An Invitation to Simulation-Based Inference

12 Sep, ASC school ’22

Alex Cole (U. of Amsterdam)
a.e.cole2@uva.nl
@a_e_cole

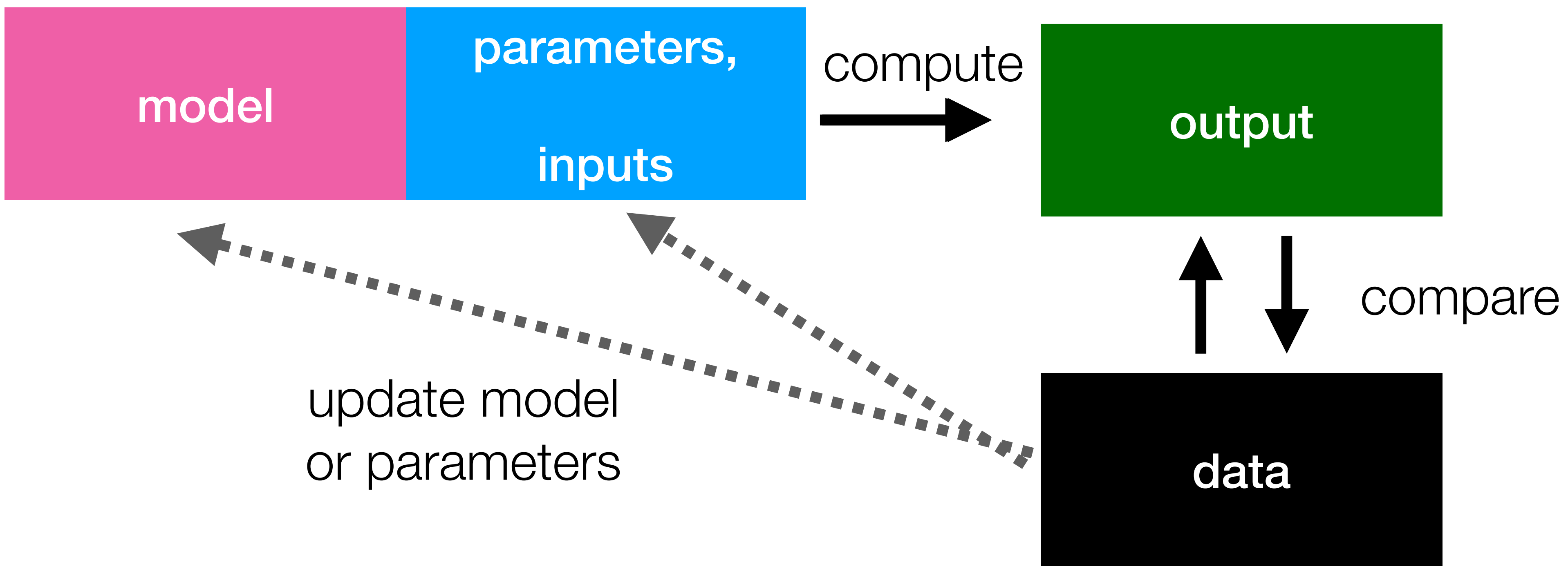
Outline

1. Forward and inverse maps in the scientific method
    1. Classical inference
        1. Drawbacks
2. Simulation-based inference
    1. Applications
    2. Software
    3. Cutting-edge considerations

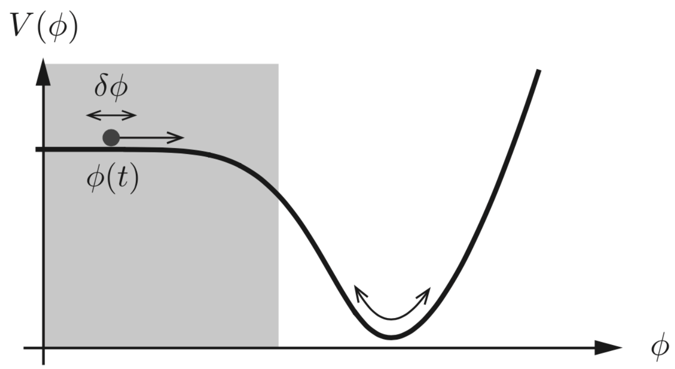
Motivation

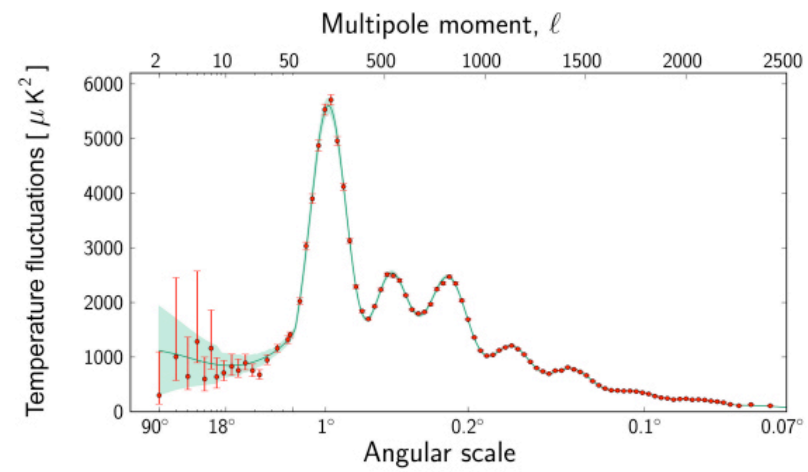
Toy Workflow

- In science, we often perform some subdiagram of:



inflation





model

parameters,

inputs

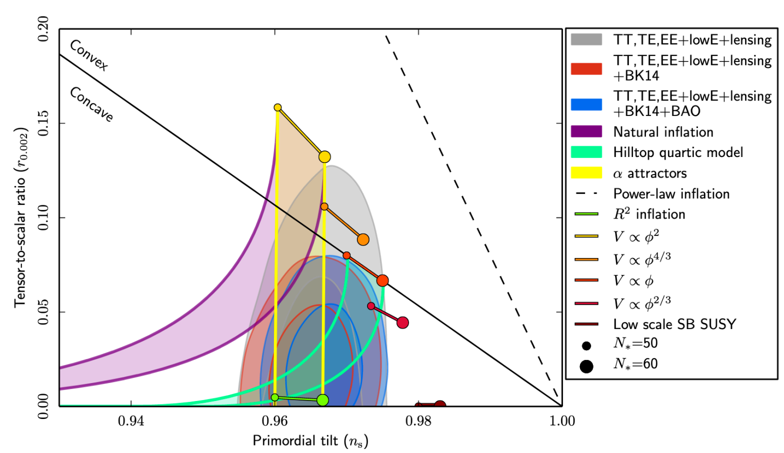
compute

output

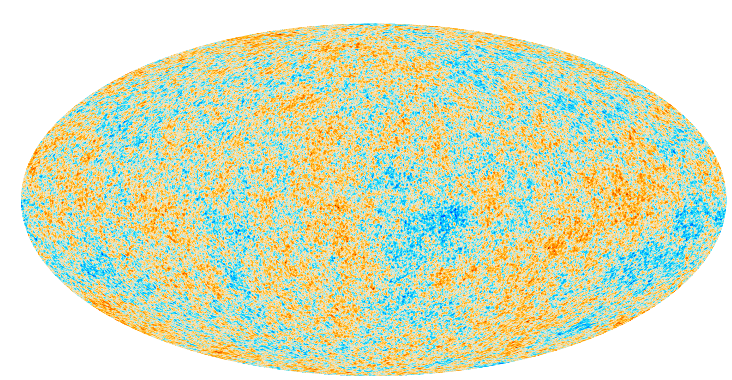
compare

constrain
inflationary models

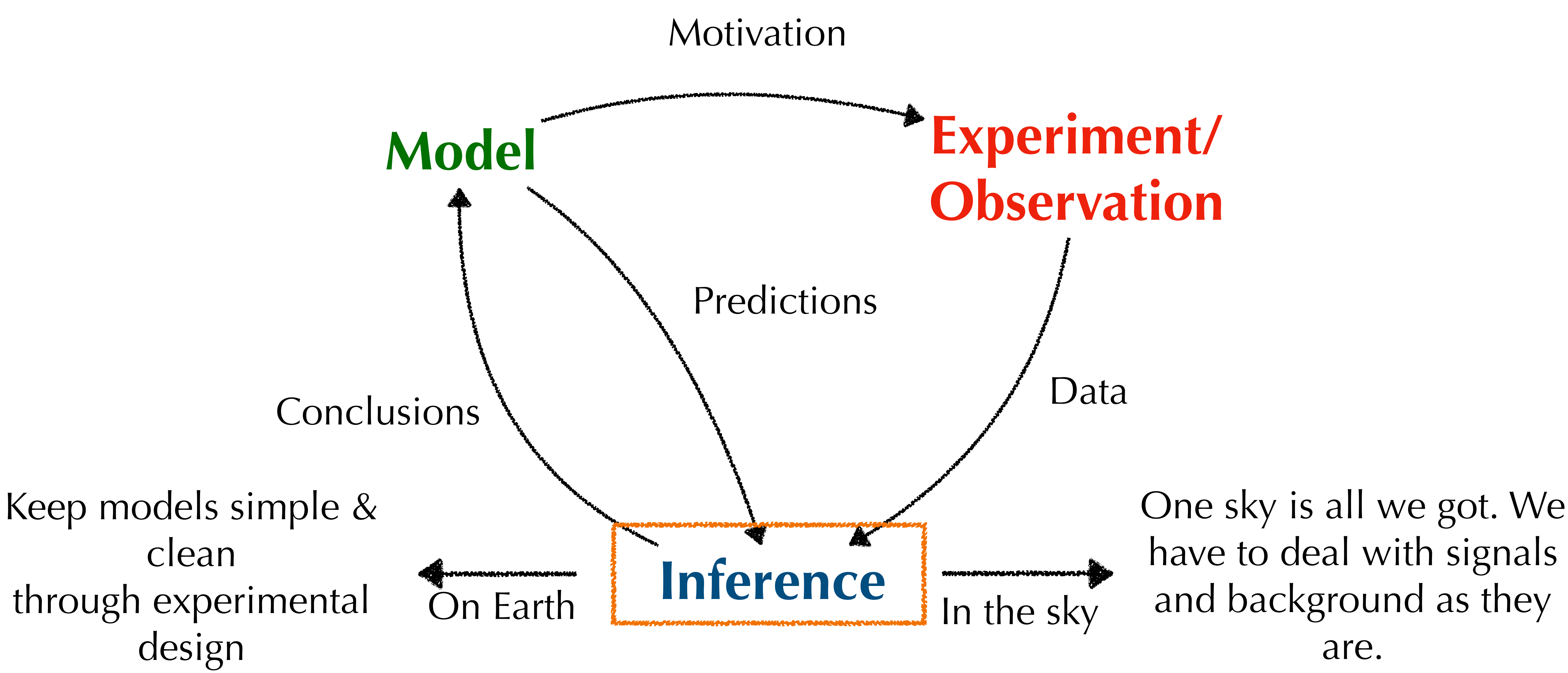
data



[Planck '18]



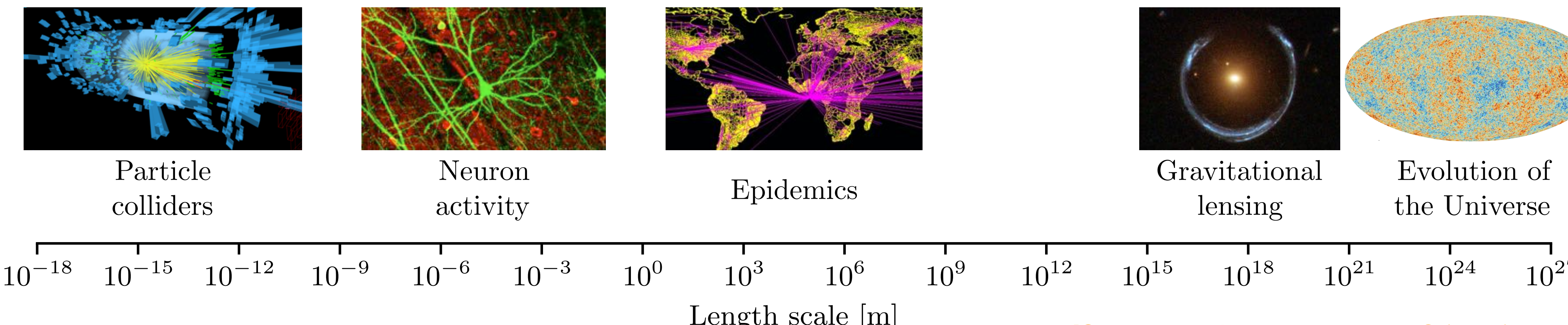
How we progress



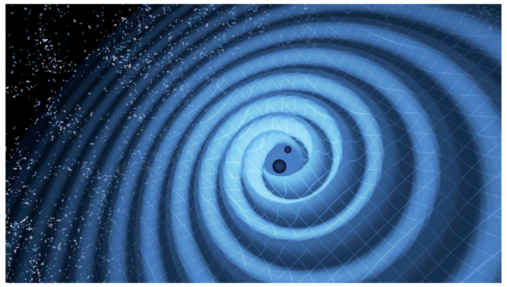
slide: C. Weniger

model $\xrightarrow{\text{compute}}$ output

- Advanced computational models allow us to simulate data across length scales:



[Cranmer, Brehmer, Louppe PNAS (2020), arXiv:1911.01429]



gravitational waves

- However, forward models are not well-suited for statistical inference.

Parameter Inference

- Go from data to constraints using **Bayes' formula**

θ: parameters

$$\underbrace{p\left(\theta \,|\, x\right)}_{\text{posterior}} = \frac{\overbrace{p\left(x \,|\, \theta\right)}^{\text{likelihood}} \, \overbrace{p(\theta)}^{\text{prior}}}{\underbrace{p(x)}_{\text{evidence}}}$$

x: data

- Classical techniques (Markov Chain Monte Carlo, Nested Sampling) use **evaluations of the likelihood** to accept/reject proposed steps, giving (weighted) samples of the **joint posterior** $p(\theta \,|\, x),\ \theta = (\theta_1, \theta_2, \ldots, \theta_D)$

Intractability

- The word **intractable** often shows up when discussing Bayesian inference.

- What is typically meant is there is a *high-dimensional integral* we don't have the resources to perform numerically, e.g.
$p(x) = \int d\theta p(x \mid \theta)p(\theta)$ (with θ high-dimensional).

