Towards reliable simulation-based inference and beyond

Hammers and Nails 2022
Machine Learning Meets Astro & Particle Physics

August 4

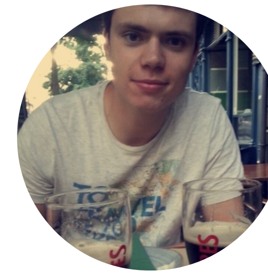
Gilles Louppe
g.louppe@uliege.be

Kyle Cranmer

Johann Brehmer

Joeri Hermans

Antoine Wehenkel



Norman Marlier

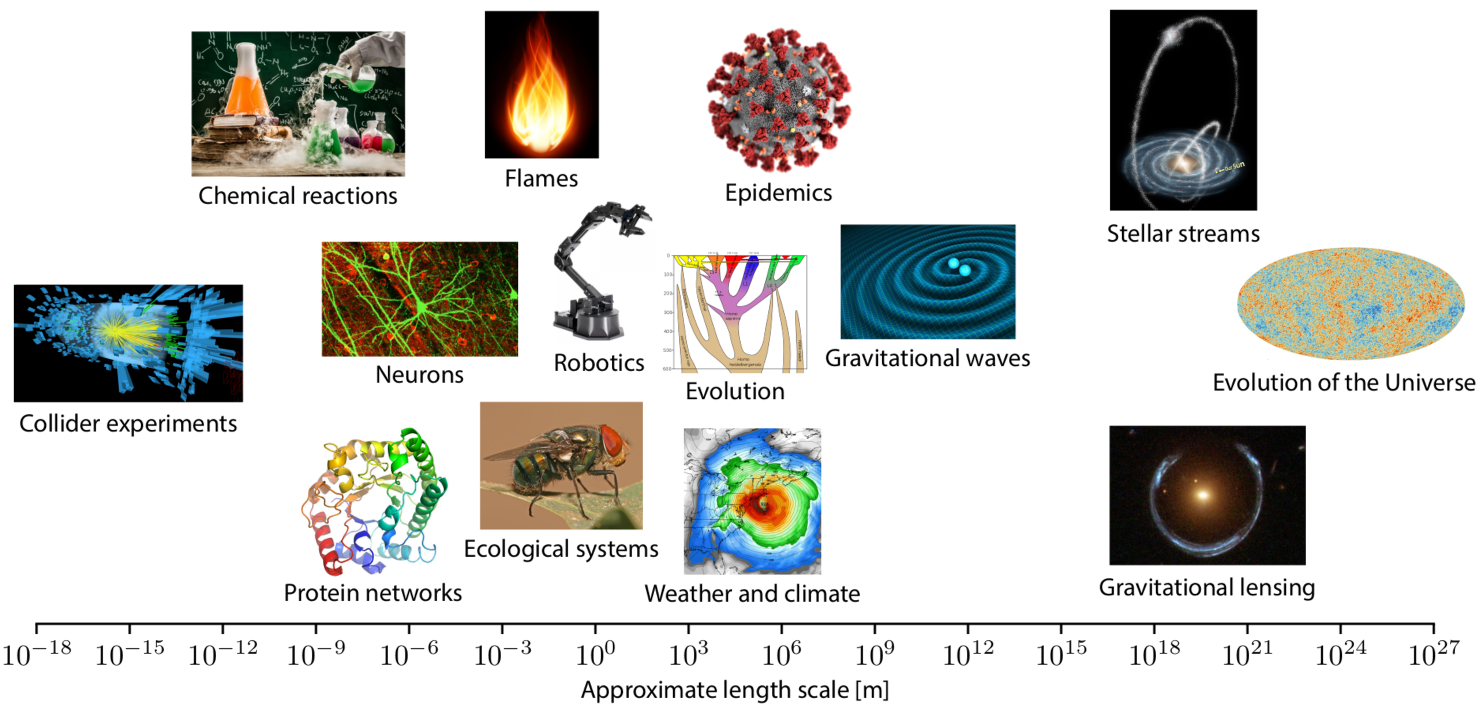
Siddharth Mishra-Sharma

Christoph Weniger

Arnaud Delaunoy

Malavika Vasist

Francois Rozet


Chemical reactions
Flames
Epidemics
Stellar streams
Neurons
Robotics
Evolution
Gravitational waves
Evolution of the Universe
Collider experiments
Ecological systems
Protein networks
Weather and climate
Gravitational lensing
10^{-18} 10^{-15} 10^{-12} 10^{-9} 10^{-6} 10^{-3} 10^{0} 10^{3} 10^{6} 10^{9} 10^{12} 10^{15} 10^{18} 10^{21} 10^{24} 10^{27}
Approximate length scale [m]

$$v_x = v\cos(\alpha), \;\; v_y = v\sin(\alpha),$$

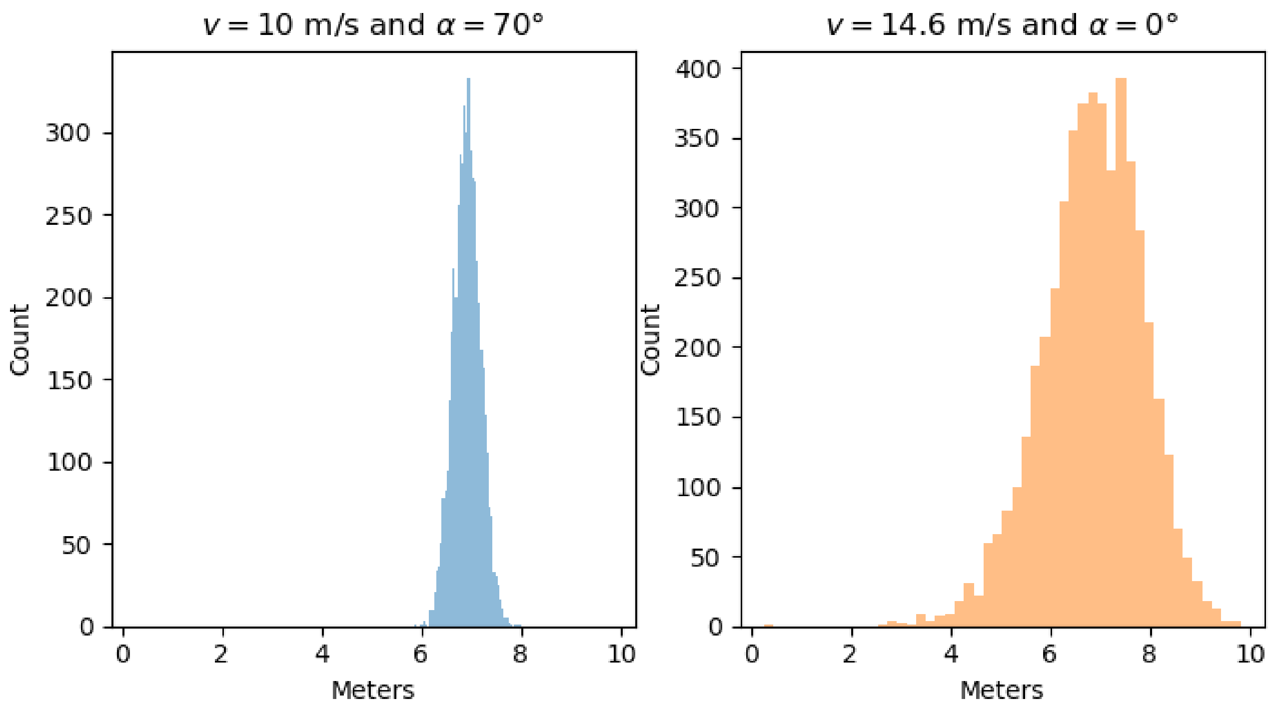
$$\frac{dx}{dt} = v_x\,, \;\; \frac{dy}{dt} = v_y, \frac{dv_y}{dt} = -G.$$

```
def simulate(v, alpha, dt=0.001):
    v_x = v * np.cos(alpha)  # x velocity m/s
    v_y = v * np.sin(alpha)  # y velocity m/s
    y = 1.1 + 0.3 * random.normal()
    x = 0.0

    while y > 0: # simulate until ball hits floor
        v_y += dt * -G  # acceleration due to gravity
        x += dt * v_x
        y += dt * v_y

    return x + 0.25 * random.normal()
```

The computer simulator defines the likelihood function $p(x|\theta)$ implicitly.



What parameter values θ are plausible given the observation x?

