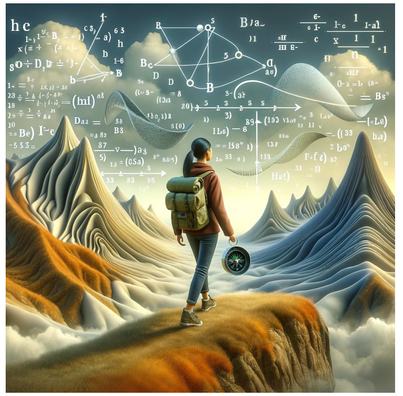
Simulation-based inference



Lecture 4: SBIverse

April 2024

Pedro Gonçalves, Anastasia Krouglova
goncalveslab.sites.vib.be/en

Guy Moss
mackelab.org

VIB

nerf

UNIVERSITÄT TÜBINGEN

machine learning
new perspectives
for science

KU LEUVEN

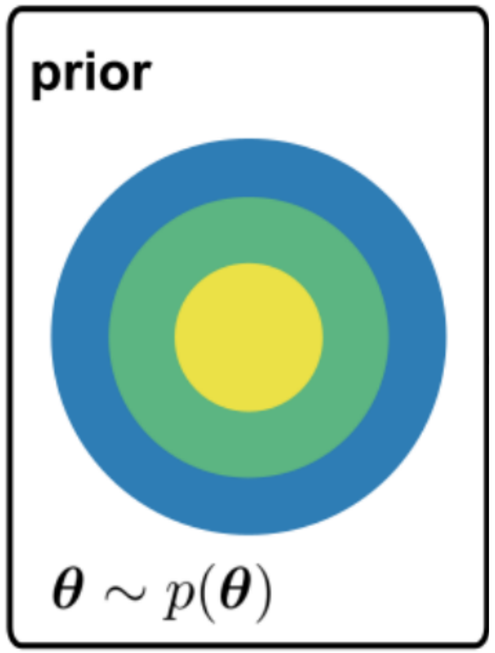
VIB.AI
CENTER FOR AI &
COMPUTATIONAL BIOLOGY

Tübingen AI
Center

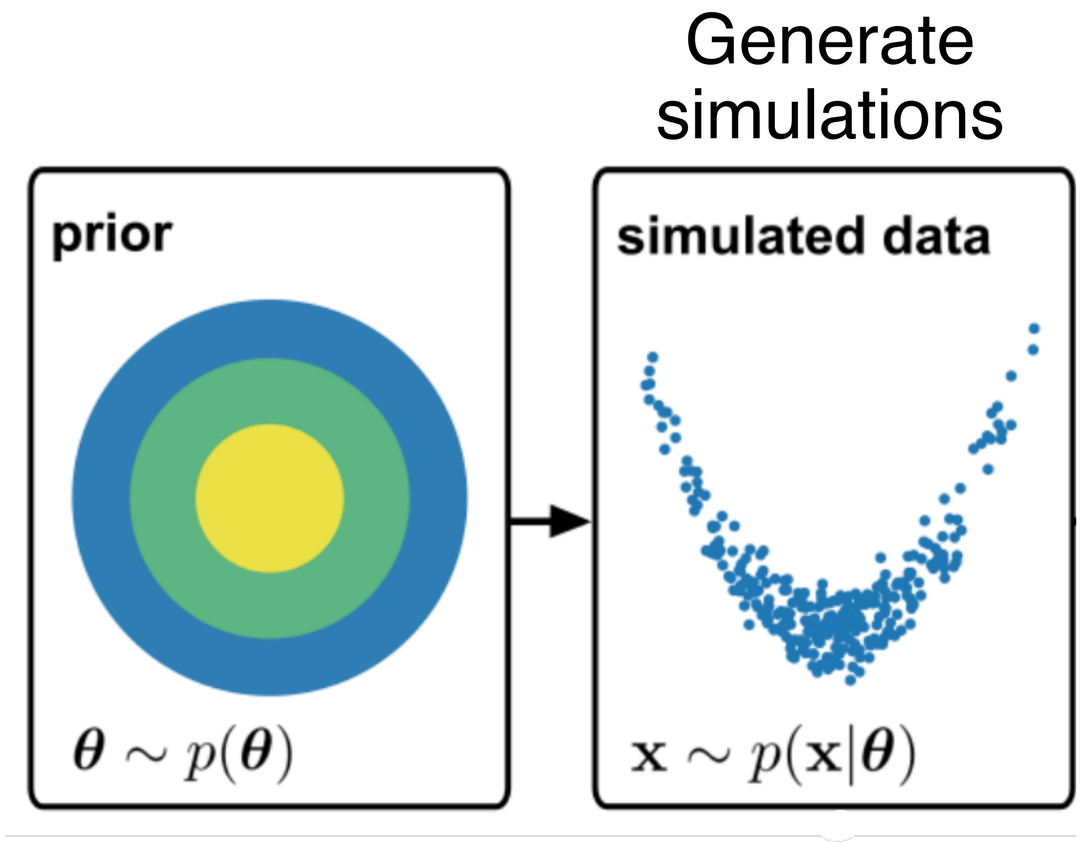
4.1 Recap: Neural Posterior Estimation (NPE)