The evidence is typically intractable

$\Longrightarrow$ MCMC, …

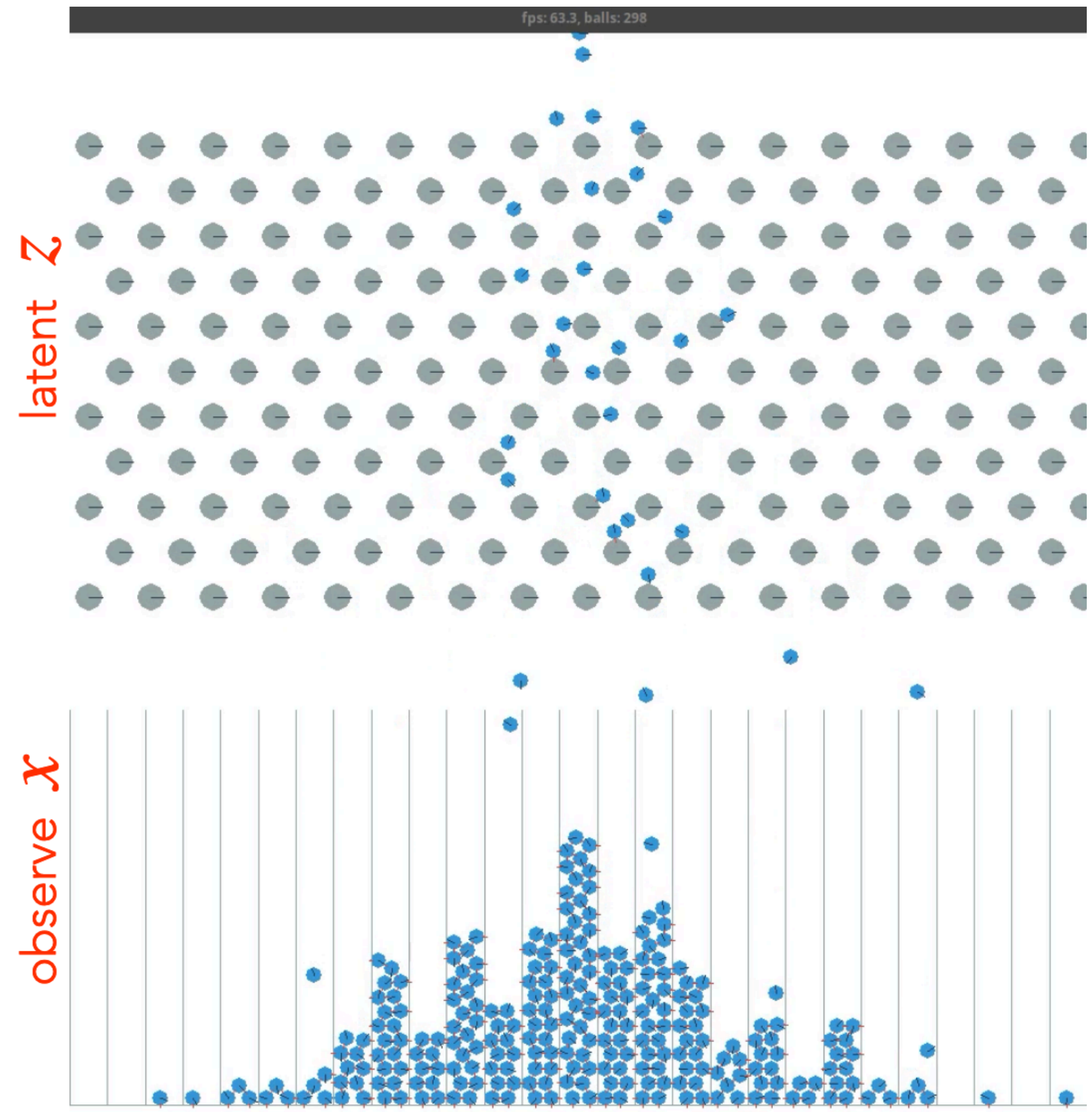
- Note that the likelihood can even be intractable,
$p(x \mid \theta) = \int d\eta p(x, z \mid \theta)$ with z latent variables.

The likelihood can also be intractable

$\Longrightarrow$???

Simulators

- **Deterministic evolution of initial state**
  - e.g. differential equations, fluid dynamics, N-body simulations…
- **Stochastic evolution**
  - e.g. Markov processes, molecular dynamics, stochastic differential equations…
- Integral over latent variables is typically intractable $p(x|\theta) = \int p(x, z|\theta)dz$



Slide: K. Cranmer

Latent vs. Nuisance

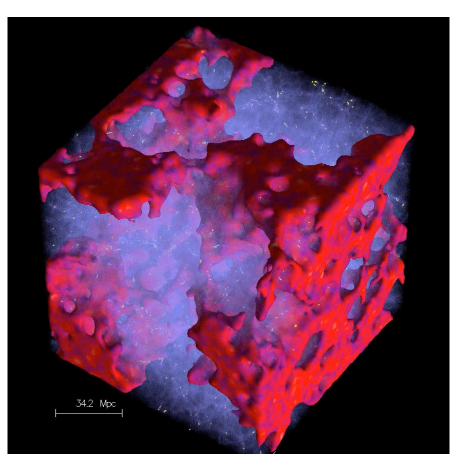
- Latent variables: unobserved "data" $p(x, \mathbf{z} \mid \theta)$
- Nuisance parameter: calibration, etc. $p(x \mid \theta, \eta)$
- Practically, the same consequence — need to integrate/ marginalize to get correct answer! This is often intractable.

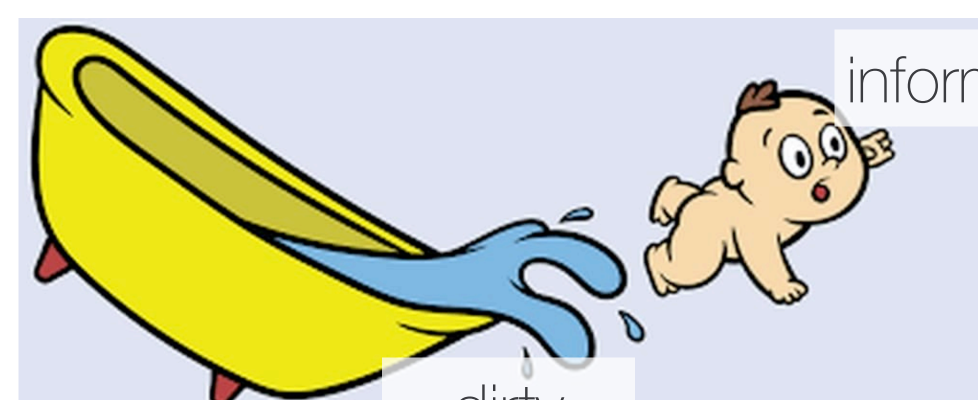
Now to formally state “the two problems of classical inference” …

Problem 1: intractable likelihood

- For most simulators, we **cannot evaluate the full likelihood**.
  - In cosmology: large-scale structure, 21-cm field, most late-time observations…
- Practitioners often **restrict** to theoretically controlled summary statistics such as the power spectrum at large scales.
  - We should worry that we're throwing the baby out with the bathwater.

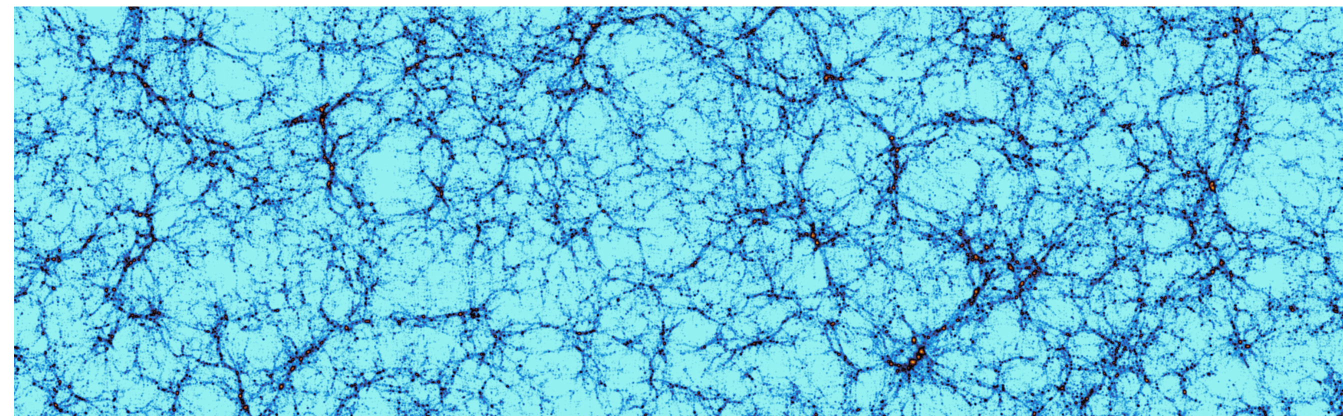
21cm field,
[SKA white paper 1210.0197]

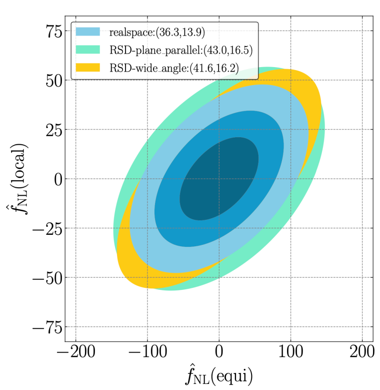




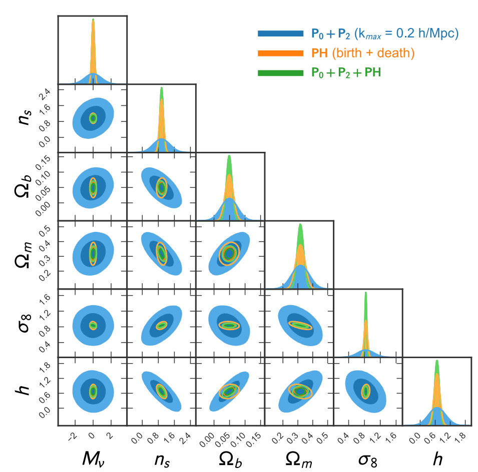
These problems clearly **demand more refined summary statistics**. One option is hand-crafted summaries, e.g. persistent homology for large-scale structure, whose **likelihoods can be approximated**. Would prefer more knobs to optimize, theoretical guarantees about saturating information content.

[Biagetti, AC, Shiu (JCAP) ’20;
AC, Biagetti, Shiu (NeurIPS wksp ’20)]



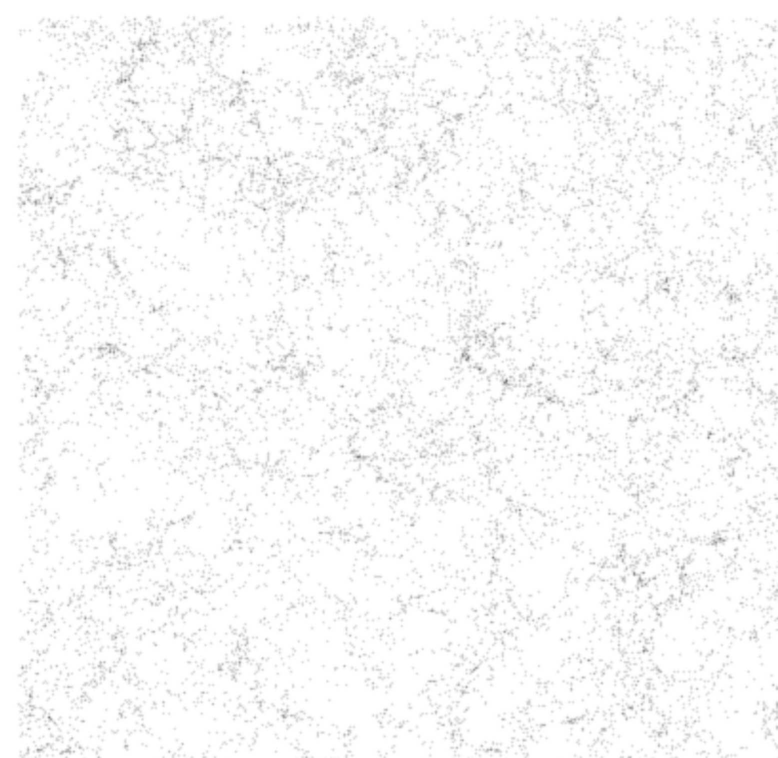


[Equilateral NG, 2203.08262]



[ΛCDM, to appear]

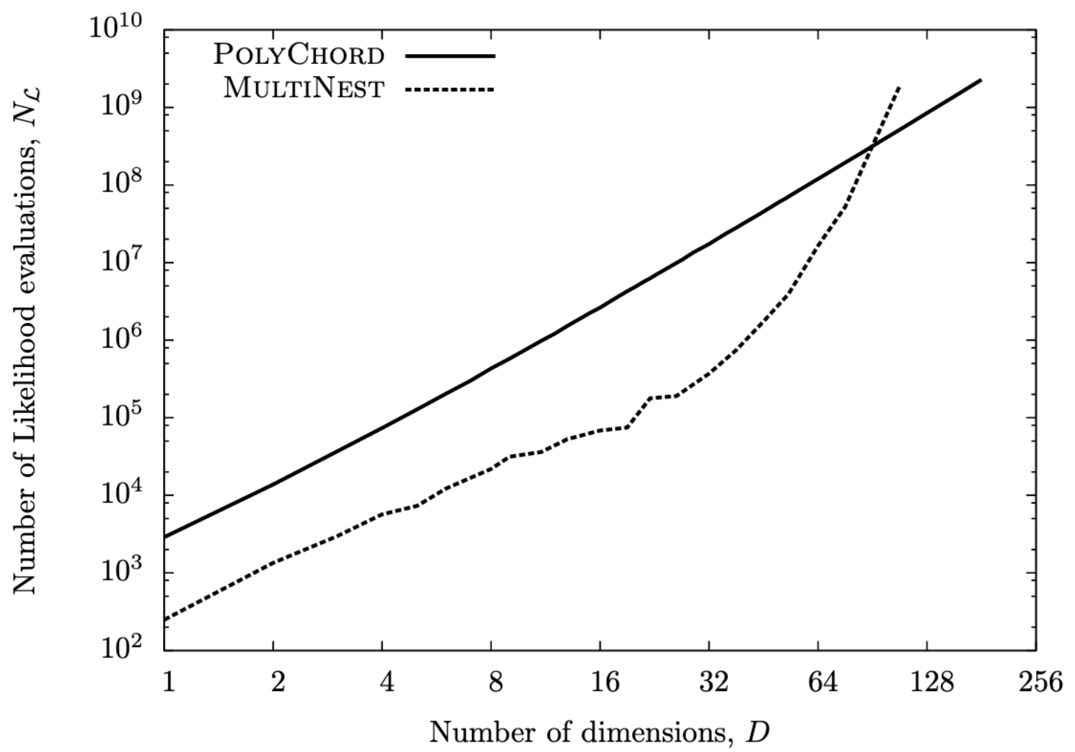
k=20, p=10, q=1, $\nu = +0.00$



Problem 2: scaling

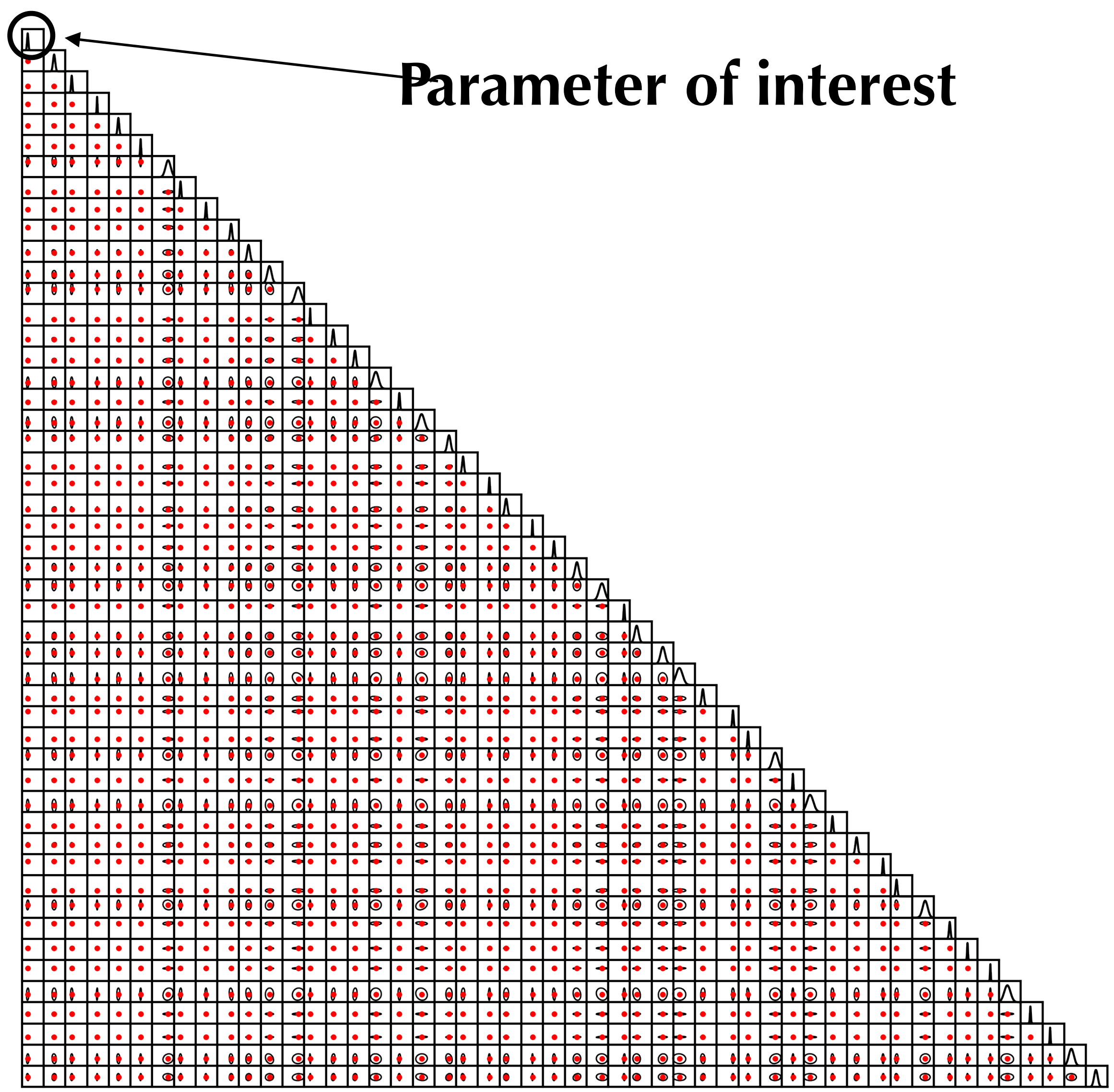
- Even if likelihood is known/ tractable:
  - For realistic inference, one must vary over instrumental calibration parameters, foreground residuals, latent variables…
  - **Sampling the joint** posterior scales poorly with parameter space dimension.