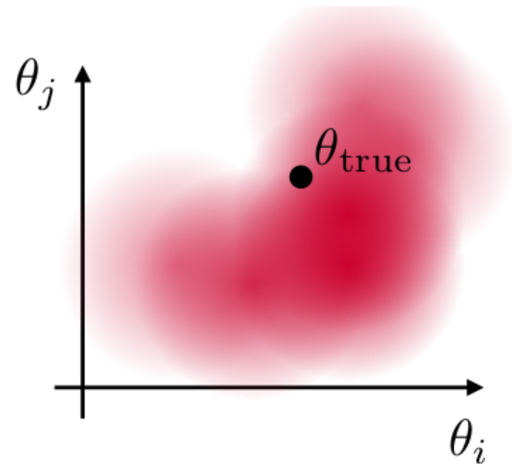
Bayesian inference

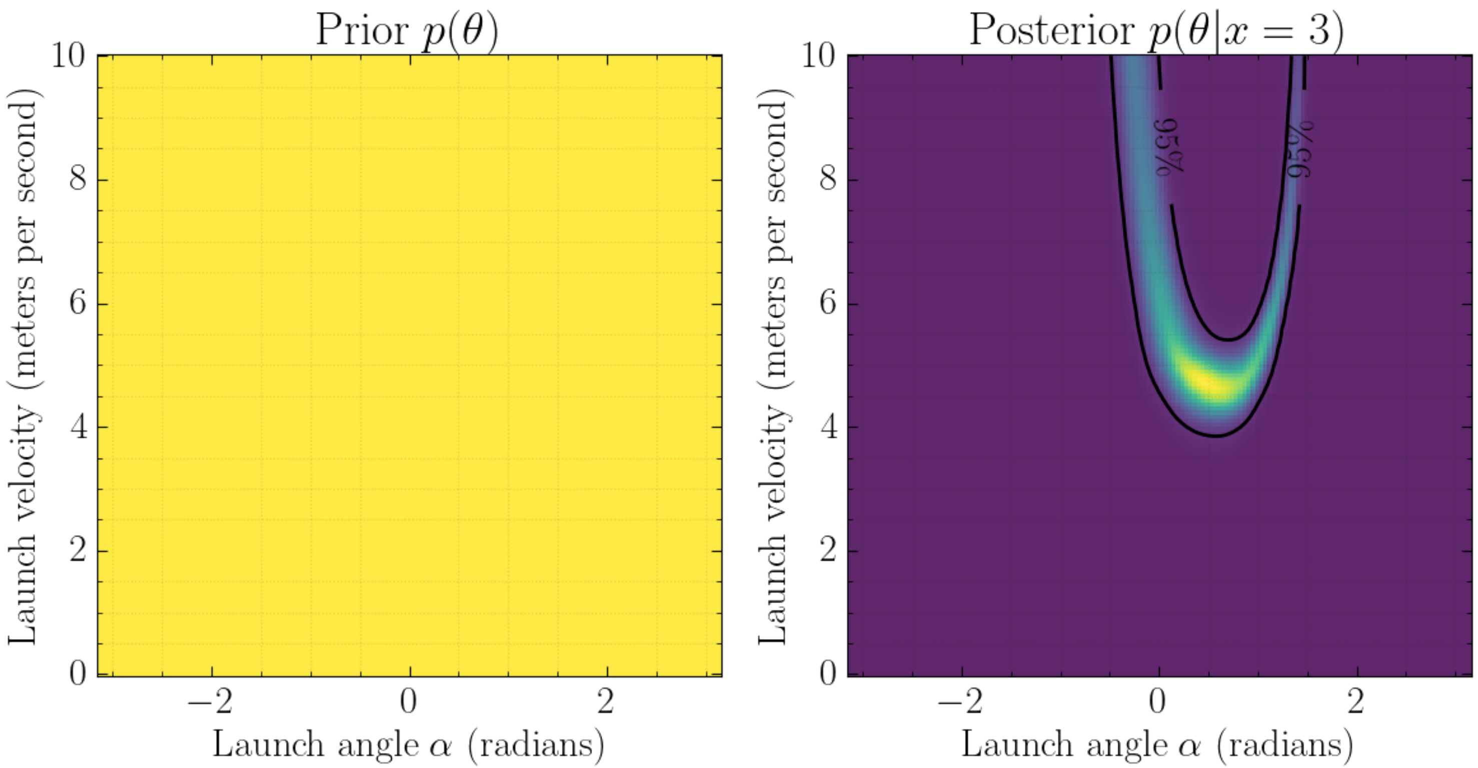
Start with

- a simulator that can generate N samples $x_i \sim p(x_i|\theta_i)$,
- a prior model $p(\theta)$,
- observed data $x_{\text{obs}} \sim p(x_{\text{obs}}|\theta_{\text{true}})$.

Then, estimate the posterior

$$p(\theta|x_{\text{obs}}) = \frac{p(x_{\text{obs}}|\theta)p(\theta)}{p(x_{\text{obs}})}$$



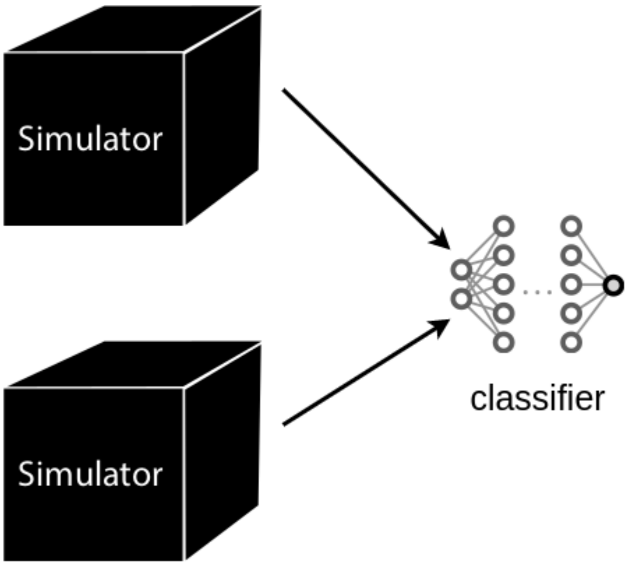

Prior $p(\theta)$
Posterior $p(\theta|x = 3)$
Launch velocity (meters per second)
Launch angle α (radians)
95%
95%
10
8
6
4
2
0
−2
0
2

Neural ratio estimation (NRE)

The likelihood-to-evidence $r(x|\theta) = \frac{p(x|\theta)}{p(x)} = \frac{p(x,\theta)}{p(x)p(\theta)}$ ratio can be learned, even if neither the likelihood nor the evidence can be evaluated:

$x, \theta \sim p(x, \theta)$

$x, \theta \sim p(x)p(\theta)$



$\hat{r}(x|\theta)$

—

Credits: Cranmer et al, 2015; Hermans et al, 2020.

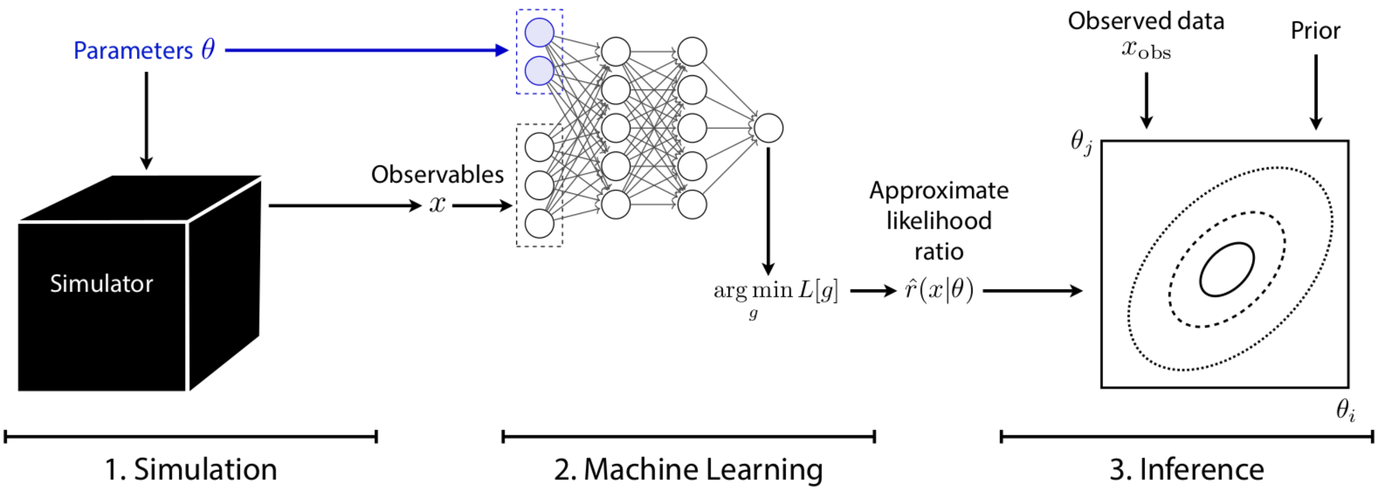
The solution d found after training approximates the optimal classifier

$$d(x,\theta) \approx d^*(x,\theta) = \frac{p(x,\theta)}{p(x,\theta) + p(x)p(\theta)}.$$

Therefore,

$$r(x|\theta) = \frac{p(x|\theta)}{p(x)} = \frac{p(x,\theta)}{p(x)p(\theta)} \approx \frac{d(x,\theta)}{1 - d(x,\theta)} = \hat{r}(x|\theta).$$

$$p(\theta|x) = \frac{p(x|\theta)p(\theta)}{p(x)} \approx \hat{r}(x|\theta)p(\theta)$$

