Improving MCMC sampling efficiency with normalizing flows

ICPS 2024, Tbilisi, Georgia

05.08.2024

Willy Weber

(willy.weber@tu-dortmund.de)

Motivation: Statistical inference

technische universität dortmund

- **Bayesian inference:**

- Updating probabilities based on **Bayes Theorem**

$$\text{posterior} = \frac{\text{likelihood} \cdot \text{prior}}{\text{evidence}}$$

- Posterior distributions oftentimes complex

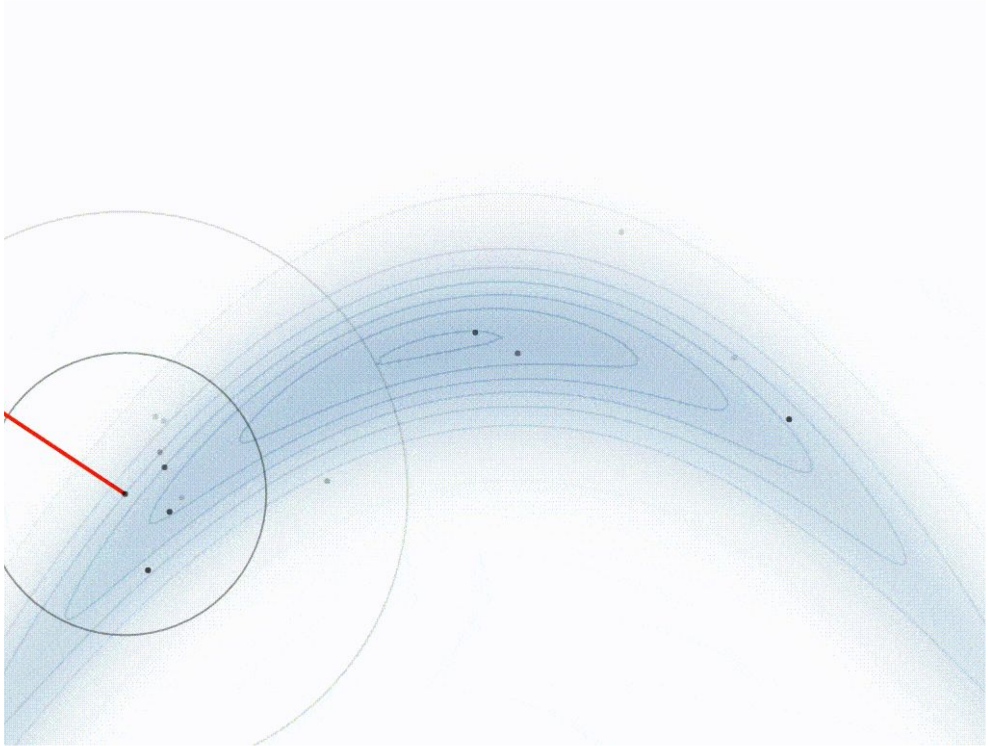
- **Bayesian Analysis Toolkit:**

BAT.jl
Bayesian Analysis Toolkit

- Framework for the use of Bayesian inference written in julia

- **Monte Carlo Markov Chain sampling** methods are used

Markov Chain Monte Carlo sampling (MCMC)

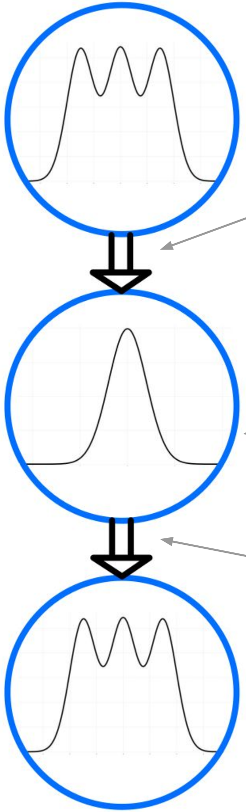


https://github.com/chi-feng/mcmc-demo

- Converges to the target distribution
- **Curse of dimensionality**

  Parameter space of a distribution grows exponentially with the number of parameters
- **Efficient proposal function** needed
  - High dimensional spaces
  - Complex distributions