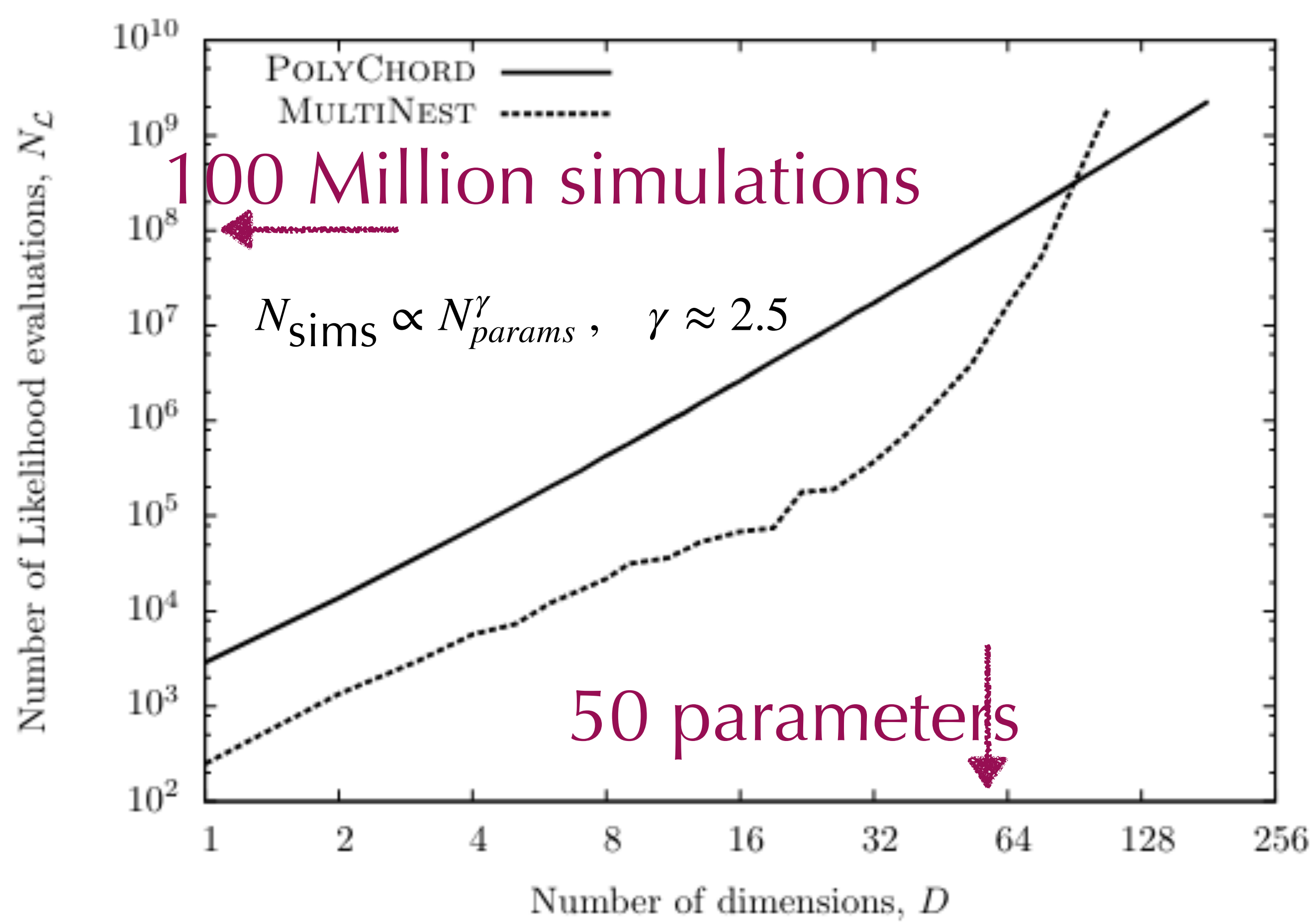
classical inference cost w/ dimension



[Handley et al. 1506.00171]

The curse of dimensionality





Feroz+ 0809.3437 (MultiNest)
Handley+ 1502.01856 (PolyChord)

slide: C. Weniger

There has to be a better way…

1. High-fidelity physics simulator
2. Deep learning
3. ???
4. Profit

2. Simulation-Based Inference

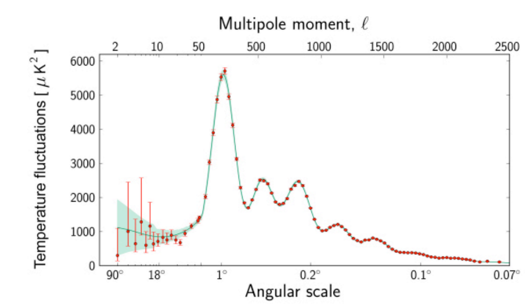
Simulators vs. Likelihoods

- Insight: running a **stochastic simulator** with input θ gives an output x that is drawn from an implicit likelihood $p(x \mid \theta)$

- "**Simulation-based inference**" or "likelihood-free inference" or "implicit likelihood inference" or … **[review: Cranmer, Brehmer, Louppe PNAS '20]**

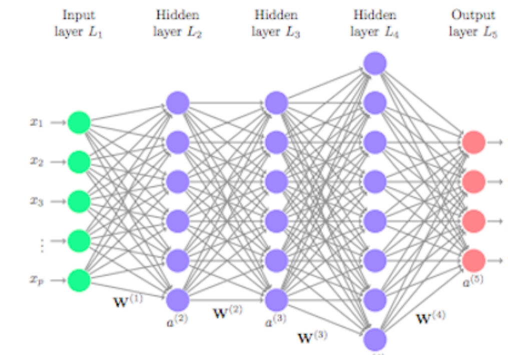
- Recent rapid progress thanks to **deep learning** algorithms **[Papamarkios et al. '19; Greenberg et al. '19; Hermans et al. '20; …]**.

θ



"simulator"

$$\theta \mapsto x$$



Neural X Estimation

- Developments use a **neural network** to approximate some quantity in Bayes' formula:

$$\boxed{p\left(\theta \mid x\right)} = \frac{p\left(x \mid \theta\right) p(\theta)}{p(x)}$$

- Neural Posterior Estimation **(NPE)**
- Neural Likelihood Estimation **(NLE)**
- Neural Ratio Estimation **(NRE)**

(Conditional) Density Estimation

cf. pydelfi **[Alsing et al. '18,'19]**
moment networks **[Jeffrey, Wandelt '20]**

- NLE and NPE both estimate normalized probability densities. Consequences:
  - **Restricted** network architecture: e.g. normalizing flow, mixture density model. Can be **difficult to train** **[Papamarkios et al. '21]**
  - For **high-dimensional data**, need a compression network.
- But: can be **good inductive bias**, especially if posterior or likelihood is "perturbation around Gaussian distribution"

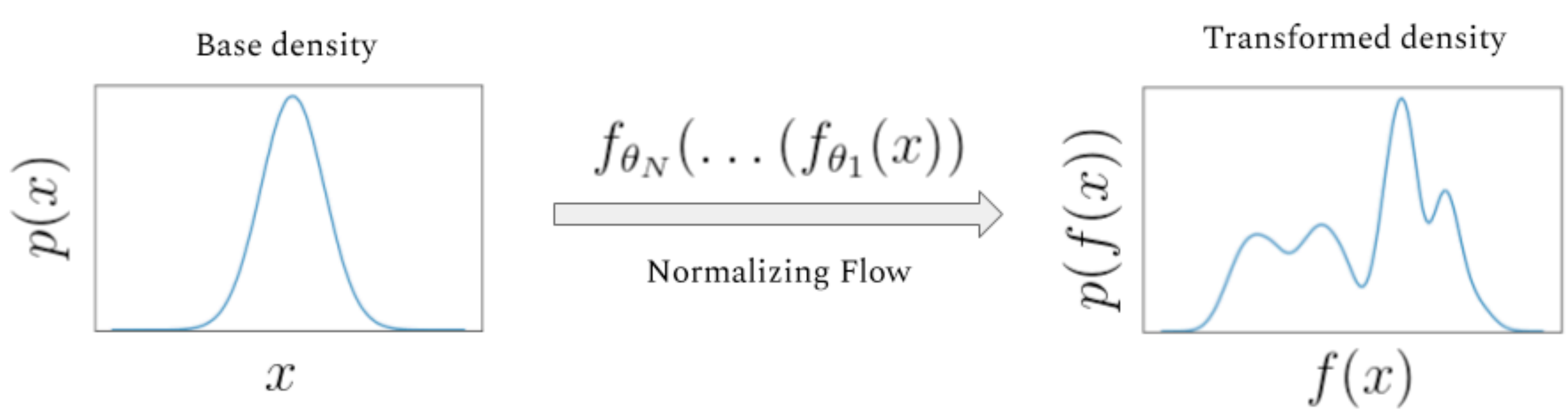
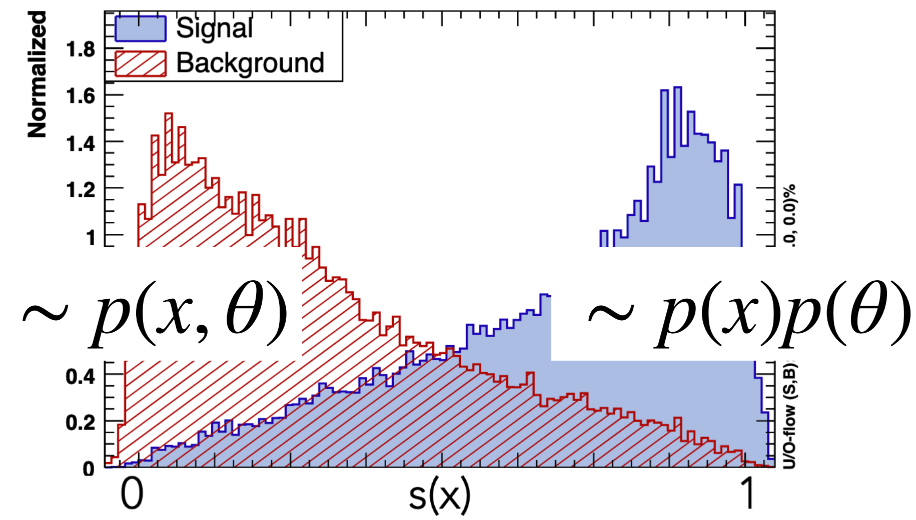


figure: https://gebob19.github.io/normalizing-flows/

Ratio Estimation

- Ratio estimation is qualitatively different.

- Train a **classifier** to distinguish data-parameter pairs (x, θ) jointly drawn $\sim p(x, \theta)$ (label $y = 1$) from marginally drawn $\sim p(x)p(\theta)$ (label $y = 0$)



likelihood-to-evidence ratio

classifier

$$\boxed{r(x, \theta)} \equiv \frac{p(\boldsymbol{x}, \boldsymbol{\theta})}{p(\boldsymbol{x})p(\boldsymbol{\theta})} = \frac{\tilde{p}(\boldsymbol{x}, \boldsymbol{\theta} \mid y = 1)}{\tilde{p}(\boldsymbol{x}, \boldsymbol{\theta} \mid y = 0)} = \frac{\boxed{\tilde{p}(y = 1 \mid \boldsymbol{x}, \boldsymbol{\theta})}}{1 - \tilde{p}(y = 1 \mid \boldsymbol{x}, \boldsymbol{\theta})}.$$

[Hastie et al., 2001; Sugiyama et al., 2012; Cranmer et. al., 2015]

Ratio Estimation

- Intuitive picture: given two probability distributions $q_1(x),\ q_2(x)$, best guess for whether x came from q_1 or q_2 is closely related to the probability ratio

$$\frac{q_1(x)}{q_2(x)}$$

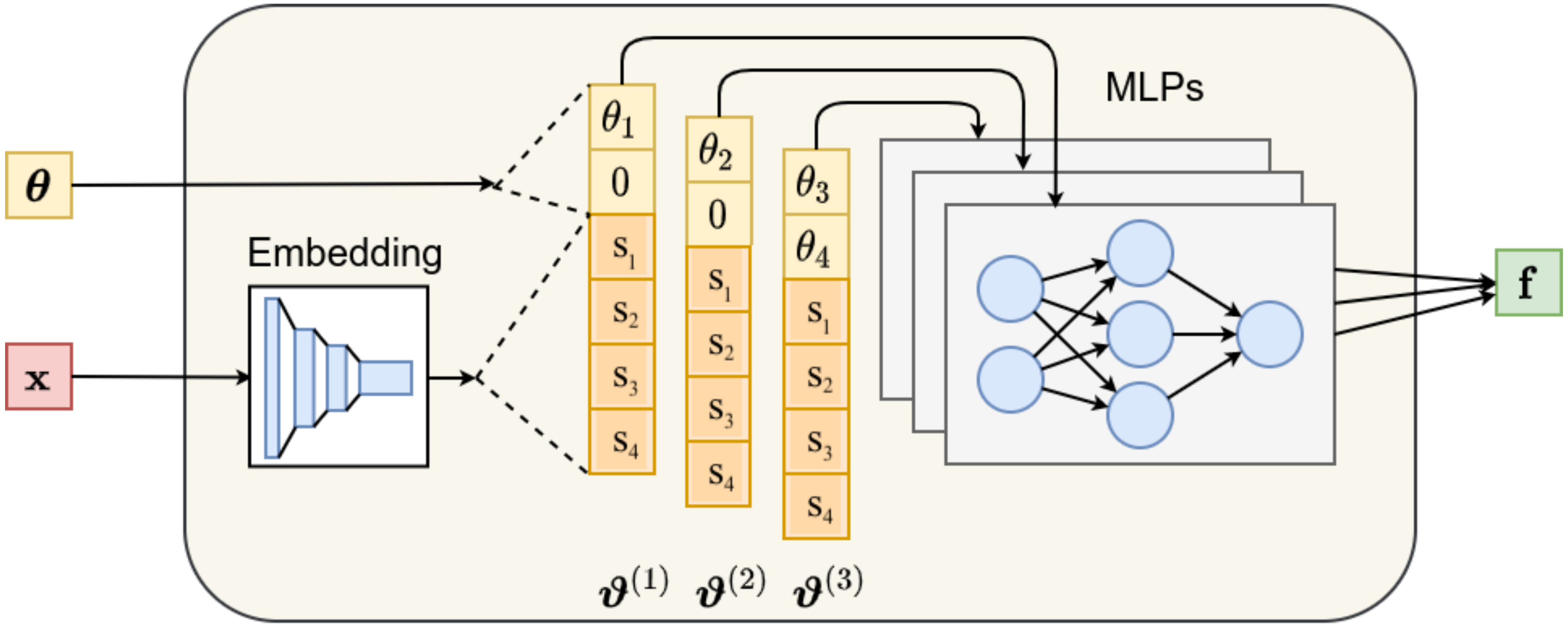
- Now take $q_1 = p(x, \theta),\ q_2 = p(x)p(\theta)$. Your classifier has learned the likelihood-to-evidence ratio!