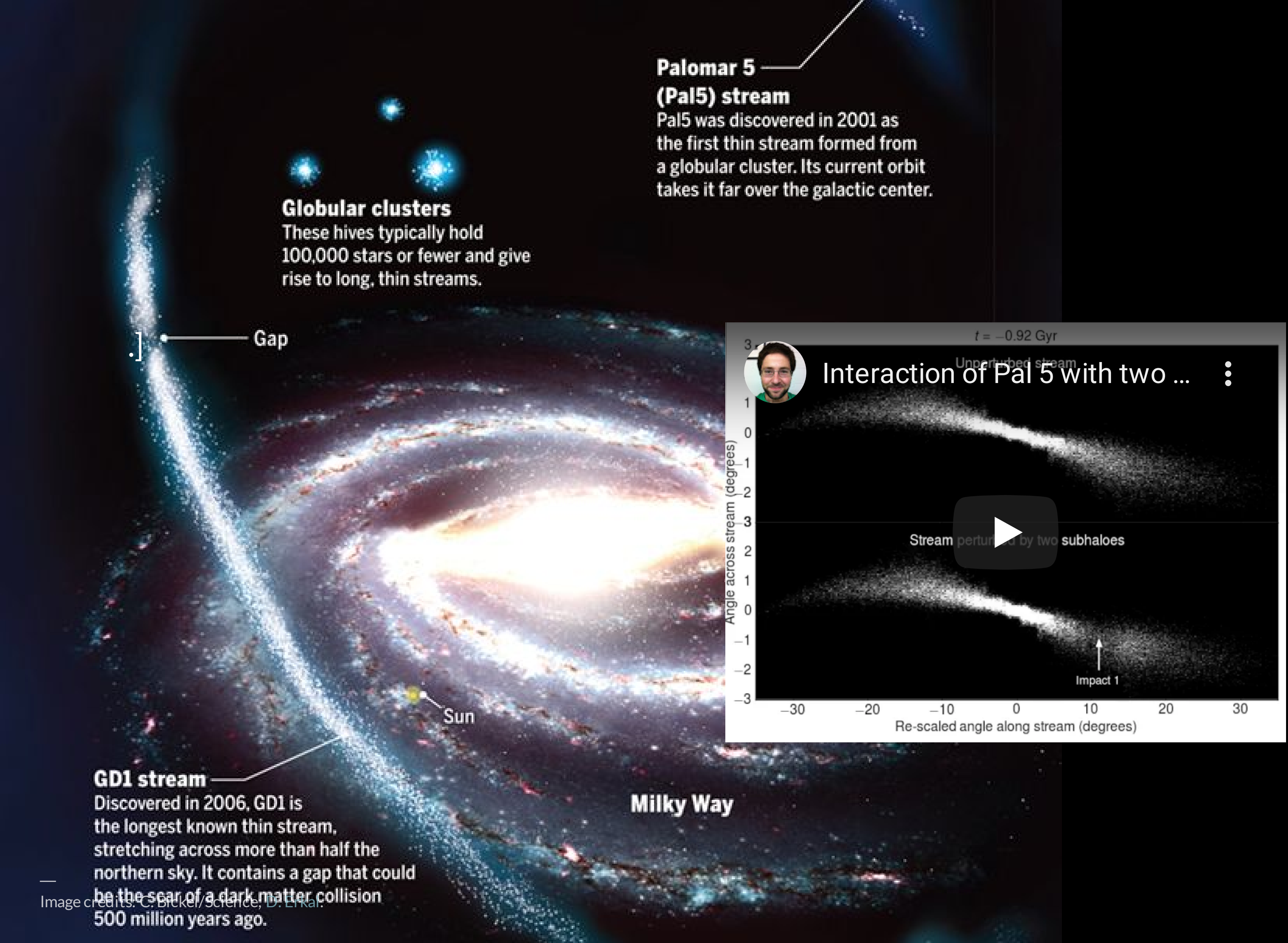


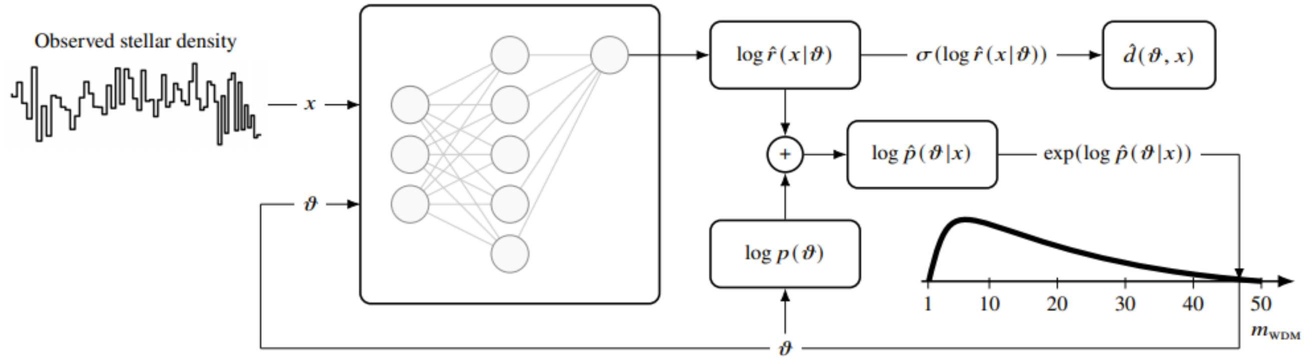
Run simulator and save data

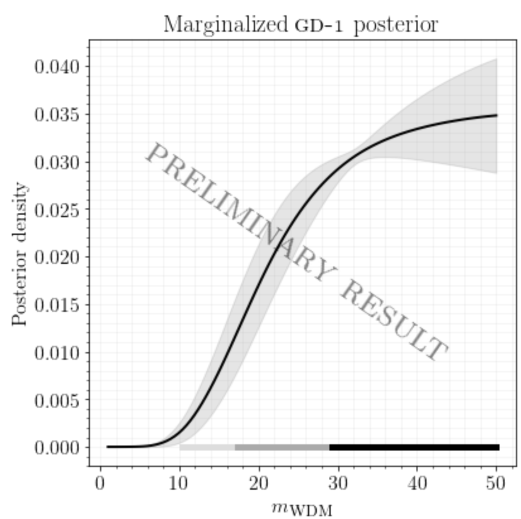
Train NN classifier, interpret as likelihood ratio estimator

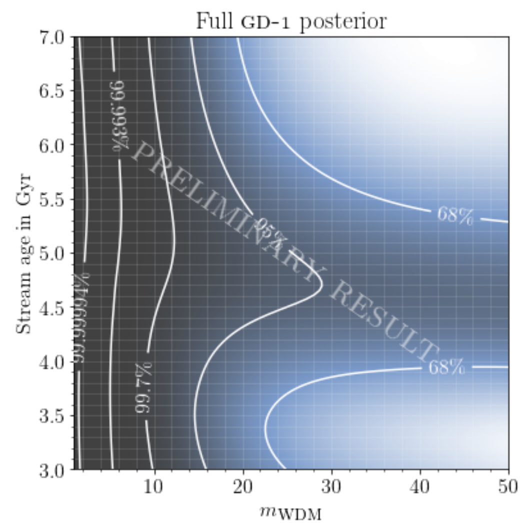
Amortized: cheap to repeat for new data

Constraining dark matter with stellar streams









—

Credits: Hermans et al, 2021.

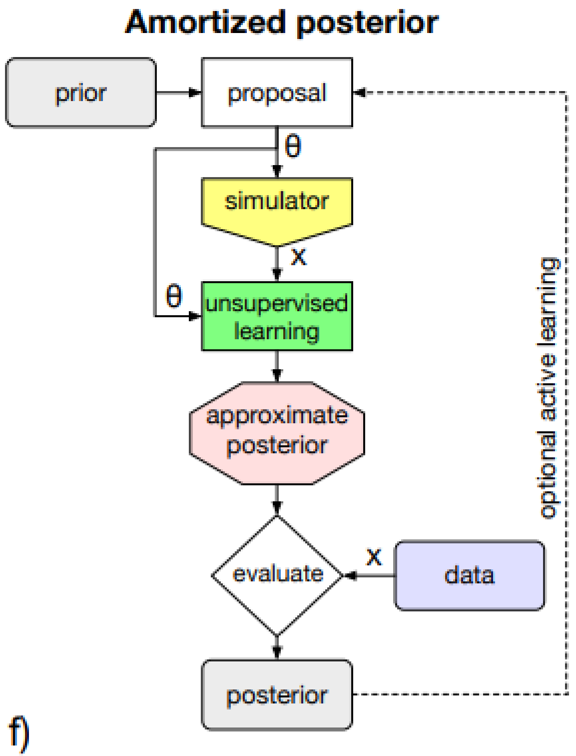
Preliminary results for GD-1 suggest a **preference for CDM over WDM**.

Neural Posterior Estimation (NPE)

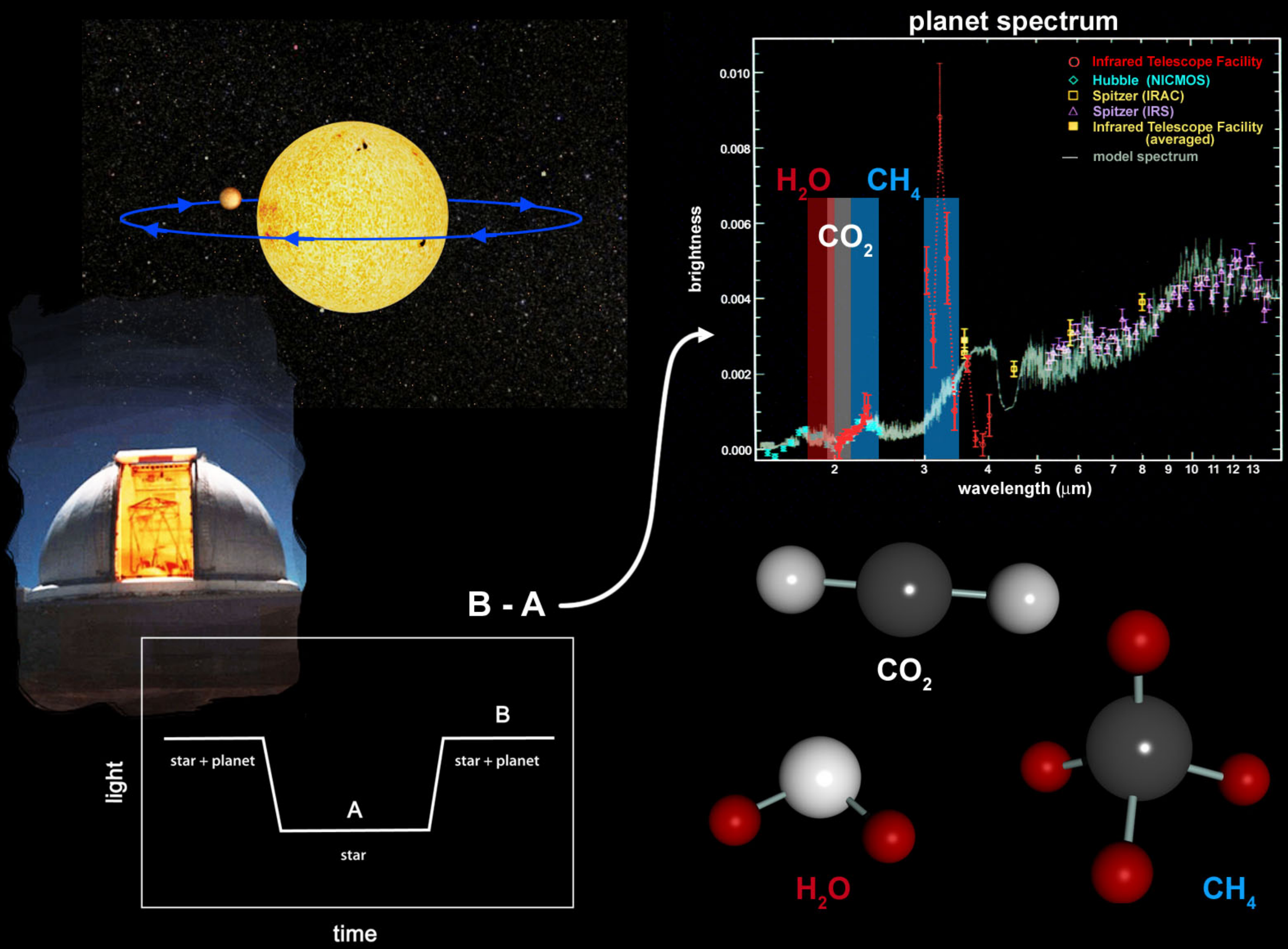
Use variational inference to directly estimate the posterior, by solving

