Functional Priors for Bayesian Deep Learning

Maurizio Filippone
Statistics Program, KAUST
August 8^{th}, 2024

Decision-Making

Decision-making is a critical step in several domains [**Norvig and Russell**, **1995**]:

- Policy-making for the environment
- Healthcare
- Society
- ...

Decision-Making

Decision-making is a critical step in several domains [**Norvig and Russell**, **1995**]:

- Policy-making for the environment
- Healthcare
- Society
- ...

$$\text{Decision Theory} = \text{Probabilistic reasoning} + \text{Utility theory}$$

Over-confidence of Deep Learning Models - Online Meme

Image prediction: ping-pong ball
Confidence: 99.99%

Illustration: Dianna “Mick” McDougall, Photo: ResNeXtGuesser

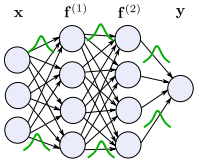
Over-confidence of Deep Learning Models - Online Meme

Image prediction: pineapple
Confidence: 99.3%

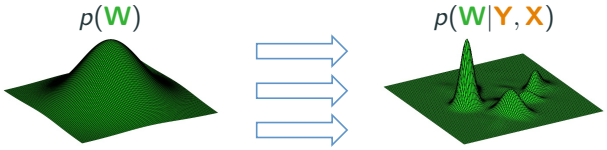
Illustration: Dianna “Mick” McDougall, Photo: ResNeXtGuesser

Bayesian Deep Nets

- Inputs : $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$
- Labels : $\mathbf{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$
- Weights : $\mathbf{W} = \{\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(L)}\}$



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

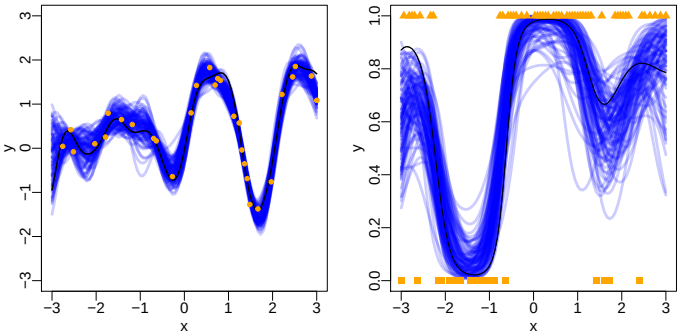


$$p(\mathbf{W}|\mathbf{Y}, \mathbf{X}) = \frac{p(\mathbf{Y}|\mathbf{X}, \mathbf{W})p(\mathbf{W})}{\int p(\mathbf{Y}|\mathbf{X}, \mathbf{W})p(\mathbf{W})d\mathbf{W}}$$

Bayesian Deep Nets

- Predictions consider an infinite number of parameter configurations

$$p(\mathbf{y}^*|\mathbf{x}^*, \mathbf{Y}, \mathbf{X}) = \int p(\mathbf{y}^*|\mathbf{x}^*, \mathbf{W}) p(\mathbf{W}|\mathbf{Y}, \mathbf{X}) d\mathbf{W}$$



Challenges with Bayesian Deep Learning

- Bayesian approaches are **usually slow** compared to non-Bayesian ones

Challenges with Bayesian Deep Learning

- Bayesian approaches are **usually slow** compared to non-Bayesian ones
  - Partial stochasticity
    [**Sharma et al.**, **AISTATS 2023**]
  - Inducing-points approximations
    [**Ritter et al.**, **NeurIPS 2021**]
  - Bayesian compression
    [**Louizos et al.**, **NeurIPS 2017**]

Challenges with Bayesian Deep Learning

- Predictive **performance is usually worse** than non-Bayesian solutions (e.g., [**Wenzel et al.**, **ICML 2020**])

Challenges with Bayesian Deep Learning

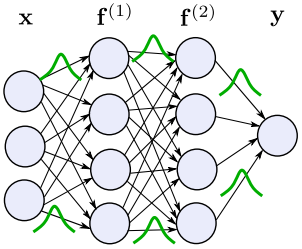
- Predictive **performance is usually worse** than non-Bayesian solutions (e.g., [**Wenzel et al.**, **ICML 2020**])
  - Ensembles
    [**Gal and Ghahramani**, **ICML 2016** – **Lakshminarayanan et al.**, **NeurIPS 2017**]
  - Better priors
    [**Tran et al.**, **JMLR 2022** – **Fortuin et al.**, **ICLR 2022**]
  - Improvements to Variational Inference for deep models
    [**Rossi et al.**, **ICML 2019**, **NeurIPS 2020**]