Vapnik's principle: "When solving a problem of interest, do not solve a more general problem as an intermediate step."

$$q_1(x), q_2(x) \;\underset{\not\longleftarrow}{\longrightarrow}\; \frac{q_1(x)}{q_2(x)}$$

Ratio Estimation

- Classifiers are very flexible in network architecture. Training is also simple.
- We still find it useful to use a "compression" or "embedding" network, which turns complex data $\boldsymbol{x}$ into features $\boldsymbol{s}$.



NB: this picture actually shows *marginal* ratio estimation, wait for next slide

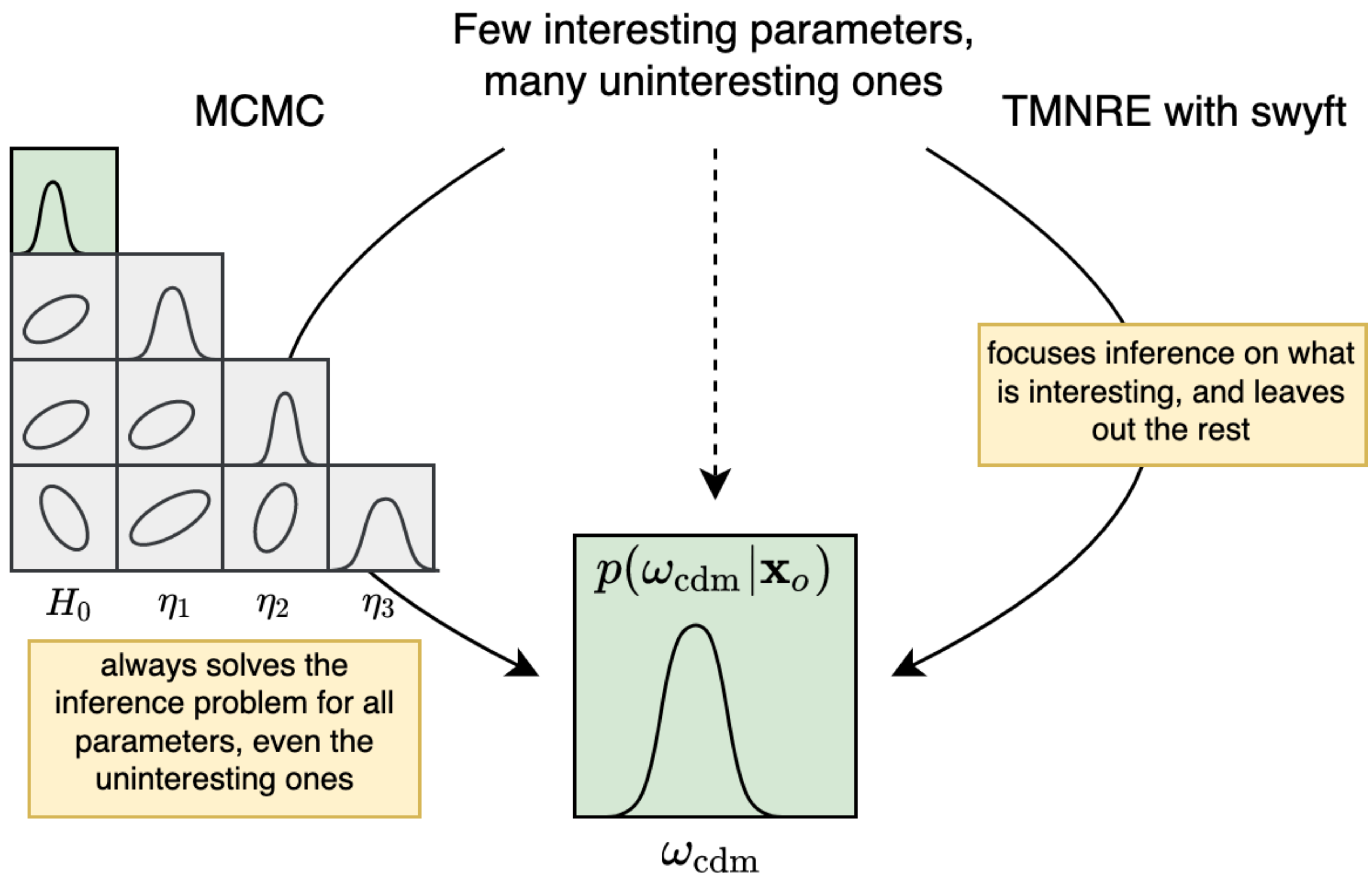
Sidebar: Marginal Estimation

- With neural methods, *automatic marginalization* is possible. [Alsing,Wandelt ’19;Hermans et al. ’19; Miller et al. ’20; Jeffrey,Wandelt ‘20]
- For comparison of various performances, see [Miller et al. ‘21]
- For example, we **define the marginal ratio**

$$r(\vartheta, x) \equiv \frac{p(x \mid \vartheta)}{p(x)} = \frac{\int d\eta \; p(x \mid \vartheta, \eta) p(\eta)}{p(x)}$$

which can be directly trained by omitting η from the information given to the classifier. We train an individual network for each marginal ratio.

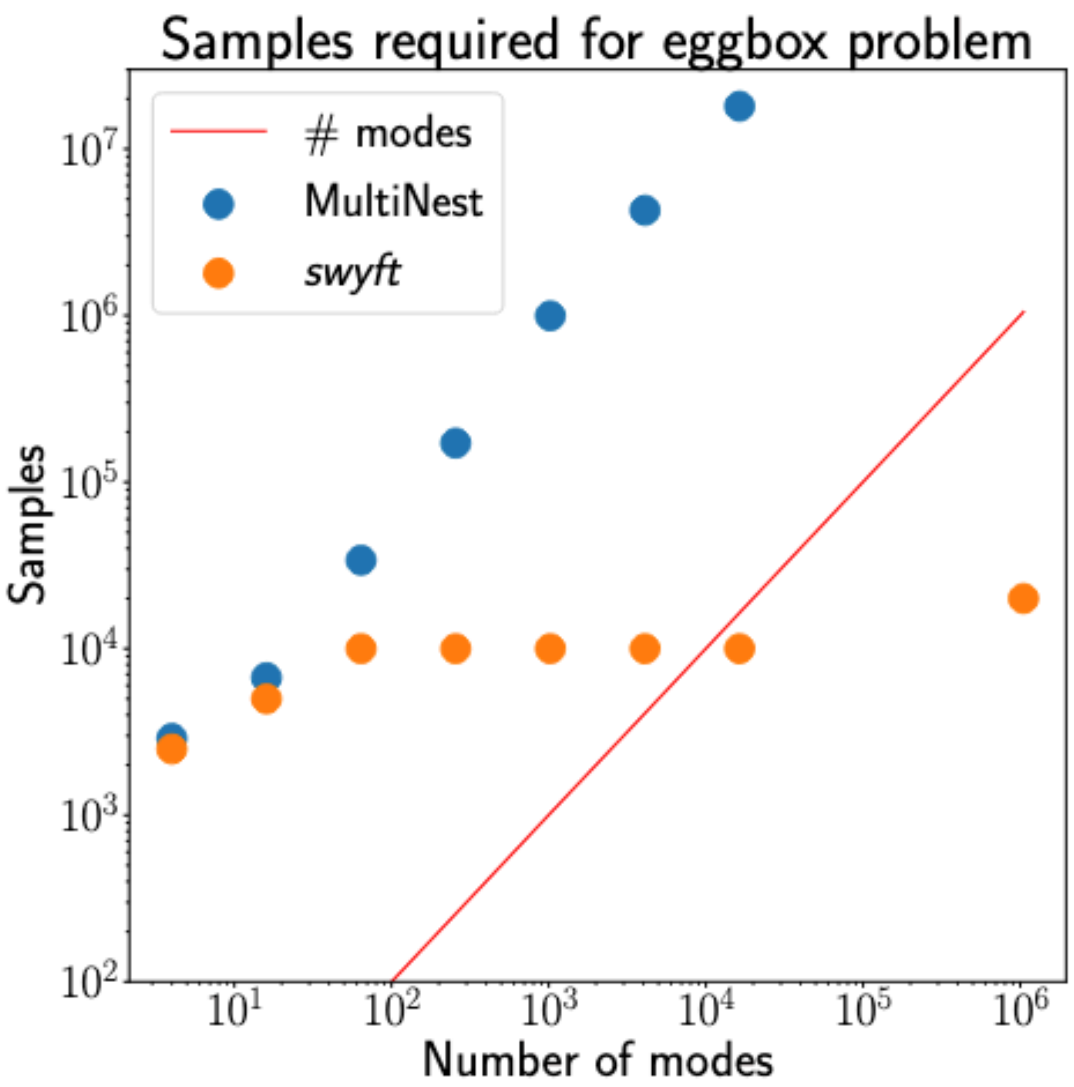
Marginal Neural Ratio Estimation

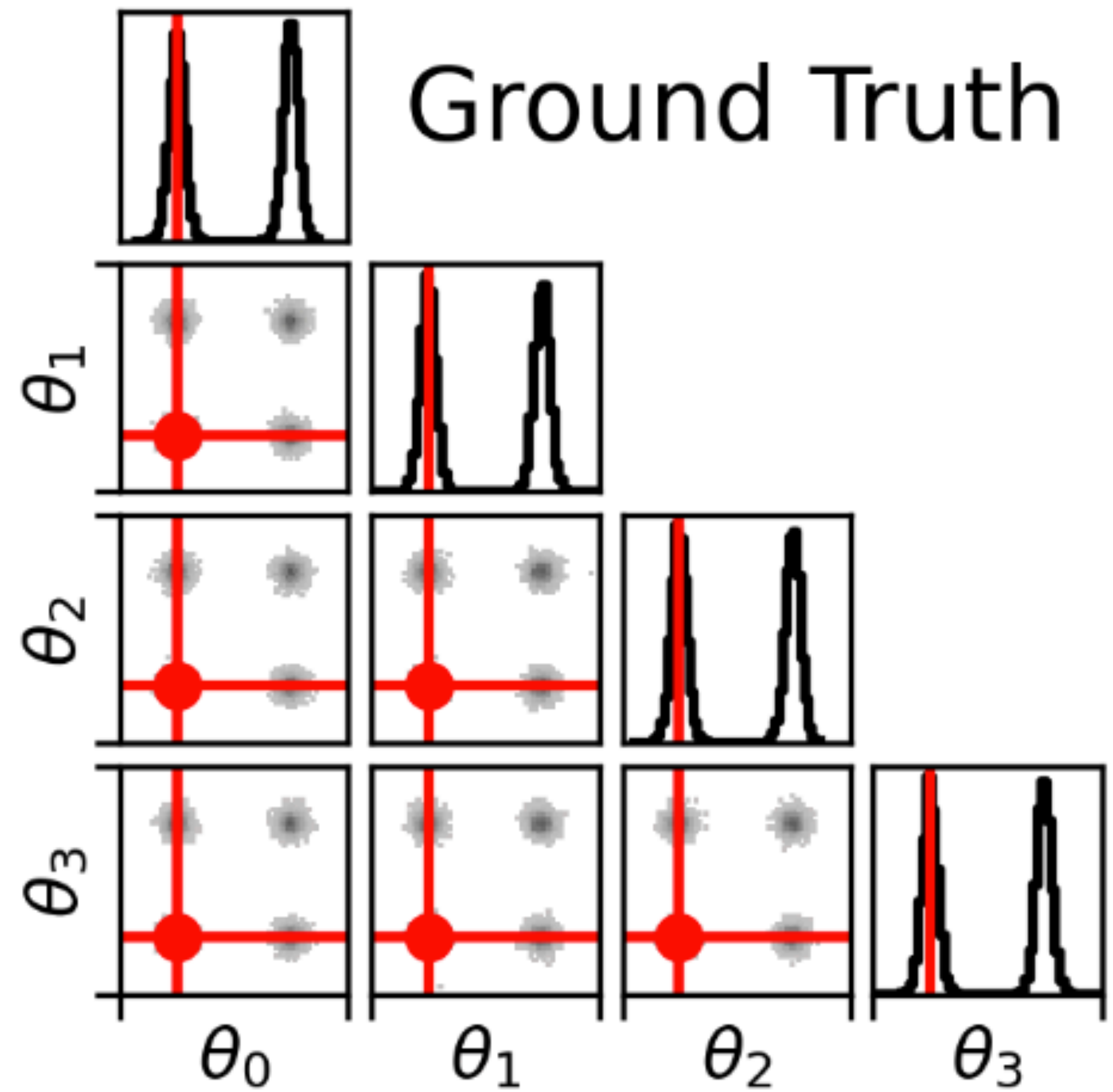


Vapnik's principle pt. 2

Some benefits of automatic marginalization

[Miller et al. ’20; Miller et al. ’21]





Sequential methods/ active learning

Sequential Methods

- **Sequential** Neural X Estimation: use proposal density to select relevant simulations for training:
  - Current posterior estimator
  - Bayesian optimization: balance between hunting for best-fit and reducing uncertainty in results.
- Note: definition of marginal X means nuisance parameters must be sampled from prior!

Sequential Neural Likelihood: Fast Likelihood-free Inference with Autoregressive Flows

George Papamakarios, University of Edinburgh — David C. Sterratt, University of Edinburgh — Iain Murray, University of Edinburgh

Automatic Posterior Transformation for Likelihood-free Inference

David S. Greenberg[1] Marcel Nonnenmacher[1] Jakob H. Macke[1]

Likelihood-free MCMC with Amortized Approximate Ratio Estimators

Joeri Hermans[1] Volodimir Begy[2] Gilles Louppe[1]

On Contrastive Learning for Likelihood-free Inference

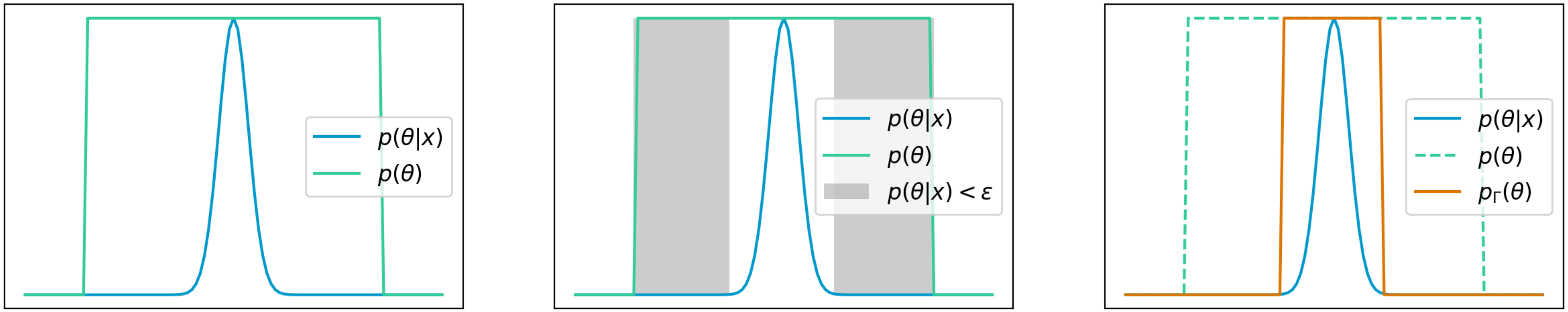
Conor Durkan[1] Iain Murray[1] George Papamakarios[2]

Sequential Neural Posterior Estimation (SNPE, Papamakarios & Murray, 2016; Lueckmann et al., 2017; Greenberg et al., 2019),

“Truncated Marginal
Neural Ratio Estimation”
(TMNRE)

Truncation

[Miller et al. ’20; Miller et al. ’21]



- Sometimes priors are much wider than posteriors. Let’s call the relevant region of parameter space Γ.