$$\min_{q_\phi} \mathbb{E}_{p(x)} \left[\mathrm{KL}(p(\theta|x)||q_\phi(\theta|x))\right]$$

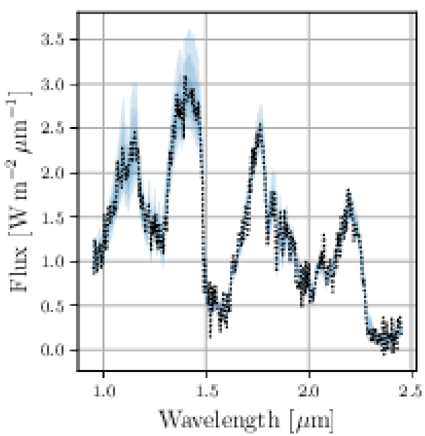
where q_ϕ is a neural density estimator, such as a normalizing flow.



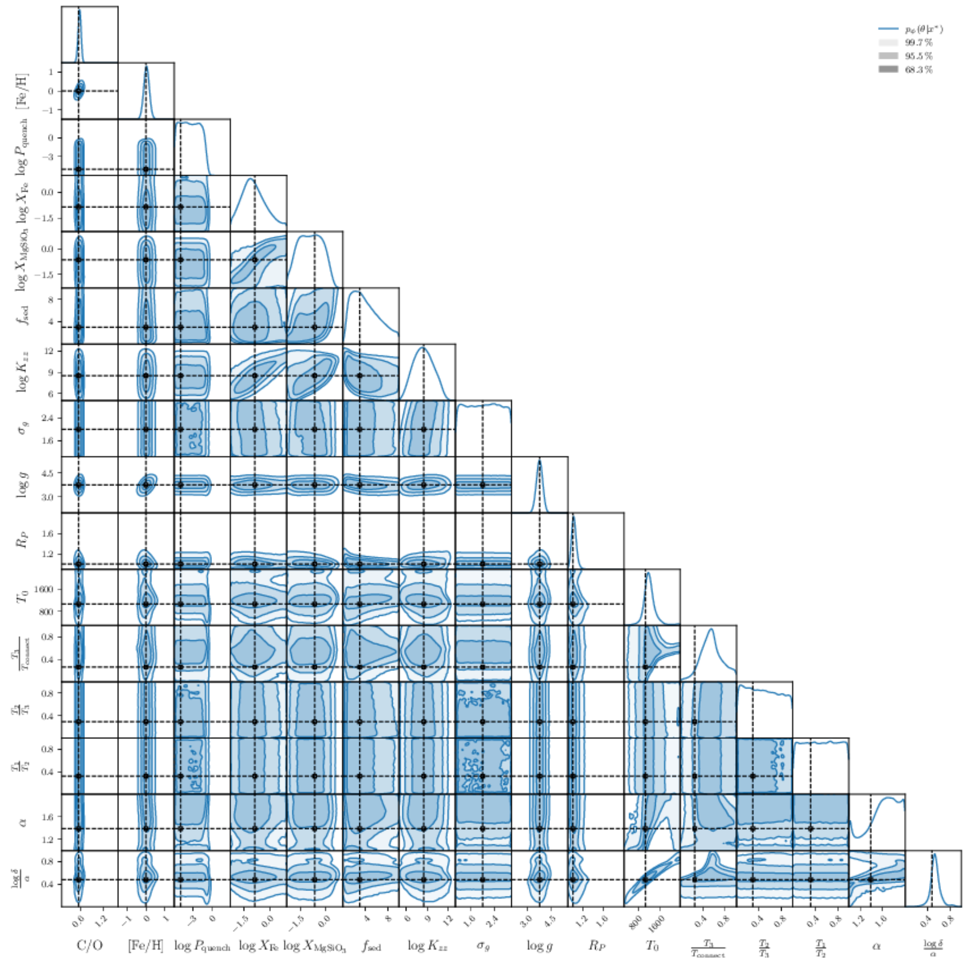
Exoplanet atmosphere characterization



—
Credits: NSA/JPL-Caltech, 2010.



Flux [W m^{-2} μm^{-1}]
Wavelength [μm]

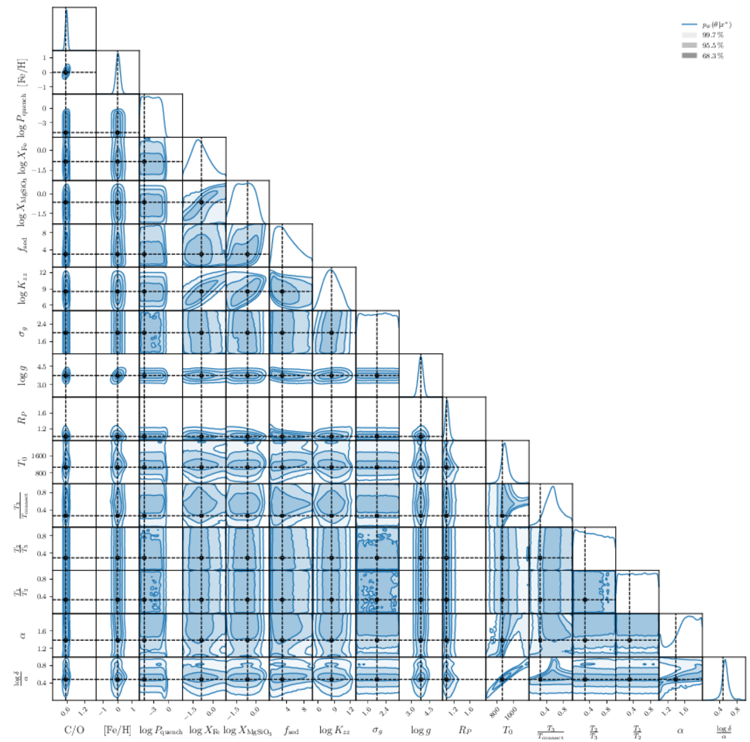


99.7 %
95.5 %
68.3 %
C/O
[Fe/H]
log P_{quench}
log X_{Fe}
log $X_{\mathrm{MgSiO_3}}$
f_{sed}
log K_{zz}
σ_g
log g
R_P
T_0
$\frac{T_3}{T_{\mathrm{connect}}}$
$\frac{T_2}{T_3}$
$\frac{T_1}{T_2}$
α
$\frac{\log \delta}{\alpha}$

Computational faithfulness

$$\hat{p}(\theta|x) = \text{sbi}(p(x|\theta), p(\theta), x)$$

We must make sure our approximate simulation-based inference algorithms can (at least) actually realize faithful inferences on the (expected) observations.

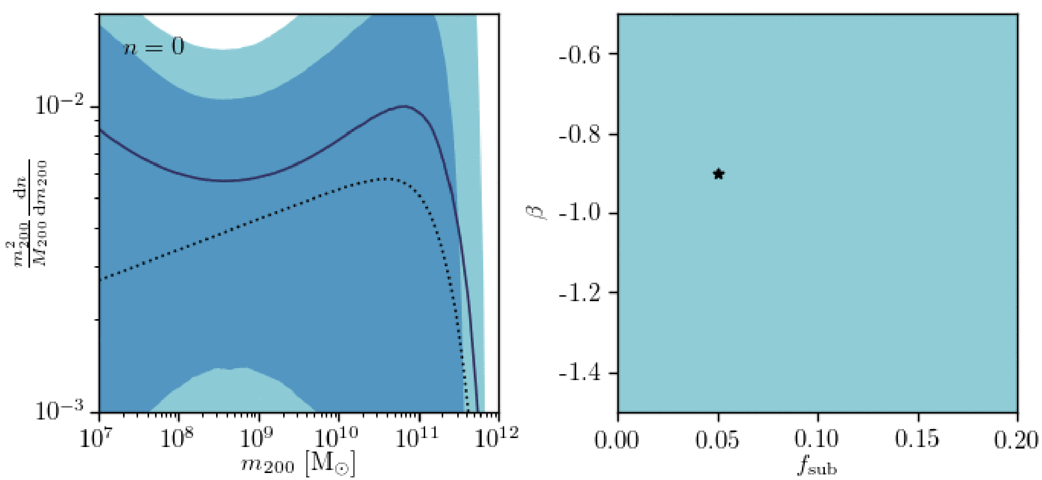


How do we know this is good enough?