NPE: step 1

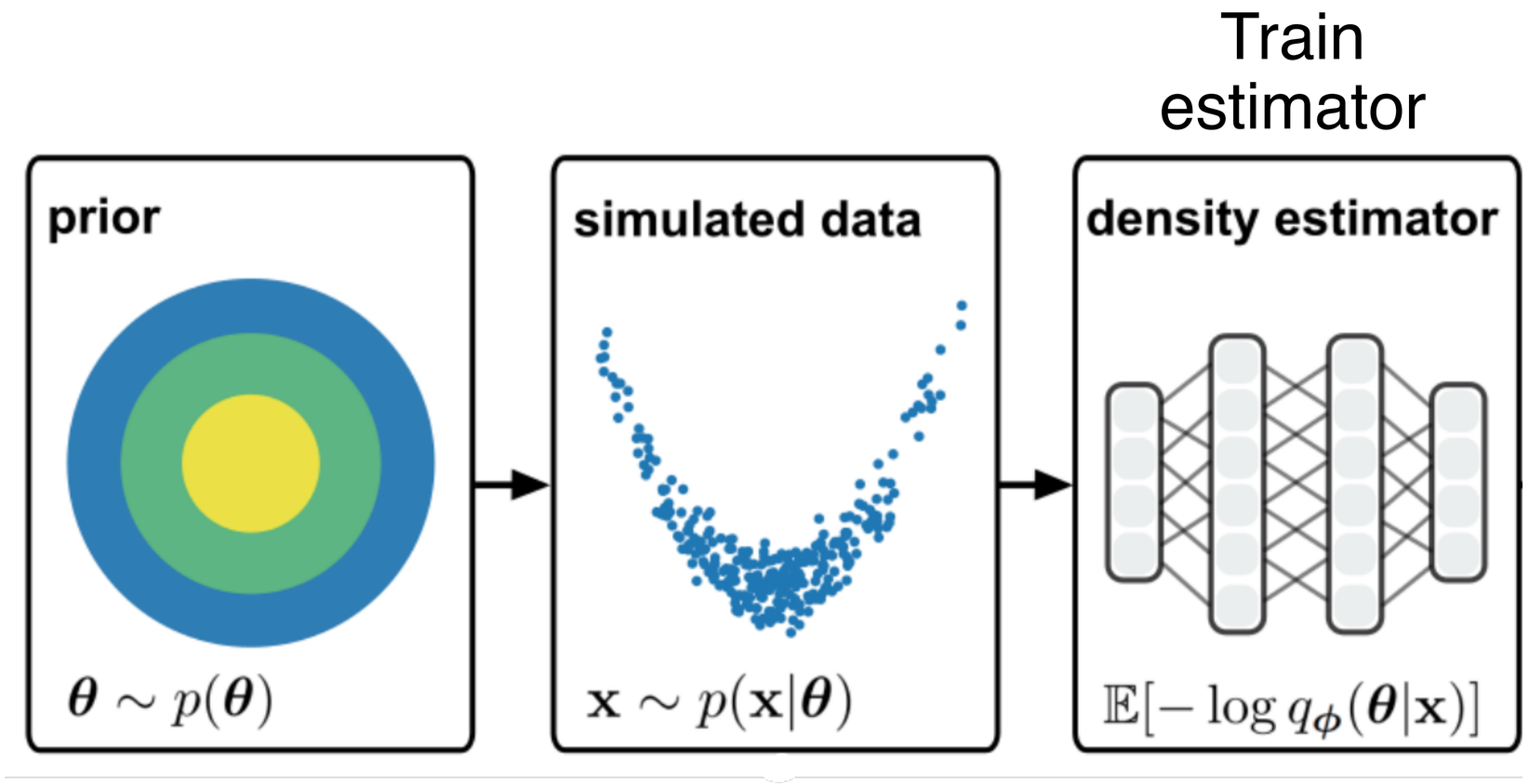
Sample from the prior



NPE: step 2

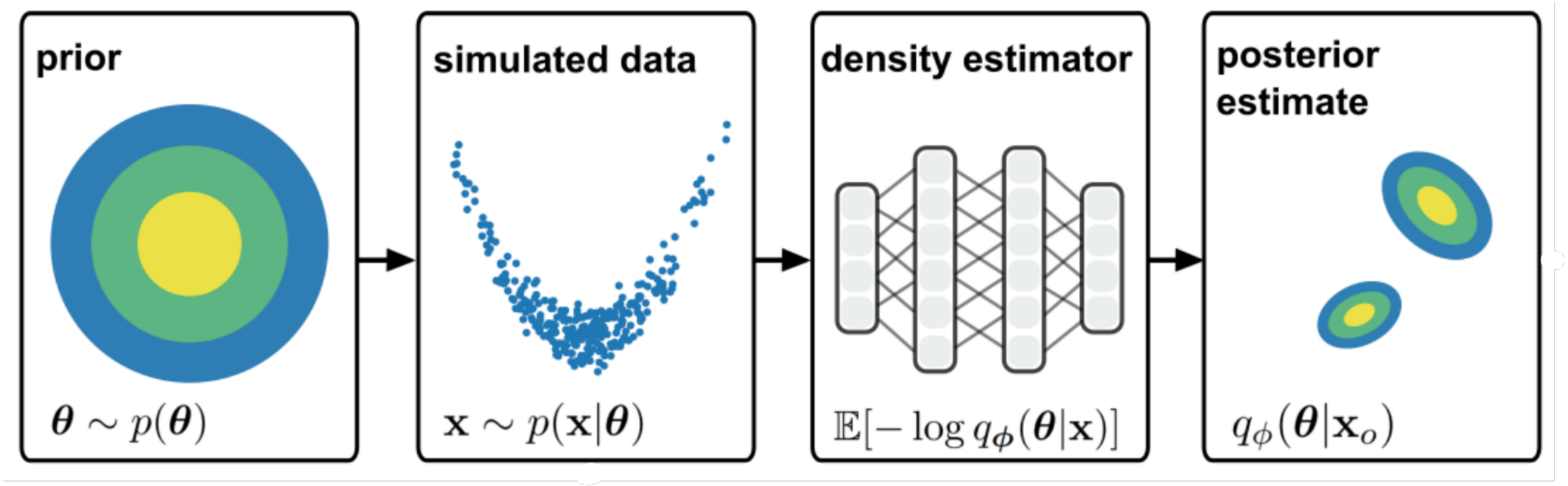

Generate simulations
prior
$\boldsymbol{\theta} \sim p(\boldsymbol{\theta})$
simulated data
$\mathbf{x} \sim p(\mathbf{x}|\boldsymbol{\theta})$

NPE: step 3


Train
estimator
prior
$\boldsymbol{\theta} \sim p(\boldsymbol{\theta})$
simulated data
$\mathbf{x} \sim p(\mathbf{x}|\boldsymbol{\theta})$
density estimator
$\mathbb{E}[-\log q_{\boldsymbol{\phi}}(\boldsymbol{\theta}|\mathbf{x})]$

