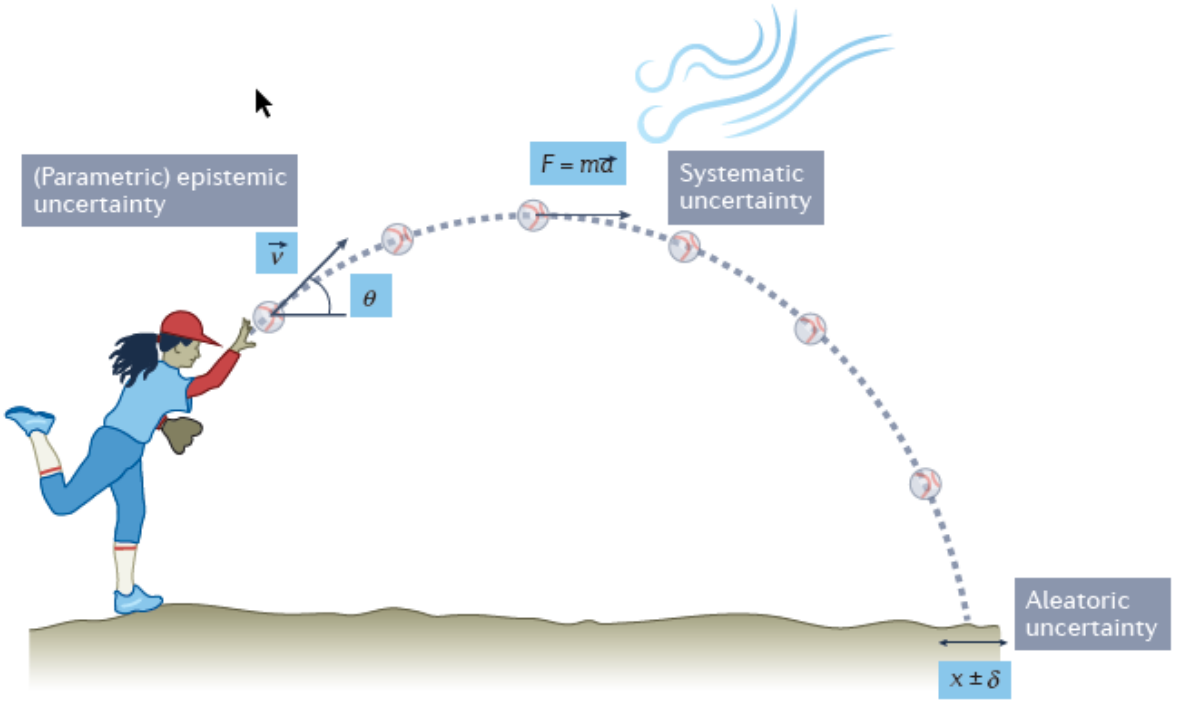
An introduction to simulation-based inference

Prof. Gilles Louppe
g.louppe@uliege.be

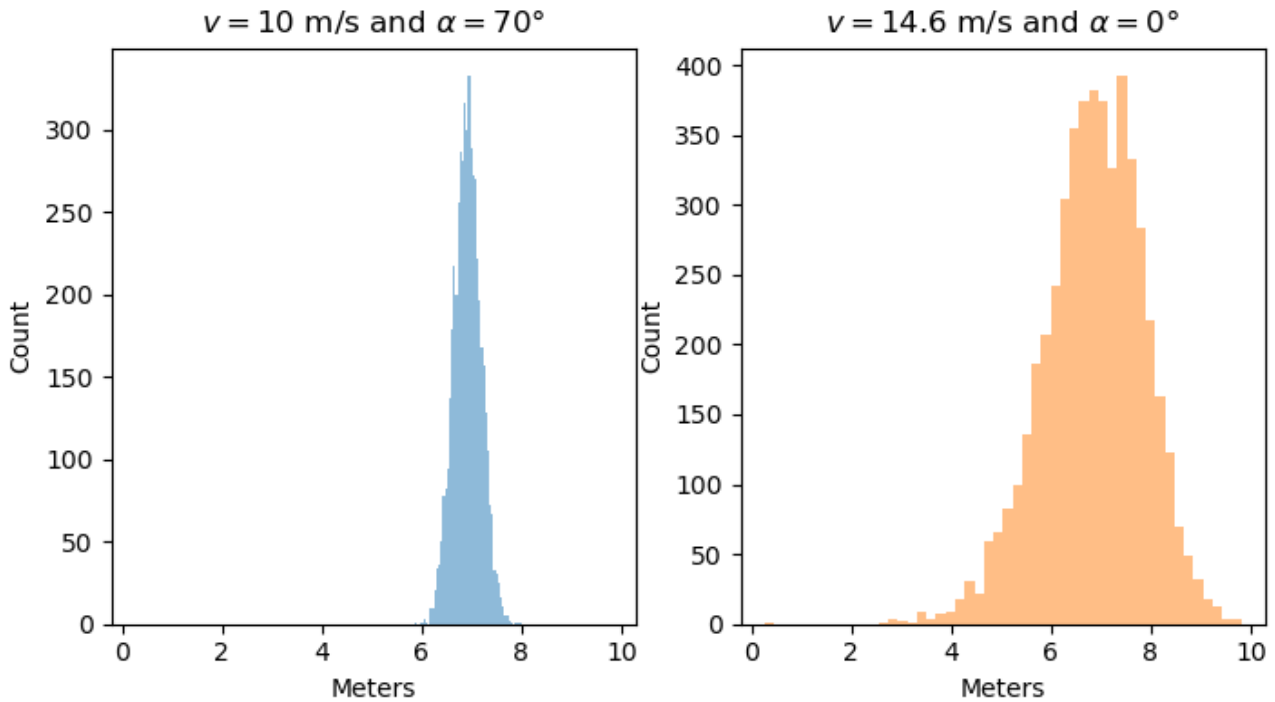
LIÈGE université


(Parametric) epistemic uncertainty
$\vec{v}$
θ
$F = m\vec{a}$
Systematic uncertainty
Aleatoric uncertainty
$x \pm \delta$

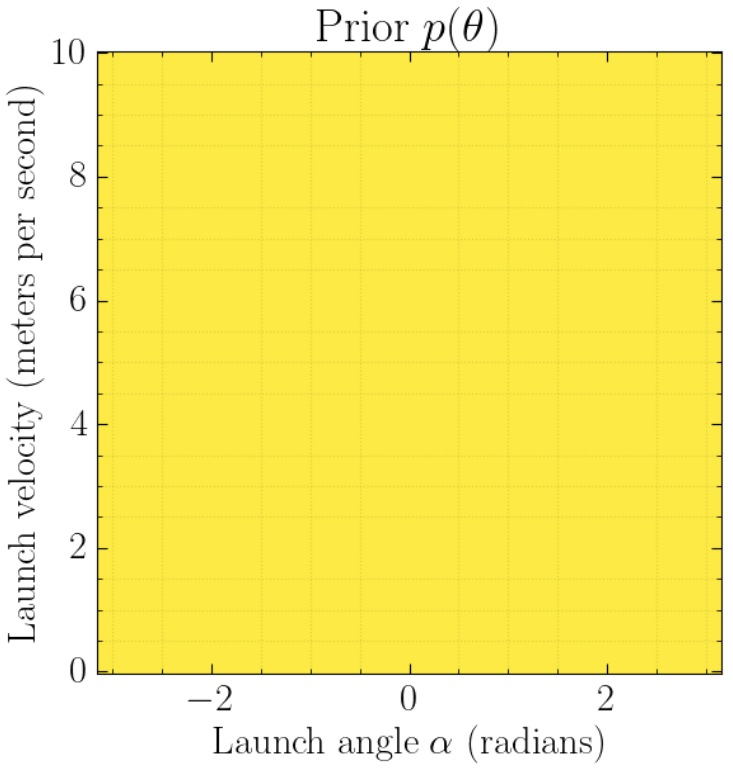
$$v_x = v\cos(\alpha), \quad v_y = v\sin(\alpha),$$

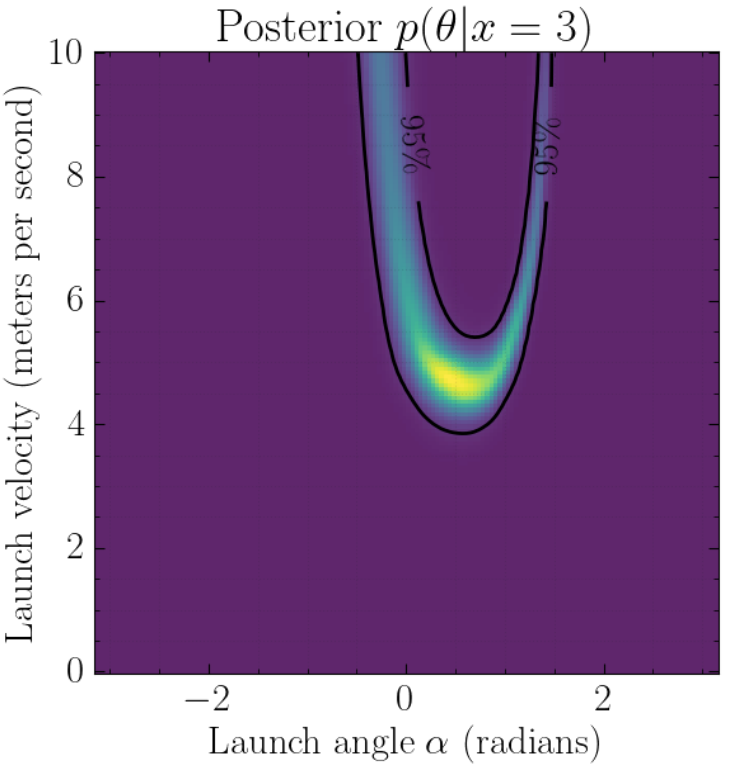
$$\frac{dx}{dt} = v_x, \quad \frac{dy}{dt} = v_y, \frac{dv_y}{dt} = -G.$$

```
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()
```


$v = 10$ m/s and $\alpha = 70°$
Count
Meters
0
50
100
150
200
250
300
0
2
4
6
8
10
$v = 14.6$ m/s and $\alpha = 0°$
Count
Meters
0
50
100
150
200
250
300
350
400
0
2
4
6
8
10

What parameter values θ are the most plausible?


Prior $p(\theta)$
Launch velocity (meters per second)
Launch angle α (radians)
0
2
4
6
8
10
−2
0
2


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

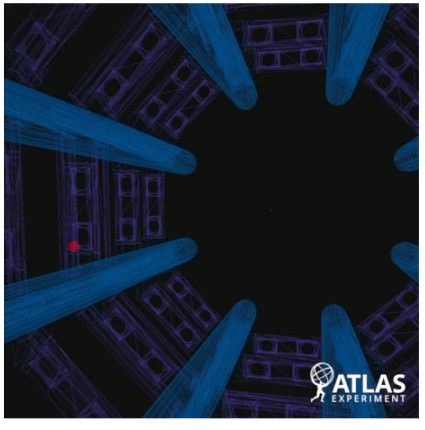
Simulation-based inference

Simulators as generative models

A simulator prescribes a generative model that can be used to simulate data $\mathbf{x}$.

Collider data

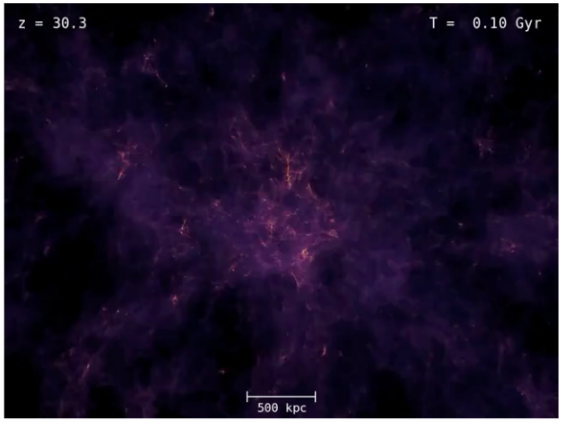
$\text{particles} \sim p(\text{particles})$



[C. Cesarotti with ATLAS]

Cosmology data

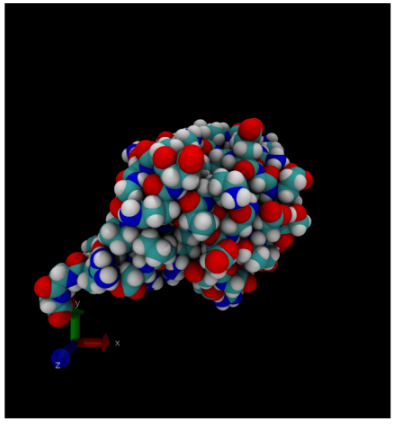
$\text{particles} \sim p(\text{particles})$



[Aquarius simulation]

Molecular dynamics

$\text{configurations} \sim p(\text{configurations})$



[E. Cances et al]

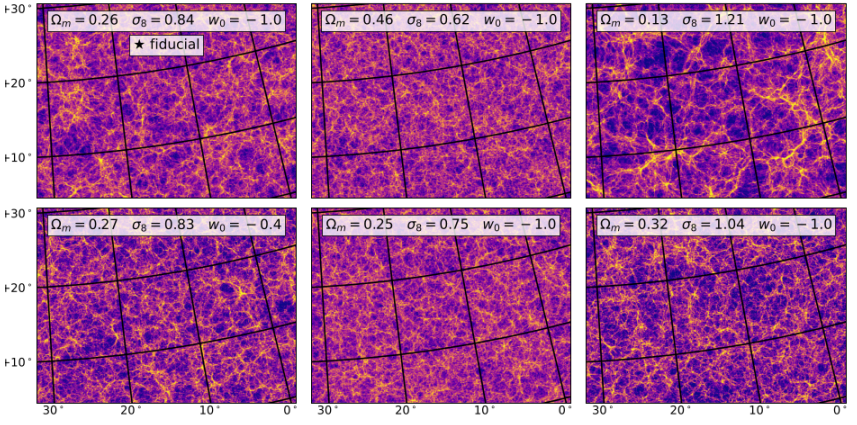
—

Conditional simulators

A conditional simulator prescribes a way to sample from the likelihood $p(\mathbf{x}|\theta)$, where θ is a set of conditioning variables or parameters.

