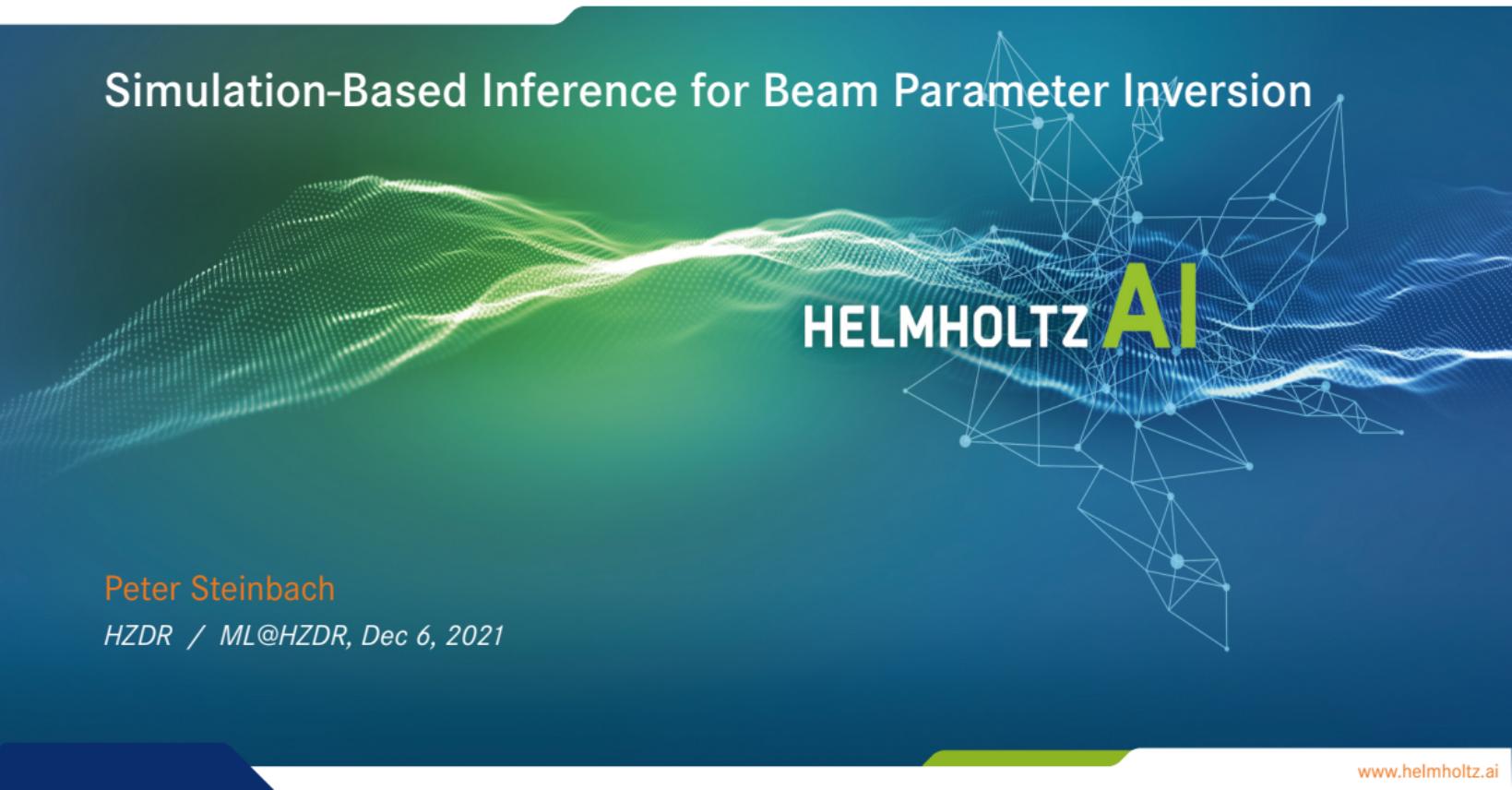


# Simulation-Based Inference for Beam Parameter Inversion

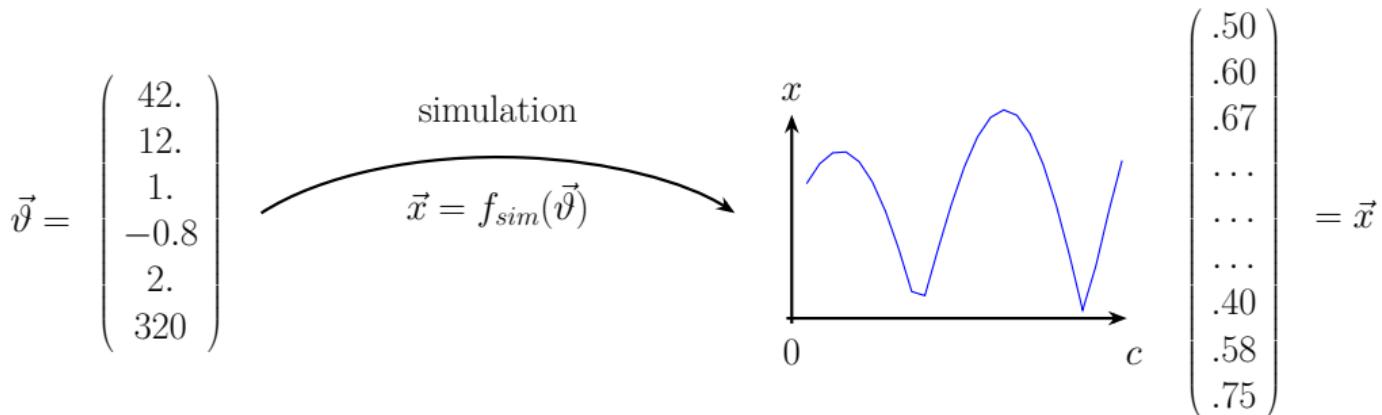
HELMHOLTZ **AI**

The background features a dynamic, abstract design. It consists of two main color-coded regions: a green area on the left and a blue area on the right. Both regions contain a wavy, undulating pattern composed of numerous small dots, giving it a textured, digital appearance. Overlaid on these waves is a complex network graph. It features a central cluster of nodes connected by lines, with several branches extending towards the edges of the frame. The nodes are small circles, and the connecting lines are thin, light-colored strokes. The overall effect is one of data flow, connectivity, and scientific inquiry.

Peter Steinbach