Mode convergence:

The maximum a posteriori estimate converges towards the nominal value θ^* for an increasing number of independent and identically distributed observables $x_i \sim p(x|\theta^*)$:

$$\lim_{N\to\infty} \arg\max_\theta p(\theta|\{x_i\}_{i=1}^N)$$
$$= \lim_{N\to\infty} \arg\max_\theta p(\theta) \prod_{x_i} r(x_i|\theta) = \theta^*$$



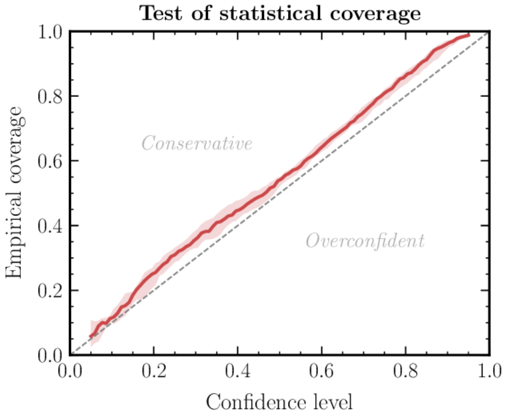
—
Credits: Brehmer et al, 2019.

A common observation at the root of several other diagnostics is to check for the **self-consistency** of the Bayesian joint distribution,

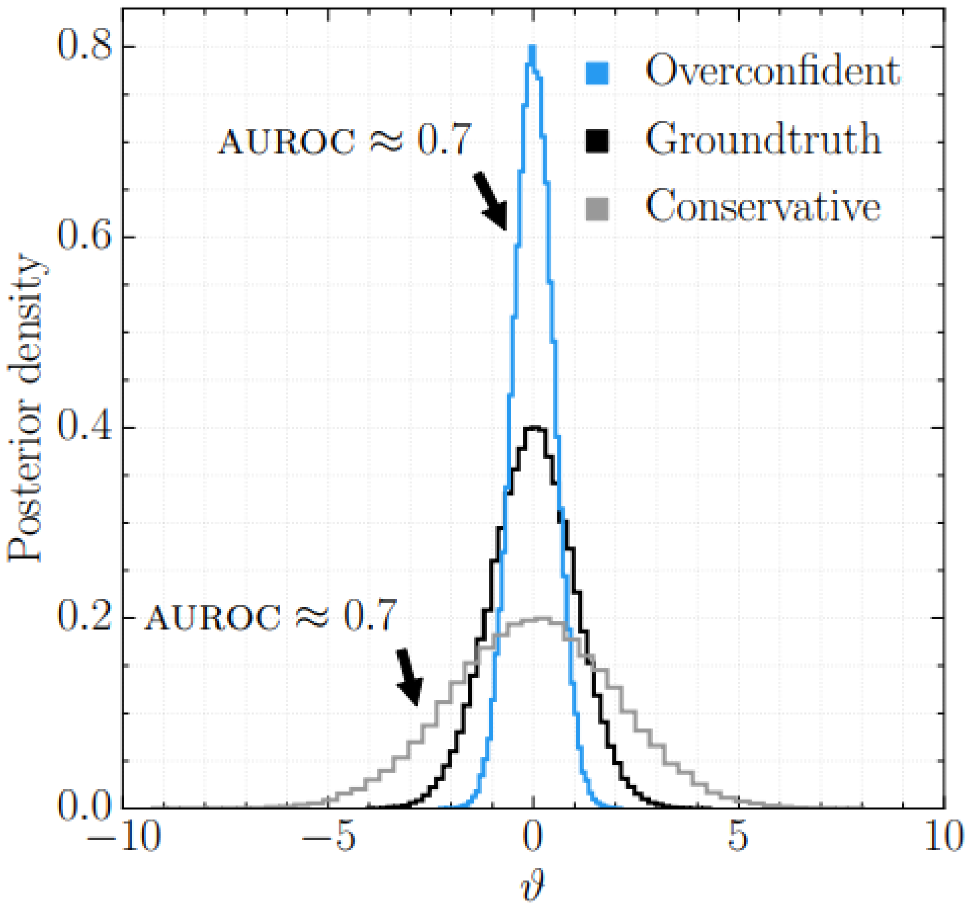
$$p(\theta) = \int p(\theta')p(x|\theta')p(\theta|x)d\theta'\, dx.$$

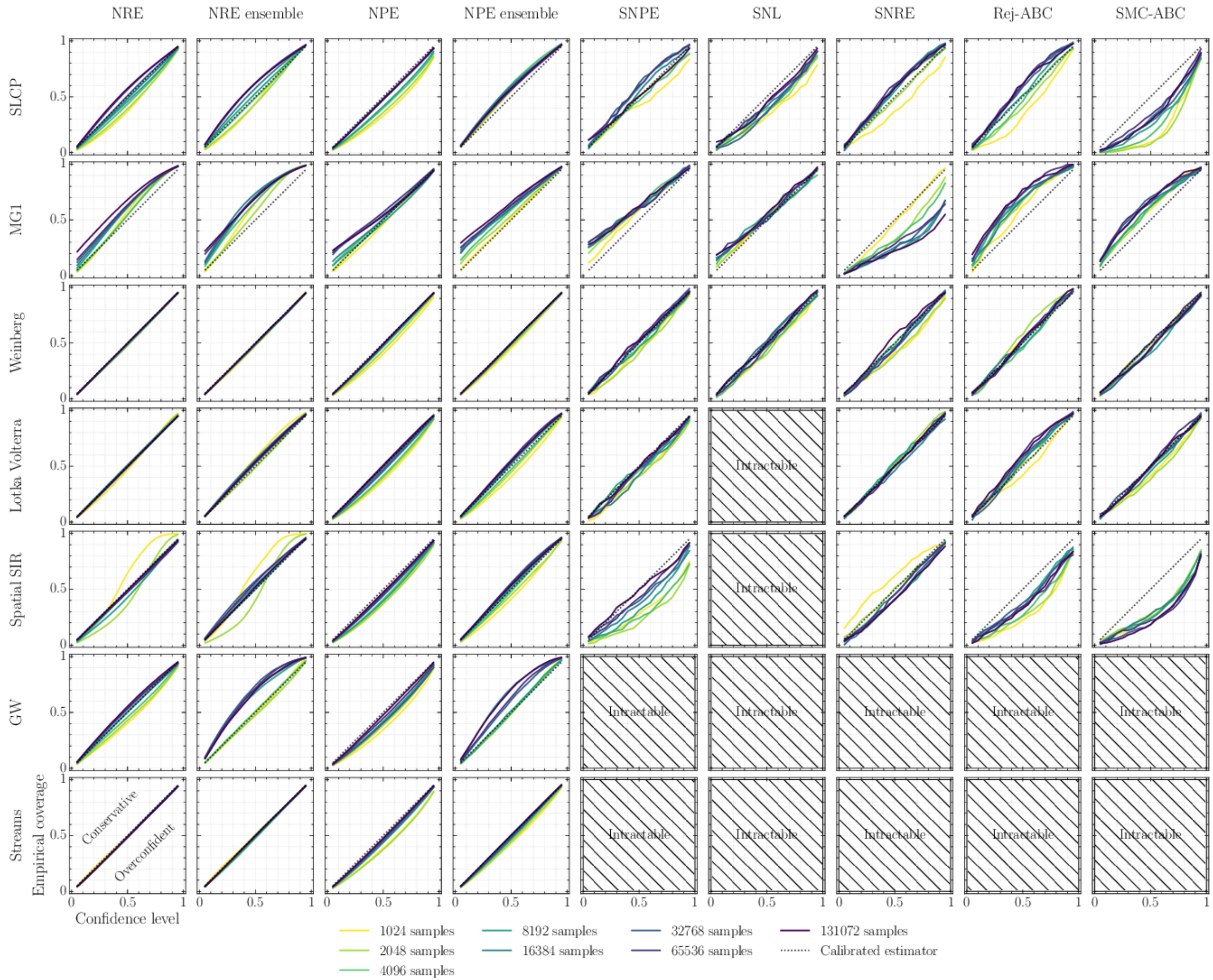
Coverage diagnostic:

- For $x, \theta \sim p(x, \theta)$, compute the $1-\alpha$ credible interval based on $\hat{p}(\theta|x)$.
- If the fraction of samples for which θ is contained within the interval is larger than the nominal coverage probability $1-\alpha$, then the approximate posterior $\hat{p}(\theta|x)$ has coverage.



—
Credits: Hermans et al, 2021; Siddharth Mishra-Sharma, 2021.


