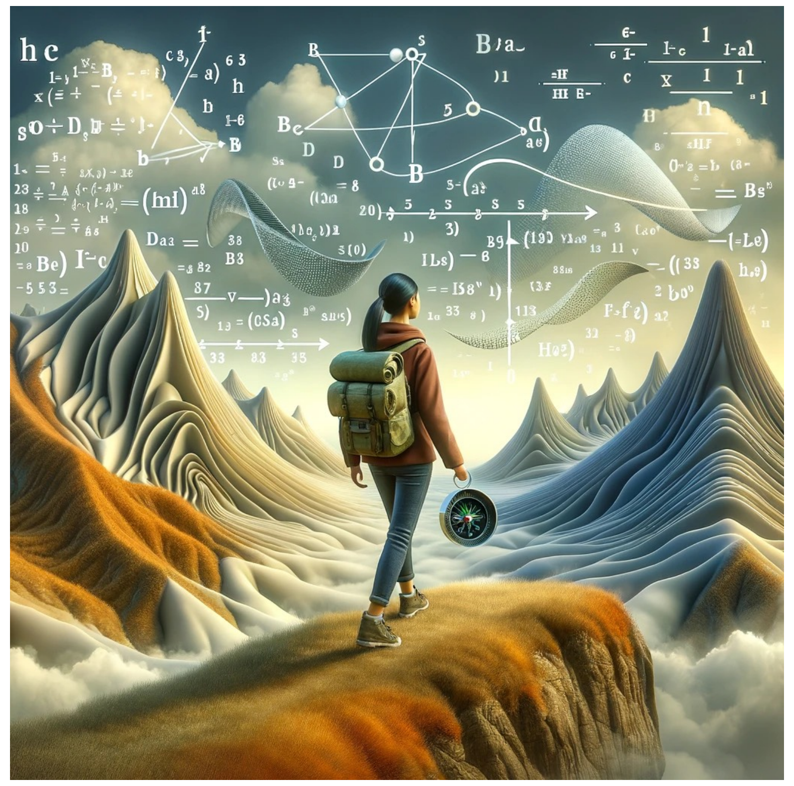
Simulation and inference in neuroscience



Lecture 9: Neural likelihood estimation

March 2025

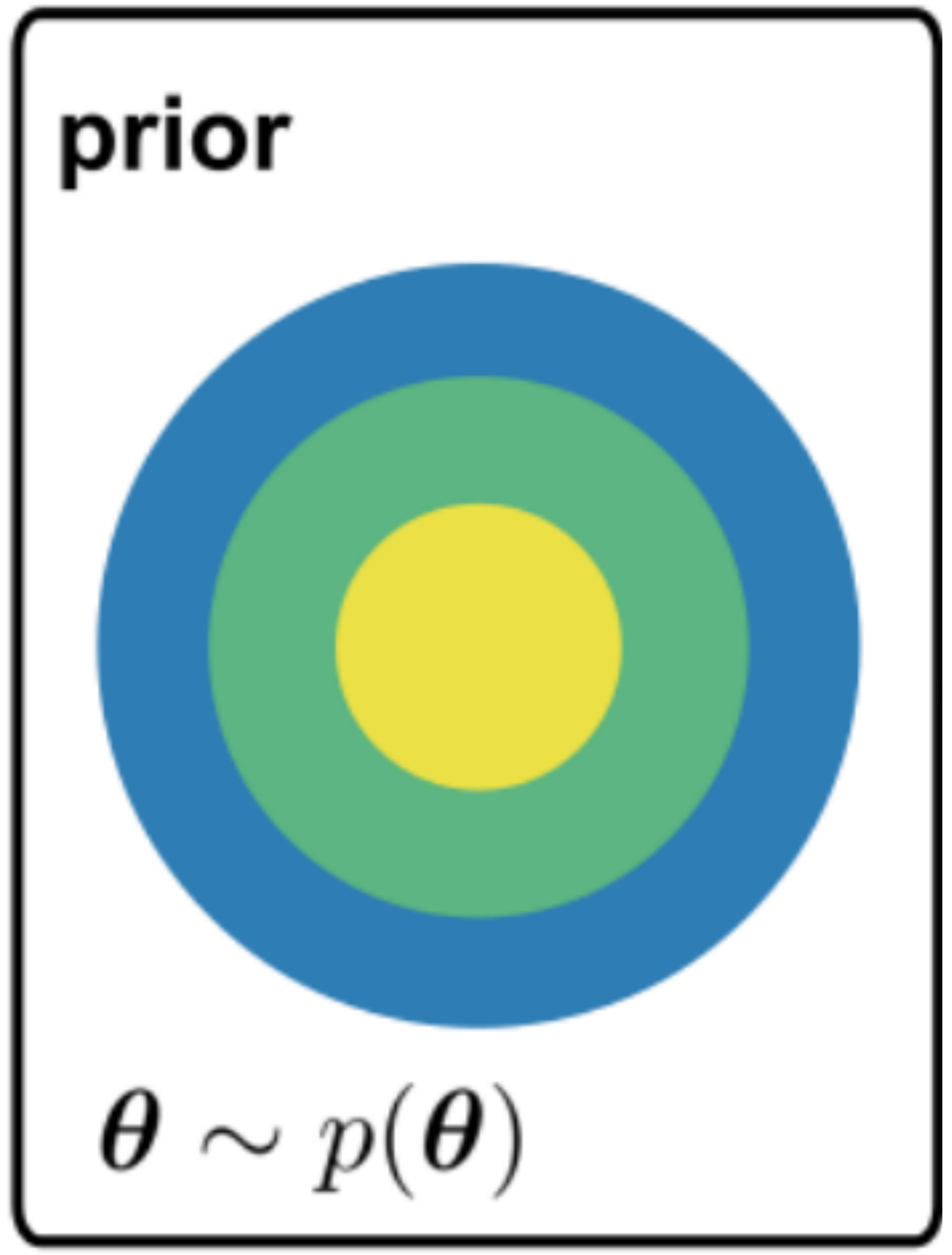
Pedro Gonçalves
goncalveslab.sites.vib.be/en