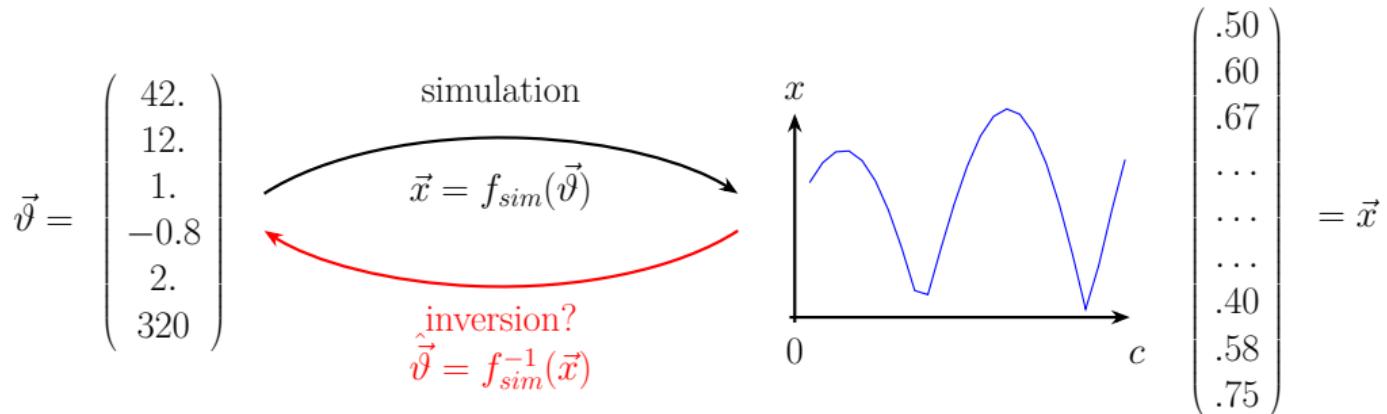
HZDR / ML@HZDR, Dec 6, 2021

# A common situation with simulations



- simulations used in many domains  
(physics, biology/medicine, chemistry, epidemiology, ...)
- approaches to simulations vary  
(mechanistic, agent based, distribution based, ...)
- simulations can be computationally challenging
- here: simulations = **forward process**

# Inversion of Simulations?



- inverse process hard to do (if at all tried)
- often, only single observables “fitted”
- simulations updated based on singular observables
- considerable human tuning involved (heuristics)

## Bayes Law

---

$$p(\vec{\vartheta}|\vec{x}) = \frac{p(\vec{x}|\vec{\vartheta}) \cdot p(\vec{\vartheta})}{\int p(\vec{x}|\vec{\vartheta})p(\vec{\vartheta})d\vec{\vartheta}}$$

