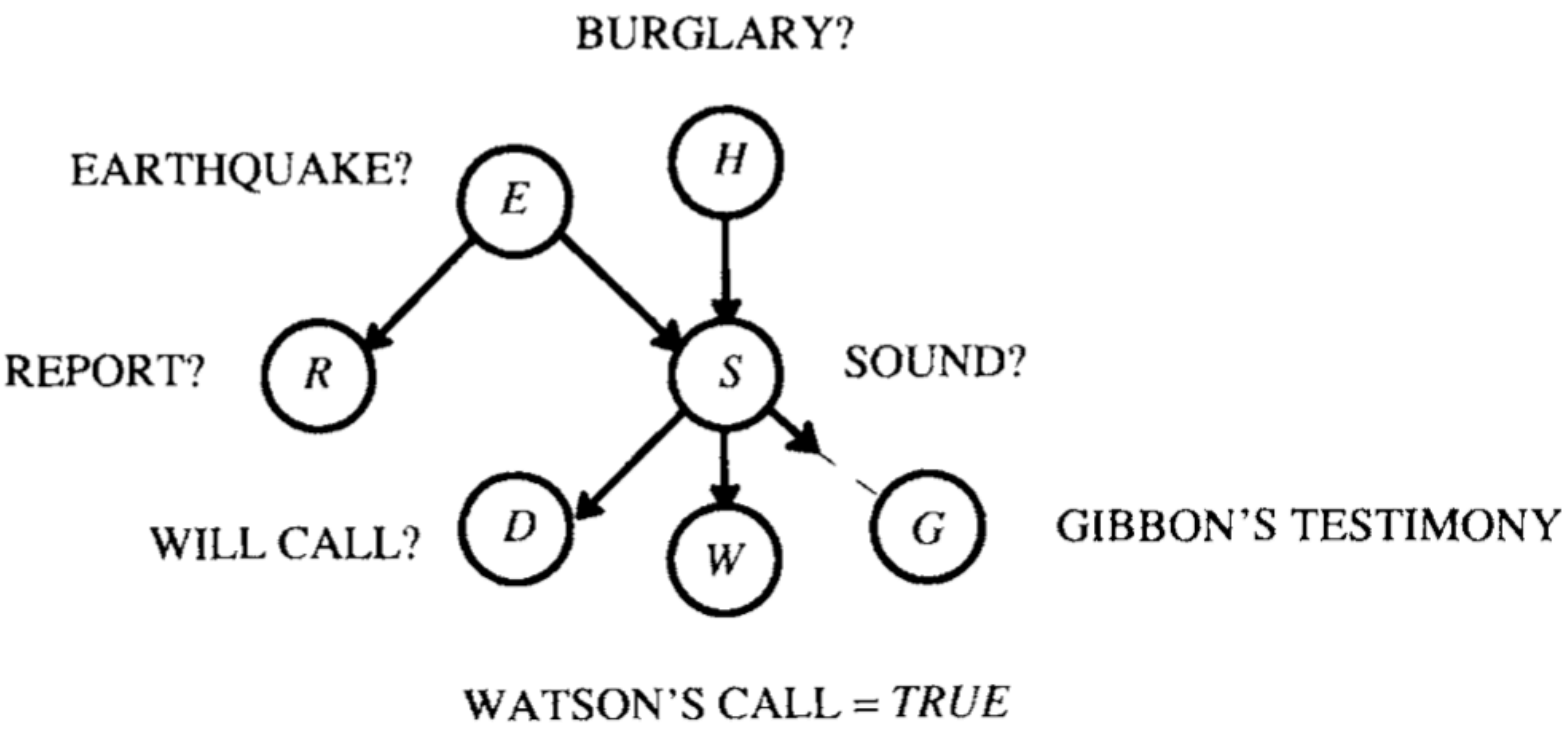
Graphical Models and Simulation-Based Inference

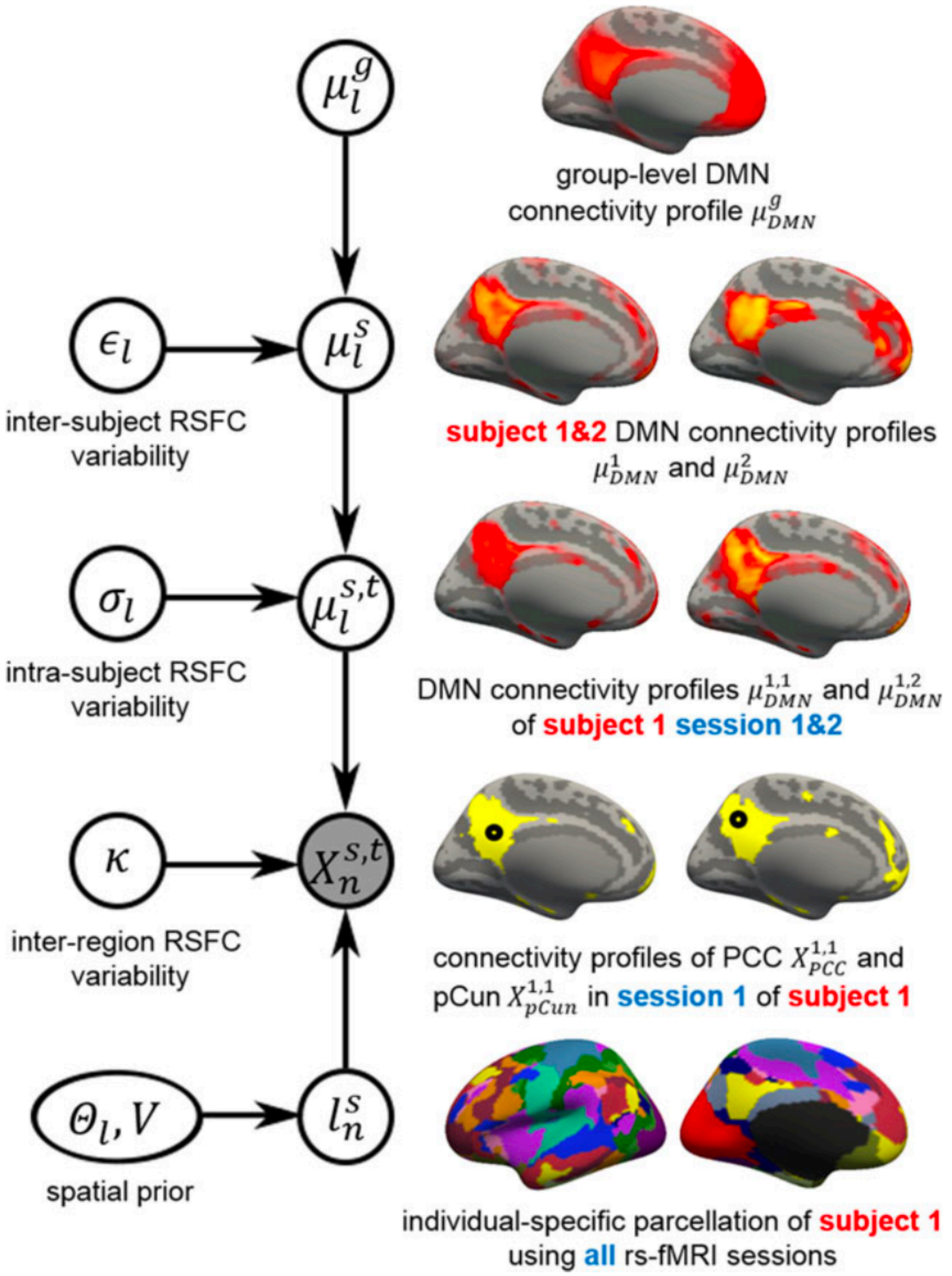
Graphical Models: Discrete Inference and Learning

Demian Wassermann 2025

Introduction to DAG and their relationship with Probability Functions (Pearl)

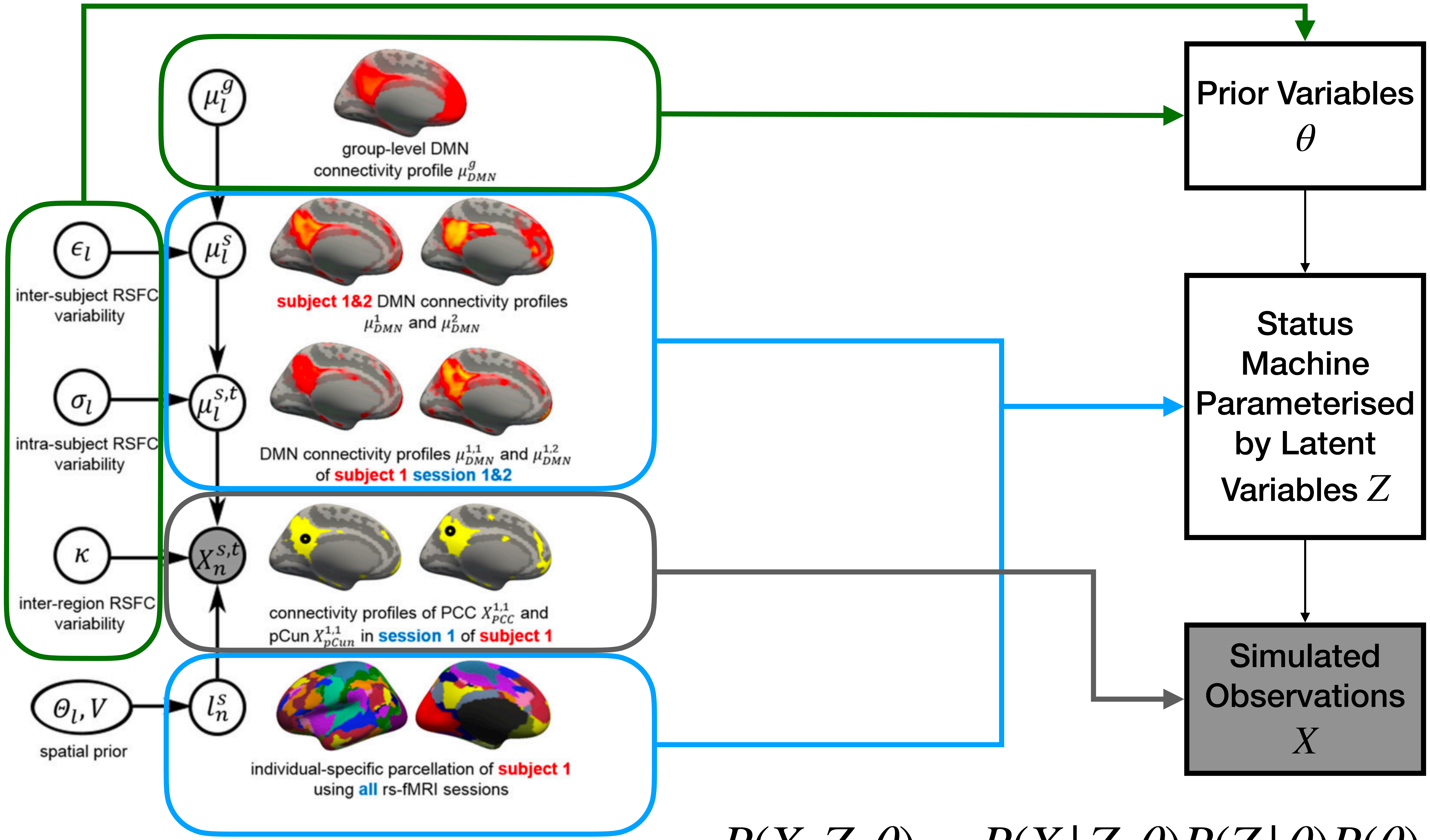


[Pearl 1987]



