Presentation at the Bayes on the Beach conference, Feb 7, 2024

The Bayesian Learning Rule

Mohammad Emtiyaz Khan

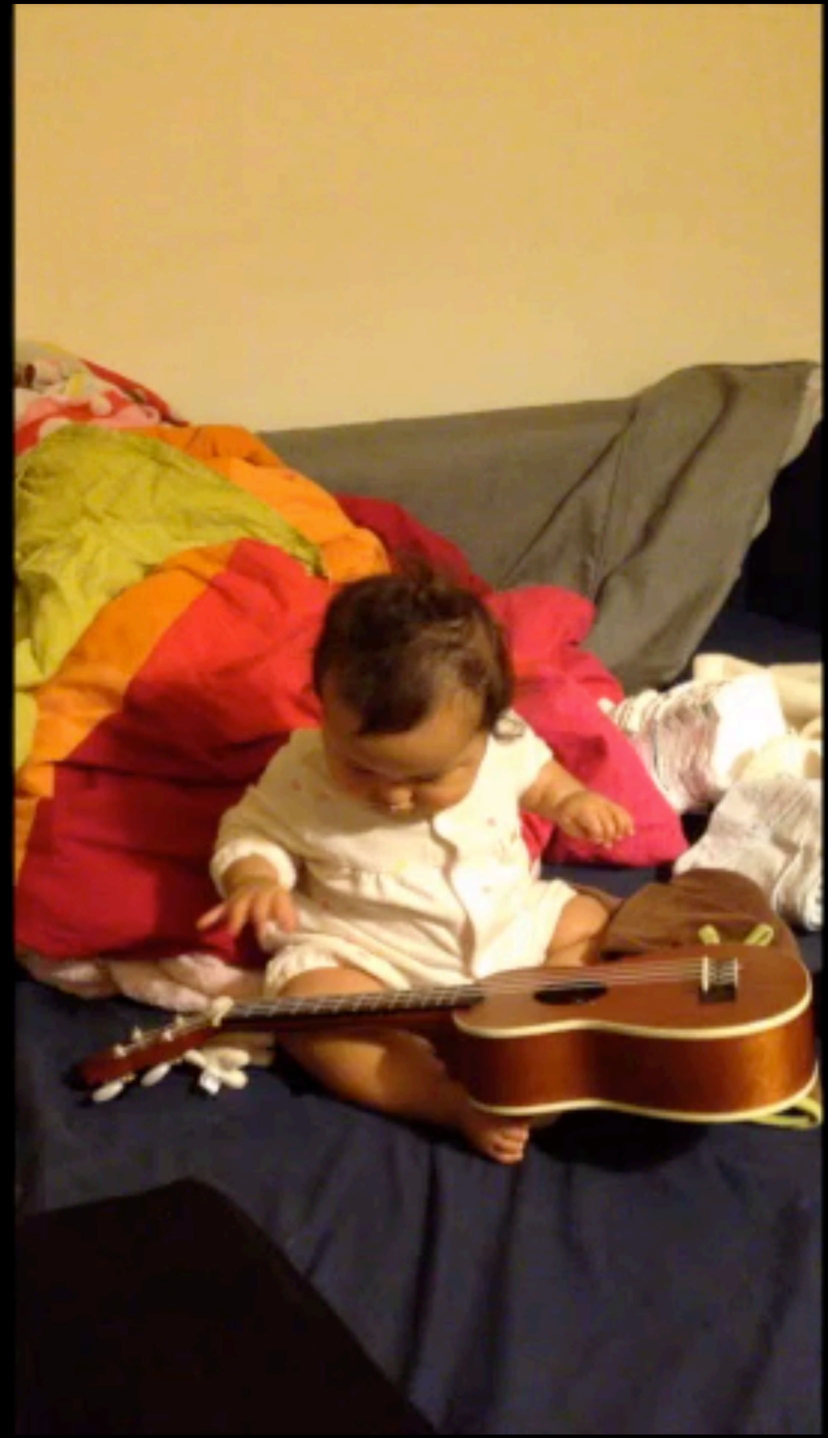
RIKEN Center for AI Project, Tokyo

http://emtiyaz.github.io

Summary of recent research at https://emtiyaz.github.io/papers/symposium_2022.pdf

Slides available at https://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 or quick to adapt

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

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
  - Help in building safe and trustworthy AI
  - To reduce "magic" in deep learning (DL)

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

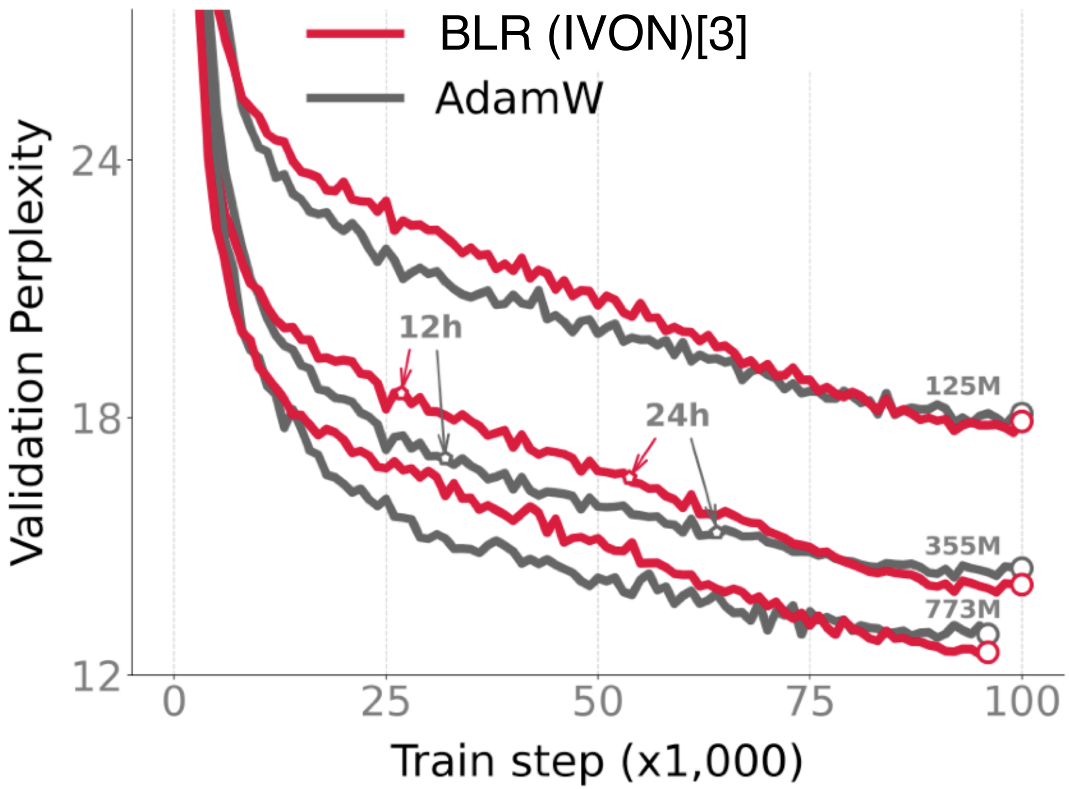
Bayesian Learning Rule [1]

- Bridge DL & Bayesian learning [2-5]
  - SOTA on GPT-2 and ImageNet [5]
- Improve other aspects of DL [5-7]
  - Calibration, uncertainty, memory etc.
  - Understand and fix model behavior
- Towards human-like quick adaptation

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. Shen et al. Variational Learning is Effective for Large Deep Networks, Under review.
6. Daheim et al. Model merging by uncertainty-based gradient matching, ICLR (2024).
7. Nickl, Xu, Tailor, Moellenhoff, Khan, The memory-perturbation equation, NeurIPS (2023)

GPT-2 with Bayes

Better performance & uncertainty at the same cost [5]



Trained on OpenWebText data (49.2B tokens).

On 773M, we get a gain of 0.5 in perplexity.

On 355M, we get a gain of 0.4 in perplexity.

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. Shen et al. Variational Learning is Effective for Large Deep Networks, Under review.

