Bayesian Numerical Integration with Neural Networks

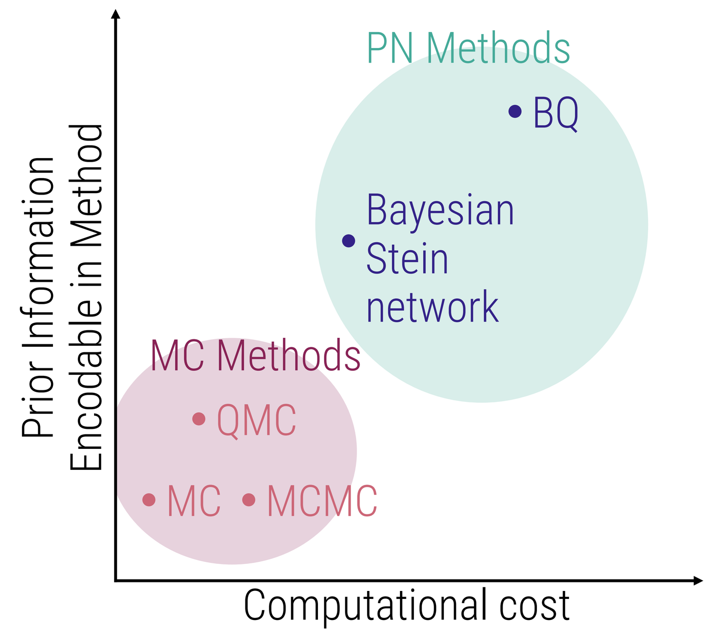
Katharina Ott [1, 2], Michael Tiemann [1] Philipp Hennig [2,3] François-Xavier Briol [4, 5]

[1]Bosch Center for Artificial Intelligence, Renningen, Germany [2]University of Tübingen, Tübingen, Germany
[3]MPI for Intelligent Systems, Tübingen, Germany [4]Department of Statistical Science, University College London, London, United Kingdom
[5]The Alan Turing Institute, London, United Kingdom

Introduction

Why Bayesian neural networks for integration tasks?

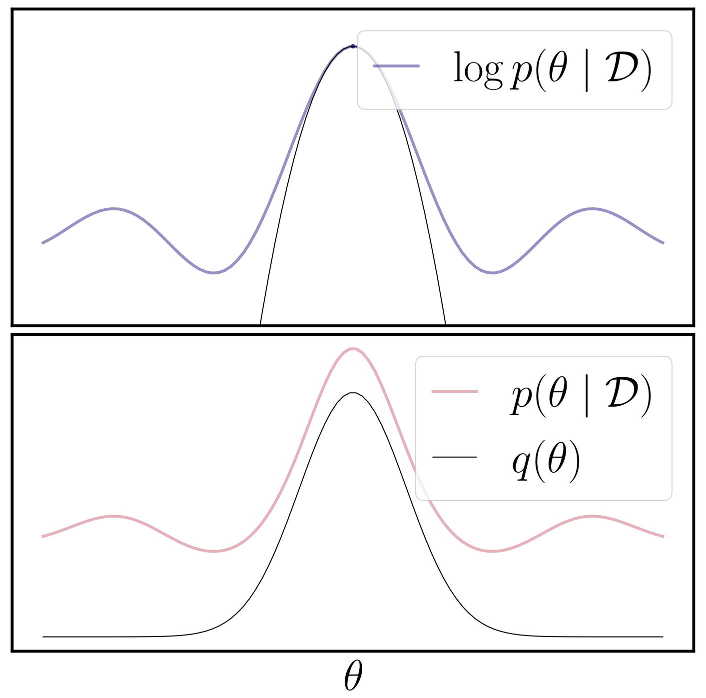
- Scaling to many data points.
- Scaling to higher dimensions.
- Memory efficient.
- Good scaling with number of data points by making use of neural network interpolation properties.
- Uncertainty estimates identify if data is too scarce to achieve desired accuracy.



Laplace Approximation

We want fast uncertainty estimates for $\theta_0 \rightarrow$ Compute posterior $p(\theta \mid \mathcal{D})$ of the networks parameters given data $\mathcal{D} \rightarrow$ Laplace approximation.