[Kong et al 2019]

Graphical Models and Simulation Systems



$\theta \sim P(\theta)$

$Z \sim P(Z|\theta)$

$X \sim P(X|Z, \theta)$

$$P(X, Z, \theta) = P(X|Z, \theta)P(Z|\theta)P(\theta) = P(X|Z)P(Z|\theta)P(Z)$$

General Inference Notation

θ: parameters   X: observations

Likelihood   Prior

$$P(\theta|X) = \frac{P(X|\theta)P(\theta)}{P(X)}$$

Posterior

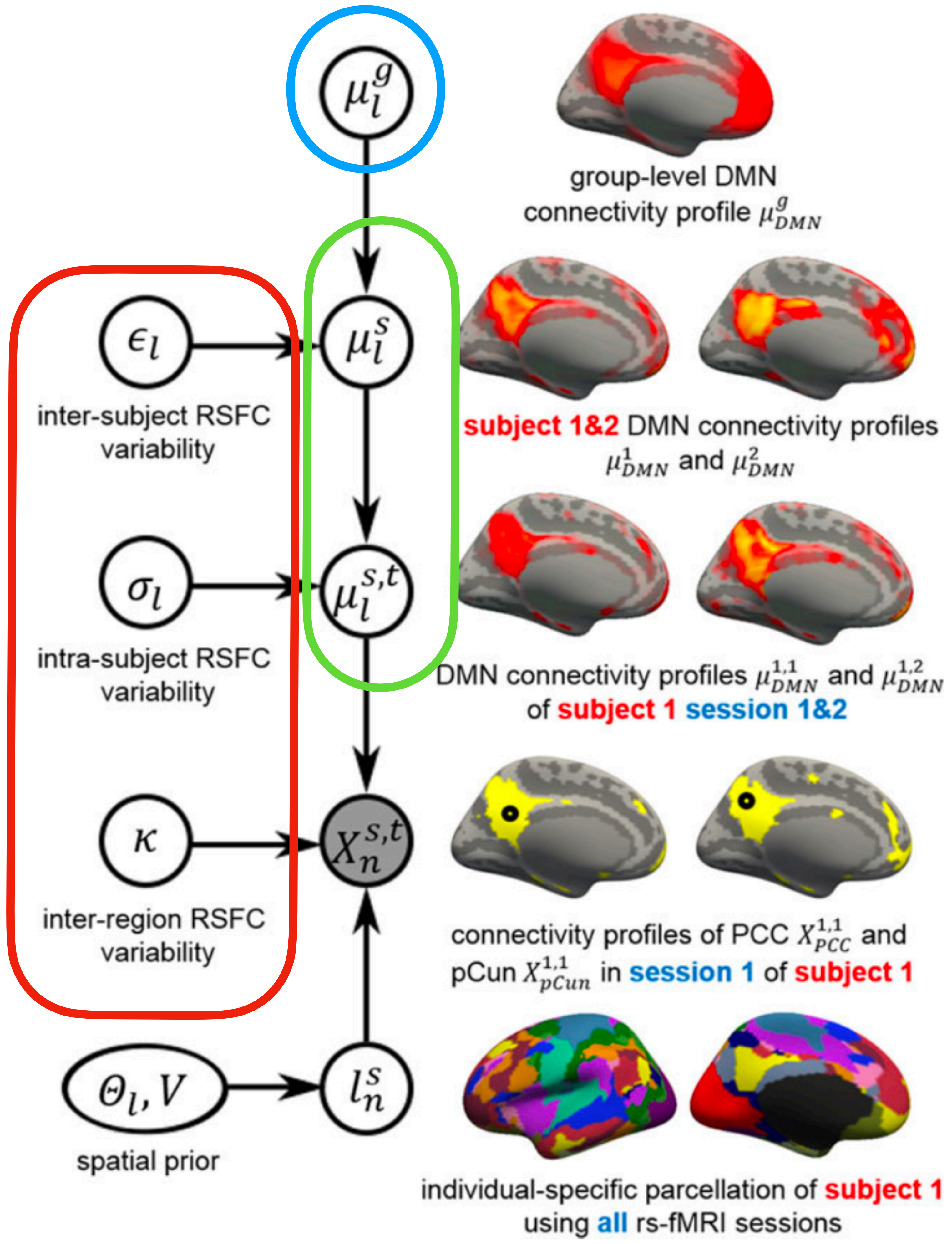
Evidence

Z: latent random variables

$$P(\theta|X) = \frac{\mathbb{E}_Z[P(X|Z,\theta)]P(\theta)}{P(X)}$$

η: nuisance random variables

$$P(\theta|X) = \frac{\mathbb{E}_\eta[P(X|\theta,\eta)]P(\theta)}{P(X)}$$



General Inference Notation

θ: parameters   X: observations

Likelihood   Prior

$$P(\theta \mid X) = \frac{P(X \mid \theta)P(\theta)}{P(X)}$$

Posterior

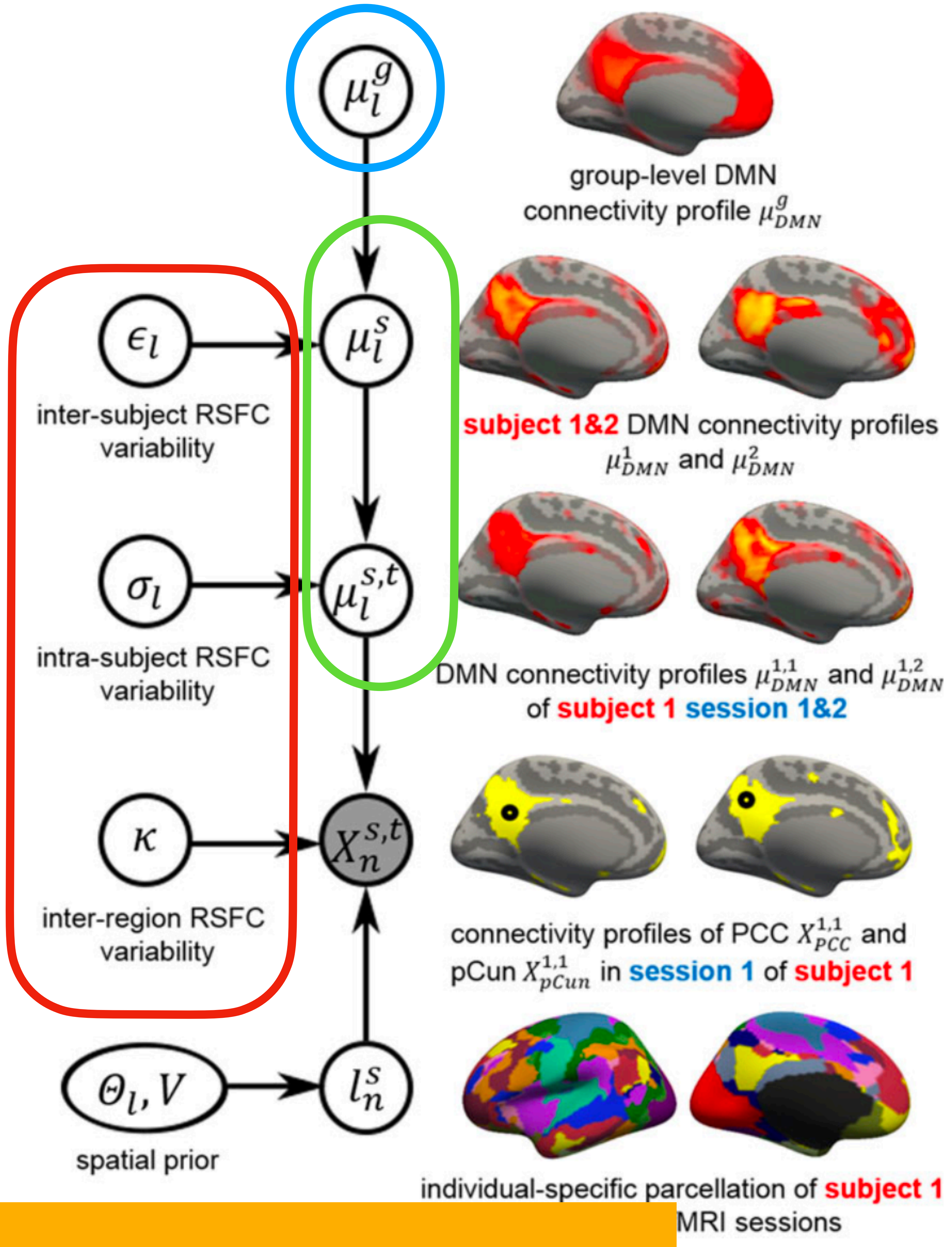
Evidence

Z: latent random variables

$$P(\theta \mid X) = \frac{\mathbb{E}_Z[P(X \mid Z, \theta)]P(\theta)}{P(X)}$$

η: nuisance random variables

$$P(\theta \mid X) = \frac{\mathbb{E}_\eta[P(X \mid \theta, \eta)]P(\theta)}{P(X)}$$



Intractable in general:
full likelihood impossible to evaluated or computation cost is extremely high

Likelihood computation is hard:
Enter Mechanistic, Example Models Galton Board

θ: parameters   X: observations

Likelihood   Prior

$$\underbrace{P(\theta|X)}_{\text{Posterior}} = \frac{P(X|\theta)P(\theta)}{\underbrace{P(X)}_{\text{Evidence}}}$$