Exponential Family

Natural parameters → λ

Sufficient Statistics → $T(\theta)$

Expectation parameters → μ

$$q(\theta) \propto \exp\left[\lambda^\top T(\theta)\right] \qquad \mu := \mathbb{E}_q[T(\theta)]$$

$$\mathcal{N}(\theta|m, S^{-1}) \propto \exp\left[-\frac{1}{2}(\theta - m)^\top S(\theta - m)\right]$$
$$\propto \exp\left[(Sm)^\top \theta + \mathrm{Tr}\left(-\frac{S}{2}\theta\theta^\top\right)\right]$$

Gaussian distribution	$q(\theta) := \mathcal{N}(\theta\|m, S^{-1})$
Natural parameters	$\lambda := \{Sm, -S/2\}$
Expectation parameters	$\mu := \{\mathbb{E}_q(\theta), \mathbb{E}_q(\theta\theta^\top)\}$

1. Wainwright and Jordan, Graphical Models, Exp Fams, and Variational Inference Graphical models 2008
2. Malago et al., Towards the Geometry of Estimation of Distribution Algos based on Exp-Fam, FOGA, 2011

Bayes and Conjugate Computations [1]

Multiplication of distribution = addition of (natural) params

Bayes rule: $\text{posterior} \propto \text{lik} \times \text{prior}$

$$e^{\lambda_{\text{post}}^\top T(\theta)} \propto e^{\lambda_{\text{lik}}^\top T(\theta)} \times e^{\lambda_{\text{prior}}^\top T(\theta)}$$

$$\text{log-posterior} = \text{log-lik} + \text{log-prior}$$

$$\lambda_{\text{post}} = \lambda_{\text{lik}} + \lambda_{\text{prior}}$$

This idea can be generalized through natural-gradients.

$$\lambda_{\text{post}} = \nabla_{\mu} \mathbb{E}_q [\text{log-lik} + \text{log-prior}]$$

Natural gradient (∇_{μ}) — Posterior "approximation" ($\mathbb{E}_q$)

1. Khan and Lin, Conjugate computation variational inference, AISTATS, 2017.

Bayes Rule as (Natural) Gradient Descent

$$\lambda_{\text{post}} \leftarrow \lambda_{\text{lik}} + \lambda_{\text{prior}}$$

Expected log-lik and log-prior are linear in μ [1]

$$\mathbb{E}_q[\text{log-lik}] = \lambda_{\text{lik}}^\top \mathbb{E}_q[T(\theta)] = \lambda_{\text{lik}}^\top \mu$$

Gradient wrt μ is simply the natural parameter

$$\nabla_\mu \mathbb{E}_q[\text{log-lik}] = \lambda_{\text{lik}}$$

So Bayes' rule can be written as (for an arbitrary q)

$$\lambda_{\text{post}} \leftarrow \nabla_\mu \mathbb{E}_q[\text{log-lik} + \text{log-prior}]$$

As an analogy, think of least-square = 1-step of Newton

1. Khan, Variational-Bayes Made Easy, AABI 2023.

Approximate Bayes

Bayes rule:

$$\text{posterior} \propto \text{lik} \times \text{prior}$$

Bayes as optimization [1], aka variational inference:

$$\min_{q \in \mathcal{Q}} \; \mathbb{E}_q[\text{log-lik}] + \text{KL}(q \| \text{prior})$$

Generalized Approx Bayesian learning:

log-lik + log-prior

$$\min_{q \in \mathcal{Q}} \; \mathbb{E}_{q(\theta)}[\ell(\theta)] - \mathcal{H}(q)$$

Entropy

Posterior approximation (expo-family)

1. Zellner, Optimal information processing and Bayes's theorem, The American Statistician, 1988.

The Bayesian Learning Rule

$$\min_{\theta} \ \ell(\theta) \qquad \text{vs} \qquad \min_{q \in \mathcal{Q}} \ \mathbb{E}_{q(\theta)}[\ell(\theta)] - \mathcal{H}(q)$$

Entropy

Posterior approximation (expo-family)

Bayesian Learning Rule [1,2] (natural-gradient descent)

Natural and Expectation parameters of q

$$\lambda \leftarrow \lambda - \rho \nabla_{\mu} \left\{ \mathbb{E}_q[\ell(\theta)] - \mathcal{H}(q) \right\}$$

Old belief   New information = natural gradients

Exploiting posterior's information geometry to derive existing algorithms as special instances by approximating q and natural gradients.

1. Khan and Rue, The Bayesian Learning Rule, JMLR, 2023
2. Khan and Lin. "Conjugate-computation variational inference…." AIstats (2017).

Warning!

- This natural gradient is different from the one what we (often) encounter in machine learning for Maximum-Likelihood
  - In MLE, the loss is the negative log probability distribution

$$\min_{\theta} - \log q(\theta) \Rightarrow F(\theta)^{-1} \nabla \log q(\theta)$$

  - Here, loss and distribution are two different entities, even possible unrelated

$$\min_{q} \mathbb{E}_q[\ell(\theta)] - \mathscr{H}(q) \Rightarrow F(\lambda)^{-1} \nabla_{\lambda} \mathbb{E}_q[\ell(\theta)]$$

Gradient Descent from Bayesian Learning Rule

(Euclidean) gradients as natural gradients

Bayesian learning rule:

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 (New)	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) (New)	Gaussian (diagonal cov.)	Gauss-Newton Hessian approx. in Adam & no square-root scaling	4.4
Variational OGN (New)	——"——	Remove delta method from OGN	4.4
BayesBiNN (New)	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 (New)	Mixture of Exp-family	None	5.4

See Section 1.3.1 in Khan and Rue, 2021

Gradient Descent from BLR

GD: $\theta \leftarrow \theta - \rho \nabla_\theta \ell(\theta)$

BLR: $m \leftarrow m - \rho \nabla_m \ell(m)$

"Global" to "local"
(the delta method)
$\mathbb{E}_q[\ell(\theta)] \approx \ell(m)$

$$m \leftarrow m - \rho \nabla_m \mathbb{E}_q[\ell(\theta)]$$

$$\lambda \leftarrow \lambda - \rho \nabla_\mu \left(\mathbb{E}_q[\ell(\theta)] - \mathcal{H}(q) \right)$$

Derived by choosing Gaussian with fixed covariance

Gaussian distribution $q(\theta) := \mathcal{N}(m, 1)$

Natural parameters $\lambda := m$

Expectation parameters $\mu := \mathbb{E}_q[\theta] = m$

Entropy $\mathcal{H}(q) := \log(2\pi)/2$

See Section 1.3.2 in Khan and Rue, 2021

Newton's Method from BLR

Newton's method: $\theta \leftarrow \theta - H_\theta^{-1}\left[\nabla_\theta \ell(\theta)\right]$

$$Sm \leftarrow (1-\rho) Sm - \rho \nabla_{\mathbb{E}_q(\theta)} \mathbb{E}_q[\ell(\theta)]$$

$$-\frac{1}{2} S \leftarrow (1-\rho)\left(-\frac{1}{2} S\right) - \rho \nabla_{\mathbb{E}_q(\theta\theta^\top)} \mathbb{E}_q[\ell(\theta)]$$

$$S \leftarrow (1-\rho) S + \rho 2 \nabla_{\mathbb{E}_q(\theta\theta^\top)} \mathbb{E}_q[\ell(\theta)]$$

$$\lambda \leftarrow \lambda - \rho \nabla_\mu \left(\mathbb{E}_q[\ell(\theta)] - \mathcal{H}(q)\right)$$

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

$$-\nabla_\mu \mathcal{H}(q) = \lambda$$

Derived by choosing a multivariate Gaussian

Gaussian distribution $q(\theta) := \mathcal{N}(\theta | m, S^{-1})$

Natural parameters $\lambda := \{Sm, -S/2\}$

Expectation parameters $\mu := \{\mathbb{E}_q(\theta), \mathbb{E}_q(\theta\theta^\top)\}$

1. Khan, et al. "Fast and scalable Bayesian deep learning by weight-perturbation in Adam." *ICML* (2018).

See Section 1.3.2 in Khan and Rue, 2021

Newton's Method from BLR

Newton's method: $\theta \leftarrow \theta - H_\theta^{-1}\left[\nabla_\theta \ell(\theta)\right]$

Set ρ =1 to get $\quad m \leftarrow m - H_m^{-1}[\nabla_m \ell(m)]$

$$m \leftarrow m - \rho S^{-1} \nabla_m \ell(m)$$
$$S \leftarrow (1-\rho) S + \rho H_m$$

Delta Method
$\mathbb{E}_q[\ell(\theta)] \approx \ell(m)$

Express in terms of gradient and Hessian of loss:

$$\nabla_{\mathbb{E}_q(\theta)} \mathbb{E}_q[\ell(\theta)] = \mathbb{E}_q[\nabla_\theta \ell(\theta)] - 2\mathbb{E}_q[H_\theta] m$$

$$\nabla_{\mathbb{E}_q(\theta\theta^\top)} \mathbb{E}_q[\ell(\theta)] = \mathbb{E}_q[H_\theta]$$

$$Sm \leftarrow (1-\rho) Sm - \rho \nabla_{\mathbb{E}_q(\theta)} \mathbb{E}_q[\ell(\theta)]$$
$$S \leftarrow (1-\rho) S - \rho 2 \nabla_{\mathbb{E}_q(\theta\theta^\top)} \mathbb{E}_q[\ell(\theta)]$$

1. Khan, et al. "Fast and scalable Bayesian deep learning by weight-perturbation in Adam." *ICML* (2018).

See Section 4.2 in Khan and Rue, 2021

RMSprop/Adam from BLR

RMSprop

$$s \leftarrow (1-\rho)s + \rho[\hat{\nabla}\ell(\theta)]^2$$

$$\theta \leftarrow \theta - \alpha(\sqrt{s}+\delta)^{-1}\hat{\nabla}\ell(\theta)$$

BLR for Gaussian approx

$$S \leftarrow (1-\rho)S + \rho(H_\theta)$$

$$m \leftarrow m - \alpha S^{-1}\nabla_\theta \ell(\theta)$$

To get RMSprop, make the following choices

- Restrict covariance to be diagonal
- Replace Hessian by square of gradients
- Add square root for scaling vector

For Adam, use a Heavy-ball term with KL divergence as momentum (Appendix E in [1])

1. Khan, et al. "Fast and scalable Bayesian deep learning by weight-perturbation in Adam." *ICML* (2018).

BLR for large deep networks

RMSprop/Adam

$$\hat{g} \leftarrow \hat{\nabla}\ell(\theta)$$
$$\hat{h} \leftarrow \hat{g}^2$$
$$h \leftarrow (1-\rho)h + \rho\hat{h}$$
$$\theta \leftarrow \theta - \alpha(\hat{g} + \delta m)/(\sqrt{h} + \delta)$$

BLR variant called
Improved Variational Online Newton (IVON)

$$\hat{g} \leftarrow \hat{\nabla}\ell(\theta) \text{ where } \theta \sim \mathcal{N}(m, \sigma^2)$$
$$\hat{h} \leftarrow \hat{g} \cdot (\theta - m)/\sigma^2$$
$$h \leftarrow (1-\rho)h + \rho\hat{h} + \rho^2(h - \hat{h})^2/(2(h+\delta))$$
$$m \leftarrow m - \alpha(\hat{g} + \delta m)/(h + \delta)$$
$$\sigma^2 \leftarrow 1/(N(h+\delta))$$

Code to be released this month!
Initialization of h (& scaling with N) matter.