Philipp Berens, Jonas Beck
https://hertie.ai/data-science/team

9.1 Recap: Neural Posterior Estimation (NPE)

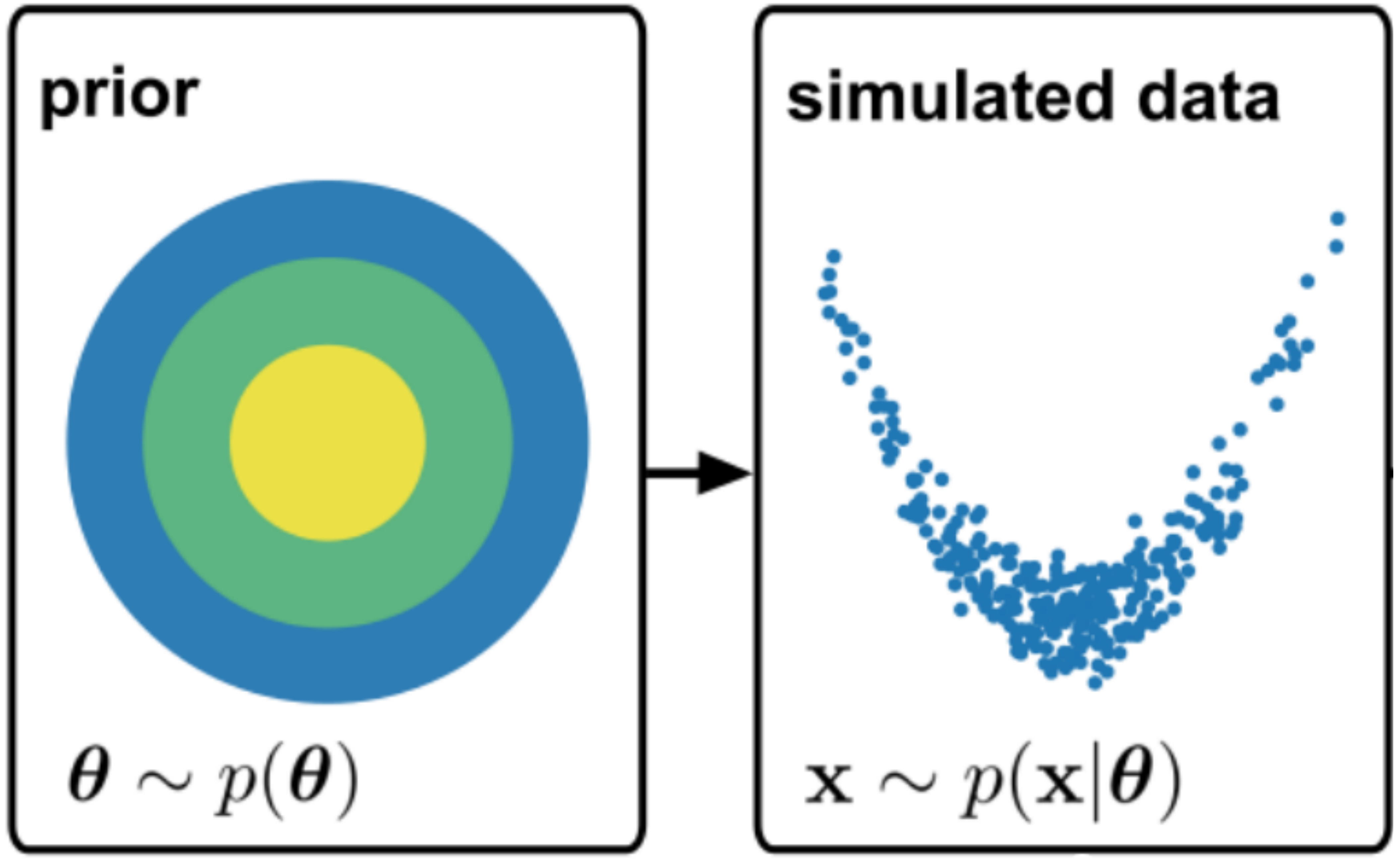
NPE: step 1

Sample from the prior