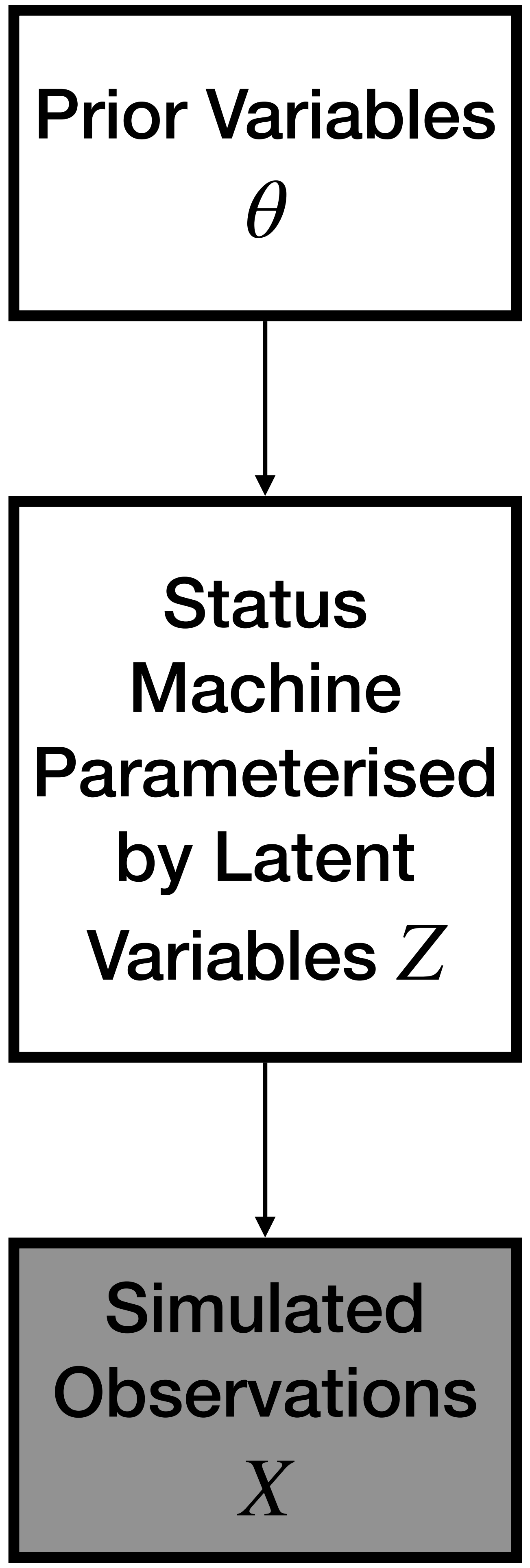
Z: latent random variables

$$P(\theta|X) = \frac{\mathbb{E}_Z[P(X|Z,\theta)]P(\theta)}{P(X)}$$

η: nuisance random variables

$$P(\theta|X) = \frac{\mathbb{E}_\eta[P(X|\theta,\eta)]P(\theta)}{P(X)}$$

Simulation-Based Inference



$\theta \sim P(\theta)$

$Z \sim P(Z|\theta)$

$X \sim P(X|Z,\theta)$

- Inference is defined as finding the θ that could be at the origin of an observation X. Specifically computing $P(\theta|X) = \mathbb{E}_Z[P(\theta, Z|X)]$
- For this, we use Bayes $P(\theta, Z|X) = \dfrac{P(X|Z,\theta)P(Z,\theta)}{P(X)}$, nonetheless the likelihood $P(X|Z,\theta)$ is often unknown or intractable.
- Hence simulation-based inference either approximates or eliminates the need for an explicit likelihood by simulating observations.

Simulation-Based Inference: Neural Network Approximations

Prior Variables θ $\quad \theta \sim P(\theta)$

↓

Status Machine Parameterised by Latent Variables Z $\quad Z \sim P(Z|\theta)$

↓

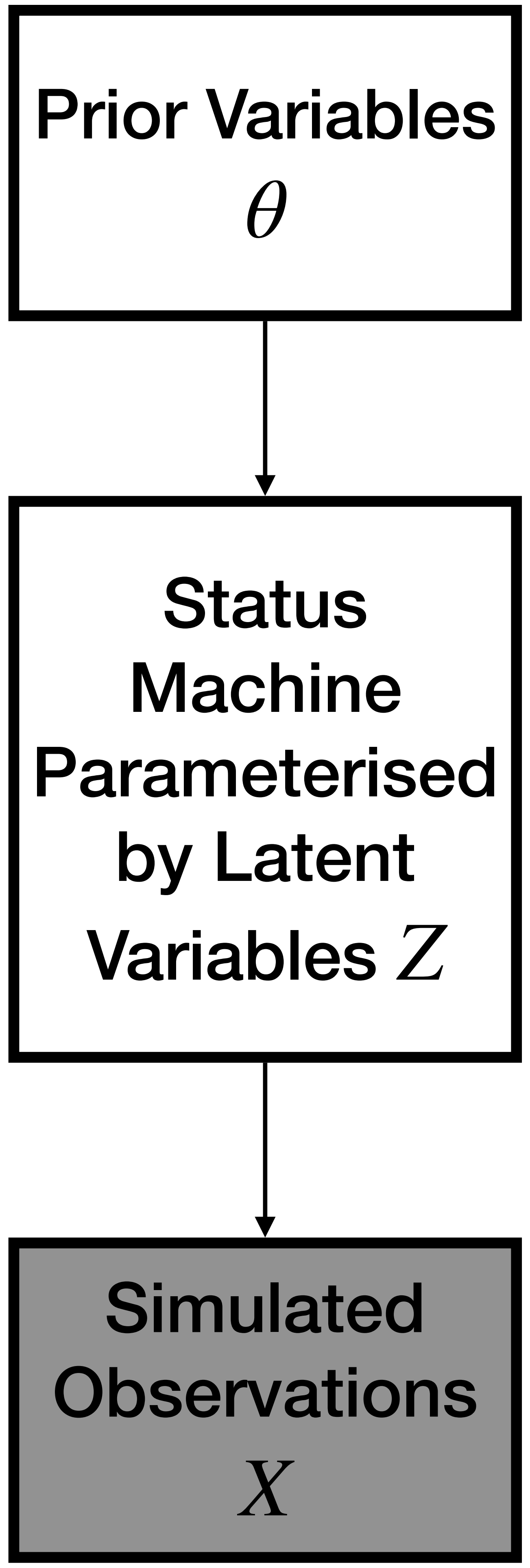
Simulated Observations X $\quad X \sim P(X|Z,\theta)$

θ: parameters X: observations

$$\underbrace{P(\theta|X)}_{\text{Posterior}} = \frac{\overbrace{P(X|\theta)}^{\text{Likelihood}}\overbrace{P(\theta)}^{\text{Prior}}}{\underbrace{P(X)}_{\text{Evidence}}}$$

- $P(\theta|X)$ approximated through "Neural Posterior" estimators
- $P(X|\theta)$ approximated through "Neural Likelihood" estimators
- $\dfrac{P(X|\theta)}{P(X)}$ approximated through the "Neural ratio" estimators

Simulation-Based Inference: Why now it works (Cranmer et al 2019)



$$\theta \sim P(\theta)$$

$$Z \sim P(Z \mid \theta)$$

$$X \sim P(X \mid Z, \theta)$$

θ: parameters X: observations

Likelihood Prior

$$P(\theta \mid X) = \frac{P(X \mid \theta) P(\theta)}{P(X)}$$

Posterior

Evidence

- Novel ML-based approaches allow us to massively generate simulated observations
- Autodifferentiation and neural network approaches are great non-linear function estimators
- Active learning can help improving sampling efficiency much better than Markov Chains