Cosmology data

$$\text{map} \sim p(\text{map} \mid \{\Omega_m, \sigma_8, w_0\})$$



[Kacprzak et al 2022]

$$x \sim p(x; \mathscr{M})$$

Model

or

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

Model parameters

What can we do with generative models?

Produce samples and make predictions

$$\mathbf{x} \sim p(\mathbf{x}|\theta)$$

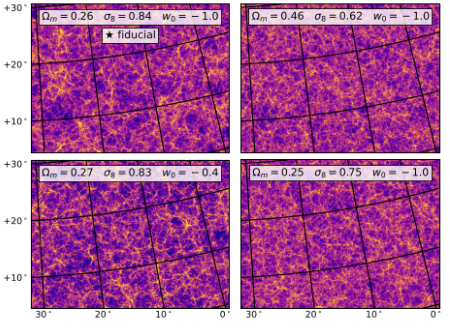
Evaluate densities

$$p(\mathbf{x}|\theta)$$

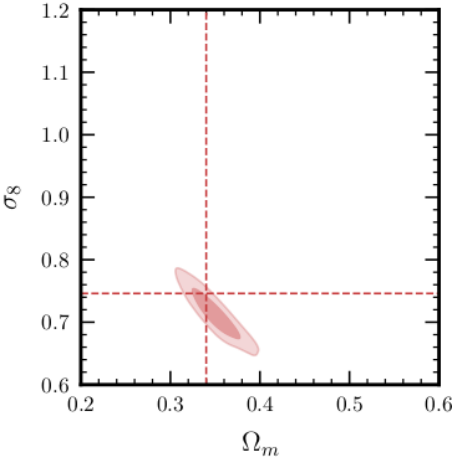
$$p(\theta|\mathbf{x}) = \frac{p(\mathbf{x}|\theta)p(\theta)}{p(\mathbf{x})}$$

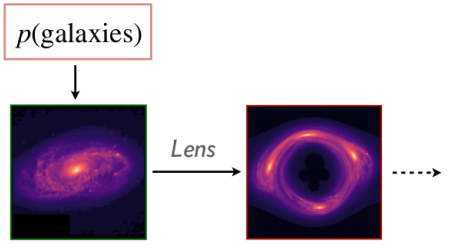
Encode complex priors

$$p(\mathbf{x})$$

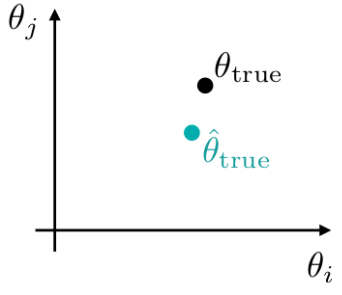


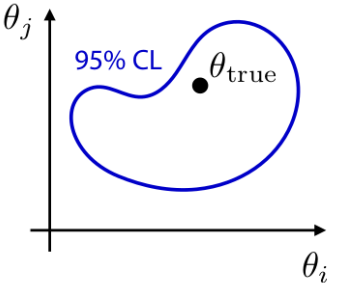
[Kacprzak et al 2022]

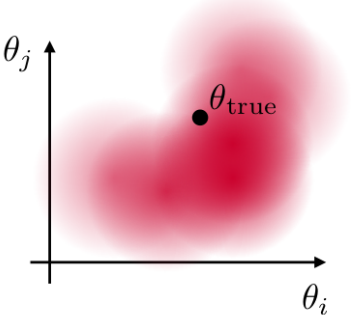




Inference







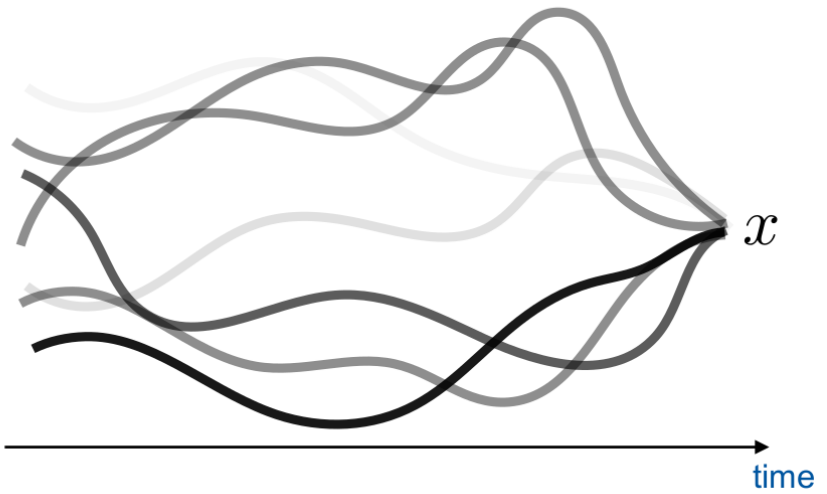
- Frequentist inference: find $\hat{\theta}$ that maximizes the likelihood $p(\mathbf{x}|\theta)$ or build a confidence interval thereof.
- Bayesian inference: compute the posterior distribution $p(\theta|\mathbf{x}) = \frac{p(\mathbf{x}|\theta)p(\theta)}{p(\mathbf{x})}$.

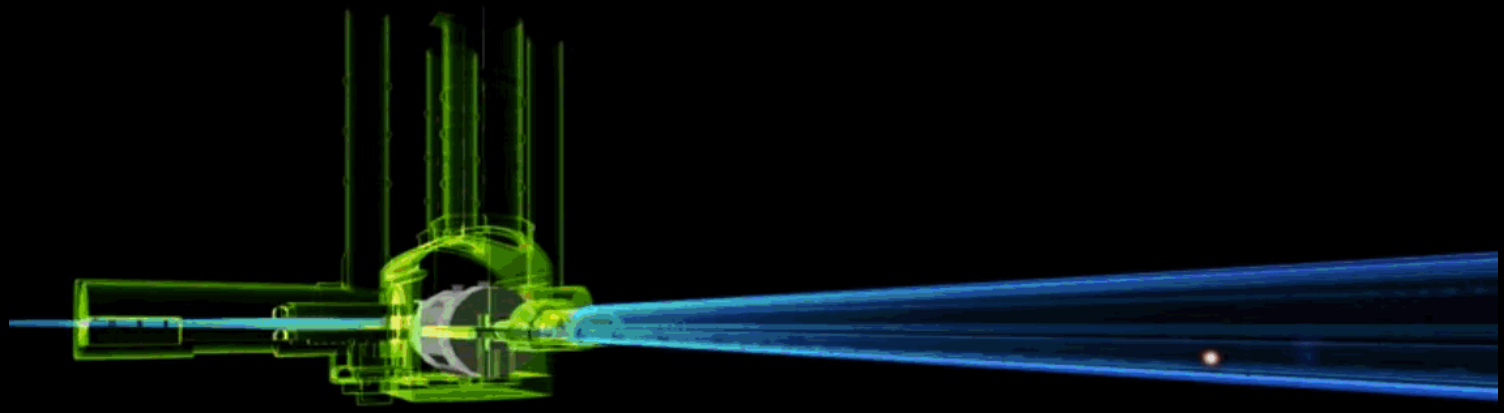
Intractable likelihoods

The (modeled) data generating process may involve additional latent variables $\mathbf{z}$ that are not observed, leading to likelihoods

$$p(\mathbf{x}|\theta) = \int p(\mathbf{x}, \mathbf{z}|\theta) d\mathbf{z}.$$

In this case, evaluating the likelihood becomes intractable.





$$p(\mathbf{z}_p|\theta)$$

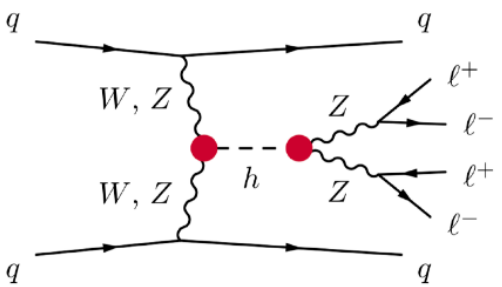
Latent variables

Parameters of interest

Parton-level momenta

Theory parameters

$z_p \longleftarrow \theta$



$$p(\mathbf{z}_s|\theta) = \int p(\mathbf{z}_p|\theta)p(\mathbf{z}_s|\mathbf{z}_p)d\mathbf{z}_p$$

Latent variables

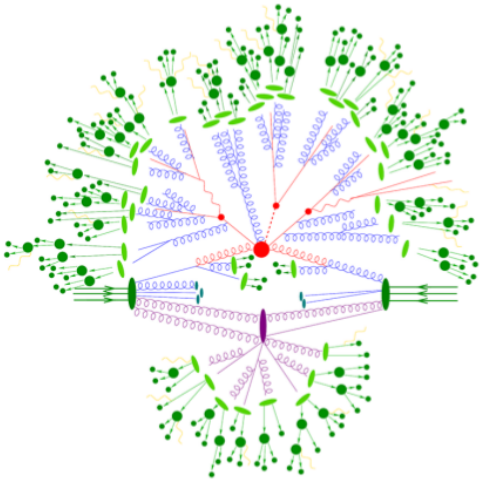
Parameters of interest

Shower splittings

Parton-level momenta

Theory parameters

$z_s \longleftarrow z_p \longleftarrow \theta$



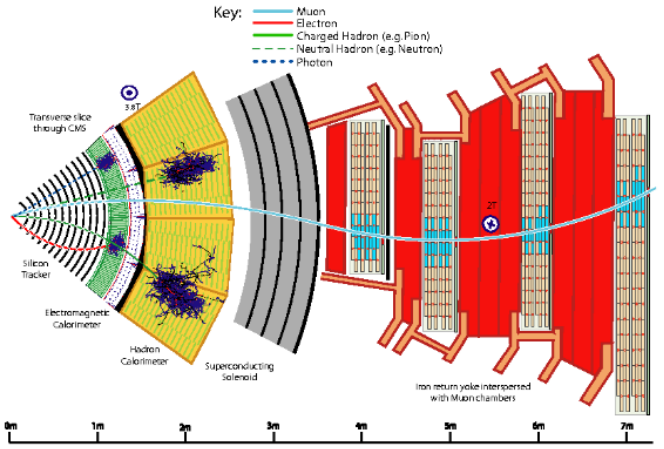
$$p(\mathbf{z}_d|\theta) = \iint p(\mathbf{z}_p|\theta)p(\mathbf{z}_s|\mathbf{z}_p)p(\mathbf{z}_d|\mathbf{z}_s)d\mathbf{z}_p d\mathbf{z}_s$$

Latent variables

Parameters of interest

Detector interactions | Shower splittings | Parton-level momenta | Theory parameters

$z_d \longleftarrow z_s \longleftarrow z_p \longleftarrow \theta$

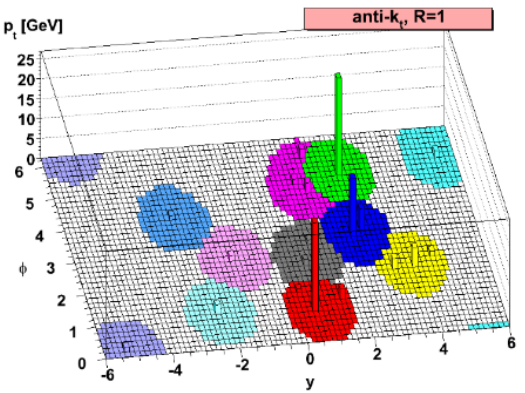


$$p(\mathbf{x}|\theta) = \underbrace{\iiint}_{\text{yikes!}} p(\mathbf{z}_p|\theta)p(\mathbf{z}_s|\mathbf{z}_p)p(\mathbf{z}_d|\mathbf{z}_s)p(\mathbf{x}|\mathbf{z}_d)d\mathbf{z}_p d\mathbf{z}_s d\mathbf{z}_d$$