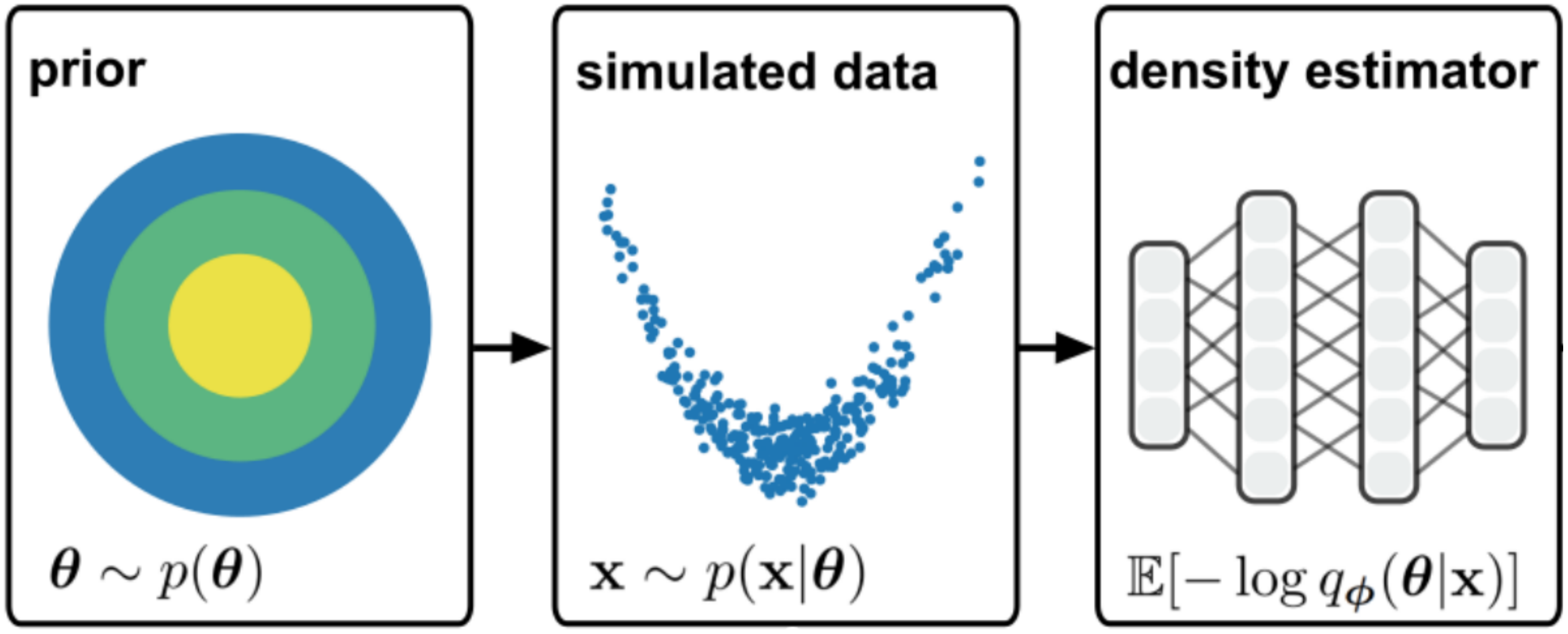
NPE: step 2

Generate simulations



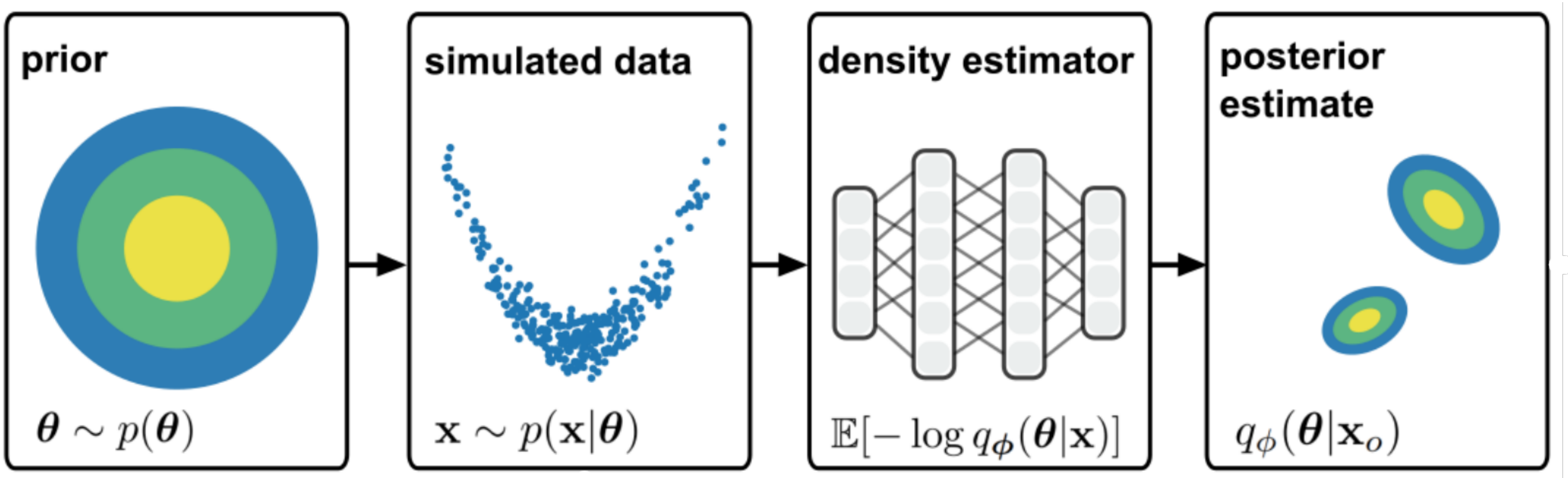
NPE: step 3

Train
estimator



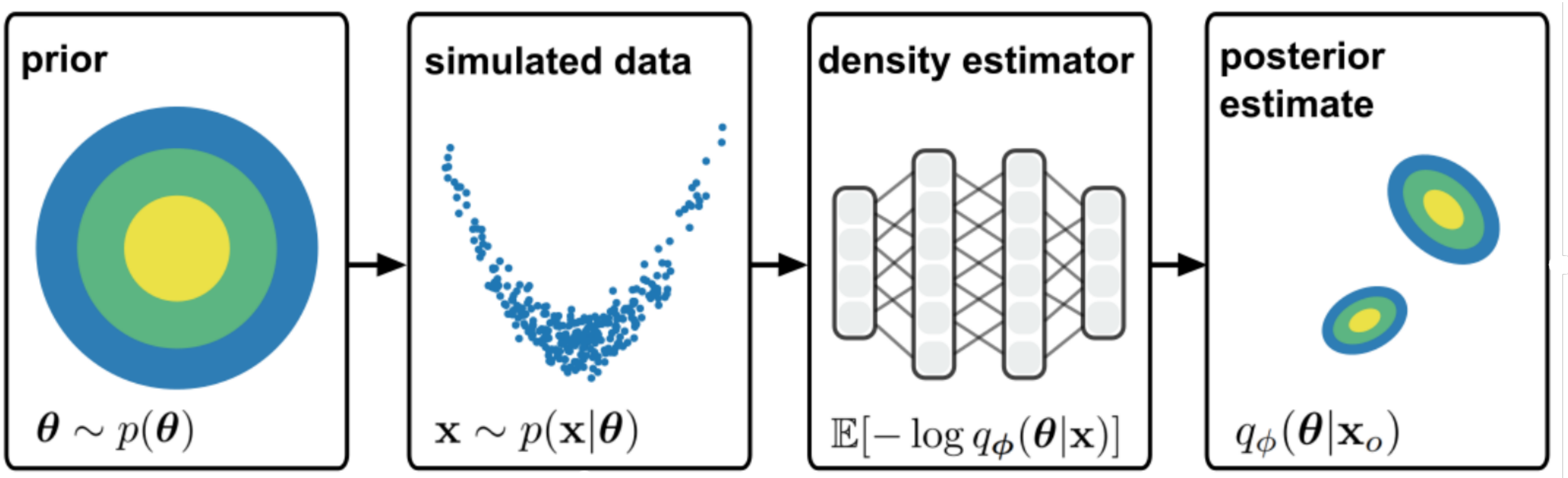
NPE: step 4