Simulation-Based Inference: Approximate Bayesian MC

Prior Variables θ

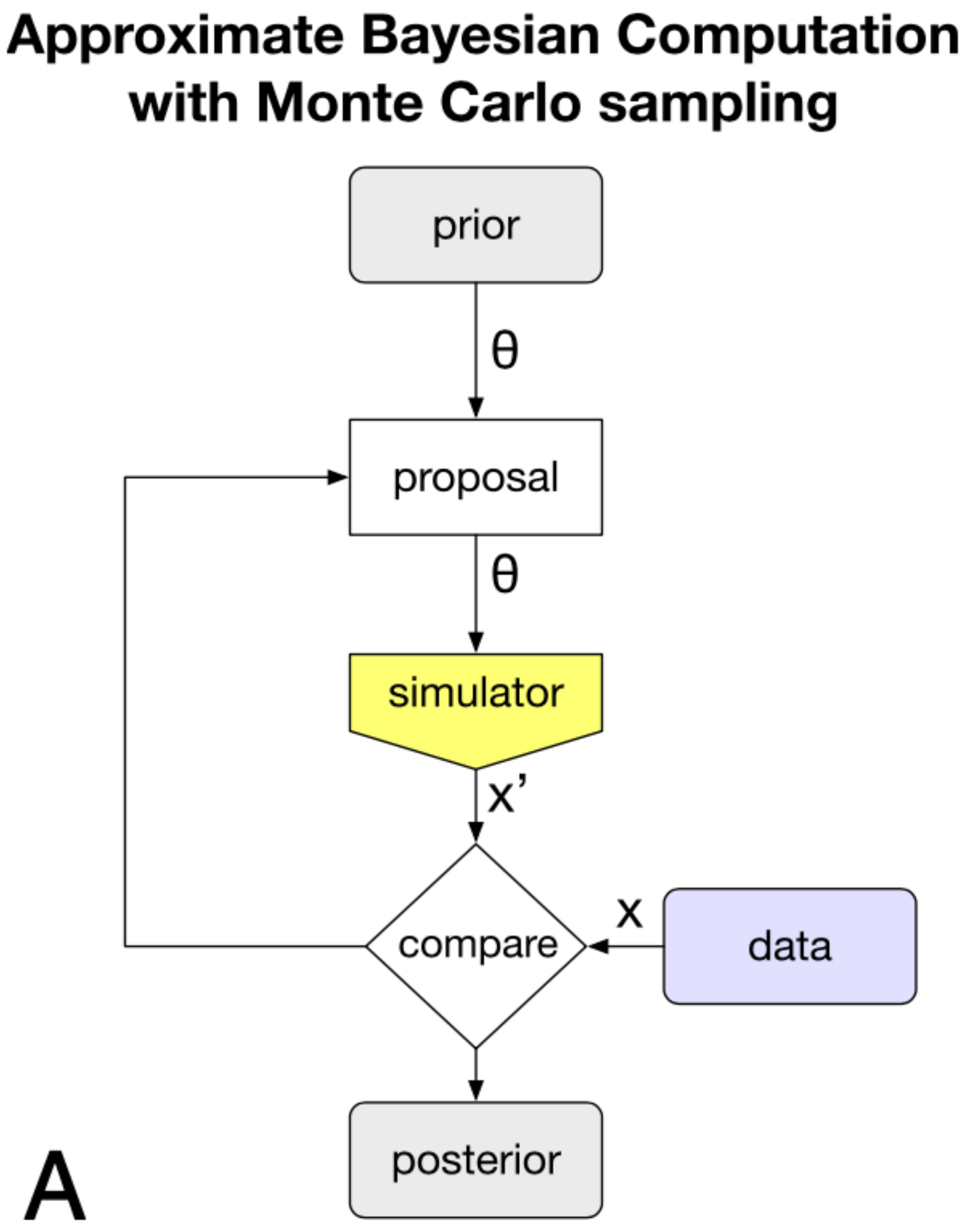
$$\theta \sim P(\theta)$$

Status Machine Parameterised by Latent Variables Z

$$Z \sim P(Z|\theta)$$

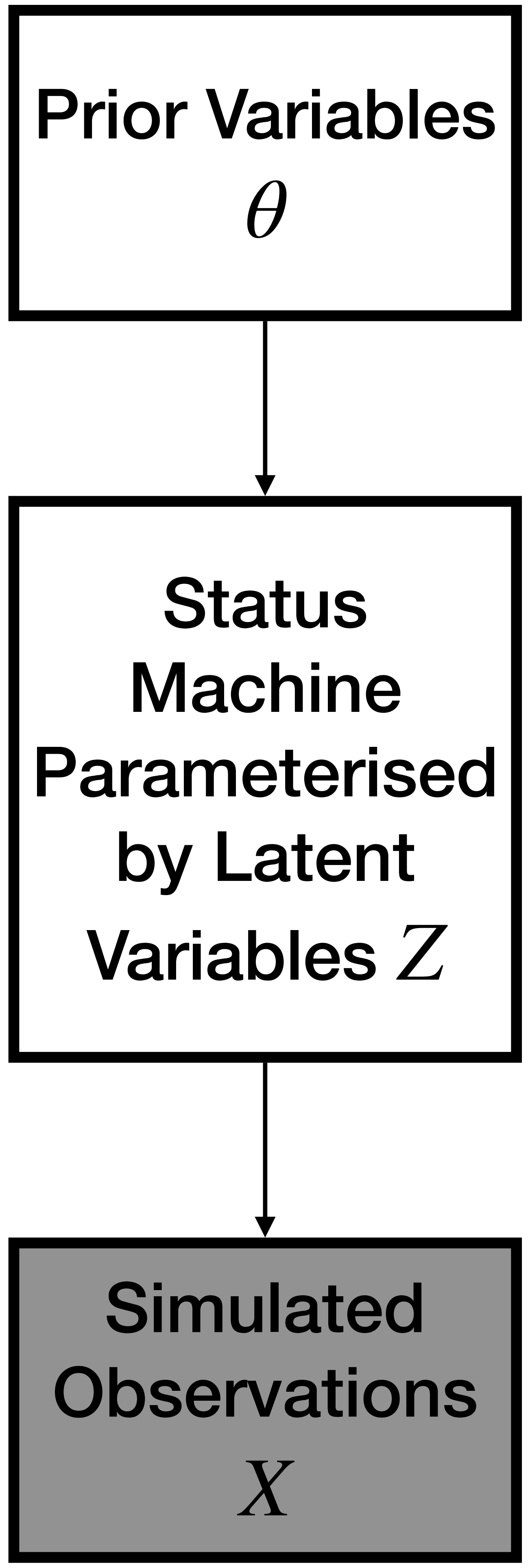
Simulated Observations X

$$X \sim P(X|Z,\theta)$$



(Cranmer et al 2019)

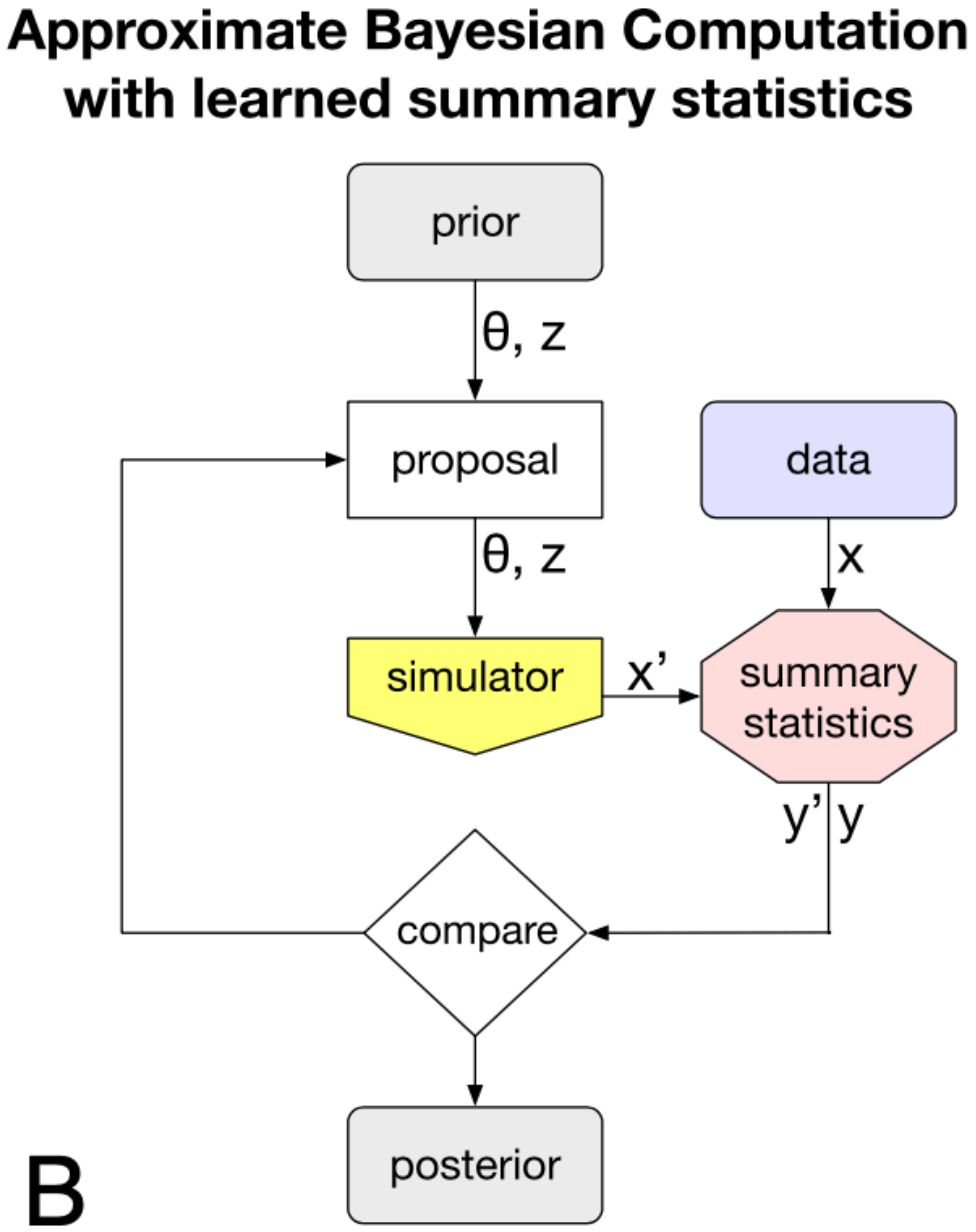
Simulation-Based Inference: Approximate Bayesian MC



$$\theta \sim P(\theta)$$

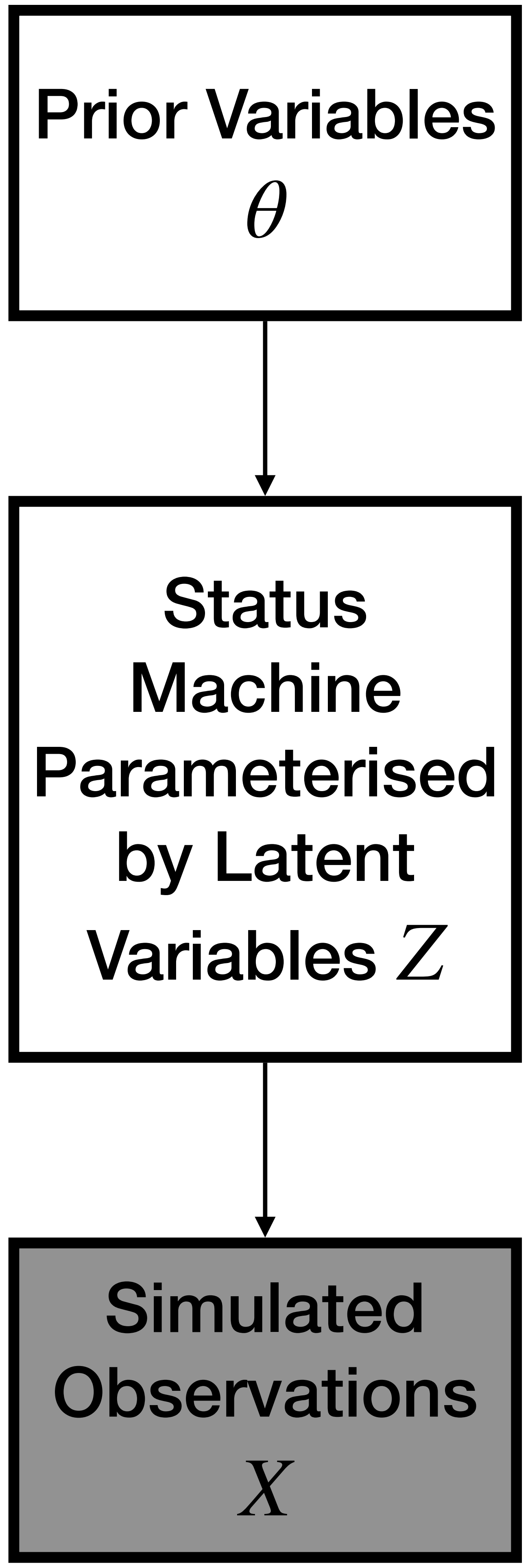
$$Z \sim P(Z|\theta)$$

$$X \sim P(X|Z,\theta)$$



(Cranmer et al 2019)

