Robust and scalable simulation-based inference

François-Xavier Briol
Department of Statistical Science
University College London
https://fxbriol.github.io/
https://fsml-ucl.github.io/

Greek Stochastics - Folegandros 2025

Robust and scalable simulation-based inference















A (slightly biased) introduction to simulation-based inference

Intractable likelihoods

Our data: $y_1, \ldots, y_n \sim \mathbb{Q}$

Intractable likelihoods

Our data: $y_1, \ldots, y_n \sim \mathbb{Q}$

Unknown data-generating process defined on the data-space $\mathcal{X}$.

Intractable likelihoods

Our data: $y_1, \ldots, y_n \sim \mathbb{Q}$

Our model: $\{\mathbb{P}_\theta\}_{\theta \in \Theta}$

Intractable likelihoods

Our data: $y_1, \ldots, y_n \sim \mathbb{Q}$

Our model: $\{\mathbb{P}_\theta\}_{\theta \in \Theta}$

Our job is to recover θ^*

Intractable likelihoods

Our data: $y_1, \ldots, y_n \sim \mathbb{Q}$

Our model: $\{\mathbb{P}_\theta\}_{\theta \in \Theta}$

Maximum likelihood:

$$\hat{\theta}_n := \arg\max_{\theta \in \Theta} \prod_{i=1}^{n} p(y_i | \theta)$$

Bayesian inference:

$$p(\theta | y_1, \ldots, y_n) \propto \prod_{i=1}^{n} p(y_i | \theta) p(\theta)$$

Intractable likelihoods

Our data: $y_1, \ldots, y_n \sim \mathbb{Q}$

Our model: $\{\mathbb{P}_\theta\}_{\theta \in \Theta}$

Maximum likelihood:

$$\hat{\theta}_n := \arg\max_{\theta \in \Theta} \prod_{i=1}^{n} p(y_i | \theta)$$

Bayesian inference:

$$p(\theta | y_1, \ldots, y_n) \propto \prod_{i=1}^{n} p(y_i | \theta) p(\theta)$$

Simulation-based inference (SBI)

- A simulator $(\mathbb{U}, G_\theta)$ such that a draw from $\mathbb{P}_\theta$ can be obtained as:

$$x_i = G_\theta(u_i)$$

Simulation-based inference (SBI)

- A simulator $\boxed{(\mathbb{U}, G_\theta)}$ such that a draw from $\mathbb{P}_\theta$ can be obtained as:

$$x_i = G_\theta(u_i)$$

Simulation-based inference (SBI)

- A simulator $(\mathbb{U}, G_\theta)$ such that a draw from $\mathbb{P}_\theta$ can be obtained as:

$$x_i = G_\theta(u_i)$$

$u_i \sim \mathbb{U}$
(randomness)

Simulation-based inference (SBI)

- A simulator $(\mathbb{U}, G_\theta)$ such that a draw from $\mathbb{P}_\theta$ can be obtained as:

$$x_i = G_\theta(u_i)$$

Deterministic map (generator)

$u_i \sim \mathbb{U}$ (randomness)

Simulation-based inference (SBI)

- A simulator $(\mathbb{U}, G_\theta)$ such that a draw from $\mathbb{P}_\theta$ can be obtained as:

Simulated data

$$x_i = G_\theta(u_i)$$

Deterministic map (generator)

$u_i \sim \mathbb{U}$ (randomness)

Simulation-based inference (SBI)

- A simulator $(\mathbb{U}, G_\theta)$ such that a draw from $\mathbb{P}_\theta$ can be obtained as:

Simulated data

$$x_i = G_\theta(u_i)$$

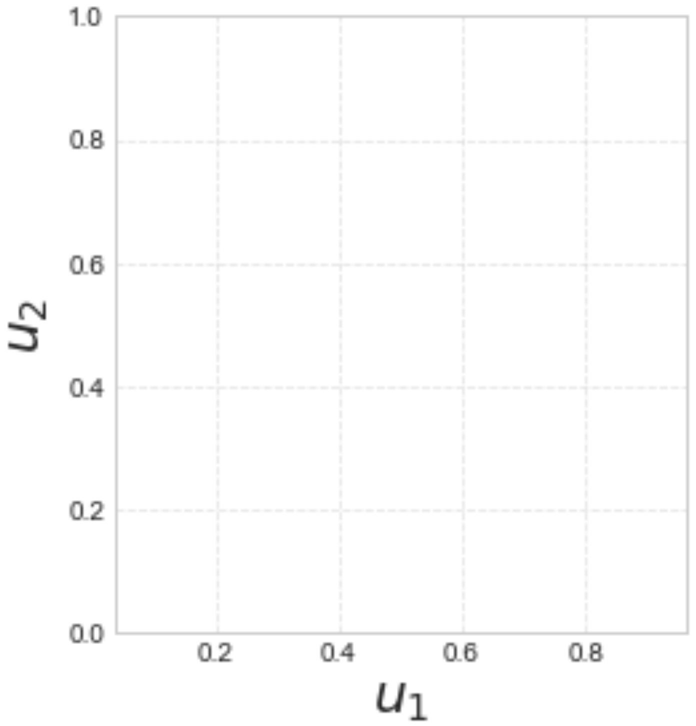
Deterministic map (generator)

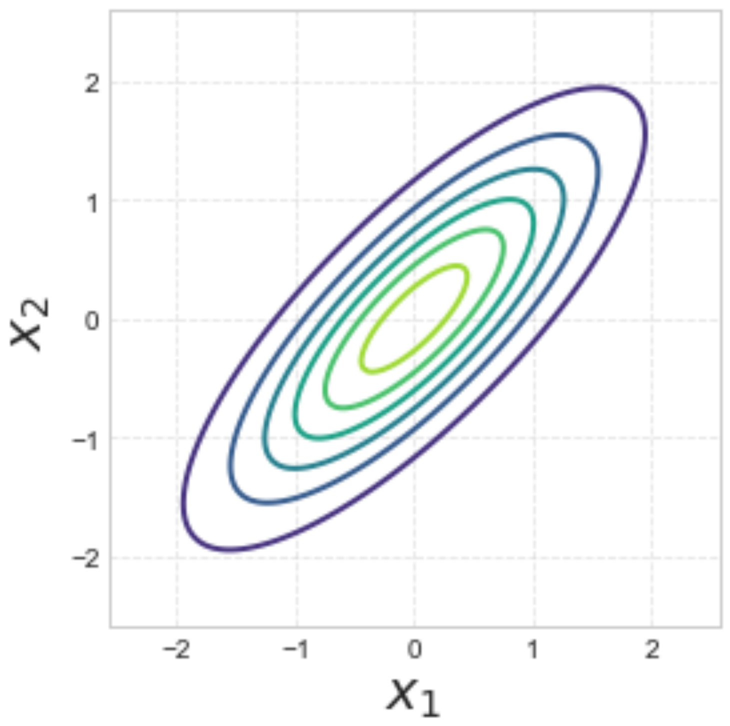
$u_i \sim \mathbb{U}$ (randomness)

Simulation-based inference: Inference using simulated data to replace evaluations of the likelihood!

A trivial simulator for Gaussians

- $\mathbb{P}_\theta := \mathcal{N}(\mu, \Sigma),\quad u_i = (u_{i1}, u_{i2})^\top,\quad u_{i1}, u_{i2} \sim \text{Unif}(0,1)\quad G_\theta(u) = \mu + L\begin{pmatrix}\Phi^{-1}(u_{i,1})\\ \Phi^{-1}(u_{i2})\end{pmatrix}$



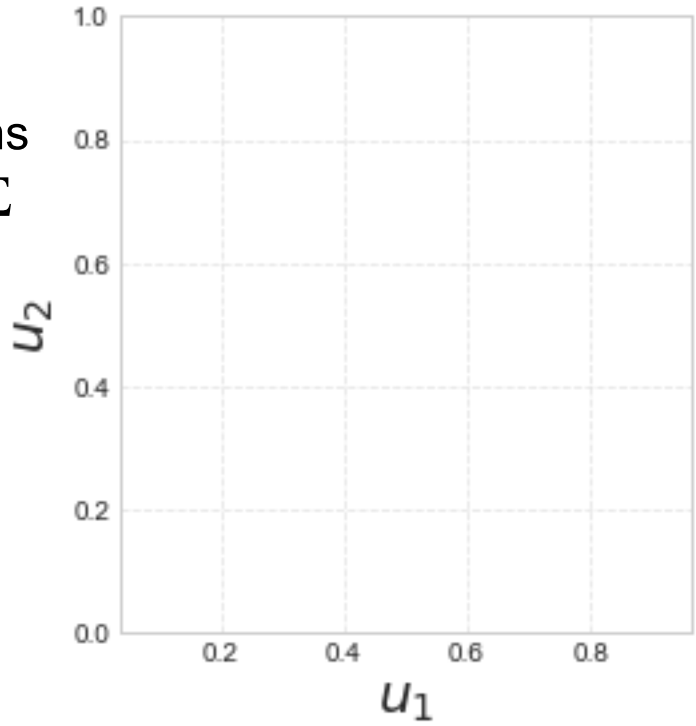


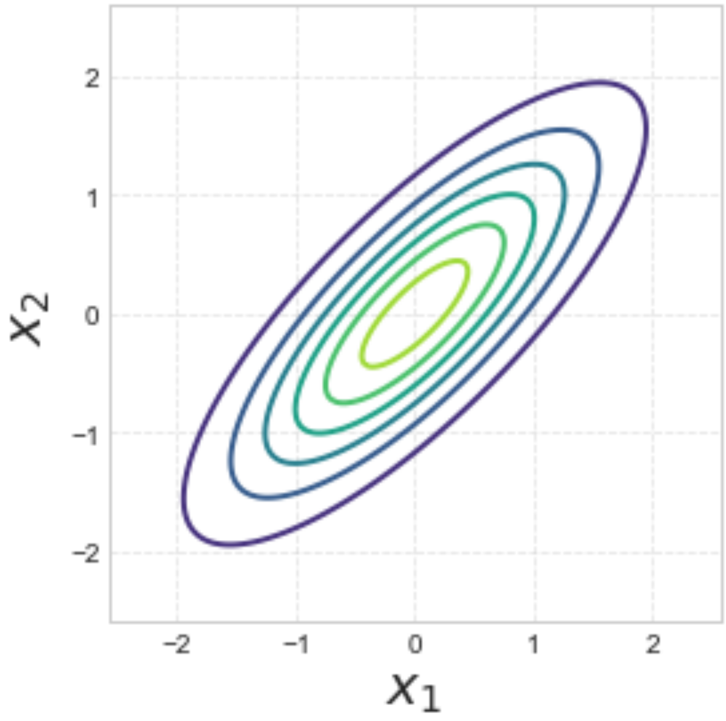
$\Sigma = LL^\top$

A trivial simulator for Gaussians

- $\mathbb{P}_\theta := \mathcal{N}(\mu, \Sigma)$, $\quad u_i = (u_{i1}, u_{i2})^\top$, $\quad u_{i1}, u_{i2} \sim \text{Unif}(0,1)$ $\quad G_\theta(u) = \mu + L \begin{pmatrix} \Phi^{-1}(u_{i,1}) \\ \Phi^{-1}(u_{i2}) \end{pmatrix}$

i.e. θ contains both μ and Σ

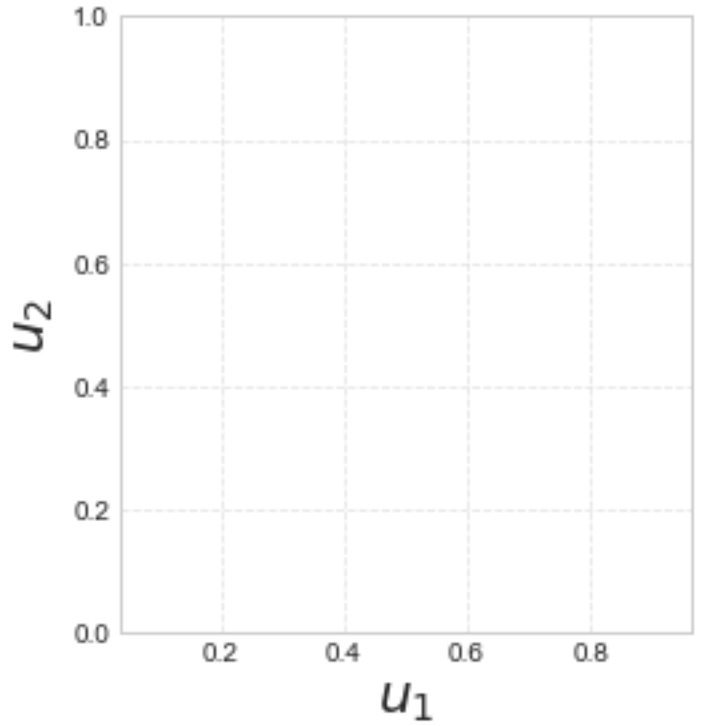


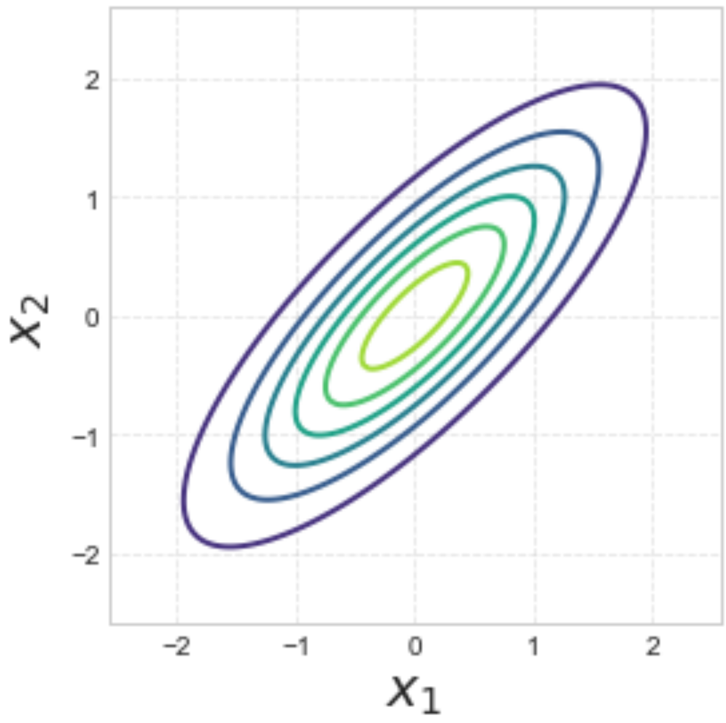


$\Sigma = LL^\top$

A trivial simulator for Gaussians

- $\mathbb{P}_\theta := \mathcal{N}(\mu, \Sigma), \quad u_i = (u_{i1}, u_{i2})^\top, \quad u_{i1}, u_{i2} \sim \text{Unif}(0,1) \quad G_\theta(u) = \mu + L \begin{pmatrix} \Phi^{-1}(u_{i,1}) \\ \Phi^{-1}(u_{i2}) \end{pmatrix}$



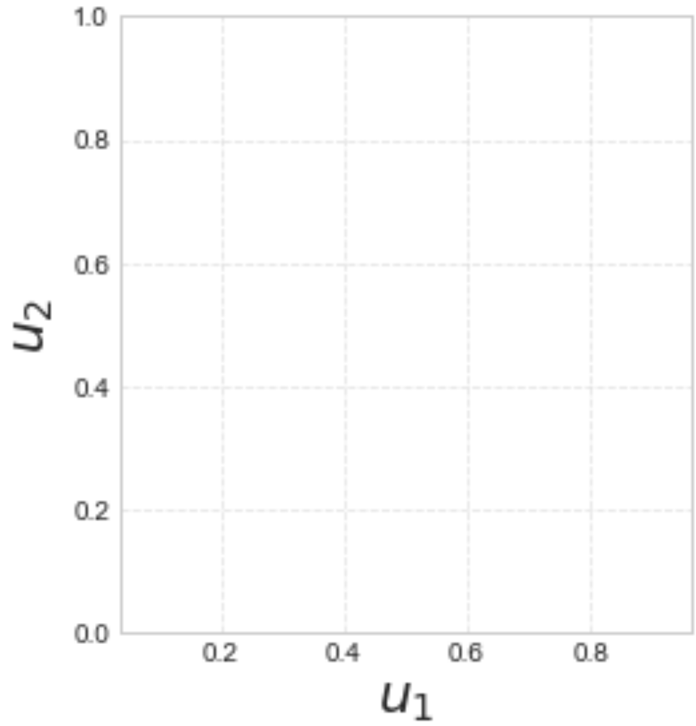


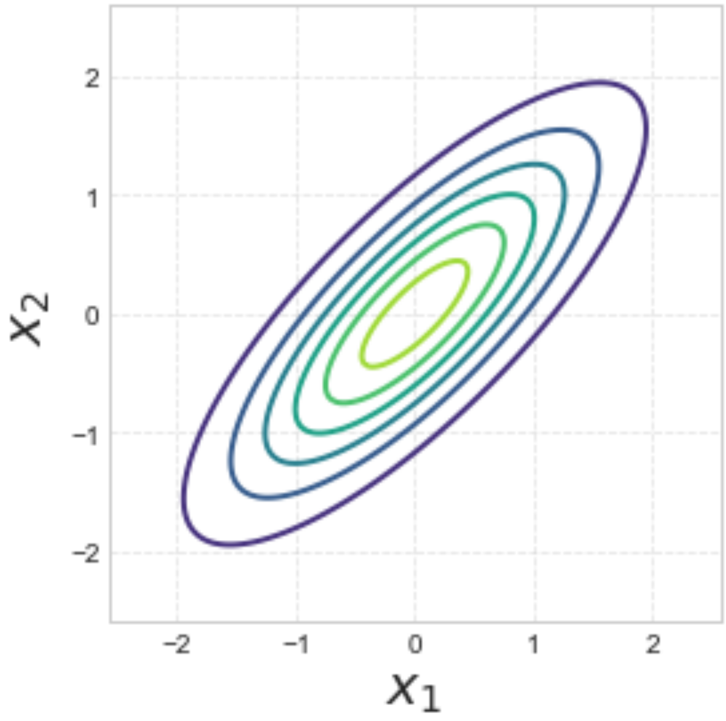
$\Sigma = LL^\top$

Inverse CDF of standard Gaussian!

A trivial simulator for Gaussians

- $\mathbb{P}_\theta := \mathcal{N}(\mu, \Sigma)$, $u_i = (u_{i1}, u_{i2})^\top$, $u_{i1}, u_{i2} \sim \text{Unif}(0,1)$ $G_\theta(u) = \mu + L \begin{pmatrix} \Phi^{-1}(u_{i,1}) \\ \Phi^{-1}(u_{i2}) \end{pmatrix}$



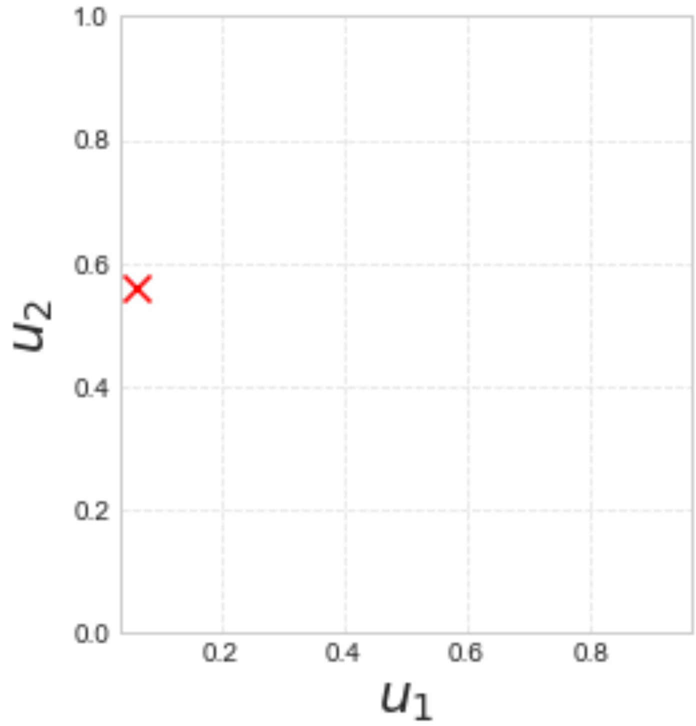


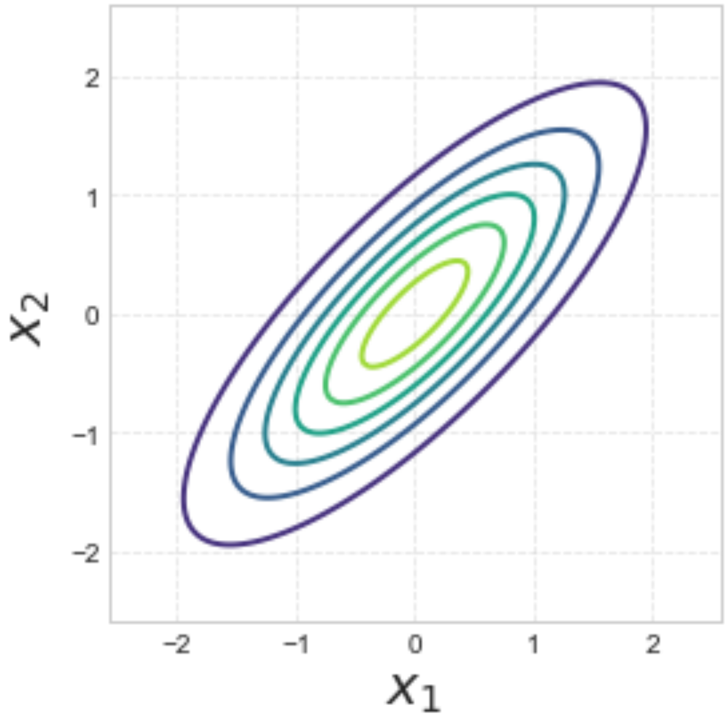
$\Sigma = LL^\top$

Cholesky!