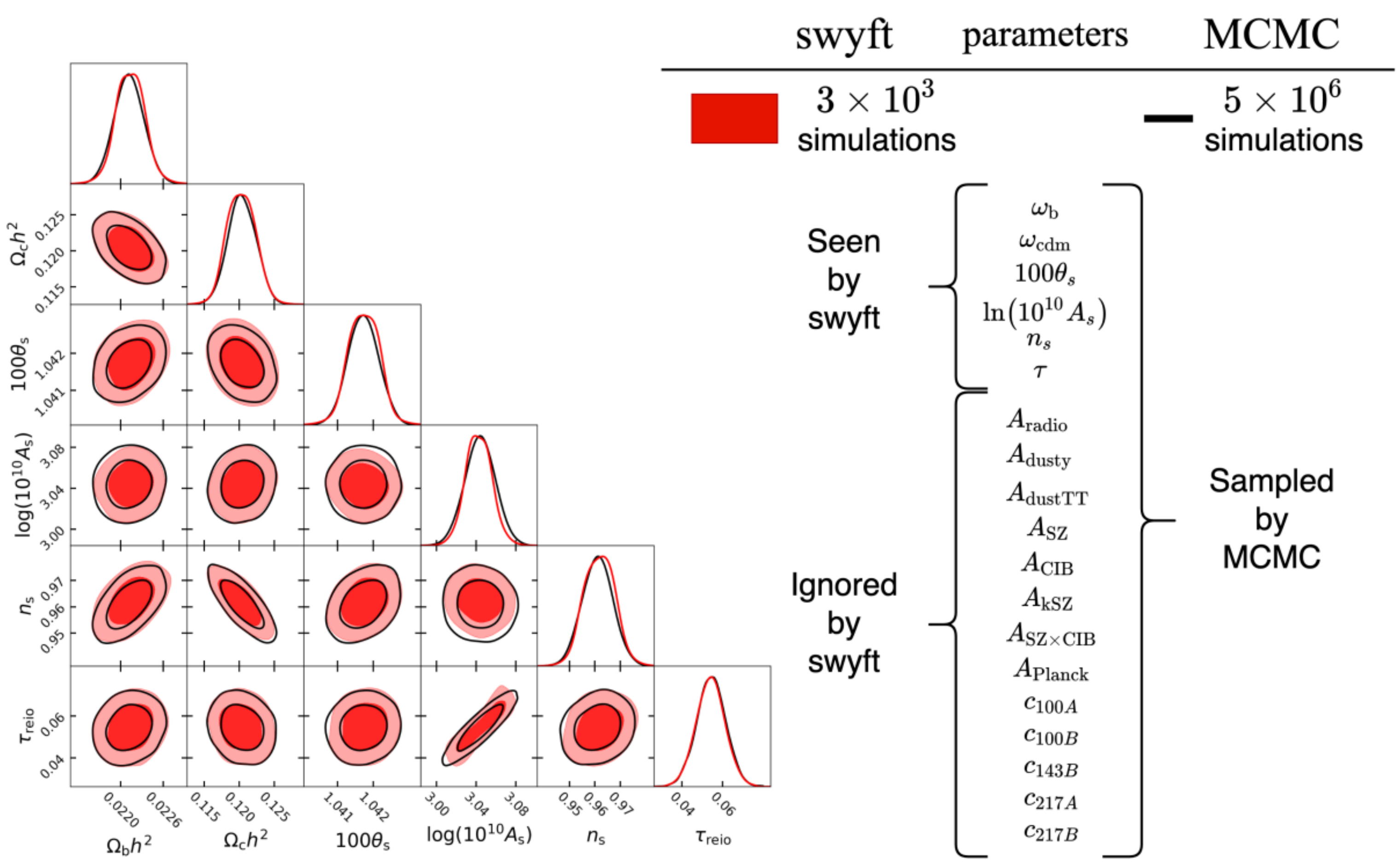
- We **zoom into the relevant region** by approximating Γ (requiring $\hat{p}(\theta\,|\,x) > \epsilon$) in a series of rounds.

- With marginal posteriors, Γ is approximated via a product of low-dimensional projections. These can reflect expected correlations.

Applications

Example- CMB PS cosmology

We can reproduce MCMC results with 3 orders of magnitude fewer simulator runs



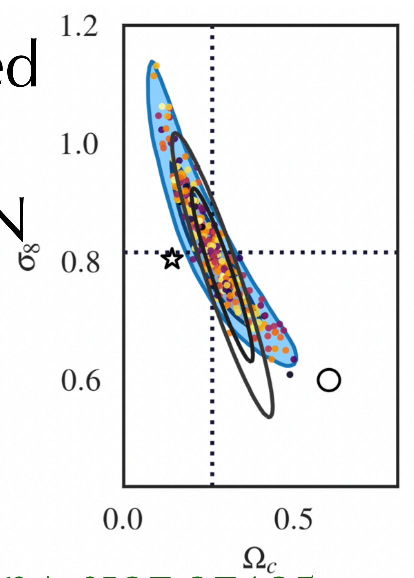
Alternative to:
Long MCMC waiting times

SWYFT
Scalable Inference

[AC et al. '21 (JCAP)]

Example - LSS and 21cm cosmology

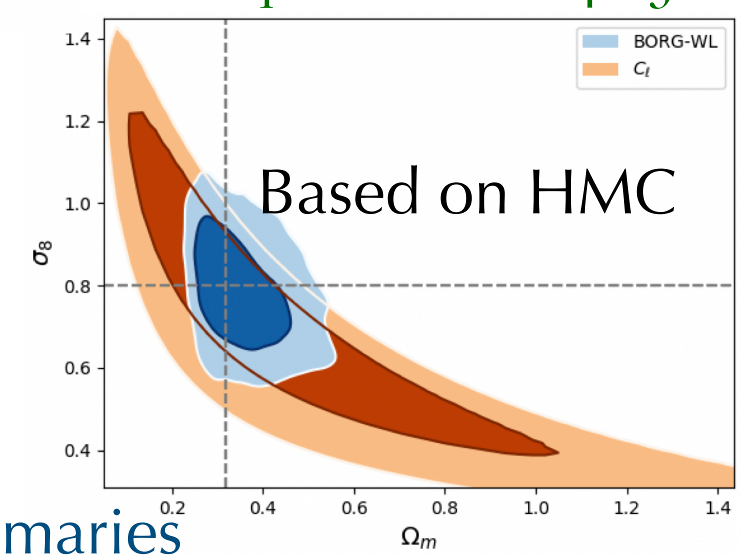
Based on CNN



Makinen+ 2107.07405

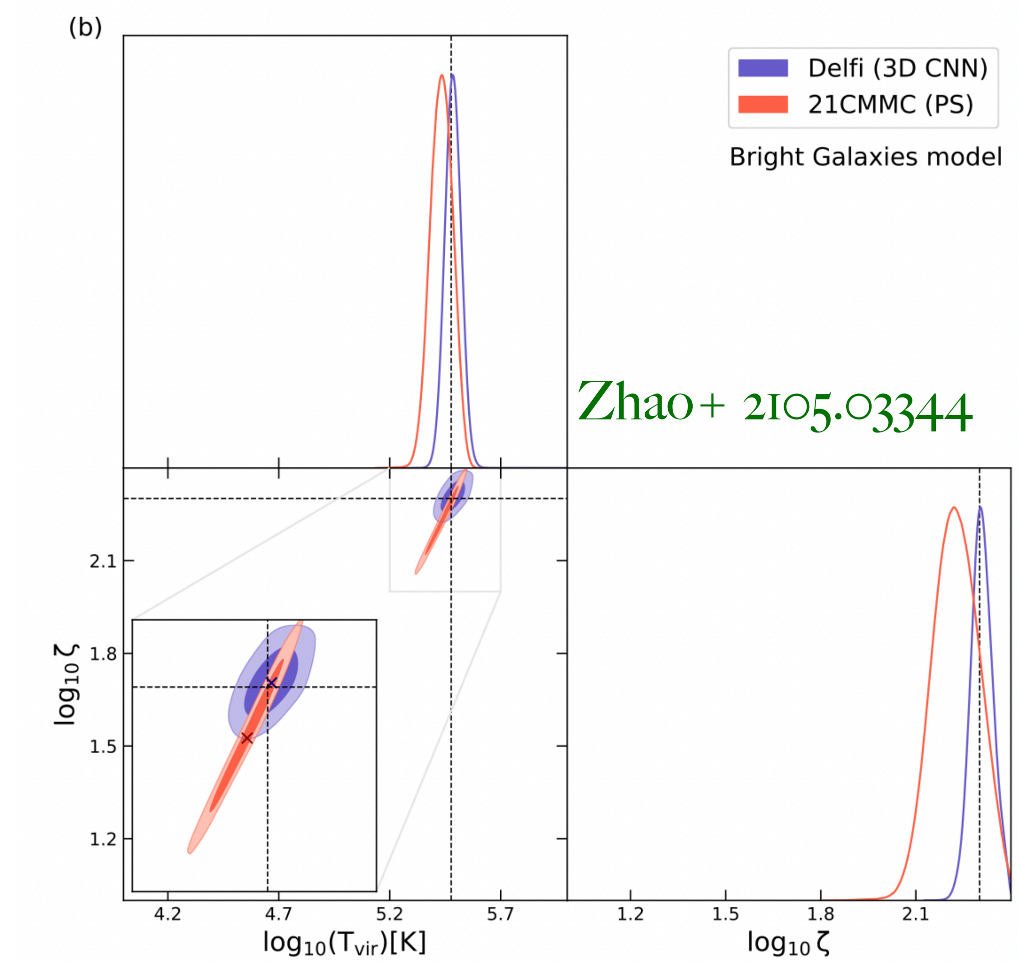
Breaking degeneracy between DM density and power-spectrum amplitude

Porqueres+ 2108.04825



Based on HMC

Alternative to:
Hand-crafted summaries



Zhao+ 2105.03344

Breaking degeneracy between ionisation parameters T_{vir} and ζ

Example - Strong lensing

SWYFT Scalable Inference

Searching light DM halos

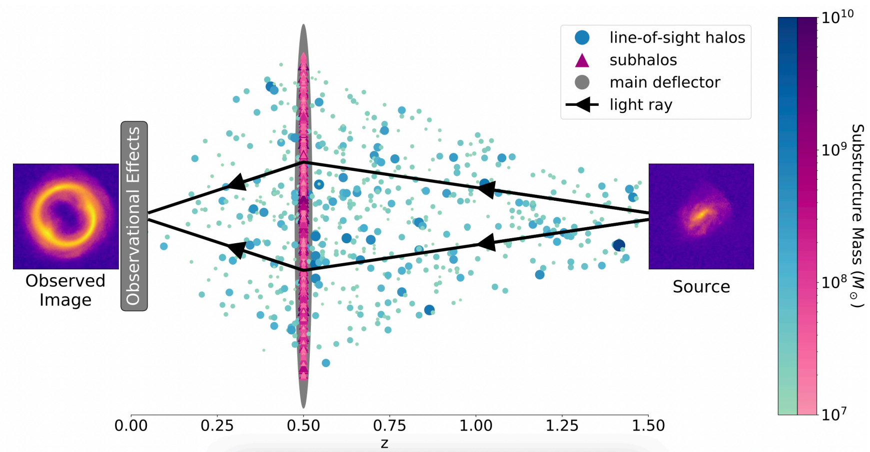
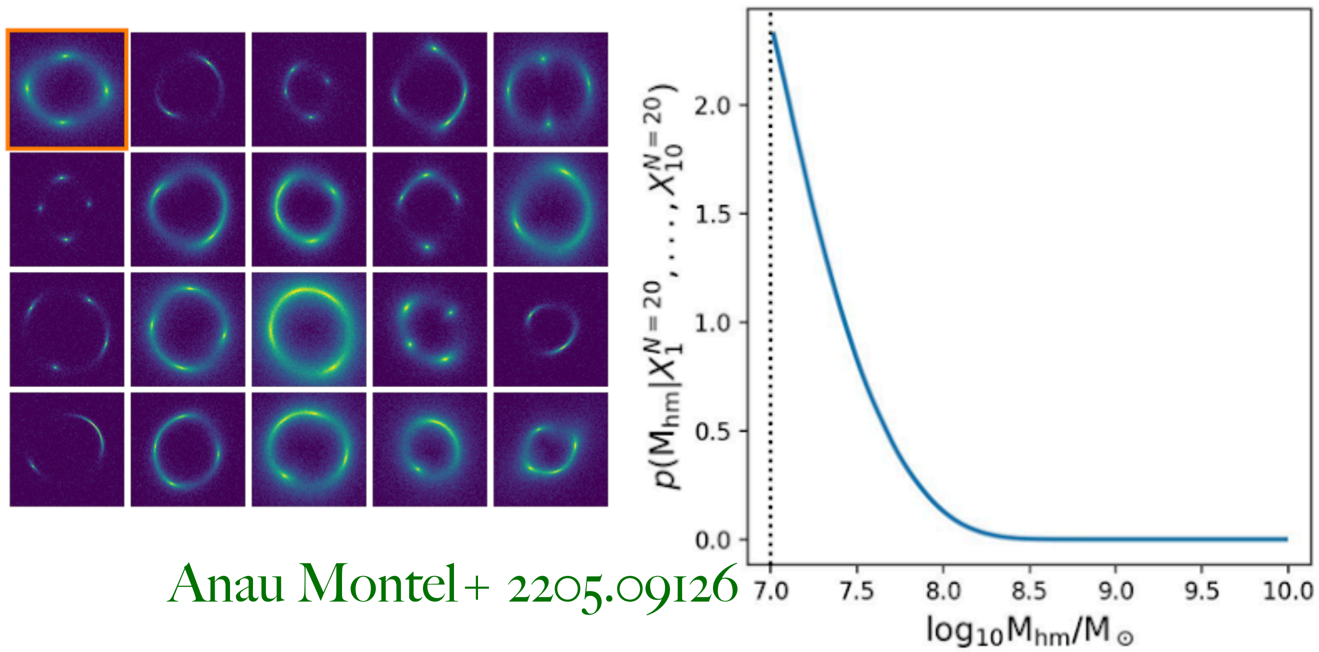


Image credit: Wagner-Carena+ 2203.00690

Halo mass function cutoff

Probing **population effects of light dark matter halos** rather than individual detections



Anau Montel+ 2205.09126

Alternative to:
HMC, parameter reduction, ABC, …

Related work: He+ 2010.13221 (similar in spirit, using ABC)

Wagner-Carena+ 2203.00690 (constraining subhalo mass function normalization)

Example — foreground removal

Single frequency CMB B-mode inference with realistic foregrounds from a single training image

Niall Jeffrey,[1,2★] François Boulanger,[1] Benjamin D. Wandelt,[3,4] Bruno Regaldo-Saint Blancard,[1,5]
Erwan Allys,[1] François Levrier[1]