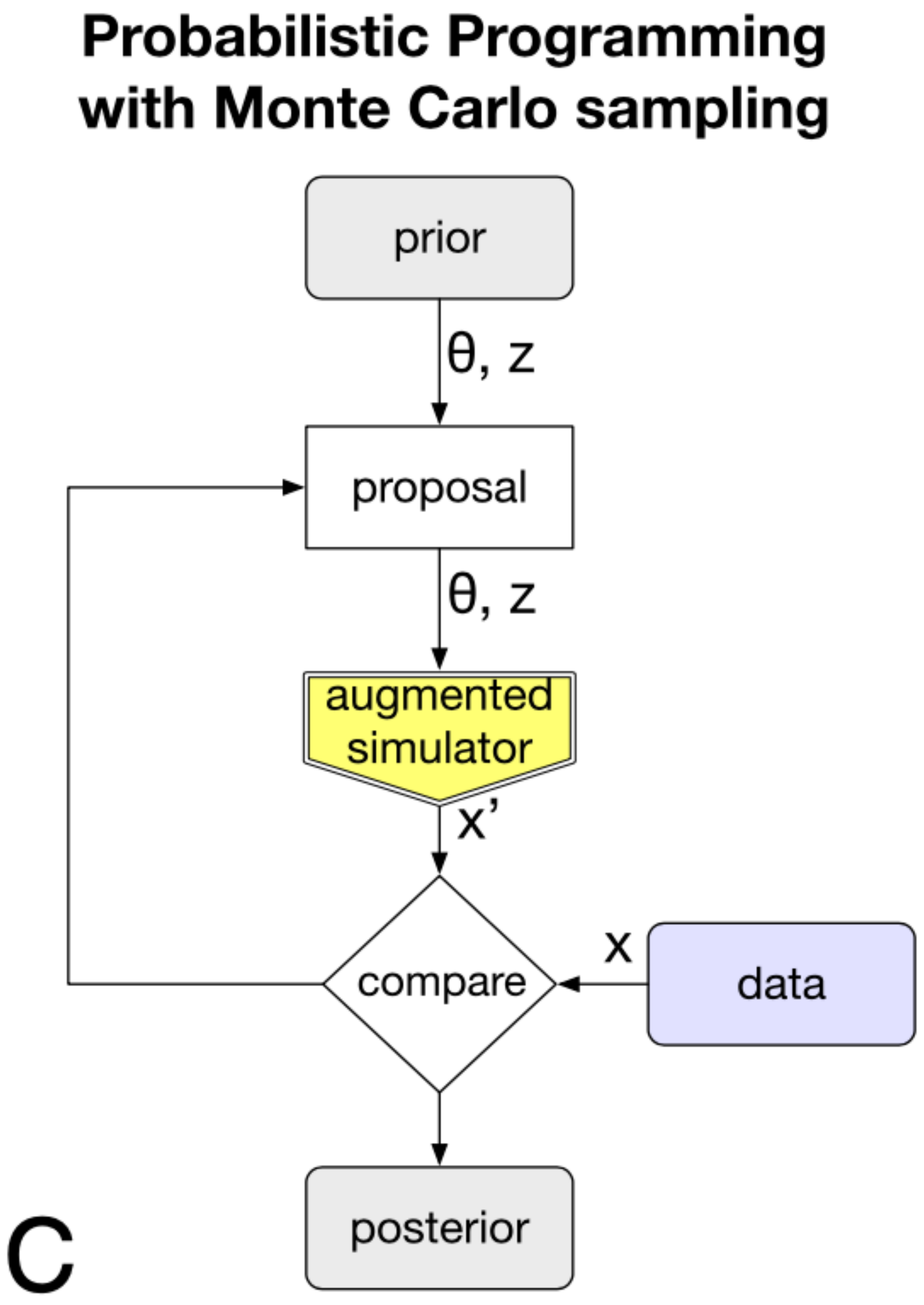
Simulation-Based Inference: Approximate Bayesian MC



$$\theta \sim P(\theta)$$

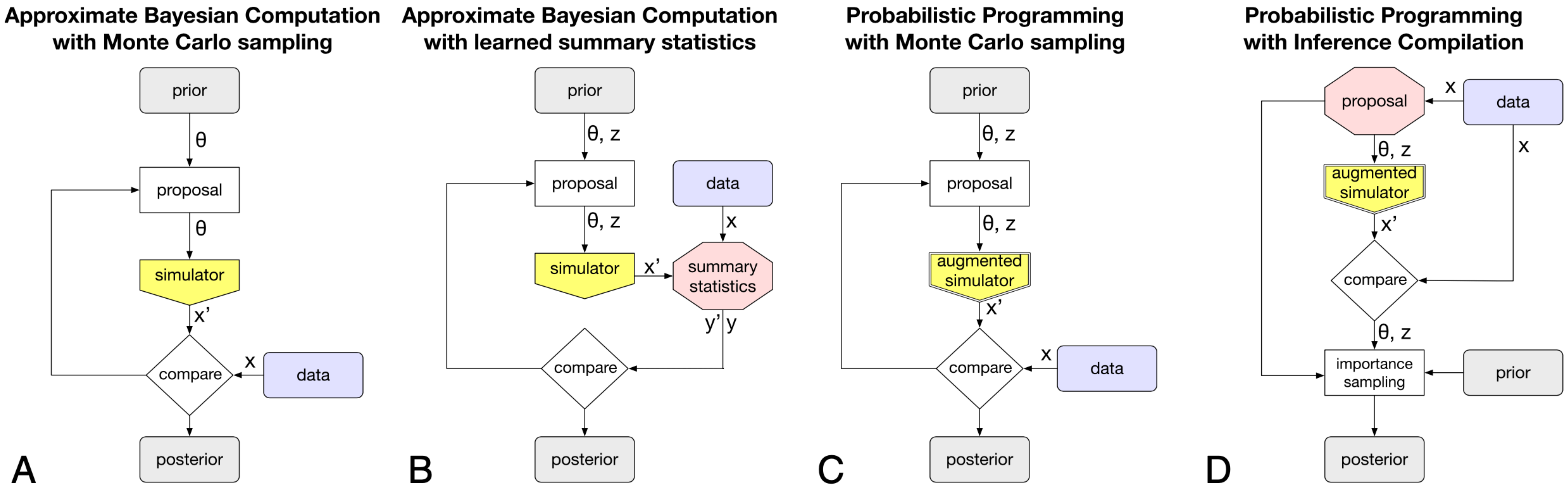
$$Z \sim P(Z|\theta)$$

$$X \sim P(X|Z,\theta)$$



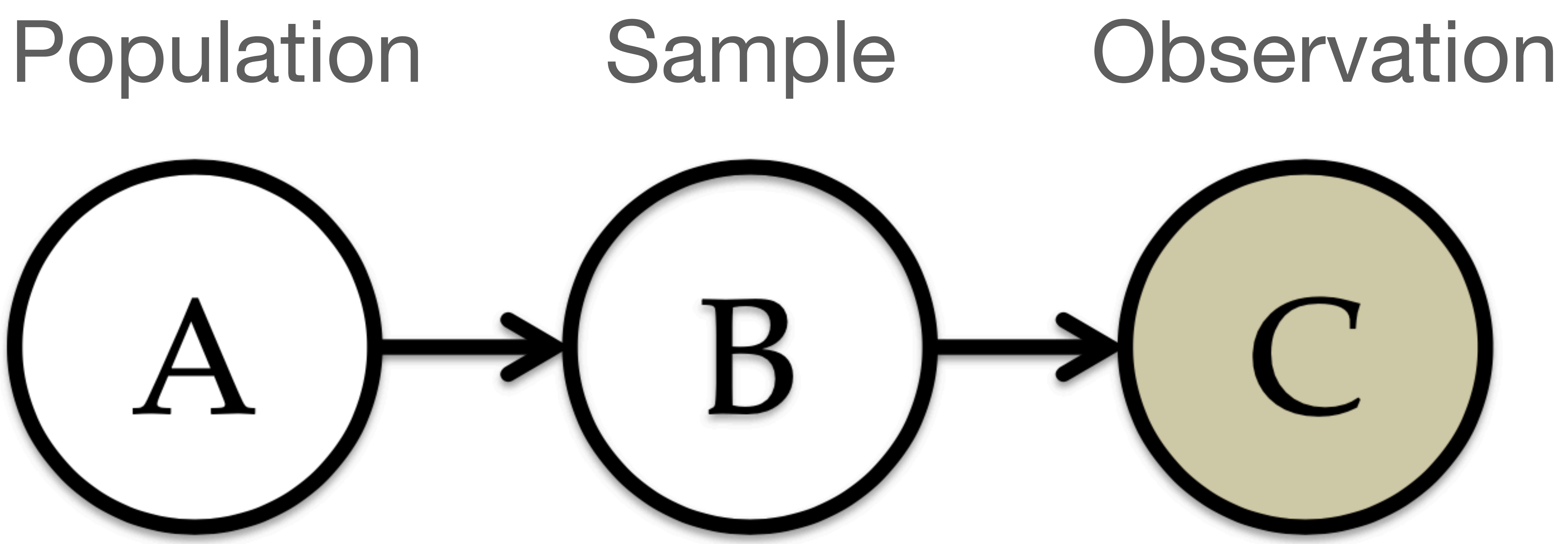
(Cranmer et al 2019)

Simulation-Based Inference: Approximate Bayesian MC



(Cranmer et al 2019)

Simulation-Based Inference: Amortization

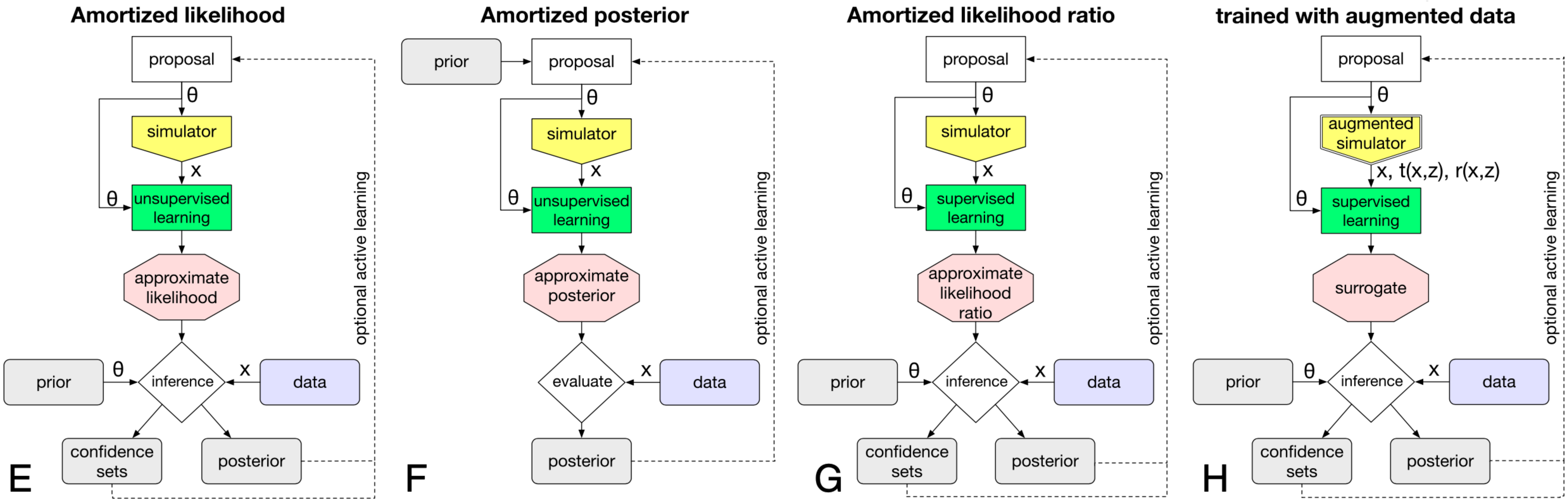


Query 1: $P(B|C) = P(C|B)P(B)/P(C)$

Query 2: $P(A|C) = \sum_{B} P(A|B)P(B|C)$

(Gershman et al 2014

Simulation-Based Inference: Amortisation Techniques



(Cranmer et al 2019)