Improve MCMC sampling with normalizing flow

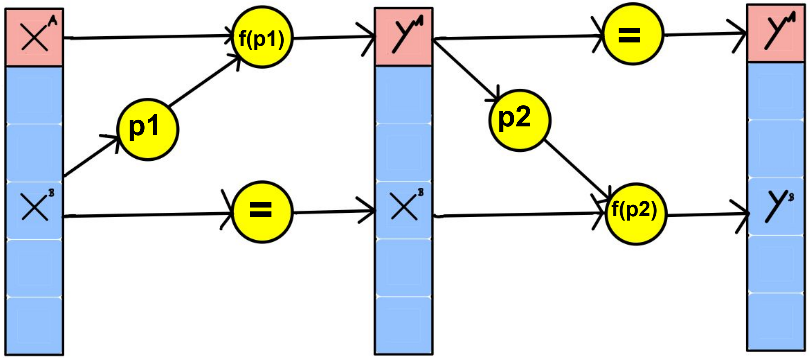


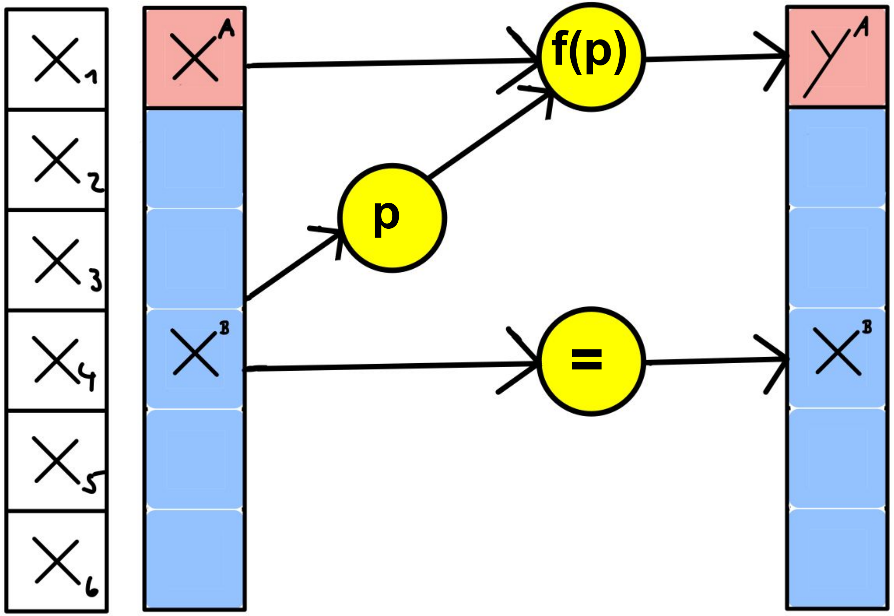
- **Transform** the target distribution into a simpler distribution (**Gaussian**)
- **Draw samples** in the simpler transformed space
- Apply the **inverse transformation** to obtain samples from the original target distribution

Coupling flows

- **Blockwise transformation** is a good choice for high-dimensional and multimodal data

- **One part of data is transformed** taking into account the correlation to the other part

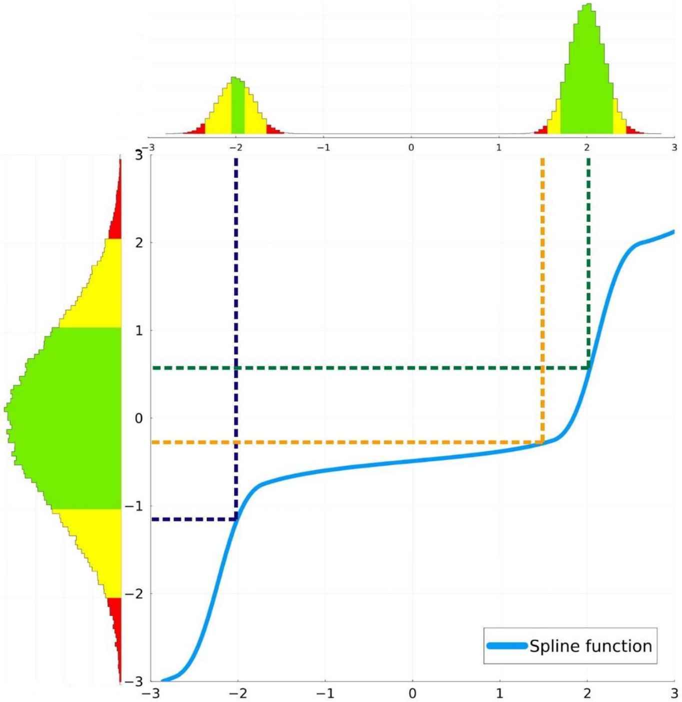




- The parameters p are learned in a way that **enables inversion**

Spline functions for transformation

Monotonous rational quadratic spline function:



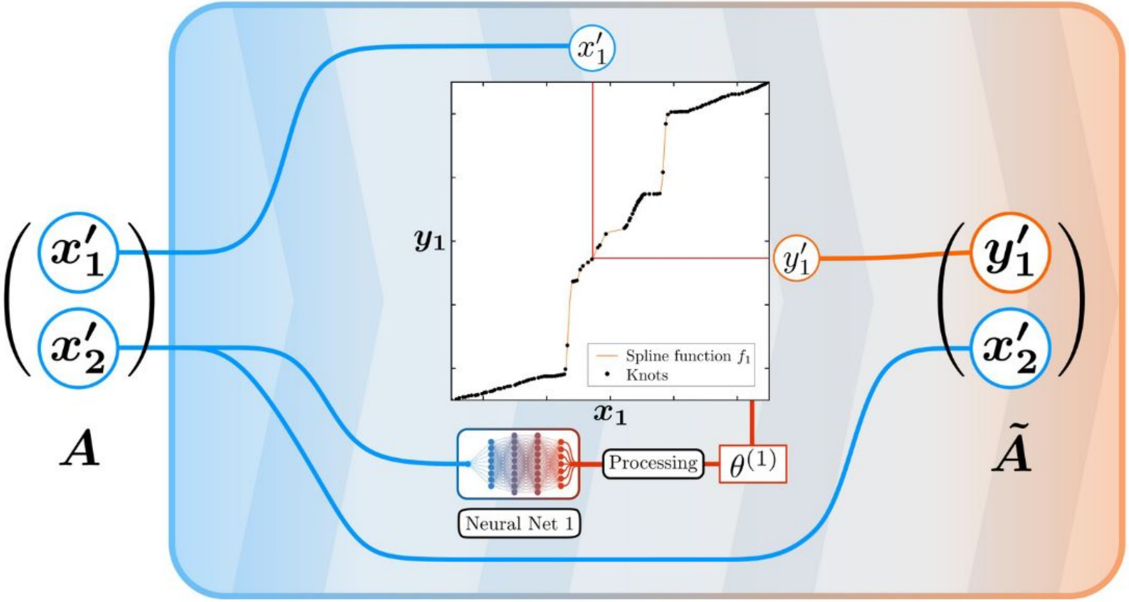
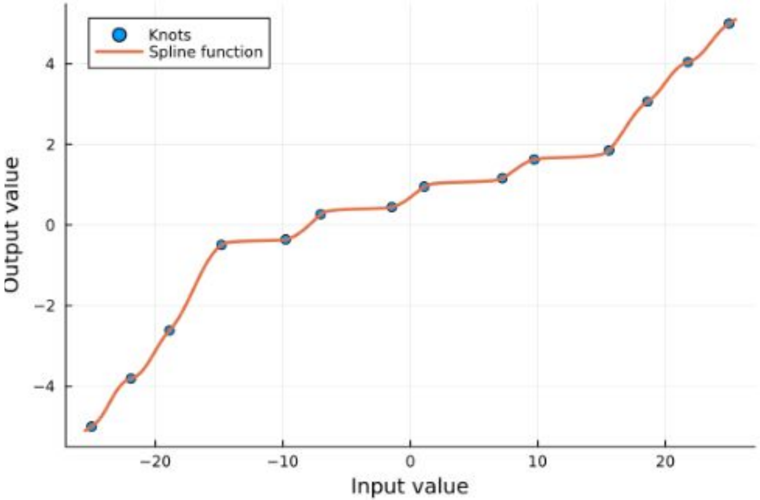
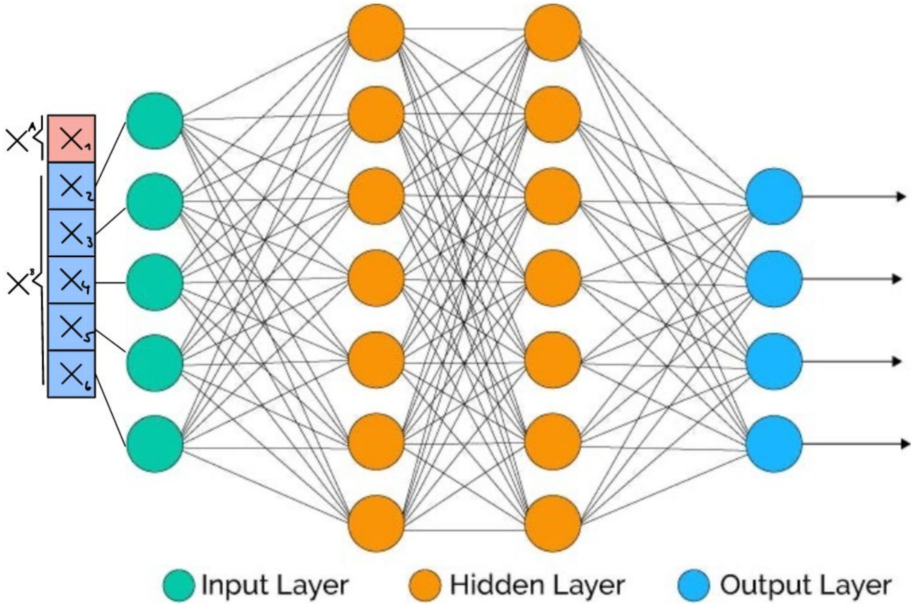