Evaluate at x_o



NPE: step 4

Evaluate at x_o

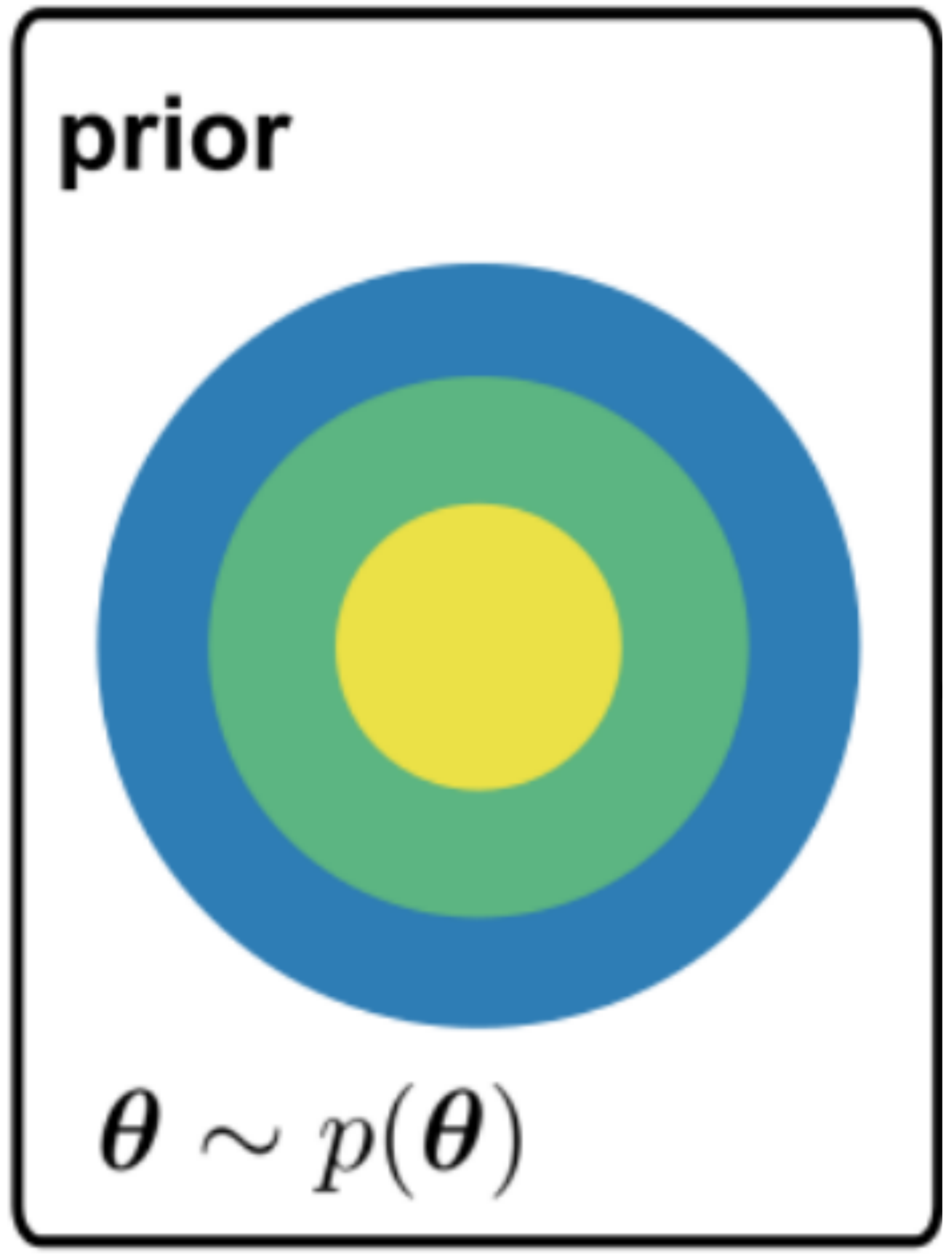


NPE is amortised: after training $q_\phi(\theta \mid x)$, we can evaluate it for any observation x_o

