Machine Learning & Bayes

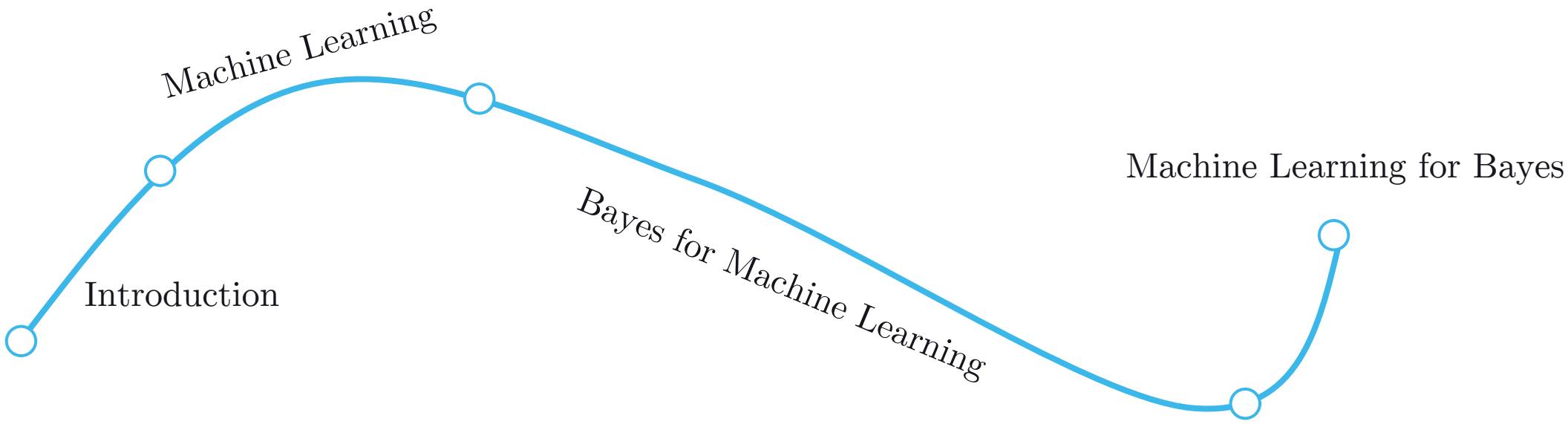
- Synergies and Challenges -

Martin Trapp

Aalto University

Learn Bayes Spring 2025 @ Karolinska Institutet

Outline

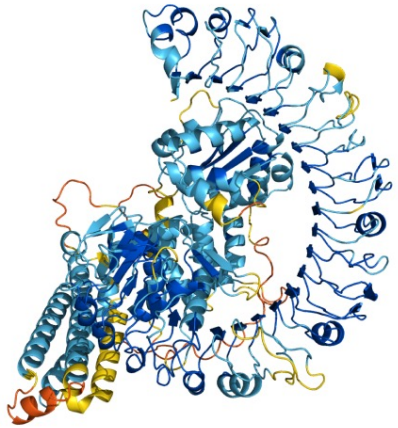

Introduction
Machine Learning
Bayes for Machine Learning
Machine Learning for Bayes

Success of Modern Machine Learning

Generative tasks

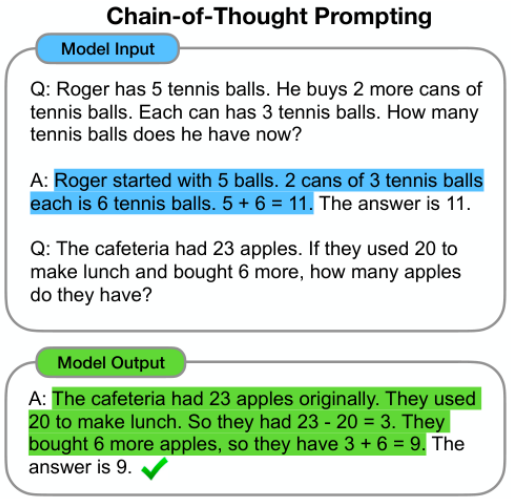
Image Generation

Predictive tasks



Protein Prediction [1]

"Reasoning" tasks



Chain-of-Thought in LLMs [2]

1: Jumper, J., et al. (2021). Highly accurate protein structure prediction with AlphaFold. Nature.
2: Wei, J., et al. (2022). Chain-of-thought prompting elicits reasoning in large language models. In NeurIPS.

Machine Learning in Healthcare

Personalised treatment

Drug discovery and development

ML-supported diagnosis

ML-assisted surgery

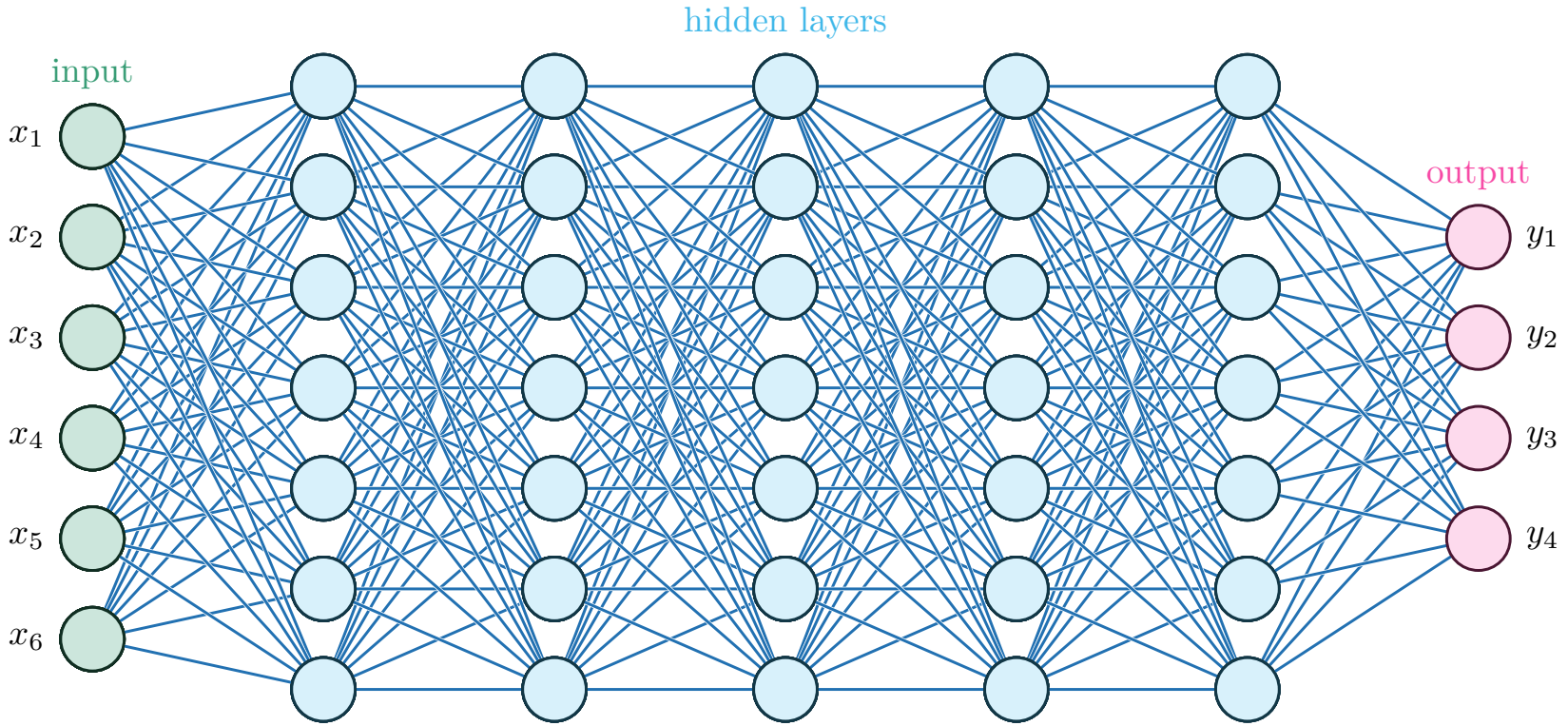
Automated image analysis

Reminder on (modern) ML

... an incomplete and inaccurate picture of machine learning...

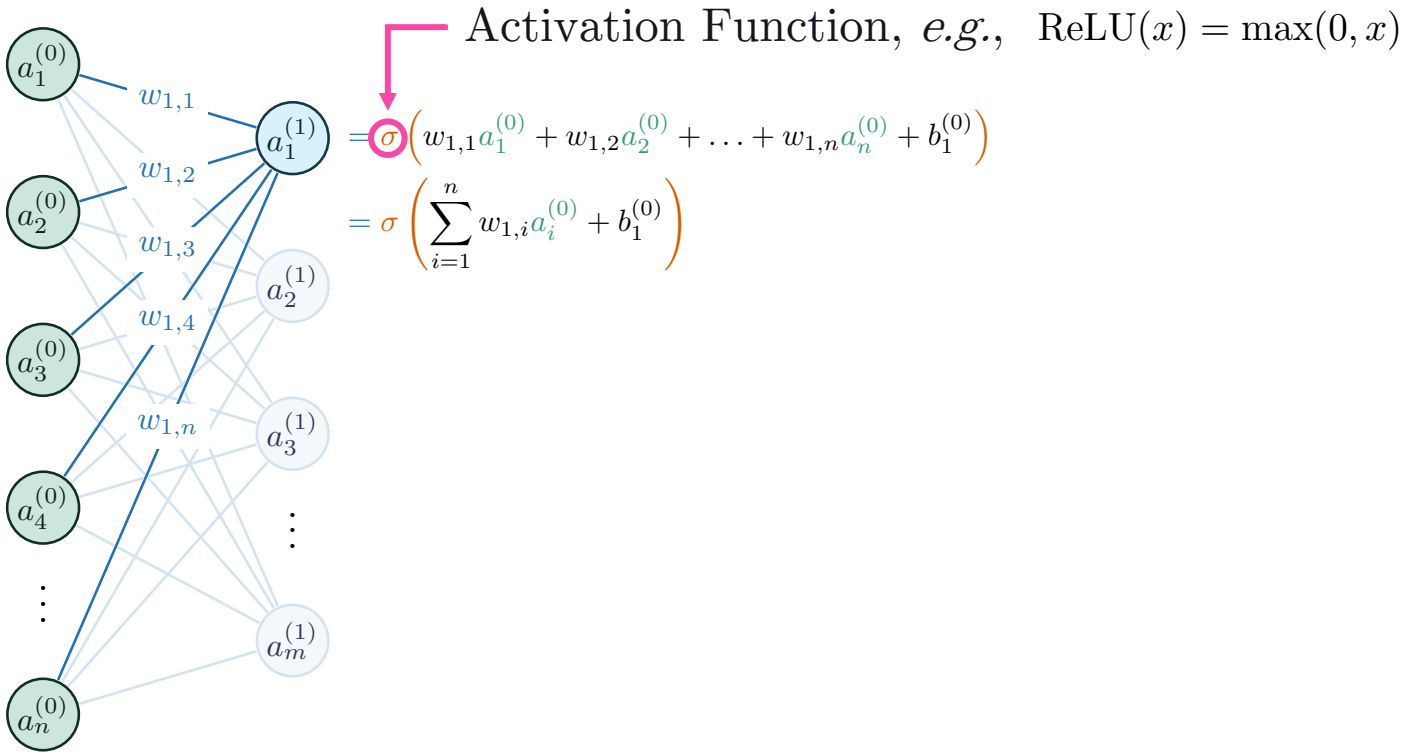
Reminder on Deep Learning

- The central objects in deep learning (modern ML) are artificial neural networks or neural network architectures (NNs).



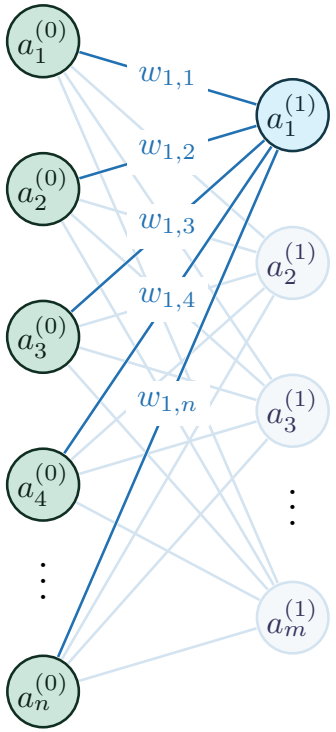
Reminder on Deep Learning

- The central objects in deep learning (modern ML) are artificial neural networks or neural network architectures (NNs).



Reminder on Deep Learning

- The central objects in deep learning (modern ML) are artificial neural networks or neural network architectures (NNs).



$$= \sigma\left(w_{1,1}a_1^{(0)} + w_{1,2}a_2^{(0)} + \ldots + w_{1,n}a_n^{(0)} + b_1^{(0)}\right)$$

$$= \sigma\left(\sum_{i=1}^{n} w_{1,i}a_i^{(0)} + b_1^{(0)}\right)$$

$$\begin{pmatrix} a_1^{(1)} \\ a_2^{(1)} \\ \vdots \\ a_m^{(1)} \end{pmatrix} = \sigma\left[\begin{pmatrix} w_{1,1} & w_{1,2} & \ldots & w_{1,n} \\ w_{2,1} & w_{2,2} & \ldots & w_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{m,1} & w_{m,2} & \ldots & w_{m,n} \end{pmatrix}\begin{pmatrix} a_1^{(0)} \\ a_2^{(0)} \\ \vdots \\ a_n^{(0)} \end{pmatrix} + \begin{pmatrix} b_1^{(0)} \\ b_2^{(0)} \\ \vdots \\ b_m^{(0)} \end{pmatrix}\right]$$

$$\mathbf{a}^{(1)} = \sigma\left(\mathbf{W}^{(0)}\mathbf{a}^{(0)} + \mathbf{b}^{(0)}\right)$$

Reminder on Deep Learning

- The central objects in deep learning (modern ML) are artificial neural networks or neural network architectures (NNs).
- NNs exploit function composition to define flexible non-linear function approximators with simple local operations.

$$f(\boldsymbol{x}) = f^{(L)} \circ \cdots \circ f^{(2)} \circ f^{(1)}(\boldsymbol{x})$$

Reminder on Deep Learning

- The central objects in deep learning (modern ML) are artificial neural networks or neural network architectures (NNs).
- NNs exploit function composition to define flexible non-linear function approximators with simple local operations.

