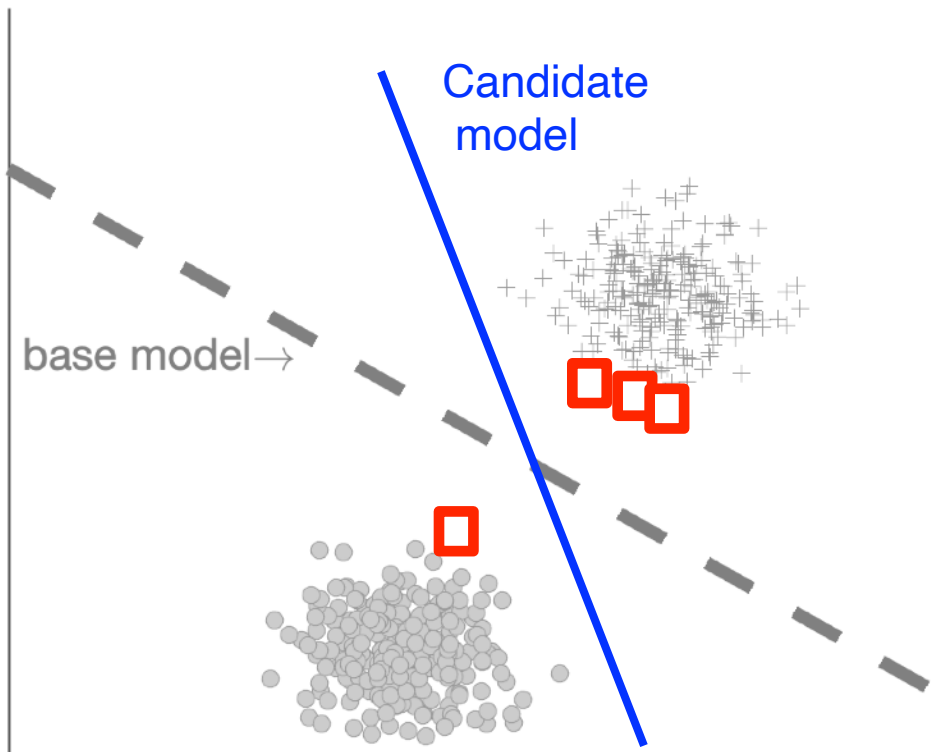
Knowledge-Adaptation Priors

Combine weight and function-space divergences

Weight-space

Function-space

$$\mathcal{K}(\theta) = \tau \mathbb{D}_w(\theta \parallel \theta_{\text{old}}) + \mathbb{D}_f(\mathbf{f}(\theta) \parallel \mathbf{f}(\theta_{\text{old}}))$$