Challenges with Bayesian Deep Learning

- Predictive **performance is usually worse** than non-Bayesian solutions (e.g., [**Wenzel et al.**, **ICML 2020**])
  - Ensembles
    [**Gal and Ghahramani**, **ICML 2016** – **Lakshminarayanan et al.**, **NeurIPS 2017**]
  - Better priors
    [**Tran et al.**, **JMLR 2022** – **Fortuin et al.**, **ICLR 2022**]
  - Improvements to Variational Inference for deep models
    [**Rossi et al.**, **ICML 2019**, **NeurIPS 2020**]

- In this talk we focus on choosing **sensible priors**

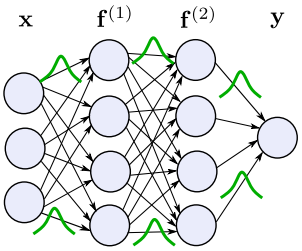
Prior for Bayesian Neural Networks



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

Specifying a sensible prior for Bayesian neural networks (BNNs) is difficult!

Prior for Bayesian Neural Networks

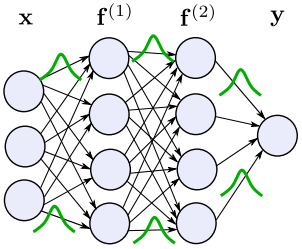


$$\mathbf{w}_i \sim \mathcal{N}(0, \mathbf{I})$$

Specifying a sensible prior for Bayesian neural networks (BNNs) is difficult!

- Neural networks are extremely **high-dimensional** and **nonidentifiable**.

Prior for Bayesian Neural Networks



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

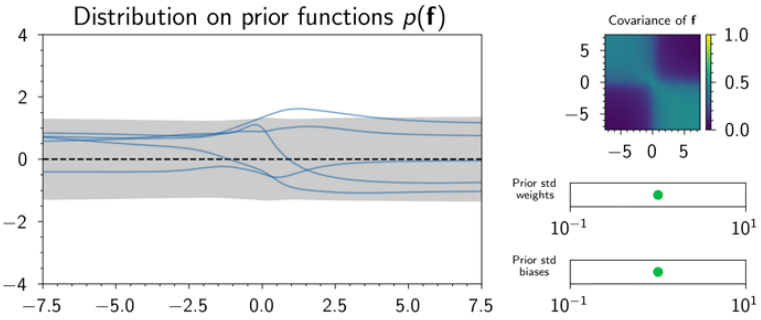
Specifying a sensible prior for Bayesian neural networks (BNNs) is difficult!

- Neural networks are extremely **high-dimensional** and **nonidentifiable**.
- Most work resorts to simple priors $\mathcal{N}(0, 1)$ and $\mathcal{N}(0, 1/D_{l-1})$.

Prior for Bayesian Neural Networks

The prior on the parameters of a Bayesian neural network (bnn) induces an *unpredictable prior over functions*.

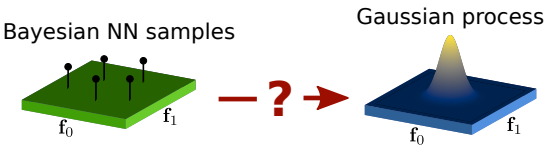
$$p(\mathbf{f}) = \int p(\mathbf{f} \mid \mathbf{w}) p(\mathbf{w}) d\mathbf{w}$$



Choosing Priors

We developed a novel framework to impose functional priors on BNNs

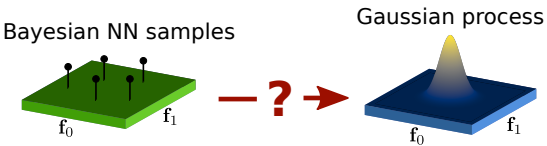
- In supervised learning, we match a functional prior $p(\mathbf{f})$ of choice
  - We focused on Gaussian processes for their simplicity and interpretability



Choosing Priors

We developed a novel framework to impose functional priors on BNNs

- In supervised learning, we match a functional prior $p(\mathbf{f})$ of choice
  - We focused on Gaussian processes for their simplicity and interpretability