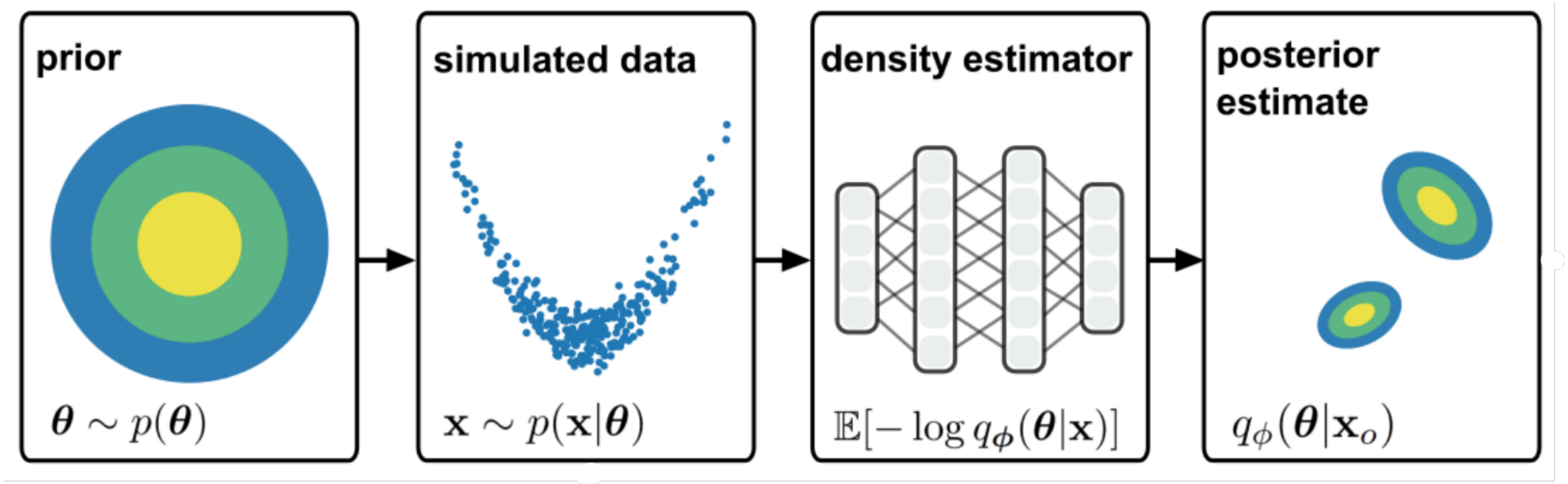
NPE: step 4

Evaluate at x_o



NPE: step 4

Evaluate at x_o

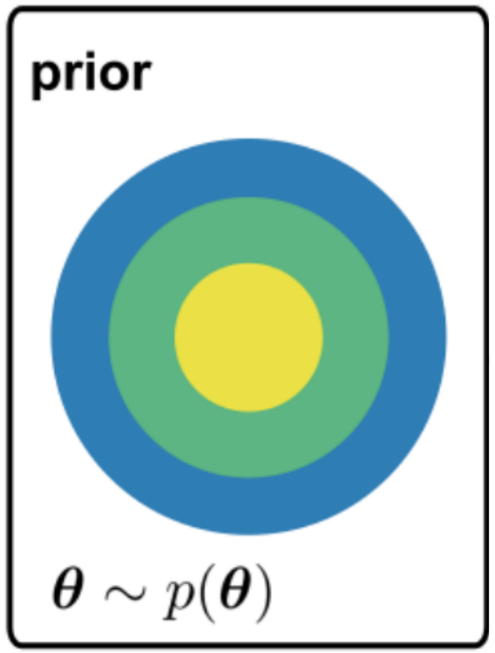


NPE is amortised: after training $q_\phi(\theta \mid x)$, we can evaluate it for any observation x_o

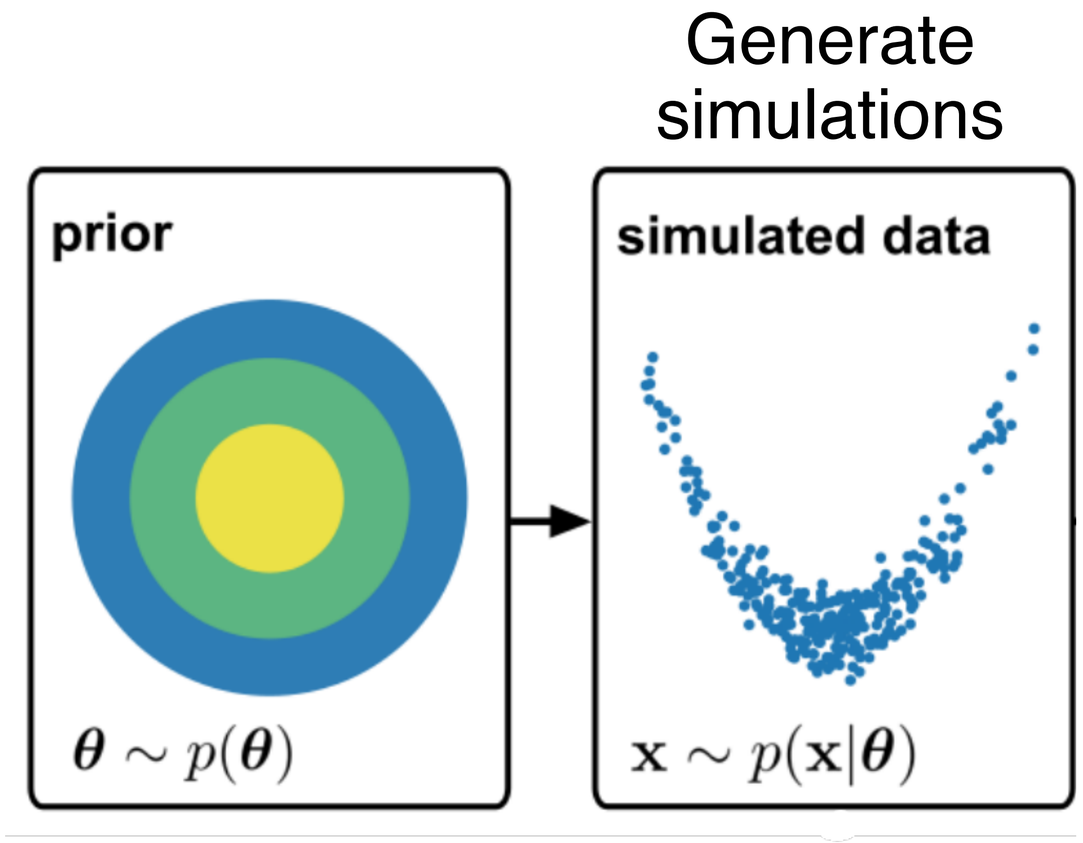
4.2 Neural Likelihood Estimation (NLE)