Image created by Michael Dudkowiak

Finding the best spline function

- With more **knots K**, a more complex spline function can be represented
- One **spline function** is defined by **3(K-1) parameters**





- Using a **dense neural network** to find an optimal spline function
- Input layer has (dimensions-1) neurons
- Output layer has 3(K-1) neurons

The Musketeer-flow-algorithm

- Each step one component is transformed based on all others

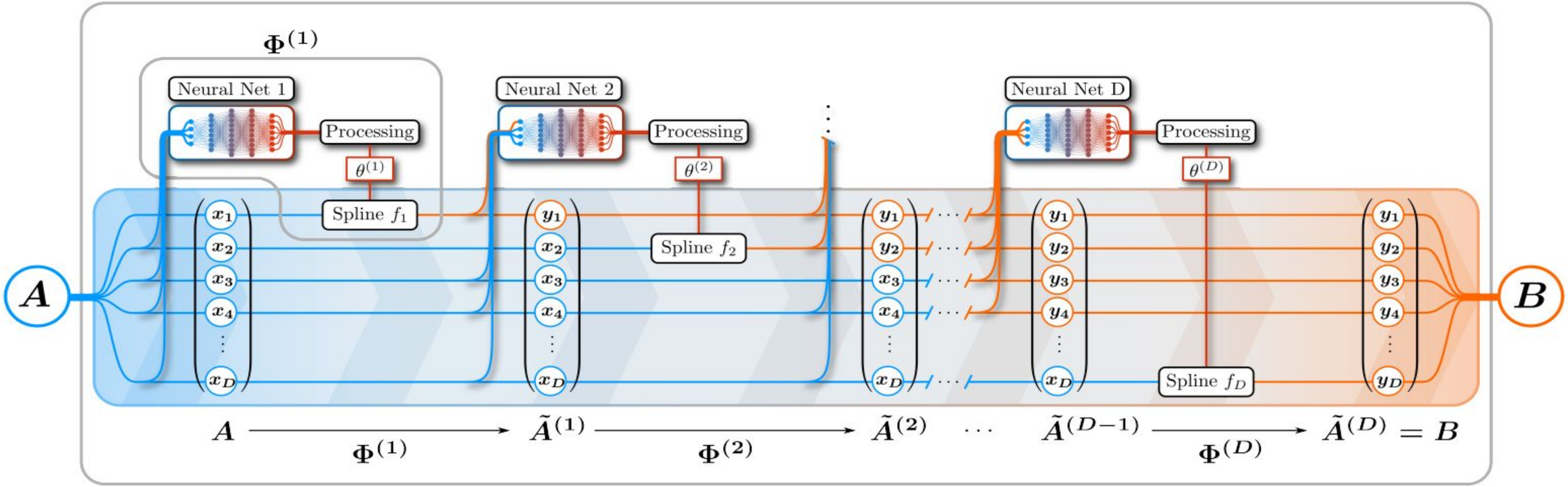
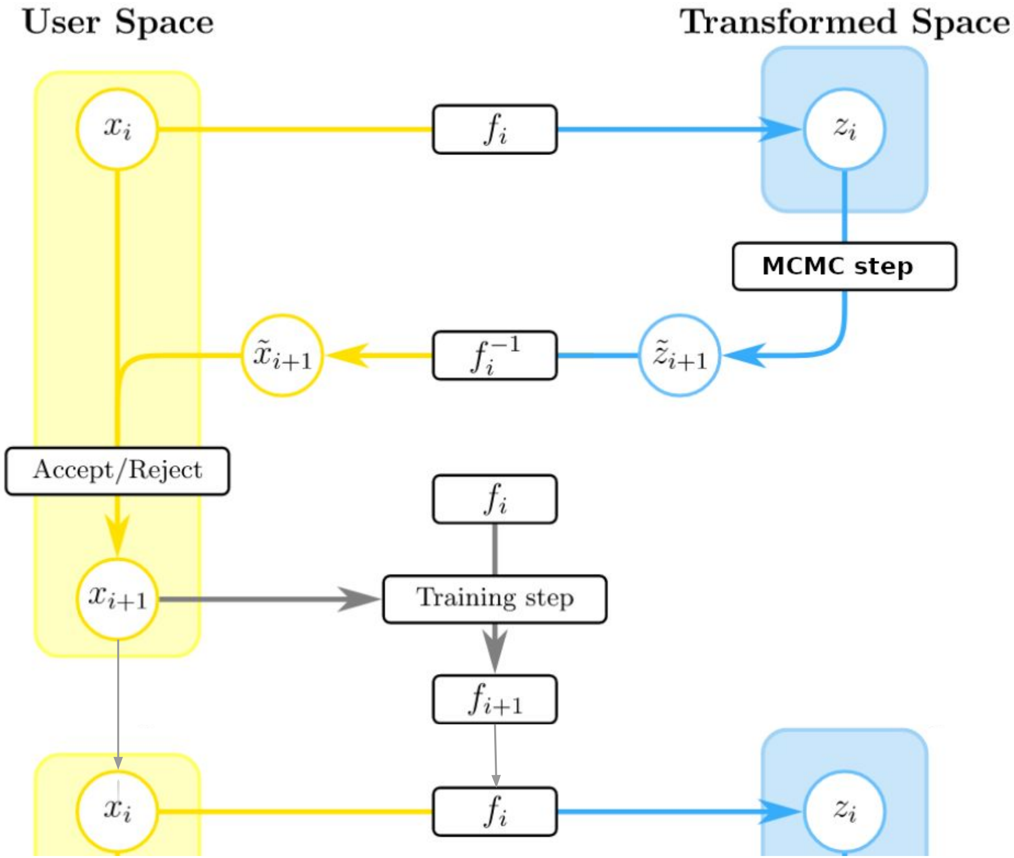