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).
4. Shen et al. “Variational Learning is effective for large neural networks.” (Under review)

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

Watch Thomas Moellenhoff's talk at https://www.youtube.com/watch?v=LQInlN5EU7E.

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

Thomas Möllenhoff[1], Yuesong Shen[2], Gian Maria Marconi[1]
Peter Nickl[1], Mohammad Emtiyaz Khan[1]

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

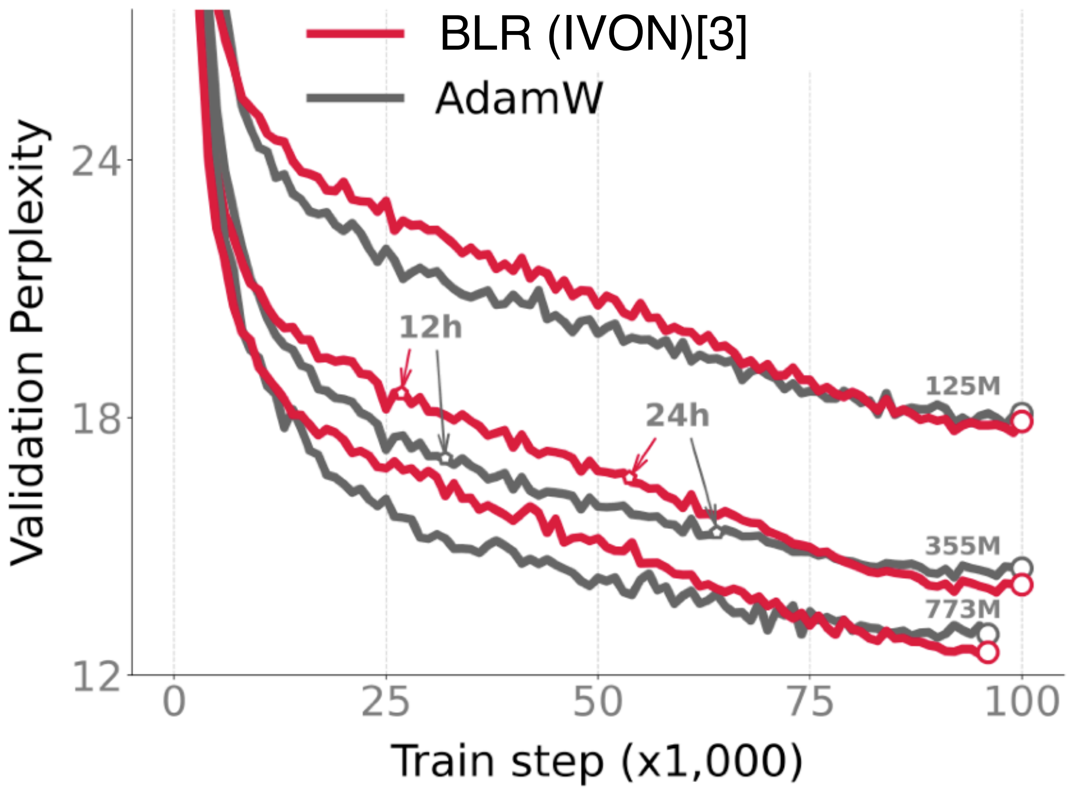
2 Computer Vision Group
Technical University of Munich, Germany

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).

GPT-2 with Bayes

Better performance and uncertainty at the same cost



Trained on OpenWebText data (49.2B tokens).

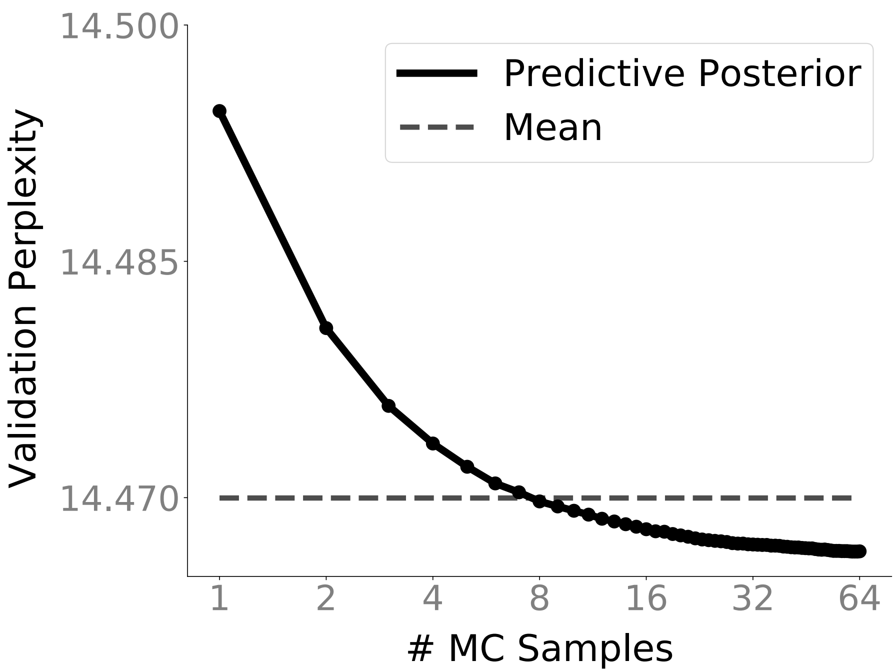
On 773M, we get a gain of 0.5 in perplexity.

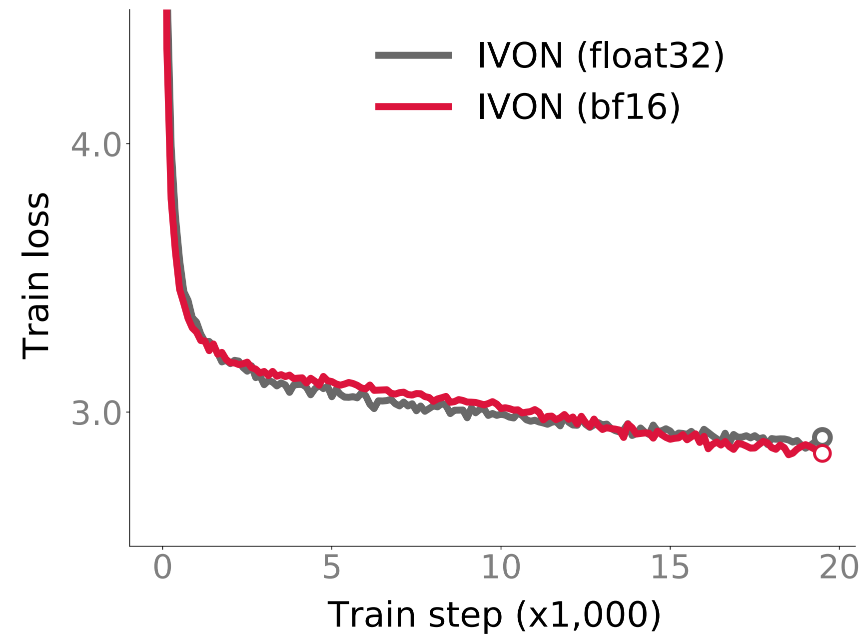
On 355M, we get a gain of 0.4 in perplexity.

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. Shen et al. “Variational Learning is effective for large neural networks.” (Under review)

GPT-2 with Bayes

Posterior averaging improve the result. Can also train on low-precision (a stable optimizer)