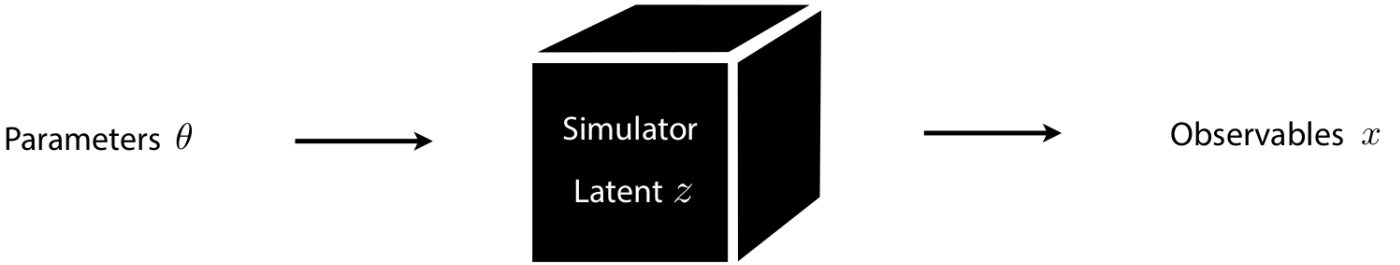
Features | Latent variables | Parameters of interest

Observables | Detector interactions | Shower splittings | Parton-level momenta | Theory parameters

$x \leftarrow z_d \leftarrow z_s \leftarrow z_p \leftarrow \theta$



[Image source: M. Cacciari, G. Salam, G. Soyez 0802.1189]

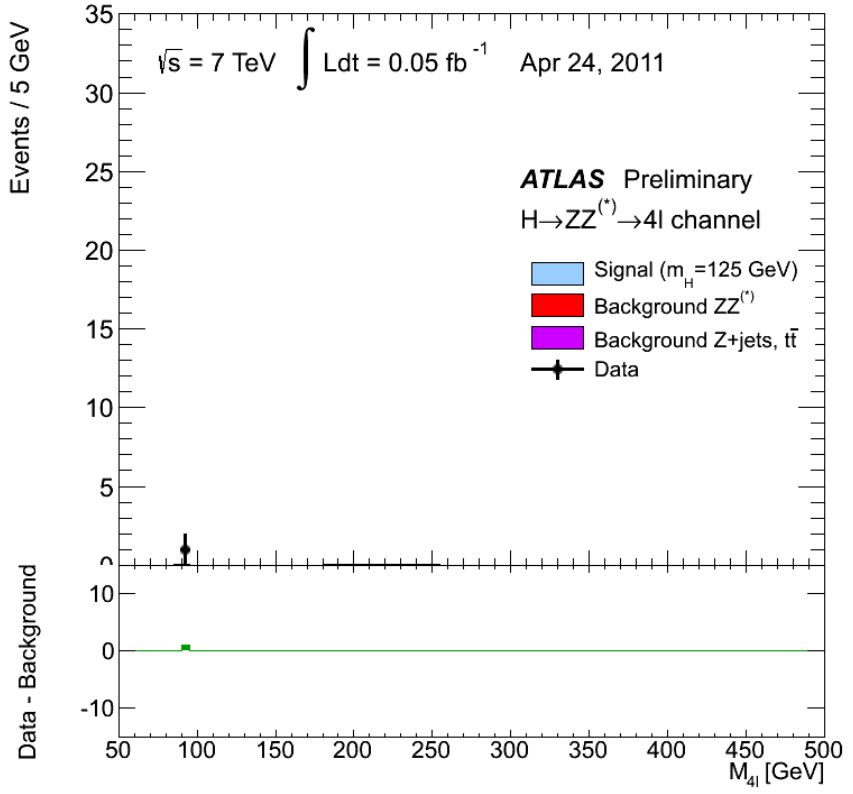


Prediction:
- Well-motivated mechanistic, causal model
- Simulator can generate samples $x \sim p(x|\theta)$

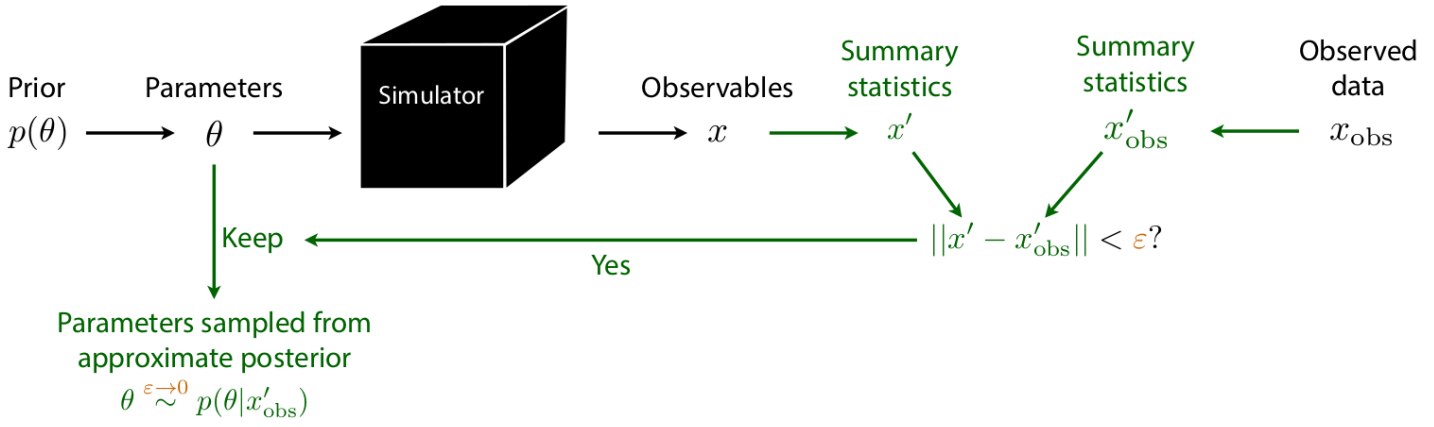
Inference:
- Interactions between low-level components lead to challenging inverse problems
- Likelihood $p(x|\theta) = \int \mathrm{d}z\; p(x, z|\theta)$ is intractable

Statistical inference becomes challenging when the likelihood $p(\mathbf{x}|\theta)$ is implicit or intractable. **Simulation-based inference algorithms are needed.**

pre-2019



(Frequentist) Approximate the likelihood $p(\mathbf{x}|\theta)$ as
$p(\mathbf{x}|\theta) \approx \hat{p}(\mathbf{x}|\theta) = p(s(\mathbf{x})|\theta)$ for some (well-chosen) summary statistic $s(\cdot)$.



(Bayesian) Approximate the posterior $p(\theta|\mathbf{x})$ using Approximate Bayesian Computation.

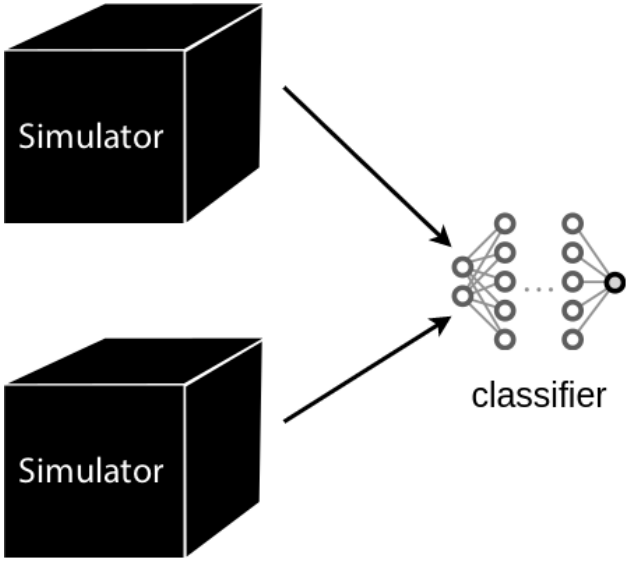
Issues:

- How to choose $\mathbf{x}' = s(\mathbf{x})$? ϵ? $||\cdot||$?
- No tractable posterior.
- Need to run new simulations for new data or new prior.

Neural ratio estimation (NRE)

The likelihood-to-evidence $r(\mathbf{x}|\theta) = \frac{p(\mathbf{x}|\theta)}{p(\mathbf{x})} = \frac{p(\theta,\mathbf{x})}{p(\theta)p(\mathbf{x})}$ ratio can estimated from a binary classifier $d(\theta, \mathbf{x})$, even if neither the likelihood nor the evidence can be evaluated.

$\theta, \mathbf{x} \sim p(\theta, \mathbf{x})$



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

$\theta, \mathbf{x} \sim p(\theta)p(\mathbf{x})$

—

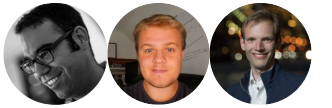
Credits: Cranmer et al, 2015; Hermans et al, 2020.

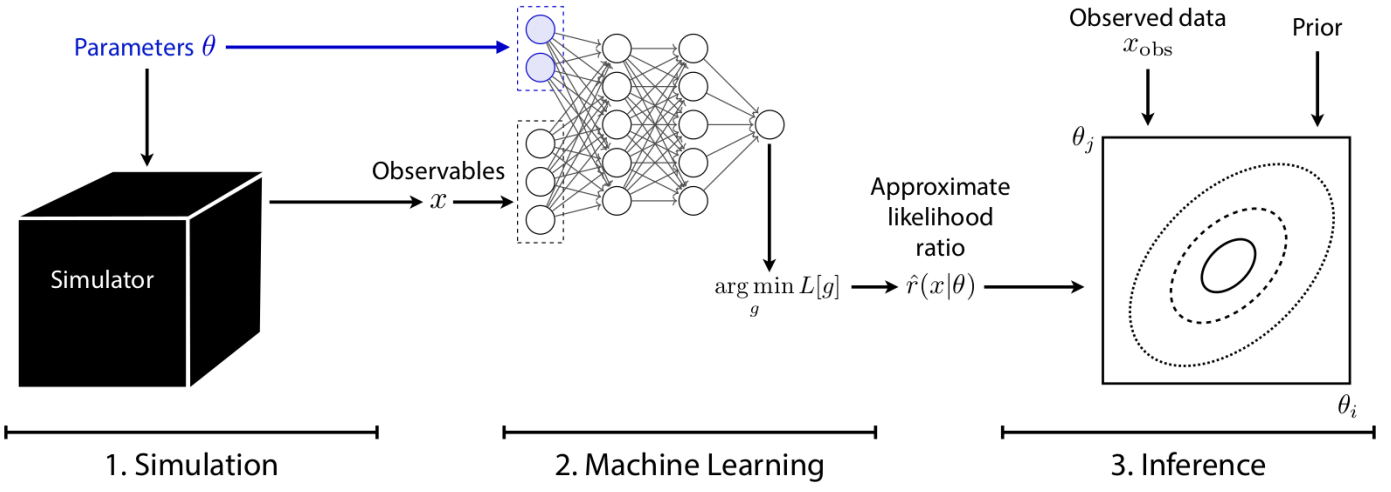
The solution d found after training approximates the optimal classifier