9.2 Neural Likelihood Estimation (NLE)

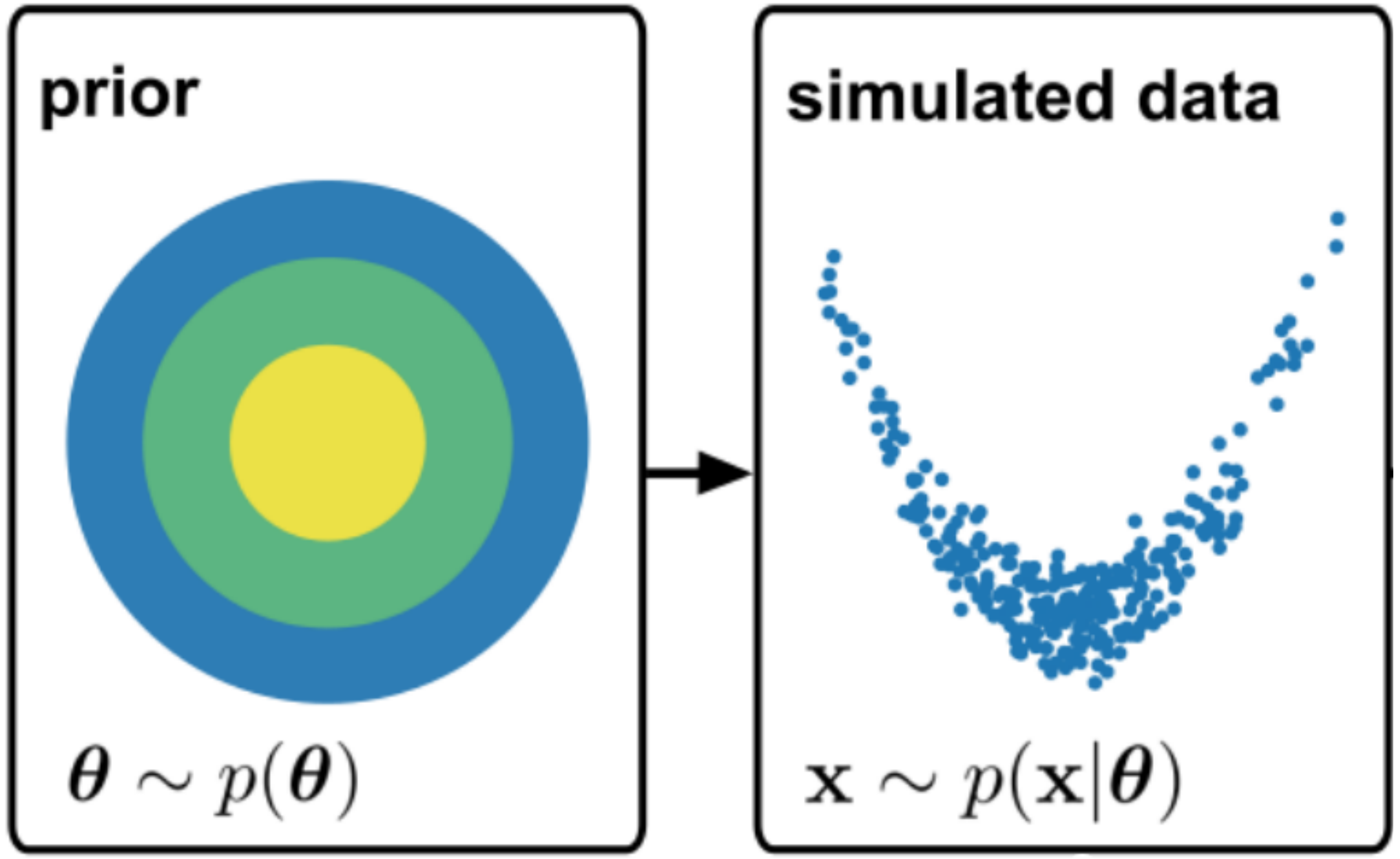
NLE: step 1

Sample from the prior



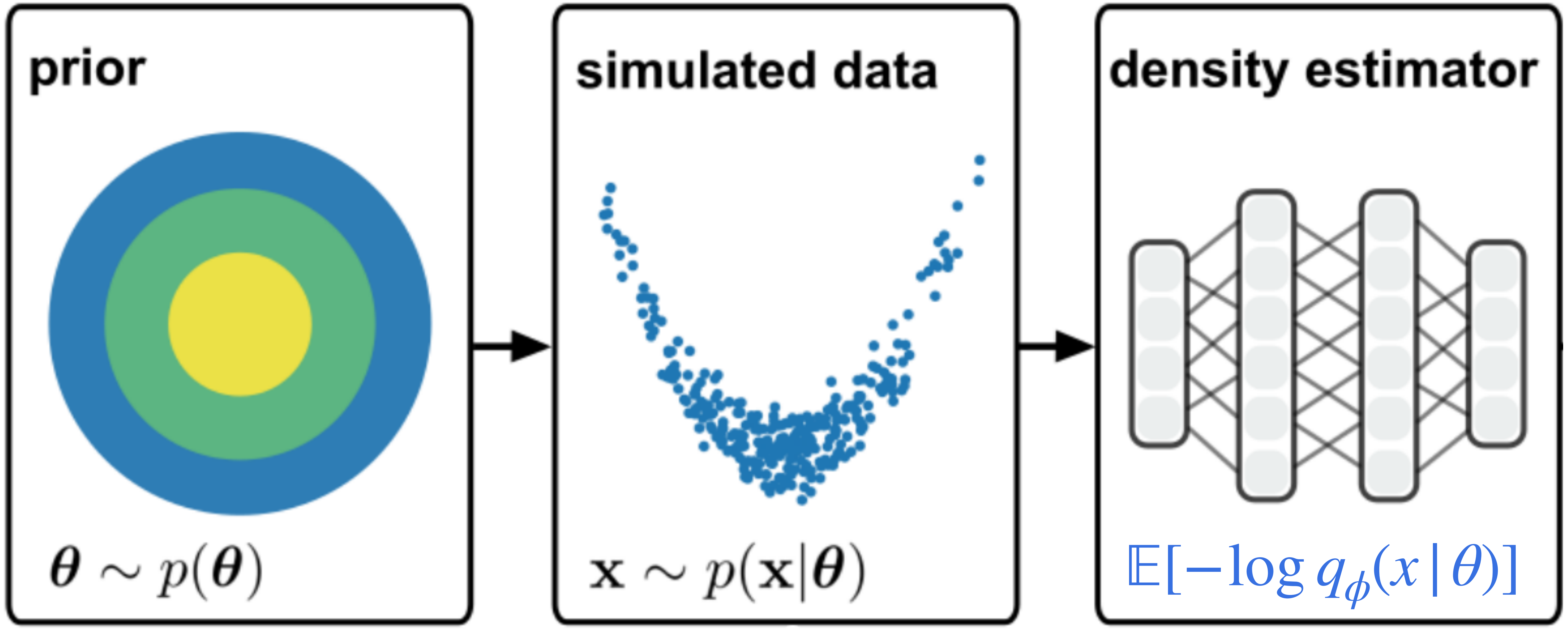
NLE: step 2