A trivial simulator for Gaussians

- $\mathbb{P}_\theta := \mathcal{N}(\mu, \Sigma)$, $\quad u_i = (u_{i1}, u_{i2})^\top$, $\quad u_{i1}, u_{i2} \sim \text{Unif}(0,1)$ $\quad G_\theta(u) = \mu + L \begin{pmatrix} \Phi^{-1}(u_{i,1}) \\ \Phi^{-1}(u_{i2}) \end{pmatrix}$

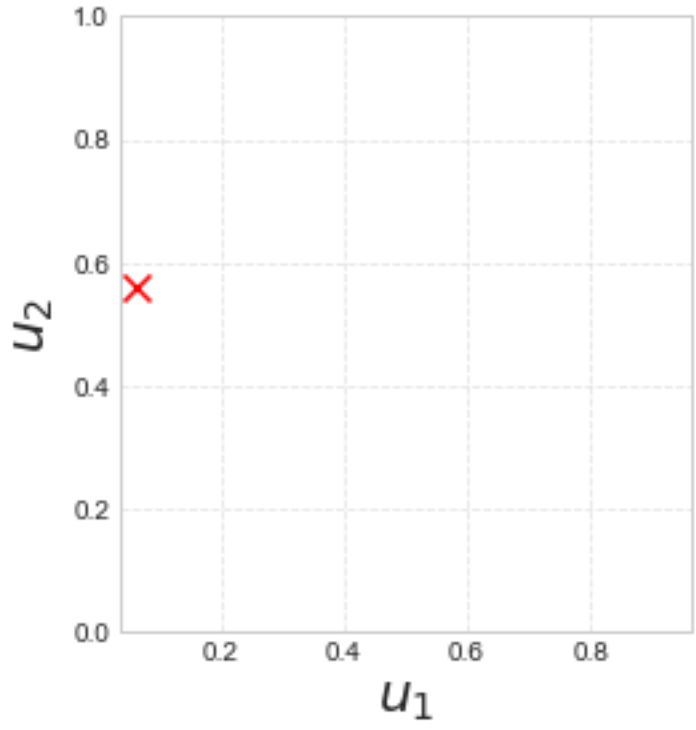


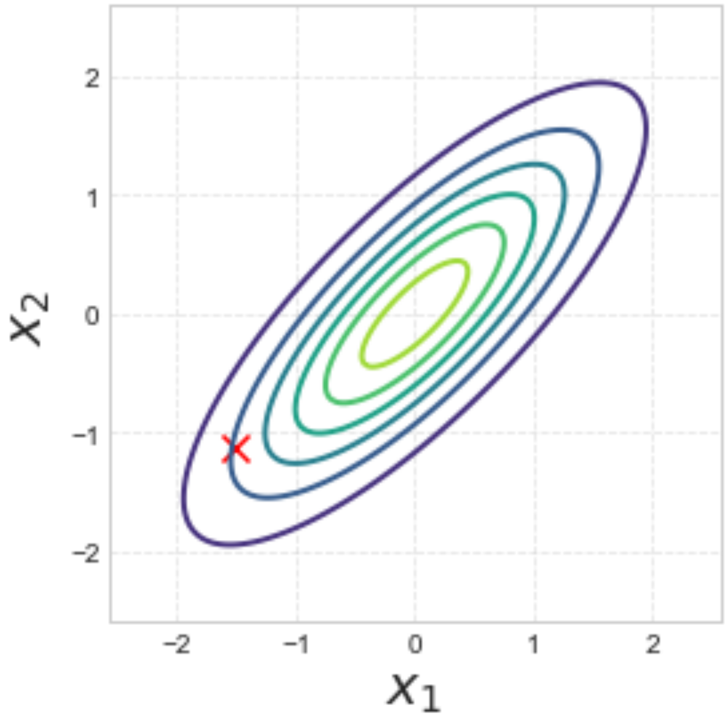


$\Sigma = LL^\top$

A trivial simulator for Gaussians

- $\mathbb{P}_\theta := \mathcal{N}(\mu, \Sigma)$, $\quad u_i = (u_{i1}, u_{i2})^\top$, $\quad u_{i1}, u_{i2} \sim \text{Unif}(0,1)$ $\quad G_\theta(u) = \mu + L \begin{pmatrix} \Phi^{-1}(u_{i,1}) \\ \Phi^{-1}(u_{i2}) \end{pmatrix}$

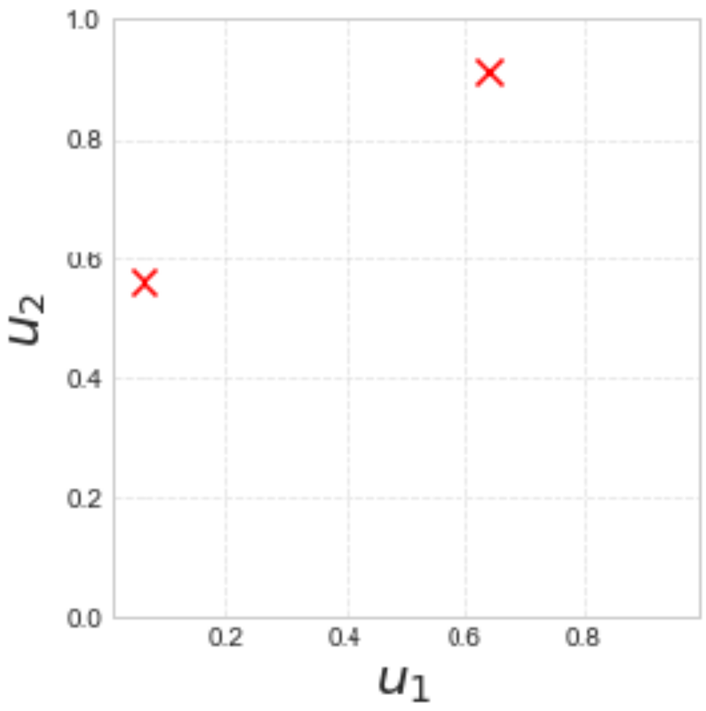


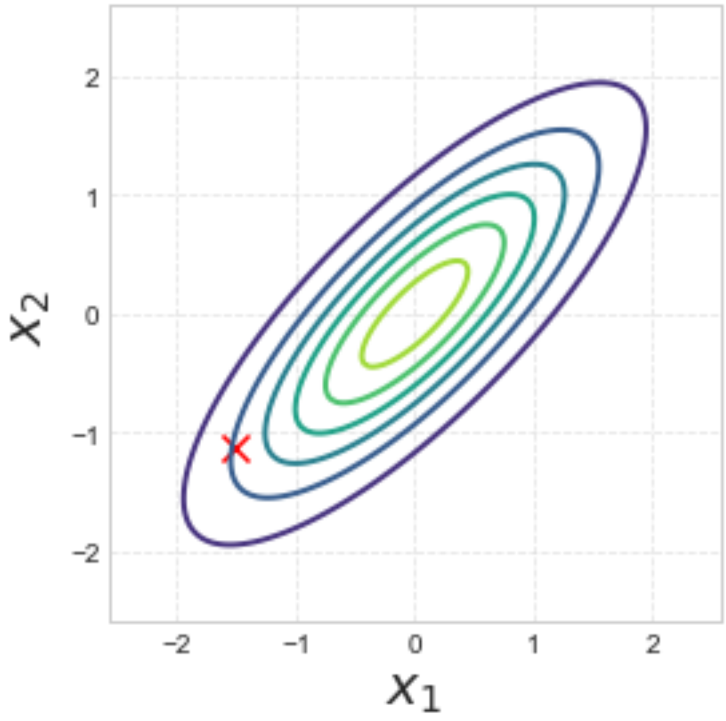


$$\Sigma = LL^\top$$

A trivial simulator for Gaussians

- $\mathbb{P}_\theta := \mathcal{N}(\mu, \Sigma)$, $\quad u_i = (u_{i1}, u_{i2})^\top$, $\quad u_{i1}, u_{i2} \sim \text{Unif}(0,1)$ $\quad G_\theta(u) = \mu + L \begin{pmatrix} \Phi^{-1}(u_{i,1}) \\ \Phi^{-1}(u_{i2}) \end{pmatrix}$

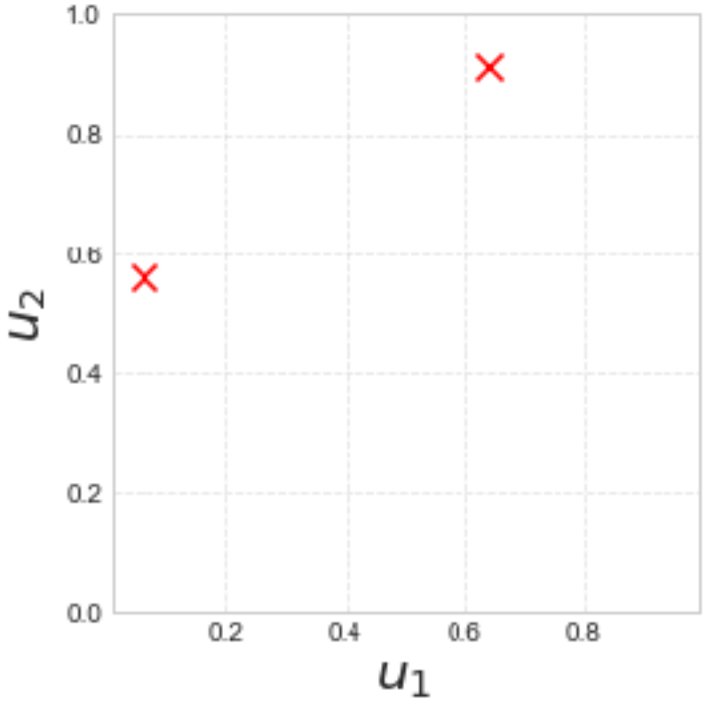


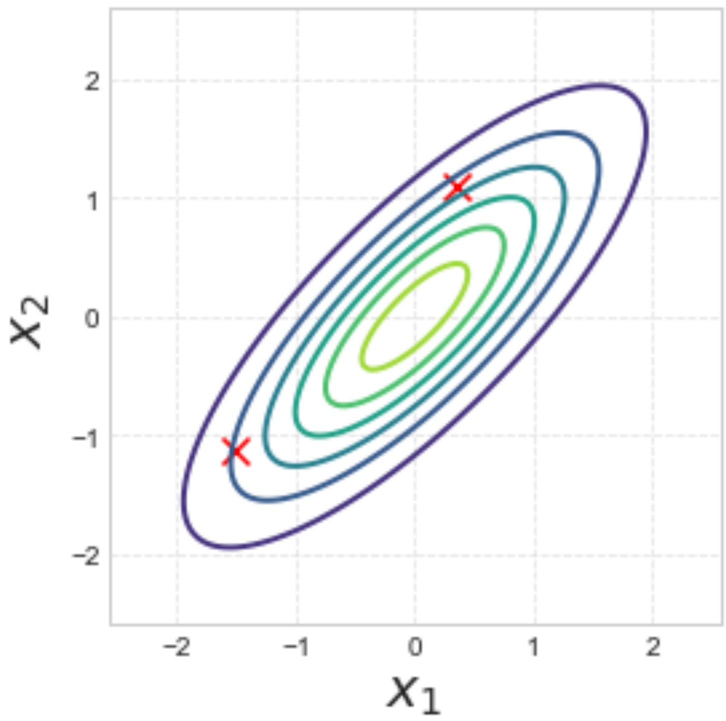


$$\Sigma = LL^\top$$

A trivial simulator for Gaussians

- $\mathbb{P}_\theta := \mathcal{N}(\mu, \Sigma)$, $\quad u_i = (u_{i1}, u_{i2})^\top$, $\quad u_{i1}, u_{i2} \sim \text{Unif}(0,1)$ $\quad G_\theta(u) = \mu + L \begin{pmatrix} \Phi^{-1}(u_{i,1}) \\ \Phi^{-1}(u_{i2}) \end{pmatrix}$

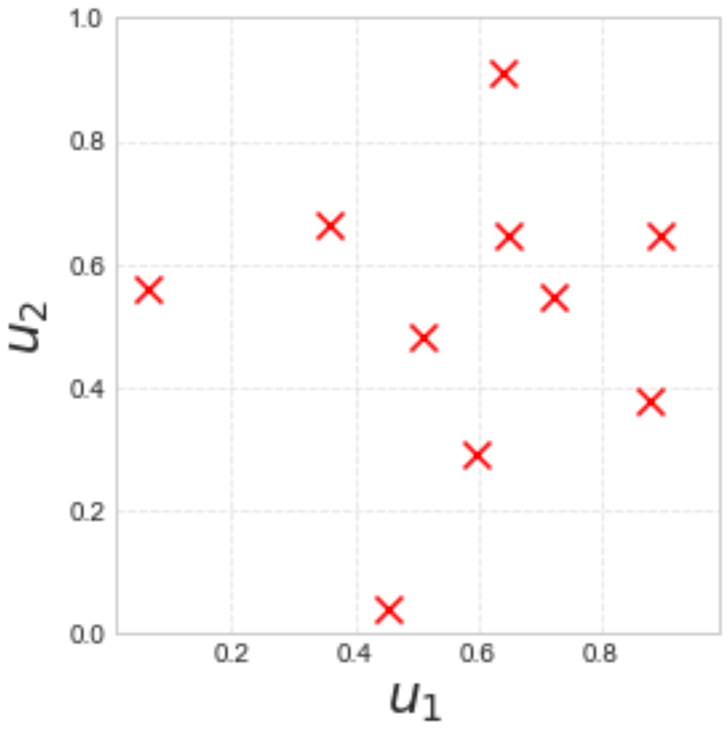


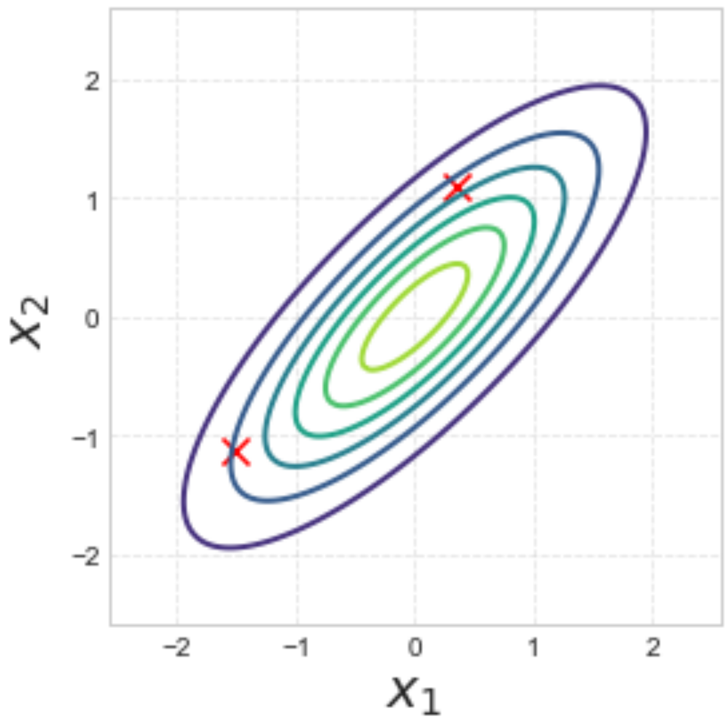


$\Sigma = LL^\top$

A trivial simulator for Gaussians

- $\mathbb{P}_\theta := \mathcal{N}(\mu, \Sigma)$, $\quad u_i = (u_{i1}, u_{i2})^\top$, $\quad u_{i1}, u_{i2} \sim \text{Unif}(0,1)$ $\quad G_\theta(u) = \mu + L \begin{pmatrix} \Phi^{-1}(u_{i,1}) \\ \Phi^{-1}(u_{i2}) \end{pmatrix}$

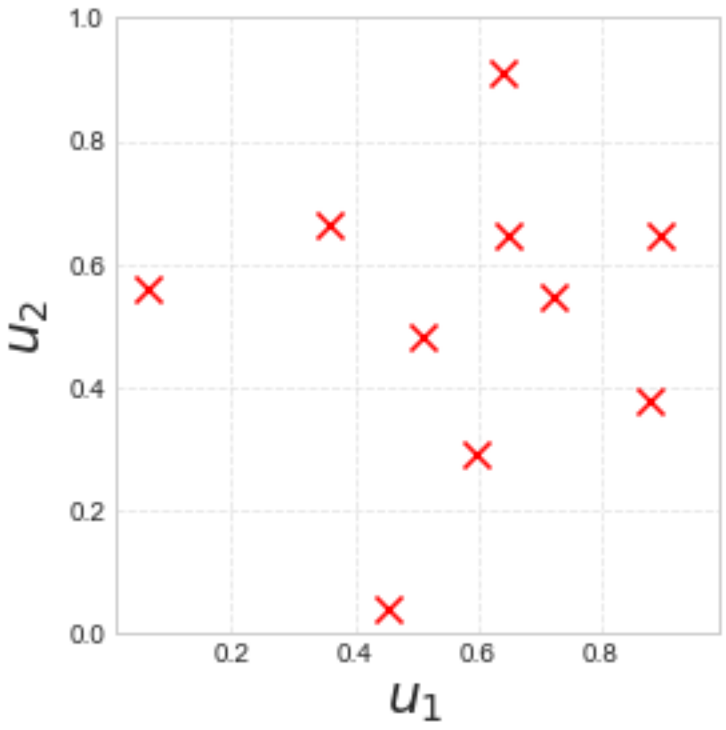


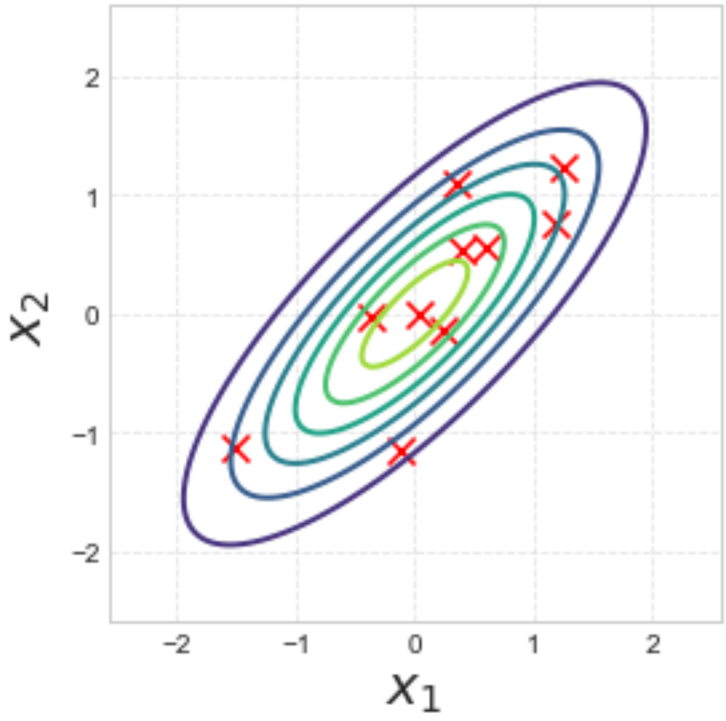


$$\Sigma = LL^\top$$

A trivial simulator for Gaussians

- $\mathbb{P}_\theta := \mathcal{N}(\mu, \Sigma)$, $\quad u_i = (u_{i1}, u_{i2})^\top$, $\quad u_{i1}, u_{i2} \sim \text{Unif}(0,1)$ $\quad G_\theta(u) = \mu + L \begin{pmatrix} \Phi^{-1}(u_{i,1}) \\ \Phi^{-1}(u_{i2}) \end{pmatrix}$

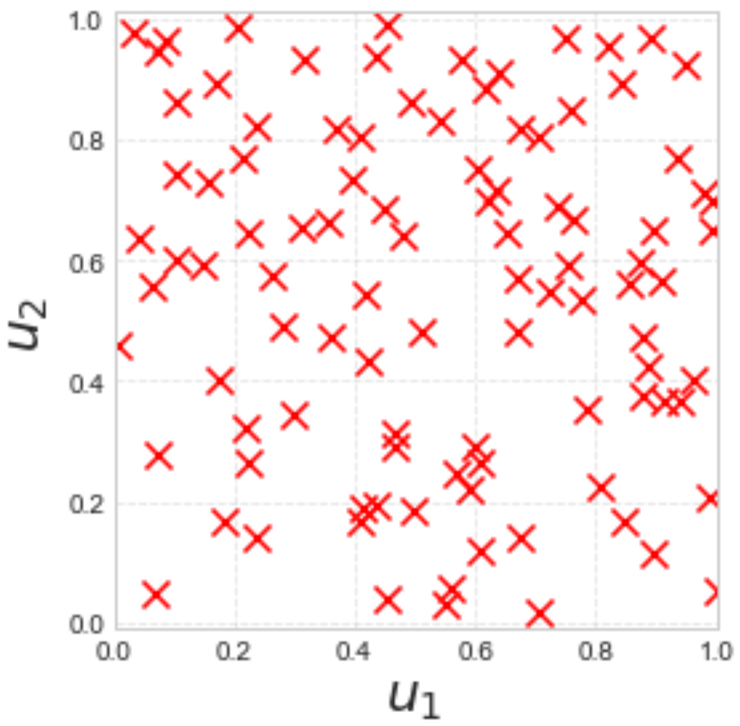


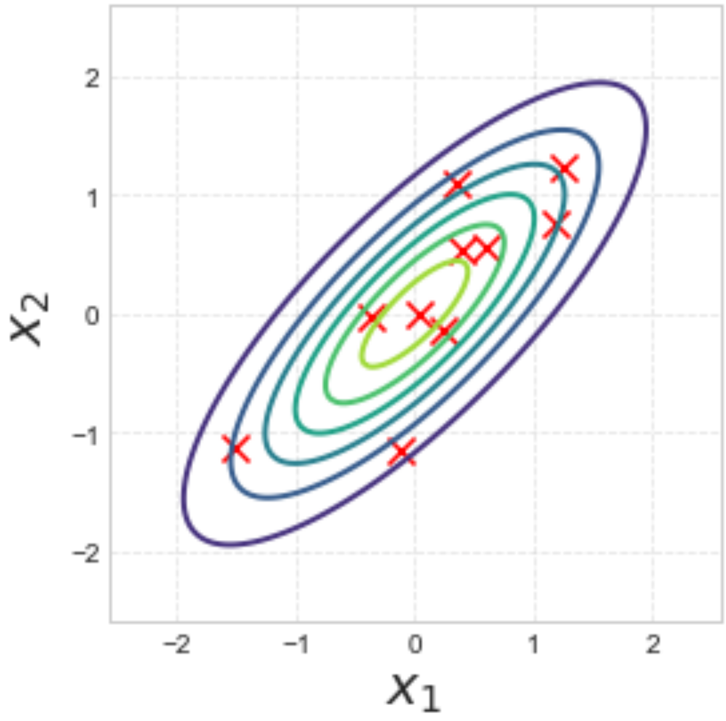


$\Sigma = LL^\top$

A trivial simulator for Gaussians

- $\mathbb{P}_\theta := \mathcal{N}(\mu, \Sigma)$, $\quad u_i = (u_{i1}, u_{i2})^\top$, $\quad u_{i1}, u_{i2} \sim \text{Unif}(0,1)$ $\quad G_\theta(u) = \mu + L \begin{pmatrix} \Phi^{-1}(u_{i,1}) \\ \Phi^{-1}(u_{i2}) \end{pmatrix}$