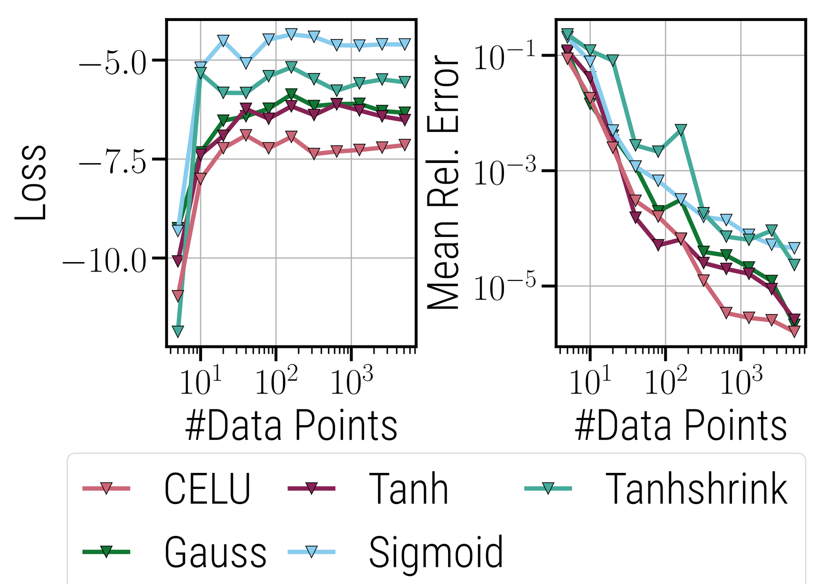
1. Note that minimum of weight regularized loss can be interpreted as mode of posterior $p(\theta \mid \mathcal{D})$ (maximum a-posteriori (MAP)) (square loss - likelihood, weight regularizer - prior) [2].
2. Normalization constant of posterior? $\rightarrow$ place Gaussian at the MAP - closed form approximation of posterior $q(\theta) \approx p(\theta \mid \mathcal{D})$.



Architectural Choices

The Stein network requires some special choices:

- **Choice of optimizer:** Use of L-BFGS, possibly due to training on entire dataset.
- **Choice of activation function:** Needs to be differentiable, CELU works best.
- **Bounded domains:** Replace $\tilde{u}(x) = u(x) \cdot \delta(x)$, where δ is 0 at the boundary, e.g., $\delta(x) = (a - x)$.



Overview

Task: We would like to compute integrals.

$$\Pi[f] = \int_{\mathcal{X}} f(x)\pi(x)dx,$$

Proposed solution: Compute the integral with an (approximate) Bayesian neural network.
How: Use Stein operator [1]:

$$\Pi\left[\mathcal{S}[u]\right] = 0,$$

where

$$\mathcal{S}[u](x) := \left(\nabla_x \log \pi(x)\right)^\top u(x) + \nabla_x u(x).$$