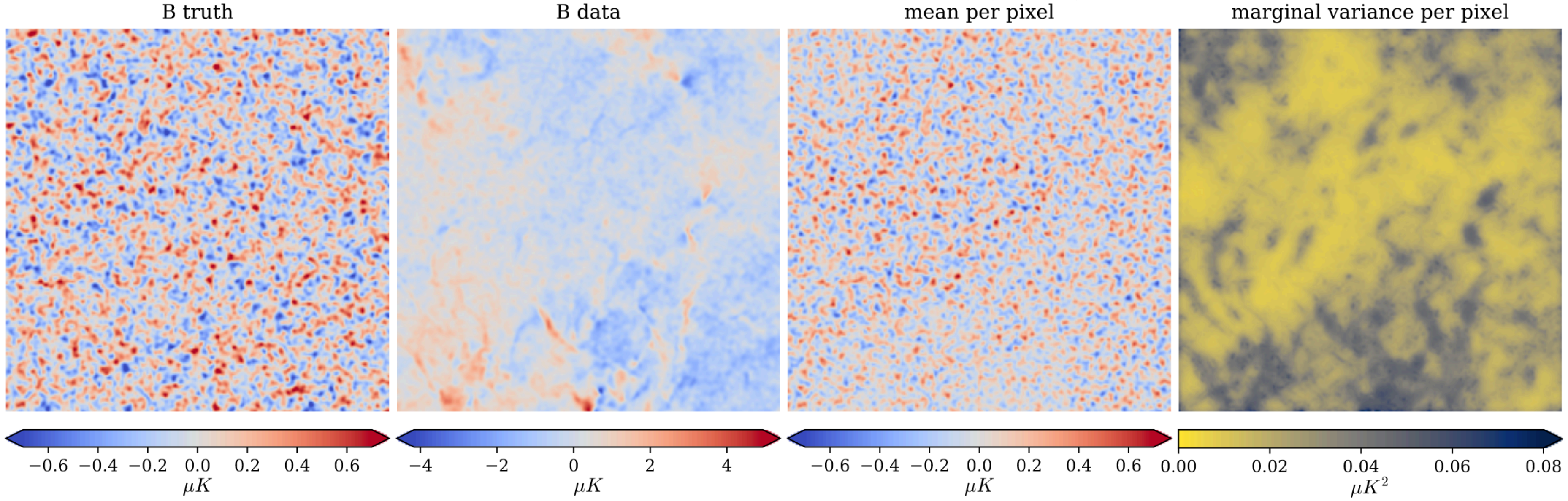
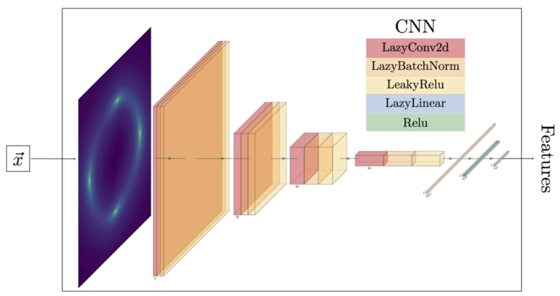


Figure 1. The two left panels show the simulated clean signal $\mathbf{s}_B$ and the foreground-contaminated data $\mathbf{d}_B$ (validation data A - section 3.3). The centre right panel shows the mean of the marginal posterior probability per pixel $\mathcal{F}(\mathbf{d}_E, \mathbf{d}_B)$ and the far right shows the variance of the marginal posterior per pixel $\mathcal{G}(\mathbf{d}_E, \mathbf{d}_B)$. This CMB signal has been inferred (the posterior probability estimated) using only a single frequency and a single training image. Patches of reduced power in the pixel posterior mean are not artefacts; the mean is expected to move closer to 0 μK when the posterior variance is higher.

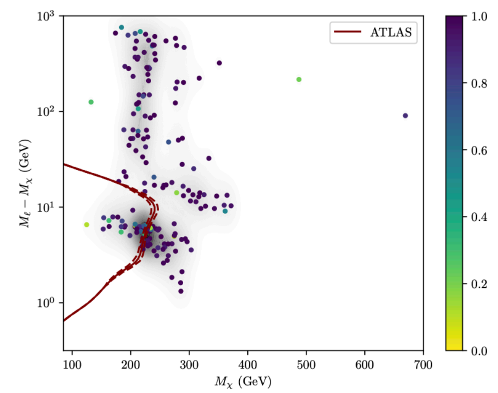
- exploit "moment network" — directly target marginal mean/variance **[Jeffrey, Wandelt]**

More examples



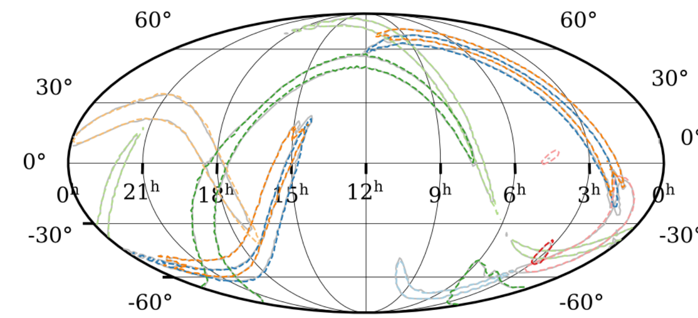
Strong lensing

Brehmer+ 1909.02005, Coogan+ 2010.07032, Legin+ 2112.05278, Wagner-Carena+ 2203.00690, Anau Montel+ 2205.09126, Coogan+ 2207.xxxxx



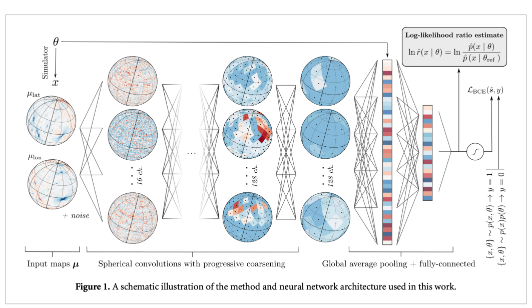
Effective field theory

Morrison+ 2203.13403



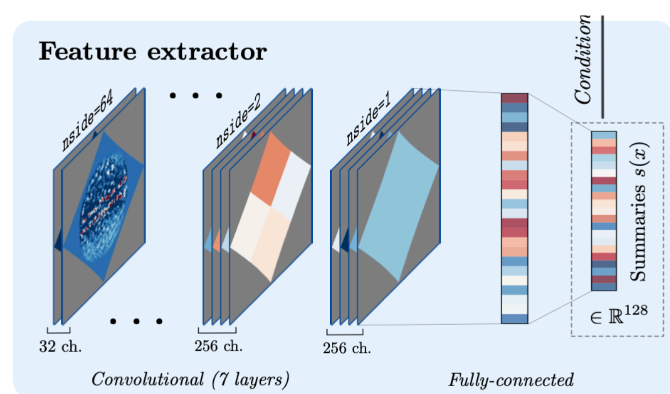
GW parameters

Delaunoy+ 2010.12931, Dax+ 2106.12594, ...



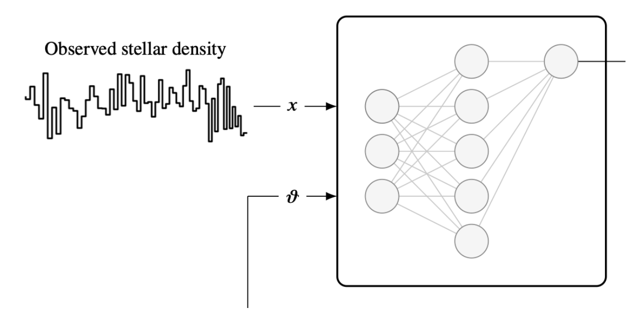
Astrometry

Mishra-Sharma+ 2110.01620



Fermi GeV excess

Mishra-Sharma+ 2110.06931



Stellar streams

Hermans+ 2011.14923

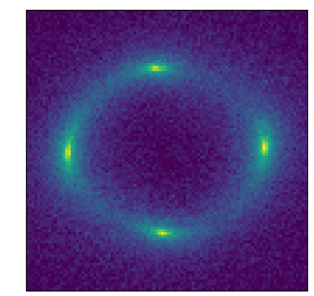
Truncation: Strong lensing

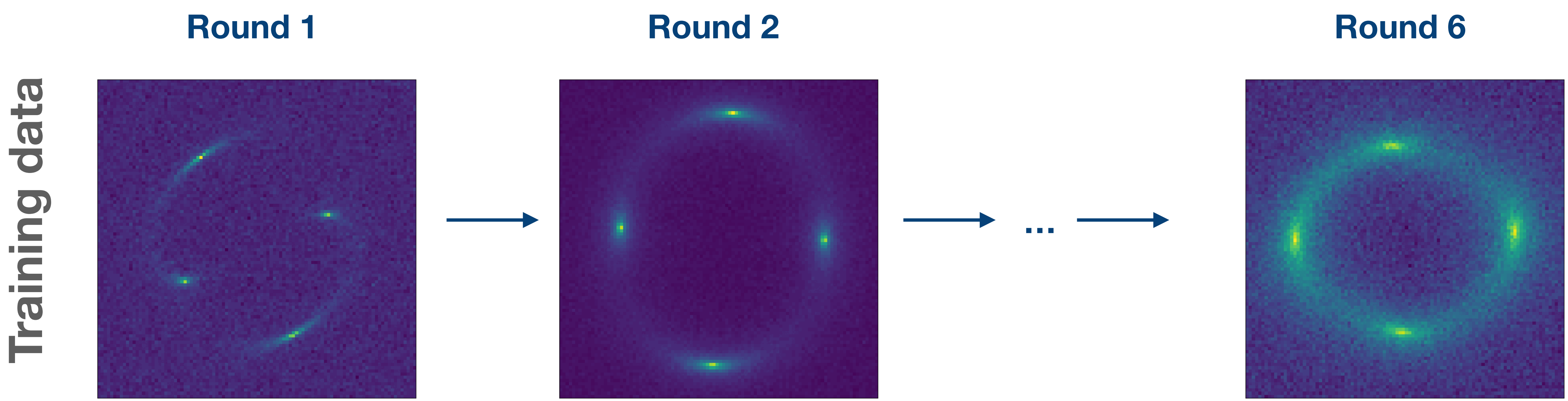
Truncation **focuses training data generation** in the regions of the parameter space most relevant for analysing a particular observation.

Algorithm: "Truncated Marginal Neural Ratio Estimation" (TMNRE)

Miller+ 2107.01214 (truncated priors)

Target mock observation





Software and Benchmarking

Software

The Journal of Open Source Software

sbi

Search

mackelab/sbi
v0.19.2 325 74

sbi
Home
Installation
Tutorials and Examples
Contribute
API Reference
FAQ
Credits

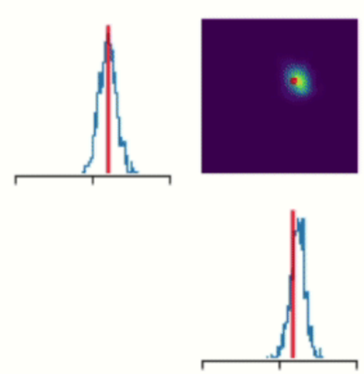
`sbi` : simulation-based inference

`sbi` : A Python toolbox for simulation-based inference.

```
prior = BoxUniform(low=zeros(2), high=2*ones(2))    # Box prior [0,2]x[0,2]
def simulator(θ): return θ + 0.1*randn_like(θ)      # Gaussian in 2D
posterior = infer(simulator, prior, method='SNPE', num_simulations=500)
```

```
Running 500 simulations.: 100%|██████████| 500/500 [00:00<00:00, 57141.55it/s]
Neural network successfully converged after 109 epochs.
```

```
samples = posterior.sample((1000,), x=observed)
pairplot(samples, points=ground_truth, **plot_style);
```



Inference can be run in a single line of code:

```
posterior = infer(simulator, prior, method='SNPE', num_simulations=1000)
```

Table of contents

https://www.mackelab.org/sbi

Software

SWYFT
Scalable Inference

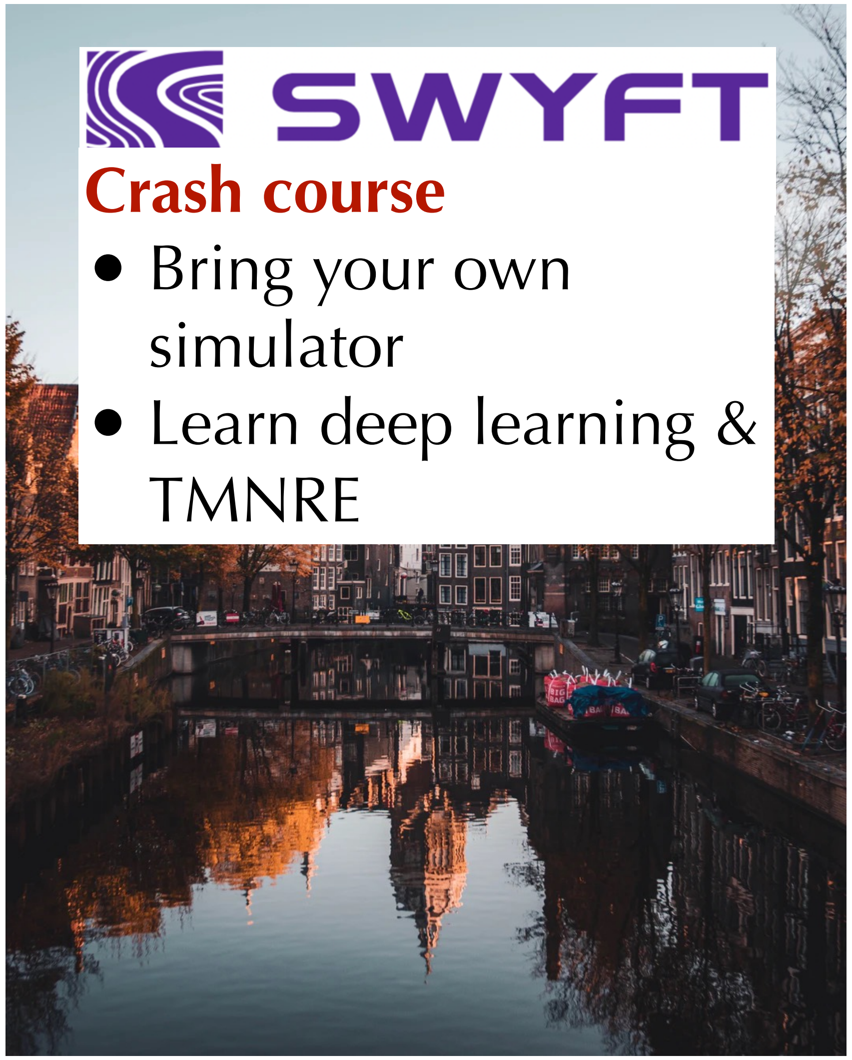
JOSS
The Journal of Open Source Software

swyft: Truncated Marginal Neural Ratio Estimation in Python

Benjamin Kurt Miller[1,2,3], **Alex Cole**[1], **Christoph Weniger**[1], **Francesco Nattino**[4], **Ou Ku**[4], **and Meiert W. Grootes**[4]

1 Gravitation Astroparticle Physics Amsterdam (GRAPPA), University of Amsterdam, Science Park 904, 1098 XH Amsterdam 2 Amsterdam Machine Learning Lab (AMLab), University of Amsterdam, Science Park 904, 1098 XH Amsterdam 3 AI4Science Lab, University of Amsterdam, Science Park 904, 1098 XH Amsterdam 4 Netherlands eScience Center, Science Park 140, 1098 XG Amsterdam, The Netherlands

- A python library built on pytorch/ lightning
- "Official" implementation of **Truncated Marginal Neural Ratio Estimation (TMNRE)** algorithm
- Makes it simple to estimate marginal posteriors for very high dimensional models
- https://github.com/undark-lab/swyft