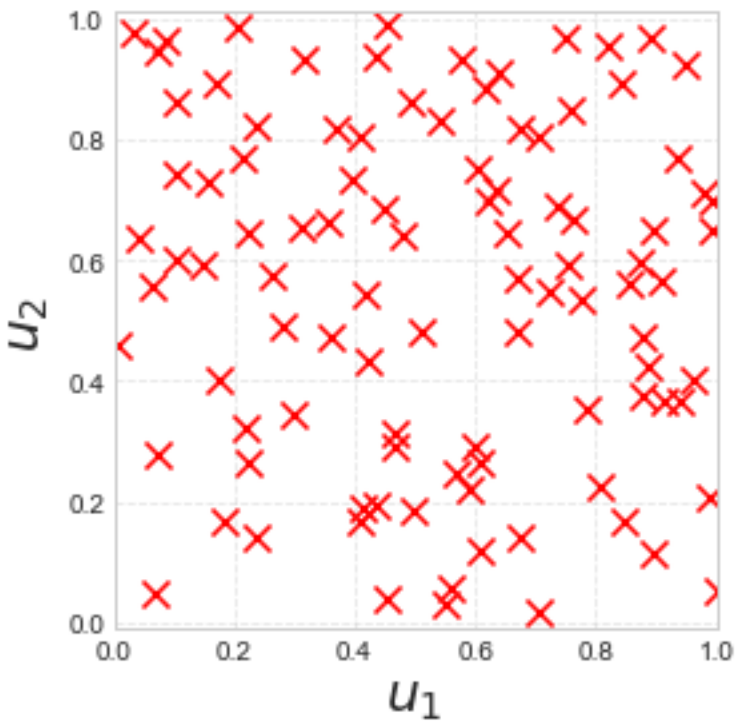


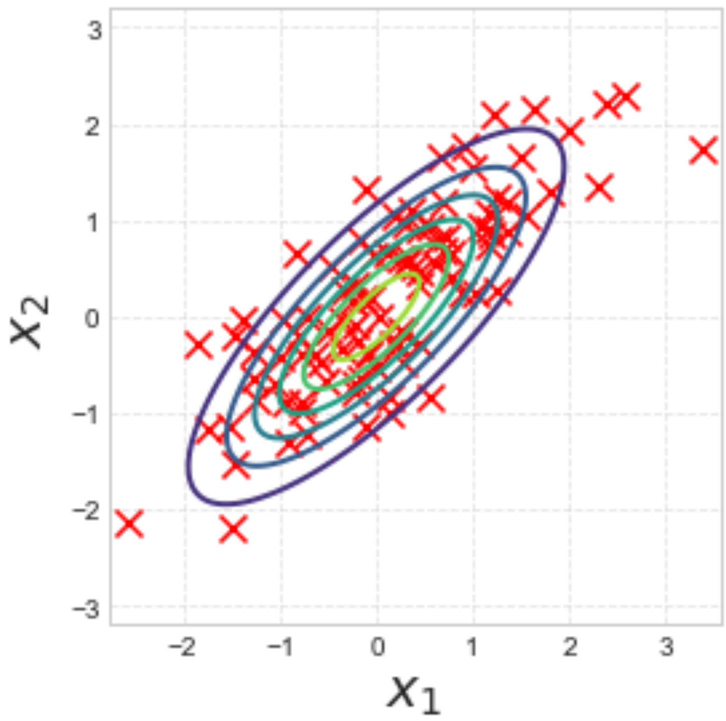


$\Sigma = LL^\top$

A trivial simulator for Gaussians

- $\mathbb{P}_\theta := \mathcal{N}(\mu, \Sigma)$, $u_i = (u_{i1}, u_{i2})^\top$, $u_{i1}, u_{i2} \sim \text{Unif}(0,1)$ $G_\theta(u) = \mu + L \begin{pmatrix} \Phi^{-1}(u_{i,1}) \\ \Phi^{-1}(u_{i2}) \end{pmatrix}$

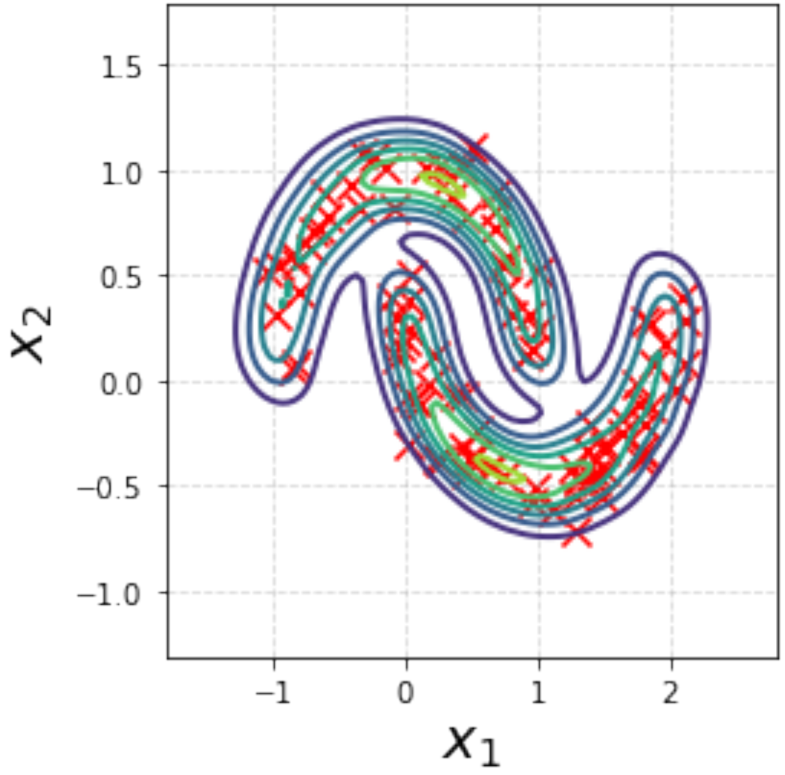




$\Sigma = LL^\top$

Some slightly less trivial simulators….

- We can create all sorts of more complex simulators by increasing the complexity of the G_θ map.

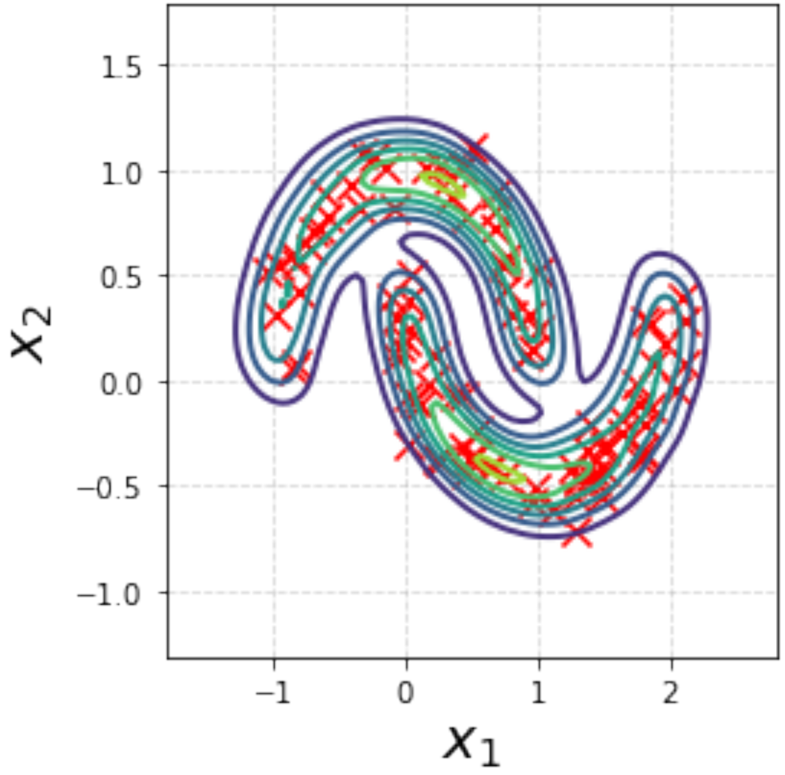


Some slightly less trivial simulators….

- We can create all sorts of more complex simulators by increasing the complexity of the G_θ map.

- Lots of classical tools from the Monte Carlo community can be used for this:

  Devroye, L. (1986). *Non-Uniform Random Variate Generation*. Springer-Verlag.



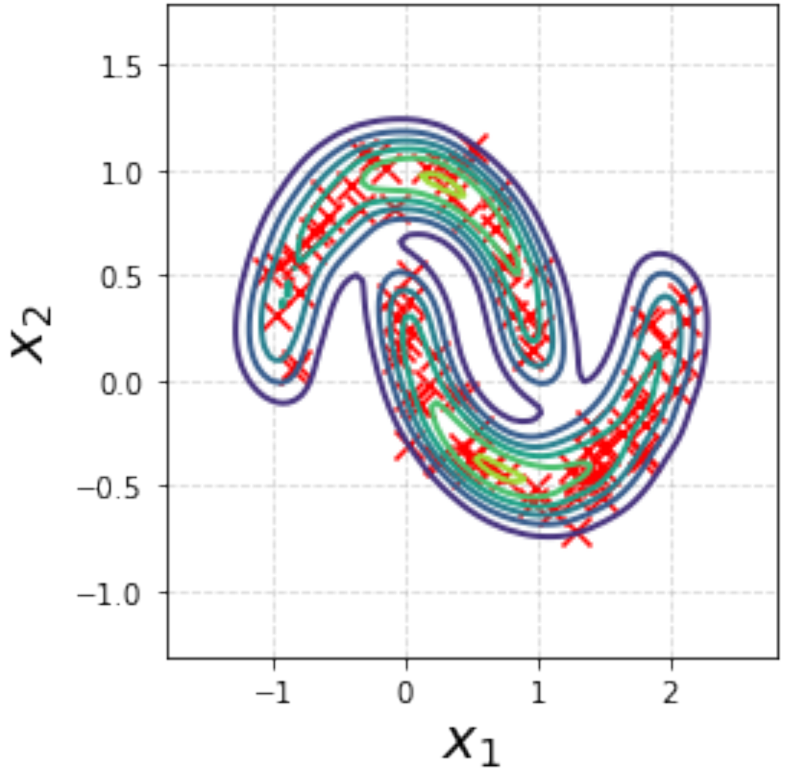
Some slightly less trivial simulators….

- We can create all sorts of more complex simulators by increasing the complexity of the G_θ map.

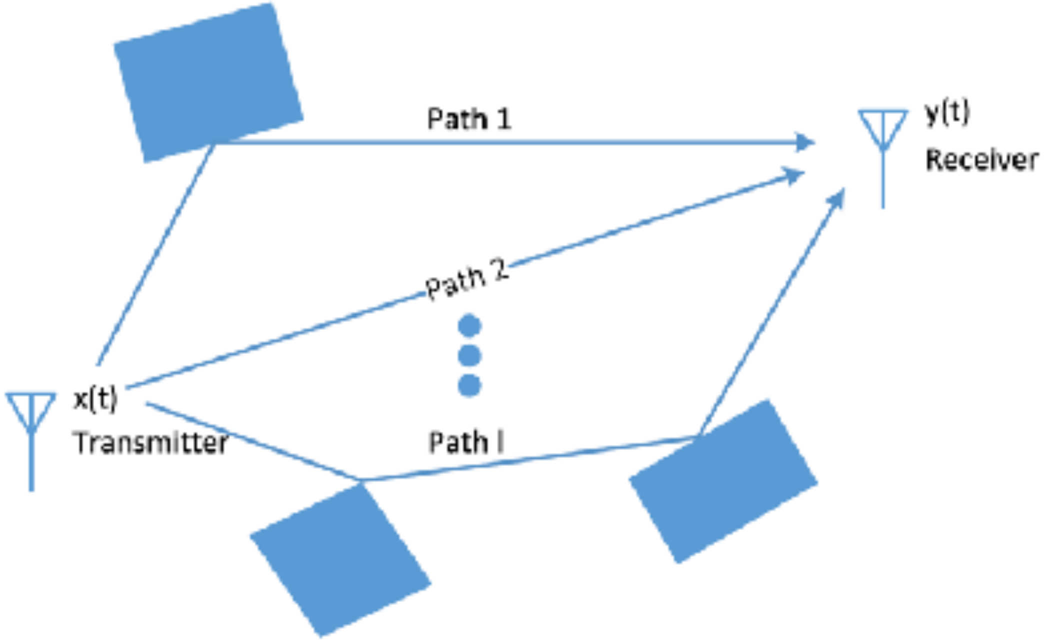
- Lots of classical tools from the Monte Carlo community can be used for this:

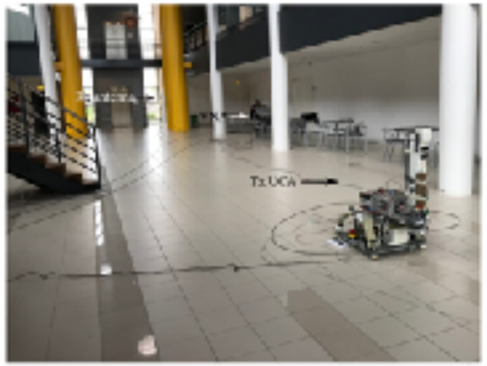
  Devroye, L. (1986). *Non-Uniform Random Variate Generation*. Springer-Verlag.

- SBI often works with simulators **carefully crafted by scientists and engineers**. These simulators are hence implementations of complex mathematical models of the phenomena being studied.



Simulators in telecommunications engineering

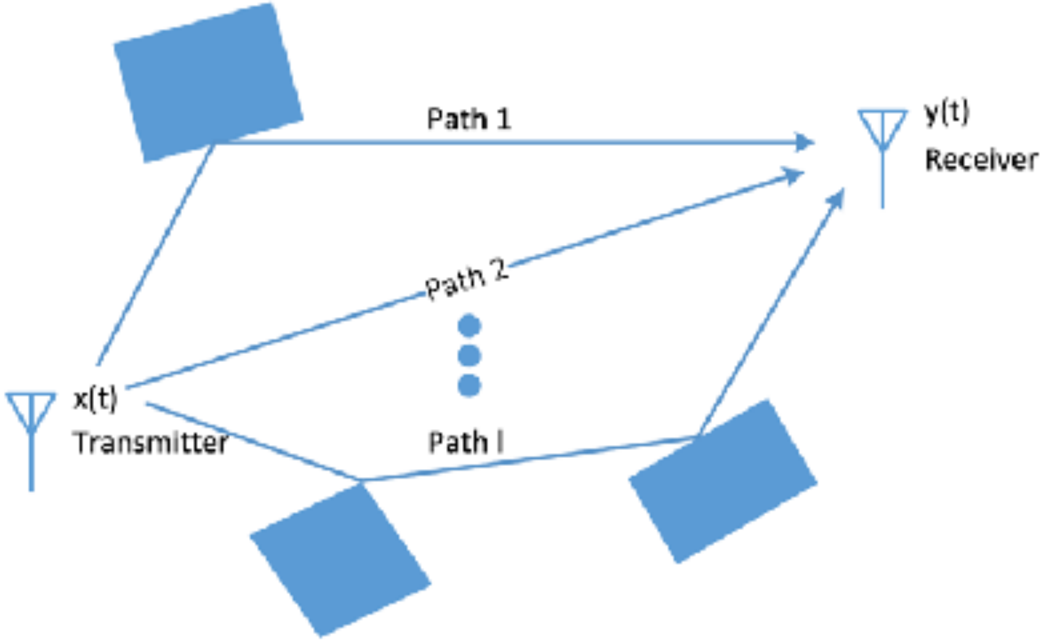


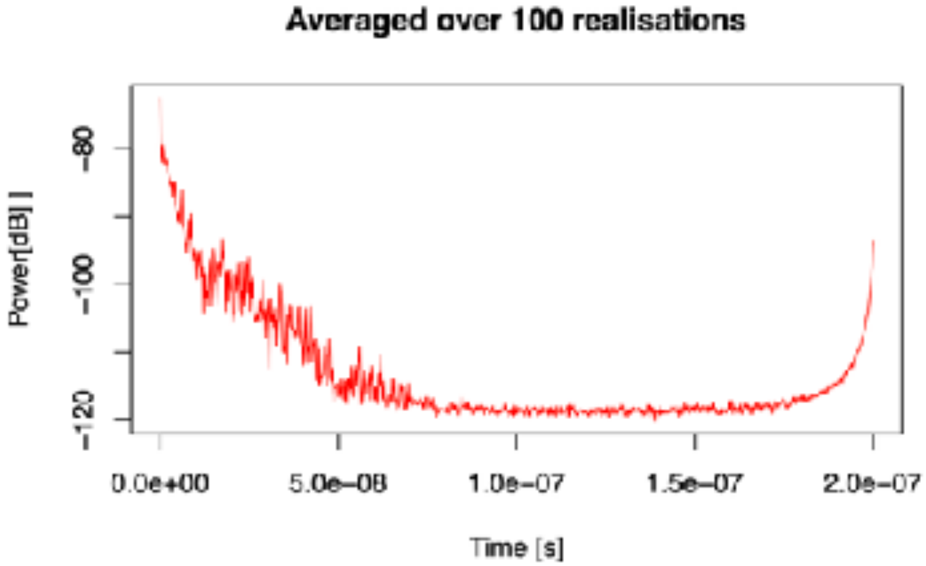


Briol, F-X., Bharti, A. (2021). Using machine learning to improve the reliability of wireless communication systems. https://www.turing.ac.uk/blog/using-machine-learning-improve-reliability-wireless-communication-systems

Bharti, A., **Briol, F-X.**, Pedersen, T. (2022). A general method for calibrating stochastic radio channel models with kernels. IEEE Transactions on Antennas and Propagation, vol. 70, no. 6, pp. 3986-4001, June 2022.

Simulators in telecommunications engineering



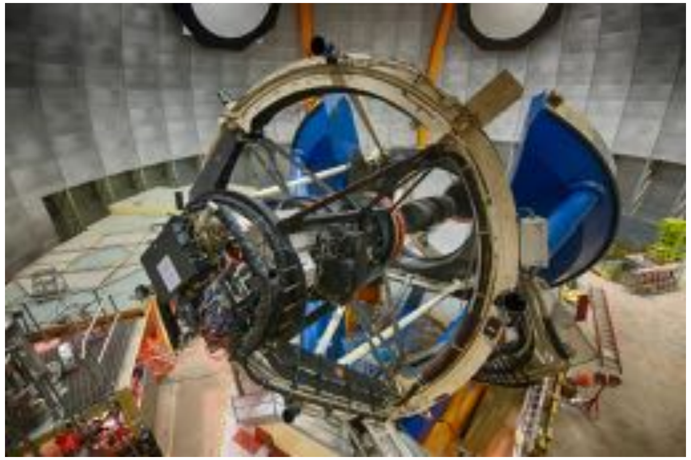


Briol, F-X., Bharti, A. (2021). Using machine learning to improve the reliability of wireless communication systems. https://www.turing.ac.uk/blog/using-machine-learning-improve-reliability-wireless-communication-systems

Bharti, A., **Briol, F-X.**, Pedersen, T. (2022). A general method for calibrating stochastic radio channel models with kernels. IEEE Transactions on Antennas and Propagation, vol. 70, no. 6, pp. 3986-4001, June 2022.

Simulators in cosmology

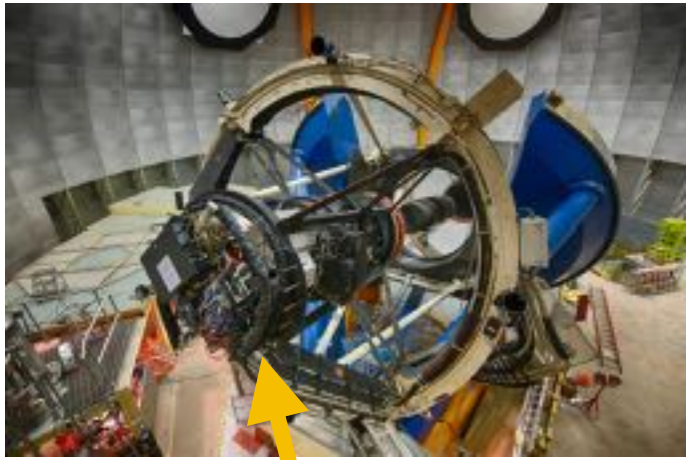
(+ ≈ 400 scientists from 25 institutions in 7 countries)



Jeffrey, N., et al. (2025). Dark energy survey year 3 results: likelihood-free, simulation-based wCDM inference with neural compression of weak-lensing map statistics. *Monthly Notices of the Royal Astronomical Society*, *536*(2), 1303–1322.