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

Therefore,

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





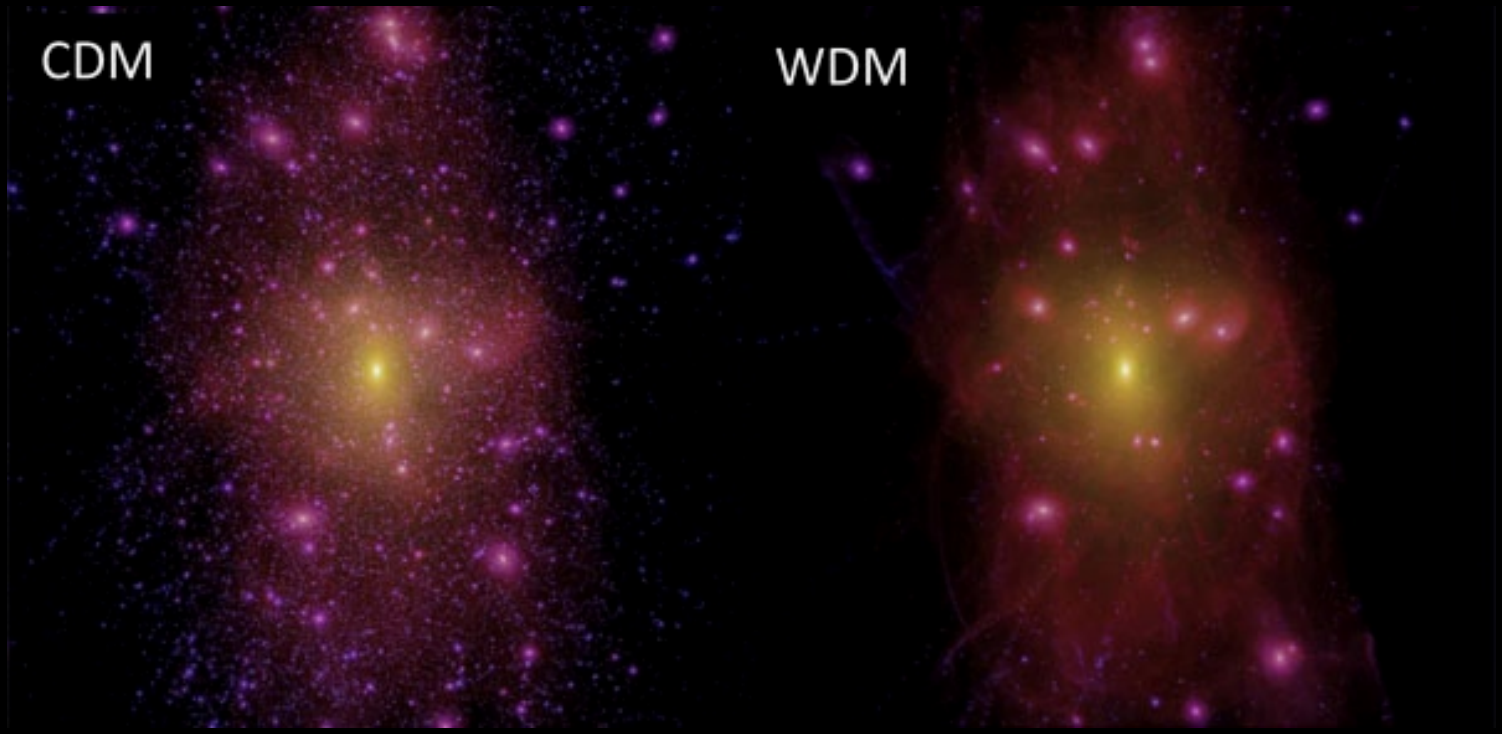
Run simulator and save data

Train NN classifier, interpret as likelihood ratio estimator

Amortized: cheap to repeat for new data

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

A case study (Hermans et al, 2021)



Can we constrain the nature of dark matter from cosmological observations?

Constraining dark matter with stellar streams

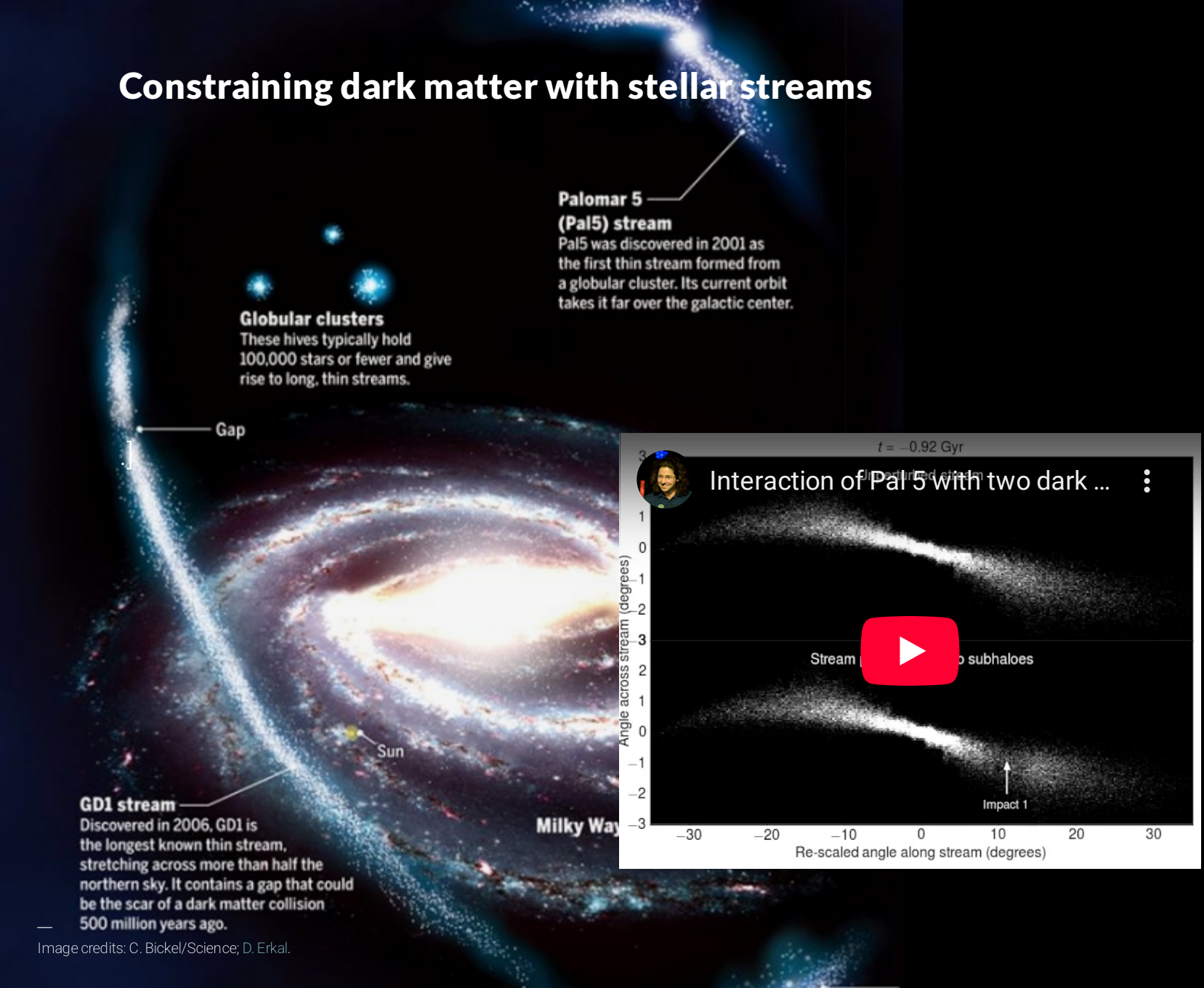
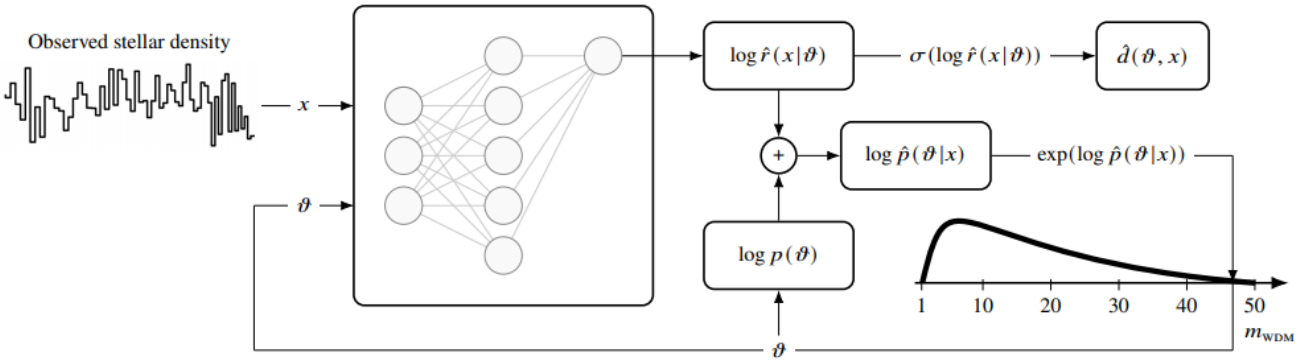
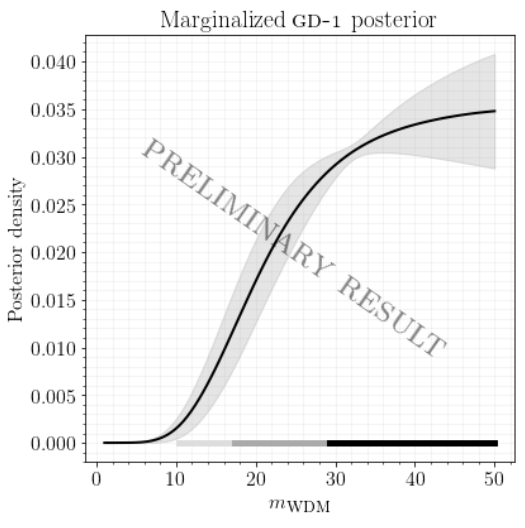


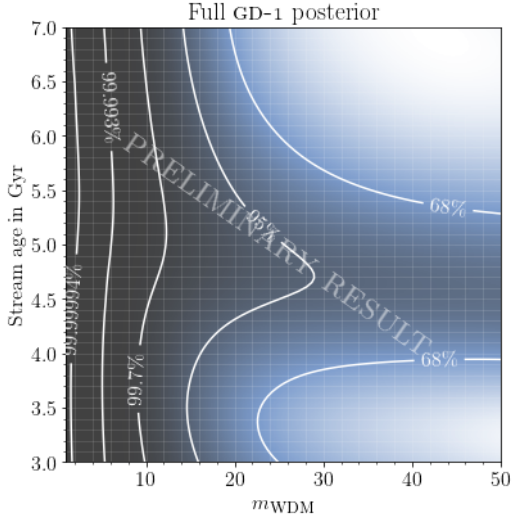
Image credits: C. Bickel/Science; D. Erkal.

$$p(m_{\rm WDM}, t_{\rm age}|{\rm GD1}) = \frac{p({\rm GD1}|m_{\rm WDM}, t_{\rm age})p(m_{\rm WDM}, t_{\rm age})}{p({\rm GD\text{-}1})}$$

NRE for stellar streams







—

Credits: Hermans et al, 2021.

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

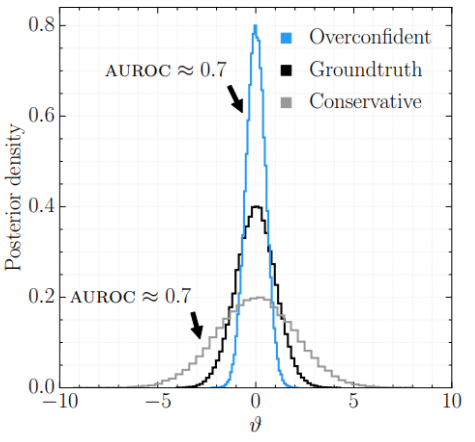
Wait a minute Gilles...
I can't claim that in a paper!
Your neural network must be wrong!

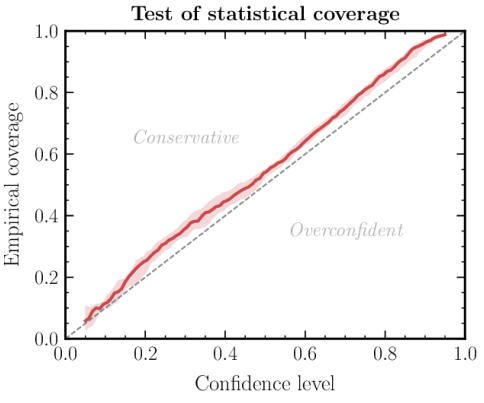
Expected coverage

$$\mathrm{EC}(\hat{p}, \alpha) = \mathbb{E}_{p(\theta,\mathbf{x})}[\theta \in \Theta_{\hat{p}(\theta|\mathbf{x})}(\alpha)]$$

If the expected coverage is close to the nominal coverage probability α, then the approximate posterior $\hat{p}$ is calibrated.

- If $\mathrm{EC} < \alpha$, then the posterior is underdispersed and overconfident.
- If $\mathrm{EC} > \alpha$, then the posterior is overdispersed and conservative.