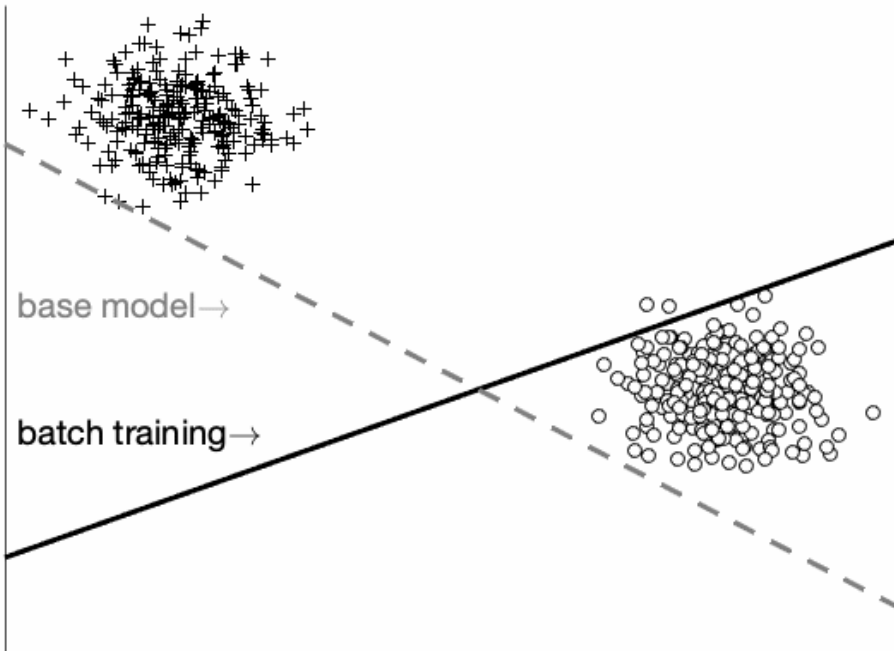


K-prior is a way to replay past gradients

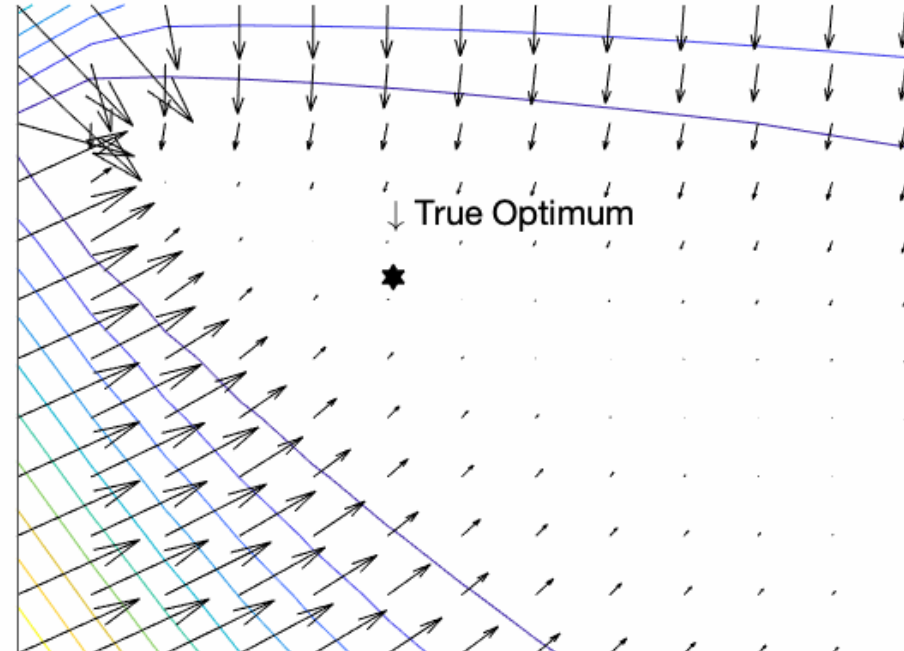
A General Principle of Adaptation

Reconstruct past gradients

M=0

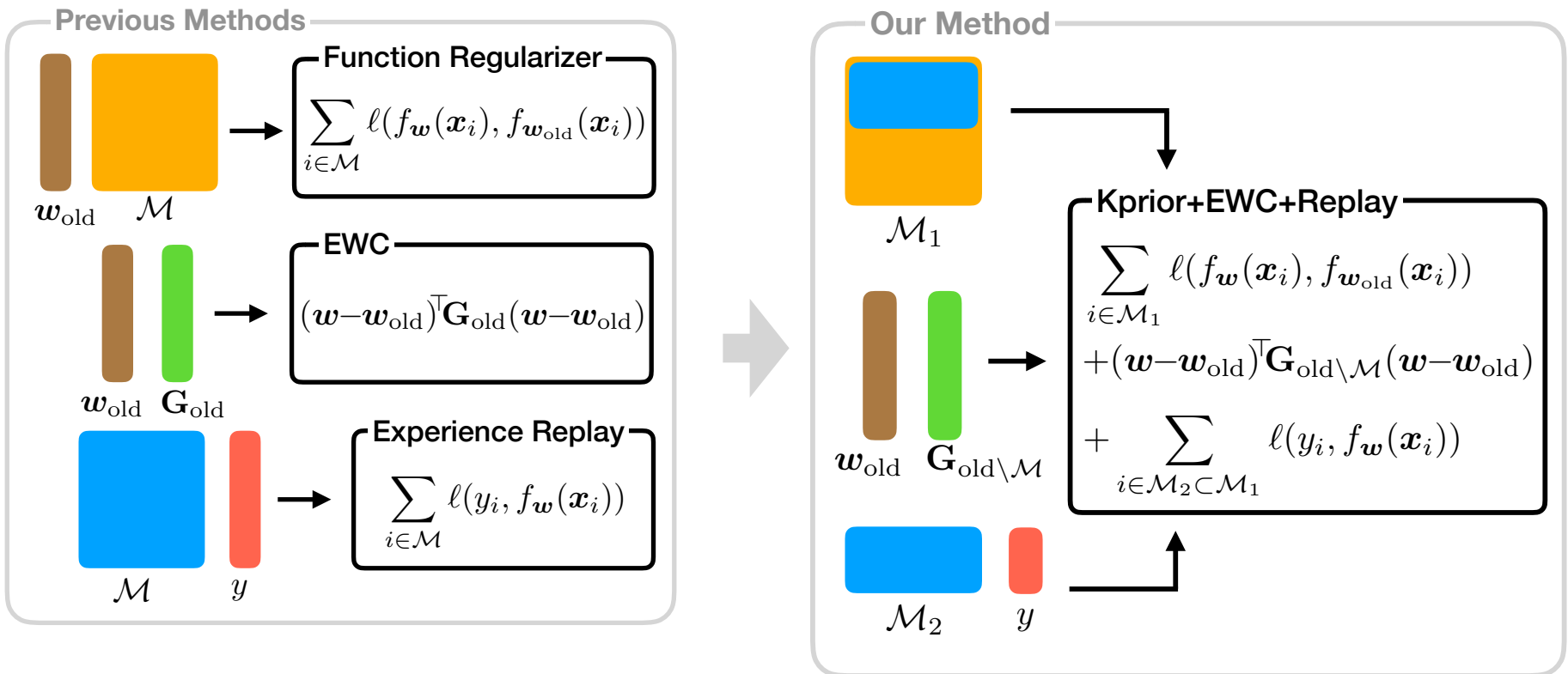


True grads (black) vs K-prior (red)