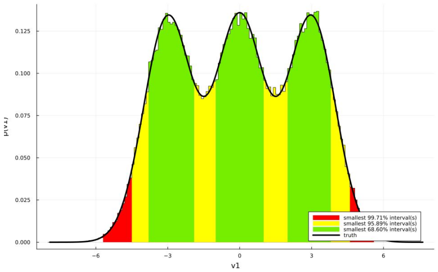
Image created by Michael Dudkowiak

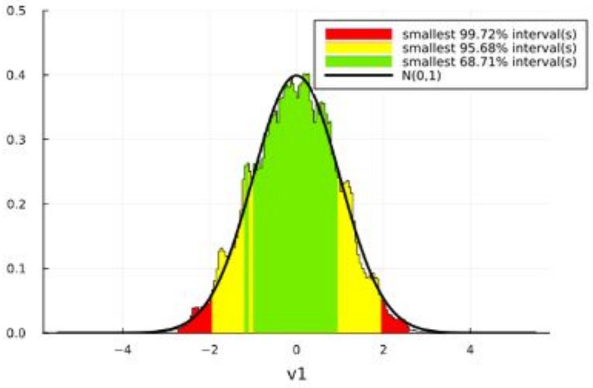
Combine MCMC sampling and flow training

- There is no pre-trained flow in reality
- **Train** a flow **during sampling** process
- Use every new sample to train the flow
- Draw **multiple samples in parallel** for efficiency

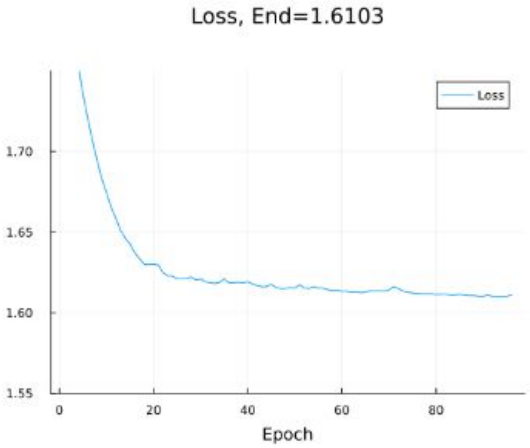


Train a normalizing flow during sampling

flow(

)= 

- Flow improves over time
- Trained on MCMC samples during sampling
- 1000 new samples per epoch



Summary

- **Normalizing flows** are interesting for the **improvement of MCMC sampling** methods
- An implementation is in development for the toolkit BAT.jl (Bayesian Analysis Toolkit)
- Initial toy experiments of training a flow during the sampling process were successful
- Next step is to study the potential for higher levels of complexity

Thank you for listening!