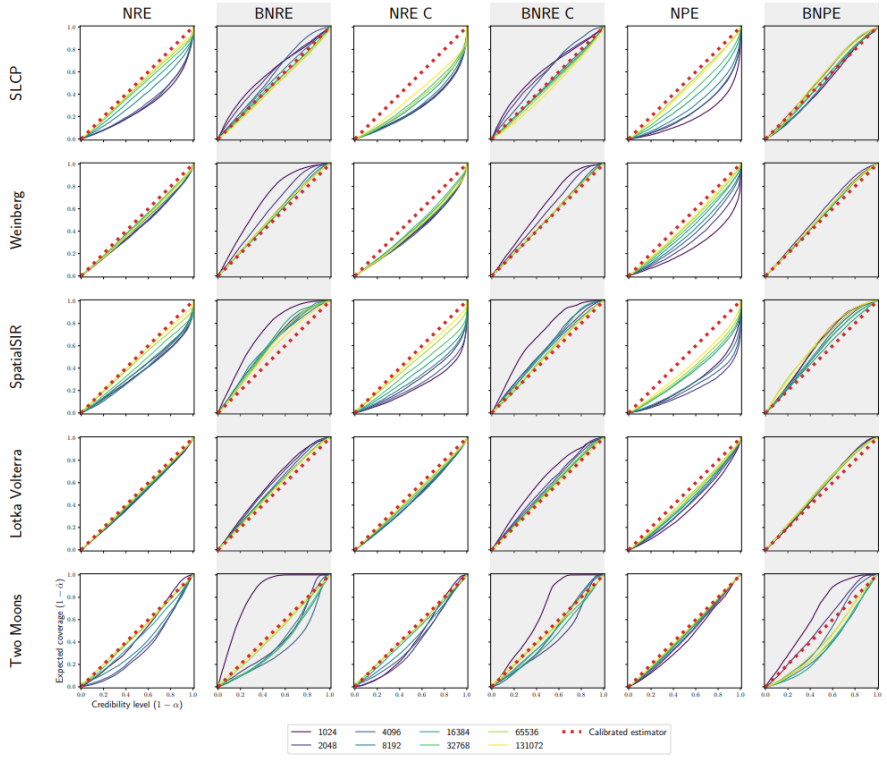




—

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

Balancing inference for conservative posteriors

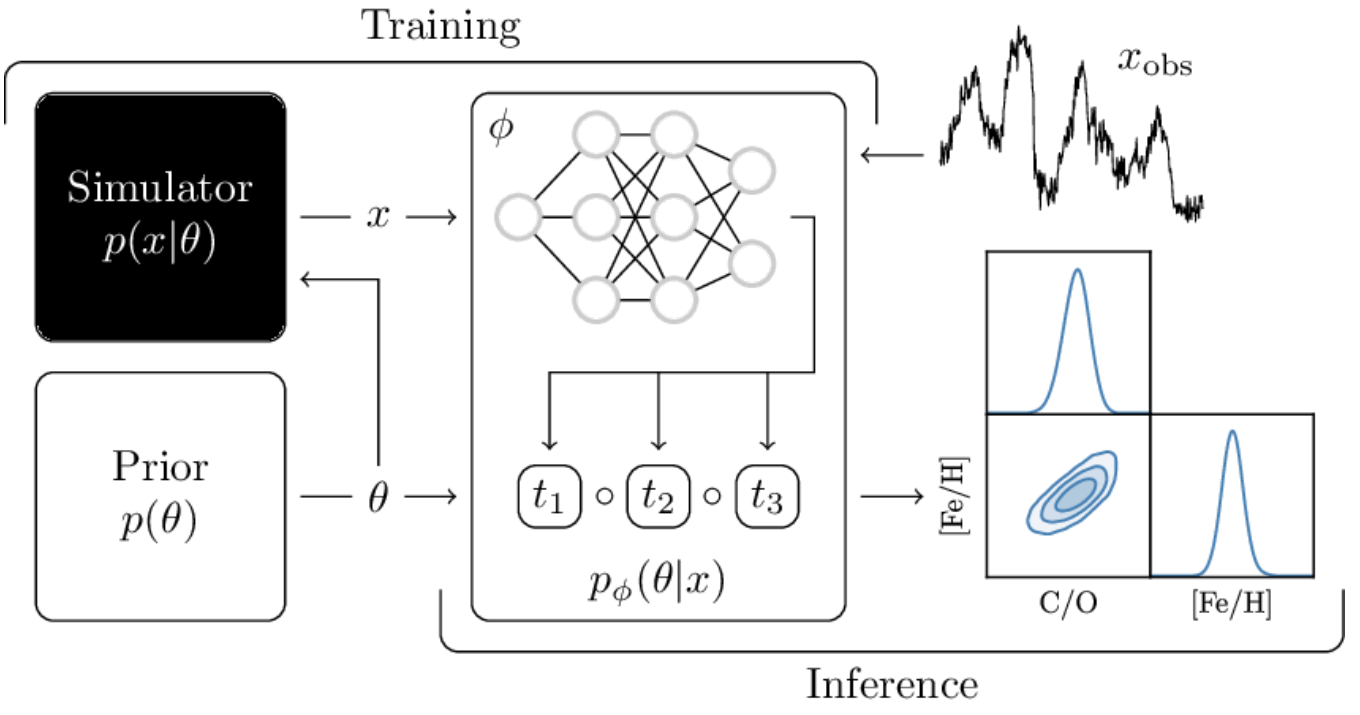


Conservative posteriors can be obtained by enforcing d to be balanced, i.e. such that $\mathbb{E}_{p(\theta,\mathbf{x})}\left[d(\theta,\mathbf{x})\right] = \mathbb{E}_{p(\theta)p(\mathbf{x})}\left[1-d(\theta,\mathbf{x})\right]$.

—

Credits: Delaunoy et al, 2022; Delaunoy et al, 2023.

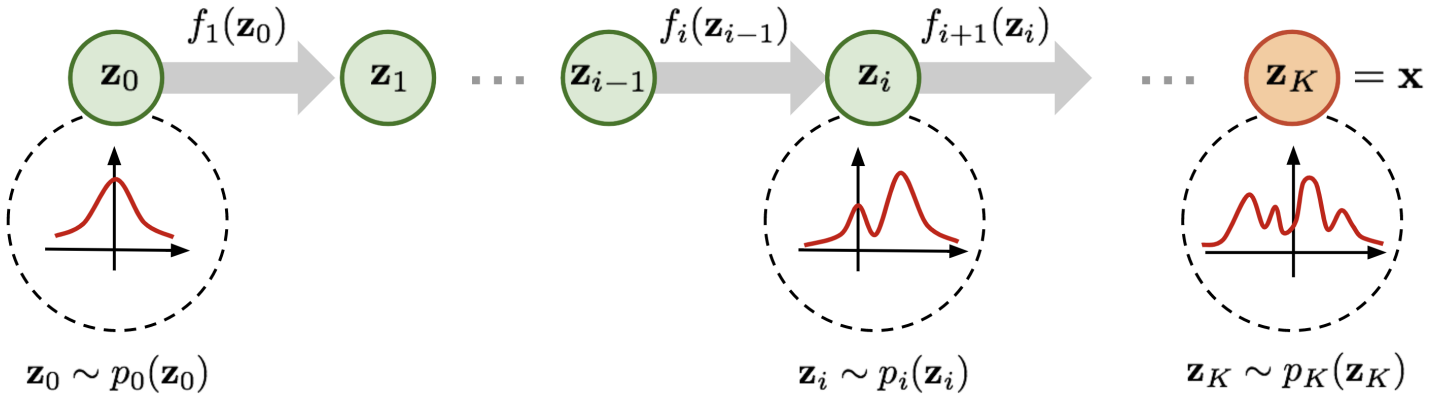
Neural Posterior Estimation (NPE)



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

Normalizing flows

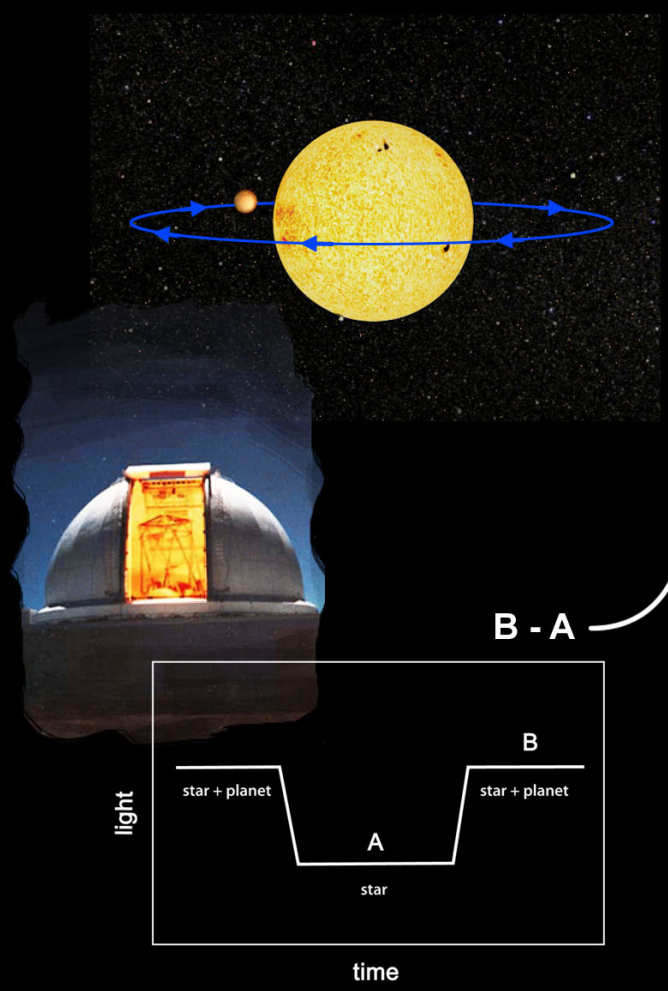
A normalizing flow is a sequence of invertible transformations f_k that map a simple distribution p_0 to a more complex distribution p_K:

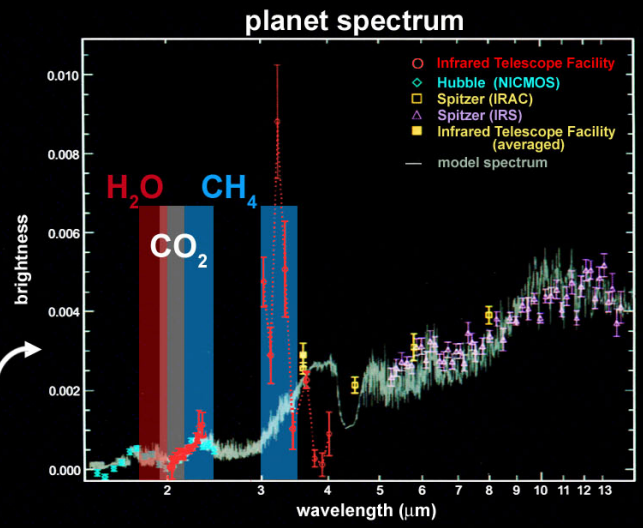


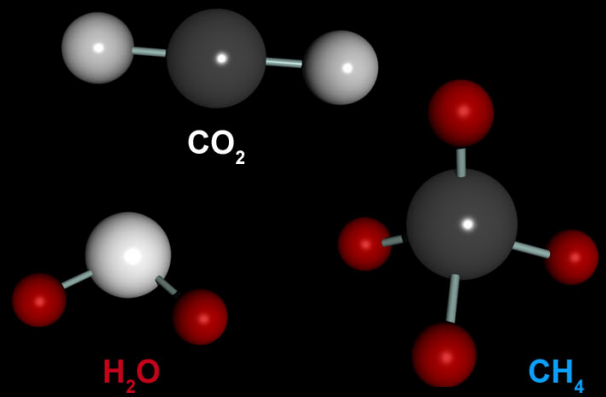
By the change of variables formula, the log-likelihood of a sample x is given by

$$\log p(\mathbf{x}) = \log p(\mathbf{z}_0) - \sum_{k=1}^{K} \log \left| \det J_{f_k}(\mathbf{z}_{k-1}) \right| .$$