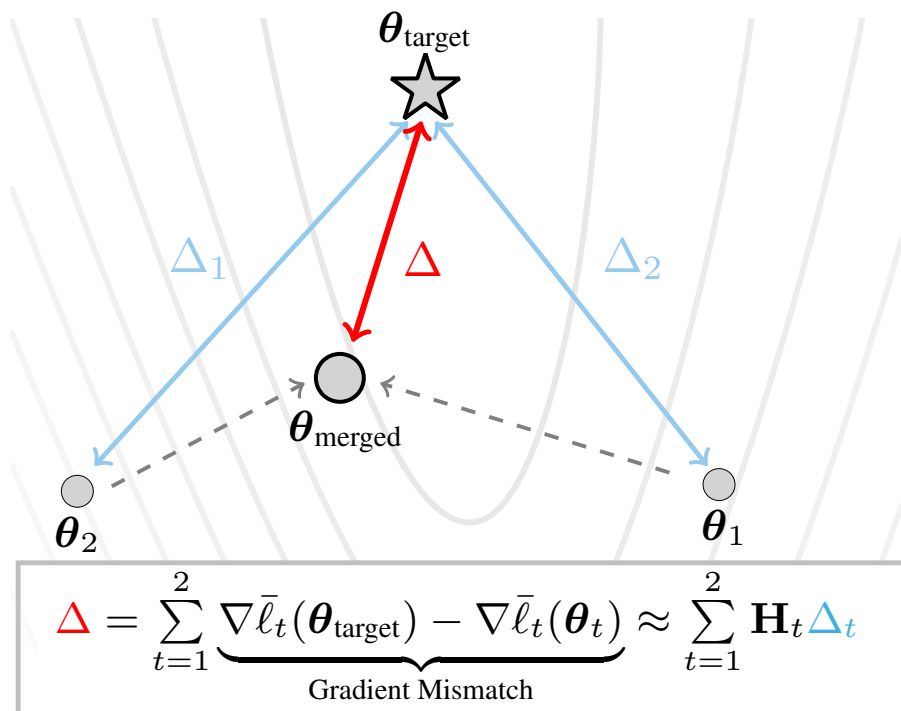


How to combine EWC + FR + Replay

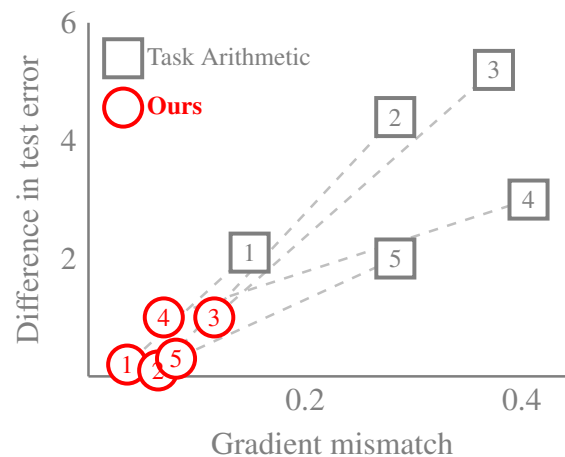
Combine approaches to (successively) reduce grad-reconstruction error



Model Merging for LLMs



RoBERTa on IMDB



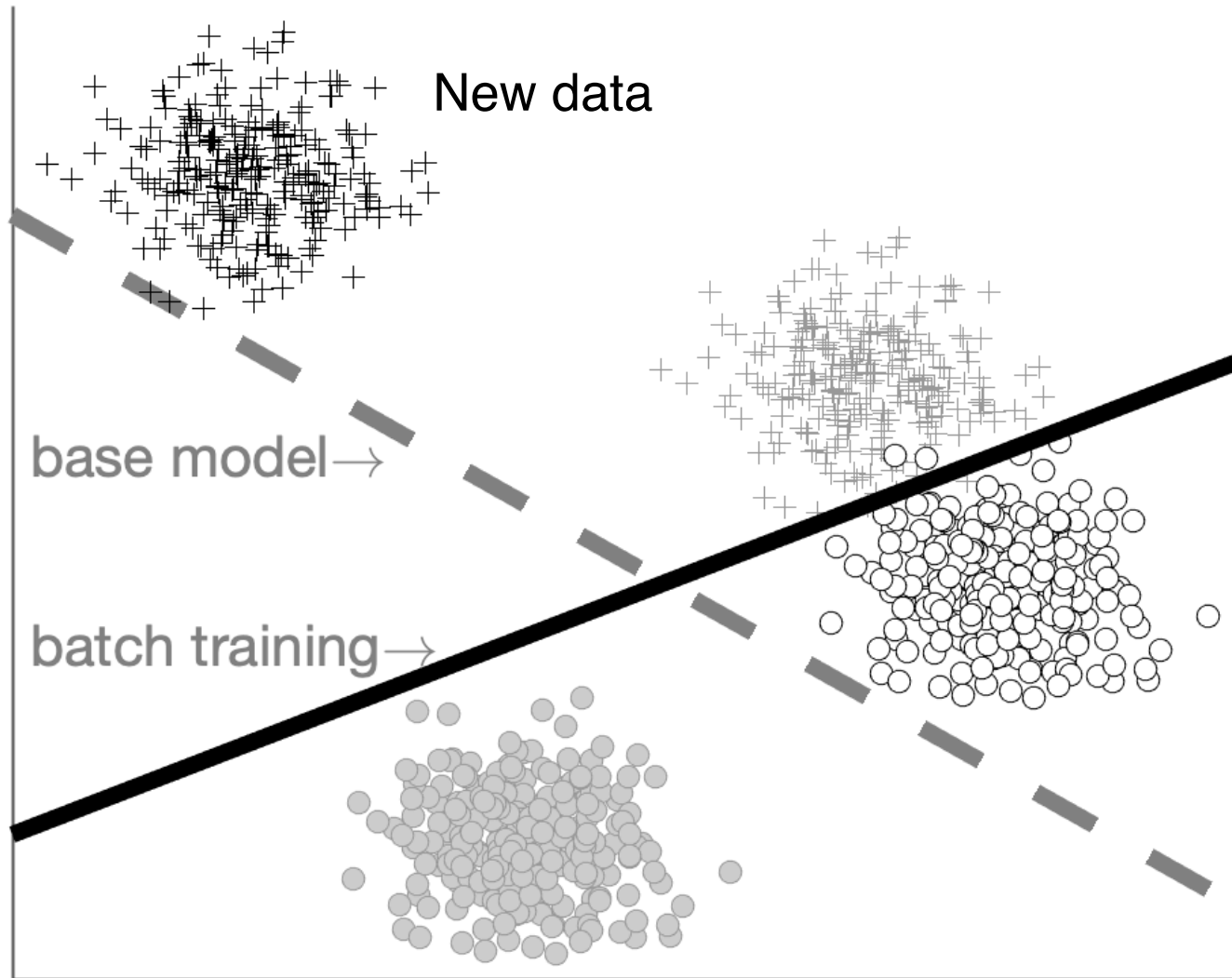
Toxicity removal from GPT (1.3B)