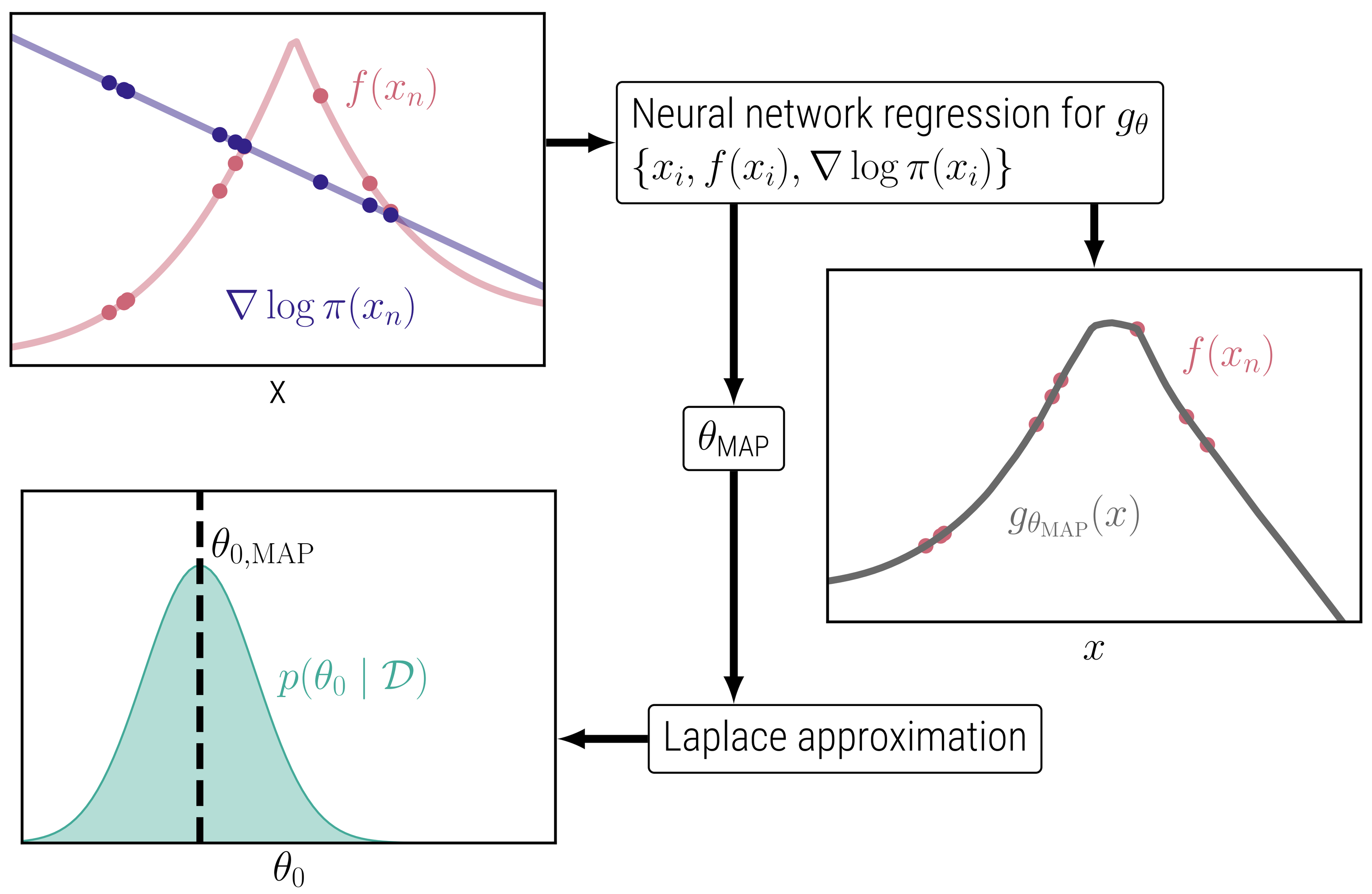
Then, we can define

$$g_\theta(x) := \mathcal{S}\left[u_{\theta_u}\right](x) + \theta_0,$$

where u_θ is a standard neural network.
If $g_\theta \approx f$ then

$$\theta_0 = \Pi[g_\theta] \approx \Pi[f].$$

Uncertainty estimates: Apply the Laplace approximation to the network.



References:

[1] Anastasiou et al. "Stein's method meets statistics: A review of some recent developments.". *arXiv*, 2021.

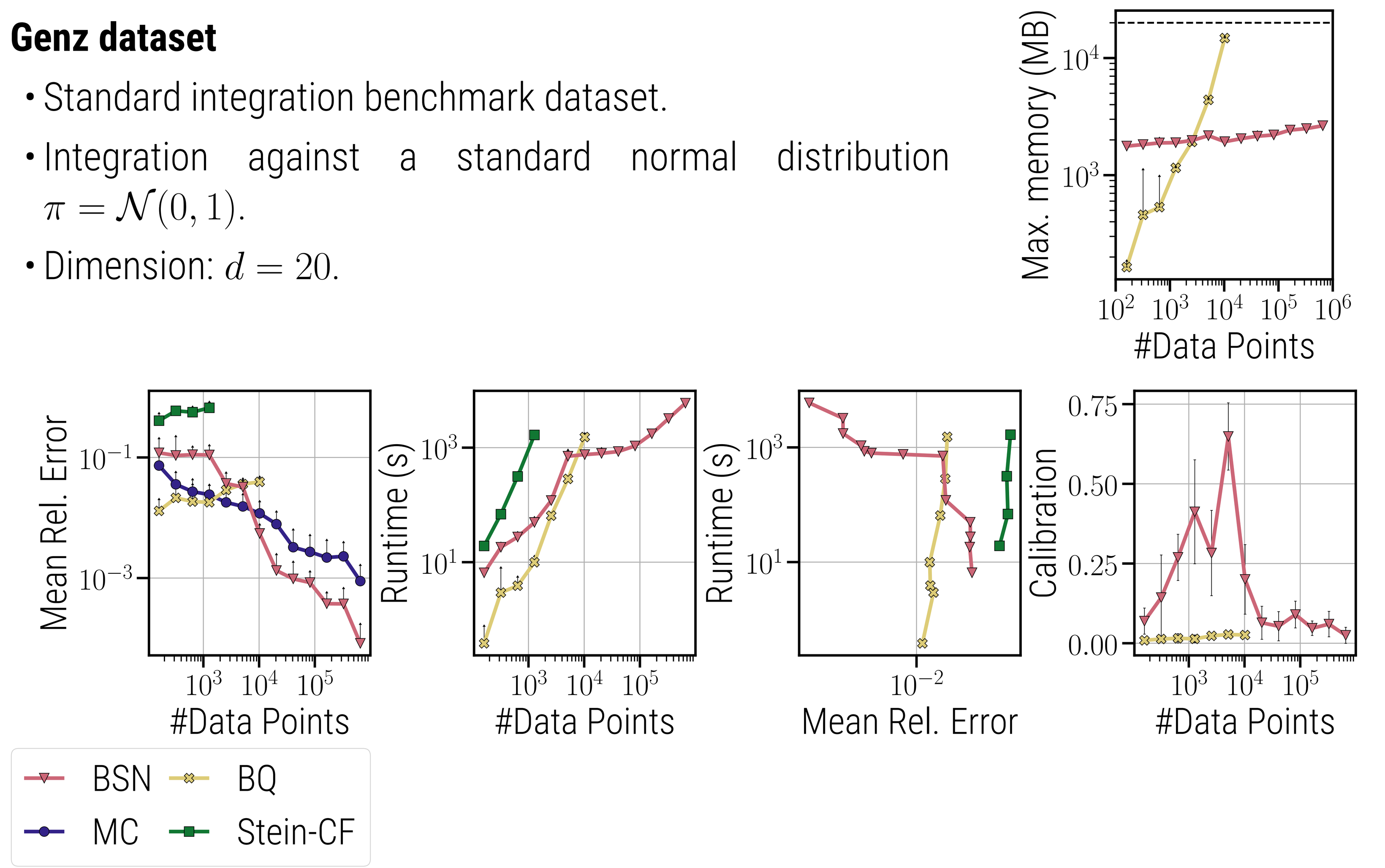
[2] MacKay "A practical Bayesian framework for backprogation networks". In *Neural computation*, 1992.