0.8
0.6
0.4
0.2
0.0
−10
−5
0
5
10
Posterior density
ϑ
AUROC ≈ 0.7
AUROC ≈ 0.7
Overconfident
Groundtruth
Conservative



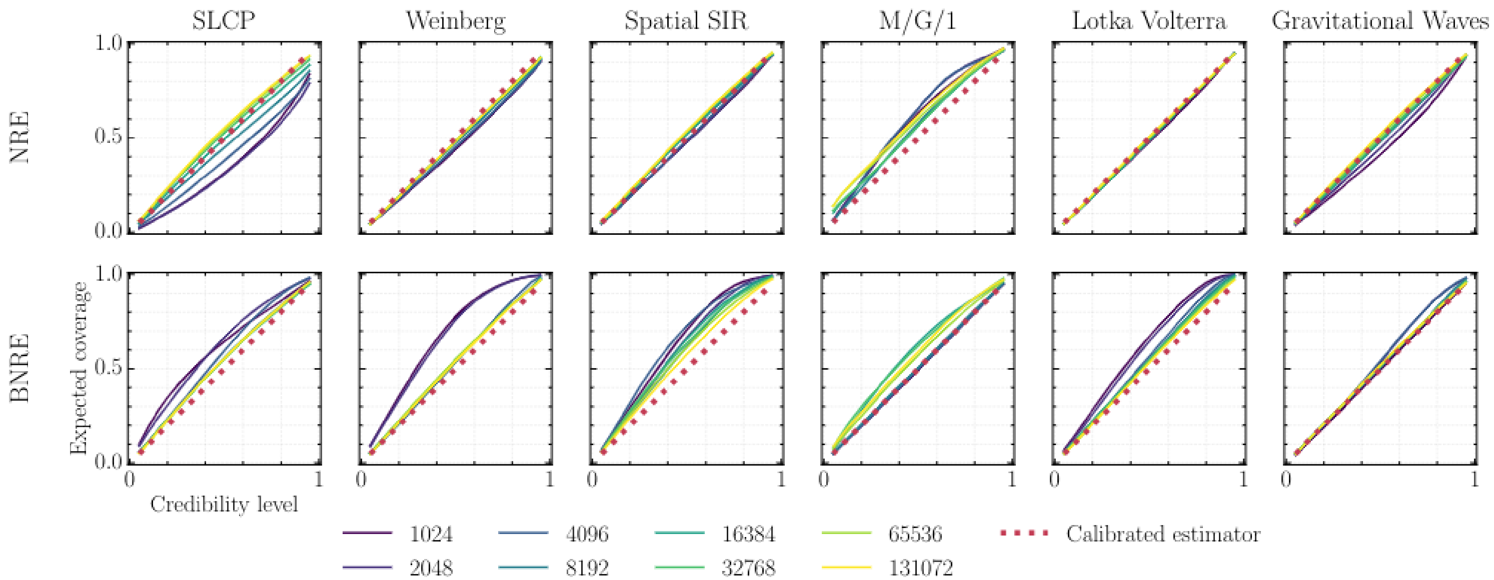
—

Credits: Hermans et al, 2021.

What if diagnostics fail?

Balanced NRE

Neural ratio estimation can be forced to be more **conservative**, hence increasing the reliability of the approximate posteriors and reducing the risk of false inferences.



Definition

A binary classifier $\hat{d}$ is balanced if

$$\mathbb{E}_{p(\theta,x)}\left[\hat{d}\left(\theta,x\right)\right] = \mathbb{E}_{p(\theta)p(x)}\left[1-\hat{d}\left(\theta,x\right)\right].$$

Theorems 1 and 2

Any balanced classifier $\hat{d}$ satisfies

$$\mathbb{E}_{p(\theta,x)}\left[\frac{d(\theta,x)}{\hat{d}\left(\theta,x\right)}\right] \geq 1 \quad \text{and} \quad \mathbb{E}_{p(\theta)p(x)}\left[\frac{1-d(\theta,x)}{1-\hat{d}\left(\theta,x\right)}\right] \geq 1.$$

Algorithm 1 Training algorithm for Balanced Neural Ratio Estimation (BNRE).

Inputs:	Implicit generative model $p(\boldsymbol{x} \mid \boldsymbol{\vartheta})$ (simulator) and prior $p(\boldsymbol{\vartheta})$
Outputs:	Approximate classifier $\hat{d}_\psi(\boldsymbol{\vartheta}, \boldsymbol{x})$ parameterized by ψ
hyper-parameters:	Balancing condition strength λ (default = 100) and batch-size n

repeat

  Sample data from the joint $\{\boldsymbol{\vartheta}_i,\ \boldsymbol{x}_i \sim p(\boldsymbol{\vartheta}, \boldsymbol{x}),\ y_i = 1\}_{i=1}^{n/2}$

  Sample data from the marginals $\{\boldsymbol{\vartheta}_i,\ \boldsymbol{x}_i \sim p(\boldsymbol{\vartheta})p(\boldsymbol{x}),\ y_i = 0\}_{i=n/2+1}^{n}$

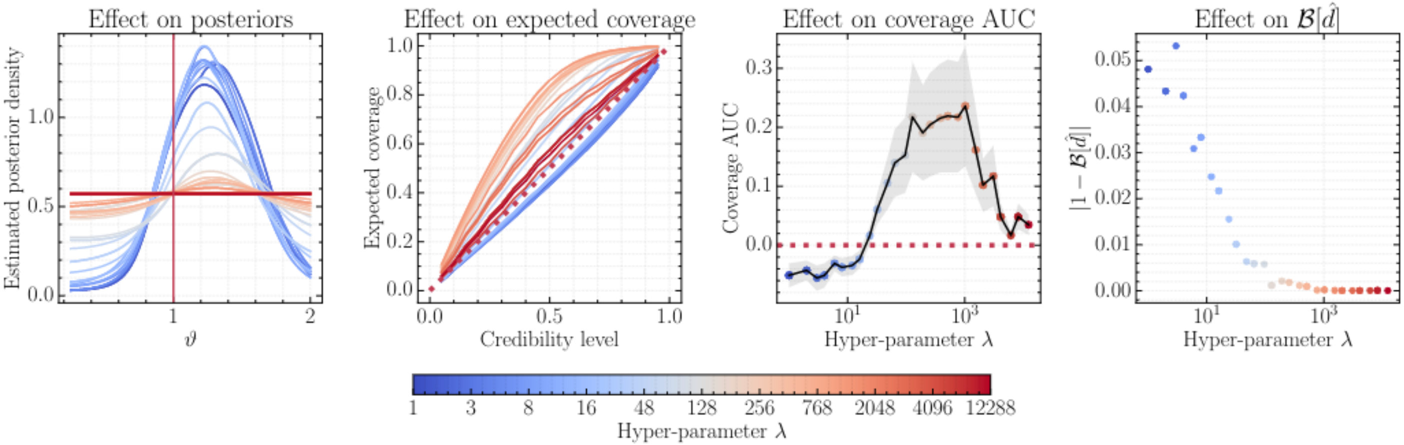
  $\mathcal{L}[\hat{d}_\psi] = -\frac{1}{n}\sum_{i=1}^{n} y_i \log \hat{d}_\psi(\boldsymbol{\vartheta}_i, \boldsymbol{x}_i) + (1 - y_i)\log(1 - \hat{d}_\psi(\boldsymbol{\vartheta}_i, \boldsymbol{x}_i))$

  $\mathcal{B}[\hat{d}_\psi] = \frac{2}{n}\sum_{i=1}^{n/2} \hat{d}_\psi(\boldsymbol{\vartheta}_i, \boldsymbol{x}_i) + \frac{2}{n}\sum_{i=n/2+1}^{n} \hat{d}_\psi(\boldsymbol{\vartheta}_i, \boldsymbol{x}_i)$

  $\psi = \texttt{minimizer\_step(params=}\psi\texttt{, loss=}\mathcal{L}[\hat{d}_\psi] + \lambda(\mathcal{B}[\hat{d}_\psi] - 1)^2\texttt{)}$

until convergence

return $\hat{d}_\psi(\boldsymbol{\vartheta}, \boldsymbol{x})$.



Effect on posteriors
Estimated posterior density
ϑ
Effect on expected coverage
Expected coverage
Credibility level
Effect on coverage AUC
Coverage AUC
Hyper-parameter λ
Effect on $\mathcal{B}[\hat{d}]$
$|1 - \mathcal{B}[\hat{d}]|$
Hyper-parameter λ
1 3 8 16 48 128 256 768 2048 4096 12288
Hyper-parameter λ

Wait a minute... What if you are model is wrong?

The observational model $p(x|\theta)$

$p(x|\theta)$ should capture the pertinent structure of the true data generating process for the inference results to be useful.

A model that does not capture every precise detail of the true data generating process can still be useful if it captures the details relevant to the particular analysis goals.

—

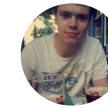
Credits: Michael Betancourt, 2020.

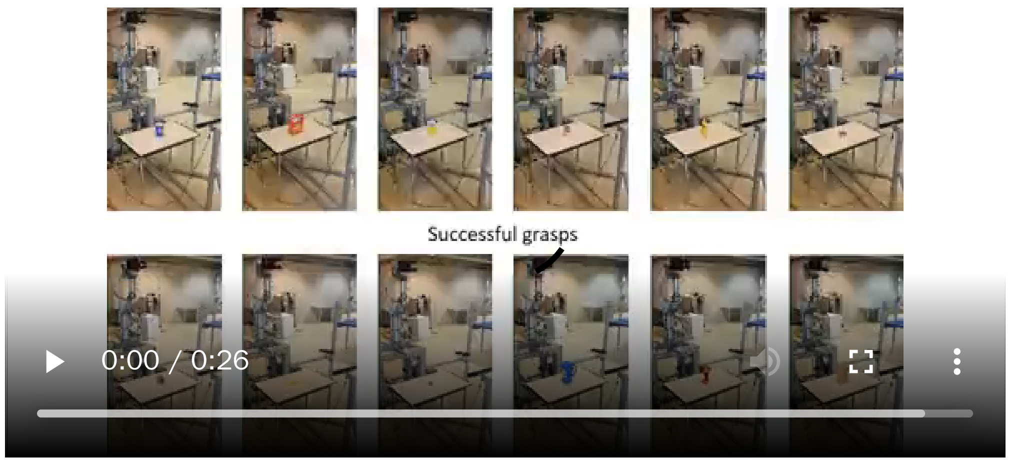
The observational model can often be made richer by including in it additional **nuisance parameters** ν that capture known unknowns.

In this case, the likelihood becomes

$$p(x|\theta) = \int p(x|\theta, \nu)p(\nu|\theta)d\nu.$$

Although nuisance parameters can reduce model misspecification, their presence and marginalization will result in increased uncertainties for the parameters θ of interest.





Nuisance parameters are used to model known unknowns in a robotic setup (e.g., camera position, table position, etc).

—

Credits: Marlier et al, 2021.