NLE: step 1

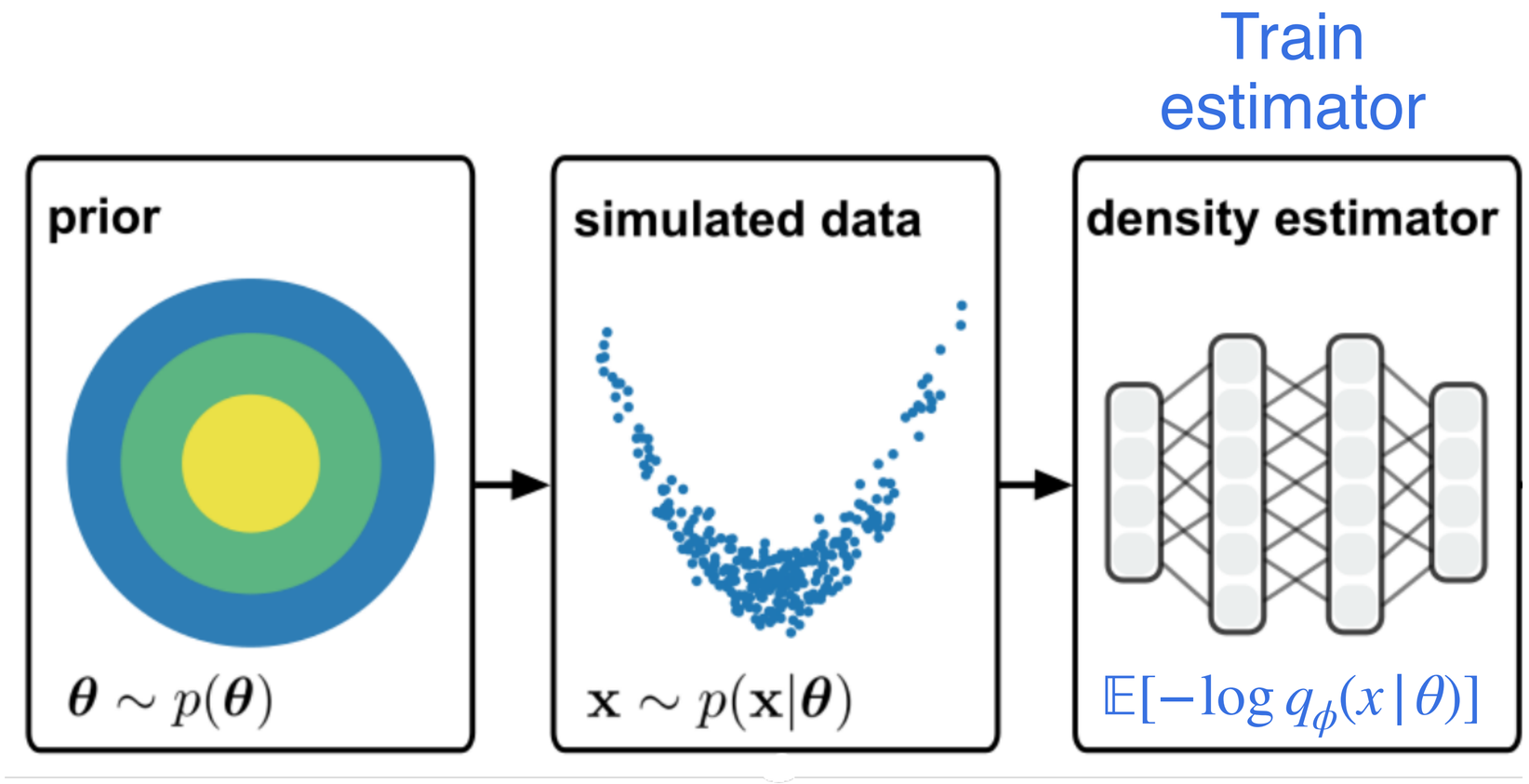
Sample from the prior



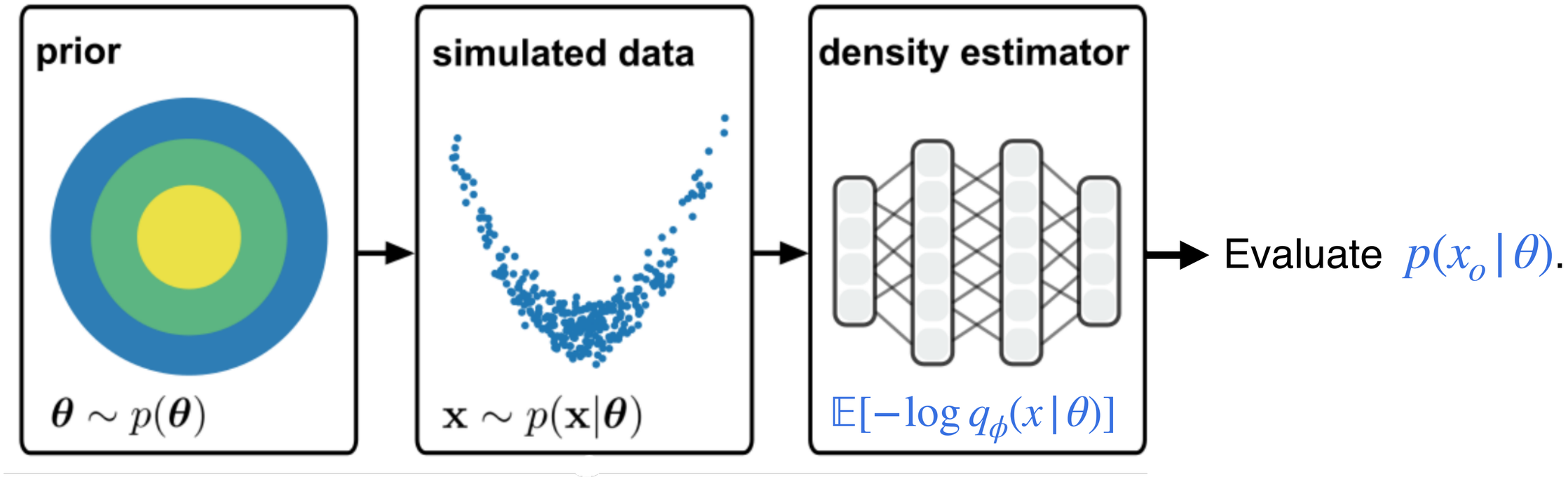
NLE: step 2


Generate simulations
prior
$\boldsymbol{\theta} \sim p(\boldsymbol{\theta})$
simulated data
$\mathbf{x} \sim p(\mathbf{x}|\boldsymbol{\theta})$

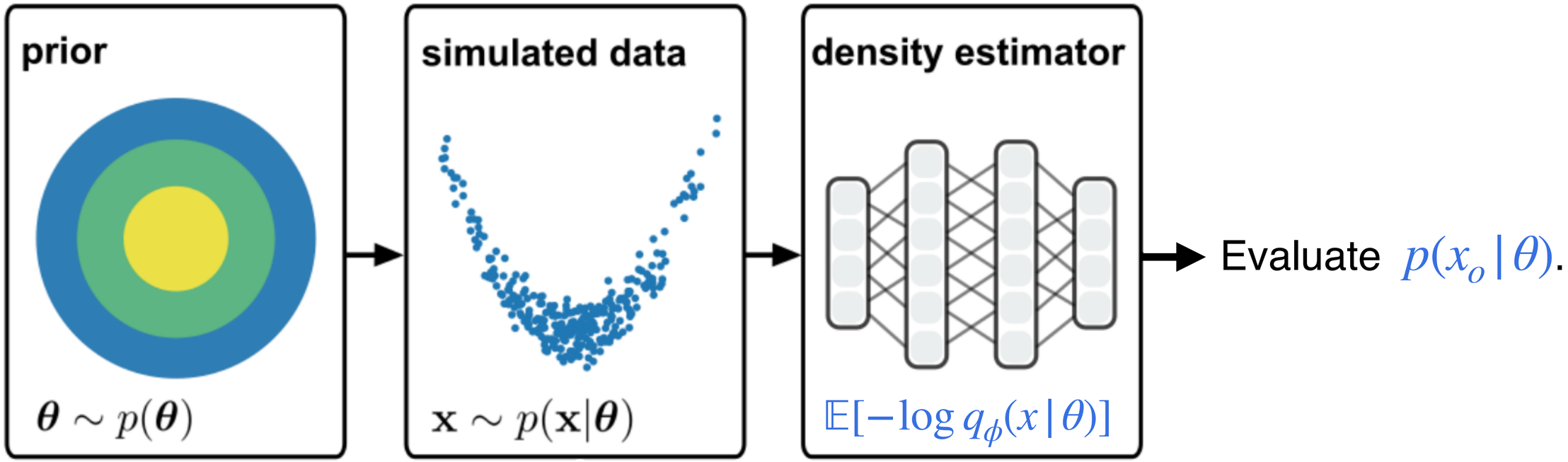
NLE: neural density estimators to learn the likelihood instead of the posterior. Step 3