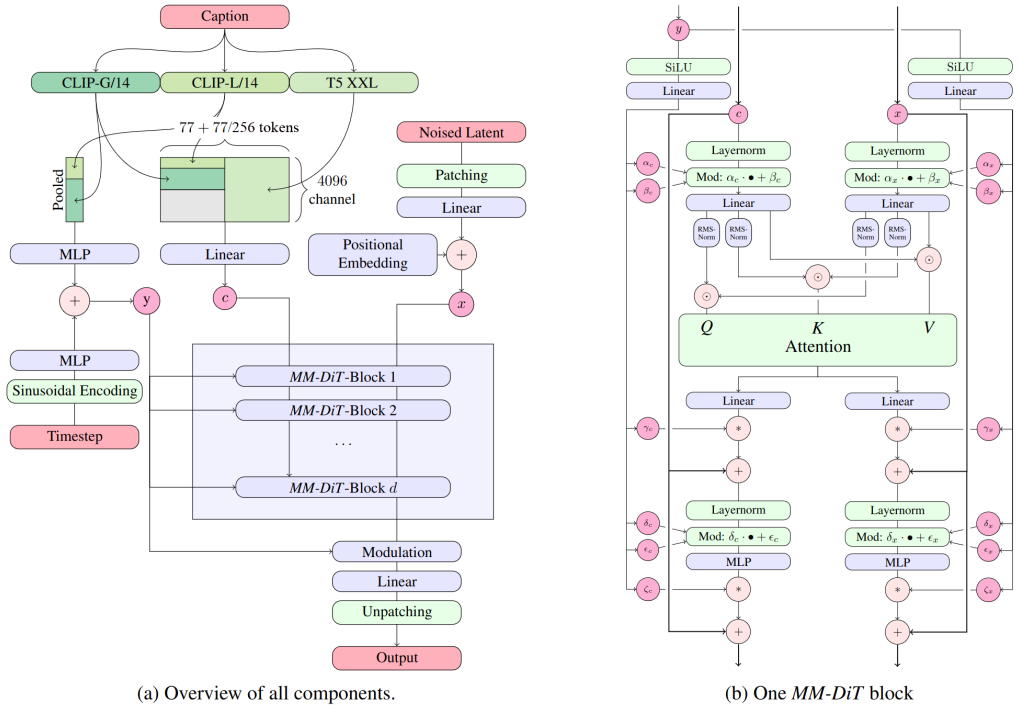
$$f(\boldsymbol{x}) = f^{(L)} \circ \cdots \circ f^{(2)} \circ f^{(1)}(\boldsymbol{x})$$

- The network weights (parameters) are learned through backpropagation (chain rule) for a given loss function.

PyTorch

Reminder on Deep Learning

In practice, modern NNs are complex systems:



(a) Overview of all components.

(b) One *MM-DiT* block

Schematics of the Stable Diffusion 3.5 Model for Image Generation

https://huggingface.co/stabilityai/stable-diffusion-3.5-large

Machine Learning in Healthcare

Personalised treatment

Drug discovery and development

ML-supported diagnosis

ML-assisted surgery

Automated image analysis

Machine Learning in Healthcare

Many Applications are High-risk!

Personalised treatment

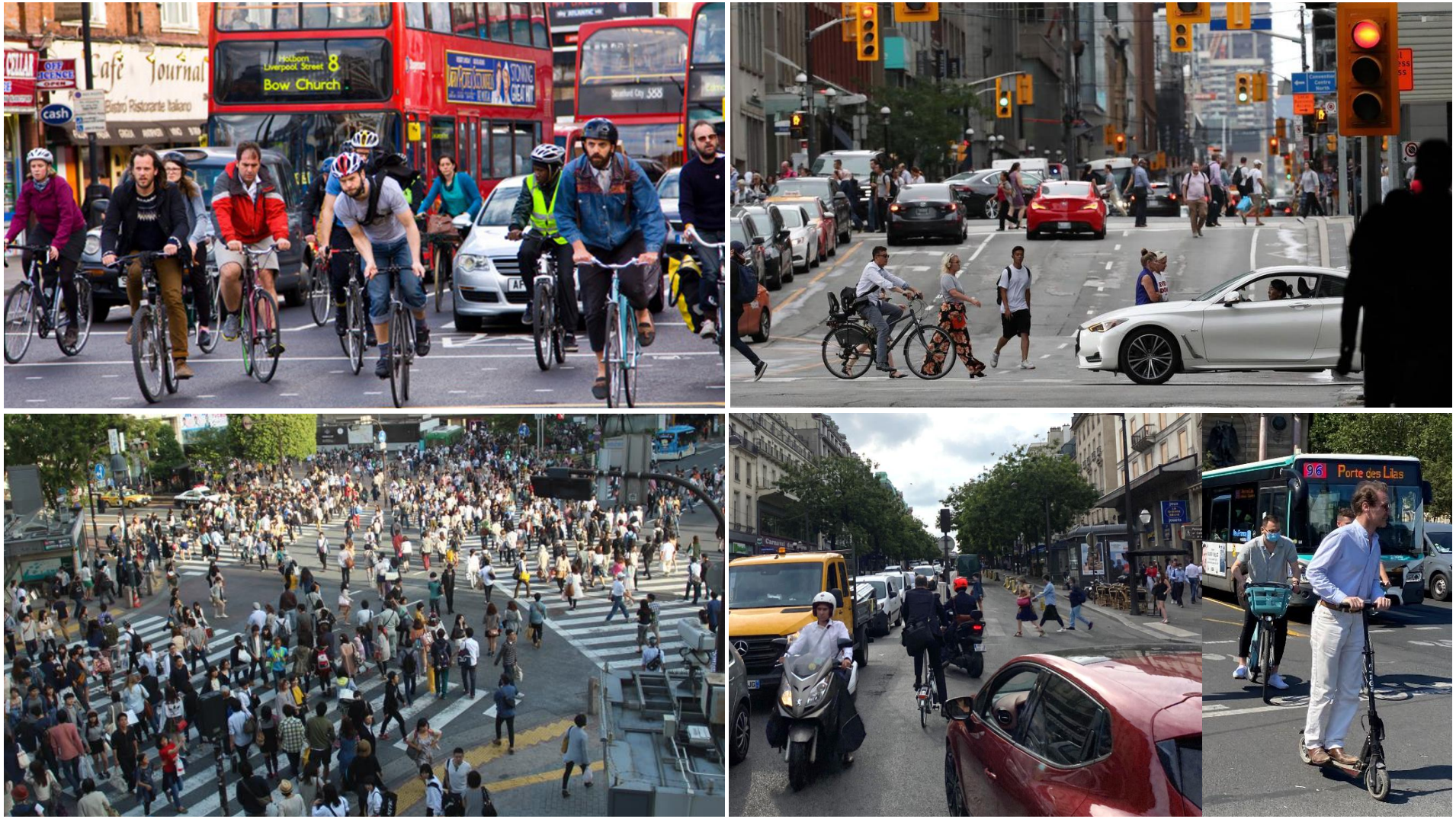
Drug discovery and development

ML-supported diagnosis

ML-assisted surgery

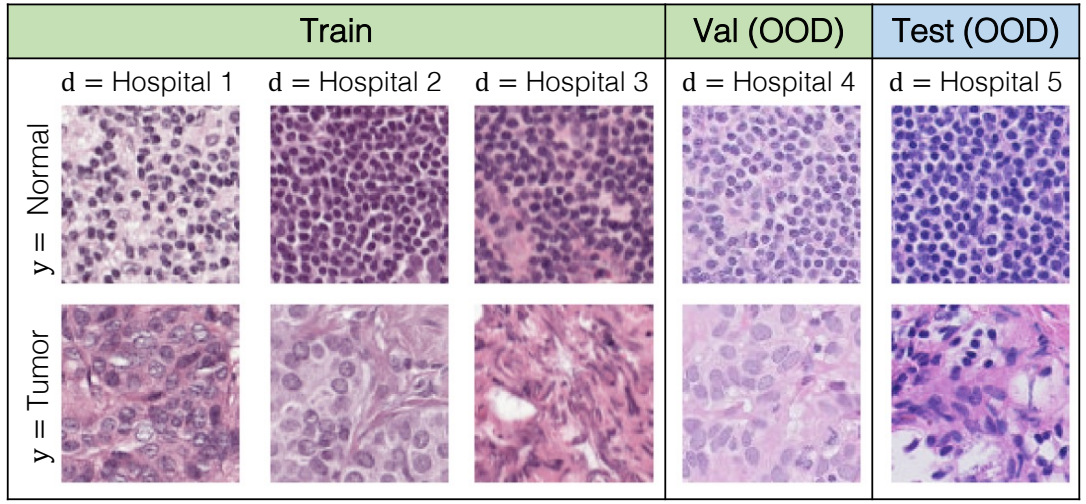
Automated image analysis

The Real World is Messy



A World of Uncertainties

Distribution Shifts

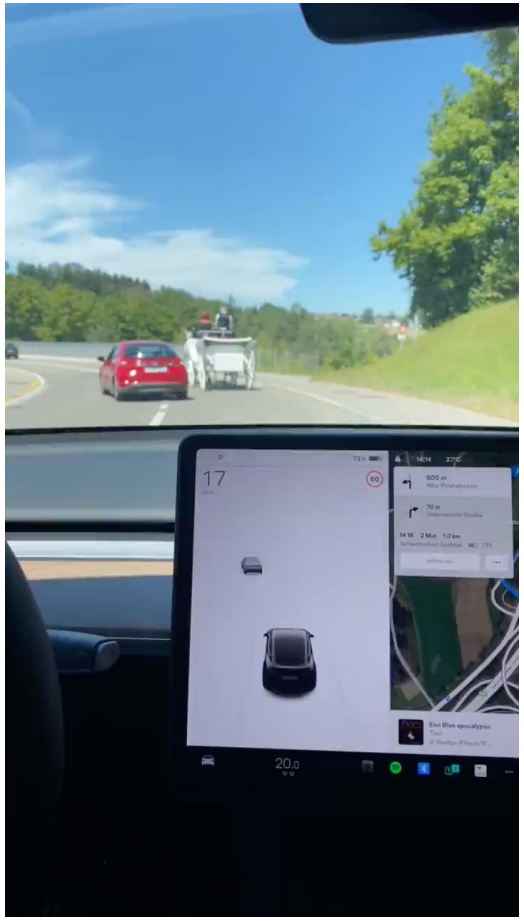


Koh, P W. et al. (2021) WILDS: A Benchmark of in-the-Wild Distribution Shifts. In *ICML*.

Various Sources of Noise

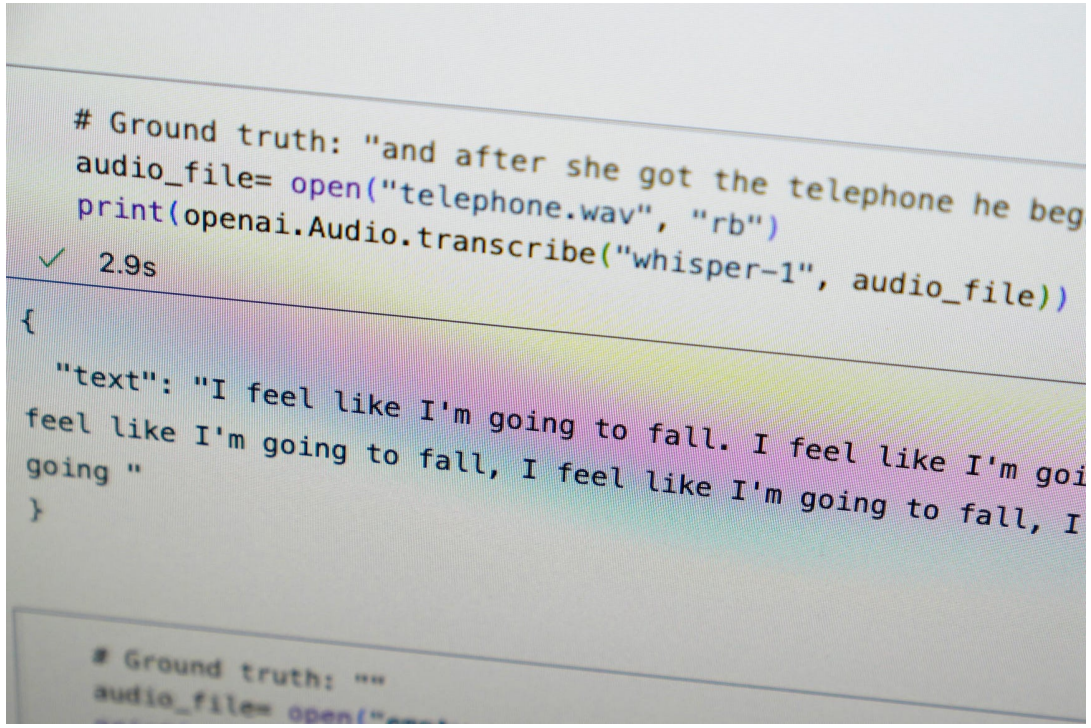
Sabour, S. et al. (2021) SpotlessSplats: Ignoring Distractors in 3D Gaussian Splatting. In *ICML*.