The prior model $p(\theta)$

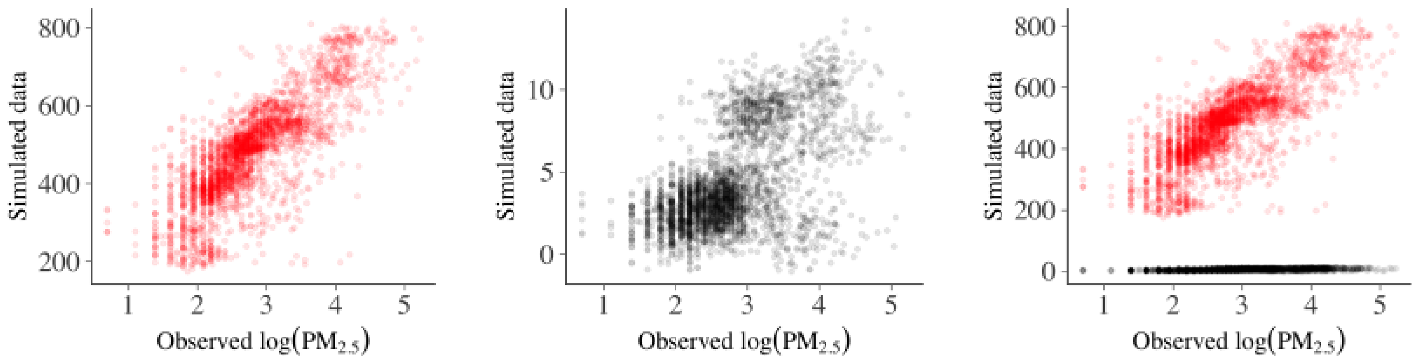
The prior model $p(\theta)$ specifies one's beliefs about the model parameters. It should reflect domain expertise.

The consequences of the prior model in the context of the observational model can be diagnosed with **prior predictive checks** to evaluate what data sets would be consistent with the prior.

A prior predictive check generates data x^{sim} according to the prior predictive distribution $p(x)$ as

$$\begin{aligned} \theta^{\text{sim}} &\sim p(\theta) \\ x^{\text{sim}} &\sim p(x|\theta^{\text{sim}}), \end{aligned}$$

or summary statistics $T(x^{\text{sim}})$ thereof.



(a) Vague priors (b) Weakly informative priors (c) Comparison

Fig. 4: *Visualizing the prior predictive distribution. Panels (a) and (b) show realizations from the prior predictive distribution using priors for the β's and τ's that are vague and weakly informative, respectively. The same $N_+(0,1)$ prior is used for σ in both cases. Simulated data are plotted on the y-axis and observed data on the x-axis. Because the simulations under the vague and weakly informative priors are so different, the y-axis scales used in panels (a) and (b) also differ dramatically. Panel (c) emphasizes the difference in the simulations by showing the red points from (a) and the black points from (b) plotted using the same y-axis.*

—
Credits: Gabry et al, 2017.

In the absence of a good prior, **neural empirical Bayes** can be used to estimate a prior distribution $p_\phi(\theta)$ by maximizing the (log) evidence of a set of observations

$$\log p_\phi(\{x_i\}_{i=1}^N) = \sum_{i=1}^N \log \int p(x_i|\theta) p_\phi(\theta) d\theta.$$

—
Credits: Vandegar et al, 2021.

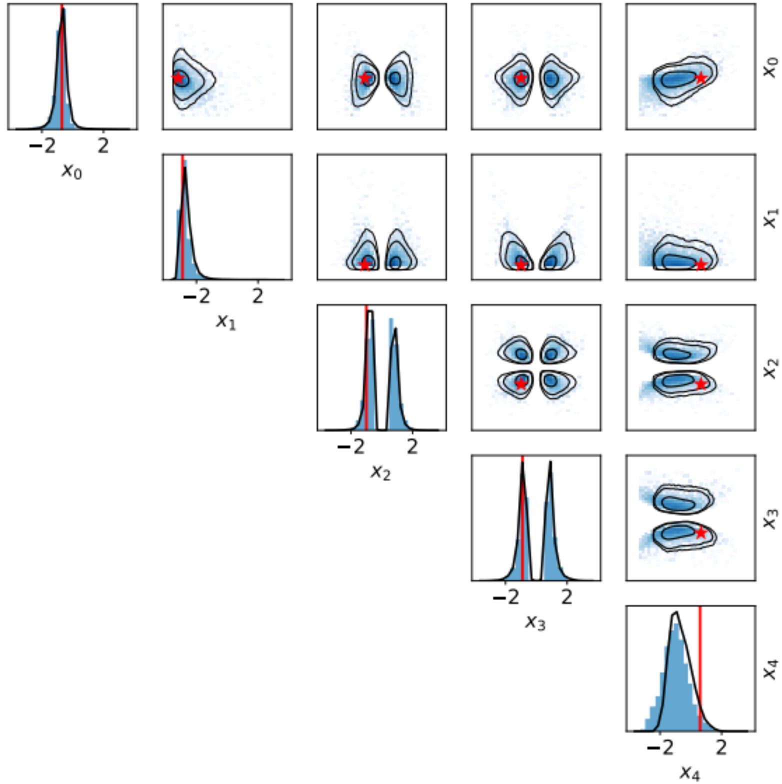


Figure 4: Posterior distribution obtained from MCMC with the exact source distribution and the exact likelihood function on SLCP in blue against the posterior distribution obtained with $q_\phi(\mathbf{y}|\mathbf{x})$ and $q_\theta(\mathbf{x})$ learned from $\mathcal{L}_{1024}$ in black (the 68-95-99.7% contours are shown). Generating source sample $\mathbf{x}$ are indicated in red. *The approximated posterior distribution closely matches the ground truth.*

—

Credits: Vandegar et al, 2021.

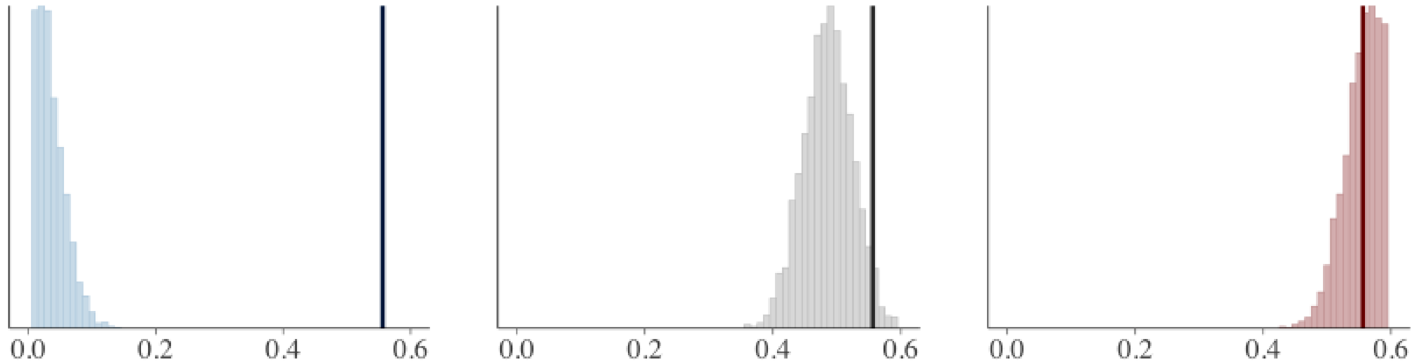
Posterior predictive checks

If a model is a good fit, then we should be able to use it to generate data that resemble the data we observe.

Formally, this can be diagnosed with posterior predictive checks that generates data x^{sim} according to the posterior predictive distribution

$$p(x^{\text{sim}}|x) = \int p(x^{\text{sim}}|\theta)p(\theta|x)d\theta,$$

or summary statistics $T(x^{\text{sim}})$ thereof.

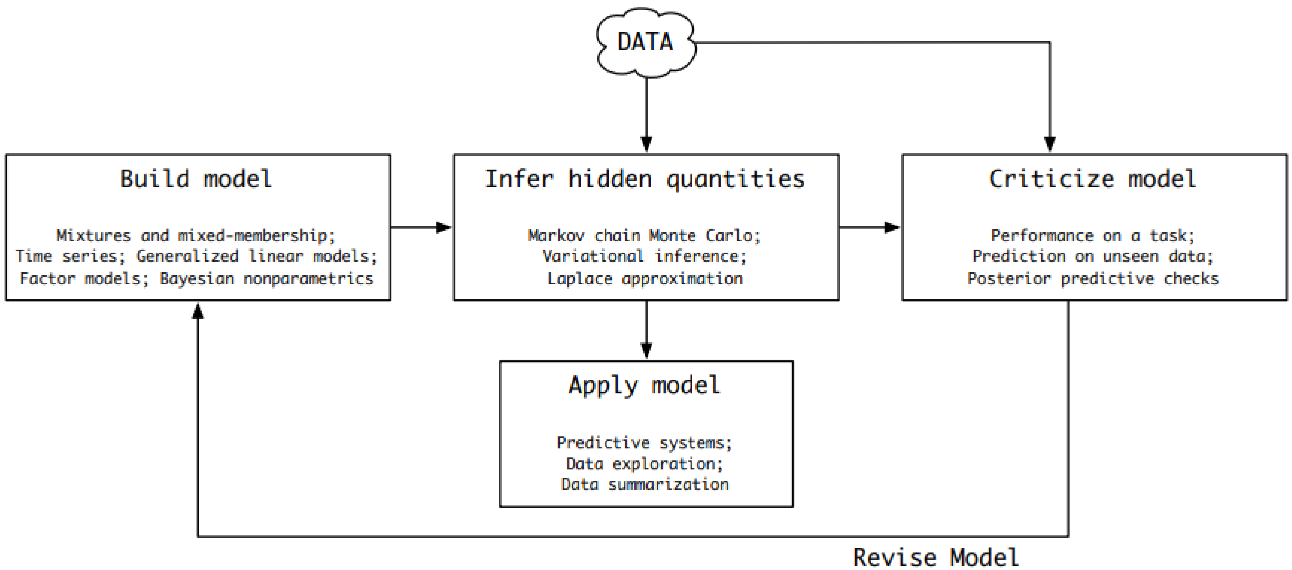


(a) Model 1 (b) Model 2 (c) Model 3

Fig. 7: *Histograms of statistics* skew(y_{rep}) *computed from 4000 draws from the posterior predictive distribution. The dark vertical line is computed from the observed data. These plots can be produced using* `ppc_stat` *in the* `bayesplot` *package.*

—
Credits: Gabry et al, 2017.