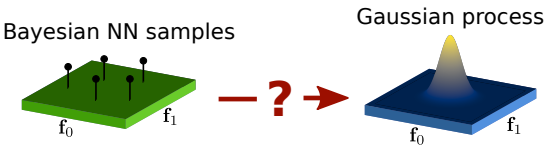


- In unsupervised learning, the functional prior is $p(\mathbf{x})$
  - Data points are samples from $p(\mathbf{x})$!
  - Prior selection can be cast as model selection

Choosing Priors

We developed a novel framework to impose functional priors on BNNs

- In supervised learning, we match a functional prior $p(\mathbf{f})$ of choice
  - We focused on Gaussian processes for their simplicity and interpretability



- In unsupervised learning, the functional prior is $p(\mathbf{x})$
  - Data points are samples from $p(\mathbf{x})$!
  - Prior selection can be cast as model selection

Matching based on Wasserstein distance

Proposed Method

- Minimize the 1-Wasserstein distance between BNN functional prior and a given functional prior
- Given a measurable space Ω, the Kantorovich dual form of the 1-Wasserstein distance between two Borel's probability measures π and ν in $\mathcal{P}(\Omega)$ is

$$W_1(\pi, \nu) = \sup_{\|\phi\|_L \leq 1} \mathbb{E}_\pi[\phi(\mathbf{x})] - \mathbb{E}_\nu[\phi(\mathbf{x})]\,,$$

  where ϕ is a 1-Lipschitz function.

✓ The objective is *fully sampled-based*!

Proposed Method

- Minimize the 1-Wasserstein distance between BNN functional prior and a given functional prior
- Given a measurable space Ω, the Kantorovich dual form of the 1-Wasserstein distance between two Borel's probability measures π and ν in $\mathcal{P}(\Omega)$ is

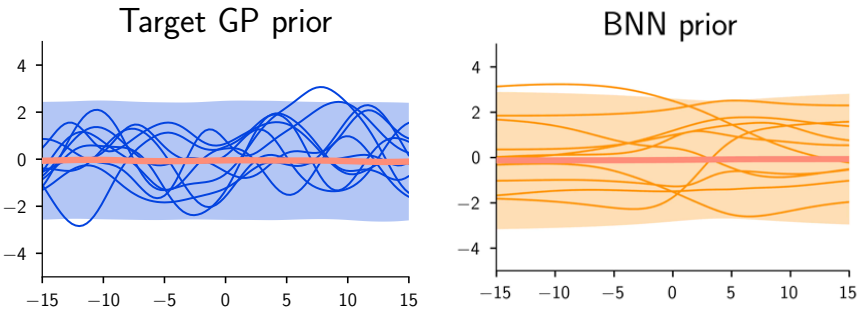
$$W_1(\pi, \nu) = \sup_{\|\phi\|_L \leq 1} \mathbb{E}_\pi[\phi(\mathbf{x})] - \mathbb{E}_\nu[\phi(\mathbf{x})]\,,$$

  where ϕ is a 1-Lipschitz function.

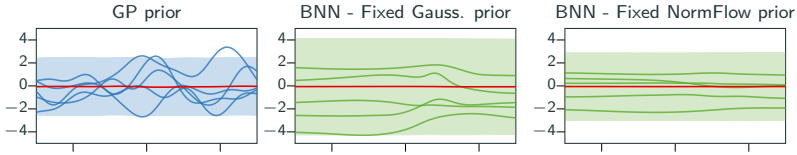
✓ The objective is *fully sampled-based*!

✓ The objective can be optimized with gradient descent algorithms with back-propagation.