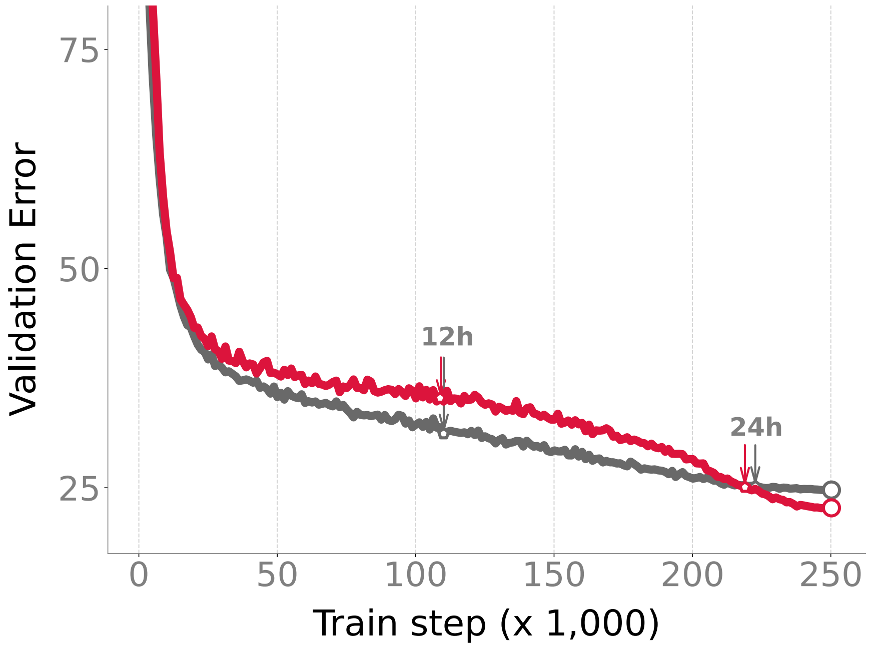


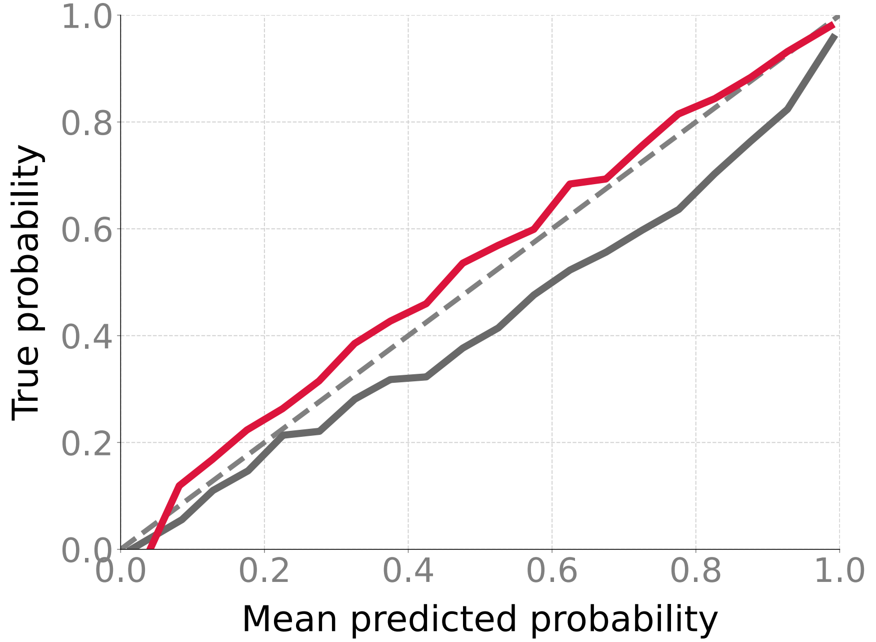


1. Khan, et al. "Fast and scalable Bayesian deep learning by weight-perturbation in Adam." *ICML* (2018).
2. Osawa et al. “Practical Deep Learning with Bayesian Principles.” NeurIPS (2019).
3. Shen et al. “Variational Learning is effective for large neural networks.” (Under review)

ImageNet on ResNet-50 (25.6M)

2% better accuracy over AdamW and 1% over SGD. Better calibration (ECE of 0.022 vs 0.066)

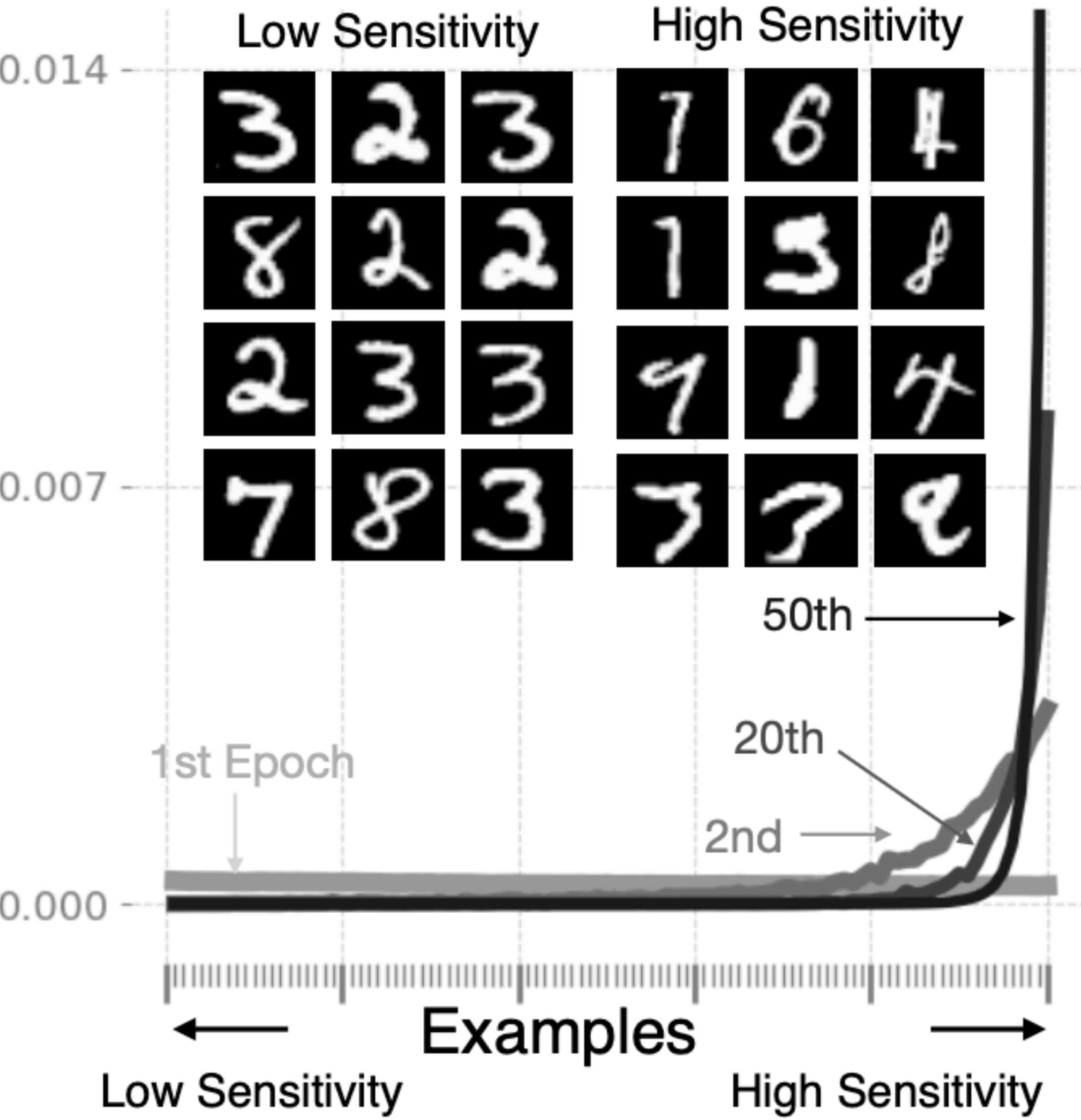




ImageNet on ResNet-50 (25.6M)

No severe overfitting like AdamW while improving accuracy over SGD consistently & better uncertainty