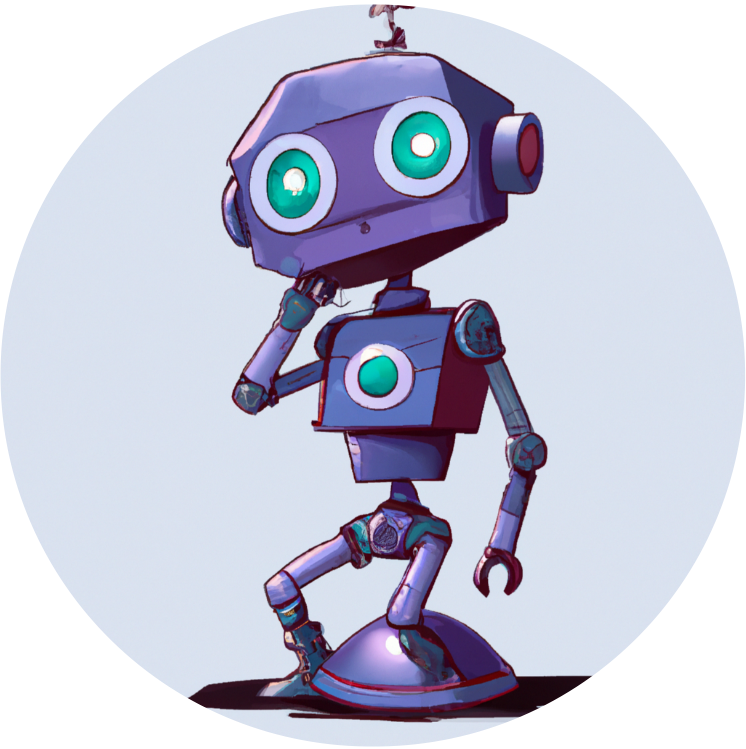
Box's loop: build, compute, critique, repeat



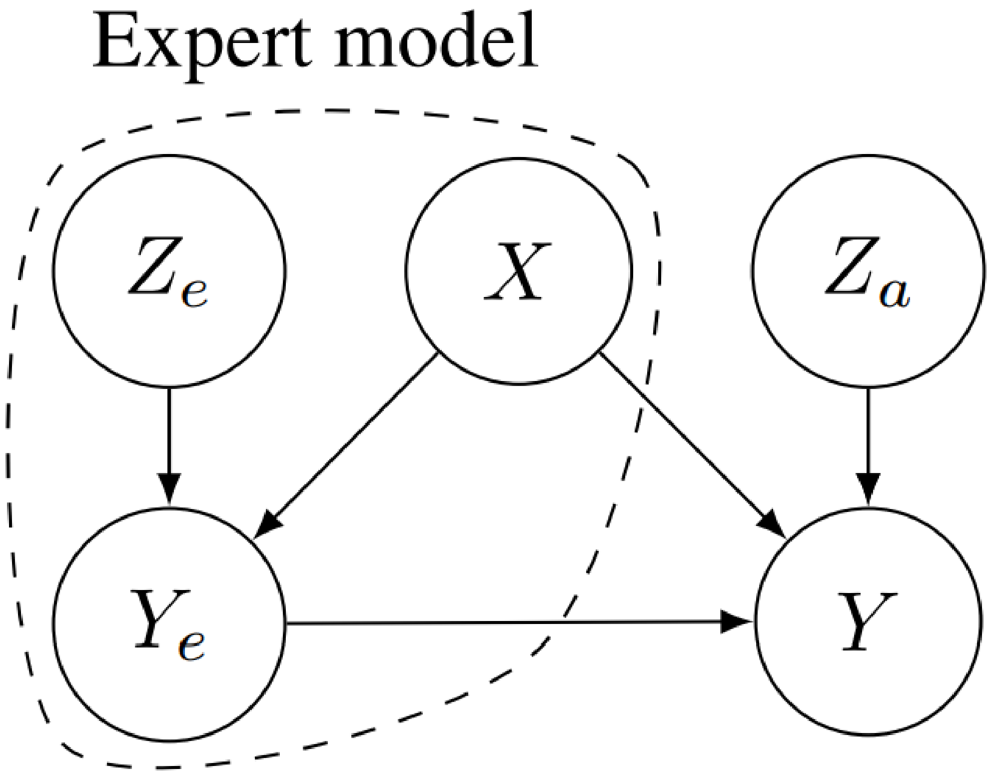
Science does not end at the inference results. Instead, they should inform the next revision of the model.

—

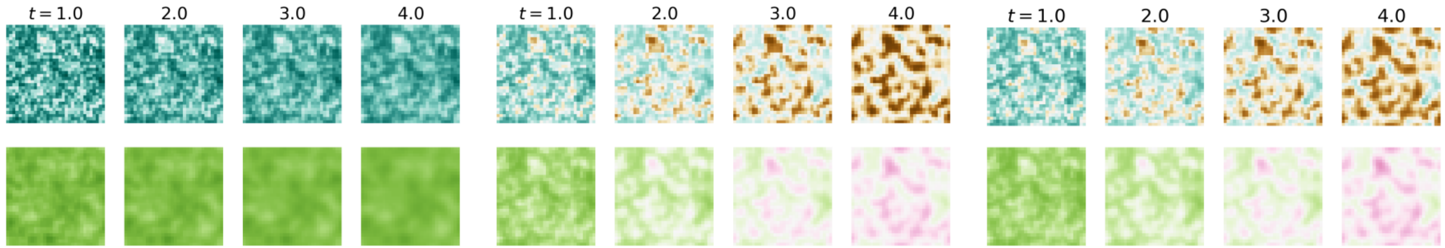
Credits: Blei, 2014.



Wait a minute... Can't I machine learn the model discrepancy?



Credits: Wehenkel et al, 2022.



(a) Param PDE (a, b), diffusion-only

(b) APHYNITY Param PDE (a, b)

(c) Ground truth simulation

—
Credits: Yin et al, 2021.

Summary

Simulation-based inference is a major evolution in the statistical capabilities for science, enabled by advances in machine learning.

Need to reliably and efficiently evaluate the quality of the posterior approximations.

Further advances will eventually augment incomplete physical models with AI.

SBI beyond Science?

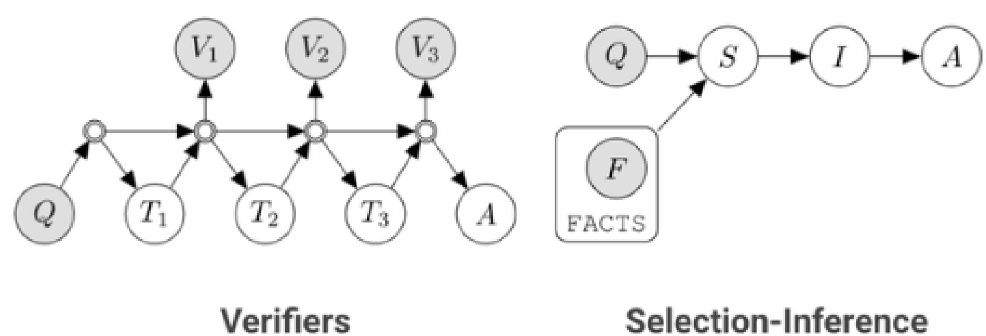
Jascha Sohl-Dickstein
@jaschasd

I think we will increasingly build systems out of many large models interacting with each other. I think the cascades perspective -- write down a probabilistic graphical model, but with every node a language model -- is the right formalism for describing these systems.

David Dohan @dmdohan · Jul 23

Happy to release our work on Language Model Cascades. Read on to learn how we can unify existing methods for interacting models (scratchpad/chain of thought, verifiers, tool-use, ...) in the language of probabilistic programming.

paper: arxiv.org/abs/2207.10342



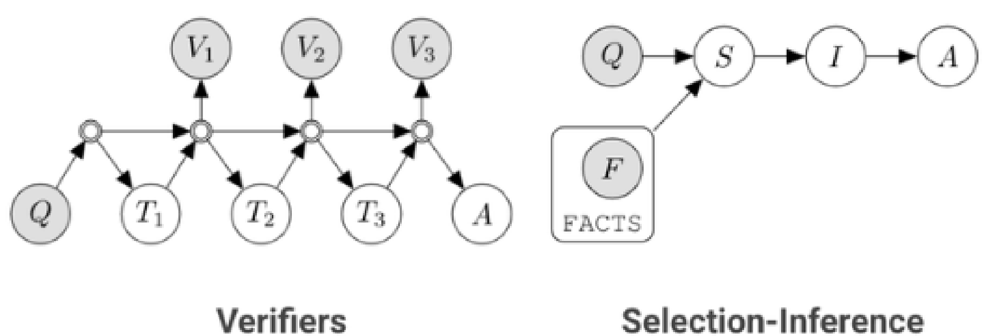
SBI beyond Science?

Jascha Sohl-Dickstein
@jaschasd

I think we will increasingly build systems out of many large models interacting with each other. I think the cascades perspective -- write down a probabilistic graphical model, but with every node a language model -- is the right formalism for describing these systems.

tor, directly into the program. Then techniques from simulation based inference, for example, can be applied to do inference in such situations (Cranmer et al., 2020).

paper: arxiv.org/abs/2207.10342



The end.