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



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



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

Solution: Bayesian inference with tractable likelihood

- $p(\theta \mid x_o) = \dfrac{p(x_o \mid \theta)p(\theta)}{p(x_o)}$
- But, $p(x_o) = \int_\theta p(x_o \mid \theta)p(\theta)d\theta$, 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).

4.3 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.

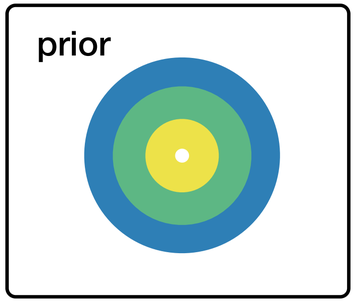
Interim summary

- 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 $\mathcal{L}(\phi) = \mathbb{E}[-\log q_\phi(x \mid \theta)]$
  4. Evaluate the estimator at x_o to get an estimate of the likelihood function $p(x_o \mid \theta)$.
  5. Estimate posterior $p(\theta \mid x_o)$ with variational inference or get samples from $p(\theta \mid x_o)$ with Markov Chain Monte Carlo (MCMC) sampling.
- 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.

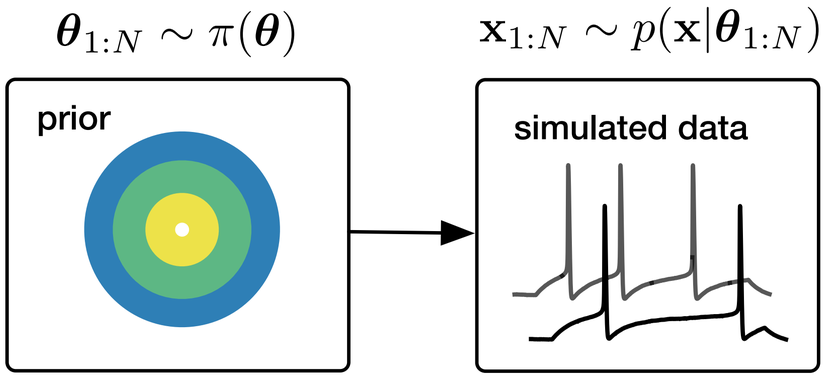
4.4 Sequential Neural Posterior Estimation (SNPE)