## Bayes Law

---

$$p(\vec{\vartheta}|\vec{x}) = \frac{p(\vec{x}|\vec{\vartheta}) \cdot p(\vec{\vartheta})}{\int p(\vec{x}|\vec{\vartheta})p(\vec{\vartheta})d\vec{\vartheta}}$$

likelihood

## Bayes Law

---

$$p(\vec{\vartheta}|\vec{x}) = \frac{p(\vec{x}|\vec{\vartheta}) \cdot p(\vec{\vartheta})}{\int p(\vec{x}|\vec{\vartheta})p(\vec{\vartheta})d\vec{\vartheta}}$$

likelihood      prior

## Bayes Law

---

$$p(\vec{\vartheta}|\vec{x}) = \frac{p(\vec{x}|\vec{\vartheta}) \cdot p(\vec{\vartheta})}{\int p(\vec{x}|\vec{\vartheta})p(\vec{\vartheta})d\vec{\vartheta}}$$

posterior

likelihood      prior

## Bayes Law

$$p(\vec{\vartheta}|\vec{x}) = \frac{\overbrace{p(\vec{x}|\vec{\vartheta})}^{\text{likelihood}} \cdot \overbrace{p(\vec{\vartheta})}^{\text{prior}}}{\int p(\vec{x}|\vec{\vartheta})p(\vec{\vartheta})d\vec{\vartheta}}$$

- likelihood  $p(\vec{x}|\vec{\vartheta})$  provided by (forward) simulation
- prior  $p(\vec{\vartheta})$  given by how we sample the simulation parameters  $\vec{\vartheta}$

## Bayes Law

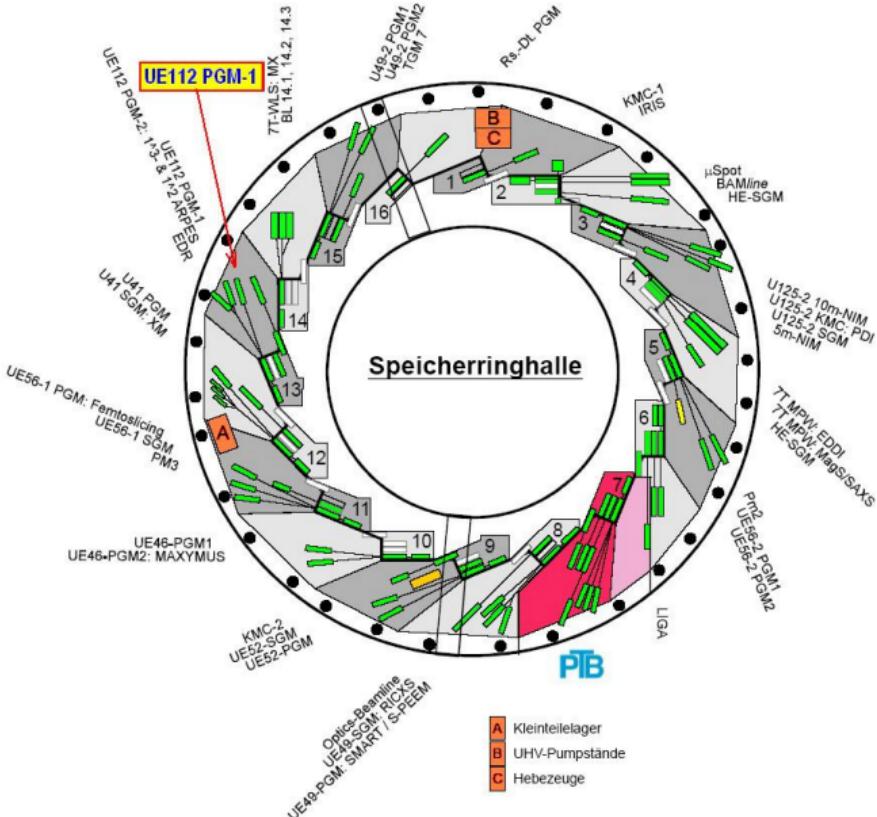
$$p(\vec{\vartheta}|\vec{x}) = \frac{p(\vec{x}|\vec{\vartheta}) \cdot p(\vec{\vartheta})}{\int p(\vec{x}|\vec{\vartheta})p(\vec{\vartheta})d\vec{\vartheta}}$$

posterior                      likelihood              prior

- likelihood  $p(\vec{x}|\vec{\vartheta})$  provided by (forward) simulation
- prior  $p(\vec{\vartheta})$  given by how we sample the simulation parameters  $\vec{\vartheta}$