Model	θ	Toxicity		Fluency
		100·Avg.	Num. Toxic	PPL(↓)
GPT2 _{117M}	θ_{LLM}	11.2	15.4 %	24.9
	TA	9.8	13.1 %	30.3
	ours	9.6 (↓0.2)	12.8 % (↓0.3)	26.9 (↓3.4)
GPT-J _{1.3B}	θ_{LLM}	11.9	16.6 %	12.6
	TA	10.7	14.5 %	12.7
	ours	10.2 (↓0.5)	14.0 % (↓0.5)	12.8 (↓0.1)

Memory-Perturbation Equation

Better Memory

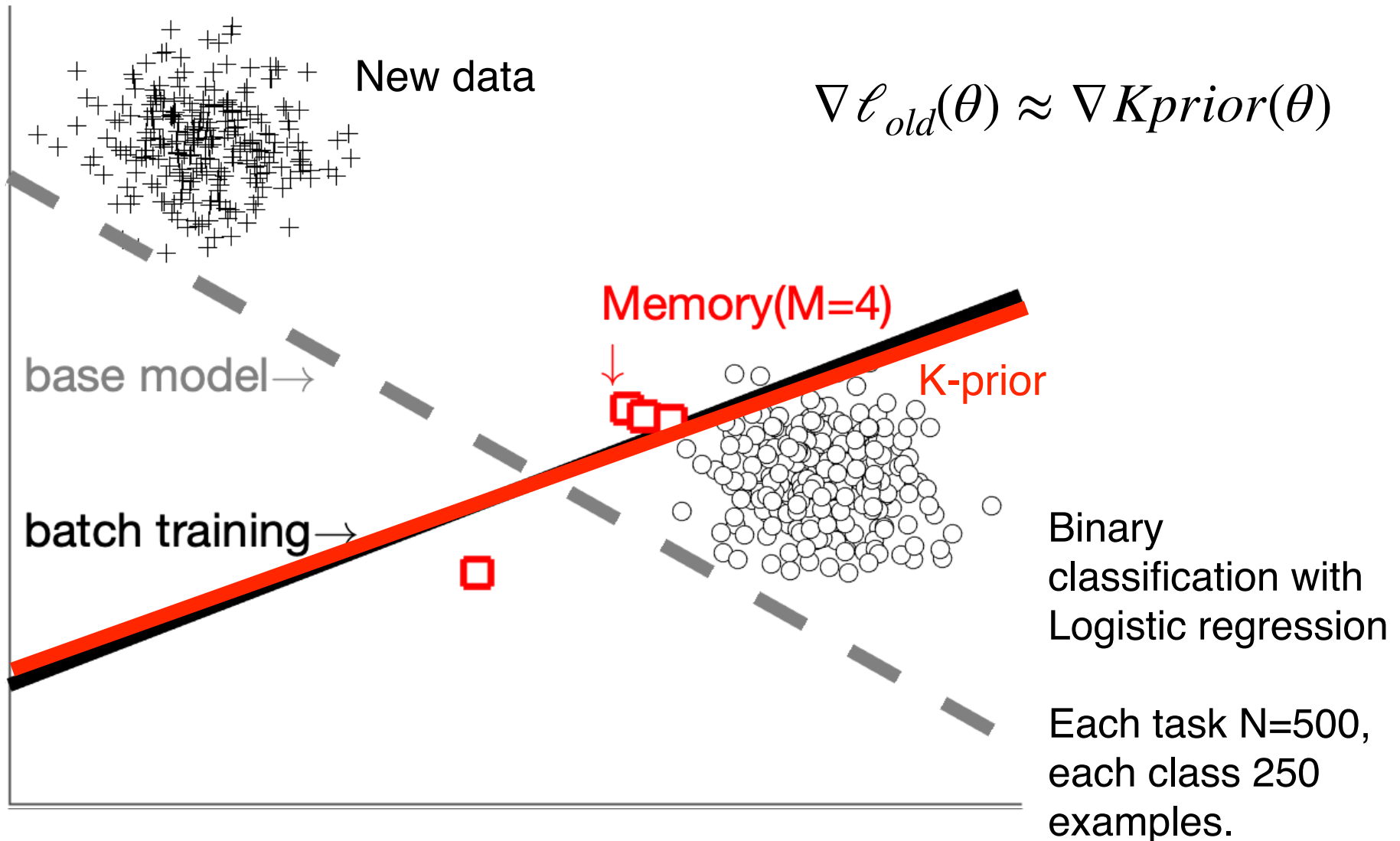
Intuition behind K-priors



Binary
classification with
Logistic regression

Each task $N=500$,
each class 250
examples.