A World of Uncertainties



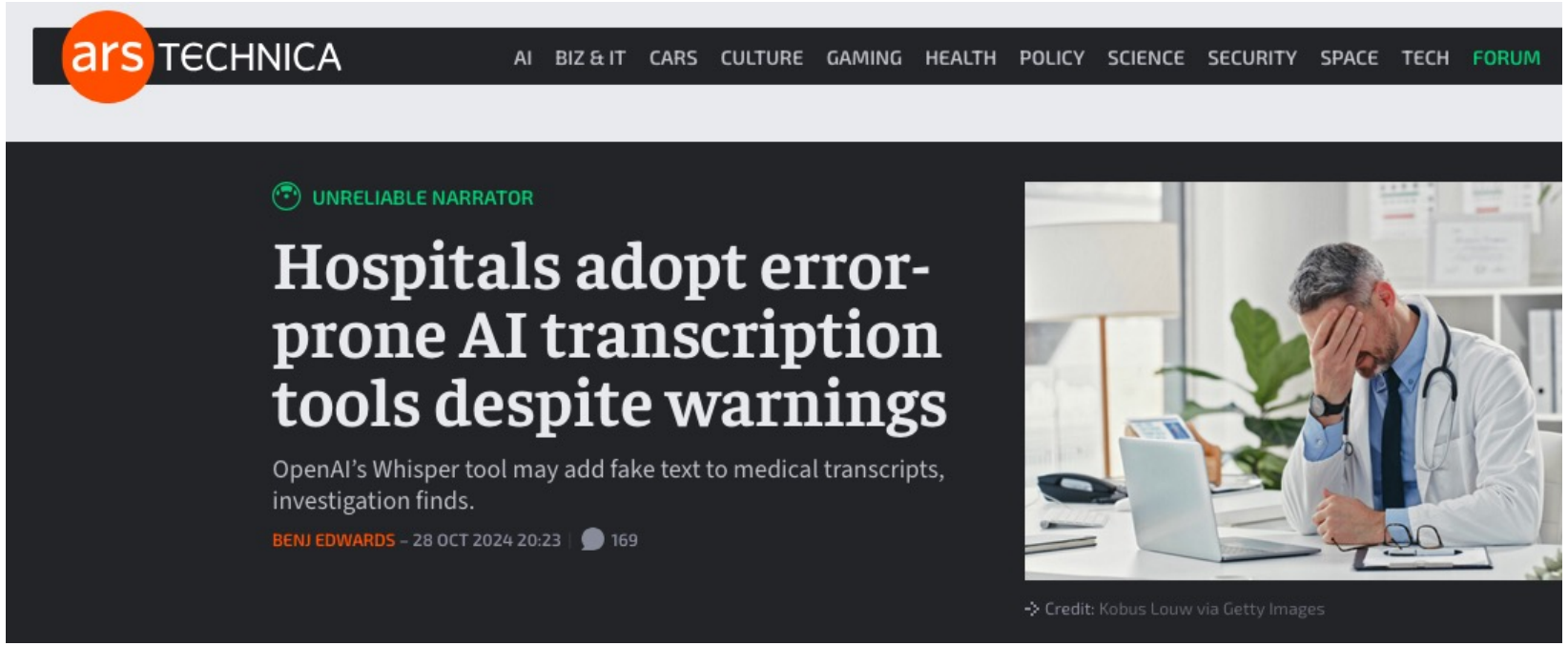
Out-of-domain data (Tesla)

A World of Uncertainties



Hallucinations (OpenAI Whisper)

A World of Uncertainties



https://arstechnica.com/ai/2024/10/hospitals-adopt-error-prone-ai-transcription-tools-despite-warnings/

Expectations for ML Systems

Systems and models should be safe, trustworthy and reliable.

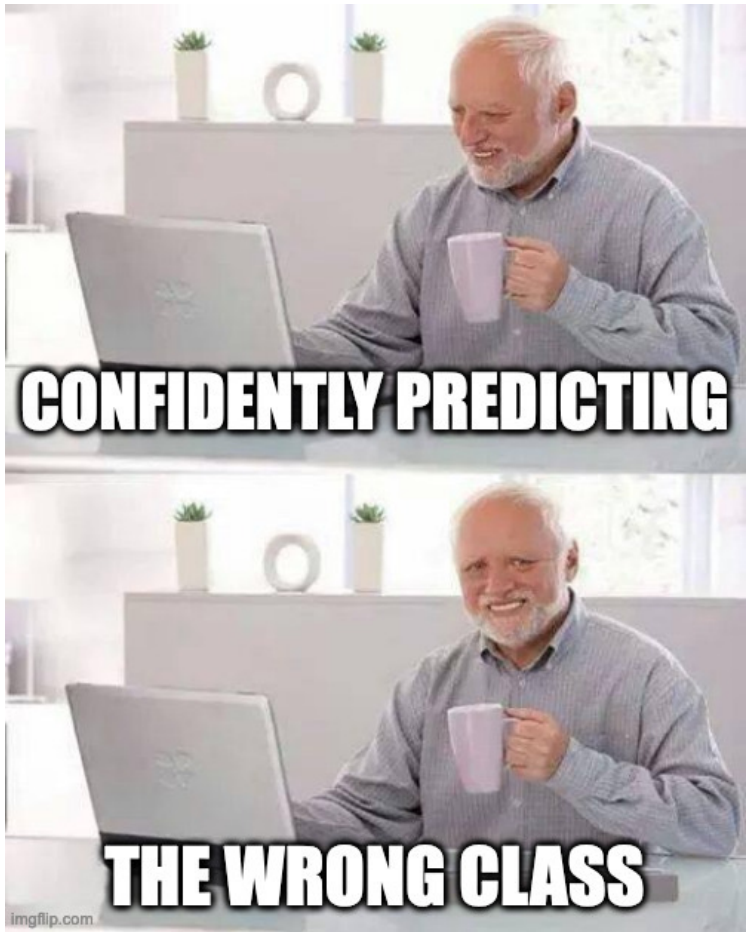
For this, we require that they are: *(incomplete list)*

- Accurate in their predictions
- Robust to input perturbations (noise, adversarial attacks)
- "Know when they don't know" (recognise out-of-domain data)
- Act if they are uncertain (*e.g.*, active learning, reasoning)

However, ...

- Accurate in their predictions
- Sensitive to pertubations[1]
- Don't know when they don't know[2]
- Overconfident in their predictions[3]

Considered "unreliable"



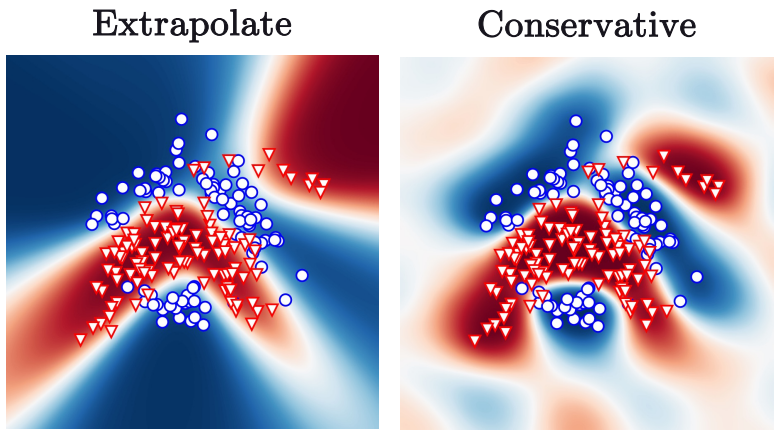
1: Szegedy, C. et al. (2014). Intriguing properties of neural networks. In *ICLR*.
2: Nalisnick, E. et al. (2019). Do deep generative models know what they don't know? In *ICLR*.
3: Guo, C. et al. (2017). On calibration of modern neural networks. In *ICML*.


WELL, WHAT DO
WE DO NOW?
imgflip.com

Bayes for Machine Learning

Bayesian Deep Learning

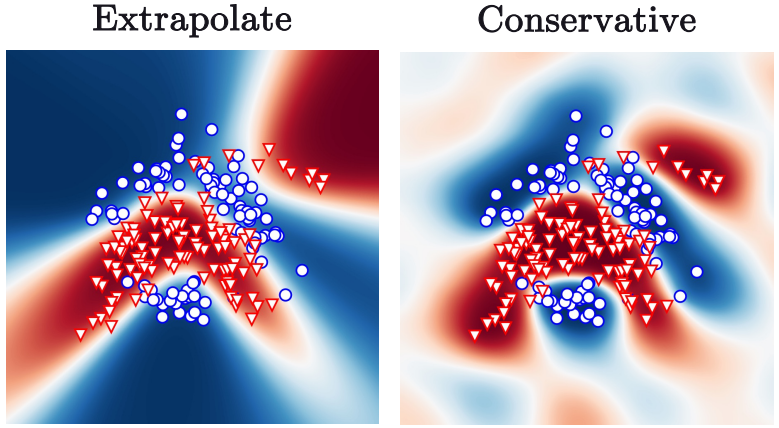
Functional Priors



Meronen, L. et al. (2021). Periodic activation functions induce stationarity. In *NeurIPS*.

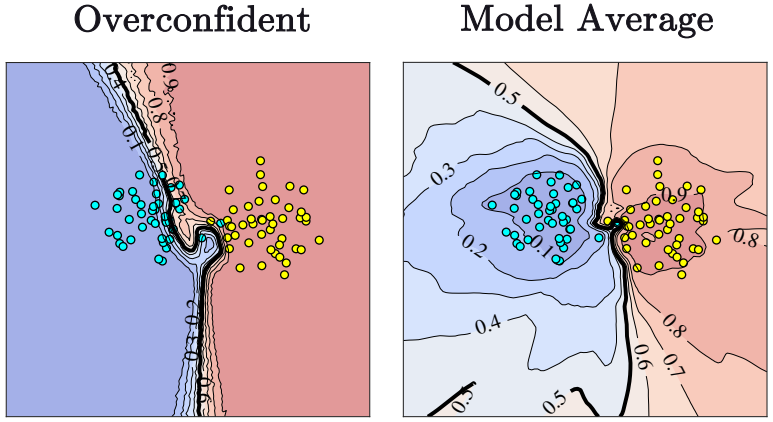
Bayesian Deep Learning

Functional Priors



Meronen, L. et al. (2021). Periodic activation functions induce stationarity. In *NeurIPS*.

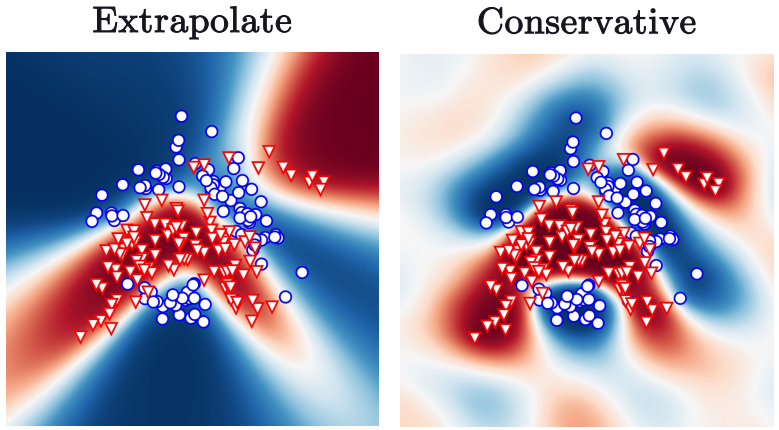
Uncertainty Quant. & Overconfidence



Kristiadi, A. et al. (2020). Being Bayesian, Even Just a Bit, Fixes Overconfidence in ReLU Networks. In *ICML*.

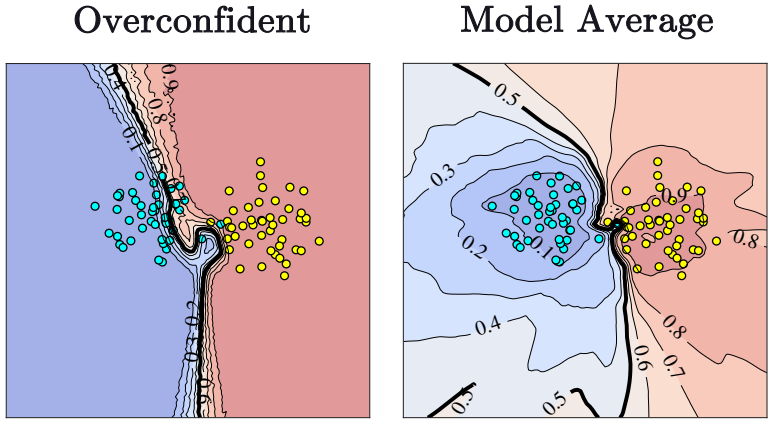
Bayesian Deep Learning

Functional Priors



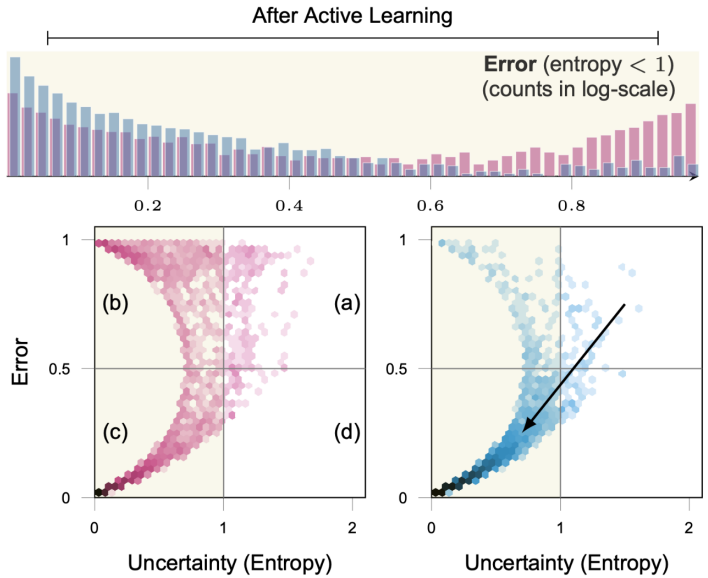
Meronen, L. et al. (2021). Periodic activation functions induce stationarity. In *NeurIPS*.

Uncertainty Quant. & Overconfidence



Kristiadi, A. et al. (2020). Being Bayesian, Even Just a Bit, Fixes Overconfidence in ReLU Networks. In *ICML*.

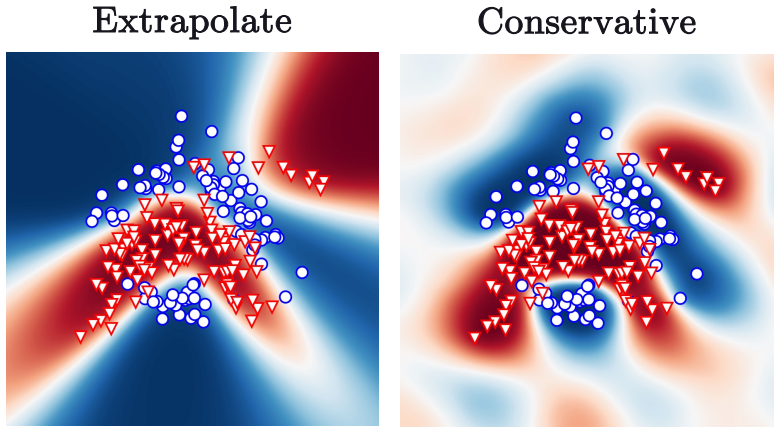
Active Learning



Baumann, A. et al. (2025). Post-hoc Probabilistic Vision-Language Models. ArXiv.