Simulation-Based Inference: Neural Network Approximations

Prior Variables θ — $\theta \sim P(\theta)$

↓

Status Machine Parameterised by Latent Variables Z — $Z \sim P(Z|\theta)$

↓

Simulated Observations X — $X \sim P(X|Z,\theta)$

θ: parameters X: observations

Likelihood Prior

$$P(\theta|X) = \frac{P(X|\theta)P(\theta)}{P(X)}$$

Posterior

Evidence

- $P(\theta|X)$ approximated through "Neural Posterior" estimators
- $P(X|\theta)$ approximated through "Neural Likelihood" estimators
- $\dfrac{P(X|\theta)}{P(X)}$ approximated through the "Neural ratio" estimators

Simulation-Based Inference:
Neural Network Approximations Through Stochastic Flows

θ: parameters X: observations

Likelihood Prior

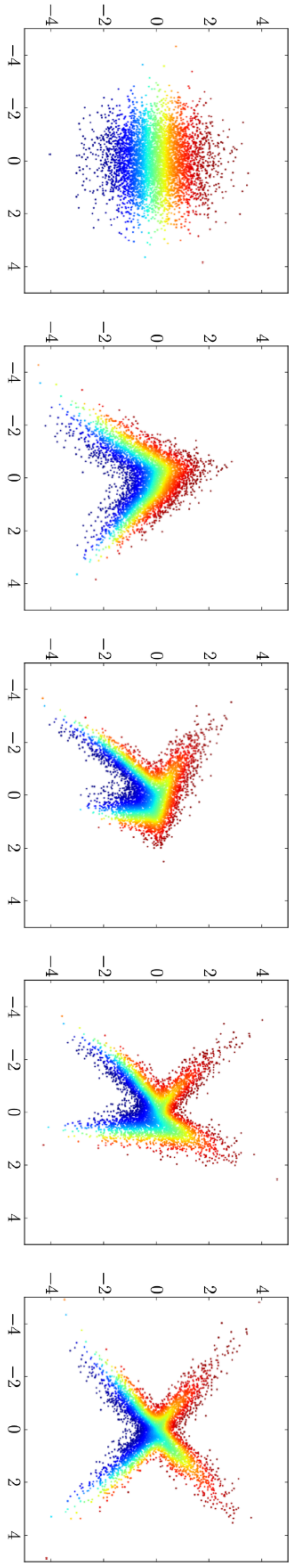
$$P(\theta\,|\,X) = \frac{P(X\,|\,\theta)P(\theta)}{P(X)}$$

Posterior

Evidence

$$f(X,\theta) = N_{\mu,\Sigma}(\phi(X,\theta))\,|\,J_{\phi}(X,\theta)\,|$$

f the Neural estimator and ϕ the stochastic flow



- $P(\theta\,|\,X)$ approximated through “Neural Posterior” estimators
- $P(X\,|\,\theta)$ approximated through “Neural Likelihood” estimators
- $\dfrac{P(X\,|\,\theta)}{P(X)}$ approximated through the “Neural ratio” estimators

Simulation-Based Inference: Automatic Posterior Transformation (Greenberg et al 2019)

θ: parameters X: observations

Likelihood Prior

$$P(\theta|X) = \frac{P(X|\theta)P(\theta)}{P(X)}$$

Posterior

Evidence

- $P(\theta|X)$ approximated through "Neural posterior" by a flow $Q_{F(x_0,\phi)}(\theta)$

- Loss function:

$$\tilde{q}_{x,\phi}(\theta) = q_{F(x,\phi)}(\theta)\frac{\tilde{p}(\theta)}{p(\theta)}\frac{1}{Z(x,\phi)}, \qquad (2)$$

- Where a proposal posterior is

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

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_ψ, 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 $q_{F(x_o,\phi)}(\theta)$

Simulation-Based Inference: Sequential Neural Likelihood (Papamakarios et al 2019)

θ: parameters   X: observations

$$\underbrace{P(\theta \mid X)}_{\text{Posterior}} = \frac{\overbrace{P(X \mid \theta)}^{\text{Likelihood}}\overbrace{P(\theta)}^{\text{Prior}}}{\underbrace{P(X)}_{\text{Evidence}}}$$

- $P(X \mid \theta)$ approximated through "Neural likelihood" by a flow $Q_\phi(X \mid \theta)$

Algorithm 1: Sequential Neural Likelihood (SNL)

Input : observed data $\mathbf{x}_o$, estimator $q_{\boldsymbol{\phi}}(\mathbf{x} \mid \boldsymbol{\theta})$, number of rounds R, simulations per round N

Output : approximate posterior $\hat{p}(\boldsymbol{\theta} \mid \mathbf{x}_o)$

set $\hat{p}_0(\boldsymbol{\theta} \mid \mathbf{x}_o) = p(\boldsymbol{\theta})$ and $\mathcal{D} = \{\}$
for $r = 1 : R$ **do**
  for $n = 1 : N$ **do**
    sample $\boldsymbol{\theta}_n \sim \hat{p}_{r-1}(\boldsymbol{\theta} \mid \mathbf{x}_o)$ with MCMC
    simulate $\mathbf{x}_n \sim p(\mathbf{x} \mid \boldsymbol{\theta}_n)$
    add $(\boldsymbol{\theta}_n, \mathbf{x}_n)$ into $\mathcal{D}$
  (re-)train $q_{\boldsymbol{\phi}}(\mathbf{x} \mid \boldsymbol{\theta})$ on $\mathcal{D}$ and set $\hat{p}_r(\boldsymbol{\theta} \mid \mathbf{x}_o) \propto q_{\boldsymbol{\phi}}(\mathbf{x}_o \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta})$
return $\hat{p}_R(\boldsymbol{\theta} \mid \mathbf{x}_o)$

Simulation-Based Inference: Neural Ratio (Hermans et al 2020)

θ: parameters   X: observations

Likelihood   Prior

$$P(\theta \mid X) = \frac{P(X \mid \theta)P(\theta)}{P(X)}$$

Posterior

Evidence

- $P(X \mid \theta)/P(X)$ approximated through "Neural ratio" by a flow $d_\phi(X \mid \theta)$

Algorithm 1 Optimization of $\mathbf{d}_\phi(\mathbf{x}, \boldsymbol{\theta})$.

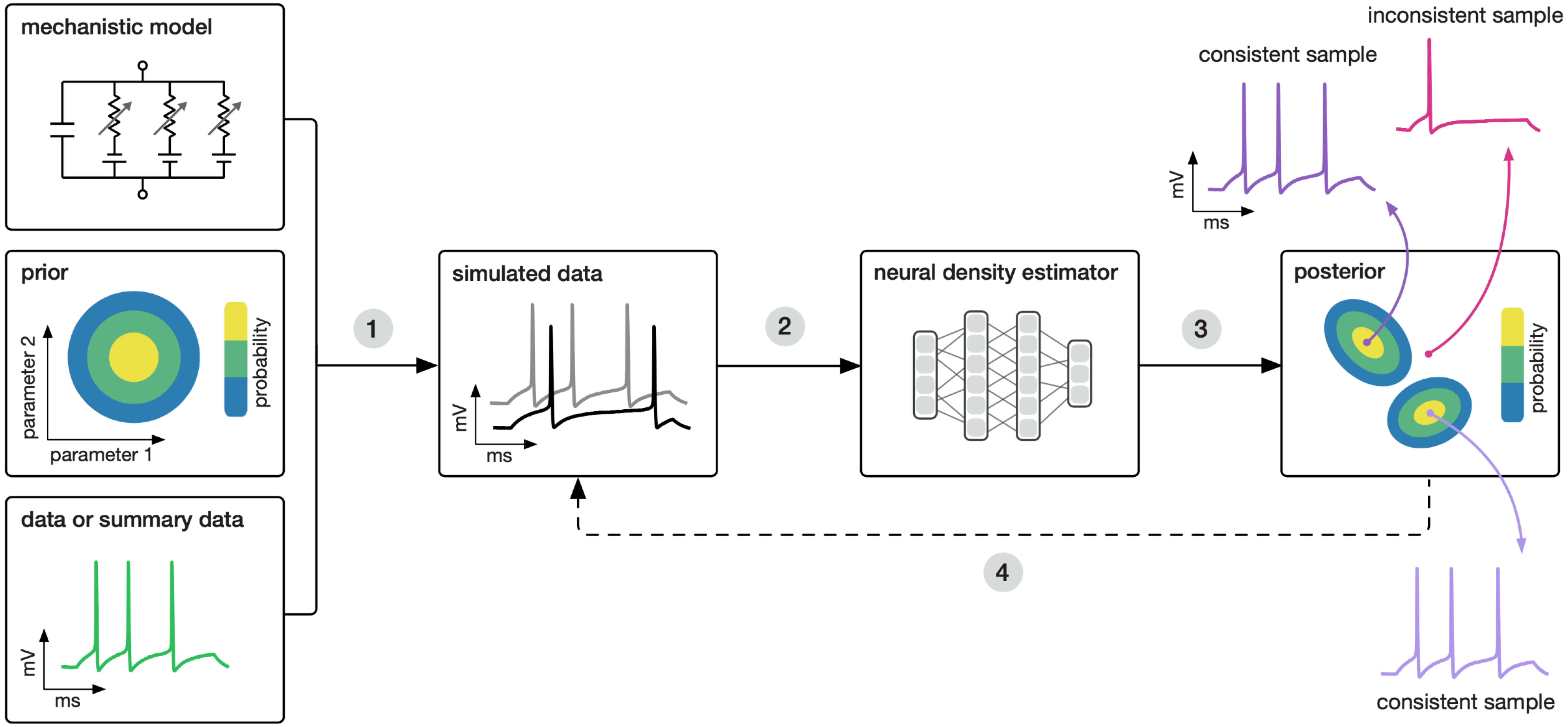
Inputs:	Criterion ℓ (e.g., BCE)
	Implicit generative model $p(\mathbf{x} \mid \boldsymbol{\theta})$
	Prior $p(\boldsymbol{\theta})$
Outputs:	Parameterized classifier $\mathbf{d}_\phi(\mathbf{x}, \boldsymbol{\theta})$
Hyperparameters:	Batch-size M

1: **while not converged do**
2: **Sample** $\boldsymbol{\theta} \leftarrow \{\boldsymbol{\theta}_m \sim p(\boldsymbol{\theta})\}_{m=1}^M$
3: **Sample** $\boldsymbol{\theta}' \leftarrow \{\boldsymbol{\theta}'_m \sim p(\boldsymbol{\theta})\}_{m=1}^M$
4: **Simulate** $\mathbf{x} \leftarrow \{\mathbf{x}_m \sim p(\mathbf{x} \mid \boldsymbol{\theta}_m)\}_{m=1}^M$
5: $\mathcal{L} \leftarrow \ell(\mathbf{d}_\phi(\mathbf{x}, \boldsymbol{\theta}), 1) + \ell(\mathbf{d}_\phi(\mathbf{x}, \boldsymbol{\theta}'), 0)$
6: $\phi \leftarrow \text{OPTIMIZER}(\phi, \nabla_\phi \mathcal{L})$
7: **end while**
8: **return** $\mathbf{d}_\phi$

eLife RESEARCH ARTICLE

Training deep neural density estimators to identify mechanistic models of neural dynamics