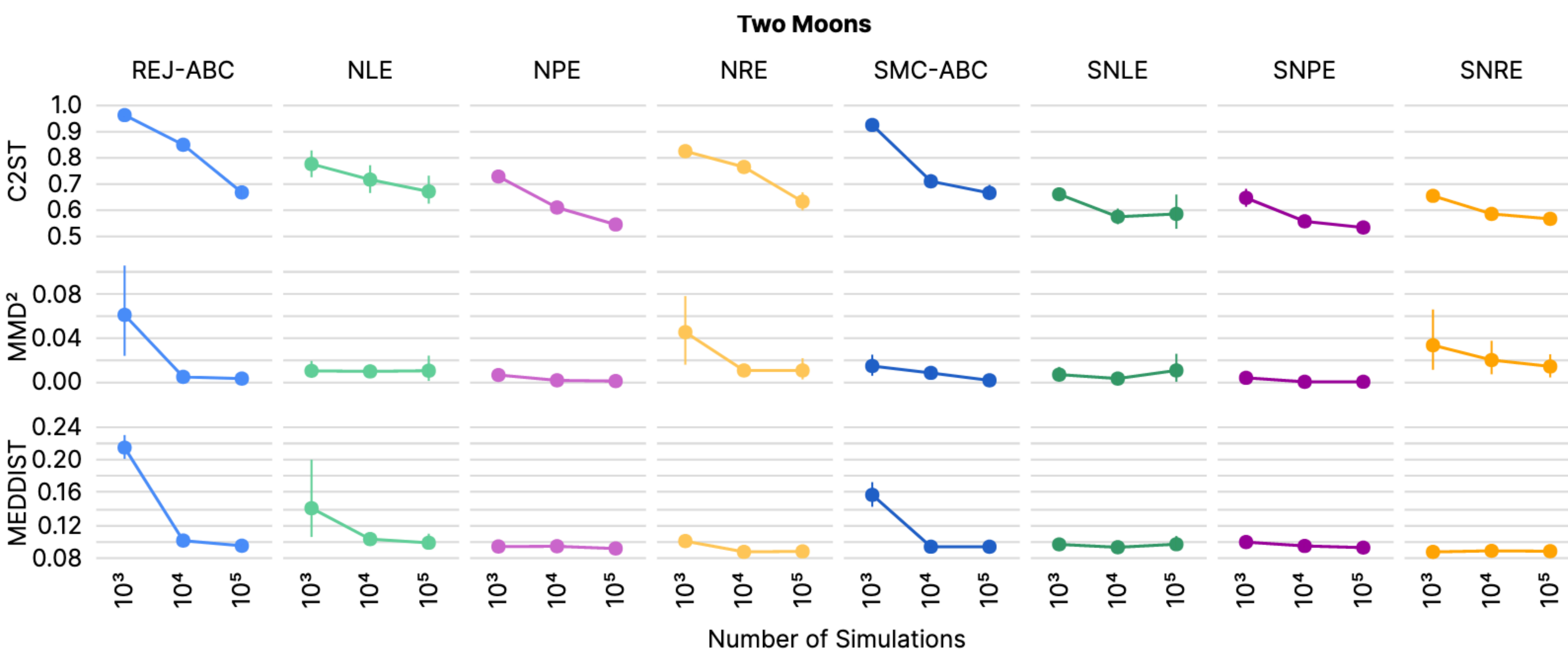
Amsterdam, Nov 2022

- Sign up to the Email list: shorturl.at/cdfw3

Benchmarking Simulation-Based Inference

Jan-Matthis Lueckmann[1,2] **Jan Boelts**[2] **David S. Greenberg**[2,3]
Pedro J. Gonçalves[4] **Jakob H. Macke**[1,2,5]

[1]University of Tübingen [2]Technical University of Munich [3]Helmholtz Centre Geesthacht
[4]Research Center caesar [5]Max Planck Institute for Intelligent Systems, Tübingen



Demo?

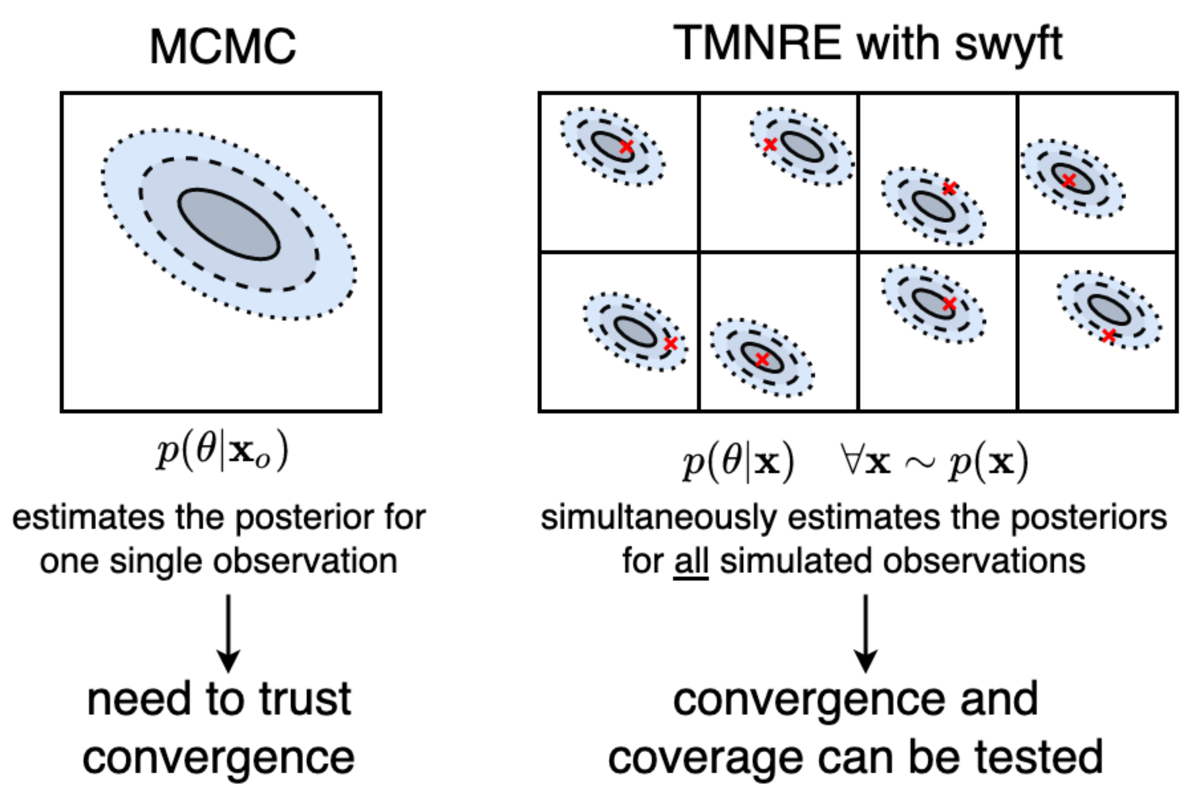
https://github.com/undark-lab/swyft/blob/master/notebooks/Examples%20-%201.%20Custom%20networks.ipynb

Consistency

Amortization and Consistency

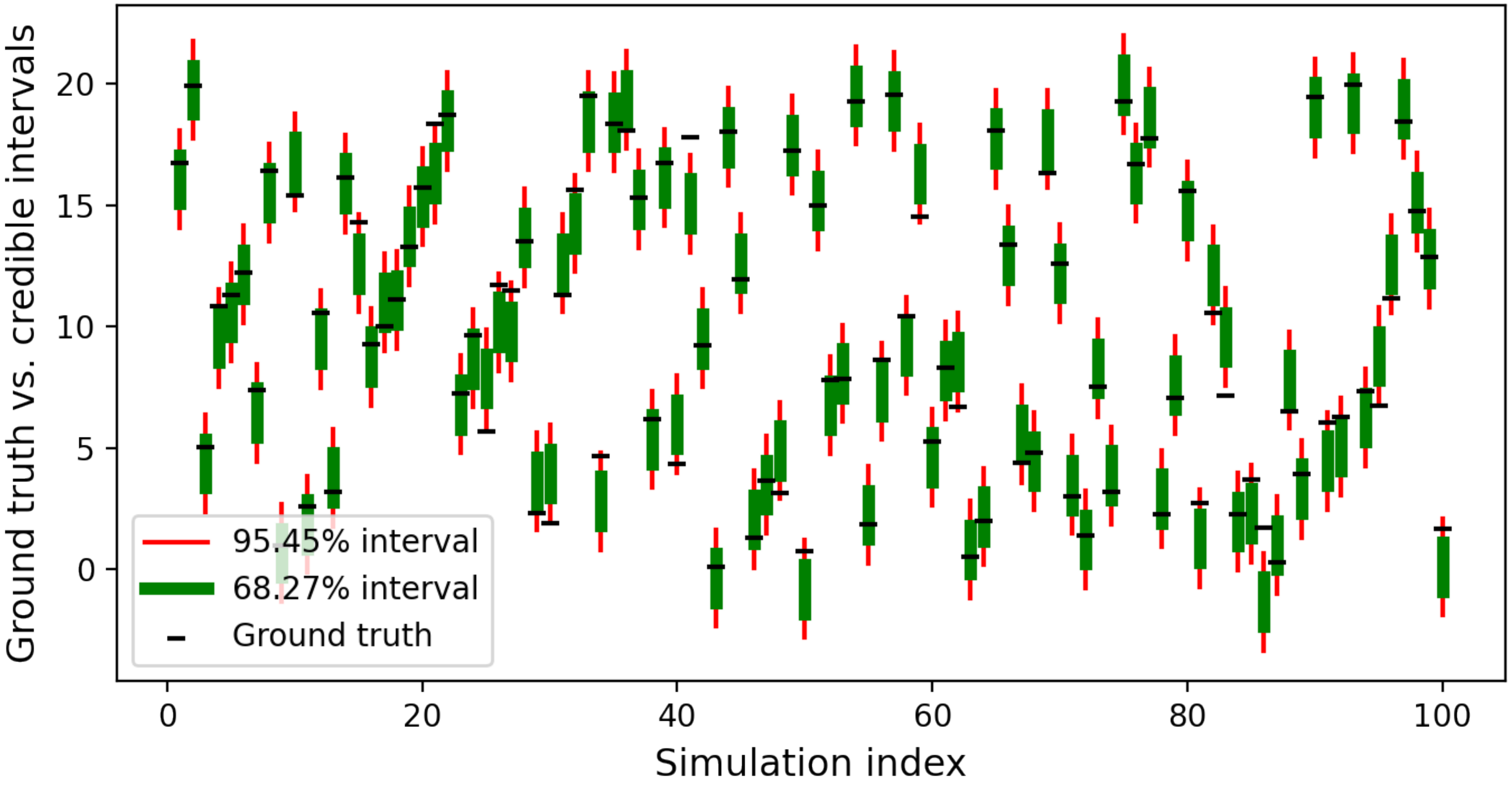
- Once trained, our network can rapidly generate posteriors for *any data* drawn from $p(x\,|\,\theta)p_\Gamma(\theta)$. Called **“amortization.”**

- This enables **rapid tests of statistical consistency** that are not possible with sampling-based methods.



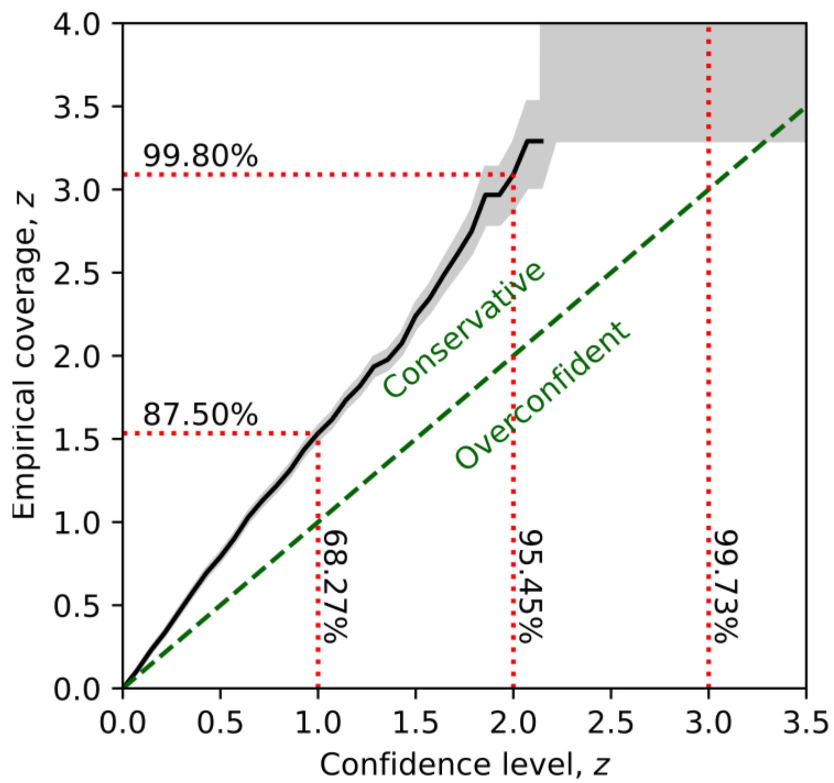
Amortization and Consistency

- We can therefore draw many samples from our simulation bank, generate posteriors, and see **how often the true parameters lie within the $N\,\%$ credible region.**

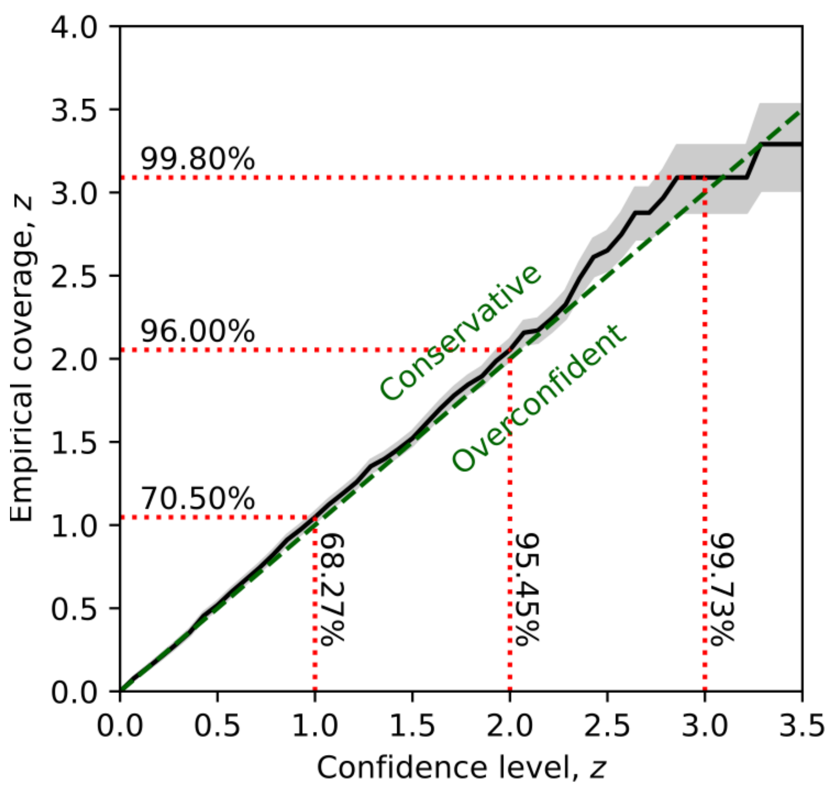


Amortization and Consistency

- We compare the network's predictions to the empirical coverage to **assess convergence** and ensure our network is not overconfident.
- This consistency test makes no reference to likelihoods or the true parameters of observed data.



still converging…



converged!