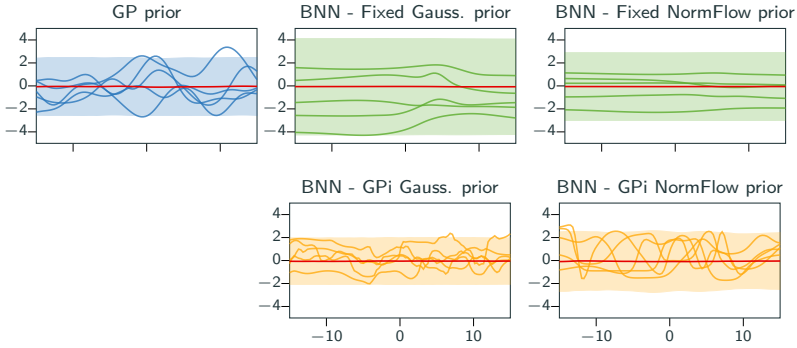
Matching BNN Prior to GP Prior



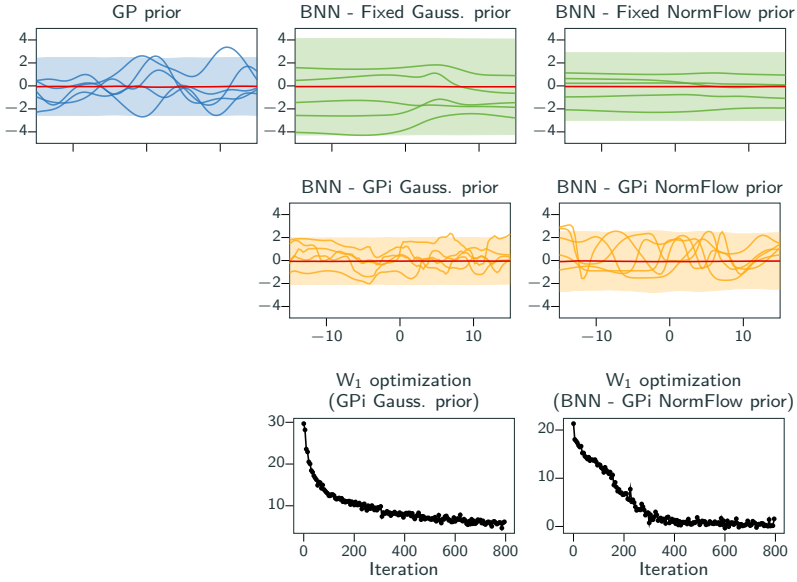
1D Regression Synthetic Data



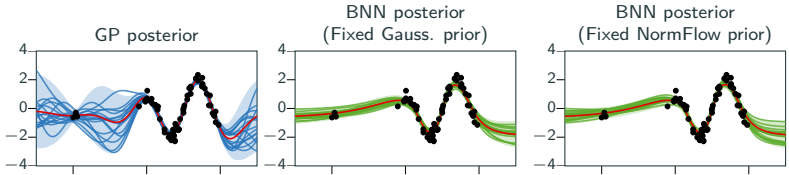
1D Regression Synthetic Data



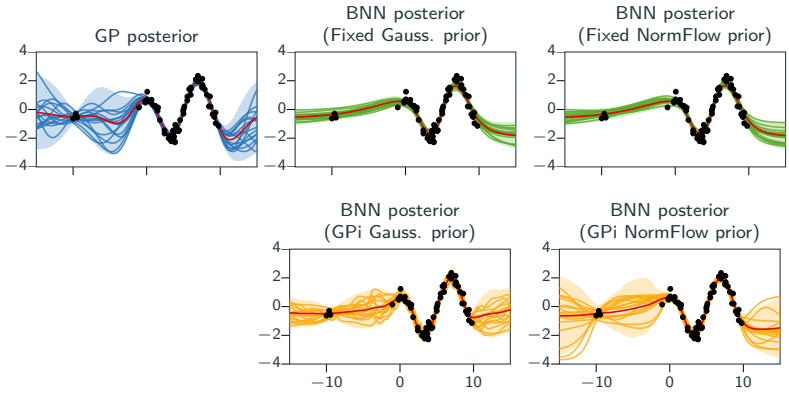
1D Regression Synthetic Data



1D Regression Synthetic Data



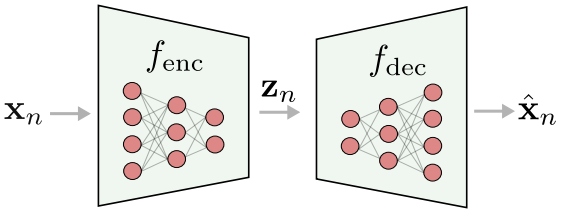
1D Regression Synthetic Data



Bayesian Convolutional Neural Networks - CIFAR-10

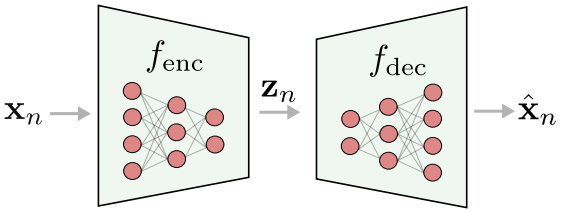
Architecture	Method	Accuracy - % (↑)	NLL (↓)
VGG16	Deep Ensemble	81.96 ± 0.33	0.7759 ± 0.0033
	Fixed Gauss. prior	81.47 ± 0.33	0.5808 ± 0.0033
	Fixed Gauss. prior + Temp. Scaling	82.25 ± 0.15	0.5398 ± 0.0015
	GPi Gauss. prior (**ours**)	83.34 ± 0.53	0.5176 ± 0.0053
	Fixed Hierar. prior	86.03 ± 0.20	0.4345 ± 0.0020
	GPi Hierar. prior (**ours**)	**87.03** ± **0.07**	**0.4127** ± **0.0007**
PRERESNET20	Deep Ensemble	87.77 ± 0.03	0.3927 ± 0.0003
	Fixed Gauss. prior	85.34 ± 0.13	0.4975 ± 0.0013
	Fixed Gauss. prior + Temp. Scaling	87.70 ± 0.11	0.3956 ± 0.0011
	GPi Gauss. prior (**ours**)	86.86 ± 0.27	0.4286 ± 0.0027
	Fixed Hierar. prior	87.26 ± 0.09	0.4086 ± 0.0009
	GPi Hierar. prior (**ours**)	**88.20** ± **0.07**	**0.3808** ± **0.0007**

Autoencoders