Simulators in cosmology

(+ $\approx$ 400 scientists from 25 institutions in 7 countries)

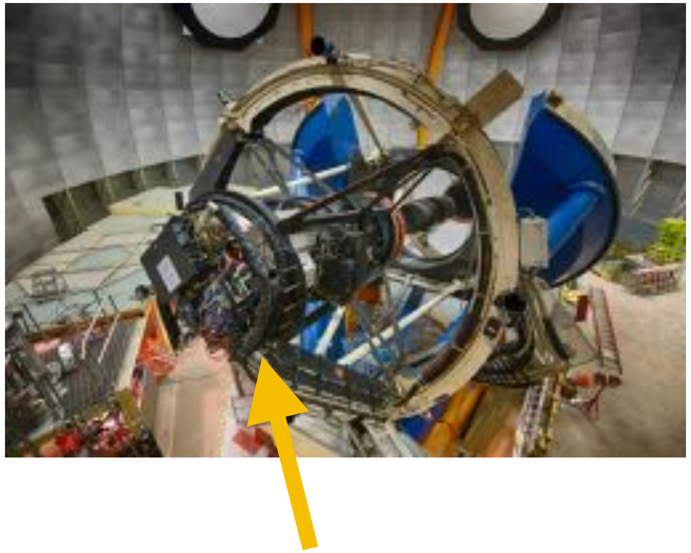


The Dark energy survey camera!

Jeffrey, N., et al. (2025). Dark energy survey year 3 results: likelihood-free, simulation-based wCDM inference with neural compression of weak-lensing map statistics. *Monthly Notices of the Royal Astronomical Society*, *536*(2), 1303–1322.

Simulators in cosmology

(+ $\approx$ 400 scientists from 25 institutions in 7 countries)

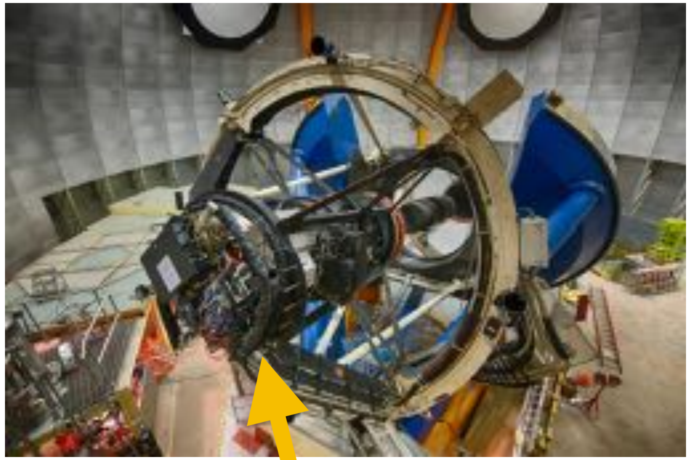


The Dark energy survey camera!

Jeffrey, N., et al. (2025). Dark energy survey year 3 results: likelihood-free, simulation-based wCDM inference with neural compression of weak-lensing map statistics. *Monthly Notices of the Royal Astronomical Society*, *536*(2), 1303–1322.

Simulators in cosmology

(+ $\approx$ 400 scientists from 25 institutions in 7 countries)



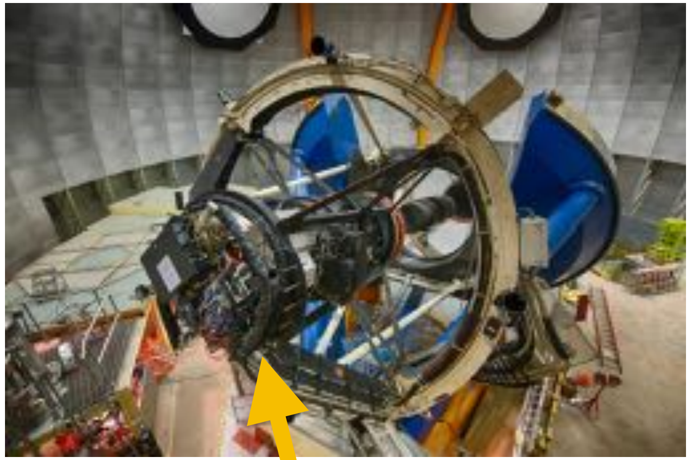
The Dark energy survey camera!

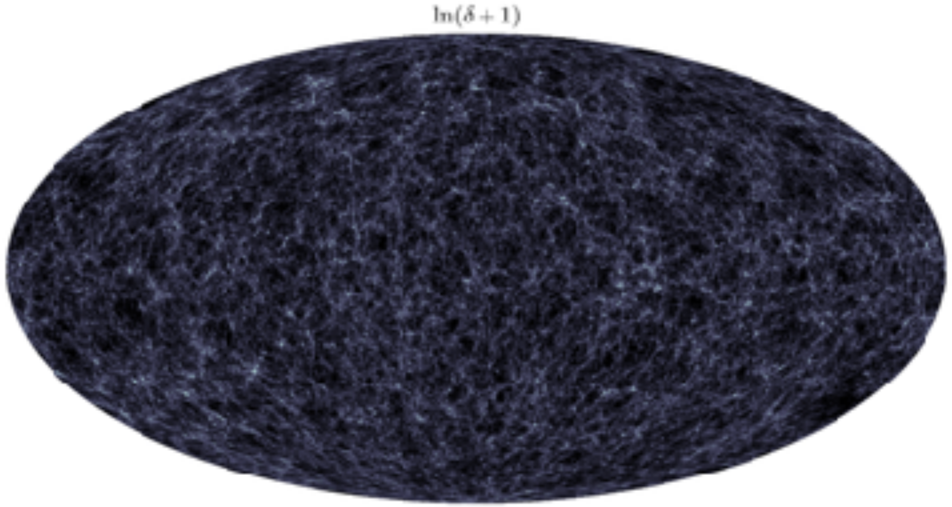
Data collected through the Dark energy survey camera

Jeffrey, N., et al. (2025). Dark energy survey year 3 results: likelihood-free, simulation-based wCDM inference with neural compression of weak-lensing map statistics. *Monthly Notices of the Royal Astronomical Society*, *536*(2), 1303–1322.

Simulators in cosmology

(+ $\approx$ 400 scientists from 25 institutions in 7 countries)





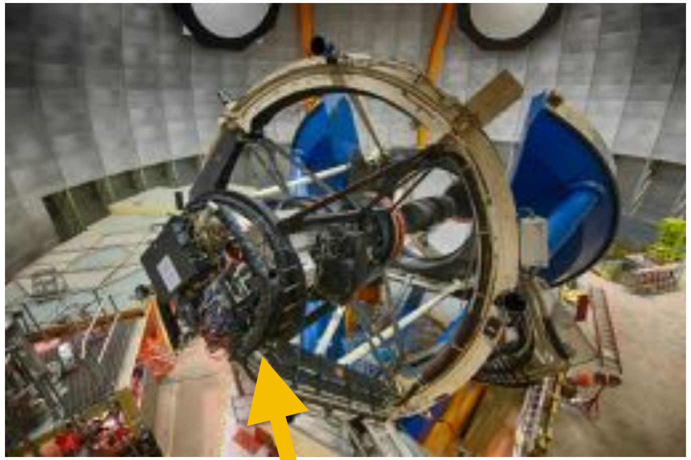
The Dark energy survey camera!

Data collected through the Dark energy survey camera

Jeffrey, N., et al. (2025). Dark energy survey year 3 results: likelihood-free, simulation-based wCDM inference with neural compression of weak-lensing map statistics. *Monthly Notices of the Royal Astronomical Society*, *536*(2), 1303–1322.

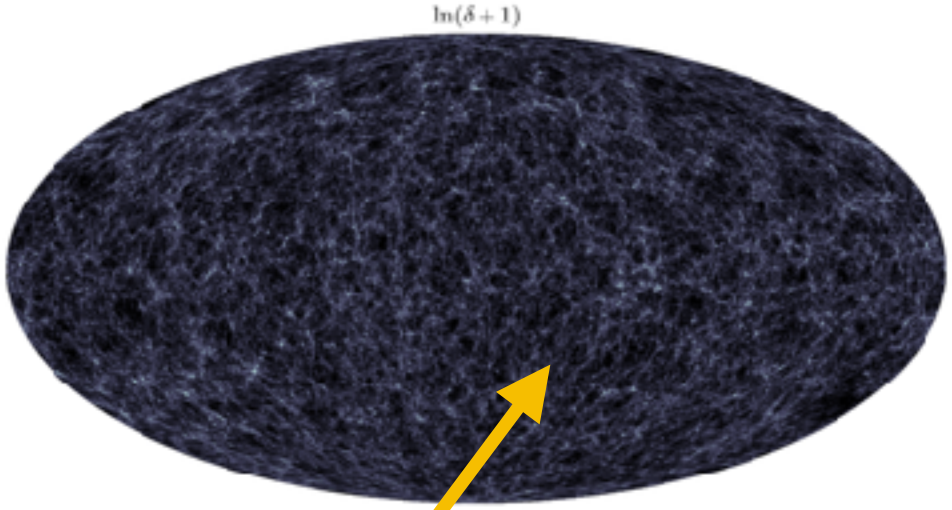
Simulators in cosmology

(+ ≈ 400 scientists from 25 institutions in 7 countries)



The Dark energy survey camera!

Data collected through the Dark energy survey camera



'Gower Street simulation' run by Niall and colleagues at UCL Physics

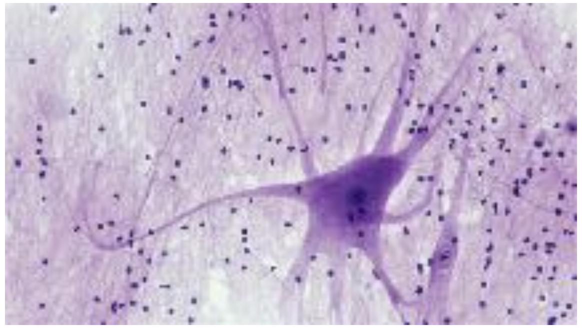
Jeffrey, N., et al. (2025). Dark energy survey year 3 results: likelihood-free, simulation-based wCDM inference with neural compression of weak-lensing map statistics. *Monthly Notices of the Royal Astronomical Society*, *536*(2), 1303–1322.

Simulators in the sciences and beyond

Particle Physics (CERN)

Simulators in the sciences and beyond

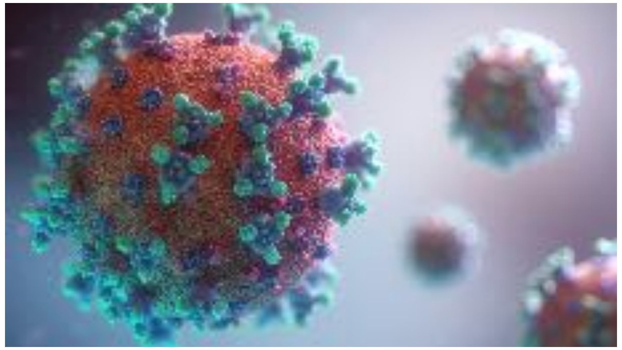
Particle Physics (CERN)



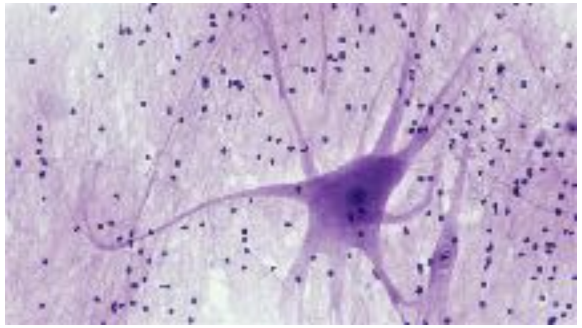
Neuroscience

Simulators in the sciences and beyond

Particle Physics (CERN)



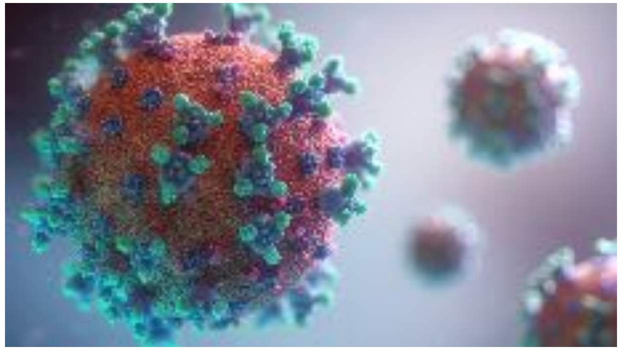
Epidemiology



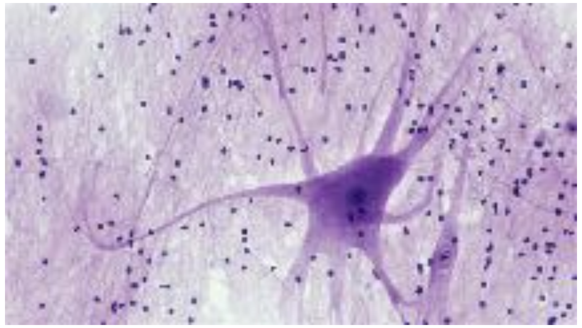
Neuroscience

Simulators in the sciences and beyond

Particle Physics (CERN)



Epidemiology

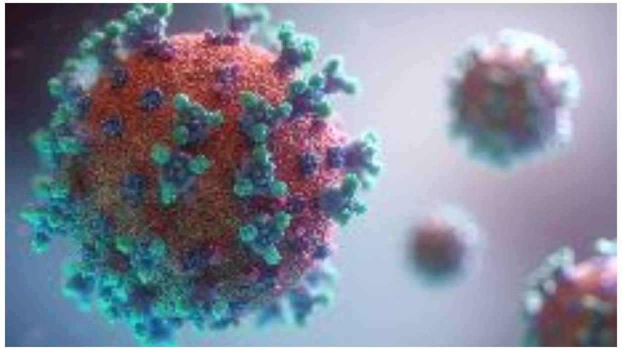


Neuroscience

Genomics

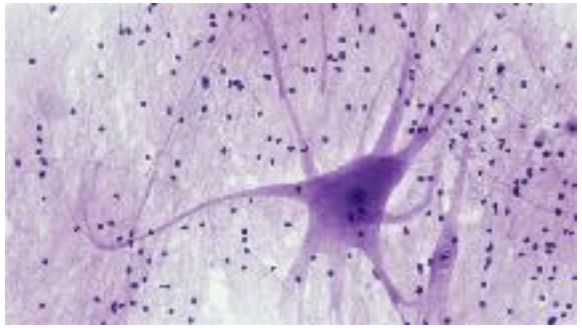
Simulators in the sciences and beyond

Particle Physics (CERN)



Epidemiology

Health monitoring (Apple)

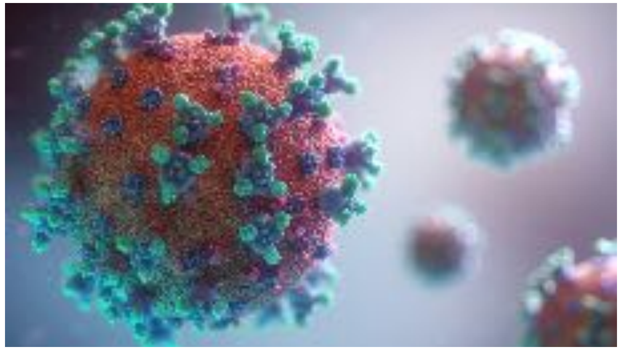


Neuroscience

Genomics

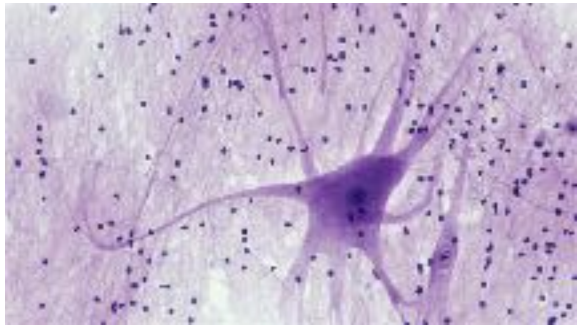
Simulators in the sciences and beyond

Particle Physics (CERN)



Epidemiology

Health monitoring (Apple)



Neuroscience

Genomics

https://simulation-based-inference.org/

Clarifying terminology

- **Are diffusion models simulators?**

https://developer.nvidia.com/blog/improving-diffusion-models-as-an-alternative-to-gans-part-1/

Clarifying terminology

- **Are diffusion models simulators?**

Yes, you can think of this process as defining a simulator!

https://developer.nvidia.com/blog/improving-diffusion-models-as-an-alternative-to-gans-part-1/

Clarifying terminology

- **Are diffusion models simulators?**

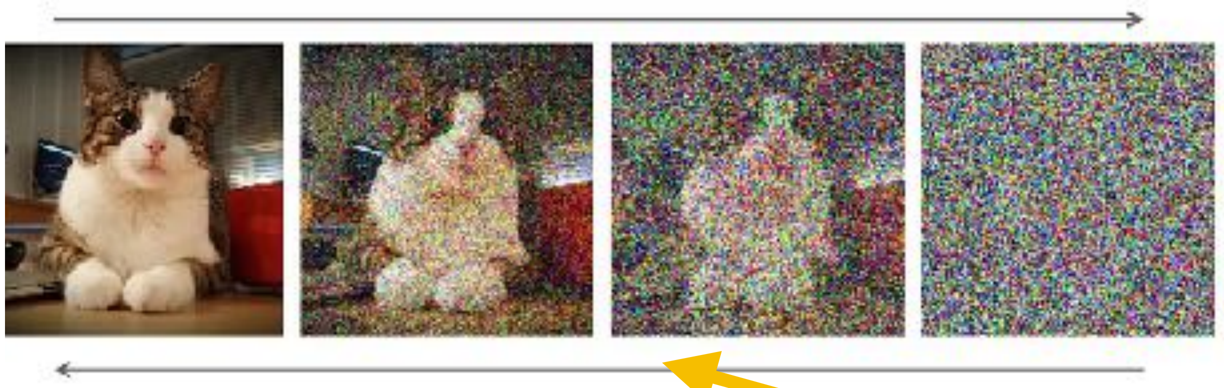
Yes, you can think of this process as defining a simulator!

- **Does this mean we are getting yet another course on diffusion models?**

https://developer.nvidia.com/blog/improving-diffusion-models-as-an-alternative-to-gans-part-1/

Clarifying terminology

- **Are diffusion models simulators?**



Yes, you can think of this process as defining a simulator!

- **Does this mean we are getting yet another course on diffusion models?**

No! In SBI, we typically have **scientifically meaningful simulators** where **the parameter θ can be interpreted.** We therefore really care about estimating it and providing **uncertainty estimates**!

https://developer.nvidia.com/blog/improving-diffusion-models-as-an-alternative-to-gans-part-1/

Any Questions?

What is coming up

- Basic methods:

Minimum distance estimation

Approximate Bayesian Computation

Neural simulation-based inference

What is coming up

- Basic methods:

Minimum distance estimation

Approximate Bayesian Computation

Neural simulation-based inference

- Discussion of the main challenges in SBI.

What is coming up

- Basic methods:

Minimum distance estimation

Approximate Bayesian Computation

Neural simulation-based inference

- Discussion of the main challenges in SBI.
- Some illustrations of recent advances:

Hikida, Y., Bharti, A., Jeffrey, N. & **Briol, F-X** (2025). Multilevel neural simulation-based inference. arXiv:2506.06087 (to appear at NeurIPS?).

Bharti, A., Huang, D., Kaski, S., & **Briol, F.-X.** (2025). Cost-aware simulation-based inference. International Conference on Artificial Intelligence and Statistics, 28–36.

Dellaporta, C., Knoblauch, J., Damoulas, T. & **Briol, F-X** (2022). Robust Bayesian inference for simulator-based models via the MMD posterior bootstrap. AISTATS, 943-970. Best paper award.

Minimum Distance Estimation

Minimum Distance Estimation

(i.e. how to be a frequentist in SBI…)

The method of moments

- Model: $\{\mathbb{P}_\theta\}_{\theta \in \Theta}$, Data-generating process: $\mathbb{Q}$, Data: $y_1, \ldots, y_n$

The method of moments

- Model: $\{\mathbb{P}_\theta\}_{\theta\in\Theta}$,   Data-generating process: $\mathbb{Q}$,   Data: $y_1, \ldots, y_n$
- **Idea:** For two distributions to be the same, their moments must match…

  Why don't we find θ such that $\mathbb{E}_{X\sim\mathbb{P}_\theta}[X] = \mathbb{E}_{X\sim\mathbb{Q}}[X]$?

The method of moments

- Model: $\{\mathbb{P}_\theta\}_{\theta \in \Theta}$, Data-generating process: $\mathbb{Q}$, Data: $y_1, \ldots, y_n$
- **Idea:** For two distributions to be the same, their moments must match…

  Why don't we find θ such that $|\mathbb{E}_{X \sim \mathbb{P}_\theta}[X] - \mathbb{E}_{X \sim \mathbb{Q}}[X]|$ is small ?