Averting A Crisis In Simulation-Based Inference

Joeri Hermans*
University of Liège
joeri.hermans@doct.uliege.be

Arnaud Delaunoy*
University of Liège
a.delaunoy@uliege.be

François Rozet
University of Liège
francois.rozet@uliege.be

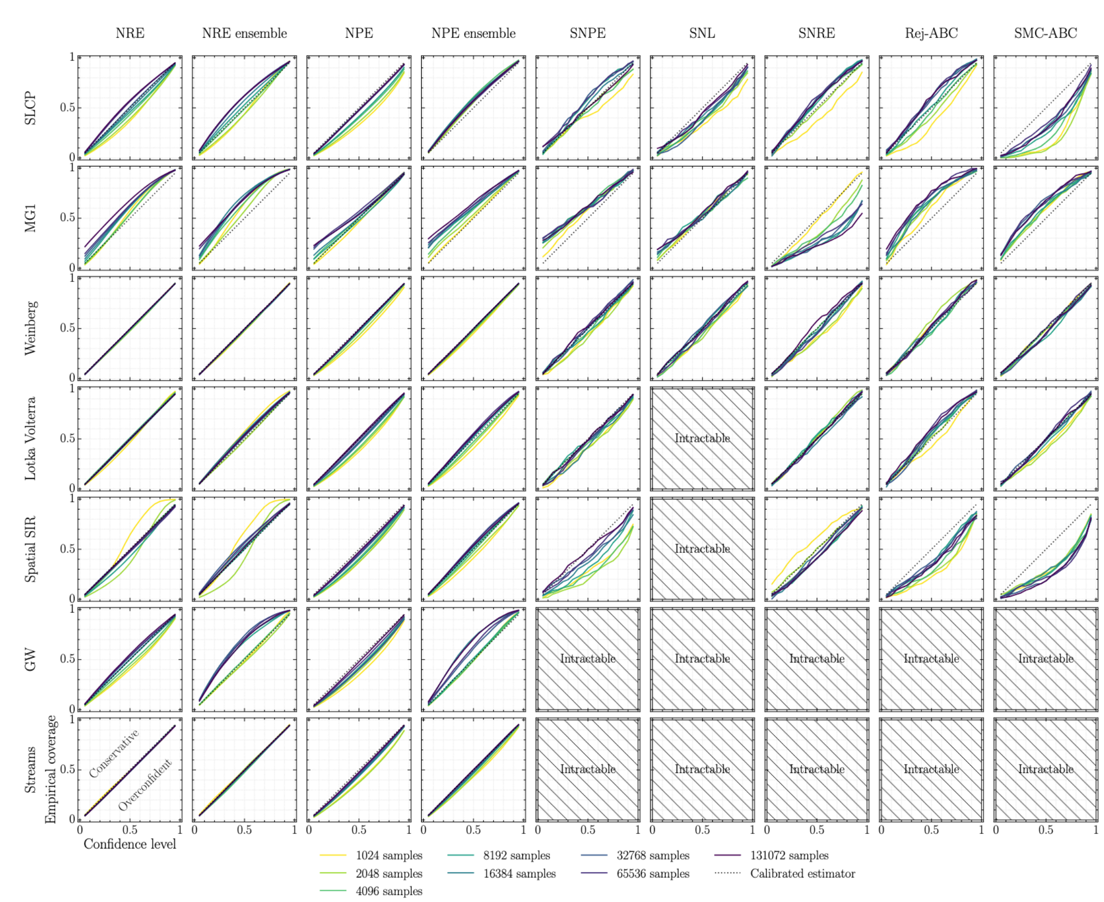
Antoine Wehenkel
University of Liège
antoine.wehenkel@uliege.be

Gilles Louppe
University of Liège
g.louppe@uliege.be

Abstract

We present extensive empirical evidence showing that current Bayesian simulation-based inference algorithms are inadequate for the falsificationist methodology of scientific inquiry. Our results collected through months of experimental computations show that all benchmarked algorithms – (S)NPE, (S)NRE, SNL and variants of ABC – may produce overconfident posterior approximations, which makes them demonstrably unreliable and dangerous if one's scientific goal is to constrain parameters of interest. We believe that failing to address this issue will lead to a well-founded trust crisis in simulation-based inference. For this reason, we argue that research efforts should now consider theoretical and methodological developments of conservative approximate inference algorithms and present research directions towards this objective. In this regard, we show empirical evidence that ensembles are consistently more reliable.

Trouble in paradise???



Towards Reliable Simulation-Based Inference with Balanced Neural Ratio Estimation

Arnaud Delaunoy* University of Liège a.delaunoy@uliege.be

Joeri Hermans* Unaffiliated joeri@peinser.com

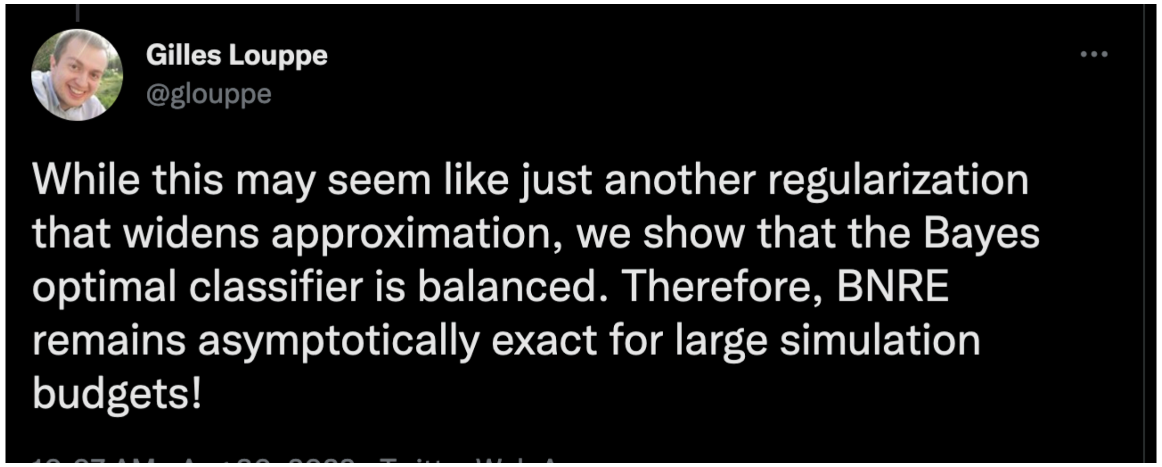
François Rozet University of Liège francois.rozet@uliege.be

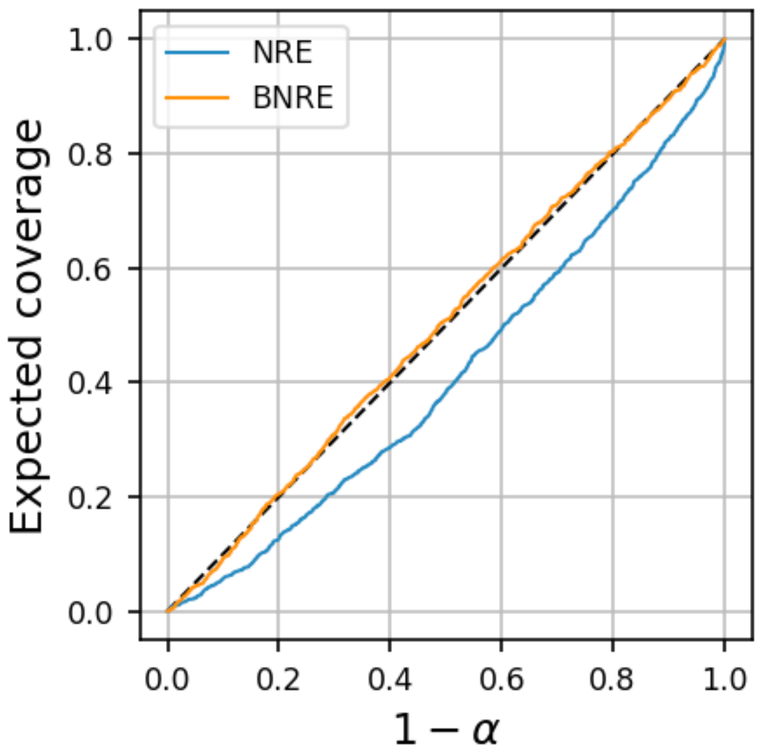
Antoine Wehenkel University of Liège antoine.wehenkel@uliege.be

Gilles Louppe University of Liège g.louppe@uliege.be

Definition 1. *A classifier $\hat{d}$ is balanced if* $\mathbb{E}_{p(\boldsymbol{\vartheta},\boldsymbol{x})}\left[\hat{d}(\boldsymbol{\vartheta},\boldsymbol{x})\right] = \mathbb{E}_{p(\boldsymbol{\vartheta})p(\boldsymbol{x})}\left[1-\hat{d}(\boldsymbol{\vartheta},\boldsymbol{x})\right]$, *or*

$$\mathbb{E}_{p(\boldsymbol{\vartheta},\boldsymbol{x})}\left[\hat{d}(\boldsymbol{\vartheta},\boldsymbol{x})\right] + \mathbb{E}_{p(\boldsymbol{\vartheta})p(\boldsymbol{x})}\left[\hat{d}(\boldsymbol{\vartheta},\boldsymbol{x})\right] = 1. \quad (3)$$





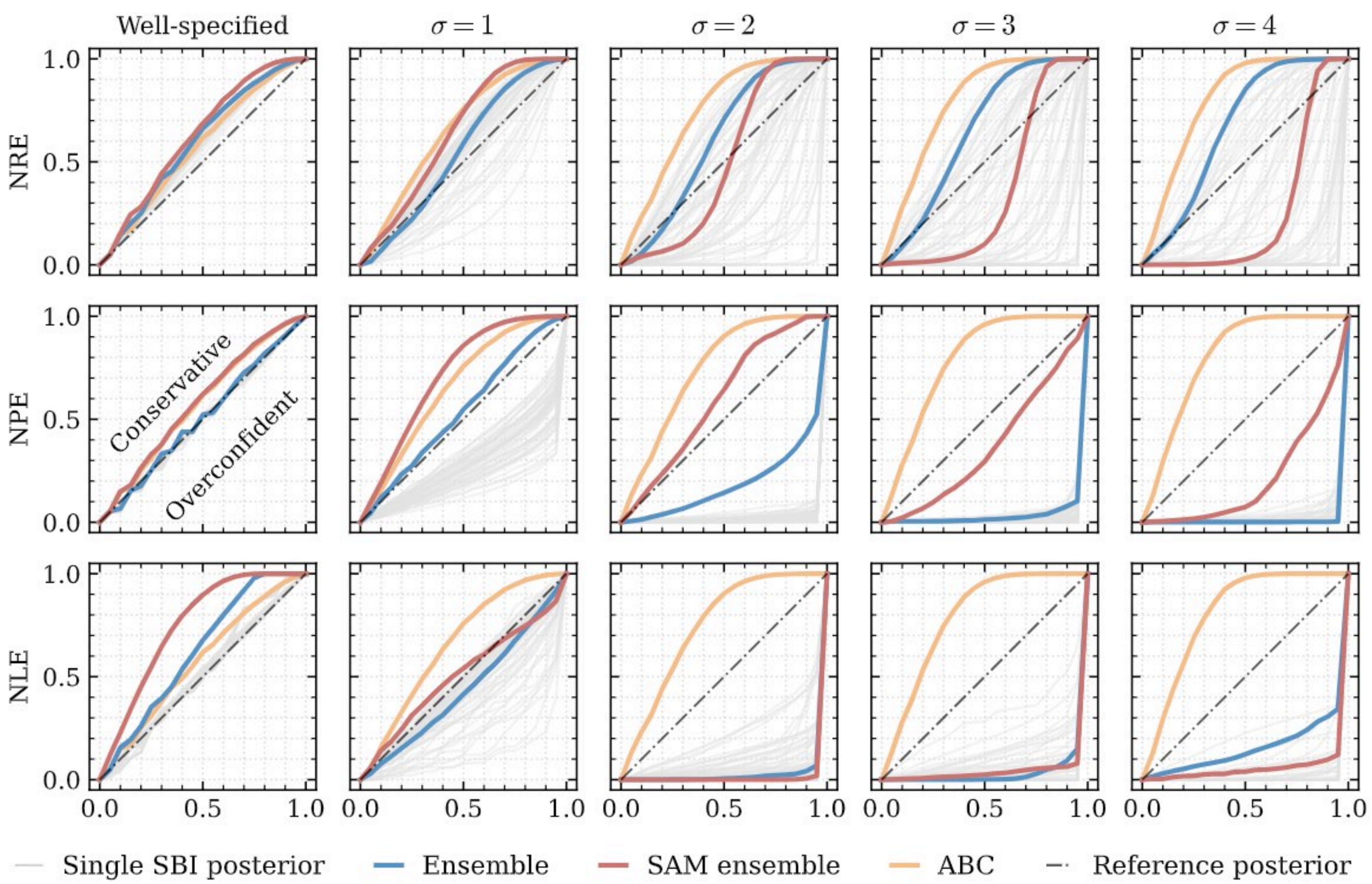
Theorem 1 shows that, in expectation over the joint distribution $p(\boldsymbol{\vartheta},\boldsymbol{x})$, a balanced classifier $\hat{d}$ tends to make predictions whose probability values $\hat{d}(\boldsymbol{\vartheta},\boldsymbol{x})$ are smaller than the exact probability values $d(\boldsymbol{\vartheta},\boldsymbol{x})$. In other words, a balanced classifier $\hat{d}$ tends to be less confident than the Bayes optimal classifier d. Similarly, Theorem 2 shows that, in expectation over the product of the marginals $p(\boldsymbol{\vartheta})p(\boldsymbol{x})$, a balanced classifier tends to make predictions whose probability values $1-\hat{d}(\boldsymbol{\vartheta},\boldsymbol{x})$ are smaller than the exact probability values $1-d(\boldsymbol{\vartheta},\boldsymbol{x})$, hence showing that a balanced classifier $\hat{d}$ tends to also be less confident than the Bayes optimal classifier d. We note however that these two

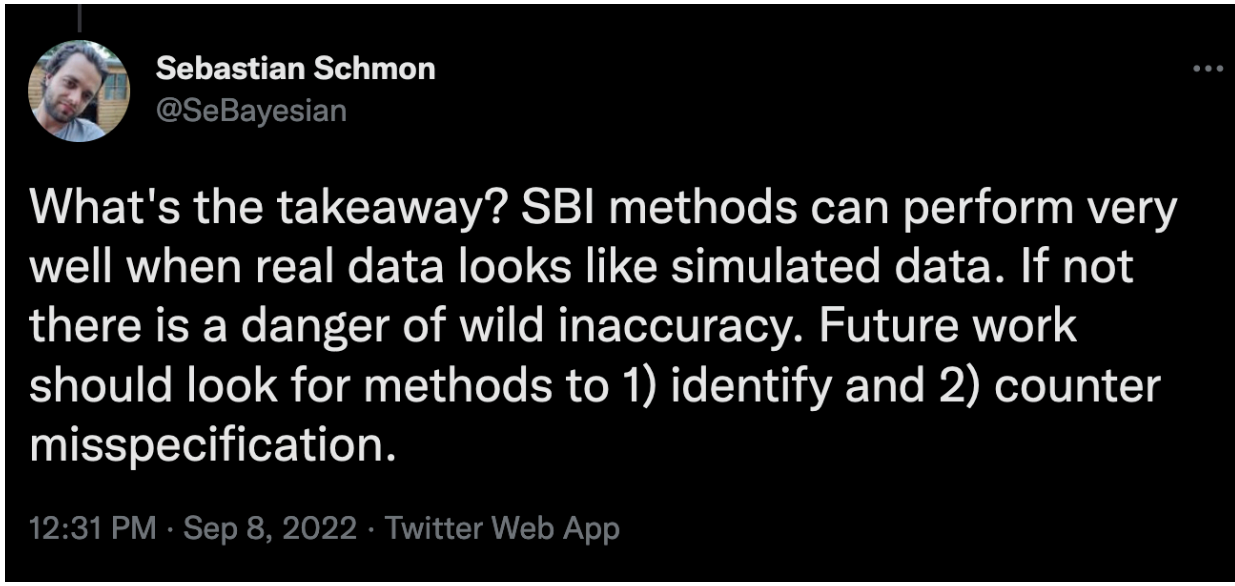
Investigating the Impact of Model Misspecification in Neural Simulation-based Inference

Patrick Cannon*[1], Daniel Ward[2], and Sebastian M. Schmon[1,3]

[1] *Improbable, UK*
[2] *School of Mathematics, Bristol University, UK*
[3] *Department of Mathematical Sciences, Durham University, UK*

Gaussian model with wrong variance





Discussion

Summary

- **Simulation-based inference** is making rapid progress with new deep learning algorithms.
- Several routes: NPE, NLE, NRE, sequential/active methods….
- Already available software implementations.

Discussion

- Many cool applications of SBI I haven't mentioned: neuroscience, epidemiology, particle physics, …
- Ongoing work examines consistency, how modifications to vanilla algorithms can avoid mistakes, improving efficiency.
- Together we can unlock the full scientific content of the data we measure!

backup slides