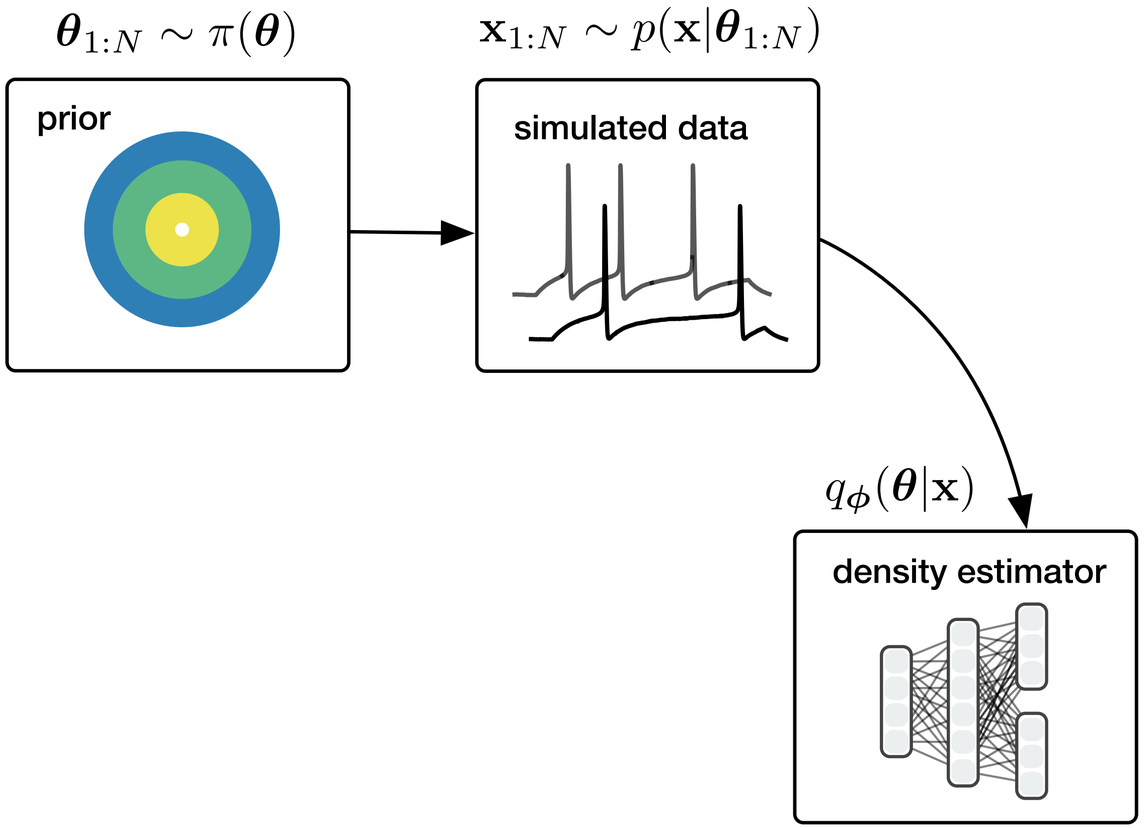
1. Sample parameters from prior

$$\boldsymbol{\theta}_{1:N} \sim \pi(\boldsymbol{\theta})$$

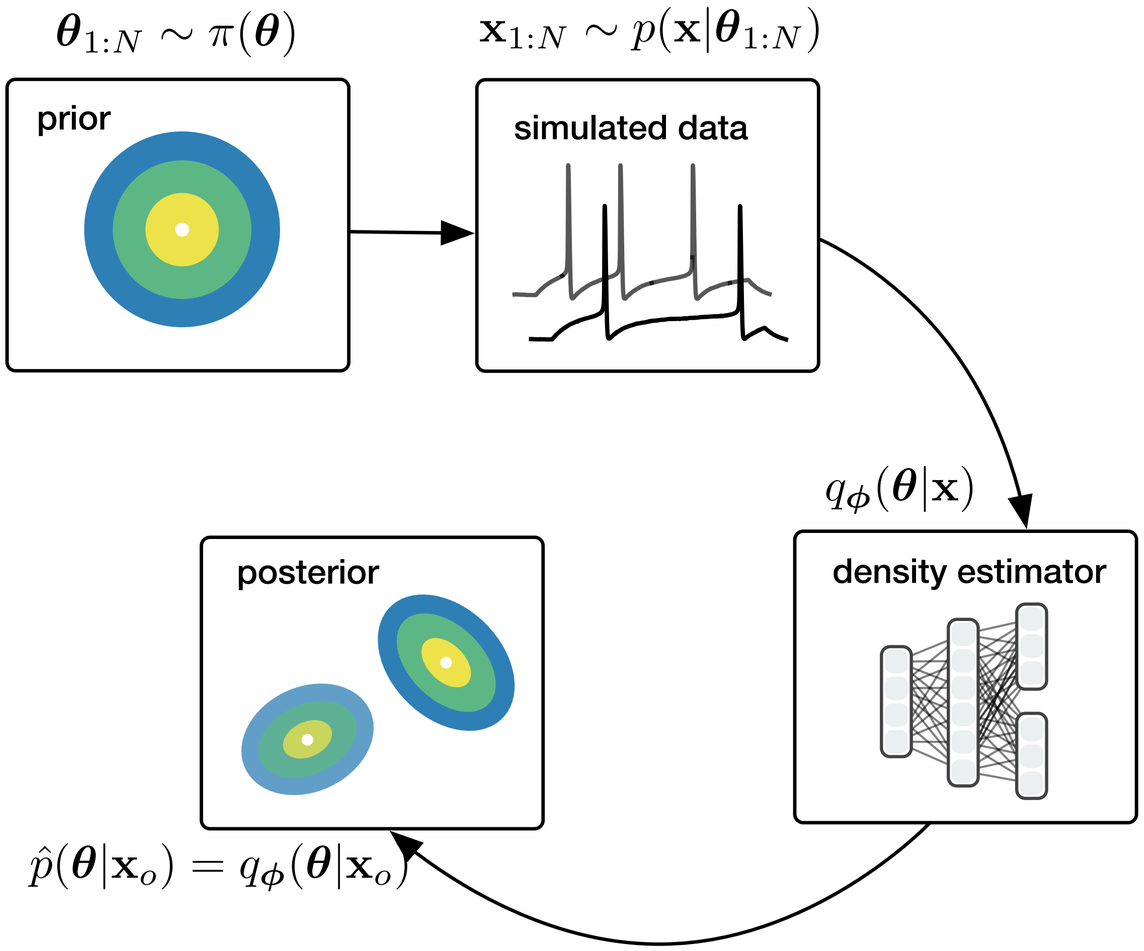


2. Simulate data from parameters





3. Train conditional density estimator to predict parameters from (simulated) data



4. Plug empirical data x_o into density estimator to calculate posterior

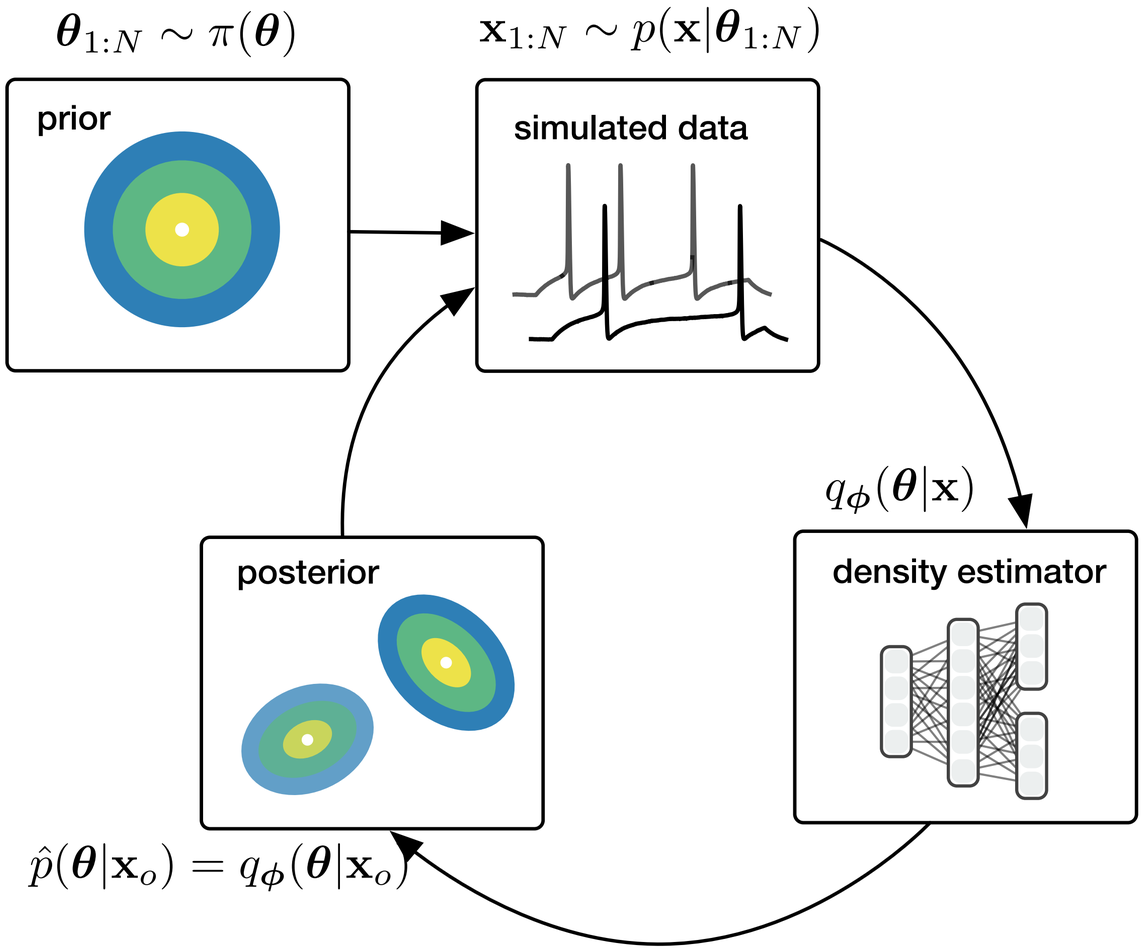
But…

- When sampling from the prior, many simulations might be very different from the observed data x_o (e.g., in the Hodgkin-Huxley model, many parameter sets might not lead to action potentials).