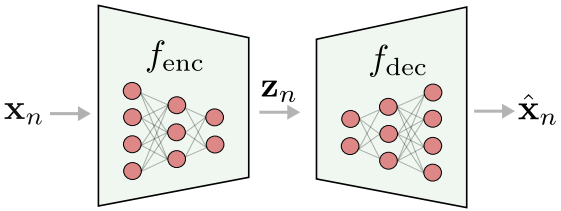
- An autoencoder (AE) is a neural network used for *unsupervised learning*

Autoencoders



- An autoencoder (AE) is a neural network used for *unsupervised learning*
- *Encoder*: transforms an unlabelled dataset, $\mathbf{x} := \{\mathbf{x}_n\}_n^N$, into latent codes, $\mathbf{z} := \{\mathbf{z}_n\}_n^N$
- *Decoder*: transforms latent codes into reconstructions, $\hat{\mathbf{x}} := \{\hat{\mathbf{x}}_n\}_n^N$

Autoencoders



- An autoencoder (AE) is a neural network used for *unsupervised learning*
- *Encoder*: transforms an unlabelled dataset, $\mathbf{x} := \{\mathbf{x}_n\}_n^N$, into latent codes, $\mathbf{z} := \{\mathbf{z}_n\}_n^N$
- *Decoder*: transforms latent codes into reconstructions, $\hat{\mathbf{x}} := \{\hat{\mathbf{x}}_n\}_n^N$
- We can do Bayesian Autoencoders! [**Tran et al.**, **NeurIPS**, **2021**]

Inductive Bias of the Optimized Priors

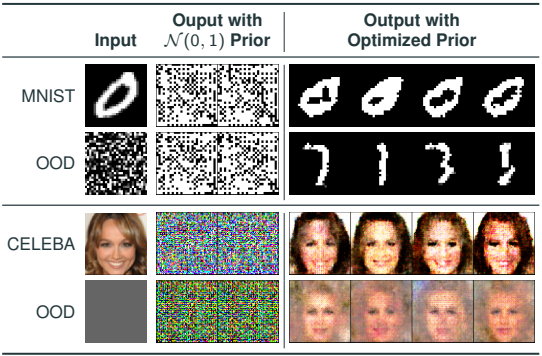


Figure 3: Realizations sampled from different priors given an input image. OOD stands for out-of-distribution.

Inductive Bias of the Optimized Priors

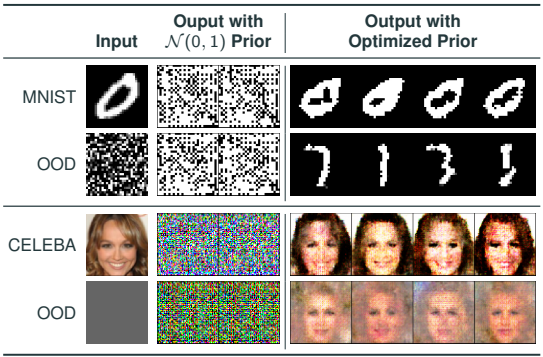


Figure 3: Realizations sampled from different priors given an input image. OOD stands for out-of-distribution.