Generate simulations

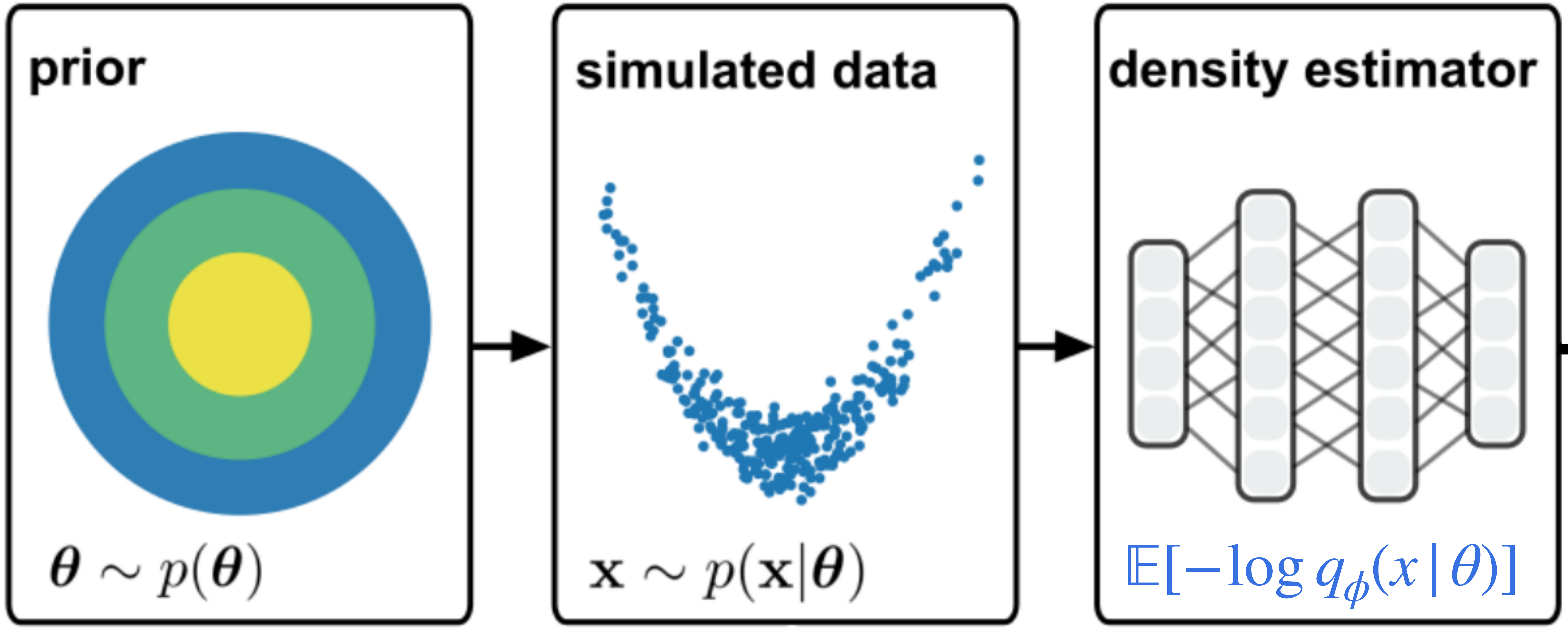


NLE: neural density estimators to learn the likelihood instead of the posterior. Step 3

Train estimator