Dataset & Model	Epochs	Method	Top-1 Acc. ↑	Top-5 Acc. ↑	NLL ↓	ECE ↓	Brier ↓
ImageNet-1k **ResNet-50** (25.6M params)	100	AdamW	74.56$_{\pm 0.24}$	92.05$_{\pm 0.17}$	1.018$_{\pm 0.012}$	0.043$_{\pm 0.001}$	0.352$_{\pm 0.003}$
		SGD	**76.18**$_{\pm 0.09}$	**92.94**$_{\pm 0.05}$	**0.928**$_{\pm 0.003}$	0.019$_{\pm 0.001}$	**0.330**$_{\pm 0.001}$
		IVON@mean	**76.14**$_{\pm 0.11}$	92.83$_{\pm 0.04}$	0.934$_{\pm 0.002}$	0.025$_{\pm 0.001}$	**0.330**$_{\pm 0.001}$
		IVON	**76.24**$_{\pm 0.09}$	**92.90**$_{\pm 0.04}$	**0.925**$_{\pm 0.002}$	**0.015**$_{\pm 0.001}$	**0.330**$_{\pm 0.001}$
	200	AdamW	+2% 75.16$_{\pm 0.14}$	92.37$_{\pm 0.03}$	1.018$_{\pm 0.003}$	0.066$_{\pm 0.002}$	0.349$_{\pm 0.002}$
		SGD	+1% 76.63$_{\pm 0.45}$	93.21$_{\pm 0.25}$	0.917$_{\pm 0.026}$	0.038$_{\pm 0.009}$	0.326$_{\pm 0.006}$
		IVON@mean	77.30$_{\pm 0.08}$	93.58$_{\pm 0.05}$	0.884$_{\pm 0.002}$	0.035$_{\pm 0.002}$	**0.316**$_{\pm 0.001}$
		IVON	**77.46**$_{\pm 0.07}$	**93.68**$_{\pm 0.04}$	**0.869**$_{\pm 0.002}$	**0.022**$_{\pm 0.002}$	**0.315**$_{\pm 0.001}$
TinyImageNet **ResNet-18** (11M params, wide)	200	AdamW	+15% 47.33$_{\pm 0.90}$	71.54$_{\pm 0.95}$	6.823$_{\pm 0.235}$	0.421$_{\pm 0.008}$	0.913$_{\pm 0.018}$
		SGD	+1% 61.39$_{\pm 0.18}$	82.30$_{\pm 0.22}$	1.811$_{\pm 0.010}$	0.138$_{\pm 0.002}$	0.536$_{\pm 0.002}$
		IVON@mean	**62.41**$_{\pm 0.15}$	**83.77**$_{\pm 0.18}$	1.776$_{\pm 0.018}$	0.150$_{\pm 0.005}$	0.532$_{\pm 0.002}$
		IVON	**62.68**$_{\pm 0.16}$	**84.12**$_{\pm 0.24}$	**1.528**$_{\pm 0.010}$	**0.019**$_{\pm 0.004}$	**0.491**$_{\pm 0.001}$
TinyImageNet **PreResNet-110** (4M params, deep)	200	AdamW	+10% 50.65$_{\pm 0.0*}$	74.94$_{\pm 0.0*}$	4.487$_{\pm 0.0*}$	0.357$_{\pm 0.0*}$	0.812$_{\pm 0.0*}$
		AdaHessian	55.03$_{\pm 0.53}$	78.49$_{\pm 0.34}$	2.971$_{\pm 0.064}$	0.272$_{\pm 0.005}$	0.690$_{\pm 0.008}$
		SGD	+2% 59.39$_{\pm 0.50}$	81.34$_{\pm 0.30}$	2.040$_{\pm 0.040}$	0.176$_{\pm 0.006}$	0.577$_{\pm 0.007}$
		IVON @mean	**60.85**$_{\pm 0.39}$	**83.89**$_{\pm 0.14}$	1.584$_{\pm 0.009}$	0.053$_{\pm 0.002}$	**0.514**$_{\pm 0.003}$
		IVON	**61.25**$_{\pm 0.48}$	**84.13**$_{\pm 0.17}$	**1.550**$_{\pm 0.009}$	**0.049**$_{\pm 0.002}$	**0.511**$_{\pm 0.003}$
CIFAR-100 **ResNet-18** (11M params, wide)	200	AdamW	+11% 64.12$_{\pm 0.43}$	86.85$_{\pm 0.51}$	3.357$_{\pm 0.071}$	0.278$_{\pm 0.005}$	0.615$_{\pm 0.008}$
		SGD	+.7% 74.46$_{\pm 0.17}$	92.66$_{\pm 0.06}$	1.083$_{\pm 0.007}$	0.113$_{\pm 0.001}$	0.376$_{\pm 0.001}$
		IVON@mean	74.51$_{\pm 0.24}$	92.74$_{\pm 0.19}$	1.284$_{\pm 0.013}$	0.152$_{\pm 0.003}$	0.399$_{\pm 0.002}$
		IVON	**75.14**$_{\pm 0.34}$	**93.30**$_{\pm 0.19}$	**0.912**$_{\pm 0.009}$	**0.021**$_{\pm 0.003}$	**0.344**$_{\pm 0.003}$



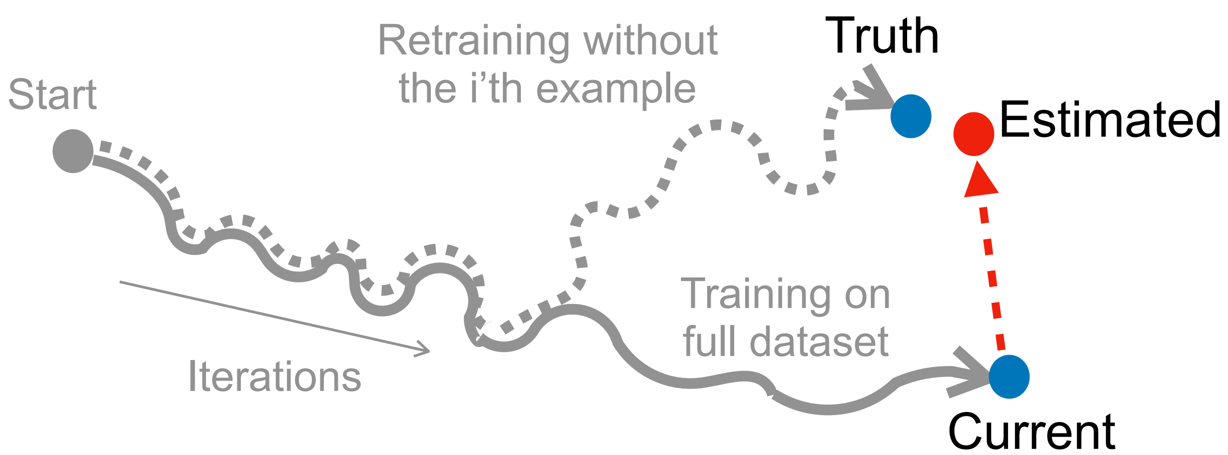
Sensitivity to data is easy to compute "during" training.

MNIST on MLP. Also work at large scale (ImageNet)

1. Nickl, Xu, Tailor, Moellenhoff, Khan, The memory-perturbation equation, NeurIPS, 2023

Sensitivity to Training Data

Past information with most influence on the present

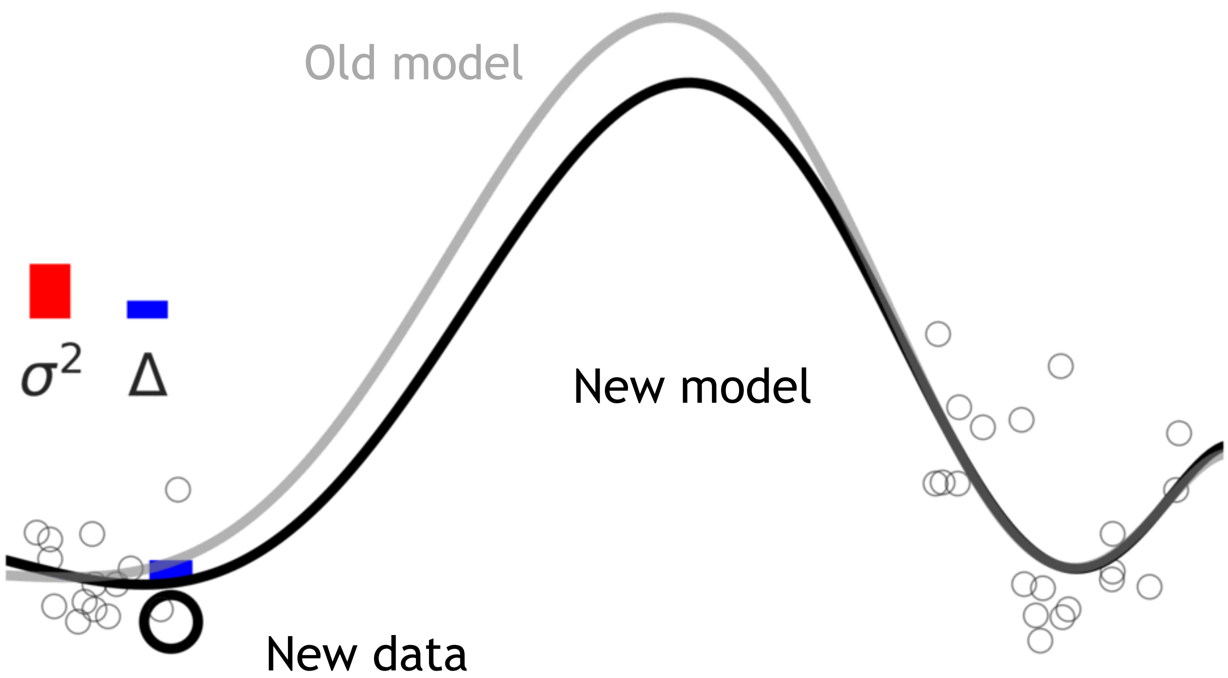


Estimating it without retraining: Using the BLR, we can recover all sorts of influence criteria used in literature.

Memory Perturbation

How sensitive is a model to its training data?