- Often, these simulations are less informative about the posterior distribution $p(\theta \mid x_o)$.

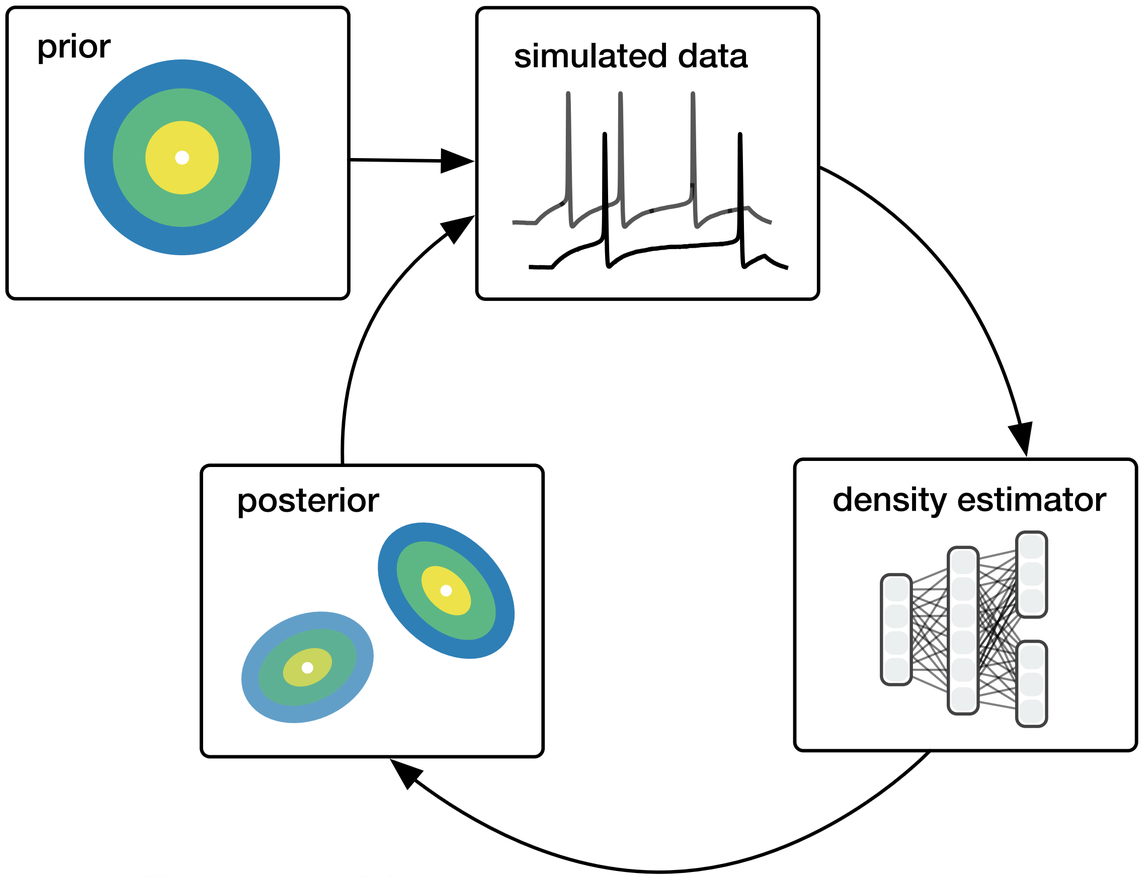
- We would prefer to sample from parameter regions that produce simulations close to x_o.

- This is crucial if we got a limited computational budget for running the simulator.



5. If needed, adaptively generate more simulations

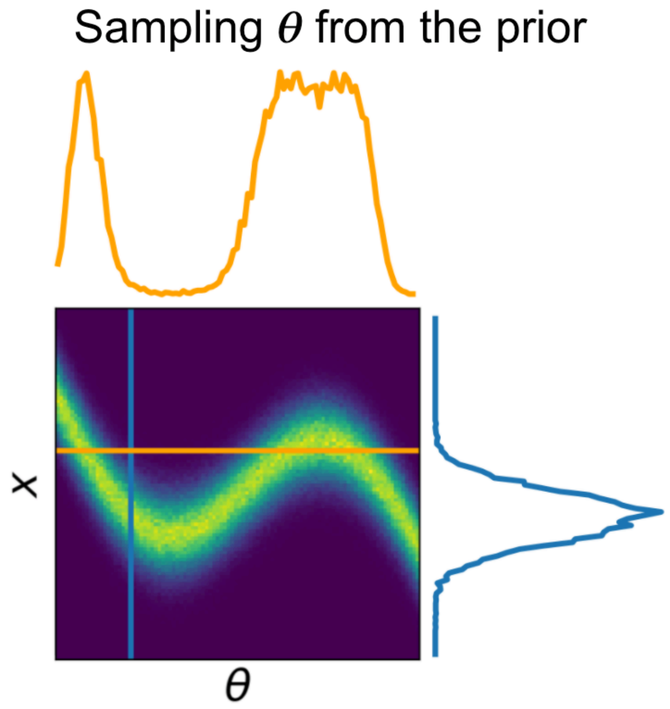
Sequential neural posterior estimation (SNPE)



Beaumont et al 2002, Blum & Francois 2010
Papamakarios & Murray NeurIPS 2016
Lueckmann, Goncalves et al NeurIPS 2017
Greenberg et al ICML 2019