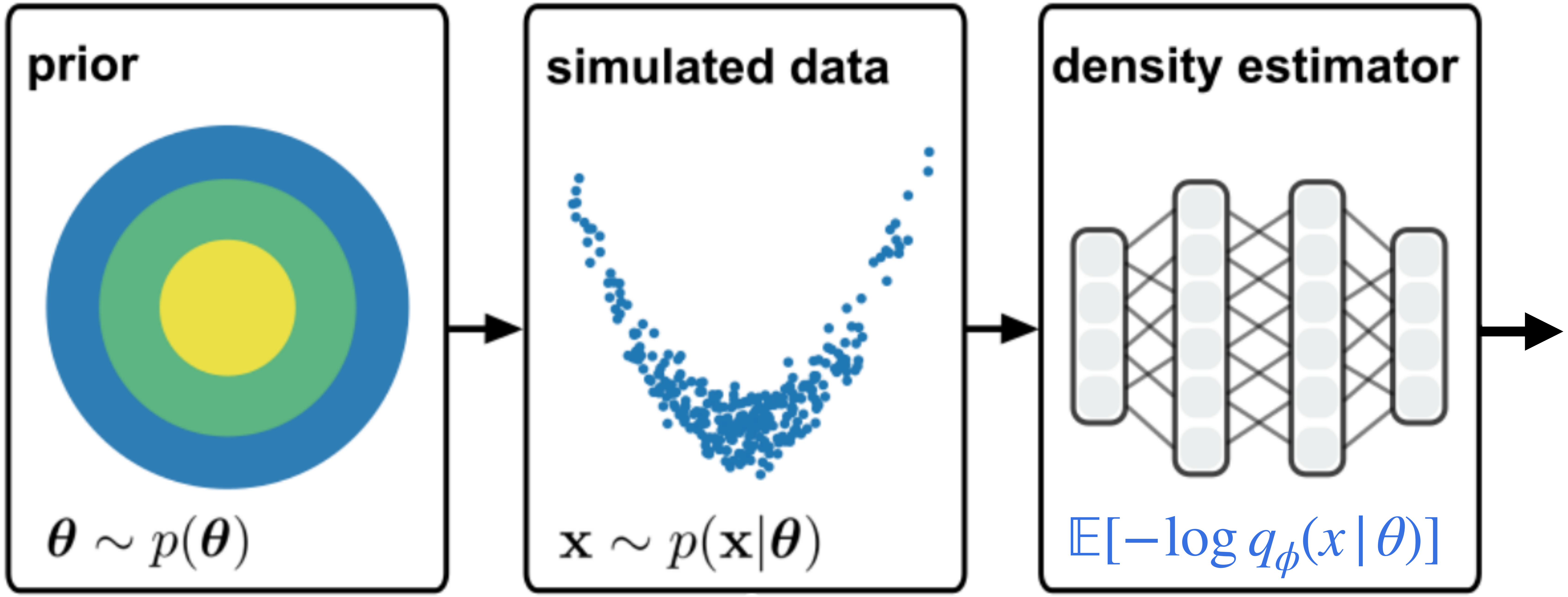


NLE: neural density estimators to learn the likelihood instead of the posterior. Step 4



Evaluate $p(x_o\,|\,\theta)$.