The method of moments

- Model: $\{\mathbb{P}_\theta\}_{\theta\in\Theta}$,   Data-generating process: $\mathbb{Q}$,   Data: $y_1, \ldots, y_n$
- **Idea:** For two distributions to be the same, their moments must match…

Why don't we find θ such that $\left|\mathbb{E}_{X\sim\mathbb{P}_\theta}[X] - \frac{1}{n}\sum_{i=1}^{n} y_i\right|$ is small ?

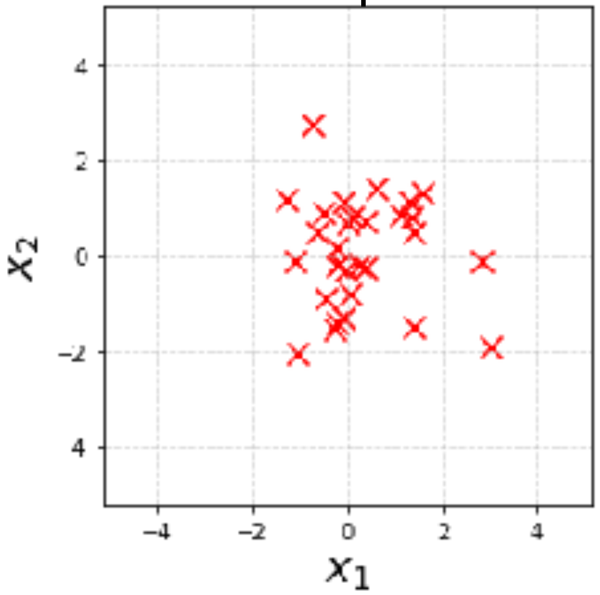
The method of moments

- Model: $\{\mathbb{P}_\theta\}_{\theta \in \Theta}$,   Data-generating process: $\mathbb{Q}$,   Data: $y_1, \ldots, y_n$
- **Idea:** For two distributions to be the same, their moments must match…

Why don't we find θ such that $\left| \mathbb{E}_{X \sim \mathbb{P}_\theta}[X] - \frac{1}{n}\sum_{i=1}^{n} y_i \right|$ is small ?

$$\mathbb{P}_\theta = \mathcal{N}(\theta, I_{2\times 2})$$

i.e. $\mathbb{E}_{X \sim \mathbb{P}_\theta}[X] = \theta$



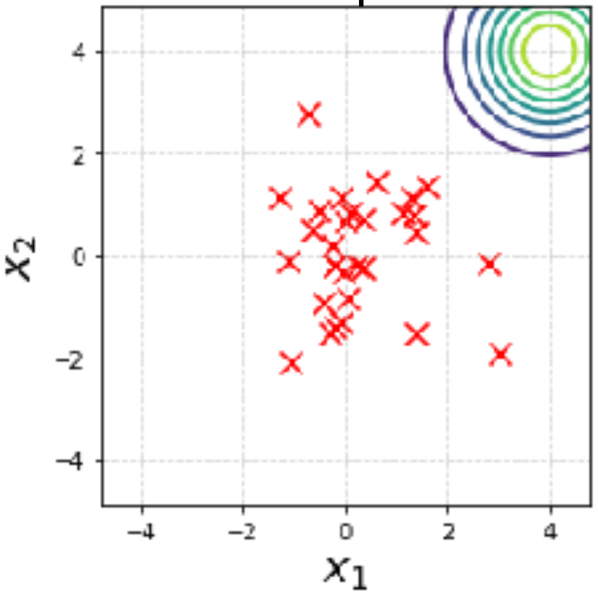
The method of moments

- Model: $\{\mathbb{P}_\theta\}_{\theta \in \Theta}$, Data-generating process: $\mathbb{Q}$, Data: $y_1, \ldots, y_n$
- **Idea:** For two distributions to be the same, their moments must match…

Why don't we find θ such that $\left| \mathbb{E}_{X \sim \mathbb{P}_\theta}[X] - \frac{1}{n} \sum_{i=1}^{n} y_i \right|$ is small ?

$\mathbb{P}_\theta = \mathcal{N}(\theta, I_{2\times 2})$

i.e. $\mathbb{E}_{X \sim \mathbb{P}_\theta}[X] = \theta$

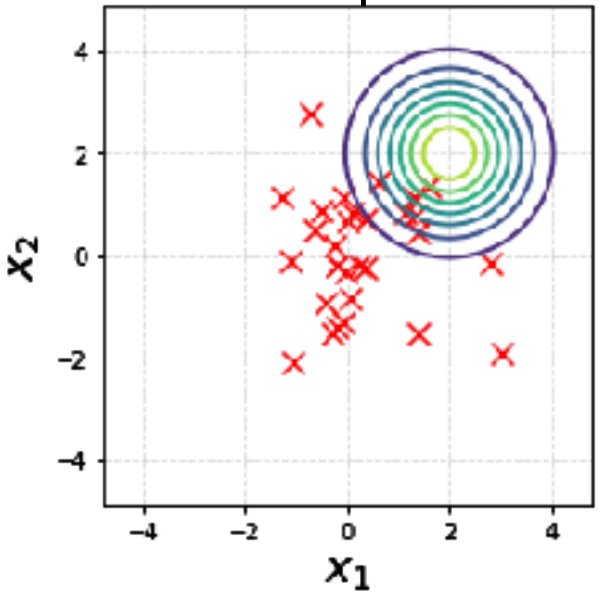


The method of moments

- Model: $\{\mathbb{P}_\theta\}_{\theta \in \Theta}$, Data-generating process: $\mathbb{Q}$, Data: $y_1, \ldots, y_n$
- **Idea:** For two distributions to be the same, their moments must match…

Why don't we find θ such that $\left| \mathbb{E}_{X \sim \mathbb{P}_\theta}[X] - \frac{1}{n}\sum_{i=1}^{n} y_i \right|$ is small ?

$\mathbb{P}_\theta = \mathcal{N}(\theta, I_{2\times 2})$
i.e. $\mathbb{E}_{X\sim\mathbb{P}_\theta}[X] = \theta$

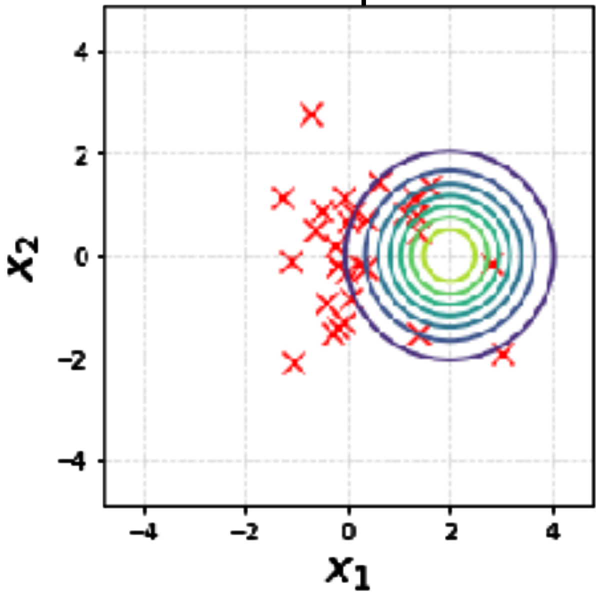


The method of moments

- Model: $\{\mathbb{P}_\theta\}_{\theta\in\Theta}$, Data-generating process: $\mathbb{Q}$, Data: $y_1, \ldots, y_n$
- **Idea:** For two distributions to be the same, their moments must match…

Why don't we find θ such that $\left| \mathbb{E}_{X\sim\mathbb{P}_\theta}[X] - \frac{1}{n}\sum_{i=1}^{n} y_i \right|$ is small ?

$$\mathbb{P}_\theta = \mathcal{N}(\theta, I_{2\times 2})$$
i.e. $\mathbb{E}_{X\sim\mathbb{P}_\theta}[X] = \theta$



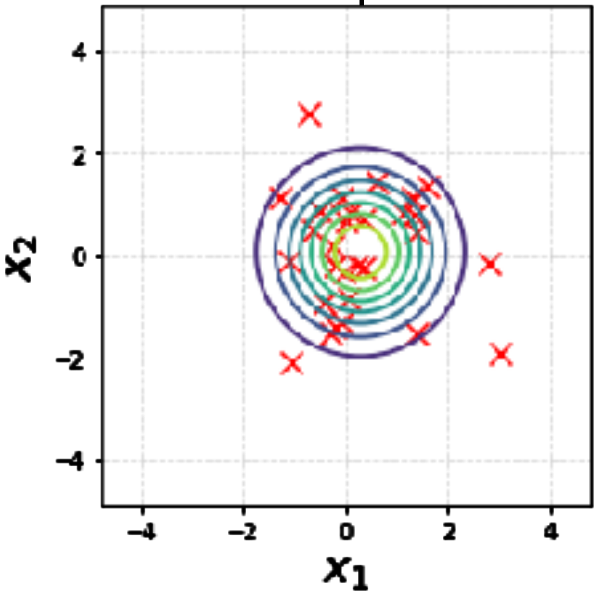
The method of moments

- Model: $\{\mathbb{P}_\theta\}_{\theta\in\Theta}$,   Data-generating process: $\mathbb{Q}$,   Data: $y_1, \ldots, y_n$
- **Idea:** For two distributions to be the same, their moments must match…

Why don't we find θ such that $\left| \mathbb{E}_{X\sim\mathbb{P}_\theta}[X] - \frac{1}{n}\sum_{i=1}^{n} y_i \right|$ is small ?

$\mathbb{P}_\theta = \mathcal{N}(\theta, I_{2\times 2})$

i.e. $\mathbb{E}_{X\sim\mathbb{P}_\theta}[X] = \theta$

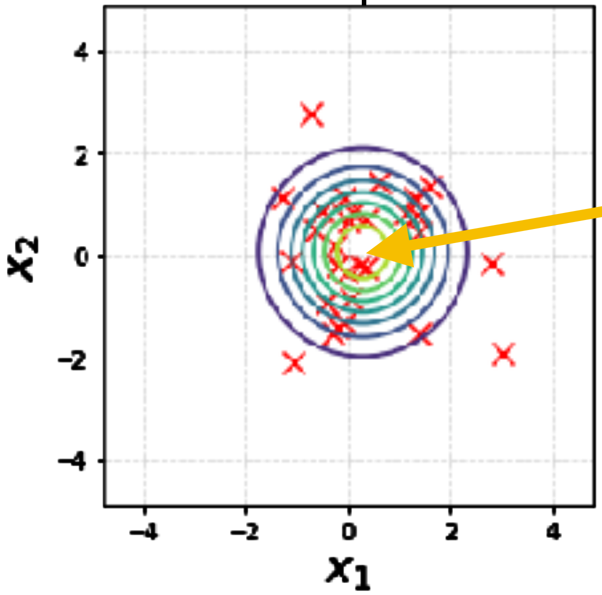


The method of moments

- Model: $\{\mathbb{P}_\theta\}_{\theta\in\Theta}$, Data-generating process: $\mathbb{Q}$, Data: $y_1, \ldots, y_n$
- **Idea:** For two distributions to be the same, their moments must match…

Why don't we find θ such that $\left| \mathbb{E}_{X\sim\mathbb{P}_\theta}[X] - \frac{1}{n}\sum_{i=1}^{n} y_i \right|$ is small ?

$$\mathbb{P}_\theta = \mathcal{N}(\theta, I_{2\times 2})$$
i.e. $\mathbb{E}_{X\sim\mathbb{P}_\theta}[X] = \theta$



$\hat{\theta}_n = (0.29, 0.07)^\top$

Not perfect, but will get there as n grows

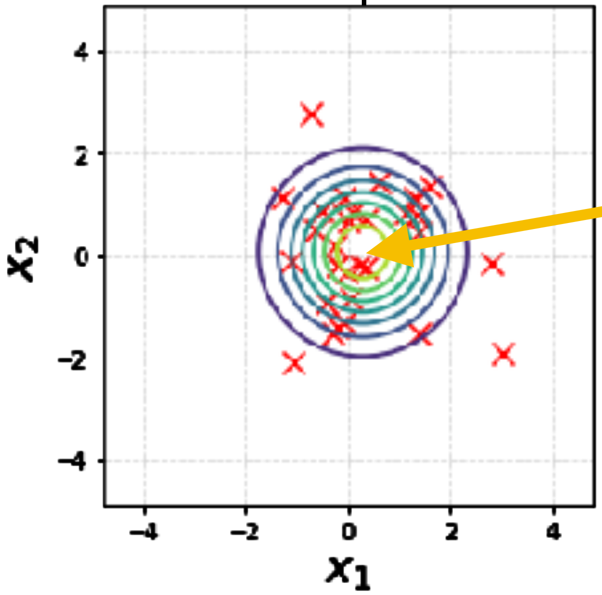
The method of moments

- Model: $\{\mathbb{P}_\theta\}_{\theta\in\Theta}$, Data-generating process: $\mathbb{Q}$, Data: $y_1, \ldots, y_n$
- **Idea:** For two distributions to be the same, their moments must match…

Why don't we find θ such that $\left| \mathbb{E}_{X\sim\mathbb{P}_\theta}[X] - \frac{1}{n}\sum_{i=1}^{n} y_i \right|$ is small ?

$$\mathbb{P}_\theta = \mathcal{N}(\theta, I_{2\times 2})$$

i.e. $\mathbb{E}_{X\sim\mathbb{P}_\theta}[X] = \theta$



$\hat{\theta}_n = (0.29, 0.07)^\top$

Not perfect, but will get there as n grows

Note: For more complex models, we may also want to compare higher moments…

The method of simulated moments

- **Problem:** We work with simulators, and so we can’t necessarily compute the mean!

McFadden, D. (1989). A method of simulated moments for estimation of discrete response models without numerical integration. *Econometrica*, *57*(5), 995–1026.

The method of <u>simulated</u> moments

- **Problem:** We work with simulators, and so we can't necessarily compute the mean!

$$\mathbb{E}_{X \sim \mathbb{P}_\theta}[X] \approx \frac{1}{n} \sum_{i=1}^{n} y_i$$

McFadden, D. (1989). A method of simulated moments for estimation of discrete response models without numerical integration. *Econometrica*, *57*(5), 995–1026.

The method of simulated moments

- **Problem:** We work with simulators, and so we can't necessarily compute the mean!

???

$$\mathbb{E}_{X \sim \mathbb{P}_\theta}[X] \approx \frac{1}{n} \sum_{i=1}^{n} y_i$$

??? ???

McFadden, D. (1989). A method of simulated moments for estimation of discrete response models without numerical integration. *Econometrica*, *57*(5), 995–1026.

The method of <u>simulated</u> moments

- **Problem:** We work with simulators, and so we can't necessarily compute the mean!

???

$$\mathbb{E}_{X \sim \mathbb{P}_\theta}[X] \approx \frac{1}{n} \sum_{i=1}^{n} y_i$$

???

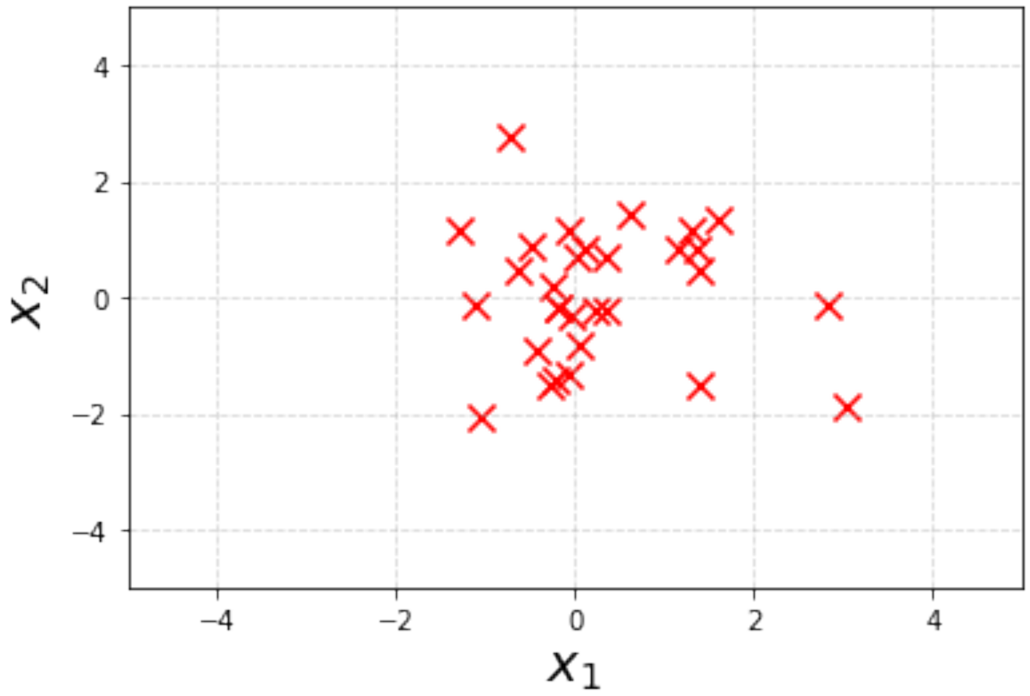
???

- **Method of simulated moments:** We repeat the method of moments, but we simulate at each iteration!

McFadden, D. (1989). A method of simulated moments for estimation of discrete response models without numerical integration. *Econometrica*, *57*(5), 995–1026.

The method of <u>simulated</u> moments

Fix grid $\theta_1, \ldots, \theta_T \in \Theta$.

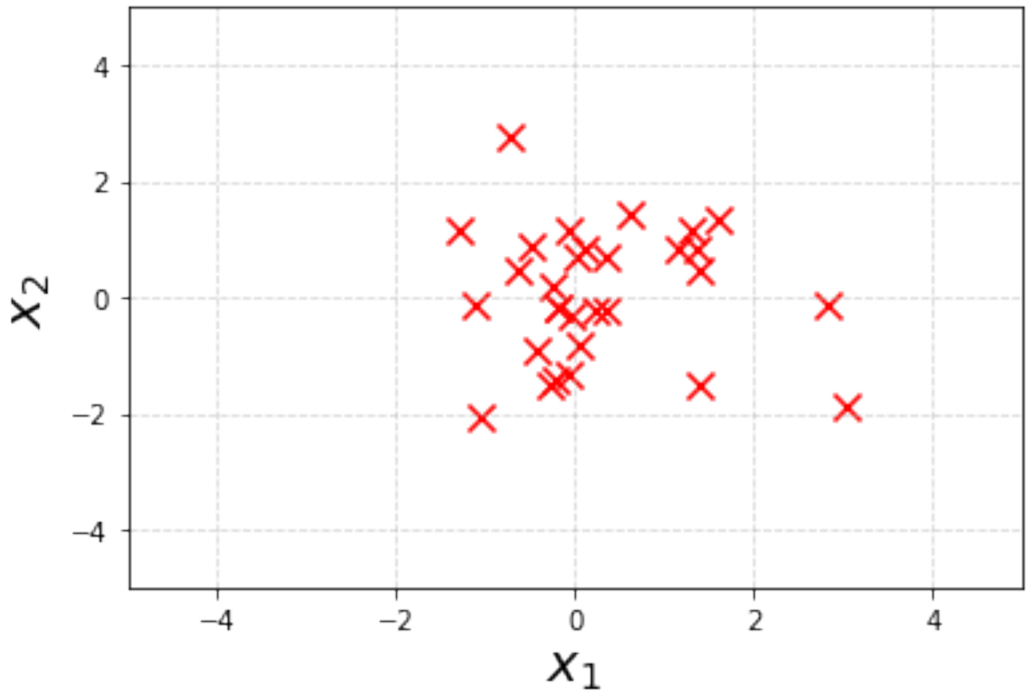


The method of simulated moments

Fix grid $\theta_1, \ldots, \theta_T \in \Theta$.

For $t \in \{1,\ldots,T\}$,

1) Simulate $x_1, \ldots, x_n \sim \mathbb{P}_{\theta_t}$



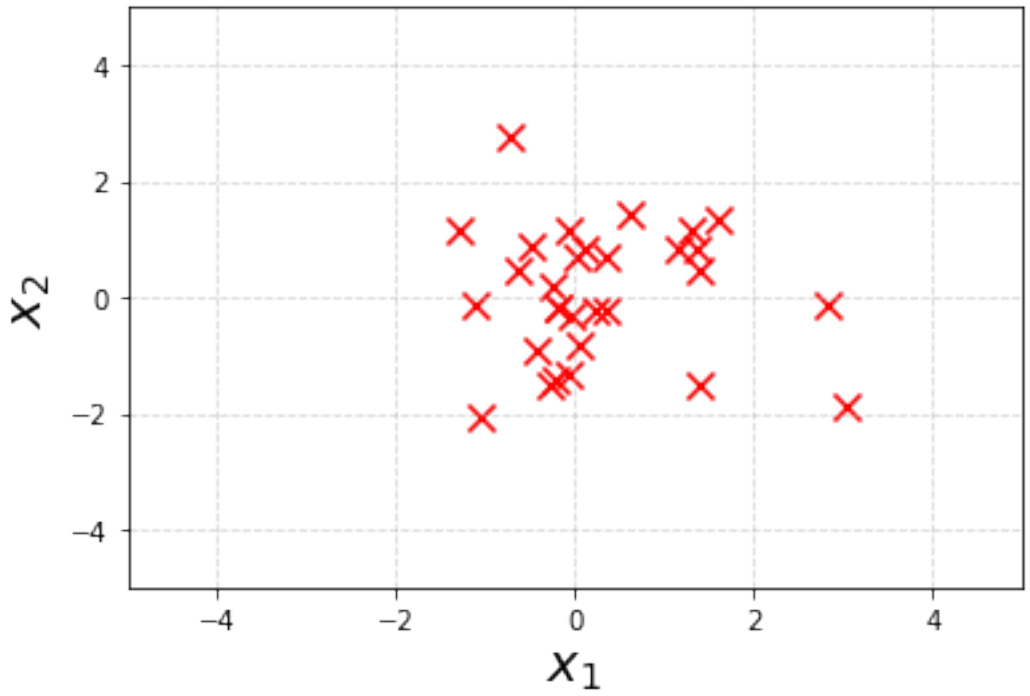
The method of <u>simulated</u> moments

Fix grid $\theta_1, \ldots, \theta_T \in \Theta$.