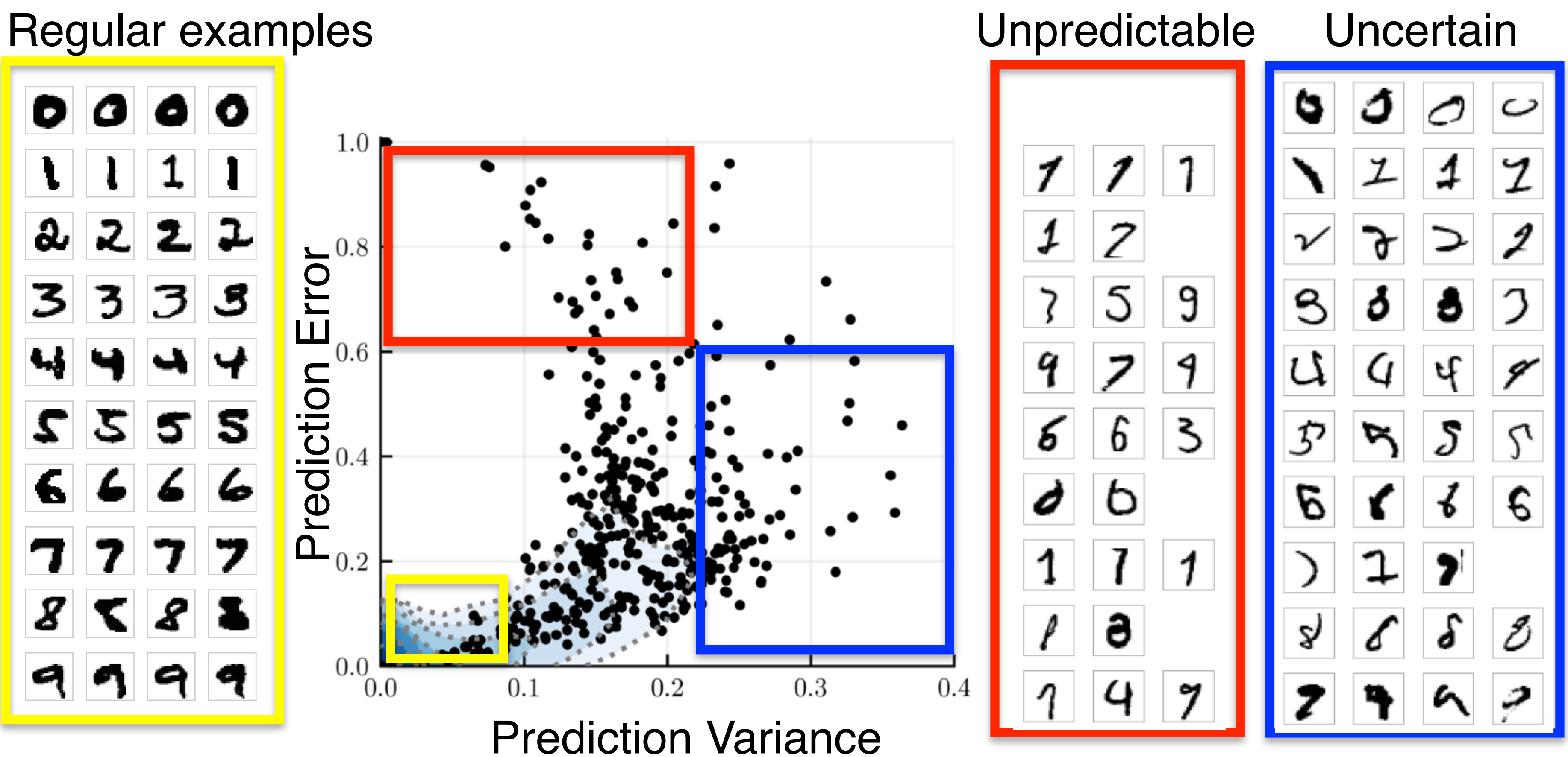
Deviation (Δ) = predictionError *predictionVariance



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

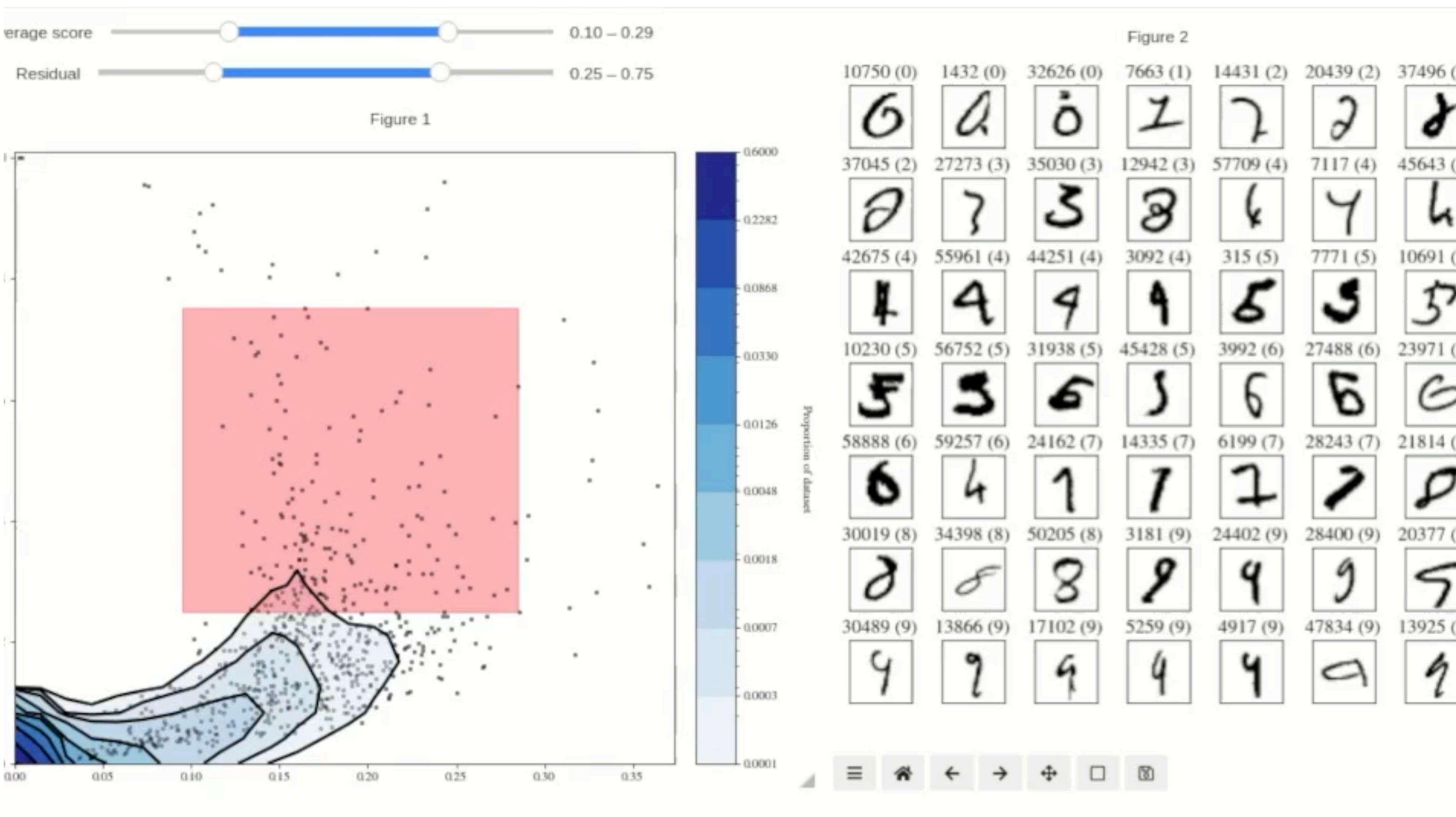
Understand generic ML models and algorithms.



1. Tailor, Chang, Swaroop, Nalisnick, Solin, Khan, Memory maps to understand models (under review)

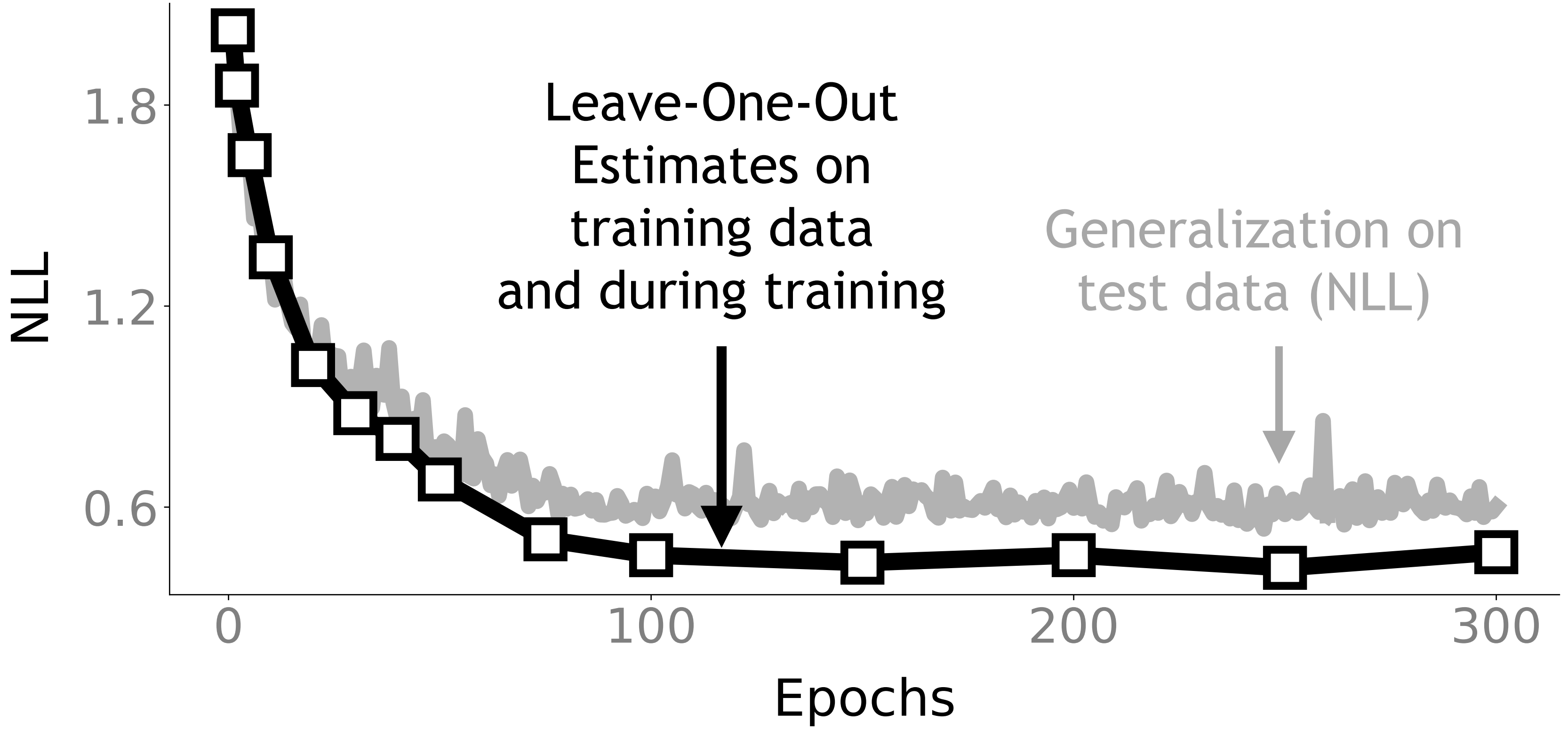
A Tool for Data-Scientists

Understand the memory of a model.