**Goal:** predict posterior to the best of our abilities!

# Why do I care?

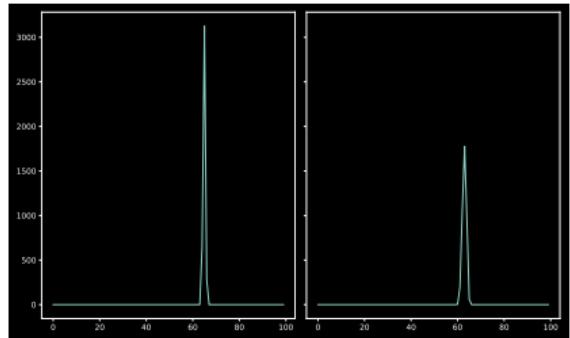
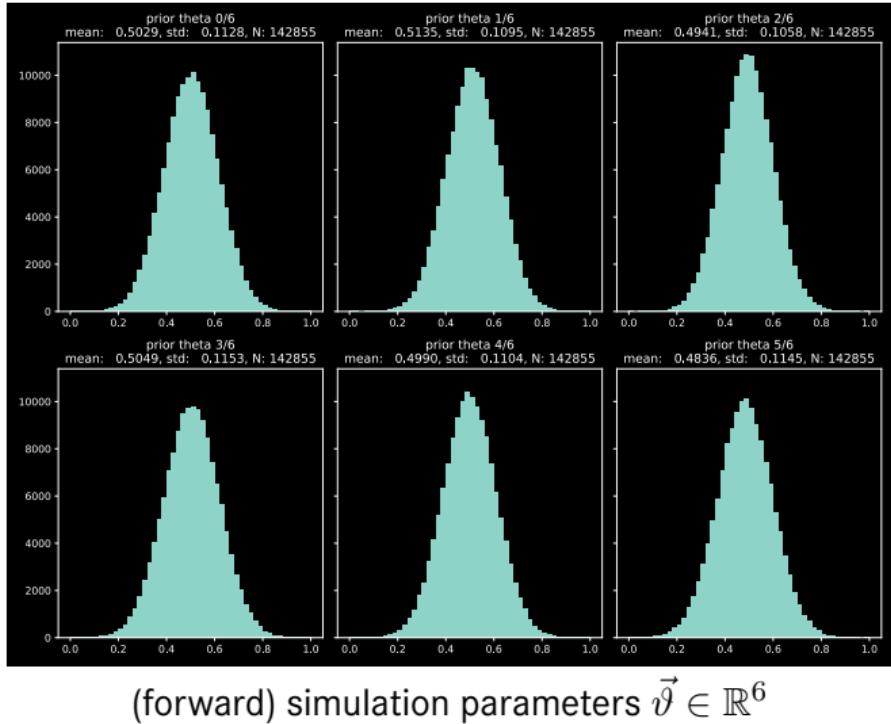


## Scientific Question

*Given a beam profile, what were the beamline optics parameters that likely produced it?*

- beamline UE112 PGM-1 at BESSY
- beam characteristics fixed at electron storage ring outlet
- forward simulations by rayUI

# My data from a surrogate



knife-edge scans to quantify beam quality at experimental station,

$$\vec{x} \in \mathbb{R}^{200}$$

(surrogate: multivariate normal of 2 dimensions plus 2 extra variables)

# Conditional Invertible Neural Networks

# GUIDED IMAGE GENERATION WITH CONDITIONAL INVERTIBLE NEURAL NETWORKS

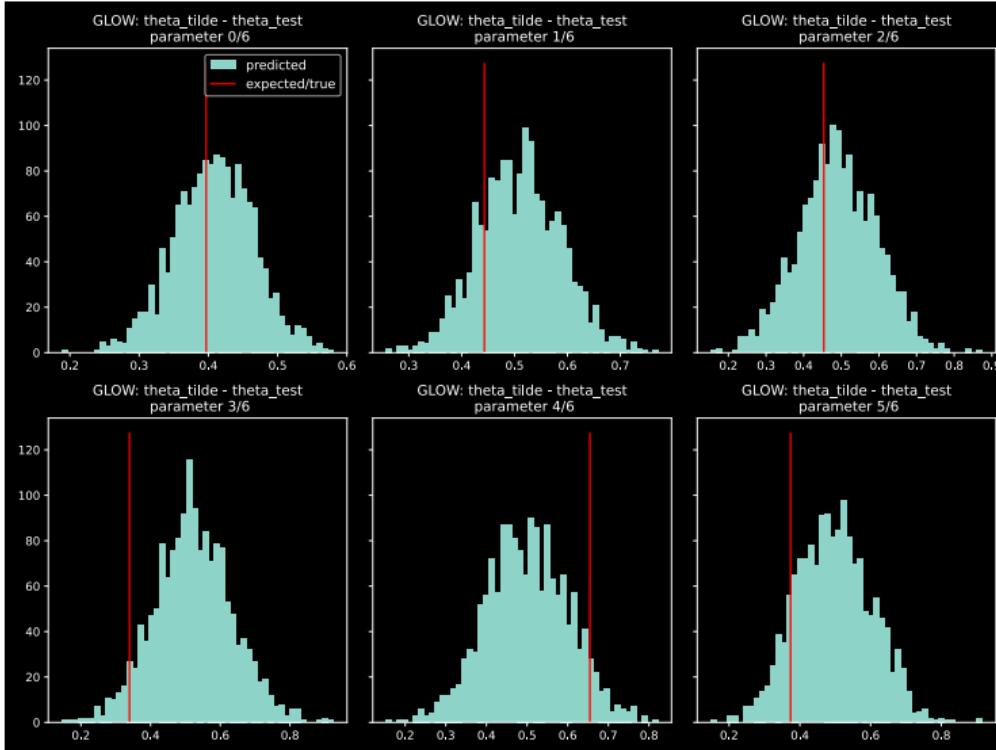
Lynton Ardizzone, Carsten Lüth, Jakob Kruse, Carsten Rother, Ullrich Köthe  
Visual Learning Lab Heidelberg