Intuition behind K-priors



How to Choose Memory?

Minimize the error in the gradients

$$\begin{aligned} & \nabla l_{\text{old}}(\theta) - \nabla K(\theta) \\ &= \sum_{i \in \mathcal{D} \setminus \mathcal{M}} \nabla f_i(\theta) [\sigma(f_i(\theta)) - \sigma(f_i(\theta_{\text{old}}))] \end{aligned}$$

\uparrow Examples not in the memory

\uparrow Feature disagreement
-> prediction variance

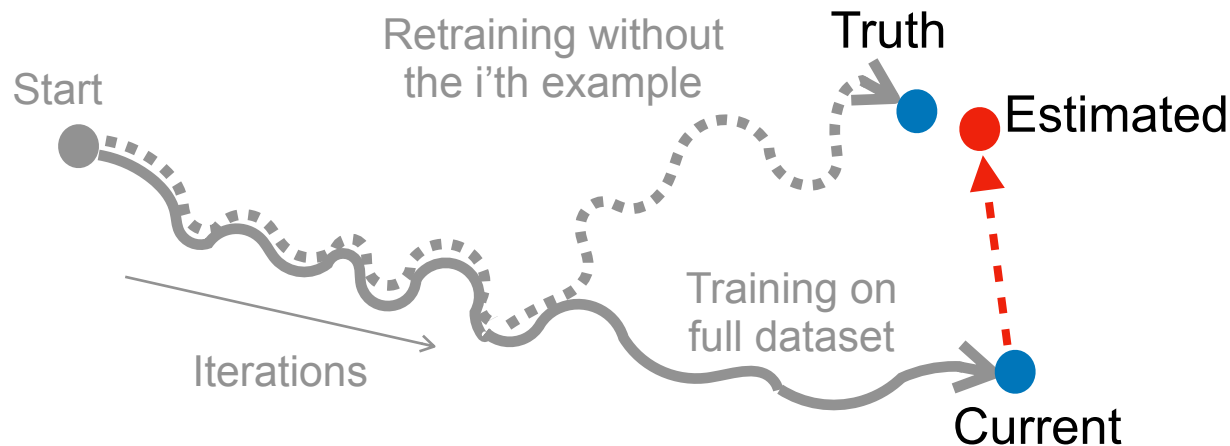
\uparrow Prediction disagreement
-> prediction error

Past and future should agree at some data examples (but, a-priori, we don't know which ones).

Use sensitivity to guess!

Memory and Sensitivity

Past information with most influence on the present

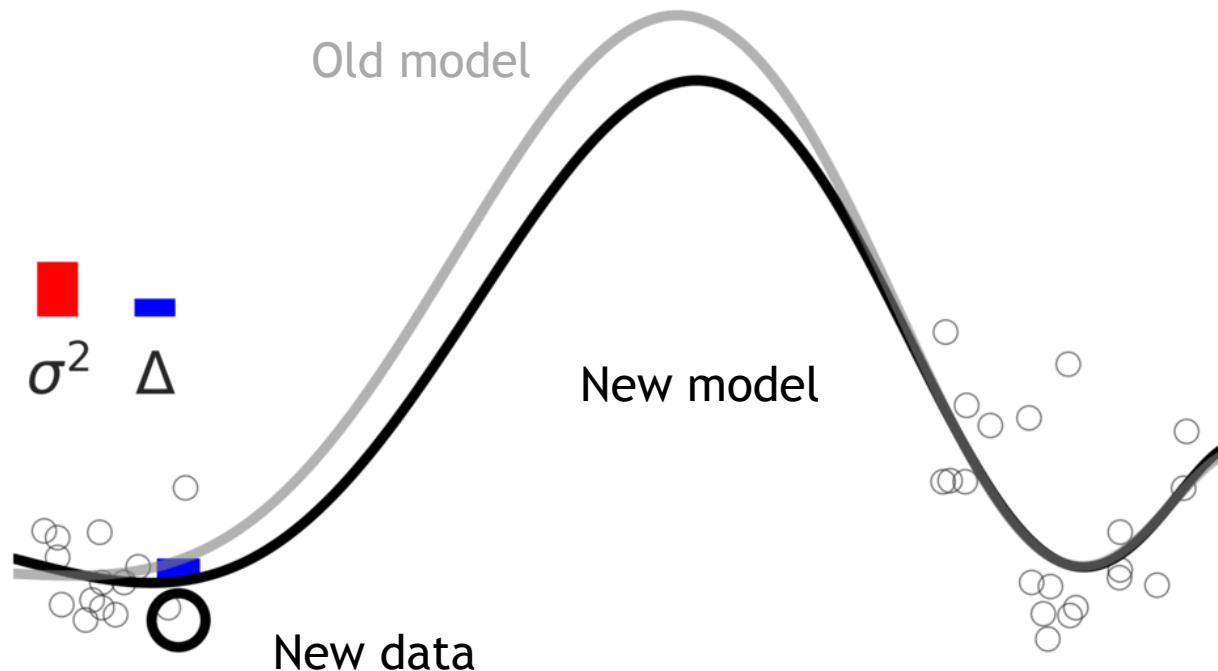


Computing the algorithm-deviation by retraining is expensive. We want to **estimate it without retraining!**

Memory Perturbation

How sensitive is a model to its training data?