Pedro J Gonçalves[1,2†*], Jan-Matthis Lueckmann[1,2†*], Michael Deistler[1,3†*], Marcel Nonnenmacher[1,2,4], Kaan Öcal[2,5], Giacomo Bassetto[1,2], Chaitanya Chintaluri[6,7], William F Podlaski[6], Sara A Haddad[8], Tim P Vogels[6,7], David S Greenberg[1,4], Jakob H Macke[1,2,3,9*]

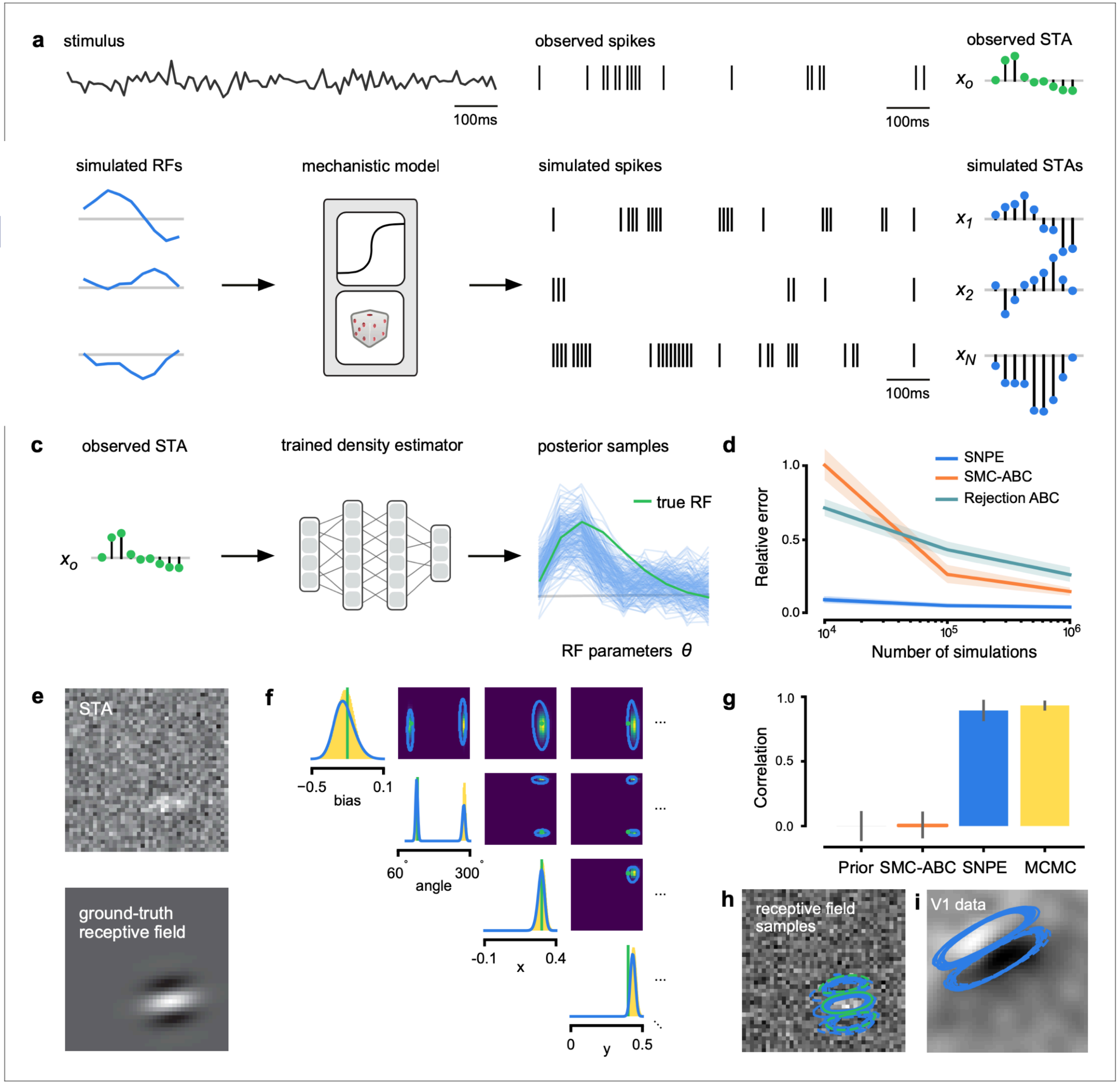


eLife

RESEARCH

Training deep neural density estimators to identify mechanistic models of neural dynamics

Pedro J Gonçalves[1,2†*], Jan-Matthis Lueckmann[1,2†*], Michael Deistler[1,3†*], Marcel Nonnenmacher[1,2,4], Kaan Öcal[2,5], Giacomo Bassetto[1,2], Chaitanya Chintaluri[6,7], William F Podlaski[6], Sara A Haddad[8], Tim P Vogels[6,7], David S Greenberg[1,4], Jakob H Macke[1,2,3,9*]

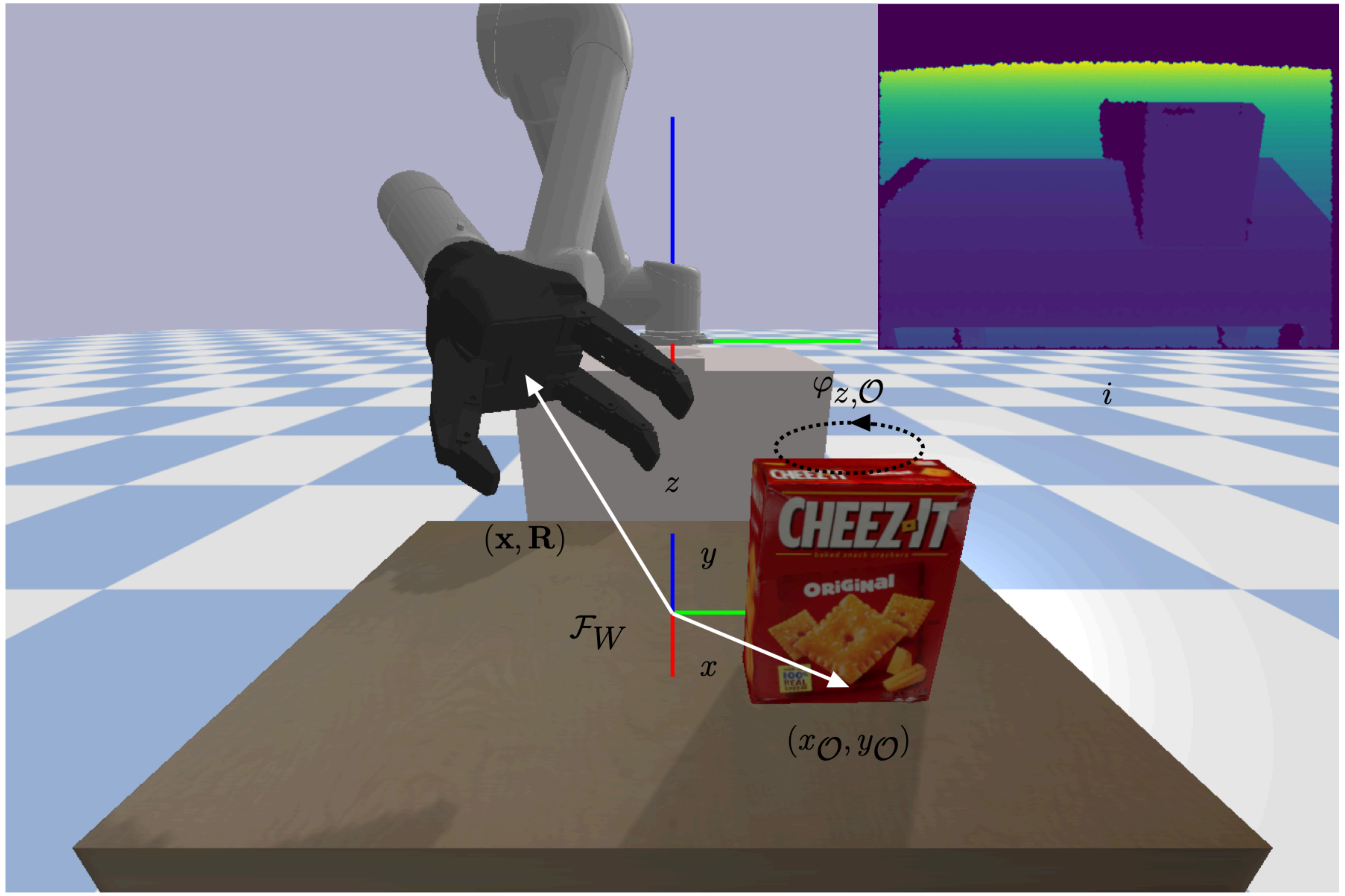


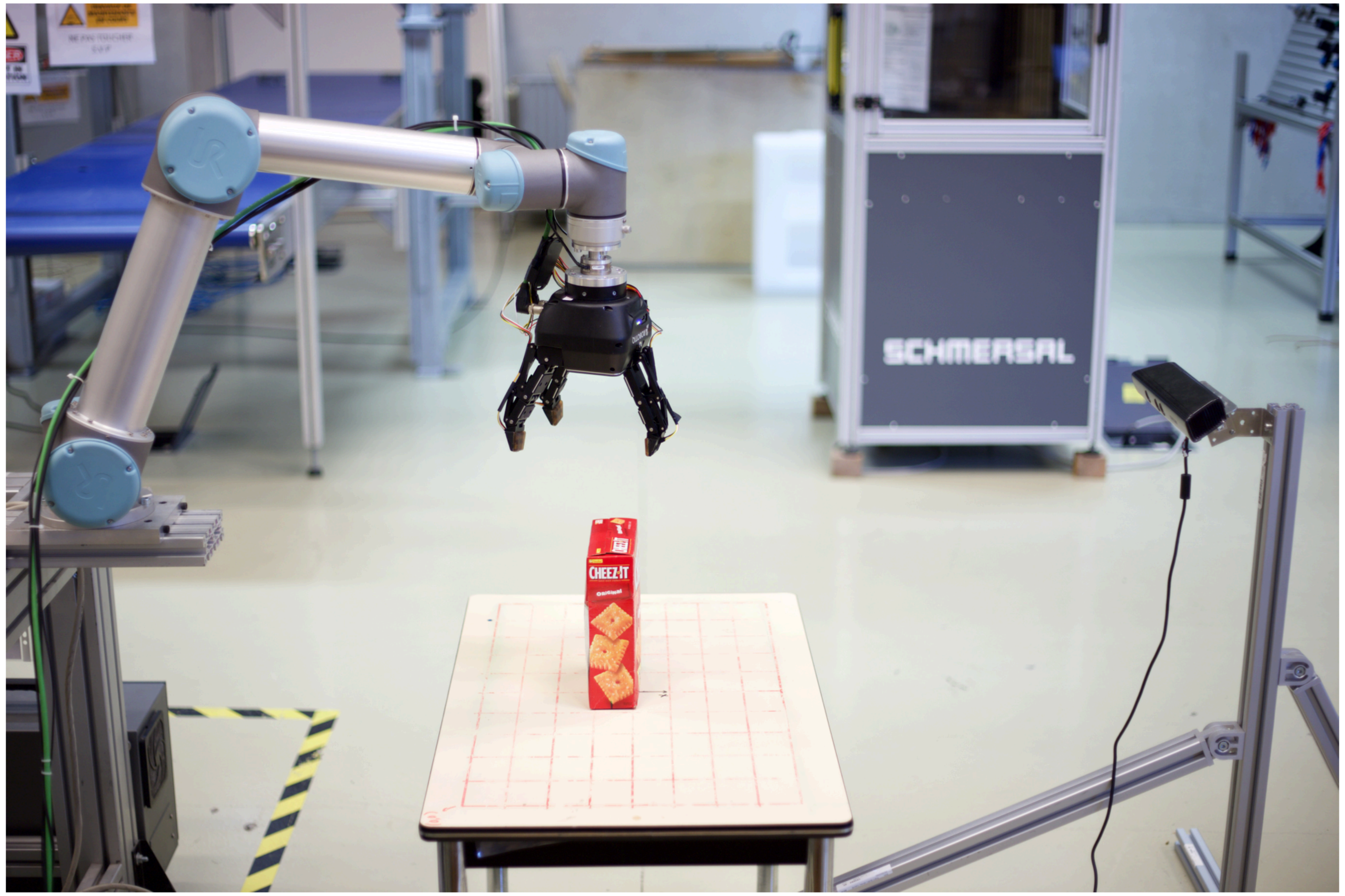
Simulation-based Bayesian inference for multi-fingered robotic grasping