## ABSTRACT

In this work, we address the task of natural image generation guided by a conditioning input. We introduce a new architecture called conditional invertible neural network (cINN). The cINN combines the purely generative INN model with an unconstrained feed-forward network, which efficiently preprocesses the conditioning input into useful features. All parameters of the cINN are jointly optimized with a stable, maximum likelihood-based training procedure. By construction, the cINN does not experience mode collapse and generates diverse samples, in contrast to e.g. cGANs. At the same time our model produces sharp images since no reconstruction loss is required, in contrast to e.g. VAEs. We demonstrate these properties for the tasks of MNIST digit generation and image colorization. Furthermore, we take advantage of our bi-directional cINN architecture to explore and manipulate

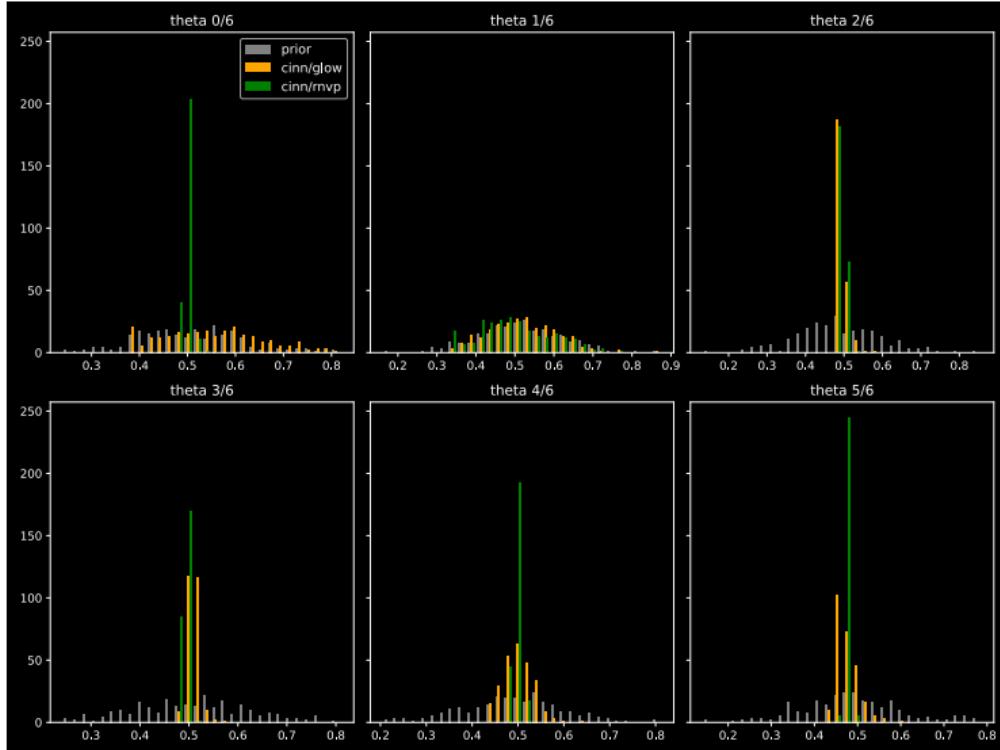


# cINN Inference on the validation set



- cINN provides posterior that can be sampled
- extract **Maximum a posteriori estimation (MAP)** estimate by mean/median

# cINN MAPs on the validation set



- training:
  - 30 epochs only
  - fixed arch: 8 layers
  - 256 units per dense layer
- inference: 256 draws per validation sample, MAP by mean
- good: posterior stays within prior support
- to improve: posterior misses out for some dimensions

# What is going on?

---

## core assumption(s)

- network  $f_{cinn}$  is sufficiently expressive
- as  $N_{\text{simulations}} \rightarrow \infty$ , network allows mapping of  $\vec{x}$  onto  $p(\vec{\vartheta}|\vec{x})$
- training dataset imposes the entire “truth” through implicit prior  $p_{\text{simulation}}(\vec{\vartheta})$