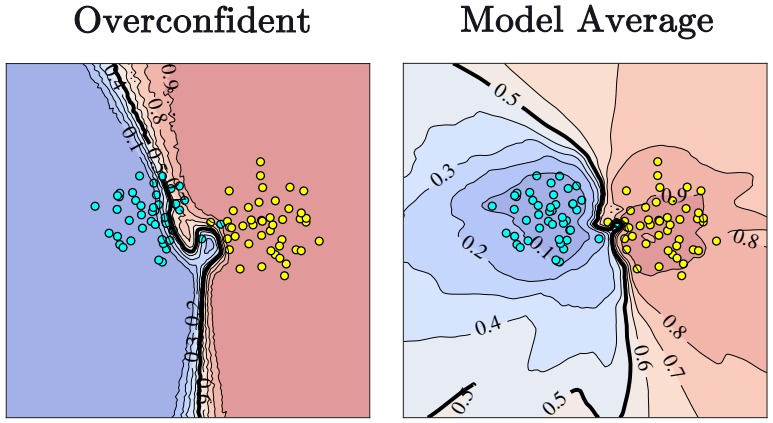
Bayesian Deep Learning

Functional Priors



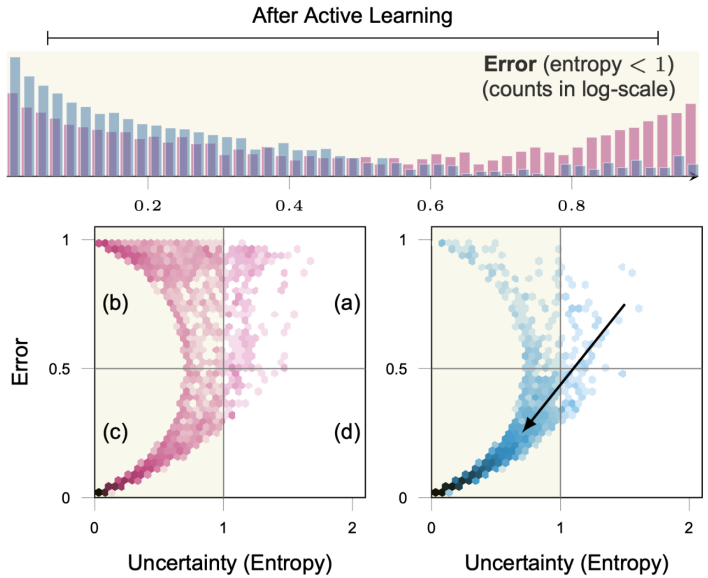
Meronen, L. et al. (2021). Periodic activation functions induce stationarity. In *NeurIPS*.

Uncertainty Quant. & Overconfidence



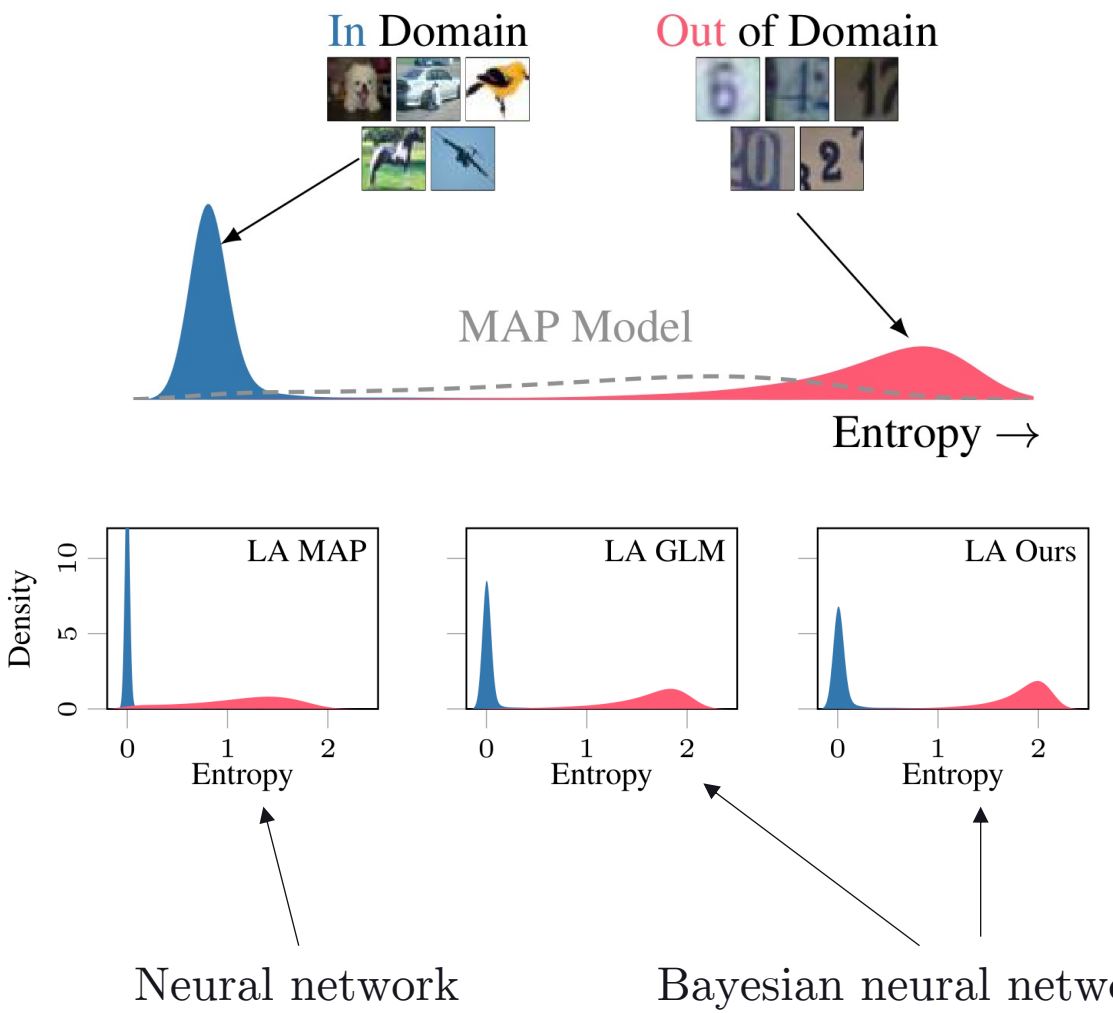
Kristiadi, A. et al. (2020). Being Bayesian, Even Just a Bit, Fixes Overconfidence in ReLU Networks. In *ICML*.

Active Learning

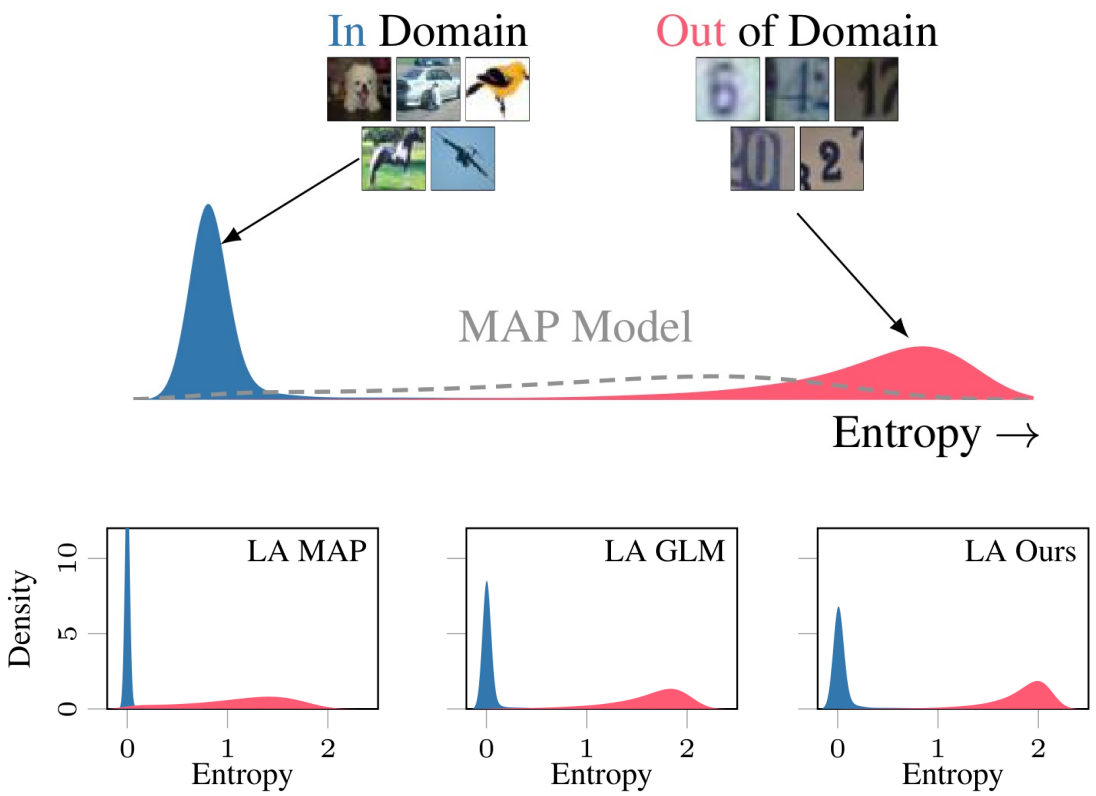


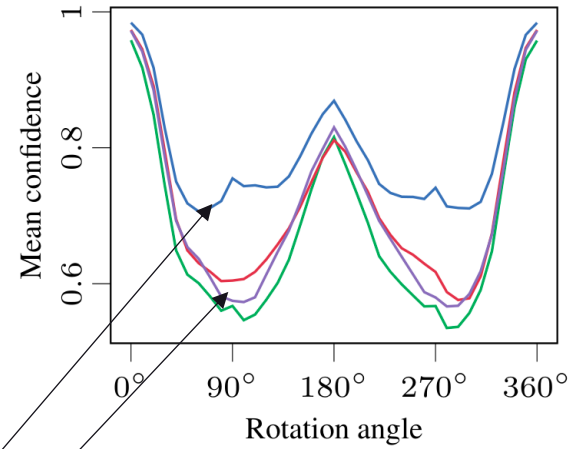
Baumann, A. et al. (2025). Post-hoc Probabilistic Vision-Language Models. ArXiv.