Exoplanet atmosphere characterization

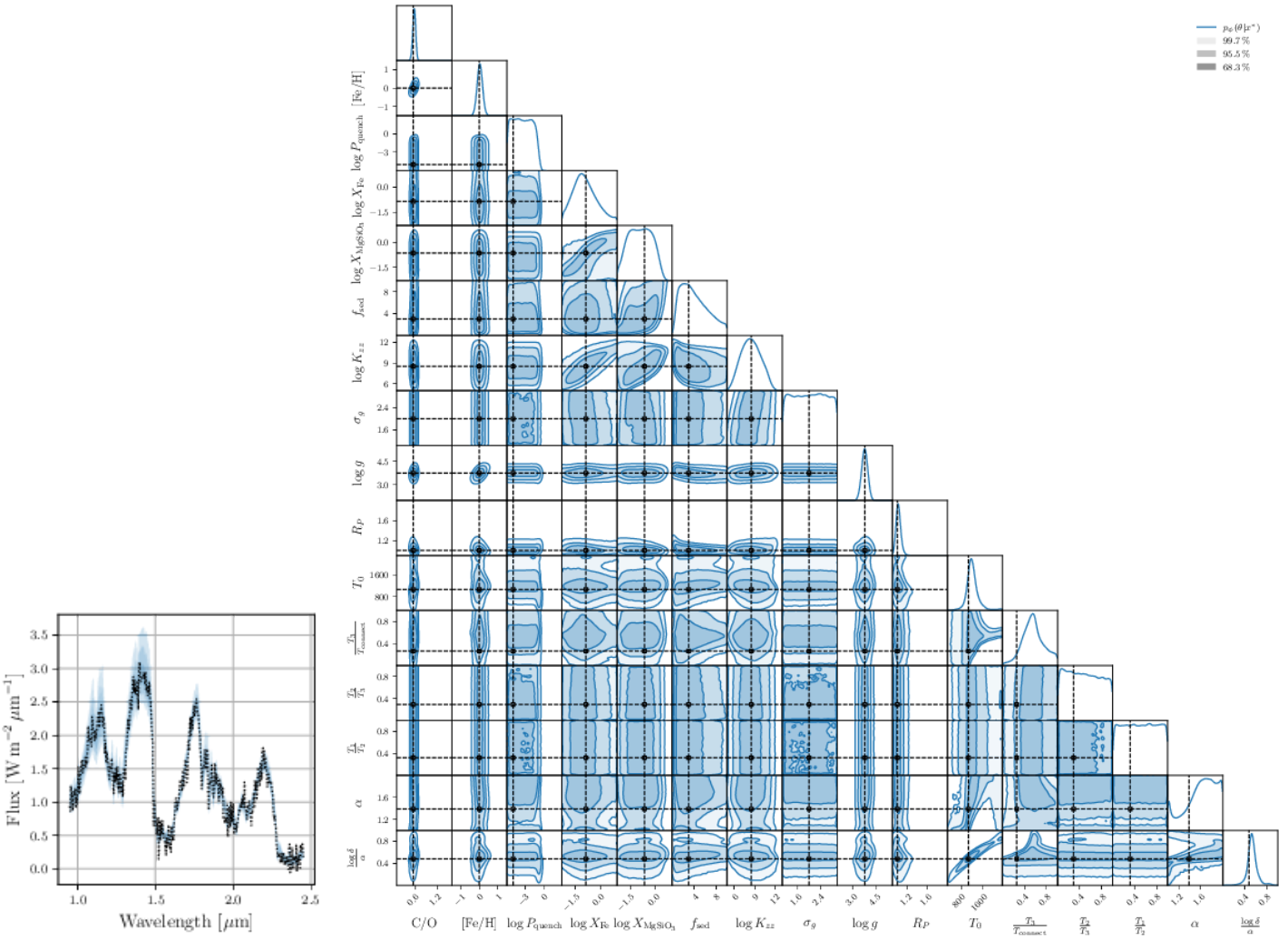






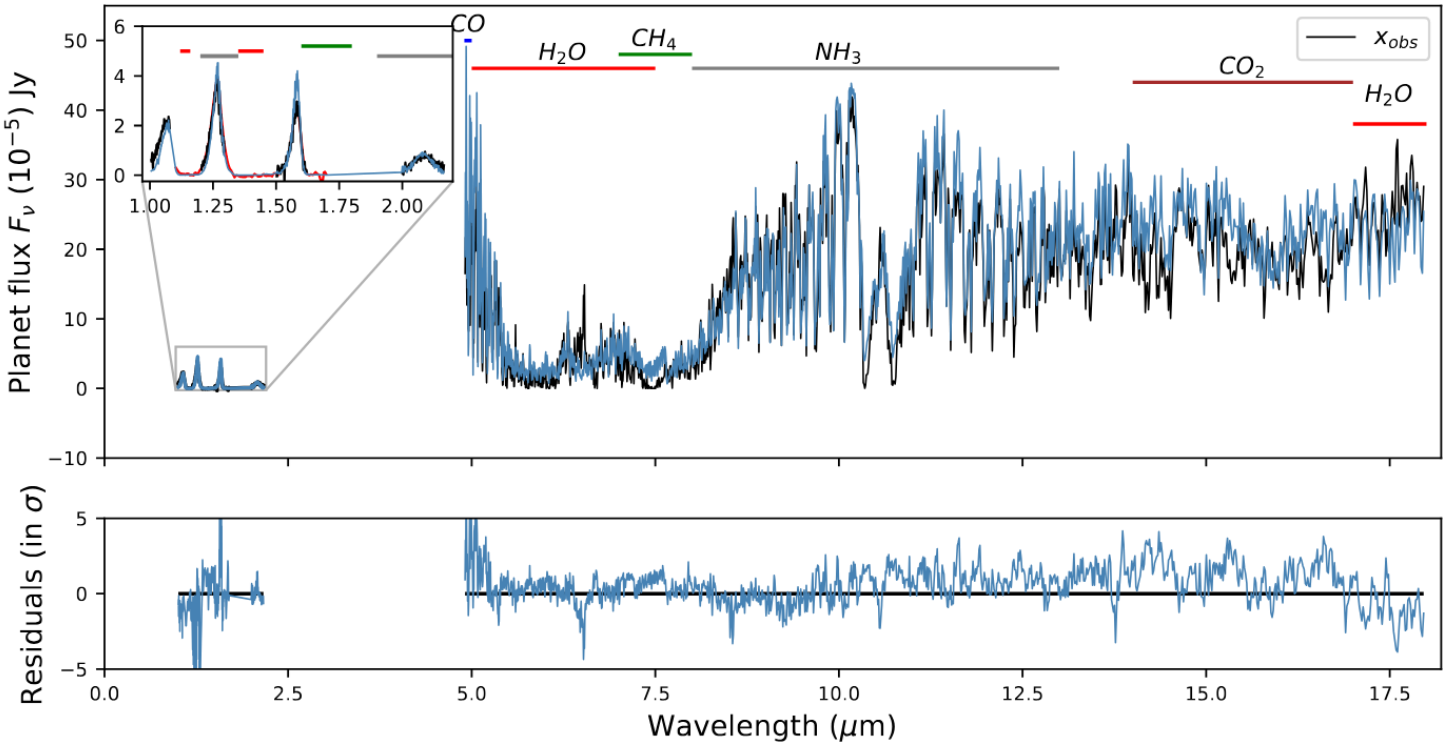
—

Credits: NSA/JPL-Caltech, 2010.



—

Credits: Vasist et al, 2023.

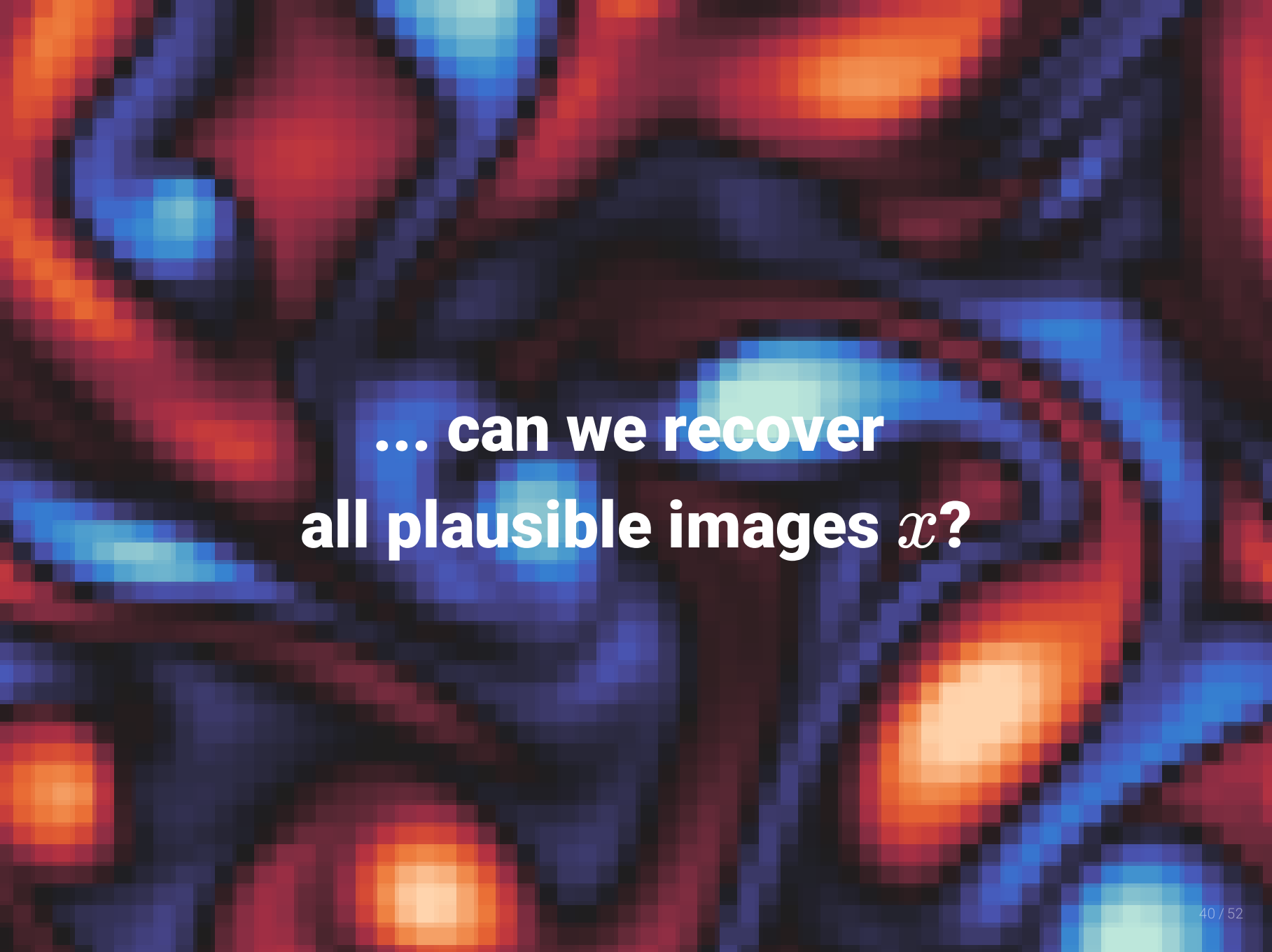


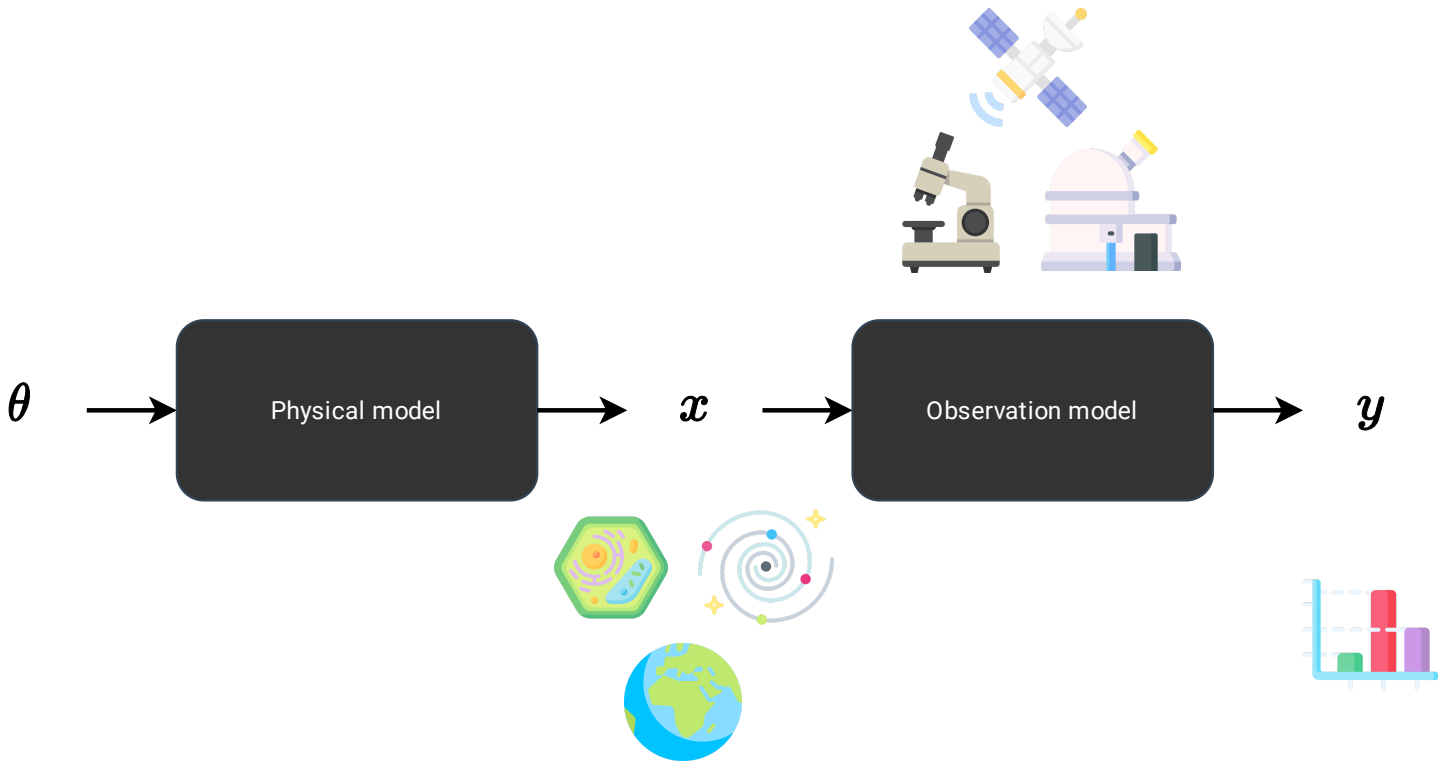
Panchromatic characterization of WISE 1738 using JWST/MIRI (Vasist et al, submitted).