For $t \in \{1, \ldots, T\}$,

1) Simulate $x_1, \ldots, x_n \sim \mathbb{P}_{\theta_t}$

2) Compute $\left| \frac{1}{n} \sum_{i=1}^{n} x_i - \frac{1}{n} \sum_{i=1}^{n} y_i \right|$



The method of simulated moments

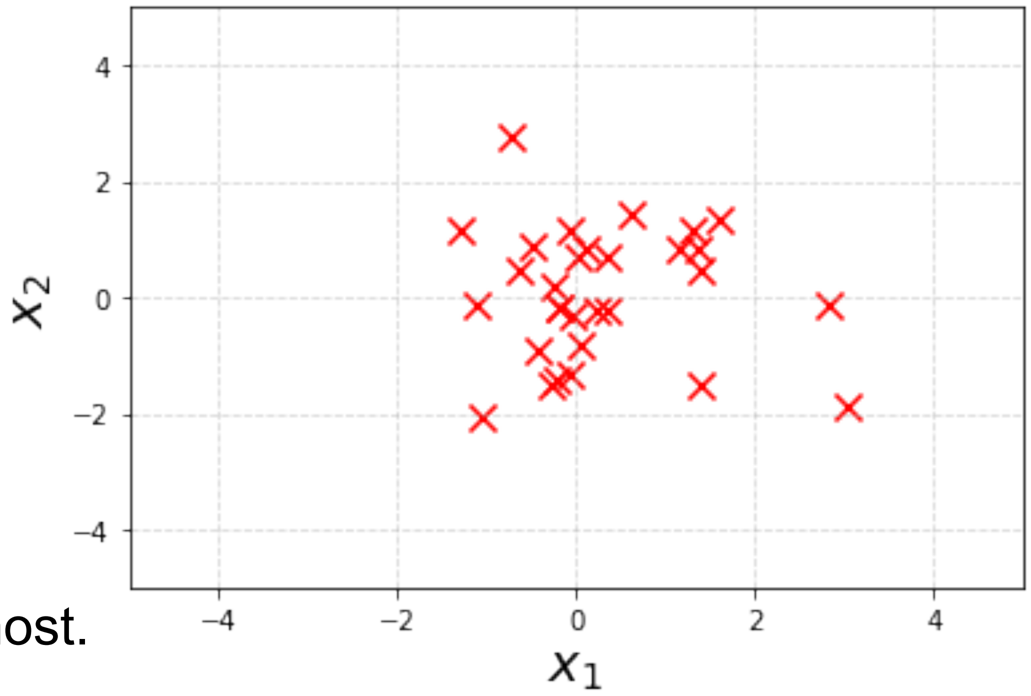
Fix grid $\theta_1, \ldots, \theta_T \in \Theta$.

For $t \in \{1, \ldots, T\}$,

1) Simulate $x_1, \ldots, x_n \sim \mathbb{P}_{\theta_t}$

2) Compute $\left| \frac{1}{n} \sum_{i=1}^{n} x_i - \frac{1}{n} \sum_{i=1}^{n} y_i \right|$

Return parameter value where moments match most.



The method of <u>simulated</u> moments

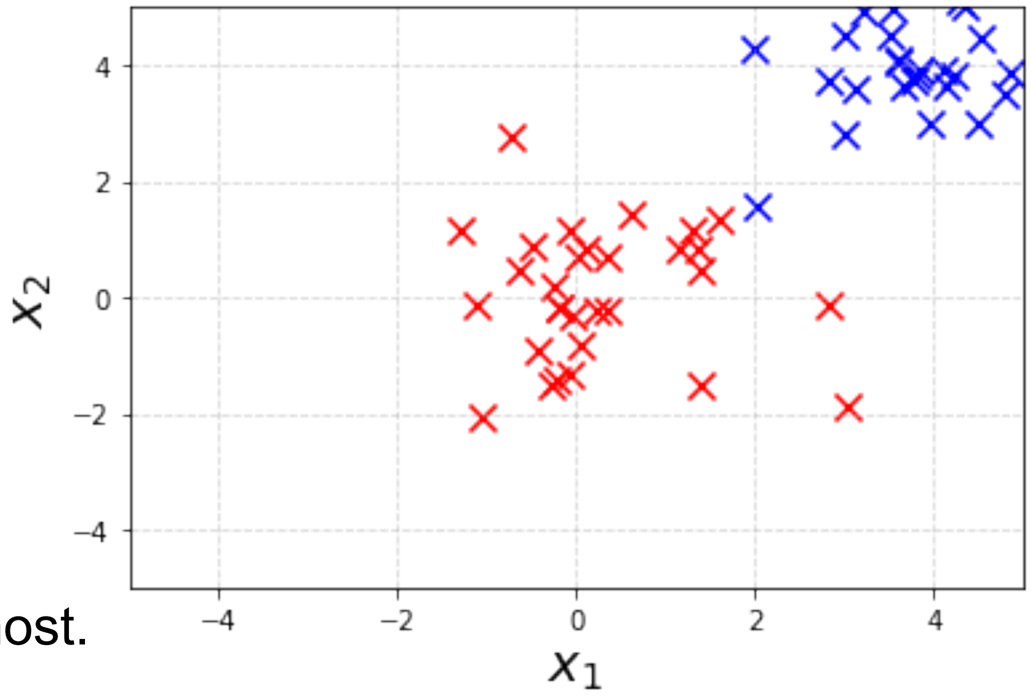
Fix grid $\theta_1, \ldots, \theta_T \in \Theta$.

For $t \in \{1, \ldots, T\}$,

1) Simulate $x_1, \ldots, x_n \sim \mathbb{P}_{\theta_t}$

2) Compute $\left| \frac{1}{n} \sum_{i=1}^{n} x_i - \frac{1}{n} \sum_{i=1}^{n} y_i \right|$

Return parameter value where moments match most.



The method of simulated moments

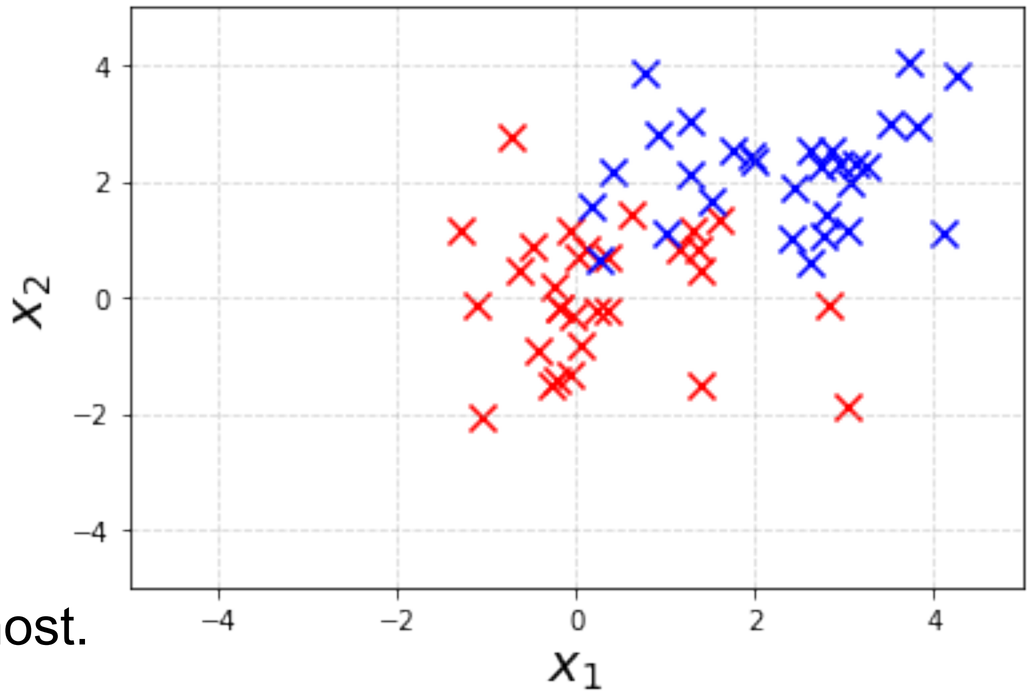
Fix grid $\theta_1, \ldots, \theta_T \in \Theta$.

For $t \in \{1,\ldots,T\}$,

1) Simulate $x_1, \ldots, x_n \sim \mathbb{P}_{\theta_t}$

2) Compute $\left| \frac{1}{n}\sum_{i=1}^{n} x_i - \frac{1}{n}\sum_{i=1}^{n} y_i \right|$

Return parameter value where moments match most.



The method of <u>simulated</u> moments

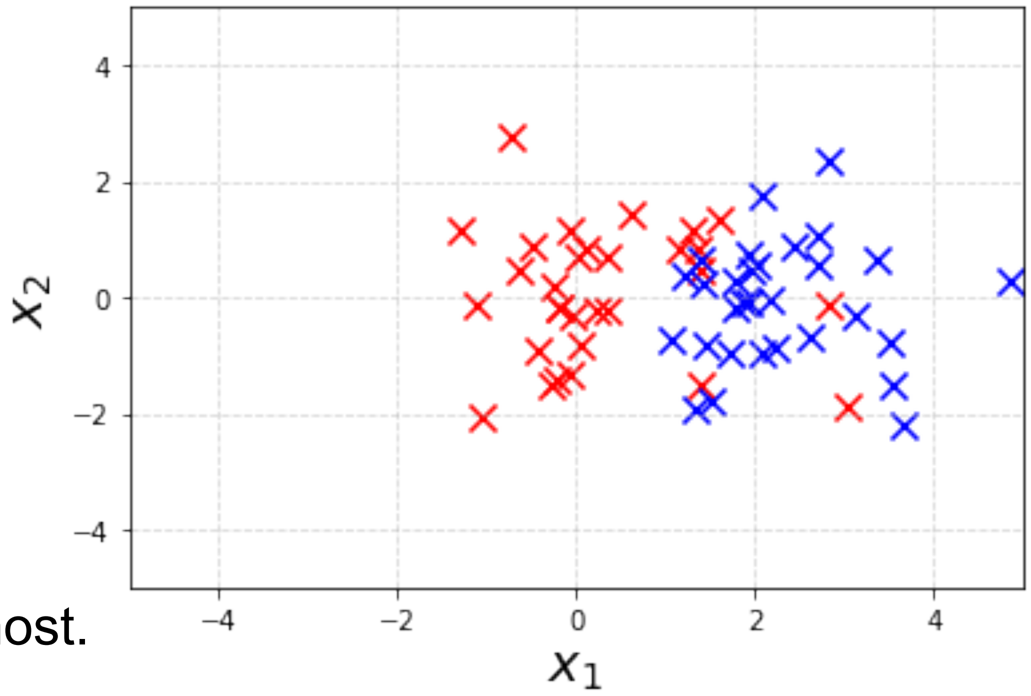
Fix grid $\theta_1, \ldots, \theta_T \in \Theta$.

For $t \in \{1, \ldots, T\}$,

1) Simulate $x_1, \ldots, x_n \sim \mathbb{P}_{\theta_t}$

2) Compute $\left| \frac{1}{n} \sum_{i=1}^{n} x_i - \frac{1}{n} \sum_{i=1}^{n} y_i \right|$

Return parameter value where moments match most.



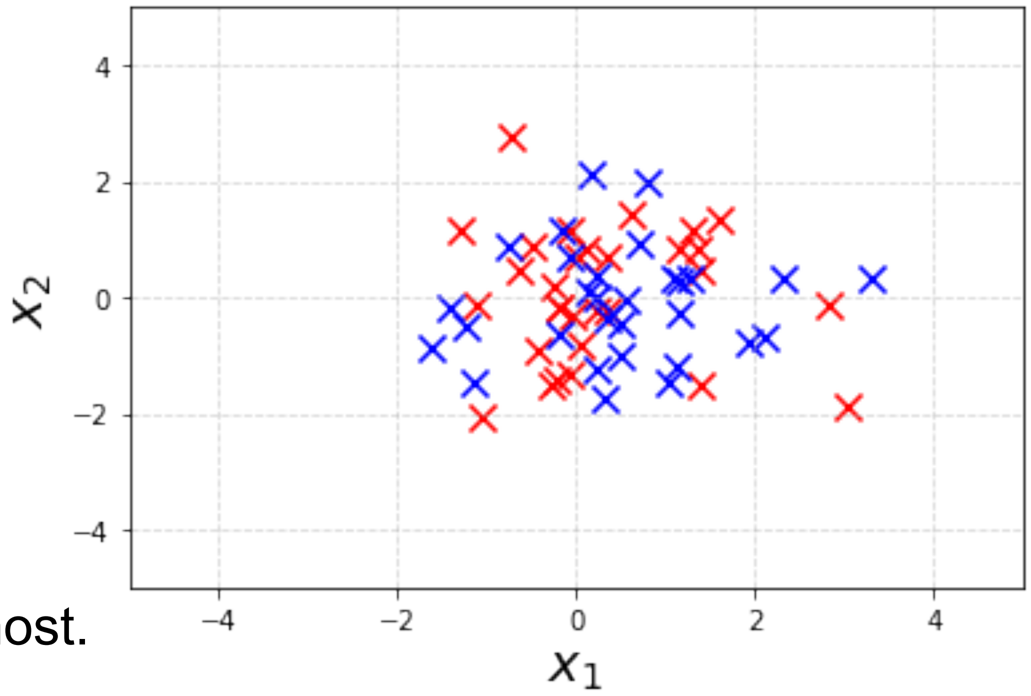
The method of simulated moments

Fix grid $\theta_1, \ldots, \theta_T \in \Theta$.

For $t \in \{1, \ldots, T\}$,

1) Simulate $x_1, \ldots, x_n \sim \mathbb{P}_{\theta_t}$
2) Compute $\left| \frac{1}{n} \sum_{i=1}^{n} x_i - \frac{1}{n} \sum_{i=1}^{n} y_i \right|$

Return parameter value where moments match most.



The method of simulated moments

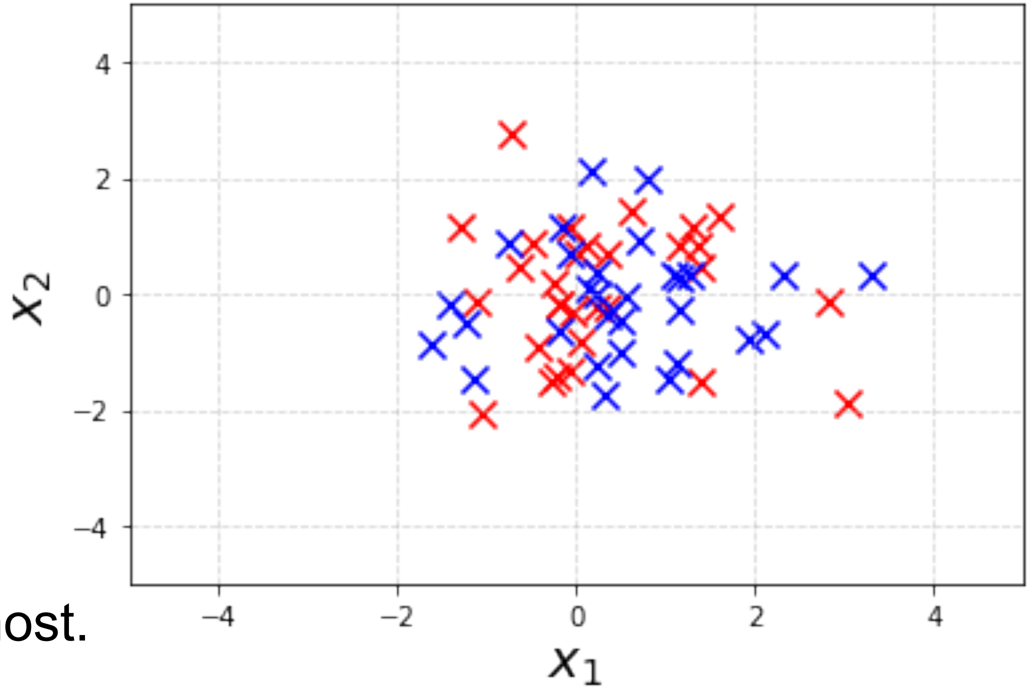
Fix grid $\theta_1, \ldots, \theta_T \in \Theta$.

For $t \in \{1,\ldots,T\}$,

1) Simulate $x_1, \ldots, x_n \sim \mathbb{P}_{\theta_t}$

2) Compute $\left| \frac{1}{n} \sum_{i=1}^{n} x_i - \frac{1}{n} \sum_{i=1}^{n} y_i \right|$

Return parameter value where moments match most.



- In practice, this is implemented much more efficiently than by grid search…

Minimum distance estimation for simulators

- A general framework for frequentists: $\theta^* := \arg\min_{\theta \in \Theta} D(\mathbb{P}_\theta, \mathbb{Q})$

Wolfowitz, J. (1957). The minimum distance method. The Annals of Mathematical Statistics, 28(1), 75–88.

Minimum distance estimation for simulators

- A general framework for frequentists: $\theta^* := \arg\min_{\theta \in \Theta} D(\mathbb{P}_\theta, \mathbb{Q})$
- The estimator: $\hat{\theta}_n := \arg\min_{\theta \in \Theta} D(\mathbb{P}_\theta, \mathbb{Q}_n) \leftarrow \mathbb{Q}_n = \frac{1}{n} \sum_{i=1}^{n} \delta_{y_i}$

Wolfowitz, J. (1957). The minimum distance method. The Annals of Mathematical Statistics, 28(1), 75–88.

Minimum distance estimation for simulators

- A general framework for frequentists: $$\theta^* := \arg\min_{\theta \in \Theta} D(\mathbb{P}_\theta, \mathbb{Q})$$
- The estimator: $$\hat{\theta}_n := \arg\min_{\theta \in \Theta} D(\mathbb{P}_\theta, \mathbb{Q}_n) \longleftarrow \mathbb{Q}_n = \frac{1}{n}\sum_{i=1}^{n} \delta_{y_i}$$
- The objective is intractable, but we can get an estimate by sampling and use stochastic optimisation:

$$D(\mathbb{P}_\theta, \mathbb{Q}_n) \approx D((\mathbb{P}_\theta)_n, \mathbb{Q}_n)$$

$$(\mathbb{P}_\theta)_n = \frac{1}{n}\sum_{i=1}^{n} \delta_{x_i}$$

$$x_1, \ldots, x_n \sim \mathbb{P}_\theta$$

Wolfowitz, J. (1957). The minimum distance method. The Annals of Mathematical Statistics, 28(1), 75–88.

Minimum distance estimation for simulators

- A general framework for frequentists: $$\theta^* := \arg\min_{\theta\in\Theta} D(\mathbb{P}_\theta, \mathbb{Q})$$

- The estimator: $$\hat{\theta}_n := \arg\min_{\theta\in\Theta} D(\mathbb{P}_\theta, \mathbb{Q}_n) \longleftarrow \mathbb{Q}_n = \frac{1}{n}\sum_{i=1}^{n} \delta_{y_i}$$

- The objective is intractable, but we can get an estimate by sampling and use stochastic optimisation:

$$D(\mathbb{P}_\theta, \mathbb{Q}_n) \approx D((\mathbb{P}_\theta)_n, \mathbb{Q}_n)$$

$$(\mathbb{P}_\theta)_n = \frac{1}{n}\sum_{i=1}^{n} \delta_{x_i}$$

$$x_1, \ldots, x_n \sim \mathbb{P}_\theta$$

- Can pick our favourite discrepancy/divergence/distance!

Wolfowitz, J. (1957). The minimum distance method. The Annals of Mathematical Statistics, 28(1), 75–88.

The choice of discrepancy

- Desirable criteria:

The choice of discrepancy

- Desirable criteria:

  (1) It should be a divergence: $D(\mathbb{P}, \mathbb{Q}) = 0 \Leftrightarrow \mathbb{P} = \mathbb{Q}$

The choice of discrepancy

- Desirable criteria:

  (1) It should be a divergence: $D(\mathbb{P}, \mathbb{Q}) = 0 \Leftrightarrow \mathbb{P} = \mathbb{Q}$

  (2) It should be easy to estimate from samples: $D(\mathbb{P}_n, \mathbb{Q}_n) \approx D(\mathbb{P}, \mathbb{Q})$

The choice of discrepancy

- Desirable criteria:

  (1) It should be a divergence: $D(\mathbb{P}, \mathbb{Q}) = 0 \Leftrightarrow \mathbb{P} = \mathbb{Q}$

  (2) It should be easy to estimate from samples: $D(\mathbb{P}_n, \mathbb{Q}_n) \approx D(\mathbb{P}, \mathbb{Q})$

  (3) It should be somewhat interpretable.

The choice of discrepancy

- Desirable criteria:

  (1) It should be a divergence: $D(\mathbb{P}, \mathbb{Q}) = 0 \Leftrightarrow \mathbb{P} = \mathbb{Q}$

  (2) It should be easy to estimate from samples: $D(\mathbb{P}_n, \mathbb{Q}_n) \approx D(\mathbb{P}, \mathbb{Q})$

  (3) It should be somewhat interpretable.

  (4) It should be robust/emphasise important differences for inference?

The choice of discrepancy

- Desirable criteria:

  (1) It should be a divergence: $D(\mathbb{P}, \mathbb{Q}) = 0 \Leftrightarrow \mathbb{P} = \mathbb{Q}$

  (2) It should be easy to estimate from samples: $D(\mathbb{P}_n, \mathbb{Q}_n) \approx D(\mathbb{P}, \mathbb{Q})$

  (3) It should be somewhat interpretable.

  (4) It should be robust/emphasise important differences for inference?