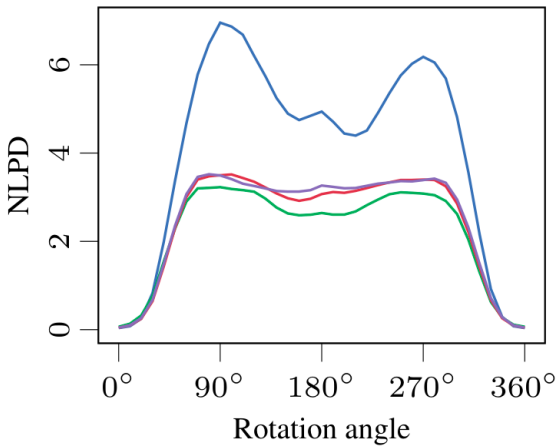
Bayesian Deep Learning



Bayesian Deep Learning



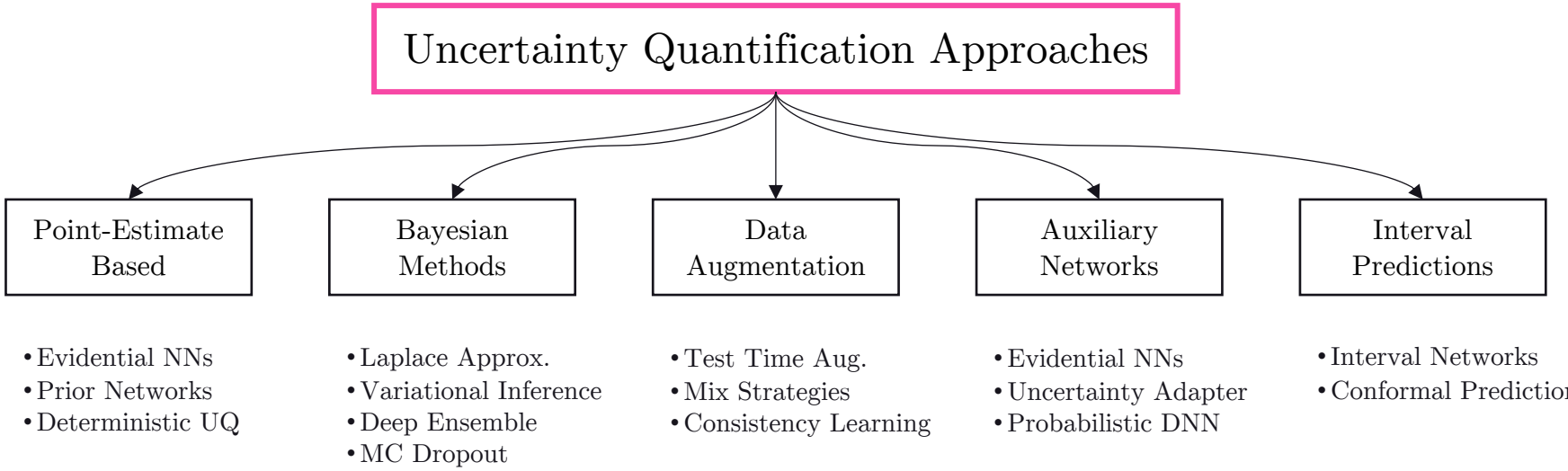




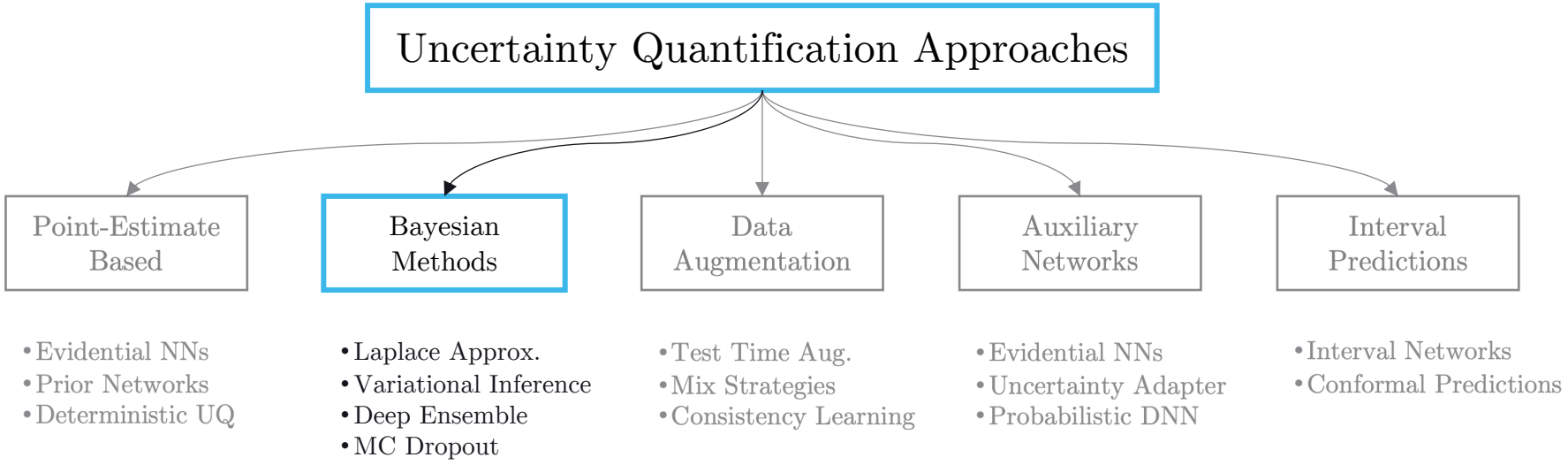
Neural network

Bayesian neural network

Uncertainty Quantification in ML



Uncertainty Quantification in ML



Uncertainty Quantification in ML

TorchUncertainty

Laplace

SUQ.py

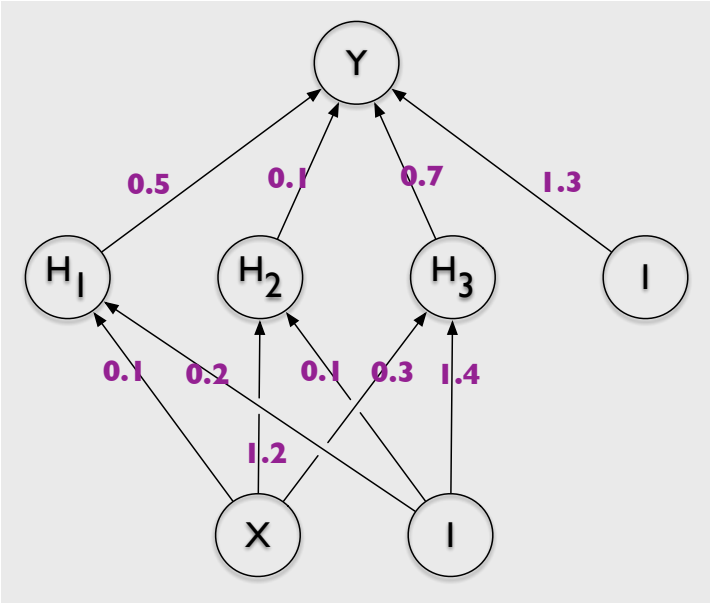
https://torch-uncertainty.github.io/

https://aleximmer.com/Laplace/

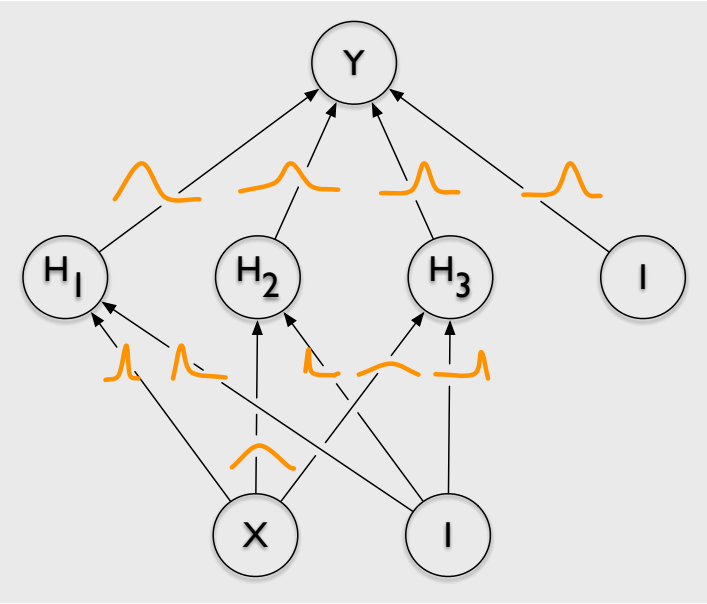
https://github.com/AaltoML/SUQ

Bayesian Deep Learning

(Deterministic) Neural Network



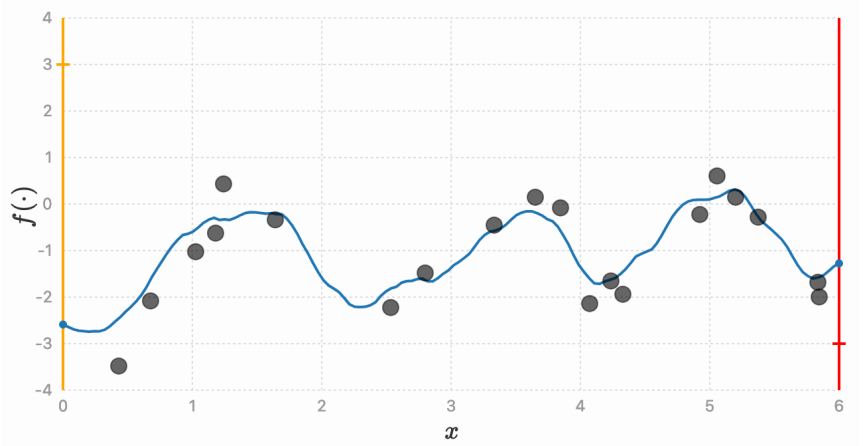
Bayesian Neural Network



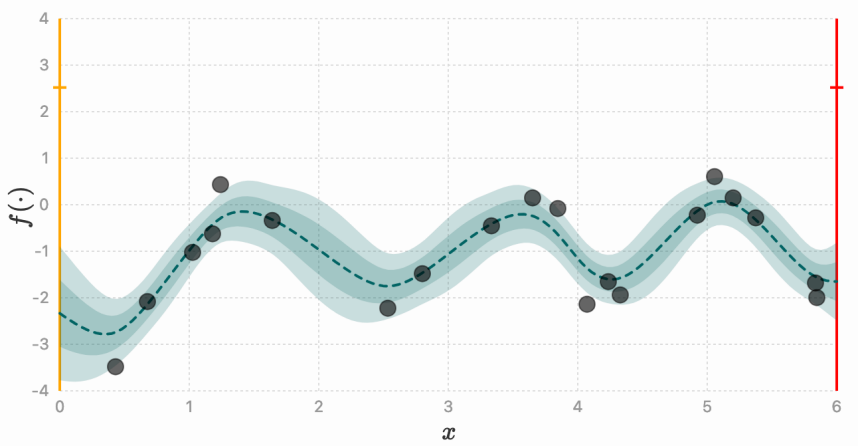
Blundell, C. et al. (2015). Weight Uncertainty in Neural Networks, In *ICML*.

Bayesian Deep Learning

(Deterministic) Neural Network



Bayesian Neural Network



Interactive Gaussian Process Visualization by Ti John (http://www.infinitecuriosity.org/vizgp/)

Bayesian Deep Learning

Computations in Bayesian Learning are notoriously hard! *in general.

Challenges in Bayesian Deep Learning:

- Hard to specify functional priors.
- Intractable high-dimensional multi-modal posterior.
  (billions of parameters)
- Models can be difficult to train from scratch.
  (high compute resource demand)

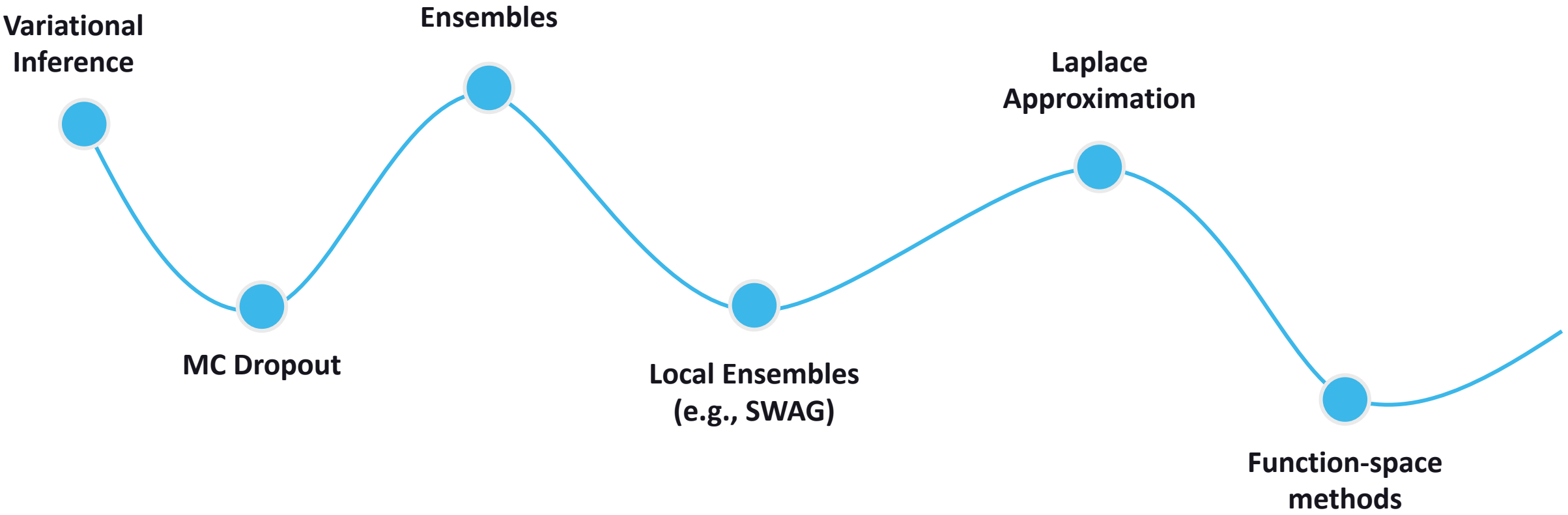
Bayesian Deep Learning

Computations in Bayesian Learning are notoriously hard! *in general.

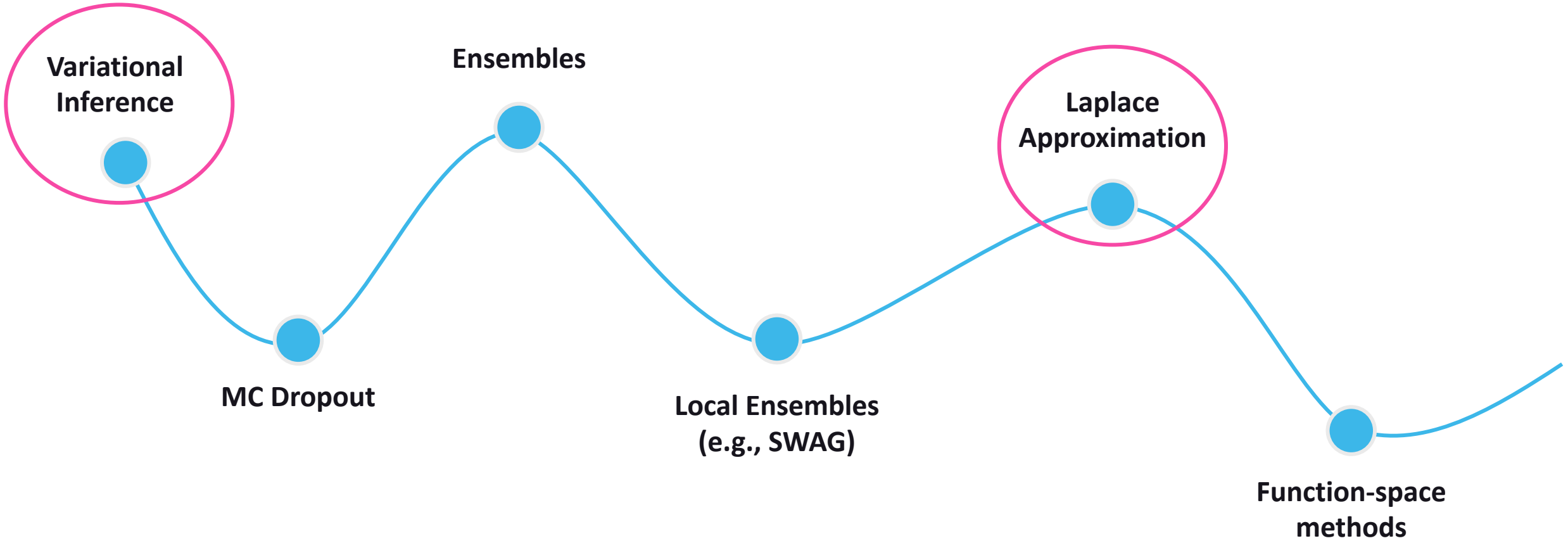
Challenges in Bayesian Deep Learning:

- Hard to specify functional priors.
- Intractable high-dimensional multi-modal posterior.
  (billions of parameters)
- Models can be difficult to train from scratch.
  (high compute resource demand)

Inference Methods in BDL



Inference Methods in BDL



Variational Inference: Big Picture

Recipe for approximating an intractable distribution $p \in \mathcal{P}$

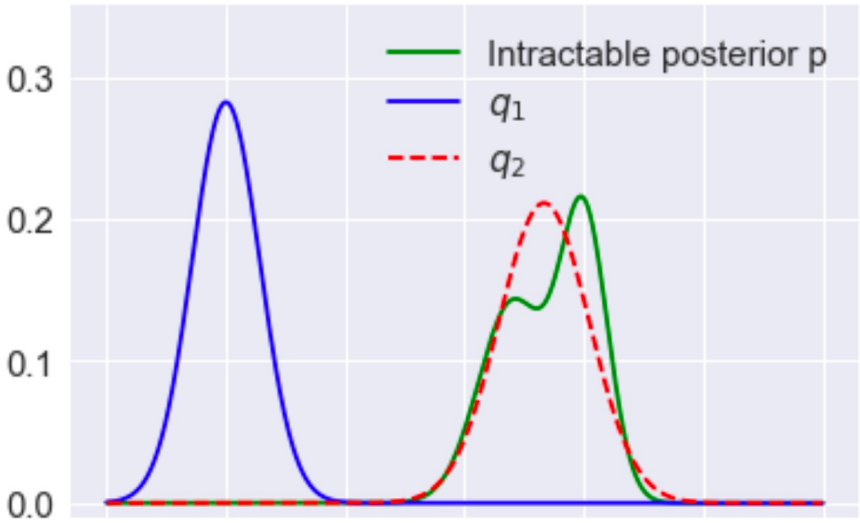
1. Define a tractable family of distributions $\mathcal{Q}$

Variational Inference: Big Picture

Recipe for approximating an intractable distribution $p \in \mathcal{P}$

1. Define a tractable family of distributions $\mathcal{Q}$
2. Define a way to compute a "distance" between distributions

$$\mathrm{D}\left(p||q_1\right) > \mathrm{D}\left(p||q_2\right)$$



Variational Inference: Big Picture

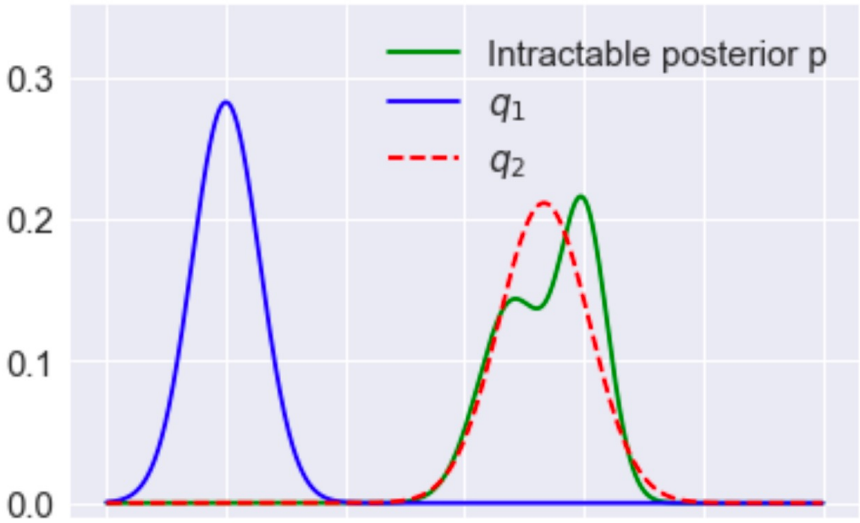
Recipe for approximating an intractable distribution $p \in \mathcal{P}$

1. Define a tractable family of distributions $\mathcal{Q}$
2. Define a way to compute a "distance" between distributions

$$\mathrm{D}\left(p||q_1\right) > \mathrm{D}\left(p||q_2\right)$$

3. Search for best approximation

$$q^\star = \arg\min_{q \in \mathcal{Q}} \mathrm{D}\left(p||q\right)$$



Reminder on KL divergence

Let P and Q be two probability distributions with p.d.f. denoted as p and q, respectively.