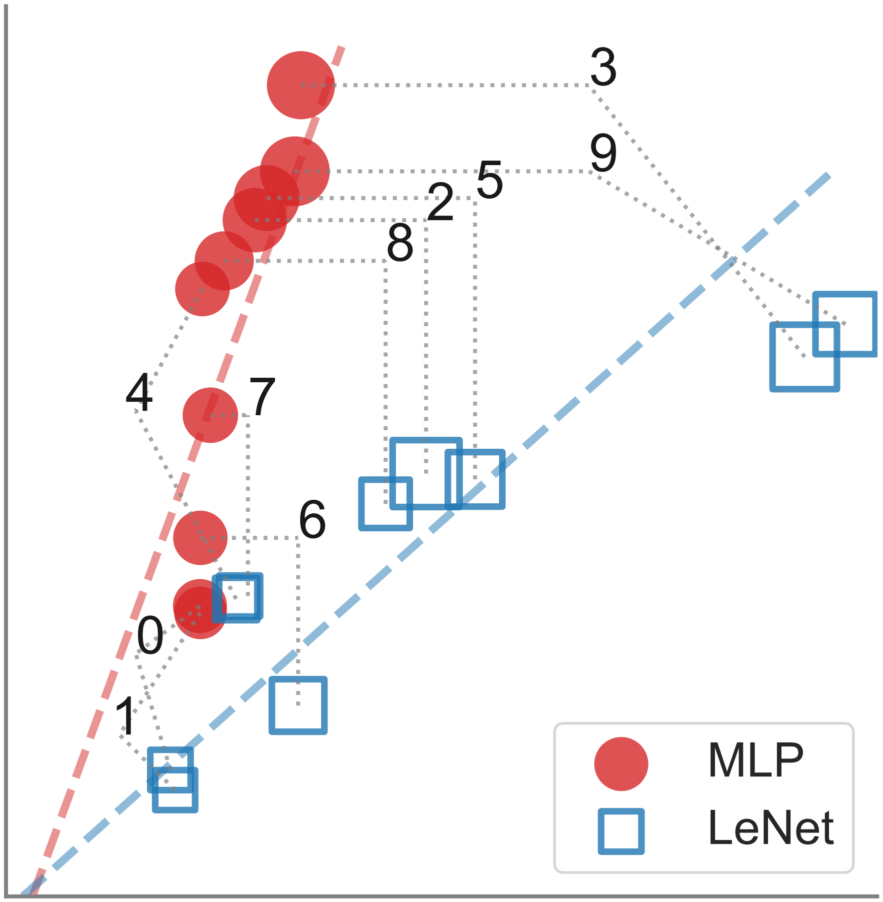
Predict Generalization during Training

CIFAR10 on ResNet-20 using IVON. SGD or Adam do not work as well.



Answering "What-If" Questions

What if we removed a class from MNIST?

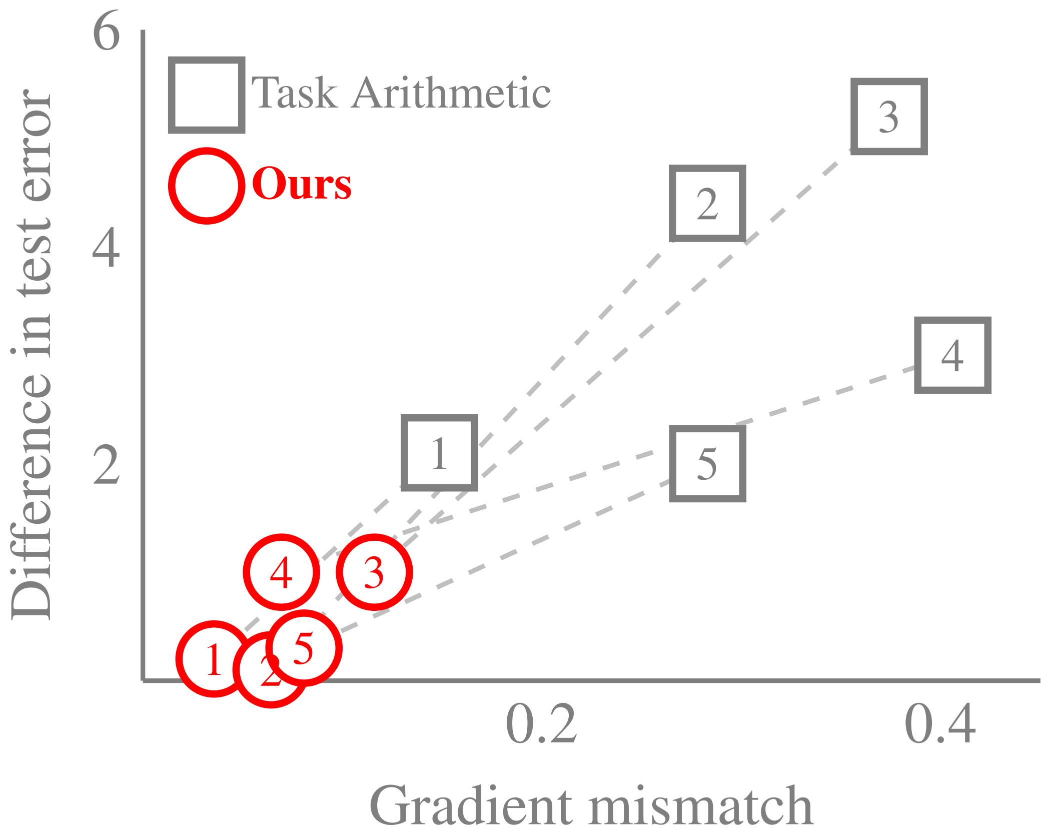


Estimates on training data (no retraining)

Test Performance (NLL) by brute-force retraining

Answering “What-If” Questions

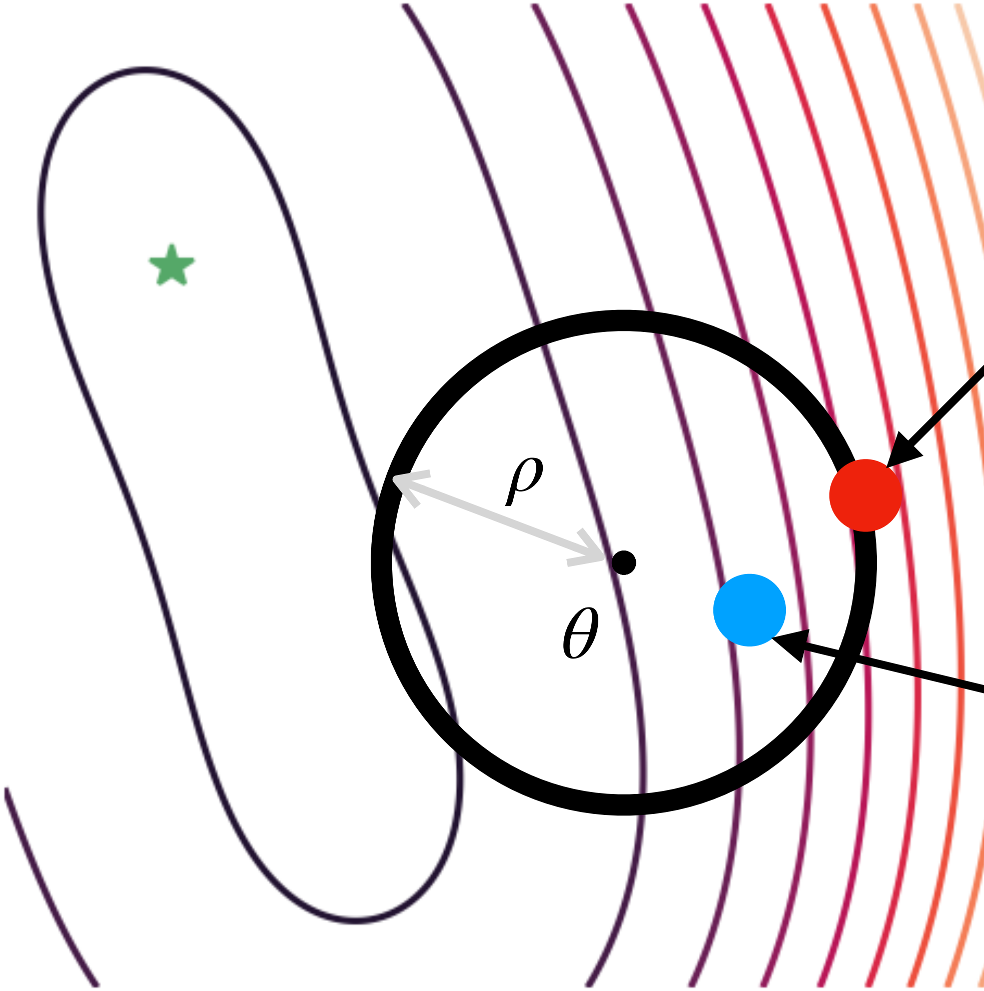
What if we merge fine-tuned large-language models?