Neural score estimation (NSE)

How do we estimate $p(x|y)$ when x is high-dimensional?

From a noisy observation y...


... can we recover
all plausible images x?



Problem statement

Given a noisy observation y, estimate the posterior distribution $p(x|y) = \frac{p(y|x)p(x)}{p(y)}$ of plausible latent states x.

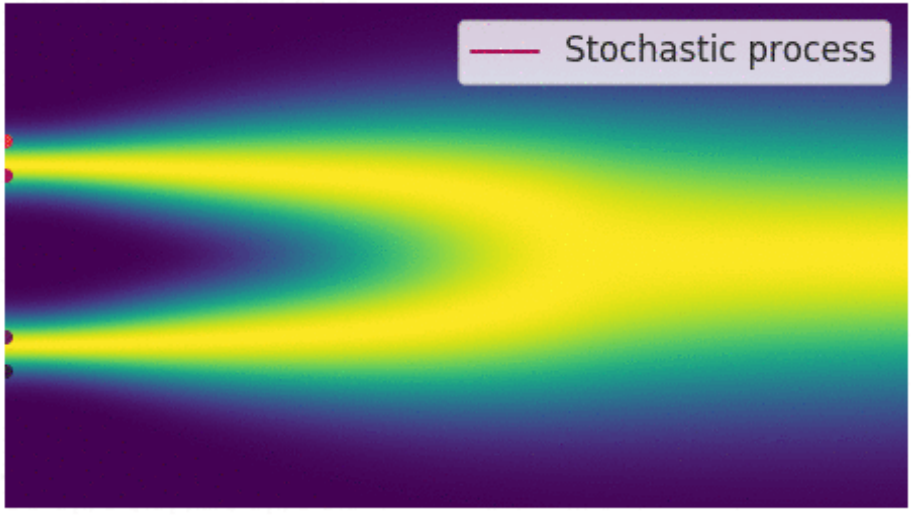
Score-based generative models 101

Samples $x \sim p(x)$ are progressively perturbed through a diffusion process described by the forward SDE

$$\mathrm{d}x_t = f_t x_t \mathrm{d}t + g_t \mathrm{d}w_t,$$

where x_t is the perturbed sample at time t, leading to a Gaussian diffusion kernel

$$p(x_t|x) = \mathcal{N}(x_t|\alpha_t x, \Sigma_t).$$



Forward diffusion process.

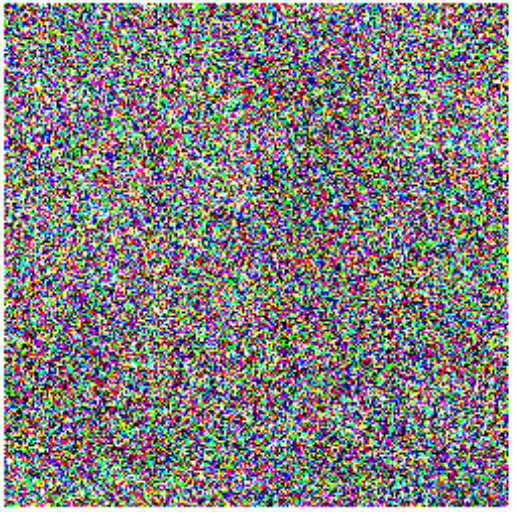
—

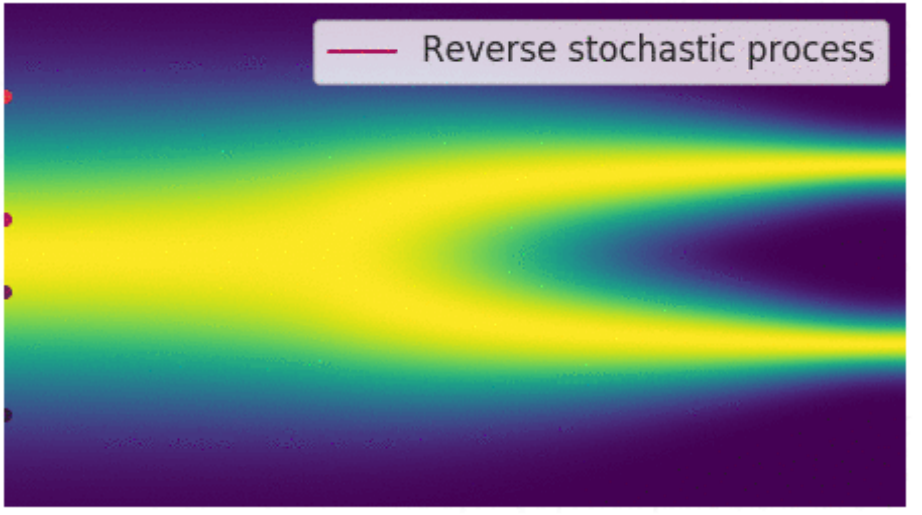
Credits: Song, 2021.

The reverse process satisfies a reverse-time SDE that can be derived analytically from the forward SDE as

$$\mathrm{d}x_t = \left[f_t x_t - \frac{1+\eta^2}{2} g_t^2 \nabla_{x_t} \log p(x_t) \right] \mathrm{d}t + \eta g_t \mathrm{d}w_t.$$

Therefore, to generate data samples $x_0 \sim p(x_0) \approx p(x)$, we can draw noise samples $x_1 \sim p(x_1) \approx \mathcal{N}(0, \Sigma_1)$ and gradually remove the noise therein by simulating the reverse SDE from $t = 1$ to 0.

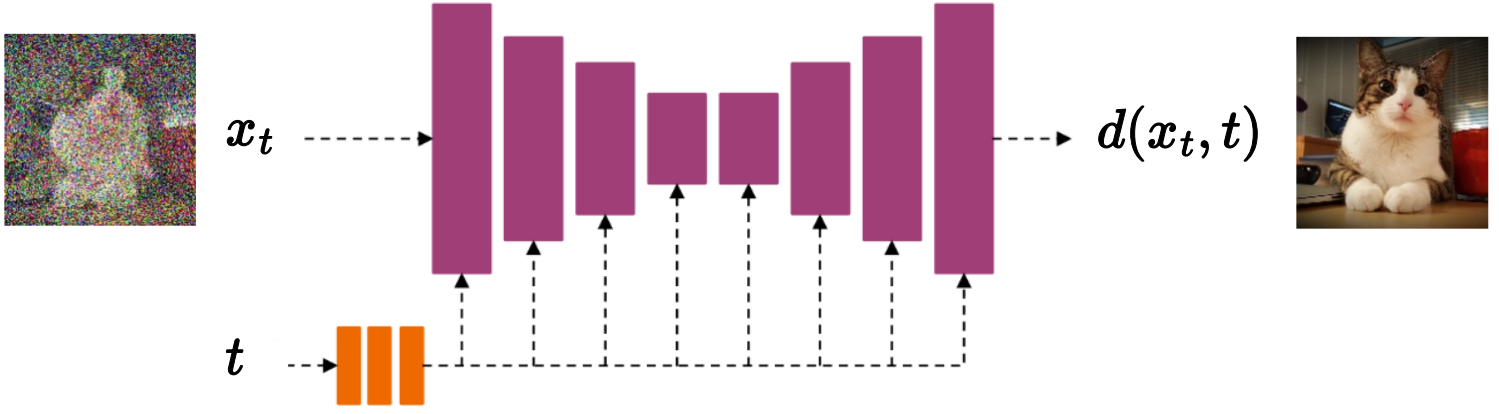




Reverse denoising process.

—

Credits: Song, 2021.



The score function $\nabla_{x_t} \log p(x_t)$ is unknown, but can be approximated by a neural network $d_\theta(x_t, t)$ by minimizing the denoising score matching objective

$$\mathbb{E}_{p(x)p(t)p(x_t|x)} \left[\lambda_t ||d_\theta(x_t, t) - x||_2^2 \right].$$

The optimal denoiser d_θ is the mean $\mathbb{E}[x|x_t]$ which, via Tweedie's formula, allows to use $s_\theta(x_t, t) = \Sigma_t^{-1}(d_\theta(x_t, t) - x_t)$ as a score estimate in the reverse SDE.

Inverting single observations

Because of the Bayes' rule, the posterior score $\nabla_{x_t} \log p(x_t|y)$ to inject in the reverse SDE can be decomposed as

$$\nabla_{x_t} \log p(x_t|y) = \nabla_{x_t} \log p(x_t) + \nabla_{x_t} \log p(y|x_t) - \cancel{\nabla_{x_t} \log p(y)}.$$

This is particularly convenient as it enables **zero-shot posterior sampling** from a diffusion prior $p(x_0)$ without having to pre-wire the neural denoiser to the observation model $p(y|x)$.

Approximating $\nabla_{x_t} \log p(y|x_t)$

We want to estimate the score $\nabla_{x_t} \log p(y|x_t)$ of the noise-perturbed likelihood

$$p(y|x_t) = \int p(y|x)p(x|x_t)\mathrm{d}x.$$

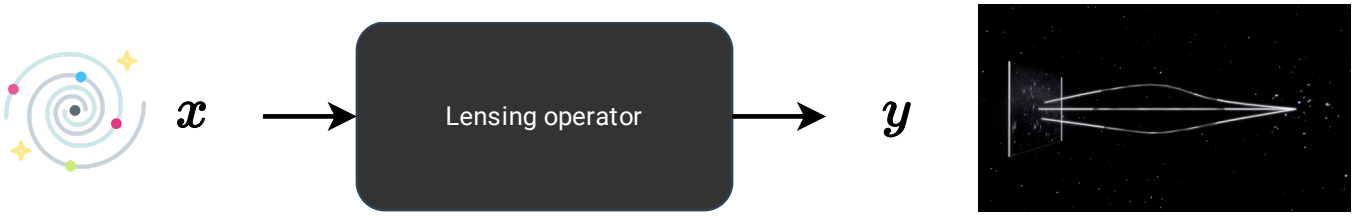
Our approach:

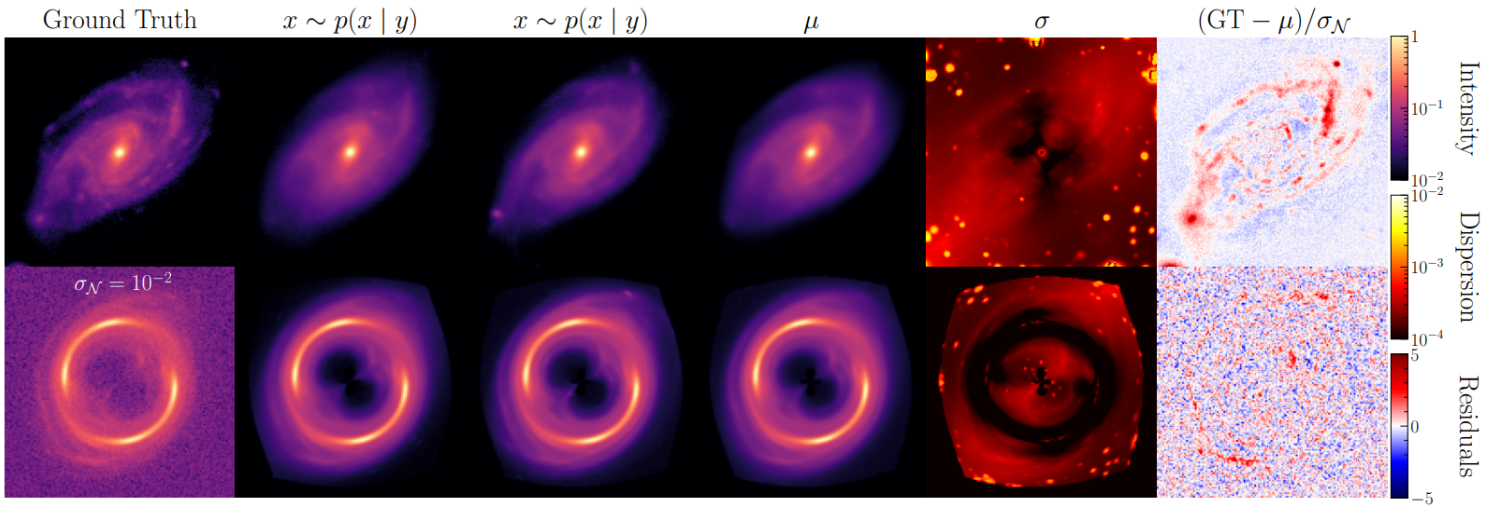
- Assume a linear Gaussian observation model $p(y|x) = \mathcal{N}(y|Ax, \Sigma_y)$.
- Assume the approximation $p(x|x_t) \approx \mathcal{N}(x|\mathbb{E}[x|x_t], \mathbb{V}[x|x_t])$, where $\mathbb{E}[x|x_t]$ is estimated by the denoiser and $\mathbb{V}[x|x_t]$ is estimated using Tweedie's covariance formula.
- Then $p(y|x_t) \approx \mathcal{N}(y|A\mathbb{E}[x|x_t], \Sigma_y + A\mathbb{V}[x|x_t]A^T)$.
- The score $\nabla_{x_t} \log p(y|x_t)$ then approximates to

$$\nabla_{x_t}\mathbb{E}[x|x_t]^T A^T(\Sigma_y + A\mathbb{V}[x|x_t]A^T)^{-1}(y - A\mathbb{E}[x|x_t]).$$

—

See also Daras et al (2024) for a survey on diffusion models for inverse problems.