Then their Kullback–Leibler divergence is given as:

$$\mathrm{D}_{\mathrm{KL}}\left(P||Q\right) = \int_x p(x) \log\left(\frac{p(x)}{q(x)}\right) \mathrm{d}x$$

Reminder on KL divergence

Let P and Q be two probability distributions with p.d.f. denoted as p and q, respectively.

Then their Kullback–Leibler divergence is given as:

$$\mathrm{D}_{\mathrm{KL}}\left(P||Q\right) = \int_x p(x) \log\left(\frac{p(x)}{q(x)}\right) \mathrm{d}x$$

$$\mathrm{D}_{\mathrm{KL}}\left(P||Q\right) \geq 0 \qquad \text{and} \qquad \mathrm{D}_{\mathrm{KL}}\left(P||Q\right) = 0 \iff P = Q$$

Reminder on ELBO

Reverse KL

$$\mathrm{D}_{\mathrm{KL}}\left(q(\theta)||p(\theta \mid \boldsymbol{x}_{1:n})\right) = \mathbb{E}_{\theta\sim q}\left[\log\left(\frac{q(\theta)}{p(\theta \mid \boldsymbol{x}_{1:n})}\right)\right]$$

Reminder on ELBO

$$\begin{aligned}
\mathrm{D}_{\mathrm{KL}}\left(q(\theta)||p(\theta \mid \boldsymbol{x}_{1:n})\right) &= \mathbb{E}_{\theta\sim q}\left[\log\left(\frac{q(\theta)}{p(\theta \mid \boldsymbol{x}_{1:n})}\right)\right] \\
&= \mathbb{E}_{\theta\sim q}\left[\log\left(\frac{q(\theta)}{\frac{p(\boldsymbol{x}_{1:n}|\theta)\,p(\theta)}{p(\boldsymbol{x}_{1:n})}}\right)\right]
\end{aligned}$$

Reminder on ELBO

$$
\begin{aligned}
\mathrm{D}_{\mathrm{KL}}\left(q(\theta)||p(\theta \mid \boldsymbol{x}_{1:n})\right) &= \mathbb{E}_{\theta\sim q}\left[\log\left(\frac{q(\theta)}{p(\theta \mid \boldsymbol{x}_{1:n})}\right)\right] \\
&= \mathbb{E}_{\theta\sim q}\left[\log\left(\frac{q(\theta)}{\frac{p(\boldsymbol{x}_{1:n}|\theta)\,p(\theta)}{p(\boldsymbol{x}_{1:n})}}\right)\right] \\
&= \mathbb{E}_{\theta\sim q}\left[\log\left(\frac{q(\theta)}{p(\theta)}\right) + \log\left(\frac{p(\boldsymbol{x}_{1:n})}{p(\boldsymbol{x}_{1:n} \mid \theta)}\right)\right]
\end{aligned}
$$

Reminder on ELBO

$$
\begin{aligned}
\mathrm{D}_{\mathrm{KL}}\left(q(\theta)||p(\theta \mid \boldsymbol{x}_{1:n})\right) &= \mathbb{E}_{\theta\sim q}\left[\log\left(\frac{q(\theta)}{p(\theta \mid \boldsymbol{x}_{1:n})}\right)\right] \\
&= \mathbb{E}_{\theta\sim q}\left[\log\left(\frac{q(\theta)}{\frac{p(\boldsymbol{x}_{1:n}|\theta)\,p(\theta)}{p(\boldsymbol{x}_{1:n})}}\right)\right] \\
&= \mathbb{E}_{\theta\sim q}\left[\log\left(\frac{q(\theta)}{p(\theta)}\right) + \log\left(\frac{p(\boldsymbol{x}_{1:n})}{p(\boldsymbol{x}_{1:n} \mid \theta)}\right)\right] \\
&= \mathrm{D}_{\mathrm{KL}}\left(q(\theta)||p(\theta)\right) - \mathbb{E}_{\theta\sim q}\left[\log p(\boldsymbol{x}_{1:n} \mid \theta)\right] + \log p(\boldsymbol{x}_{1:n})
\end{aligned}
$$

Reminder on ELBO

$$\mathrm{D}_{\mathrm{KL}}\left(q(\theta)||p(\theta \mid \boldsymbol{x}_{1:n})\right) = \mathbb{E}_{\theta\sim q}\left[\log\left(\frac{q(\theta)}{p(\theta \mid \boldsymbol{x}_{1:n})}\right)\right]$$

$$\vdots$$

$$= \mathrm{D}_{\mathrm{KL}}\left(q(\theta)||p(\theta)\right) - \mathbb{E}_{\theta\sim q}\left[\log p(\boldsymbol{x}_{1:n} \mid \theta)\right] + \log p(\boldsymbol{x}_{1:n})$$

Reminder on ELBO

$$\mathrm{D}_{\mathrm{KL}}\left(q(\theta)||p(\theta \mid \boldsymbol{x}_{1:n})\right) = \mathbb{E}_{\theta\sim q}\left[\log\left(\frac{q(\theta)}{p(\theta \mid \boldsymbol{x}_{1:n})}\right)\right]$$

$$\vdots$$

$$= \mathrm{D}_{\mathrm{KL}}\left(q(\theta)||p(\theta)\right) - \mathbb{E}_{\theta\sim q}\left[\log p(\boldsymbol{x}_{1:n} \mid \theta)\right] + \log p(\boldsymbol{x}_{1:n})$$

$$\log p(\boldsymbol{x}_{1:n}) = -\mathrm{D}_{\mathrm{KL}}\left(q(\theta)||p(\theta)\right) + \mathbb{E}_{\theta\sim q}\left[\log p(\boldsymbol{x}_{1:n} \mid \theta)\right] + \underbrace{\mathrm{D}_{\mathrm{KL}}\left(q(\theta)||p(\theta \mid \boldsymbol{x}_{1:n})\right)}_{>0}$$

Reminder on ELBO

$$\mathrm{D}_{\mathrm{KL}}\left(q(\theta)||p(\theta \mid \boldsymbol{x}_{1:n})\right) = \mathbb{E}_{\theta\sim q}\left[\log\left(\frac{q(\theta)}{p(\theta \mid \boldsymbol{x}_{1:n})}\right)\right]$$

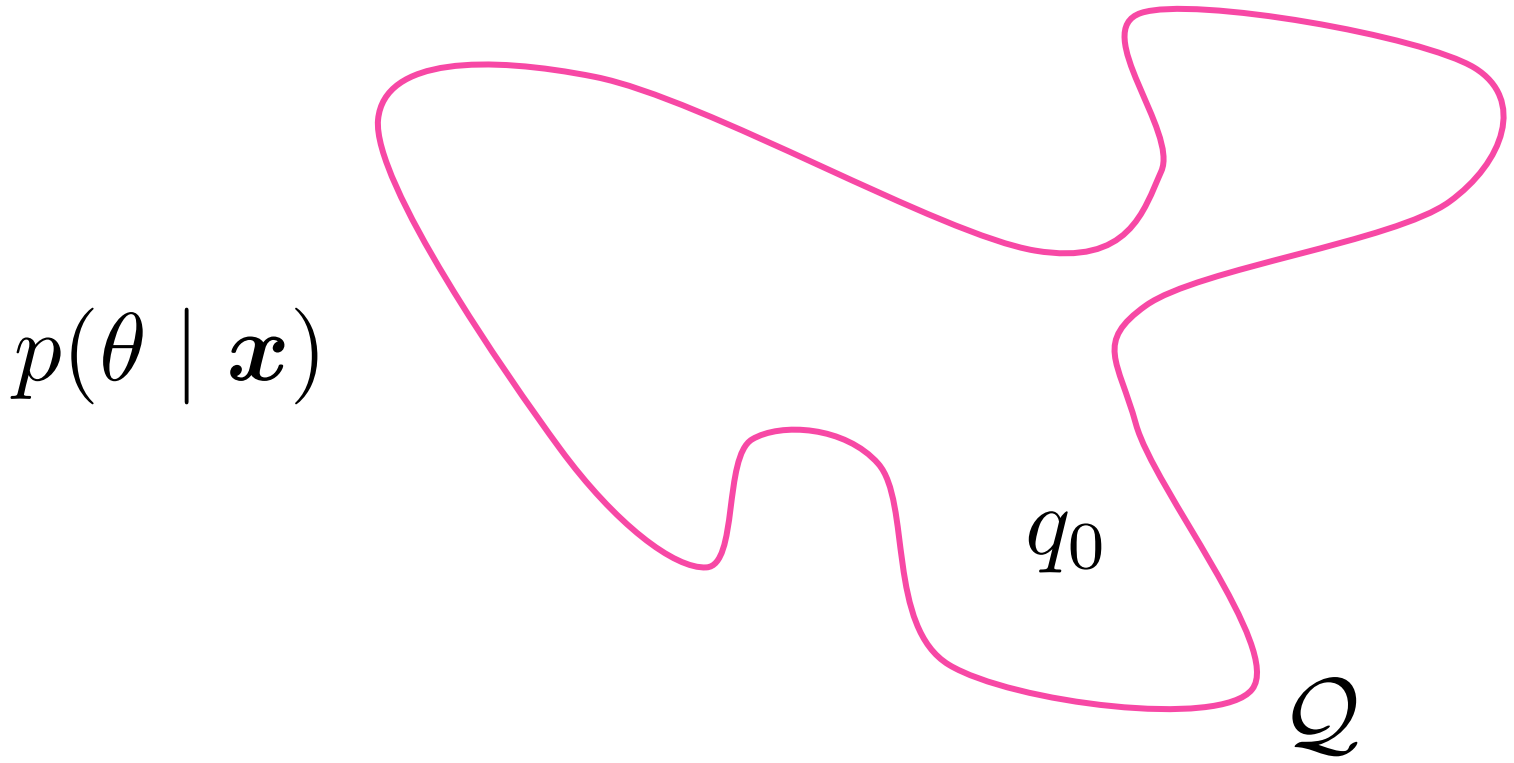
$$\vdots$$

$$= \mathrm{D}_{\mathrm{KL}}\left(q(\theta)||p(\theta)\right) - \mathbb{E}_{\theta\sim q}\left[\log p(\boldsymbol{x}_{1:n} \mid \theta)\right] + \log p(\boldsymbol{x}_{1:n})$$

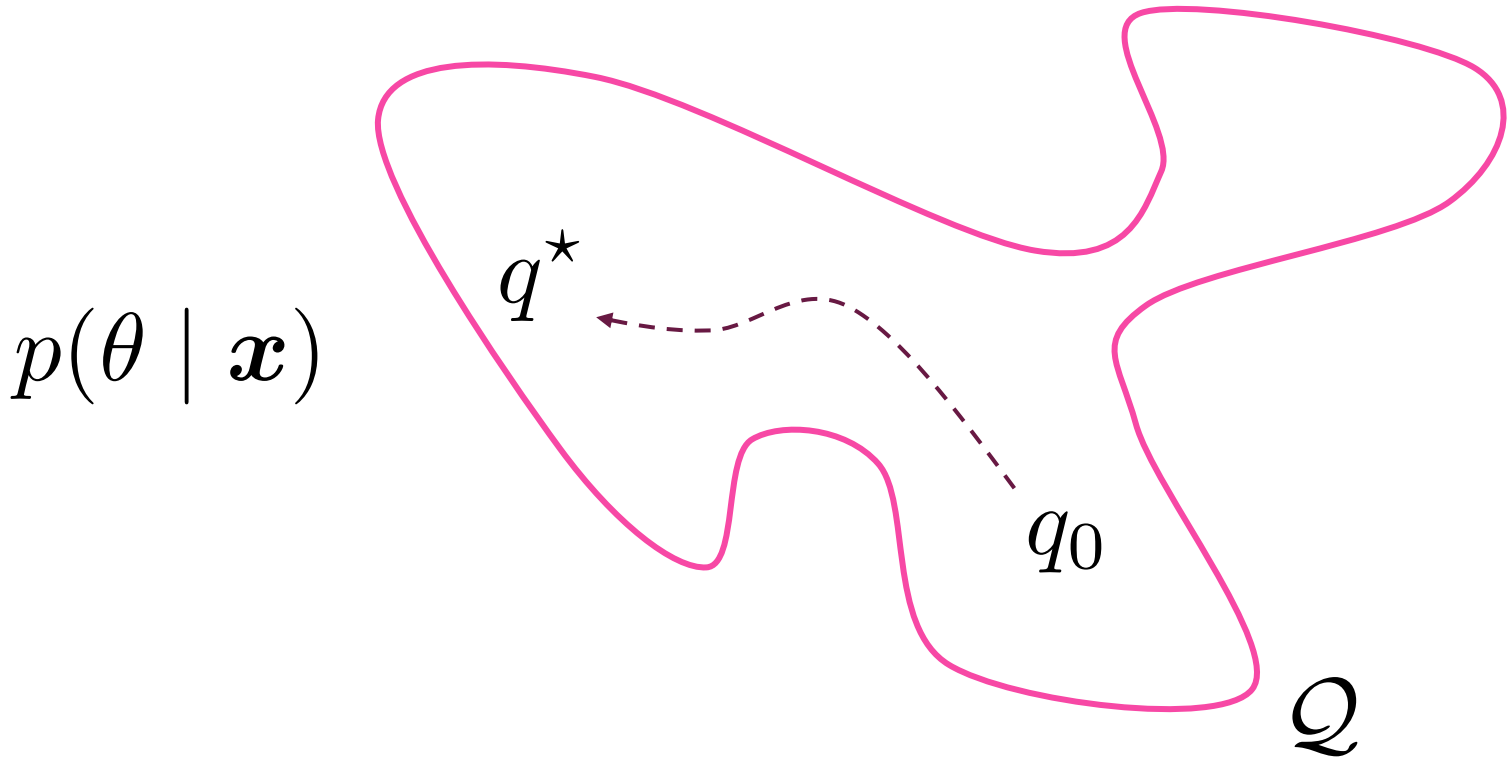
$$\log p(\boldsymbol{x}_{1:n}) = -\mathrm{D}_{\mathrm{KL}}\left(q(\theta)||p(\theta)\right) + \mathbb{E}_{\theta\sim q}\left[\log p(\boldsymbol{x}_{1:n} \mid \theta)\right] + \underbrace{\mathrm{D}_{\mathrm{KL}}\left(q(\theta)||p(\theta \mid \boldsymbol{x}_{1:n})\right)}_{\geq 0}$$