Link to paper

Experiments

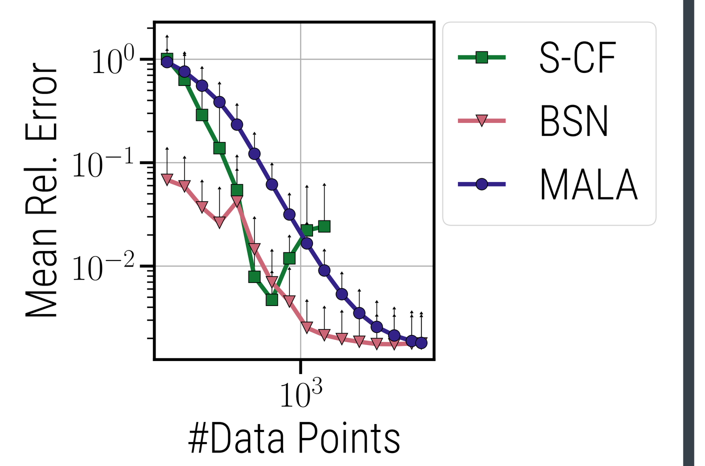
Genz dataset

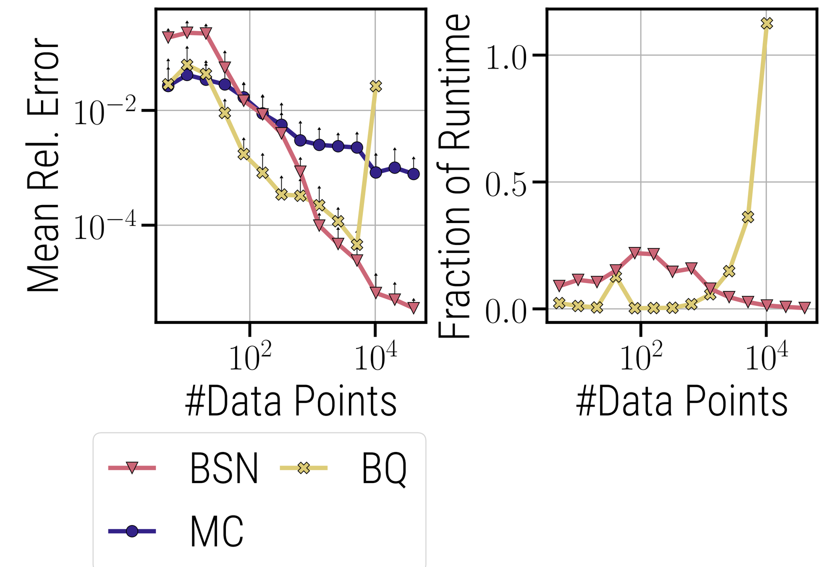
- Standard integration benchmark dataset.
- Integration against a standard normal distribution $\pi = \mathcal{N}(0, 1)$.
- Dimension: $d = 20$.



Parameter inference for ODEs

- Prior on parameters + measurements $\rightarrow$ Task: find posterior mean.
- Sampling expensive (expensive $\nabla \log \pi$).
- π not available in normalized form $\rightarrow$ BQ impractical.





Wind farm dataset

- Power output of a wind farm.
- Uncertainties over e.g., wind direction, turbulence intensity.
- Sampling slow (~ 2 min) - runtime of Bayesian quadrature and BSN negligible.

Summary

- BSN provides uncertainty estimates of good quality, identifies regions of low accuracy.
- BSN scales well to large number of data points and higher dimensions - regime where BQ becomes infeasible.
- BSN handles wide range of integration densities (e.g., if only available in unnormalized form).

katharina.ott@tum.de