Norman Marlier
University of Liège
norman.marlier@uliege.be

Olivier Brüls
University of Liège
o.bruls@uliege.be

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



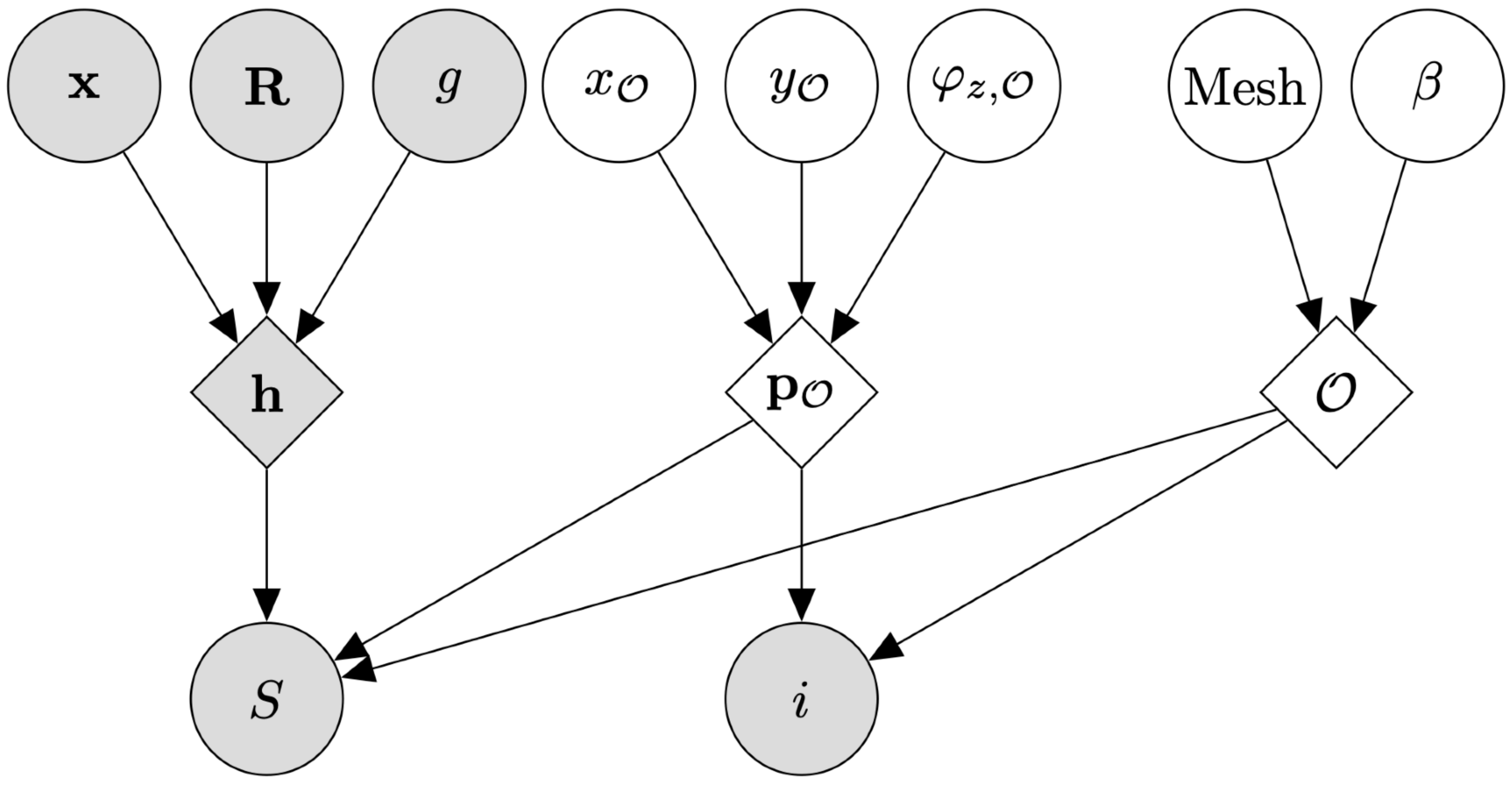


Simulation-based Bayesian inference for multi-fingered robotic grasping

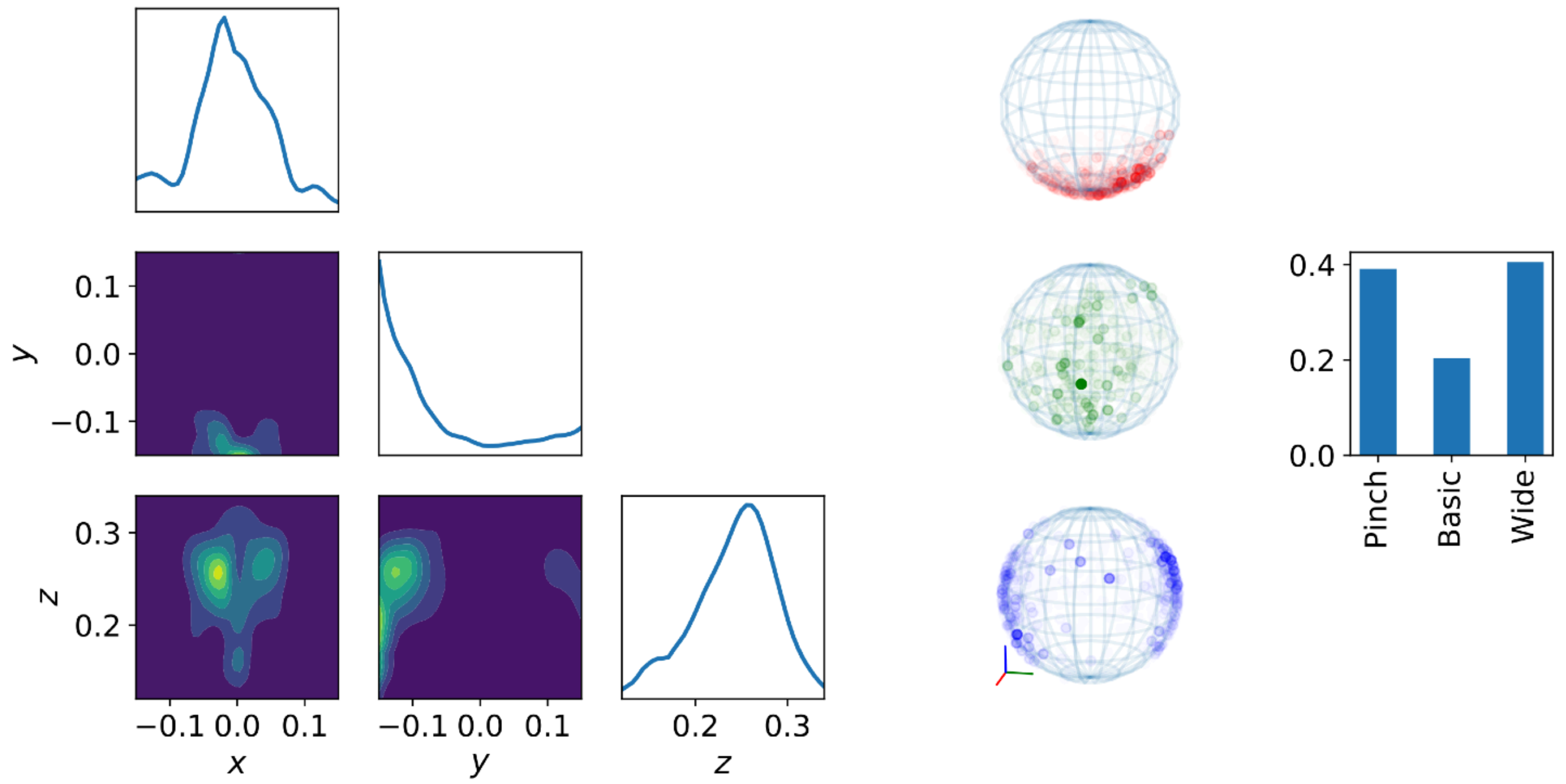
Norman Marlier
University of Liège
norman.marlier@uliege.be

Olivier Brüls
University of Liège
o.bruls@uliege.be

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



(a)



Variable	Prior
x	uniform$(-0.15, 0.15)$
y	uniform$(-0.15, 0.15)$
z	uniform$(0.12, 0.34)$
$\mathbf{R}$	mixture of power spherical(μ_i, κ)
g	categorical$(\{\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\})$
$x_{\mathcal{O}}$	uniform$(-0.05, 0.05)$
$y_{\mathcal{O}}$	uniform$(-0.05, 0.05)$
$\varphi_{z,\mathcal{O}}$	uniform$(-\pi, \pi)$
Mesh	uniform in the set of objects
β	uniform(0.9, 1.1)

(b)

Figure 2: (a) Probabilistic graphical model of the environment. Gray nodes correspond to observed variables and white nodes to unobserved variables. (b) Prior distributions.

SIMULATION-BASED BAYESIAN INFERENCE FOR MULTI-FINGERED ROBOTIC GRASPING

Norman Marlier
University of Liège
norman.marlier@uliege.be

Olivier Brüls
University of Liège
o.bruls@uliege.be

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