# What is going on?

---

## core assumption(s)

- network  $f_{cinn}$  is sufficiently expressive
- as  $N_{\text{simulations}} \rightarrow \infty$ , network allows mapping of  $\vec{x}$  onto  $p(\vec{\vartheta}|\vec{x})$
- training dataset imposes the entire “truth” through implicit prior  $p_{\text{simulation}}(\vec{\vartheta})$

## Let's reconsider

- goal: infer  $\vec{\vartheta}|\vec{x}_o$  on observation  $\vec{x}_o$
- **but:** the global learned posterior may not be too informative at  $p(\vec{\vartheta}|\vec{x}_o)$  (as we fixed the prior)

# What is going on?

## core assumption(s)

- network  $f_{cinn}$  is sufficiently expressive
- as  $N_{\text{simulations}} \rightarrow \infty$ , network allows mapping of  $\vec{x}$  onto  $p(\vec{\vartheta}|\vec{x})$
- training dataset imposes the entire “truth” through implicit prior  $p_{\text{simulation}}(\vec{\vartheta})$

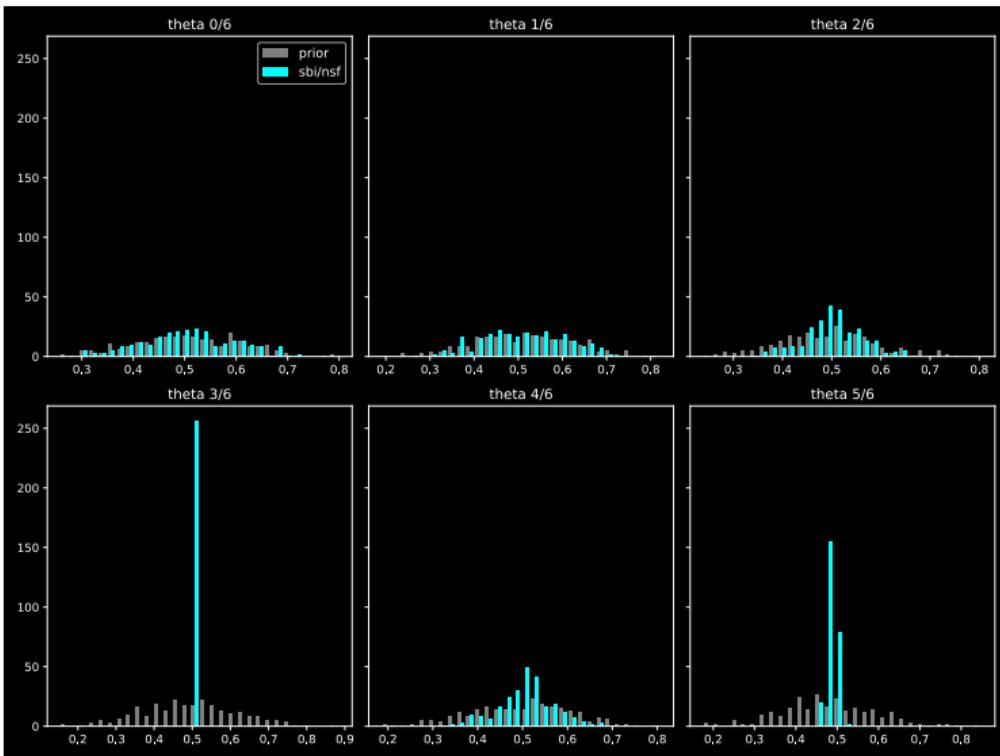
Let's reconsider

- goal: infer  $\vec{\vartheta}|\vec{x}_o$  on observation  $\vec{x}_o$
- **but:** the global learned posterior may not be too informative at  $p(\vec{\vartheta}|\vec{x}_o)$  (as we fixed the prior)