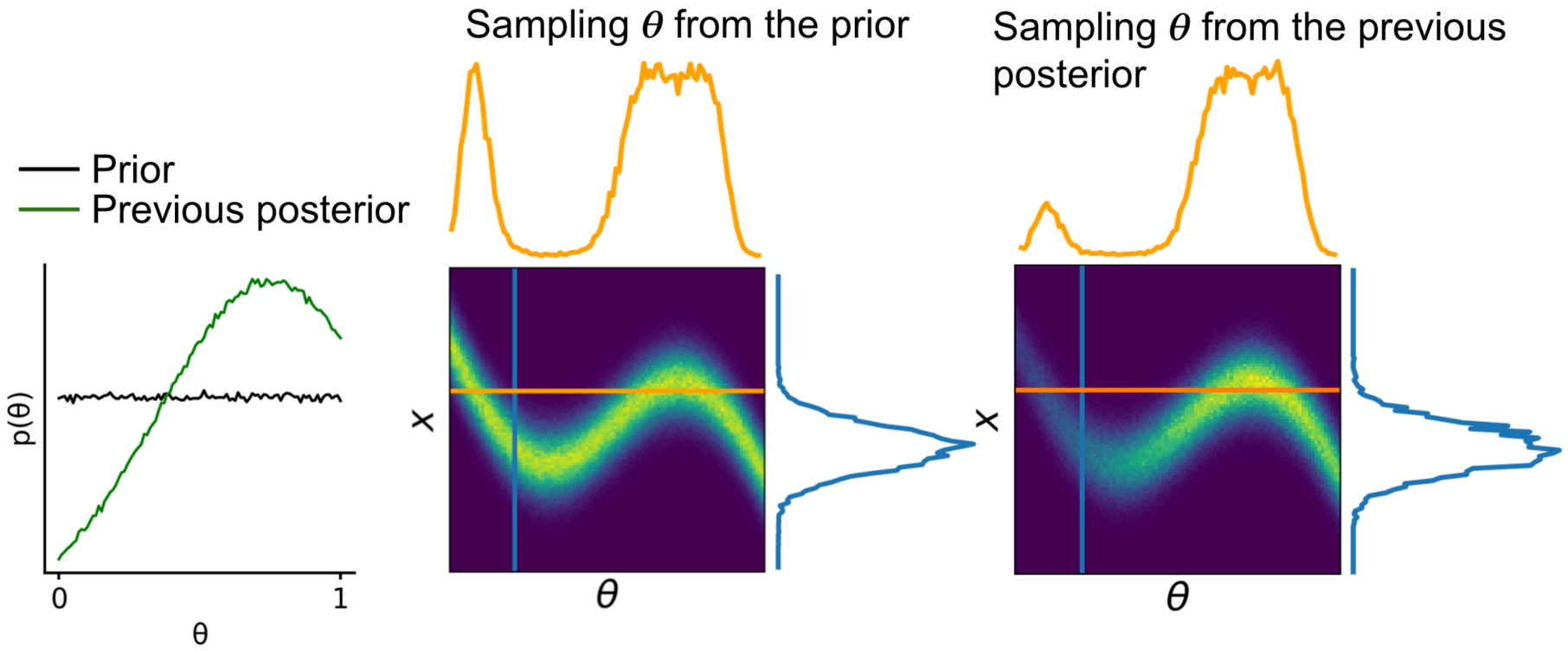
Single-round inference

- When drawing parameters from the prior and simulating them, we obtain (θ, x) from the joint distribution $p(\theta, x) = p(\theta)p(x \mid \theta)$.



Central problem of multi-round inference

- If our parameters are sampled from the previous posterior, we obtain (θ, x) from the joint distribution $\tilde{p}(\theta, x) = \tilde{p}(\theta)p(x \mid \theta)$.



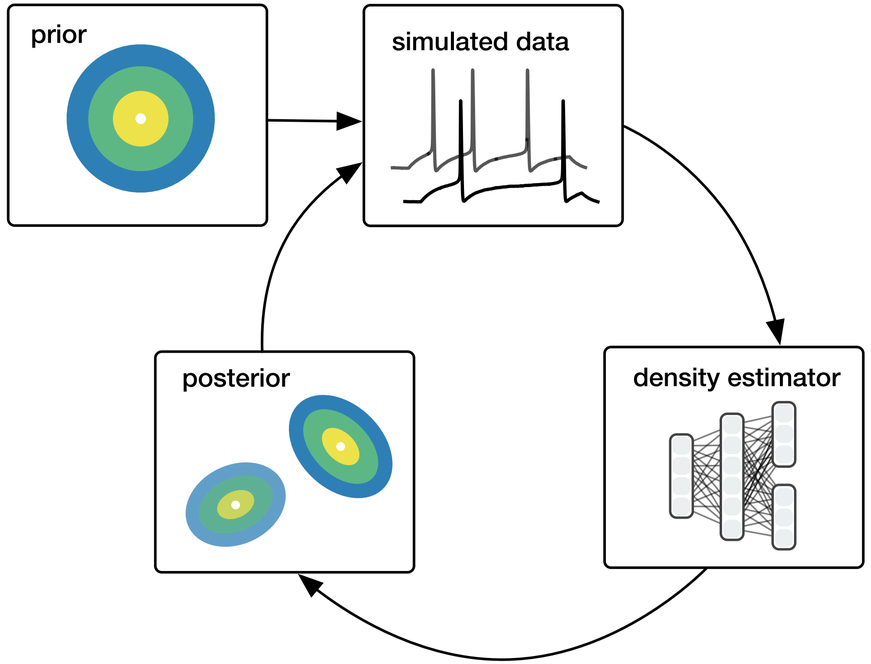
How do we correct for using a proposal distribution $\tilde{p}(\theta)$?

- NPE. When sampling from the prior, $\theta_n \sim p(\theta)$, we minimise negative log-likelihood to train the conditional density estimator: $-\sum_n \log q_\phi(\theta_n \,|\, x_n)$ to get $q_\phi(\theta \,|\, x_o) \approx p(\theta \,|\, x_o)$.

- What if $\theta_n \sim \tilde{p}(\theta)$ (proposal distribution/prior)?

What if $\theta_n \sim \tilde{p}(\theta)$?

- Some sequential algorithms:
  1. SNPE-A (Papamakarios & Murray NeurIPS 2016)
  2. SNPE-B (Lueckmann, Goncalves,..., Macke, NeurIPS 2017)
  3. SNPE-C (Greenberg, Nonnenmacher, Macke, ICML 2019) (NOT COVERED HERE)
- Many more not covered.



SNPE-A

- SNPE-A minimises the same loss function as in NPE, but applies a post-hoc analytical correction.
- If we minimise $-\sum_n \log q_\phi(\theta_n \mid x_n)$, where $\tilde{p}(\theta_n, x_n) = \tilde{p}(\theta_n) p(x_n \mid \theta_n)$, we get

$$q_\phi(\theta \mid x_o) \propto p(\theta \mid x_o) \frac{\tilde{p}(\theta)}{p(\theta)}$$

- The learned posterior $q_\phi(\theta \mid x_o)$ is adjusted, by analytically dividing it by $\tilde{p}(\theta)$ and multiplying it by $p(\theta)$.
- Potential issues:
  1. Can be unstable;
  2. Proposals need to be Gaussian.

SNPE-B

- SNPE-B minimises an importance-weighted loss function, directly approximating the posterior and therefore not requiring a post-hoc correction.
- When minimising loss function $-\sum_n \frac{p(\theta_n)}{\tilde{p}(\theta_n)} \log q_\phi(\theta_n \mid x_n)$, we get $q_\phi(\theta \mid x_o) \approx p(\theta \mid x_o)$
- Potential issues:
  1. Importance weights can lead to high variance in loss function.

Lecture 4: the universe of SBI

- 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 $p(\theta \mid x_o)$.
- Sequential SBI methods are developed for increased efficiency in the number of simulations.
- But, these methods do not allow amortised inference!!
- There are many methods (SNPE, SNLE, SNRE…) and variants, many that we did not cover in this course.
- Sequential methods tend to perform better than non-sequential ones.
- But still lots of challenges ahead in SBI (e.g., scalability).
- We built a benchmarking framework for SBI algorithms that can be used and extended by the community.

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