RoBERTa
on IMDB

1. Daheim et al. Model merging by uncertainty-based gradient matching, ICLR (2024).

SAM as an Optimal relaxation of Bayes



SAM: $\sup_{|\epsilon|<\rho} \ell(\theta + \epsilon)$

Our work: Fenchel Biconjugate

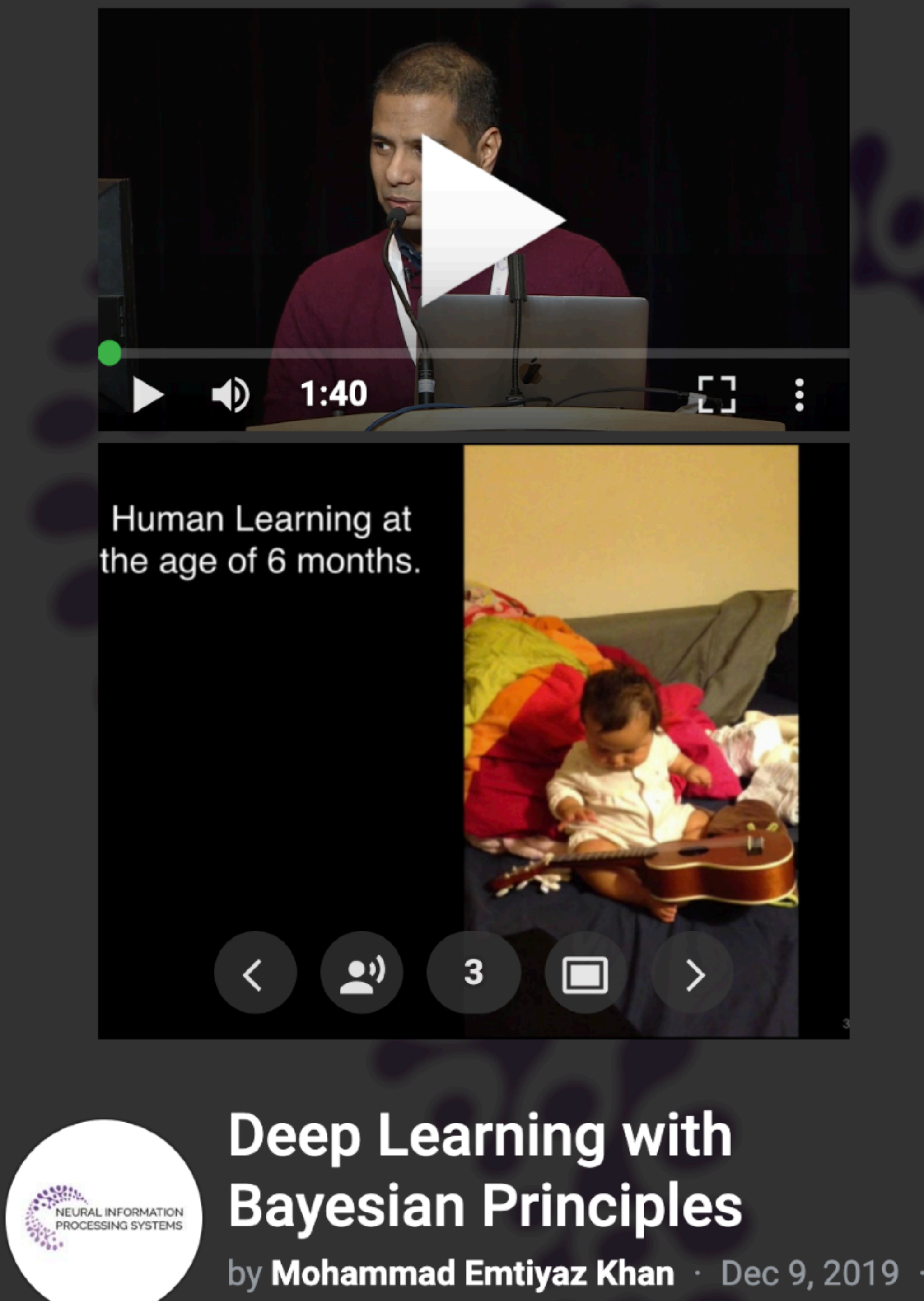
Bayes: $\mathbb{E}_{\epsilon \sim \mathcal{N}(0,\sigma^2)}[\ell(\theta + \epsilon)]$

1. Foret et al. Sharpness-Aware Minimization for Efficiently Improving Generalization, ICLR, 2021
2. Moellenhoff and Khan, SAM as an Optimal Relaxation of Bayes, Under review, 2022

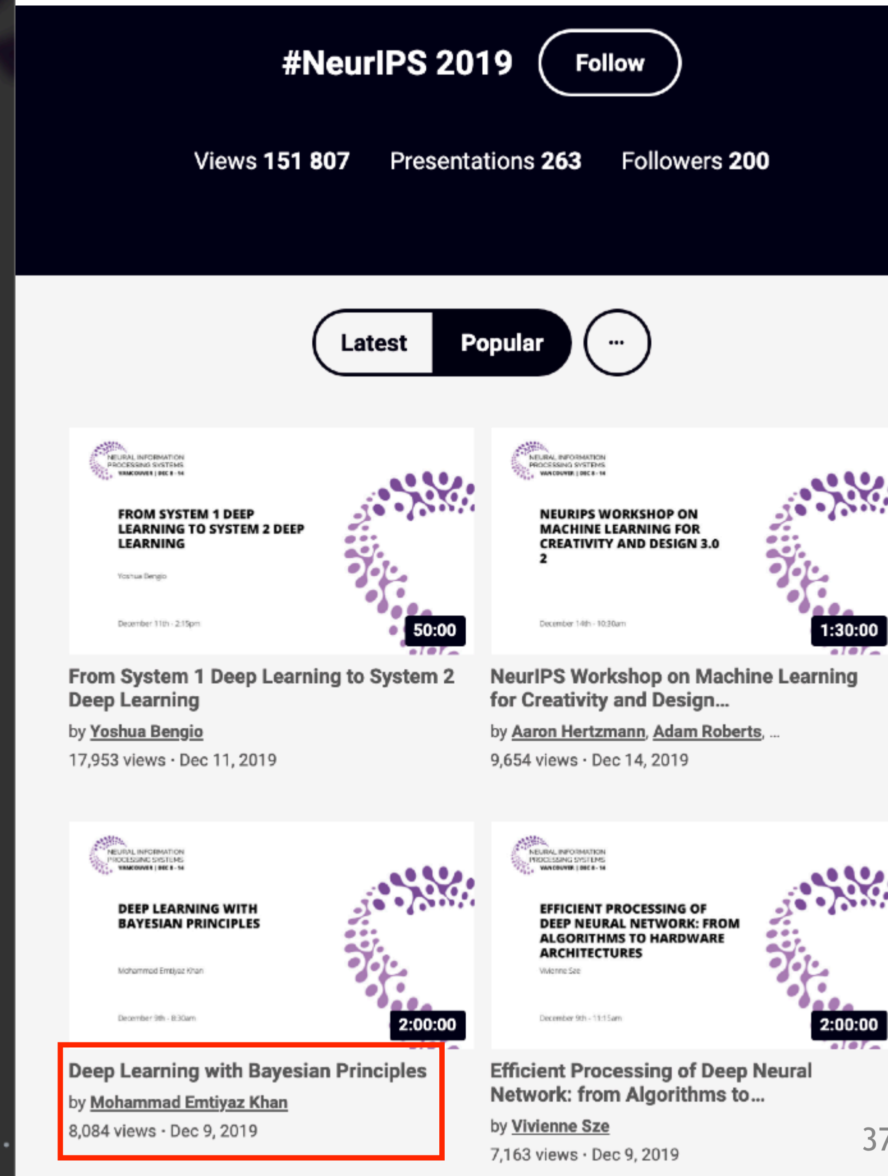
Bayesian Learning Rule [1]

- Bridge DL & Bayesian learning [2-5]
  - SOTA on GPT-2 and ImageNet [5]
- Improve DL [5-7]
  - Calibration, uncertainty, memory etc.
  - Understand and fix model behavior
- Towards human-like quick adaptation

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. Shen et al. Variational Learning is Effective for Large Deep Networks, Under review.
6. Daheim et al. Model merging by uncertainty-based gradient matching, ICLR (2024).
7. Nickl, Xu, Tailor, Moellenhoff, Khan, The memory-perturbation equation, NeurIPS (2023)



NeurIPS 2019 Tutorial



The webpage is available at https://bayesduality.github.io/, and Twitter account @BayesDuality

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 (2021-2027) and Kakenhi Grants (2019-2021).

Team Approx-Bayes

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).

We are always looking for new collaborations.

Keigo Nishida
Postdoctoral Researcher
RIKEN BDR

Zhedong Liu
Postdoctoral Researcher

Peter Nickl
Research Assistant

Joseph Austerweil
Visiting Scientist
University of Winsconsin-Madison

Pierre Alquier
Visiting Scientist
ESSEC Business School

Dharmesh Tailor
Remote Collaborator
University of Amsterdam