- **Integral probability metrics:**

$$D(\mathbb{P}, \mathbb{Q}) := \sup_{f \in \mathscr{F}} \left| \mathbb{E}_{X \sim \mathbb{P}}[f(X)] - \mathbb{E}_{X \sim \mathbb{Q}}[f(X)] \right|$$

The choice of discrepancy

- Desirable criteria:

  (1) It should be a divergence: $D(\mathbb{P}, \mathbb{Q}) = 0 \Leftrightarrow \mathbb{P} = \mathbb{Q}$

  (2) It should be easy to estimate from samples: $D(\mathbb{P}_n, \mathbb{Q}_n) \approx D(\mathbb{P}, \mathbb{Q})$

  (3) It should be somewhat interpretable.

  (4) It should be robust/emphasise important differences for inference?

- **Integral probability metrics:**

$$D(\mathbb{P}, \mathbb{Q}) := \sup_{f \in \mathscr{F}} \left| \mathbb{E}_{X \sim \mathbb{P}}[f(X)] - \mathbb{E}_{X \sim \mathbb{Q}}[f(X)] \right|$$

An entire (quite possibly infinite) family of moments

The choice of discrepancy

- Desirable criteria:

  (1) It should be a divergence: $D(\mathbb{P},\mathbb{Q}) = 0 \Leftrightarrow \mathbb{P} = \mathbb{Q}$

  (2) It should be easy to estimate from samples: $D(\mathbb{P}_n,\mathbb{Q}_n) \approx D(\mathbb{P},\mathbb{Q})$

  (3) It should be somewhat interpretable.

  (4) It should be robust/emphasise important differences for inference?

- **Integral probability metrics:**

$$D(\mathbb{P},\mathbb{Q}) := \sup_{f \in \mathscr{F}} \left| \mathbb{E}_{X\sim\mathbb{P}}[f(X)] - \mathbb{E}_{X\sim\mathbb{Q}}[f(X)] \right|$$

An entire (quite possibly infinite) family of moments

This will typically make things intractable unless $\mathscr{F}$ is picked carefully!

The Wasserstein distance

$$W(\mathbb{P}, \mathbb{Q}) := \sup_{f \in \mathcal{F}_W} \left| \mathbb{E}_{X \sim \mathbb{P}}[f(X)] - \mathbb{E}_{X \sim \mathbb{Q}}[f(X)] \right|$$

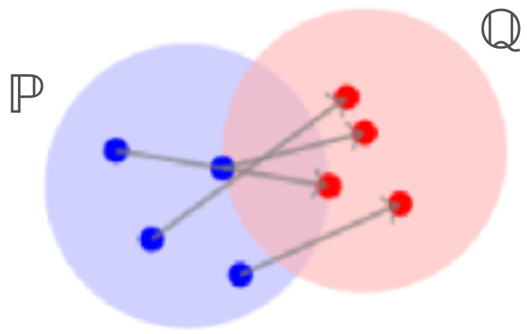
$$\mathcal{F}_W := \{f : \mathcal{X} \to \mathbb{R} : |f(x) - f(y)| \leq \|x - y\|\}$$

The Wasserstein distance

$$W(\mathbb{P}, \mathbb{Q}) := \sup_{f \in \mathcal{F}_W} \left| \mathbb{E}_{X \sim \mathbb{P}}[f(X)] - \mathbb{E}_{X \sim \mathbb{Q}}[f(X)] \right|$$

$$\mathcal{F}_W := \{f : \mathcal{X} \to \mathbb{R} : |f(x) - f(y)| \leq \|x - y\|\}$$

- Well-known interpretation as the **cost of moving mass** from $\mathbb{P}$ to $\mathbb{Q}$!



Credit for figure:

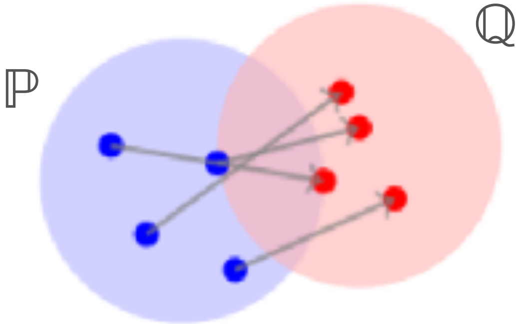
The Wasserstein distance

$$W(\mathbb{P}, \mathbb{Q}) := \sup_{f \in \mathscr{F}_W} \left| \mathbb{E}_{X \sim \mathbb{P}}[f(X)] - \mathbb{E}_{X \sim \mathbb{Q}}[f(X)] \right|$$

$$\mathscr{F}_W := \{ f : \mathscr{X} \to \mathbb{R} : |f(x) - f(y)| \le \|x - y\| \}$$

- Well-known interpretation as the **cost of moving mass** from $\mathbb{P}$ to $\mathbb{Q}$!

(1) Divergence ✓



Credit for figure:

The Wasserstein distance

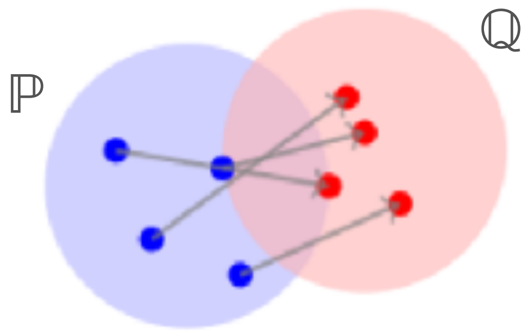
$$W(\mathbb{P}, \mathbb{Q}) := \sup_{f \in \mathcal{F}_W} \left| \mathbb{E}_{X \sim \mathbb{P}}[f(X)] - \mathbb{E}_{X \sim \mathbb{Q}}[f(X)] \right|$$

$$\mathcal{F}_W := \{f : \mathcal{X} \to \mathbb{R} : |f(x) - f(y)| \leq \|x - y\| \}$$

- Well-known interpretation as the **cost of moving mass** from $\mathbb{P}$ to $\mathbb{Q}$!

(1) Divergence ✓

(2) Easy to estimate ~



Credit for figure:

The Wasserstein distance

$$W(\mathbb{P}, \mathbb{Q}) := \sup_{f \in \mathcal{F}_W} \left| \mathbb{E}_{X \sim \mathbb{P}}[f(X)] - \mathbb{E}_{X \sim \mathbb{Q}}[f(X)] \right|$$

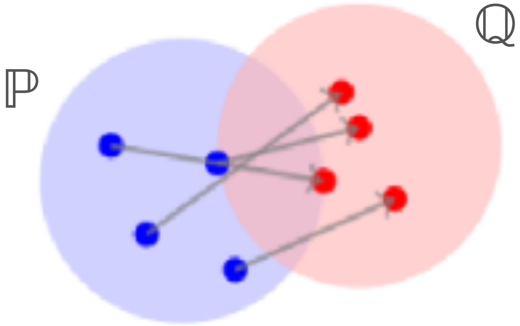
$$\mathcal{F}_W := \{ f : \mathcal{X} \to \mathbb{R} : |f(x) - f(y)| \leq \|x - y\| \}$$

- Well-known interpretation as the **cost of moving mass** from $\mathbb{P}$ to $\mathbb{Q}$!

(1) Divergence

(2) Easy to estimate ~

(3) Interpretable



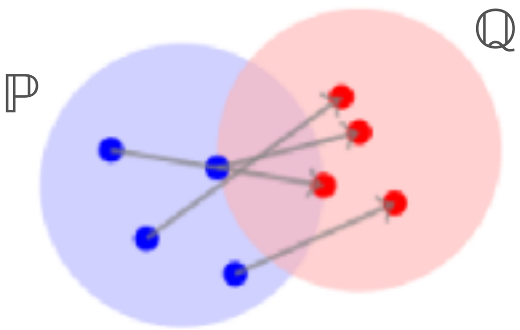
Credit for figure:

The Wasserstein distance

$$W(\mathbb{P},\mathbb{Q}) := \sup_{f\in\mathscr{F}_W} \left| \mathbb{E}_{X\sim\mathbb{P}}[f(X)] - \mathbb{E}_{X\sim\mathbb{Q}}[f(X)] \right|$$

$$\mathscr{F}_W := \{f : \mathscr{X} \to \mathbb{R} : |f(x) - f(y)| \leq \|x-y\|\}$$

- Well-known interpretation as the **cost of moving mass** from $\mathbb{P}$ to $\mathbb{Q}$!



(1) Divergence ✓

(2) Easy to estimate ~

(3) Interpretable ✓

(4) Robust ✗

Credit for figure:

Minimum Wasserstein estimators

- Once we have n samples from $\mathbb{P}_\theta$ and $\mathbb{Q}$, this turns into an optimal transport which can be solved in $O(n \log n)$ in $d = 1$ and $O(n^3)$ for $d > 1$.

Minimum Wasserstein estimators

- Once we have n samples from $\mathbb{P}_\theta$ and $\mathbb{Q}$, this turns into an optimal transport which can be solved in $O(n \log n)$ in $d = 1$ and $O(n^3)$ for $d > 1$.

Minimum Wasserstein estimators

- Once we have n samples from $\mathbb{P}_\theta$ and $\mathbb{Q}$, this turns into an optimal transport which can be solved in $O(n \log n)$ in $d = 1$ and $O(n^3)$ for $d > 1$.

- This leads to the following estimator, usually approximated with stochastic optimisation:

$$\hat{\theta}_n := \arg\min_{\theta \in \Theta} \mathsf{W}(\mathbb{P}_\theta, Q_n)$$

Bassetti, F., Bodini, A., & Regazzini, E. (2006). On minimum Kantorovich distance estimators. *Statistics & Probability Letters*, *76*, 1298–1302.

Bernton, E., Jacob, P. E., Gerber, M., & Robert, C. P. (2017). Inference in generative models using the Wasserstein distance. *Information and Inference: A Journal of the IMA*, *8*(4), 657–676.

The maximum mean discrepancy

$$\text{MMD}(\mathbb{P}, \mathbb{Q}) := \sup_{f \in \mathcal{F}_{\text{MMD}}} \left| \mathbb{E}_{X \sim \mathbb{P}}[f(X)] - \mathbb{E}_{X \sim \mathbb{Q}}[f(X)] \right| \qquad \mathcal{F}_{\text{MMD}} := \{f : \mathcal{X} \to \mathbb{R} : \|f\|_{\mathcal{H}_k} \leq 1\}$$

The maximum mean discrepancy

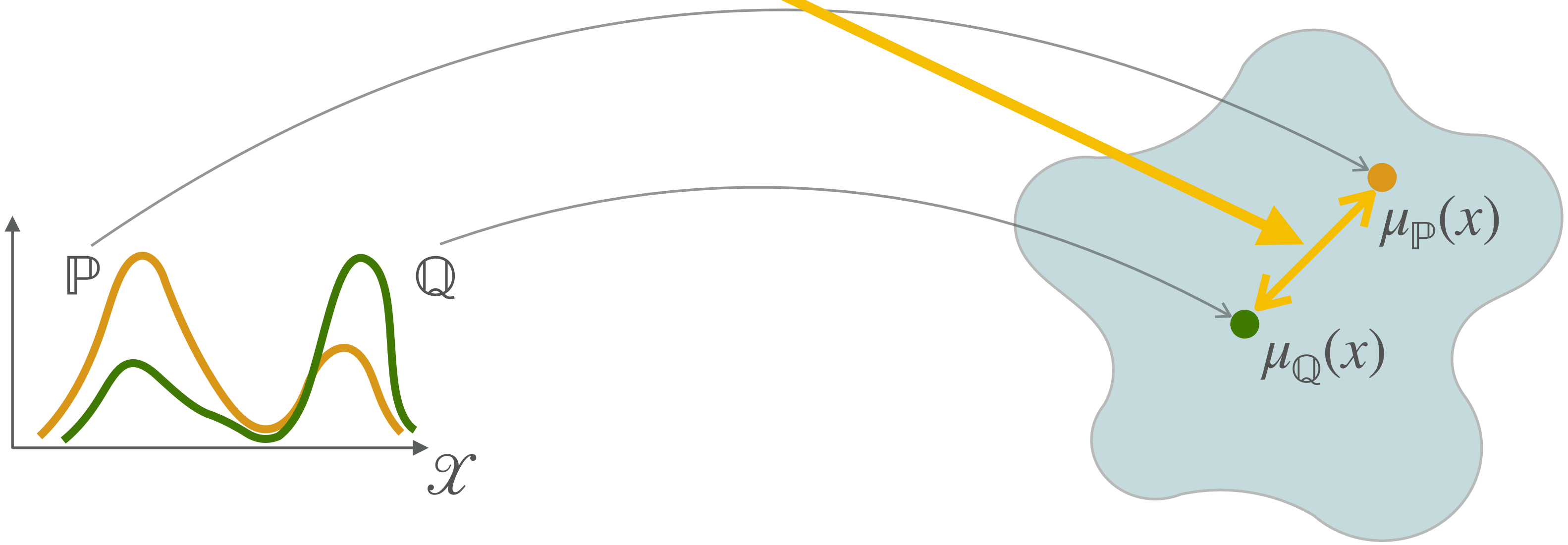
$$\text{MMD}(\mathbb{P}, \mathbb{Q}) := \sup_{f \in \mathcal{F}_{\text{MMD}}} \left| \mathbb{E}_{X \sim \mathbb{P}}[f(X)] - \mathbb{E}_{X \sim \mathbb{Q}}[f(X)] \right| \qquad \mathcal{F}_{\text{MMD}} := \{f : \mathcal{X} \to \mathbb{R} : \|f\|_{\mathcal{H}_k} \leq 1\}$$

$$= \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|_{\mathcal{H}_k} \qquad \mu_{\mathbb{P}}(x) = \int_{\mathcal{X}} k(x, x')\mathbb{P}(\mathrm{d}x')$$

The maximum mean discrepancy

$$\mathrm{MMD}(\mathbb{P},\mathbb{Q}) := \sup_{f\in\mathscr{F}_{\mathrm{MMD}}} \left| \mathbb{E}_{X\sim\mathbb{P}}[f(X)] - \mathbb{E}_{X\sim\mathbb{Q}}[f(X)] \right| \qquad \mathscr{F}_{\mathrm{MMD}} := \{f:\mathscr{X}\to\mathbb{R} : \|f\|_{\mathscr{H}_k} \leq 1\}$$

$$= \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|_{\mathscr{H}_k} \qquad \mu_{\mathbb{P}}(x) = \int_{\mathscr{X}} k(x,x')\mathbb{P}(\mathrm{d}x')$$



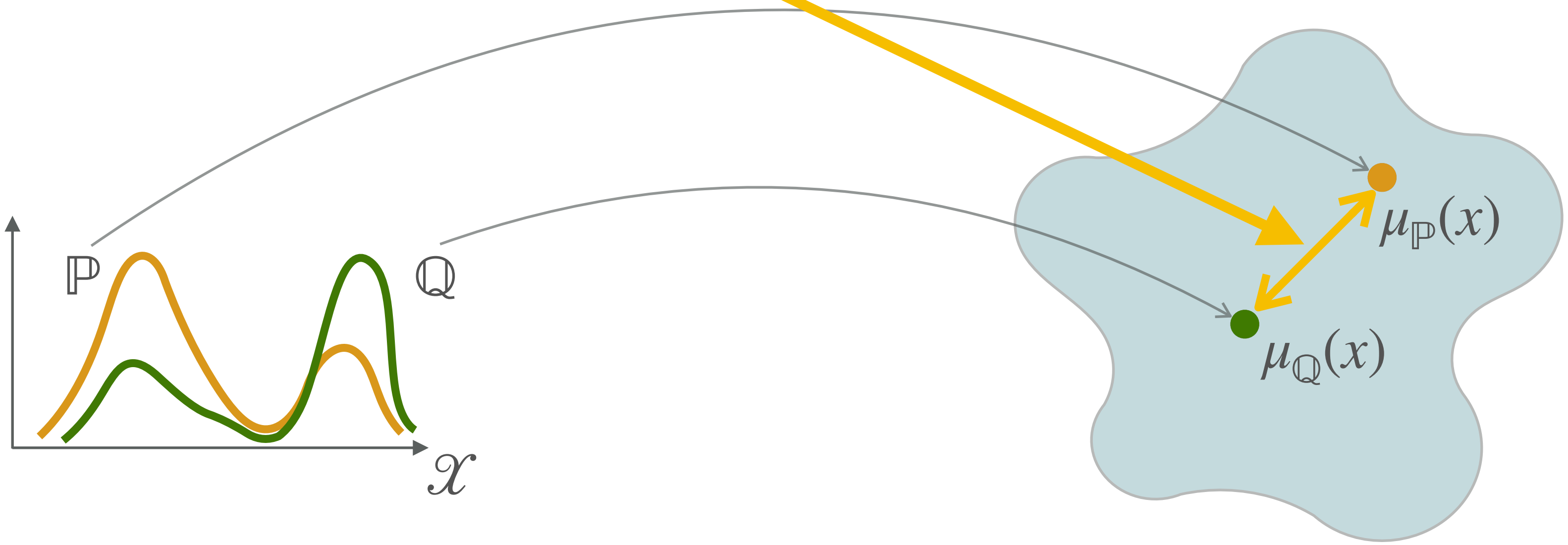
Credit for figure:

The maximum mean discrepancy

$$\mathrm{MMD}(\mathbb{P},\mathbb{Q}) := \sup_{f\in\mathcal{F}_{\mathrm{MMD}}} \left| \mathbb{E}_{X\sim\mathbb{P}}[f(X)] - \mathbb{E}_{X\sim\mathbb{Q}}[f(X)] \right| \qquad \mathcal{F}_{\mathrm{MMD}} := \{f : \mathcal{X} \to \mathbb{R} : \|f\|_{\mathcal{H}_k} \leq 1\}$$

$$= \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|_{\mathcal{H}_k} \qquad \mu_{\mathbb{P}}(x) = \int_{\mathcal{X}} k(x,x')\mathbb{P}(\mathrm{d}x')$$

(1) Divergence ✓

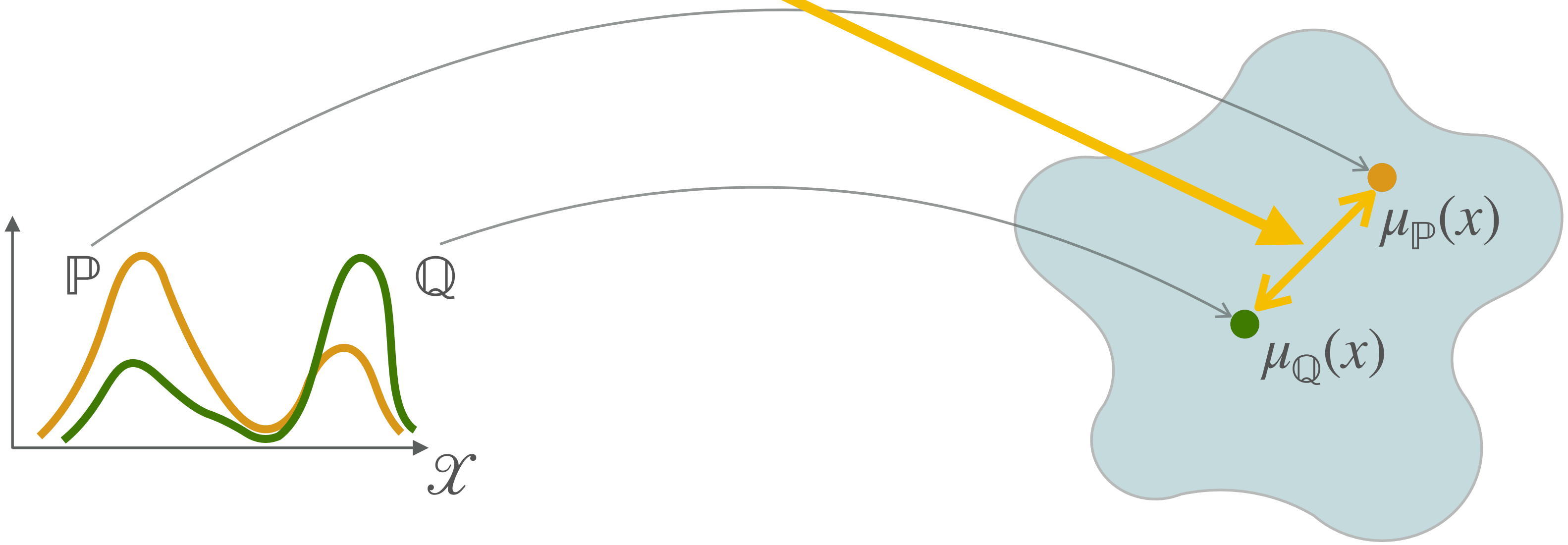


Credit for figure:

The maximum mean discrepancy

$$\mathrm{MMD}(\mathbb{P},\mathbb{Q}) := \sup_{f \in \mathcal{F}_{\mathrm{MMD}}} \left| \mathbb{E}_{X\sim\mathbb{P}}[f(X)] - \mathbb{E}_{X\sim\mathbb{Q}}[f(X)] \right| \qquad \mathcal{F}_{\mathrm{MMD}} := \{f : \mathcal{X} \to \mathbb{R} : \|f\|_{\mathcal{H}_k} \leq 1\}$$

$$= \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|_{\mathcal{H}_k} \qquad \mu_{\mathbb{P}}(x) = \int_{\mathcal{X}} k(x,x')\mathbb{P}(\mathrm{d}x')$$