$$\geq -\mathrm{D}_{\mathrm{KL}}\left(q(\theta)||p(\theta)\right) + \mathbb{E}_{\theta\sim q}\left[\log p(\boldsymbol{x}_{1:n} \mid \theta)\right] = \mathrm{ELBO}\ \text{(Evidence Lower Bound)}$$

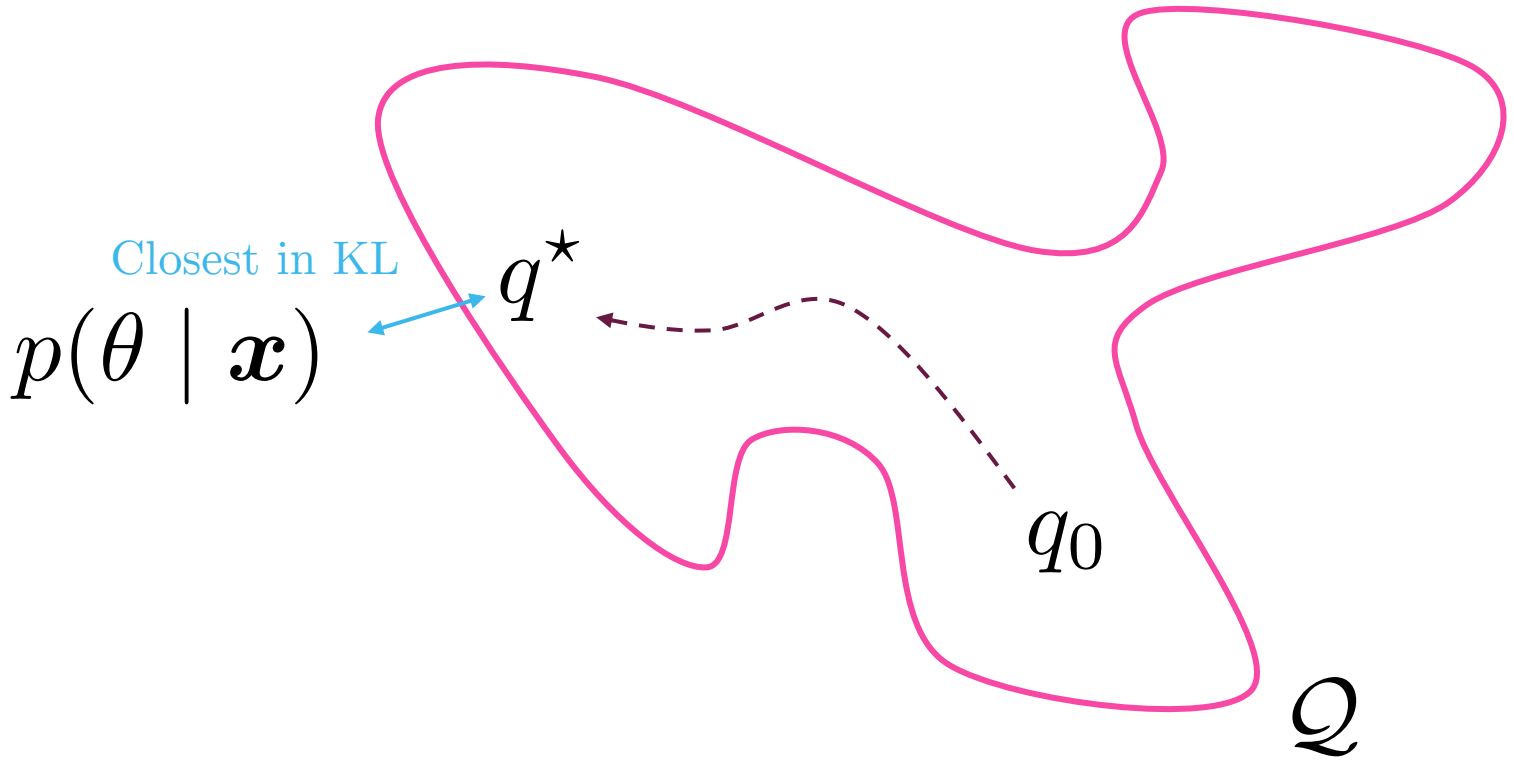
Variational Inference



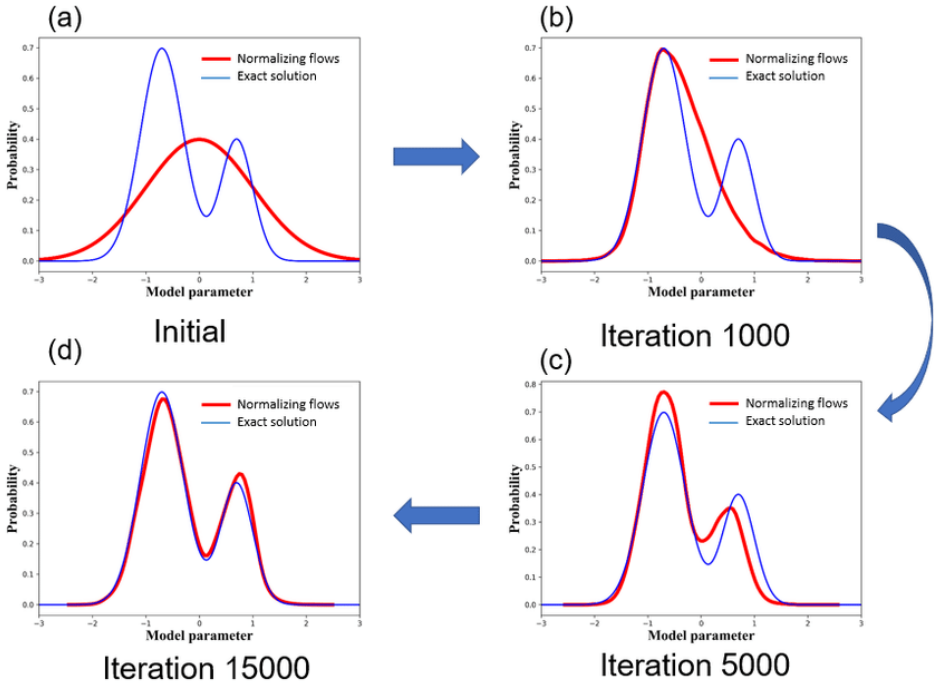
Variational Inference



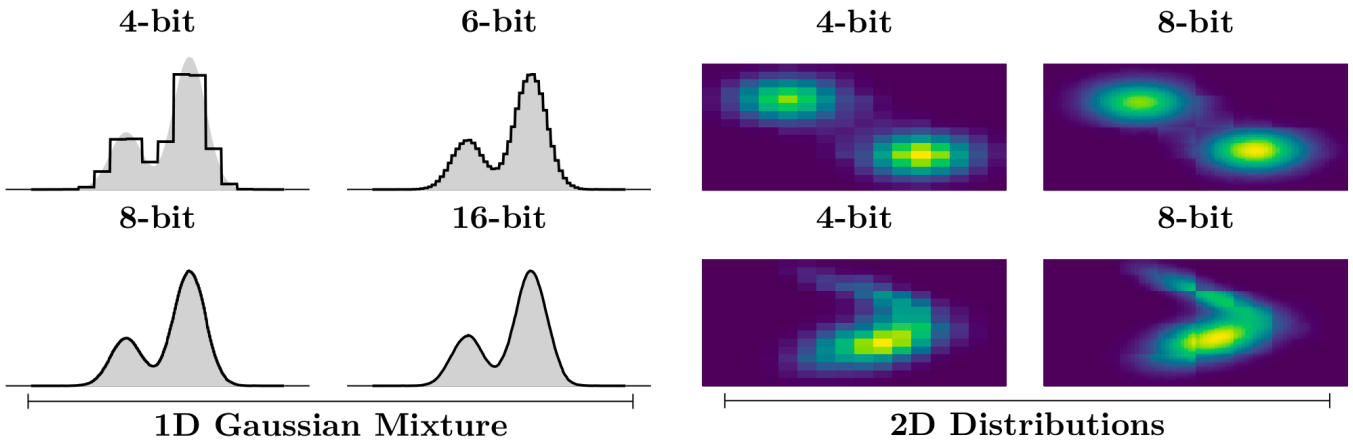
Variational Inference



Variational Inference

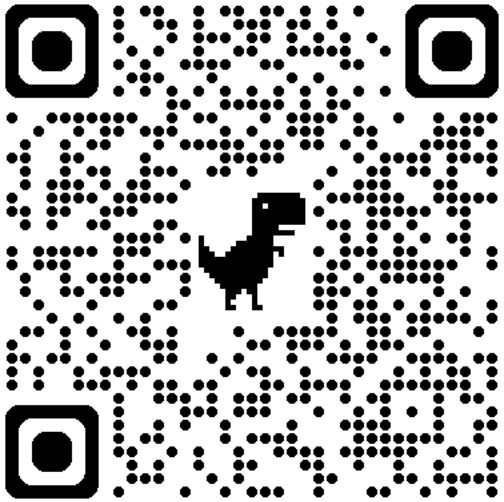


Rezende, D. et al. (2015). Variational Inference with Normalizing Flows. In *ICML*.



Sladek, A. et al. (2025). Approximate Bayesian Inference via Bitstring Representations. In *UAI*.

Variational Inference



https://lacerbi.github.io/blog/2024/vi-is-inference-is-optimization/

Variational Inference: Takeaways

- VI turns inference into an optimisation problem.
- Can scale to large data sets, but is challenging to optimise
- Can exhibit high variance in stochastic gradients
- Tendency to underestimate the variance if reverse KL is used

- Recent work has shown that VI can be effective for large-scale models.

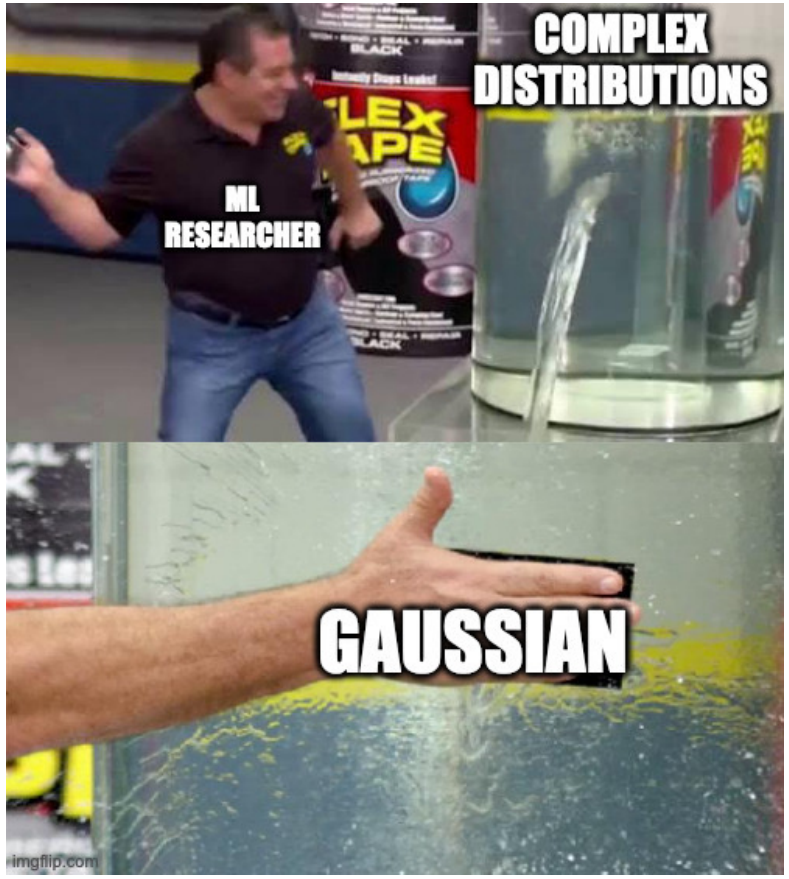
Shen, Y. et al (2024). Variational Learning is Effective for Large Deep Networks. In *ICML*.

Laplace Approximation

What if training a model "from scratch" is too expensive?

Laplace Approximation

What if training a model "from scratch" is too expensive?