Deviation (Δ) = predictionError * predictionVariance

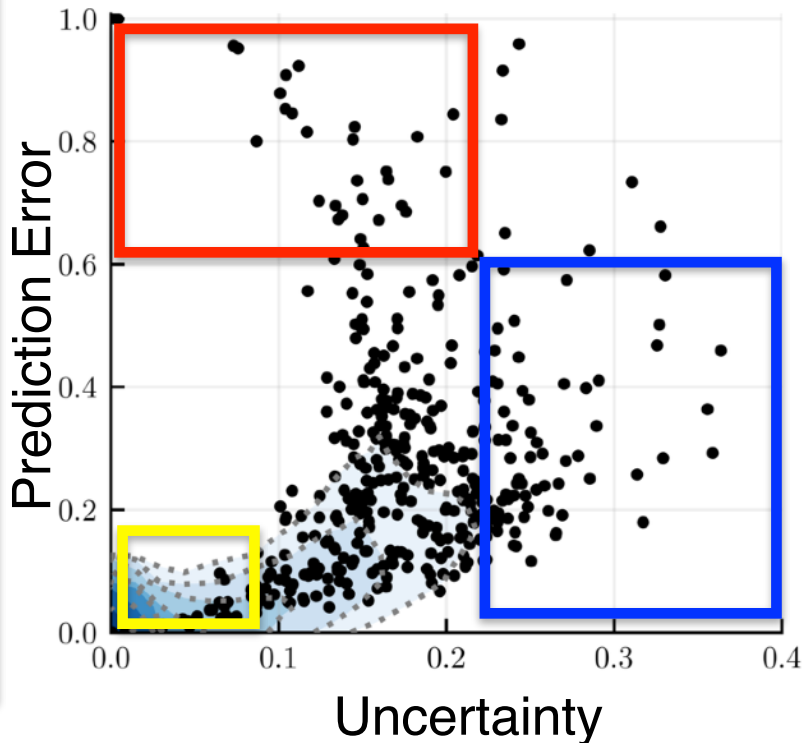
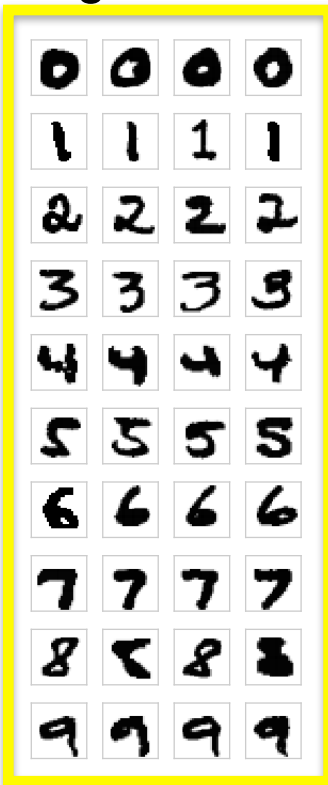


1. Cook. Detection of Influential Observations in Linear Regression. Technometrics. ASA 1977
2. Nickl, Xu, Tailor, Moellenhoff, Khan, The memory-perturbation equation, NeurIPS, 2023

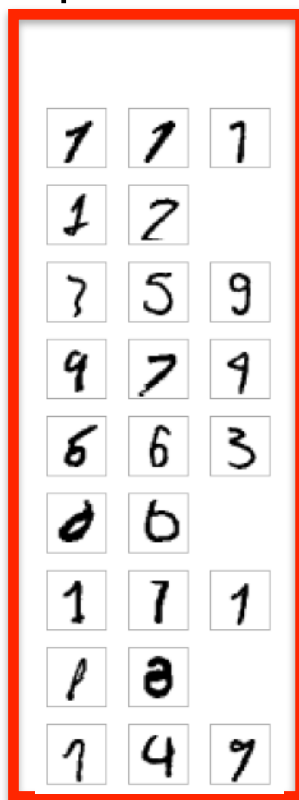
Memory Maps using the BLR

Understand generic ML models and algorithms.