**sequential neural density estimation (SNPE) with**  
[www.mackelab.org/sbi](http://www.mackelab.org/sbi)

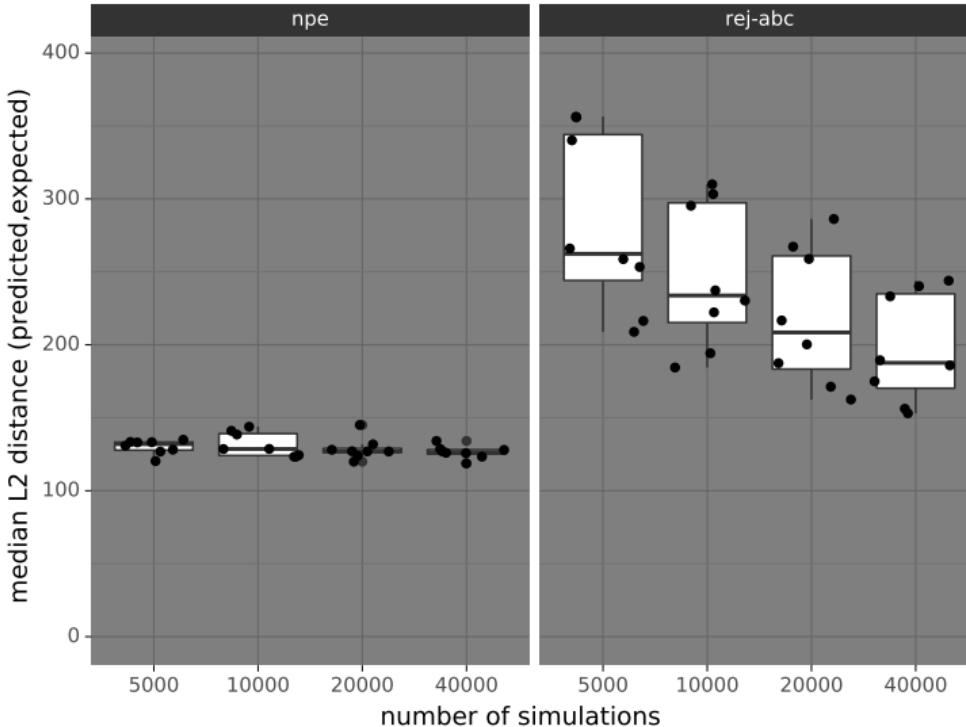
# (Sequential) neural density estimation

# NPE: sbi MAPs on the validation set



- training:
  - >100 epochs only
  - fixed arch: 8 layers, neural spine flow  
*Durkan et al, 2019*
  - 256 units per layer
- training: only one round (NPE) to produce amortized posterior
- inference: 256 draws per validation sample, MAP by mean
- good: posterior stays within prior support
- to improve: posterior misses out for two dimensions (expected)

## presently: comparison to other methods



- using surrogate simulation of a beamline (projected multivariate normal)
- integrating into `sbibm`
- plan: compare to more SBI based approaches (sequential and non-sequential)

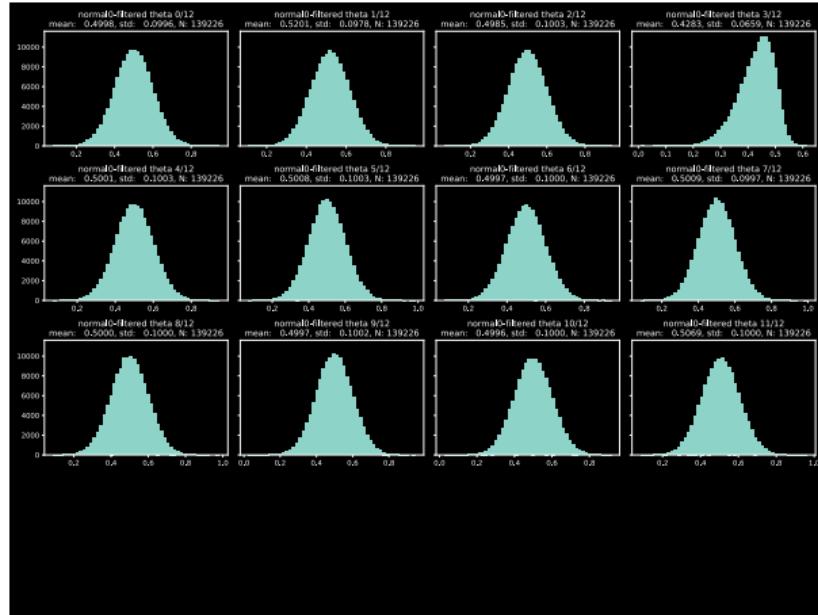
## Summary

---

- simulation-based inference (*sbi*) can be a tool for inversion (either NPE with cINN or SNPE)
- *sbi* is a light weight and yet flexible framework
- neural spline flow used in SNPE appears more flexible
- quality control & comparison:
  - currently: how do other methods compare ([Lueckmann et al, 2021](#))
  - sample based metrics ([Naeem et al, 2020](#))
  - simulation based calibration ([Talts et al, 2018](#))

# the real world

---



**reproducibility, openness & team work = key!**  
(all results from above were from toy simulations)

## Further Reading

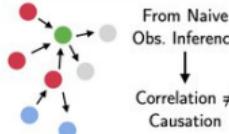
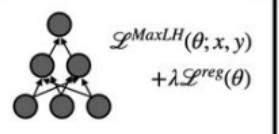
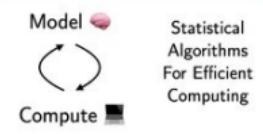
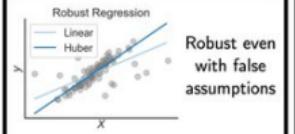
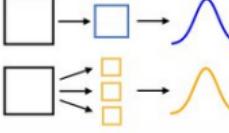
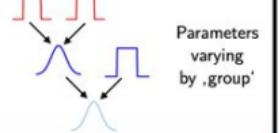
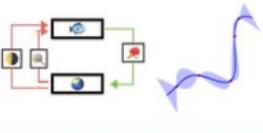
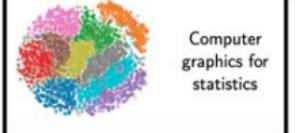
---