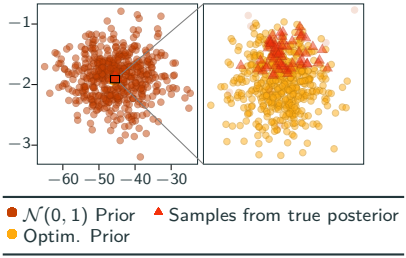
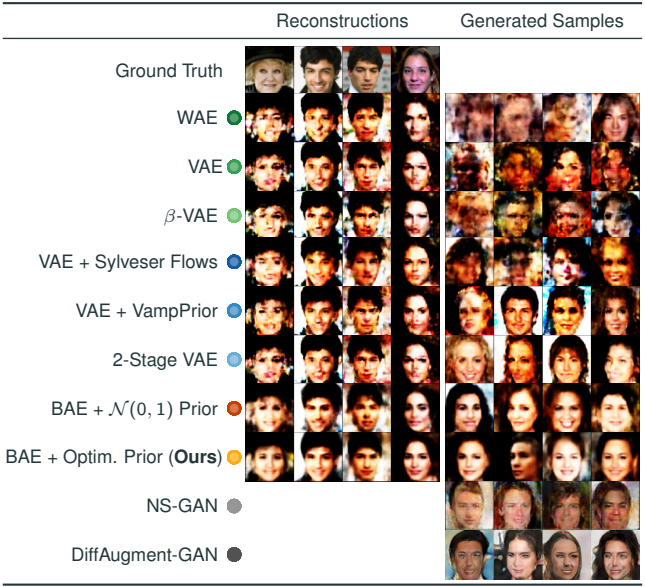
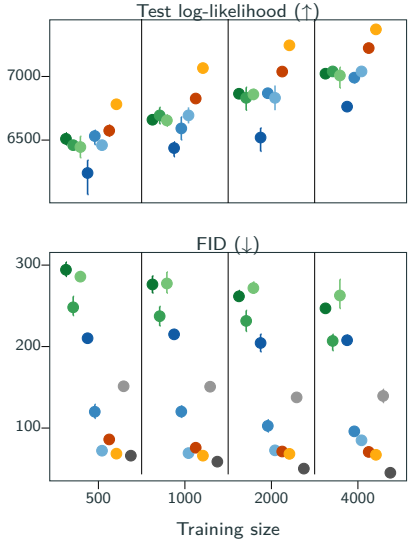


Figure 4: Visualization in 2D of samples from priors and posteriors of BAE parameters.

The hypothesis space of the optimized prior is reduced to regions close to the true posterior

Experiments on CelebA Dataset





Conclusions

- Choosing sensible priors for deep models is very important and difficult.
- Proposed a novel objective based on the Wasserstein distance.
  - Impose sensible priors for BNNs in function space.
  - Showed empirical benefits on a large variety of BNNs on supervised and unsupervised tasks.
  - Demonstrated that a fully Bayesian treatment of Bayesian deep learning models provides large performance gains.

Conclusions

- Choosing sensible priors for deep models is very important and difficult.

- Proposed a novel objective based on the Wasserstein distance.
  - Impose sensible priors for BNNs in function space.
  - Showed empirical benefits on a large variety of BNNs on supervised and unsupervised tasks.
  - Demonstrated that a fully Bayesian treatment of Bayesian deep learning models provides large performance gains.
- Physics-based priors
- Partially stochastic BNNs
- Acceleration (novel hardware, compression, ...)

References

[1] B.-H. Tran, S. Rossi, D. Milios, and M. Filippone. All You Need is a Good Functional Prior for Bayesian Deep Learning. *Journal of Machine Learning Research*, 23 (74), 1-56, 2022.

[2] B.-H. Tran, S. Rossi, D. Milios, P. Michiardi, E. V. Bonilla, and M. Filippone. Model Selection for Bayesian Autoencoders. *NeurIPS* 2021.

[3] B.-H. Tran, B. Shahbaba, S. Mandt, M. Filippone. Fully Bayesian Autoencoders with Latent Sparse Gaussian Processes. *ICML* 2023.

[4] A. Zammit-Mangion, M. D. Kaminski, B.-H. Tran, M. Filippone, and N. Cressie. Spatial Bayesian neural networks. *Spatial Statistics*, 60:100825, 2024.

Thank you!

Questions?