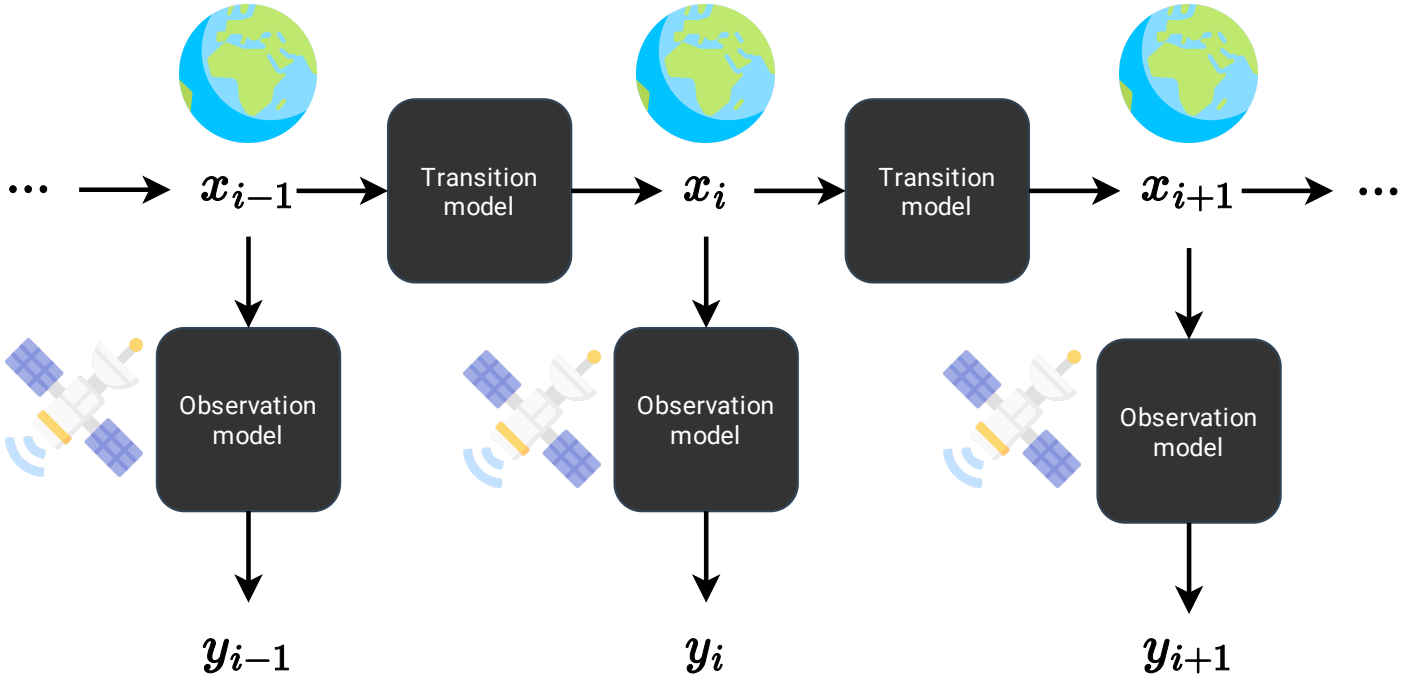


Plausible galaxy images $\boldsymbol{x}$ can be recovered from lensed observations $\boldsymbol{y}$ by zero-shot posterior sampling from a diffusion prior $\boldsymbol{p(x)}$.

—

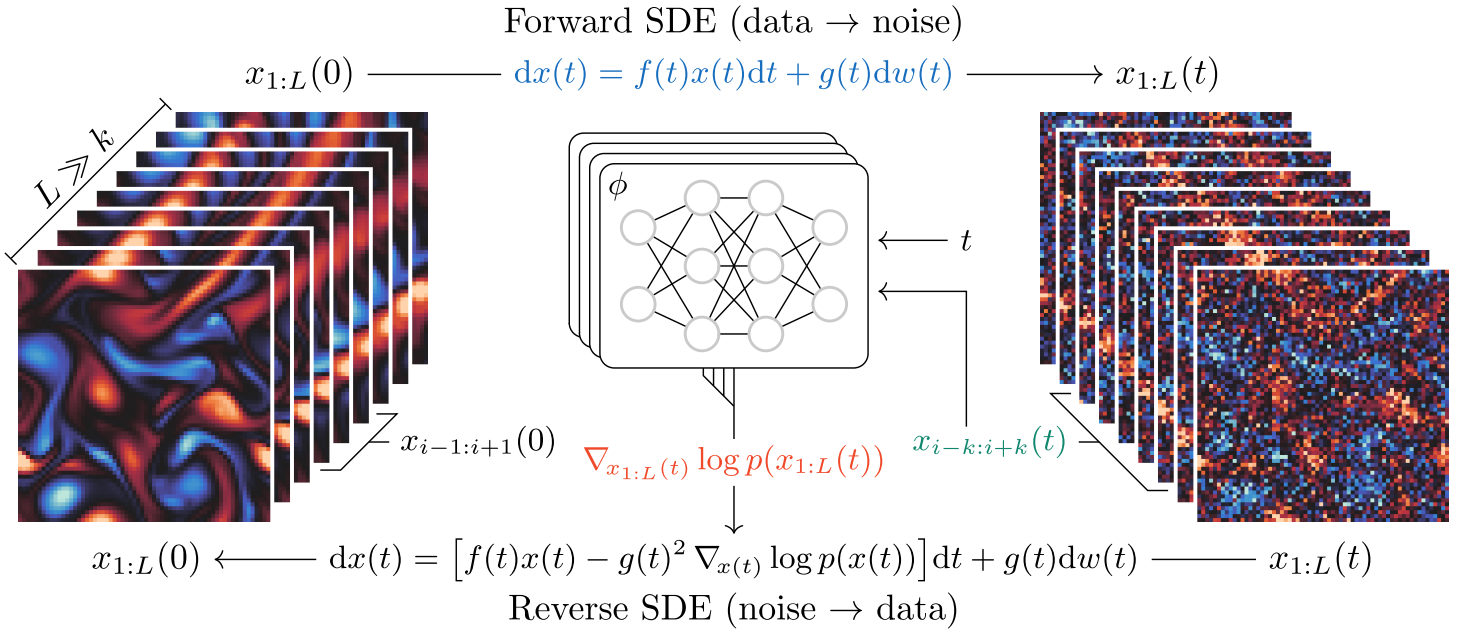
Credits: Adam et al., 2022 (arXiv:2211.03812).

Score-based data assimilation in dynamical systems



The goal of **data assimilation** is to estimate plausible trajectories $x_{1:L}$ given one or more noisy observations y (or $y_{1:L}$) as the posterior

$$p(x_{1:L}|y) = \frac{p(y|x_{1:L})}{p(y)} p(x_0) \prod_{i=1}^{L-1} p(x_{i+1}|x_i).$$

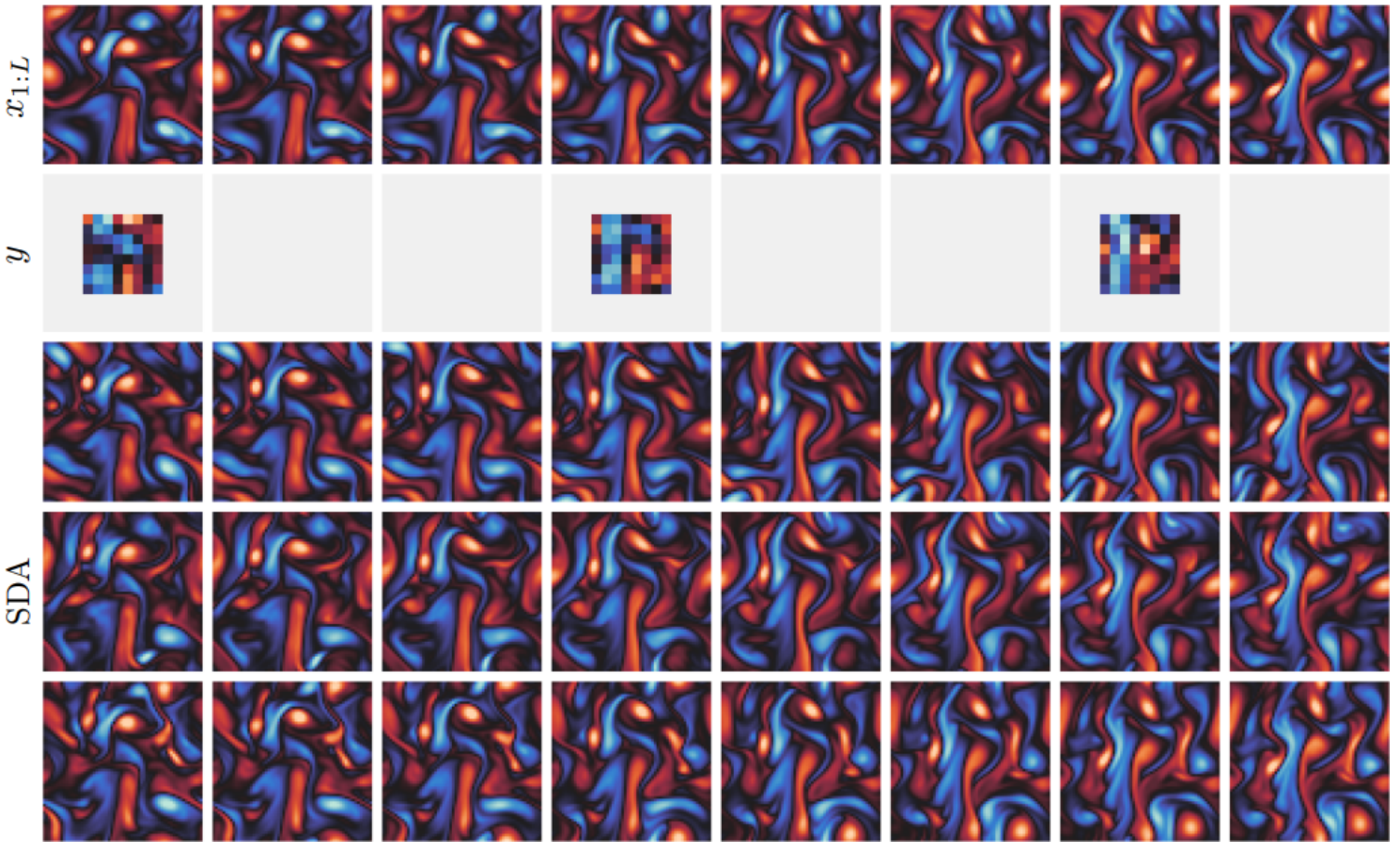


Our approach:

- Build a score-based generative model $p(x_{1:L})$ of arbitrary-length trajectories*.
- Use zero-shot posterior sampling to generate plausible trajectories from noisy observations y.

—

*:The score of a (noise perturbed) trajectory can be approximated by a sum of scores. See paper for details.



Sampling trajectories $x_{1:L}$ from
noisy, incomplete and coarse-grained observations y.

Summary

Simulation-based inference is a major evolution in the statistical capabilities for science, as it enables the analysis of complex models and data without simplifying assumptions.

Obstacles remain to be overcome, such as the curse of dimensionality, the need for large amounts of data, or the necessary robustness of the inference network.