- nice blog post with pytorch code samples
- “Normalizing Flows for Probabilistic Modeling and Inference,” G. Papamakarios et al, [arXiv:1912.02762](https://arxiv.org/abs/1912.02762), 2019.
- “Normalizing Flows: An Introduction and Review of Current Methods”, I. Kobyzev et al, [arXiv:1908.09257](https://arxiv.org/abs/1908.09257), 2019.
- “Glow: Generative Flow with Invertible 1x1 Convolutions”, Kingma et al, [arXiv:1807.03039](https://arxiv.org/abs/1807.03039), 2018.

# Backup

# What are the Most Important Statistical Ideas of the Past 50 Years?

- Andrew Gelman & Aki Vehtari (2021) -

<b>1</b> Counterfactual Causal Inference	<b>3</b> Overparametrized Models & Regularization	<b>5</b> Generic Computation Algorithms	<b>7</b> Robust Inference	Commonalities & $\Delta s$
 <p>From Naive Obs. Inference Correlation ≠ Causation</p> <p>Econometrics, Epidemiology, Psychology, Stats, CS</p>	 <p>Deep Learning, L1/L2/ElasticNet, Sparse Structure Discovery</p>	 <p>EM, DP, MCMC, ABC, Variational Inference</p>	 <p>Robust standard errors, leverage, M-open world</p>	<ul style="list-style-type: none"><li>○ Codify 'philosophy'</li><li>○ Data &amp; Model Split</li><li>○ Scalability w. Data</li><li>○ Connecting ideas</li><li>○ Theory <math>\Leftrightarrow</math> Applied</li><li>○ Interlinked ideas</li></ul>
<b>2</b> Bootstrapping & Simulation-Based Inference	<b>4</b> Multilevel Models	<b>6</b> Adaptive Decision Analysis	<b>8</b> Exploratory Data Analysis	Looking Back/Forward
 <p>Substitute computation for mathematical analysis</p>	 <p>Bayesian models, partial pooling local &amp; global info <math>\rightarrow</math> combination</p>	 <p>Deep RL, Bayesian Optimisation, A/B Testing, Online Learning</p>	 <p>Computer graphics for statistics</p> <p>Open-ended discovery, Interpret, Data <math>\Leftrightarrow</math> Model <math>\Leftrightarrow</math> Predictions</p>	<ul style="list-style-type: none"><li>○ 20'-70': Latent Var. Models, Exp. Design, Sampling Theory, ...</li><li>○ Future: Continuation Unit Tests for Stats, Interpretable ML, ...</li></ul>

#MLCollage - @RobertTLange [17/52]

Simulation-based inference discovered recently  
(Cranmer et al, 2020, Gelman & Vehtari, 2021)

**Algorithm 1** APT with per-round proposal updates

**Input:** simulator with (implicit) density  $p(x|\theta)$ , data  $x_o$ , prior  $p(\theta)$ , density family  $q_\psi$ , neural network  $F(x, \phi)$ , simulations per round  $N$ , number of rounds  $R$ .

```

 $\tilde{p}_1(\theta) := p(\theta)$ 
for  $r = 1$  to  $R$  do
    for  $j = 1$  to  $N$  do
        Sample  $\theta_{r,j} \sim \tilde{p}_r(\theta)$ 
        Simulate  $x_{r,j} \sim p(x|\theta_{r,j})$ 
    end for
     $\phi \leftarrow \operatorname{argmin}_{\phi} \sum_{i=1}^r \sum_{j=1}^N -\log \tilde{q}_{x_{i,j}, \phi}(\theta_{i,j})$       using (2)
     $\tilde{p}_{r+1}(\theta) := q_{F(x_o, \phi)}(\theta)$ 
end for
return  $q_{F(x_o, \phi)}(\theta)$ 
```

- sequentially update prior to form a proposal prior  $\tilde{p}(\vartheta)$  (and a  $\tilde{p}(\vartheta|\vec{x})$ )
- loss function has to be adapted
- cINNs can be used as conditional density estimator
- **Greenberg et al, 2019:**  
“Learning with such ‘atomic’ proposals has an intuitive interpretation: we are training the network to solve multiple choice test problems, of the format “which of these  $\vartheta$ ’s generated this  $x$ ?”