Regular examples



Unpredictable

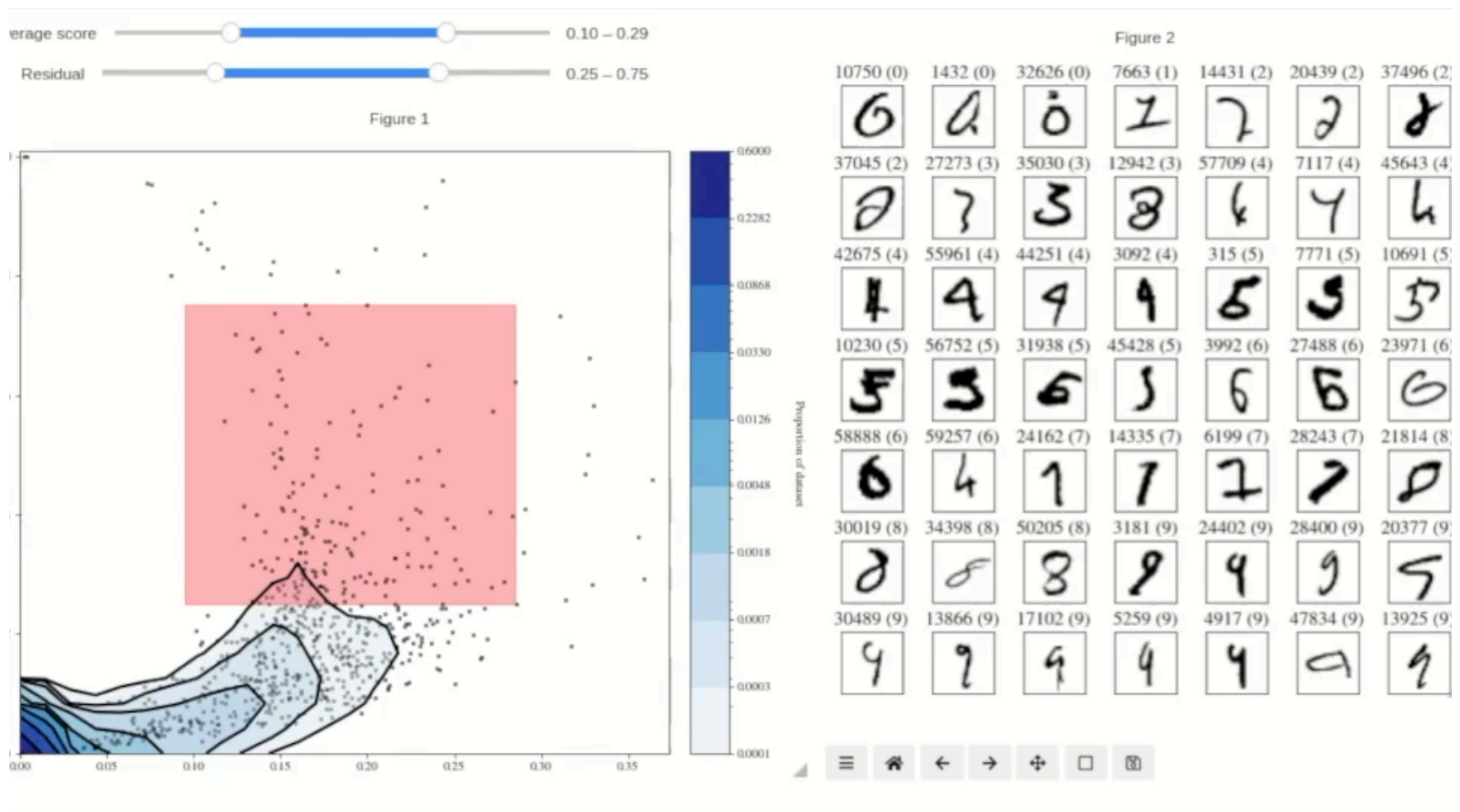


Uncertain



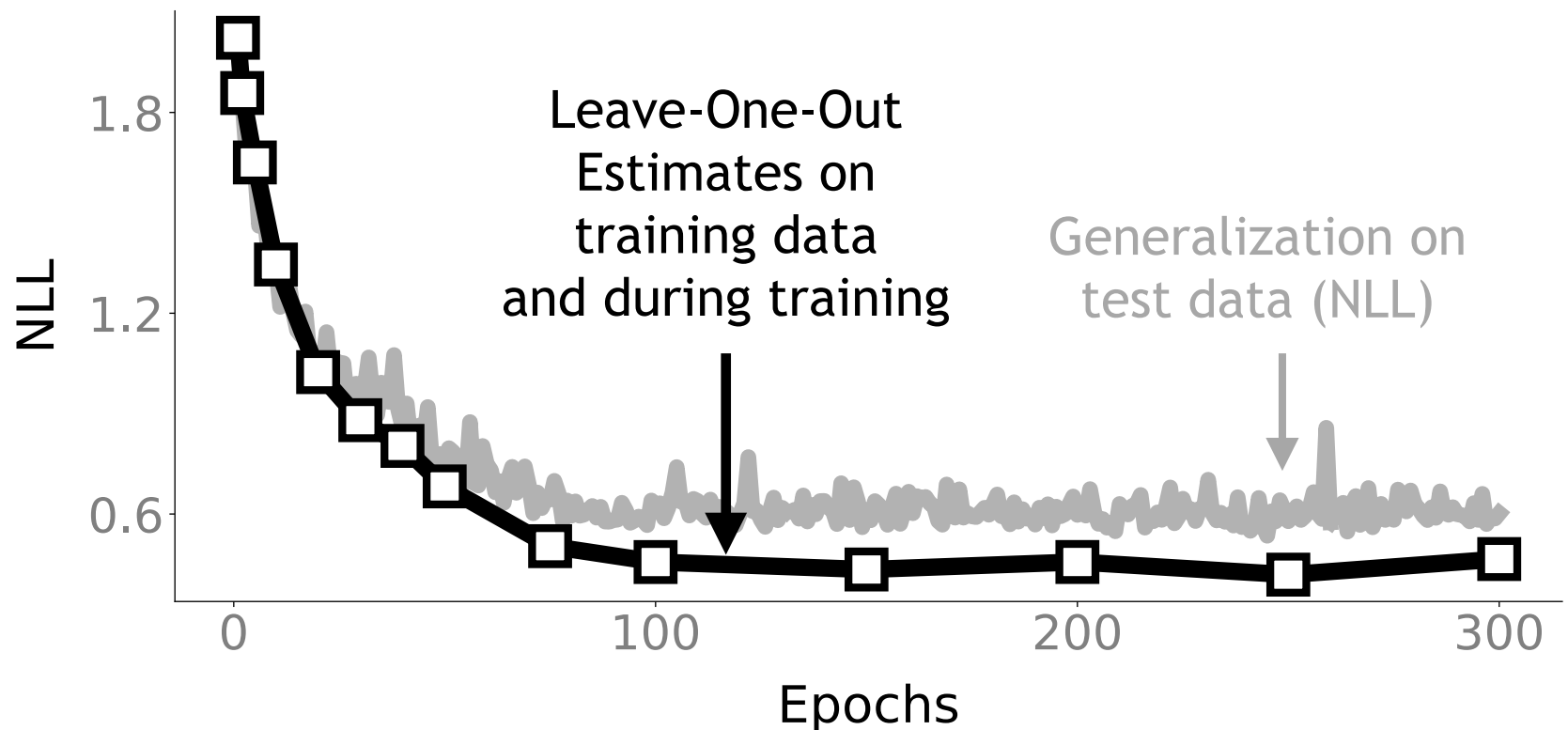
A Tool for Data-Scientists

Understand the memory of a model.



Predict Generalization during Training

CIFAR10 on ResNet-20 using iBLR [1]. SGD also works but better uncertainty gives better estimates.



Towards Quick Adaptation

- Better **uncertainty** [1-4]
 - Bayesian Learning rule (BLR)
- Better **regularization** [5-8]
 - Knowledge-Adaptation Priors (K-priors)
- Better **memory** [9]
 - Memory Perturbation Equation (MPE)

1. Khan and Rue, The Bayesian Learning Rule, JMLR (2023).
2. Khan, et al. Fast and scalable Bayesian deep learning by weight-perturbation in Adam, ICML (2018).
3. Osawa et al. Practical Deep Learning with Bayesian Principles, NeurIPS (2019).
4. Lin et al. Handling the positive-definite constraints in the BLR, ICML (2020).
5. Khan and Swaroop. Knowledge-Adaptation Priors, NeurIPS (2021)
6. Pan et al. Continual deep learning by functional regularisation of memorable past, NeurIPS (2020)
7. Daxberger et al. Improving CL by Accurate Gradient Reconstruction of the Past, TMLR (2023).
8. Daheim et al. Model merging by uncertainty-based gradient matching, arXiv 2023.
9. Nickl, Xu, Tailor, Moellenhoff, Khan, The memory-perturbation equation, NeurIPS (2023)

The Bayes-Duality Project