NLE: neural density estimators to learn the likelihood instead of the posterior. Step 4



Evaluate $p(x_o \mid \theta)$.

How can we evaluate $p(\theta \mid x_o)$?

Solution: Bayesian inference with tractable likelihood

- $p(\theta | x_o) = \dfrac{p(x_o | \theta)p(\theta)}{p(x_o)}$
- But, $p(x_o) = \int_\theta p(x_o | \theta)p(\theta)d\theta$, which is intractable in general.
- Two main strategies (not covered in class; suggested reading material at the end of slides):
    1. Variational Inference
    2. Markov Chain Monte Carlo Sampling

Neural likelihood estimation (NLE)

- The five main steps of NLE:
  1. Sample from the prior: $\theta_n \sim p(\theta)$
  2. Run simulations: $x_n \sim p(x \mid \theta_n)$
  3. Train a neural density estimator $q_\phi(x \mid \theta)$ by minimising $\mathscr{L}(\phi) = \mathbb{E}[-\log q_\phi(x \mid \theta)]$
  4. Evaluate the estimator at x_o to get an estimate of the likelihood function $p(x_o \mid \theta)$.
  5. Get samples from $p(\theta \mid x_o)$ with Markov Chain Monte Carlo (MCMC) sampling or estimate posterior $p(\theta \mid x_o)$ with variational inference.

- After training $q_\phi(x \mid \theta)$, we can evaluate it for any observation x_o, but need to estimate the posterior $p(\theta \mid x_o)$ for each x_o (step 5 above).

9.3 When to use NLE instead of NPE

NPE

- Amortized inference: after training, we can evaluate $p(\theta \mid x_o)$ for any observation x_o.
- Requires special corrections if θ is not sampled from prior $p(\theta)$ in the training data (more on this later).
- For high-dimensional parameter space (θ), learning $p(\theta \mid x)$ can be very challenging.

NLE

- Easy to deal with i.i.d. observations: $p(x_1^o, x_2^o, \ldots, x_m^o \mid \theta) = \prod_n p(x_n^o \mid \theta)$.
- Can use training data with θ from any distribution.
- For high-dimensional observations x, learning $p(x \mid \theta)$ can be very challenging.
- Requires MCMC.

Learning $p(\theta \mid x)$ directly vs. learning $p(x \mid \theta)$ for MCMC sampling

- Consider $\dim(x)$ and $\dim(\theta)$. Learning neural density estimators in high-dimensional spaces is hard, but neural nets can take high-dimensional input easily. So, use NPE when $\dim(x) >> \dim(\theta)$, and NLE when $\dim(x) << \dim(\theta)$.

- Consider structure in x or θ. When one of these is an image (or time series), we could use a CNN (RNN) to process it as input. Specialized neural density estimators also exist for structured outputs. Other structure (graphs, sets, etc.) can also be exploited.

- Feasibility of MCMC depends on the shape and dimension of the posterior.

- All of these considerations are active areas of research, and the set of SBI problems for which these methods have been tested remains small.

Lecture 9: Neural likelihood estimation

- In NLE, we (1) approximate an unknown likelihood function $p(x_o \mid \theta)$ by minimising the KL-divergence to our model q_ϕ, (2) use "standard" Bayesian inference tools to get an approximation to the posterior $p(\theta \mid x_o)$.
- The five main steps of NLE:
  1. Sample from the prior: $\theta_n \sim p(\theta)$
  2. Run simulations: $x_n \sim p(x \mid \theta_n)$
  3. Train a neural density estimator $q_\phi(x \mid \theta)$ by minimising $\mathscr{L}(\phi) = \mathbb{E}[-\log q_\phi(x \mid \theta)]$
  4. Evaluate the estimator at x_o to get an estimate of the likelihood function $p(x_o \mid \theta)$.
  5. Get samples from $p(\theta \mid x_o)$ with Markov Chain Monte Carlo (MCMC) sampling or estimate posterior $p(\theta \mid x_o)$ with variational inference.
- After training $q_\phi(x \mid \theta)$, we can evaluate it for any observation x_o, but need to estimate the posterior $p(\theta \mid x_o)$ for each x_o (step 5 above).

Further reading on sampling and variational inference

- Nice introduction to MCMC and variational inference at https://towardsdatascience.com/bayesian-inference-problem-mcmc-and-variational-inference-25a8aa9bce29
- Variational Inference: A Review for Statisticians. (2018) David M. Blei, Alp Kucukelbir, Jon D. McAuliffe
- An Introduction to MCMC for Machine Learning. (2003) Christophe Andrieu, Nando de Freitas, Arnaud Doucet & Michael I. Jordan