Laplace Approximation

$$\log p(\theta \mid \boldsymbol{x}) = \overbrace{\underbrace{\log p(\theta)}_{\text{L2 regularization}} + \underbrace{\log p(\boldsymbol{x} \mid \theta)}_{\text{Cross-Entropy}}}^{\text{Negated Loss}} - \underbrace{C}_{\text{Unknown}}$$

Laplace Approximation

$$\log p(\theta \mid \boldsymbol{x}) = \overbrace{\underbrace{\log p(\theta)}_{\text{L2 regularization}} + \underbrace{\log p(\boldsymbol{x} \mid \theta)}_{\text{Cross-Entropy}}}^{\text{Negated Loss}} - \underbrace{C}_{\text{Unknown}}$$

$$\underbrace{\log p(\theta) + \log p(\boldsymbol{x} \mid \theta)}_{\ell(\theta)} \approx \ell(\theta^*) + \boldsymbol{J}_{\ell|_{\theta=\theta^*}} (\theta - \theta^*) - \frac{1}{2} \boldsymbol{H}_{\ell|_{\theta=\theta^*}} (\theta - \theta^*)^2$$

Laplace Approximation

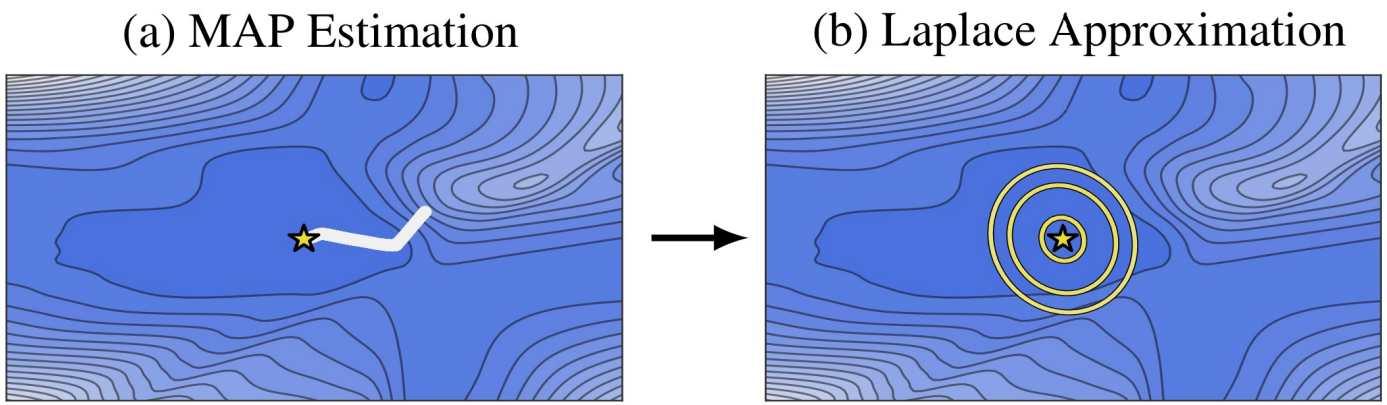
$$\log p(\theta \mid \boldsymbol{x}) = \overbrace{\underbrace{\log p(\theta)}_{\text{L2 regularization}} + \underbrace{\log p(\boldsymbol{x} \mid \theta)}_{\text{Cross-Entropy}}}^{\text{Negated Loss}} - \underbrace{C}_{\text{Unknown}}$$

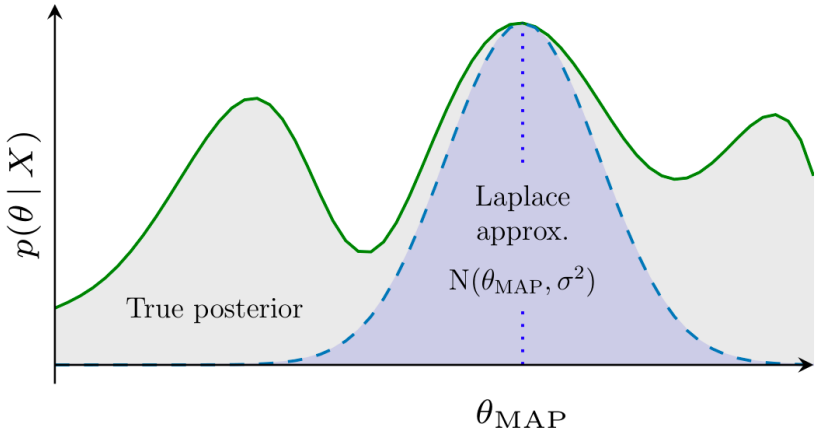
$$\underbrace{\log p(\theta) + \log p(\boldsymbol{x} \mid \theta)}_{\ell(\theta)} \approx \ell(\theta^*) + \boldsymbol{J}_{\ell|_{\theta=\theta^*}} (\theta - \theta^*) - \frac{1}{2} \boldsymbol{H}_{\ell|_{\theta=\theta^*}} (\theta - \theta^*)^2$$

$$= \ell(\theta^*) - \frac{1}{2} \boldsymbol{H}_{\ell|_{\theta=\theta^*}} (\theta - \theta^*)^2$$

$$\propto \log \mathcal{N}(\theta^*, \boldsymbol{H}^{-1}_{\ell|_{\theta=\theta^*}})$$

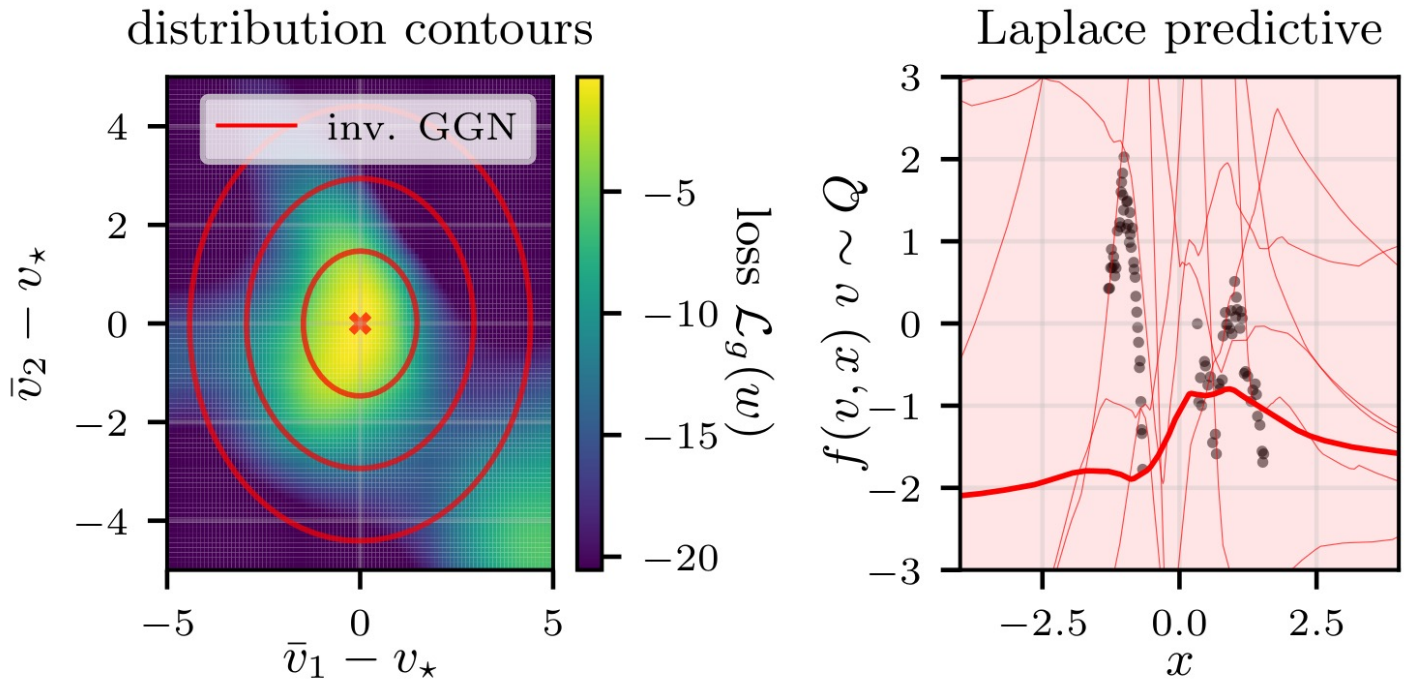
Laplace Approximation





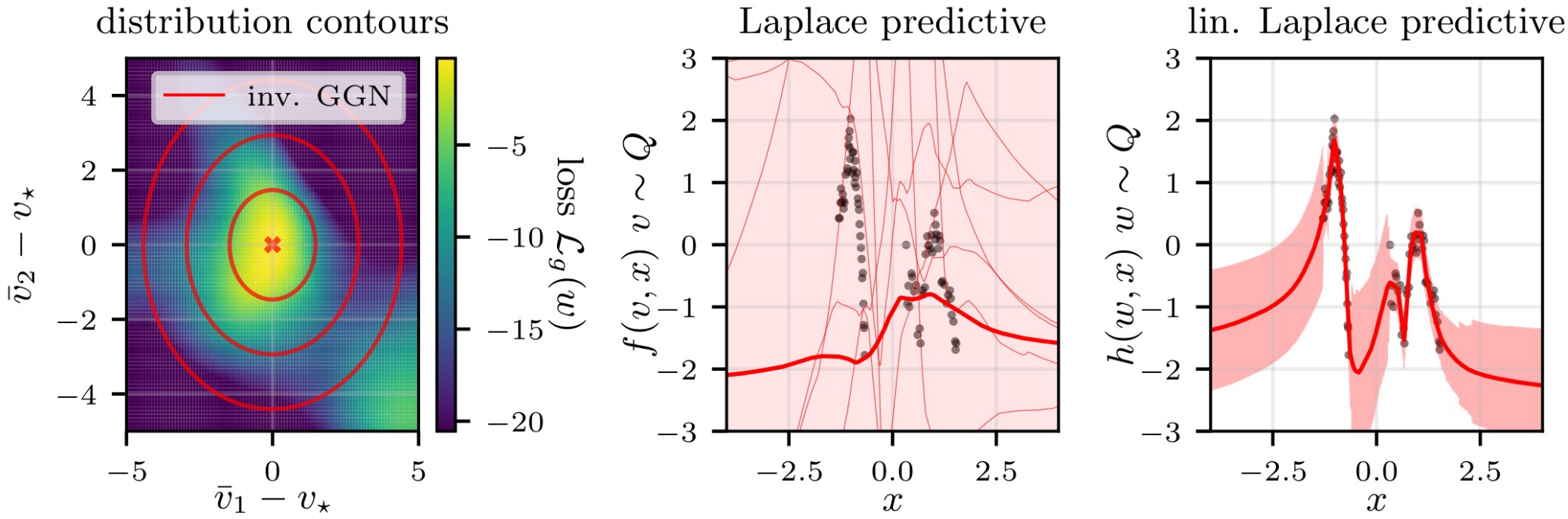
Daxberger, E. et al. (2021) Laplace Redux - Effortless Bayesian Deep Learning. In *NeurIPS*.
Roy, S. et al. (2022). Uncertainty-guided Source-free Domain Adaptation. In *ECCV*.

Linearised Laplace: Problem Setting



Antoran, J. (2024). Scalable Bayesian Inference in the Era of Deep Learning: From Gaussian Processes to Deep Neural Networks. PhD thesis.

Linearised Laplace: Problem Setting



Antoran, J. (2024). Scalable Bayesian Inference in the Era of Deep Learning: From Gaussian Processes to Deep Neural Networks. PhD thesis.

Linearised Laplace

The linearised Laplace approximates the predictive distribution:

$$f(x;\theta) \approx f(x;\theta_{\text{MAP}}) + \nabla f(x;\theta)\,|_{\theta_{\text{MAP}}}^{\top}\,(\theta - \theta_{\text{MAP}}) = \hat{f}(x;\theta)$$

Linearised Laplace

The linearised Laplace approximates the predictive distribution:

$$f(x;\theta) \approx f(x;\theta_{\text{MAP}}) + \nabla f(x;\theta)\,|_{\theta_{\text{MAP}}}^{\top}\,(\theta - \theta_{\text{MAP}}) = \hat{f}(x;\theta)$$

$$p(y^\star \mid x^\star, \boldsymbol{x}, \boldsymbol{y}) \approx \int \mathcal{N}(y^\star \mid \hat{f}(x;\theta), \sigma)\,\mathcal{N}(\theta \mid \theta_{\text{MAP}}, \Sigma)\mathrm{d}\theta$$

Laplace Approximation: Takeaways

- Does not require retraining (post-hoc).
- Requires linearisation in the case of deep learning models.
- Crude and only local approximation.
- Estimation of the Hessian can be difficult.

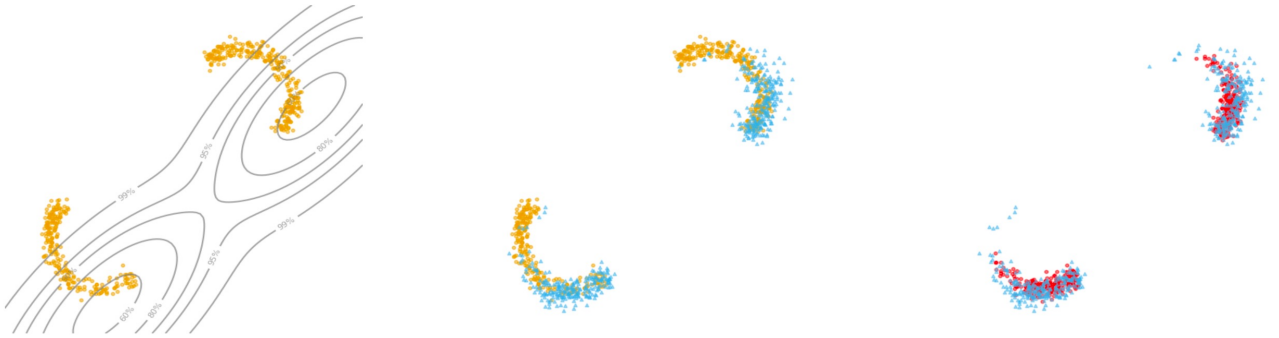
- Crude but useful (easy to use) "tool" for Bayesian inference in deep learning.

Machine Learning for Bayes

Amortized Inference

- NNs are powerful function approximators and generators (e.g., diffusion models)
- How can we use NNs for approximate Bayesian inference?

- → Emerging research on amortized inference with NNs (condition and predict)



(a) Prior v samples (b) PriorGuide v samples (c) PriorGuide v retrained

Chang, P. et al. (2025). Inference-Time Prior Adaptation in Simulation-Based Inference via Guided Diffusion Models. ICLR workshop.

Recap

- Deep learning methods can be problematic in high-risk domains.
- The Bayesian approach to deep learning can help reduce overconfidence, quantify uncertainties, and utilise uncertainties in decision-making.
- Prior specification is challenging and an open question.
- Performing computations is challenging, but promising avenues exist.
- Machine learning can help scaling Bayesian inference, e.g., through amortisation.

Thanks for your attention!