Toward AI that learns adaptively, robustly, and continuously, like humans



Emtiyaz Khan

Research director
(Japan side)

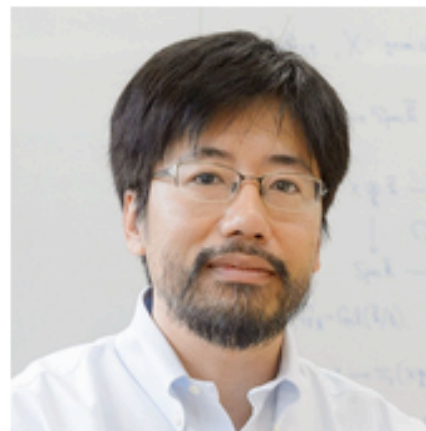
Approx-Bayes team at
RIKEN-AIP and OIST



Julyan Arbel

Research director
(France side)

Statify-team, Inria
Grenoble Rhône-Alpes



Kenichi Bannai

Co-PI (Japan side)

Math-Science Team at
RIKEN-AIP and Keio
University



Rio Yokota

Co-PI
(Japan side)

Tokyo Institute of
Technology

Received total funding of around **USD 3 million** through JST's CREST-ANR and Kakenhi Grants.

Approximate Bayesian Inference Team

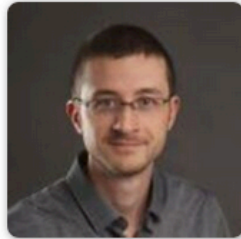
<https://team-approx-bayes.github.io/>



Emtiyaz Khan
Team Leader



Thomas Möllenhoff
Research Scientist

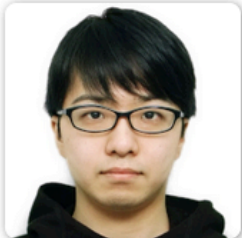


Geoffrey Wolfer
Special Postdoctoral
Resesarcher



**Hugo Monzón
Maldonado**
Postdoctoral
Researcher

Many thanks to our group members and collaborators (many not on this slide).



Keigo Nishida
Postdoctoral
Researcher
RIKEN BDR



Gian Maria Marconi
Postdoctoral
Researcher



Lu Xu
Postdoctoral
Researcher



Peter Nickl
Research Assistant

We have open positions and are always looking for new collaborations.



Etash Guha
Intern
Georgia Tech



Joseph Austerweil
Visiting Scientist
*University of
Winsconsin-Madison*



Pierre Alquier
Visiting Scientist
*ESSEC Business
School*



Dharmesh Tailor
Remote Collaborator
*University of
Amsterdam*