(1) Divergence ✓

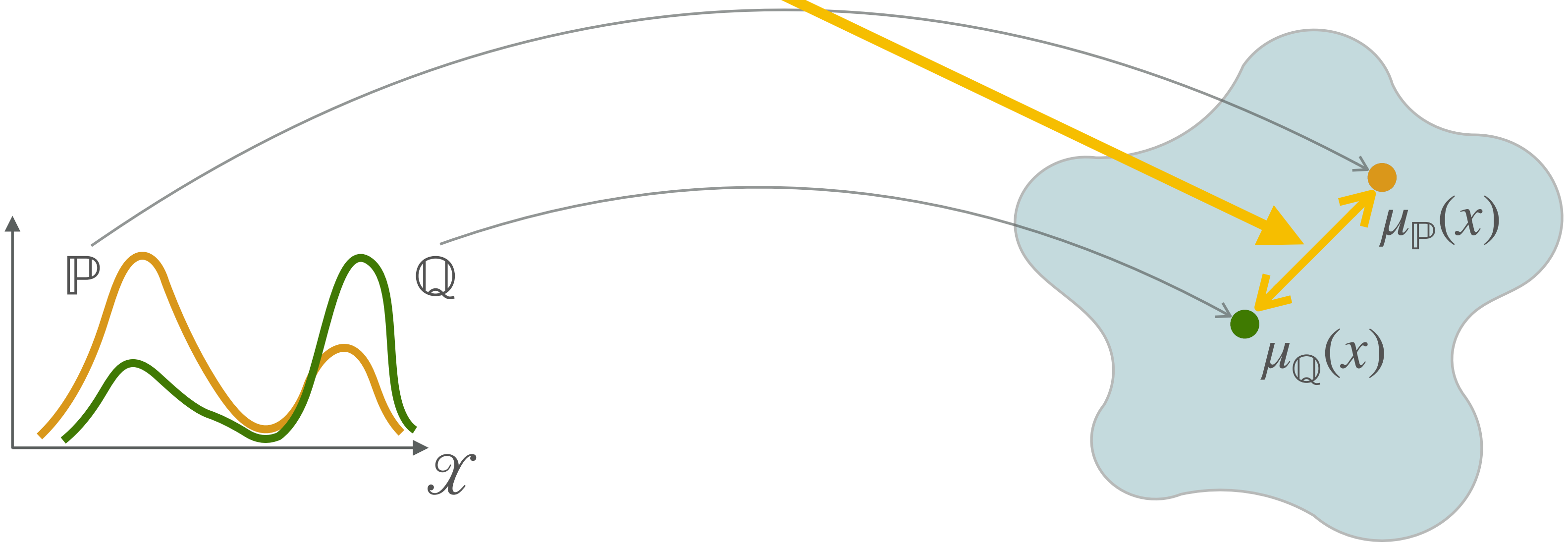
(2) Easy to estimate ✓

Credit for figure:

The maximum mean discrepancy

$$\mathrm{MMD}(\mathbb{P},\mathbb{Q}) := \sup_{f\in\mathscr{F}_{\mathrm{MMD}}} \left| \mathbb{E}_{X\sim\mathbb{P}}[f(X)] - \mathbb{E}_{X\sim\mathbb{Q}}[f(X)] \right| \qquad \mathscr{F}_{\mathrm{MMD}} := \{f : \mathscr{X} \to \mathbb{R} : \|f\|_{\mathscr{H}_k} \leq 1\}$$

$$= \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|_{\mathscr{H}_k} \qquad \mu_{\mathbb{P}}(x) = \int_{\mathscr{X}} k(x,x')\mathbb{P}(\mathrm{d}x')$$



(1) Divergence ✓

(2) Easy to estimate ✓

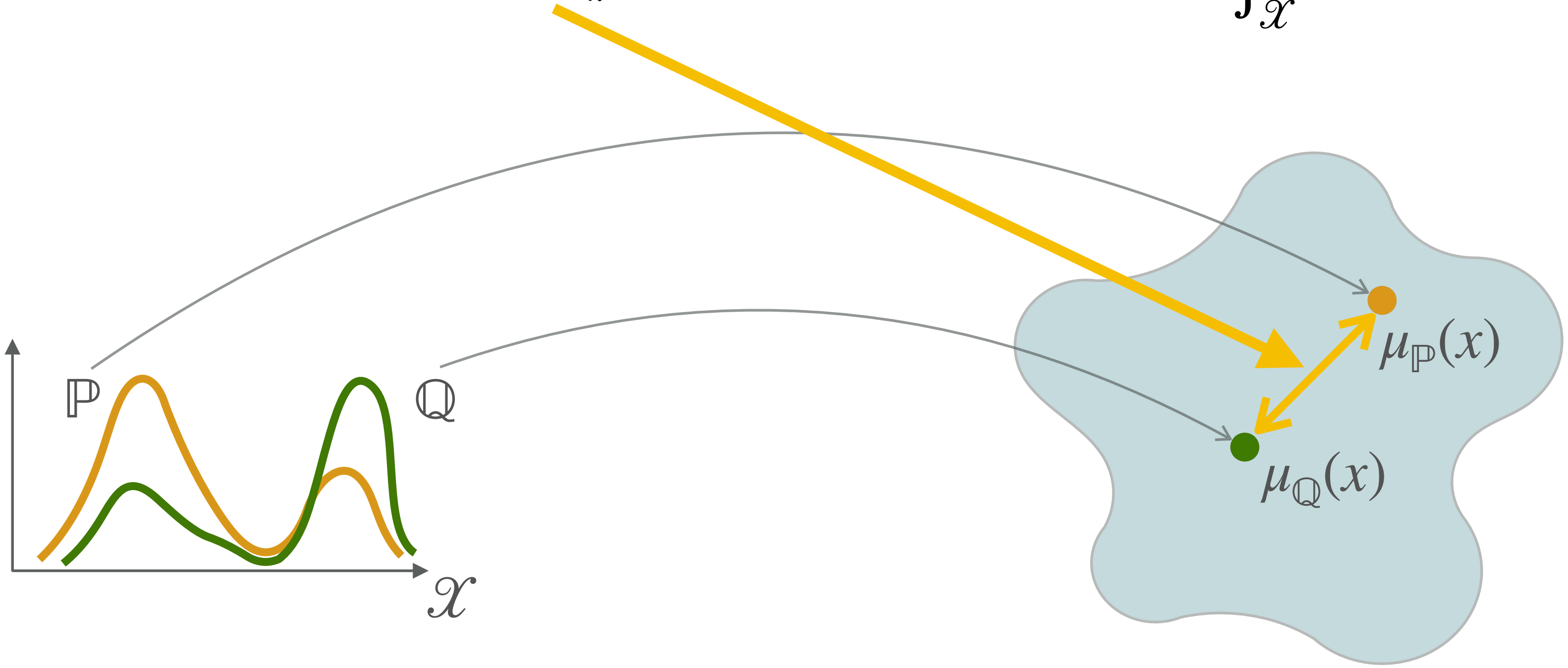
(3) Interpretable ~

Credit for figure:

The maximum mean discrepancy

$$\mathrm{MMD}(\mathbb{P},\mathbb{Q}) := \sup_{f \in \mathscr{F}_{\mathrm{MMD}}} \left| \mathbb{E}_{X\sim\mathbb{P}}[f(X)] - \mathbb{E}_{X\sim\mathbb{Q}}[f(X)] \right| \qquad \mathscr{F}_{\mathrm{MMD}} := \{f : \mathscr{X} \to \mathbb{R} : \|f\|_{\mathscr{H}_k} \leq 1\}$$

$$= \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|_{\mathscr{H}_k} \qquad \mu_{\mathbb{P}}(x) = \int_{\mathscr{X}} k(x,x')\mathbb{P}(\mathrm{d}x')$$



(1) Divergence

(2) Easy to estimate

(3) Interpretable

(4) Robust

Credit for figure:

Minimum MMD estimators

- Thanks to the ‘reproducing property’, we get:

$$\text{MMD}^2(\mathbb{P},\mathbb{Q}) = \int_{\mathcal{X}}\int_{\mathcal{X}} k(x,y)\mathbb{P}(dx)\mathbb{P}(dy) - 2\int_{\mathcal{X}}\int_{\mathcal{X}} k(x,y)\mathbb{P}(dx)\mathbb{Q}(dy) + \int_{\mathcal{X}}\int_{\mathcal{X}} k(x,y)\mathbb{Q}(dx)\mathbb{Q}(dy)$$

Minimum MMD estimators

- Thanks to the 'reproducing property', we get:

$$\text{MMD}^2(\mathbb{P}, \mathbb{Q}) = \int_{\mathcal{X}} \int_{\mathcal{X}} k(x, y)\mathbb{P}(dx)\mathbb{P}(dy) - 2 \int_{\mathcal{X}} \int_{\mathcal{X}} k(x, y)\mathbb{P}(dx)\mathbb{Q}(dy) + \int_{\mathcal{X}} \int_{\mathcal{X}} k(x, y)\mathbb{Q}(dx)\mathbb{Q}(dy)$$

- A natural estimator from sample consists of approximating the integrals with Monte Carlo!

Minimum MMD estimators

- Thanks to the 'reproducing property', we get:

$$\text{MMD}^2(\mathbb{P}, \mathbb{Q}) = \int_{\mathcal{X}} \int_{\mathcal{X}} k(x, y)\mathbb{P}(dx)\mathbb{P}(dy) - 2\int_{\mathcal{X}} \int_{\mathcal{X}} k(x, y)\mathbb{P}(dx)\mathbb{Q}(dy) + \int_{\mathcal{X}} \int_{\mathcal{X}} k(x, y)\mathbb{Q}(dx)\mathbb{Q}(dy)$$

- A natural estimator from sample consists of approximating the integrals with Monte Carlo!

- This leads to:

$$\hat{\theta}_n := \arg\min_{\theta \in \Theta} \text{MMD}^2(\mathbb{P}_\theta, Q_n)$$

Briol, F.-X., Barp, A., Duncan, A. B., & Girolami, M. (2019). Statistical inference for generative models with maximum mean discrepancy. *arXiv:1906.05944*.

Chérief-Abdellatif, B.-E., & Alquier, P. (2022). Finite sample properties of parametric MMD estimation: robustness to misspecification and dependence. *Bernoulli*, *28*(1), 181–213.

Any Questions?

Approximate Bayesian Computation

(From now on we will mostly be Bayesian!)

Approximate Bayesian computation (ABC)

- Recall that we would like to approximate:

$$p(\theta \mid y_1, \ldots, y_n) \propto \prod_{i=1}^{n} p(y_i \mid \theta) p(\theta)$$

Marin, J.-M., Pudlo, P., Robert, C. P., & Ryder, R. J. (2012). Approximate Bayesian computational methods. *Statistics and Computing*, *22*, 1167–1180.

Beaumont, M. A. (2019). Approximate Bayesian computation. *Annual Review of Statistics and Its Application*, *6*, 379–403.

Approximate Bayesian computation (ABC)

- Recall that we would like to approximate:

$$p(\theta \mid y_1) \propto p(y_1 \mid \theta) p(\theta)$$

(Only for notational simplicity)

Marin, J.-M., Pudlo, P., Robert, C. P., & Ryder, R. J. (2012). Approximate Bayesian computational methods. *Statistics and Computing*, *22*, 1167–1180.

Beaumont, M. A. (2019). Approximate Bayesian computation. *Annual Review of Statistics and Its Application*, *6*, 379–403.

Approximate Bayesian computation (ABC)

- Recall that we would like to approximate:

$$p(\theta \mid y_1) \propto p(y_1 \mid \theta) p(\theta)$$

(Only for notational simplicity)

- Now suppose we have a 'bump function'/'convolution kernel' K_ϵ, then we can define:

$$q_{\text{ABC}}(\theta \mid y_1) \propto \left[\int_{\mathcal{X}} K_\epsilon(\|x_1 - y_1\|) p(x_1 \mid \theta) dx_1 \right] p(\theta)$$

Marin, J.-M., Pudlo, P., Robert, C. P., & Ryder, R. J. (2012). Approximate Bayesian computational methods. *Statistics and Computing*, *22*, 1167–1180.

Beaumont, M. A. (2019). Approximate Bayesian computation. *Annual Review of Statistics and Its Application*, *6*, 379–403.

Approximate Bayesian computation (ABC)

- Recall that we would like to approximate:

$$p(\theta \mid y_1) \propto p(y_1 \mid \theta) p(\theta)$$

(Only for notational simplicity)

- Now suppose we have a 'bump function'/'convolution kernel' K_ϵ, then we can define:

$$q_{\mathrm{ABC}}(\theta \mid y_1) \propto \left[\underbrace{\int_{\mathcal{X}} K_\epsilon(\|x_1 - y_1\|) p(x_1 \mid \theta) dx_1}_{\text{Surrogate likelihood!}} \right] p(\theta)$$

Marin, J.-M., Pudlo, P., Robert, C. P., & Ryder, R. J. (2012). Approximate Bayesian computational methods. *Statistics and Computing*, *22*, 1167–1180.

Beaumont, M. A. (2019). Approximate Bayesian computation. *Annual Review of Statistics and Its Application*, *6*, 379–403.

Approximate Bayesian computation (ABC)

- Recall that we would like to approximate:

$$p(\theta \mid y_1) \propto p(y_1 \mid \theta) p(\theta)$$

(Only for notational simplicity)

- Now suppose we have a 'bump function'/'convolution kernel' K_ϵ, then we can define:

$$q_{\text{ABC}}(\theta \mid y_1) \propto \left[\int_{\mathcal{X}} K_\epsilon(\|x_1 - y_1\|) p(x_1 \mid \theta) dx_1 \right] p(\theta)$$

Marin, J.-M., Pudlo, P., Robert, C. P., & Ryder, R. J. (2012). Approximate Bayesian computational methods. *Statistics and Computing*, *22*, 1167–1180.

Beaumont, M. A. (2019). Approximate Bayesian computation. *Annual Review of Statistics and Its Application*, *6*, 379–403.

Approximate Bayesian computation (ABC)

- Recall that we would like to approximate:

$$p(\theta \mid y_1, \ldots, y_n) \propto \prod_{i=1}^{n} p(y_i \mid \theta) p(\theta)$$

- Now suppose we have a 'bump function'/'convolution kernel' K_ϵ, then we can define:

$$q_{\mathrm{ABC}}(\theta \mid y_1, \ldots, y_n) \propto \int_{\mathcal{X}} \ldots \int_{\mathcal{X}} K_\epsilon(\|x - y\|) \prod_{i=1}^{n} p(x_i \mid \theta) p(\theta) dx_1 \ldots dx_n$$

Marin, J.-M., Pudlo, P., Robert, C. P., & Ryder, R. J. (2012). Approximate Bayesian computational methods. *Statistics and Computing*, *22*, 1167–1180.

Beaumont, M. A. (2019). Approximate Bayesian computation. *Annual Review of Statistics and Its Application*, *6*, 379–403.

A simple ABC sampler

$$q_{\text{ABC}}(\theta \,|\, y_1, \ldots, y_n) \propto \int_{\mathcal{X}} \ldots \int_{\mathcal{X}} K_\epsilon(\|x - y\|) \prod_{i=1}^{n} p(x_i \,|\, \theta) p(\theta) dx_1 \ldots dx_n$$

A simple ABC sampler

$$q_{\mathrm{ABC}}(\theta \mid y_1, \ldots, y_n) \propto \int_{\mathcal{X}} \cdots \int_{\mathcal{X}} K_\epsilon(\|x - y\|) \prod_{i=1}^{n} p(x_i \mid \theta) p(\theta) dx_1 \ldots dx_n$$

This is still intractable though!!

A simple ABC sampler

$$q_{\text{ABC}}(\theta \mid y_1, \ldots, y_n) \propto \int_{\mathcal{X}} \ldots \int_{\mathcal{X}} K_\epsilon(\|x - y\|) \prod_{i=1}^{n} p(x_i \mid \theta) p(\theta) dx_1 \ldots dx_n$$

Sampler for the ABC posterior

A simple ABC sampler

$$q_{\text{ABC}}(\theta \mid y_1, \ldots, y_n) \propto \int_{\mathcal{X}} \cdots \int_{\mathcal{X}} K_\epsilon(\|x - y\|) \prod_{i=1}^{n} p(x_i \mid \theta) p(\theta) dx_1 \ldots dx_n$$

<u>Sampler for the ABC posterior</u>

- For $t \in \{1, \ldots, T\}$:

A simple ABC sampler

$$q_{\text{ABC}}(\theta \mid y_1, \ldots, y_n) \propto \int_{\mathcal{X}} \cdots \int_{\mathcal{X}} K_\epsilon(\|x - y\|) \prod_{i=1}^{n} p(x_i \mid \theta) p(\theta) dx_1 \ldots dx_n$$

Sampler for the ABC posterior

- For $t \in \{1, \ldots, T\}$:
    - Sample from the prior: $\theta_t \sim p(\theta)$.

A simple ABC sampler

$$q_{\text{ABC}}(\theta \mid y_1, \ldots, y_n) \propto \int_{\mathcal{X}} \ldots \int_{\mathcal{X}} K_\epsilon(\|x - y\|) \prod_{i=1}^{n} p(x_i \mid \theta) p(\theta) dx_1 \ldots dx_n$$

Sampler for the ABC posterior

- For $t \in \{1,\ldots,T\}$:
    - Sample from the prior: $\theta_t \sim p(\theta)$.
    - Simulate from the model: $x_{t1}, \ldots, x_{tn} \sim p(x \mid \theta_t)$.

A simple ABC sampler

$$q_{\text{ABC}}(\theta \mid y_1, \ldots, y_n) \propto \int_{\mathcal{X}} \cdots \int_{\mathcal{X}} K_\epsilon(\|x - y\|) \prod_{i=1}^{n} p(x_i \mid \theta) p(\theta) dx_1 \ldots dx_n$$

Sampler for the ABC posterior

- For $t \in \{1, \ldots, T\}$:
  - Sample from the prior: $\theta_t \sim p(\theta)$.
  - Simulate from the model: $x_{t1}, \ldots, x_{tn} \sim p(x \mid \theta_t)$.

We use the simulator:
$x_{ti} = G_{\theta_t}(u_i), u_i \sim \mathbb{U}$

A simple ABC sampler

$$q_{\mathrm{ABC}}(\theta \mid y_1, \ldots, y_n) \propto \int_{\mathcal{X}} \cdots \int_{\mathcal{X}} K_\epsilon(\|x - y\|) \prod_{i=1}^{n} p(x_i \mid \theta) p(\theta) dx_1 \ldots dx_n$$

Sampler for the ABC posterior

- For $t \in \{1, \ldots, T\}$:
    - Sample from the prior: $\theta_t \sim p(\theta)$.
    - Simulate from the model: $x_{t1}, \ldots, x_{tn} \sim p(x \mid \theta_t)$.
    - Weight θ_t with probability proportional to $K_\epsilon(\|x - y\|)$.

We use the simulator:
$x_{ti} = G_{\theta_t}(u_i), u_i \sim \mathbb{U}$

A simple ABC sampler

$$q_{\text{ABC}}(\theta \mid y_1, \ldots, y_n) \propto \int_{\mathcal{X}} \cdots \int_{\mathcal{X}} K_\epsilon(\|x - y\|) \prod_{i=1}^{n} p(x_i \mid \theta) p(\theta) dx_1 \ldots dx_n$$

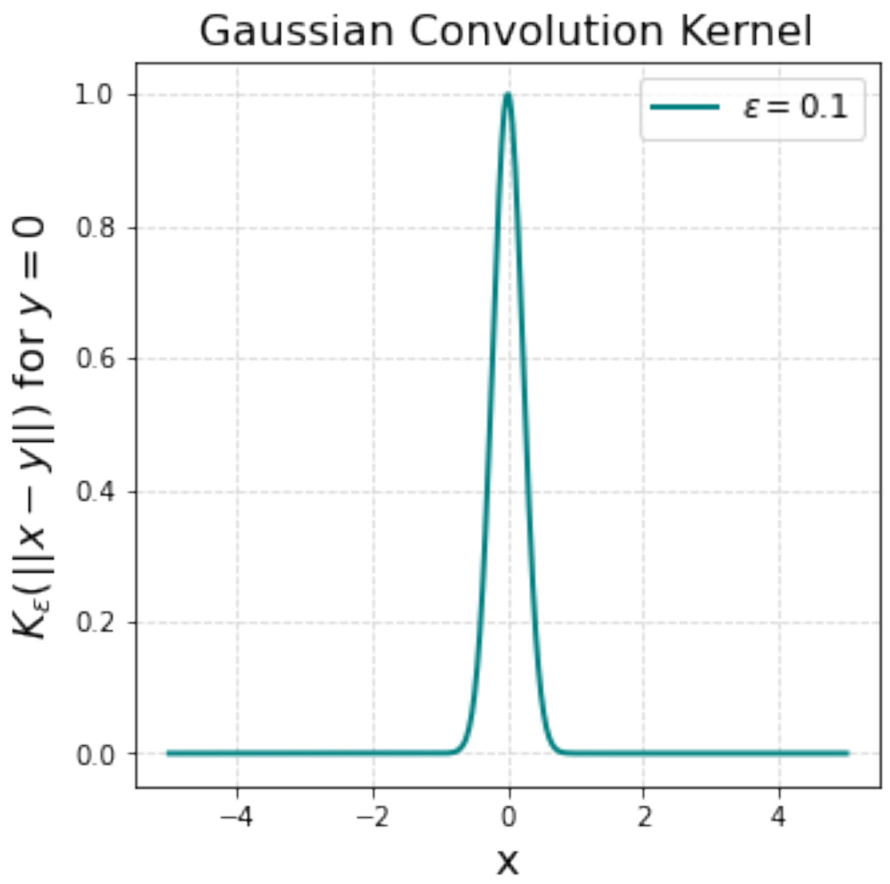
Sampler for the ABC posterior

- For $t \in \{1, \ldots, T\}$:
  - Sample from the prior: $\theta_t \sim p(\theta)$.
  - Simulate from the model: $x_{t1}, \ldots, x_{tn} \sim p(x \mid \theta_t)$.
  - Weight θ_t with probability proportional to $K_\epsilon(\|x - y\|)$.

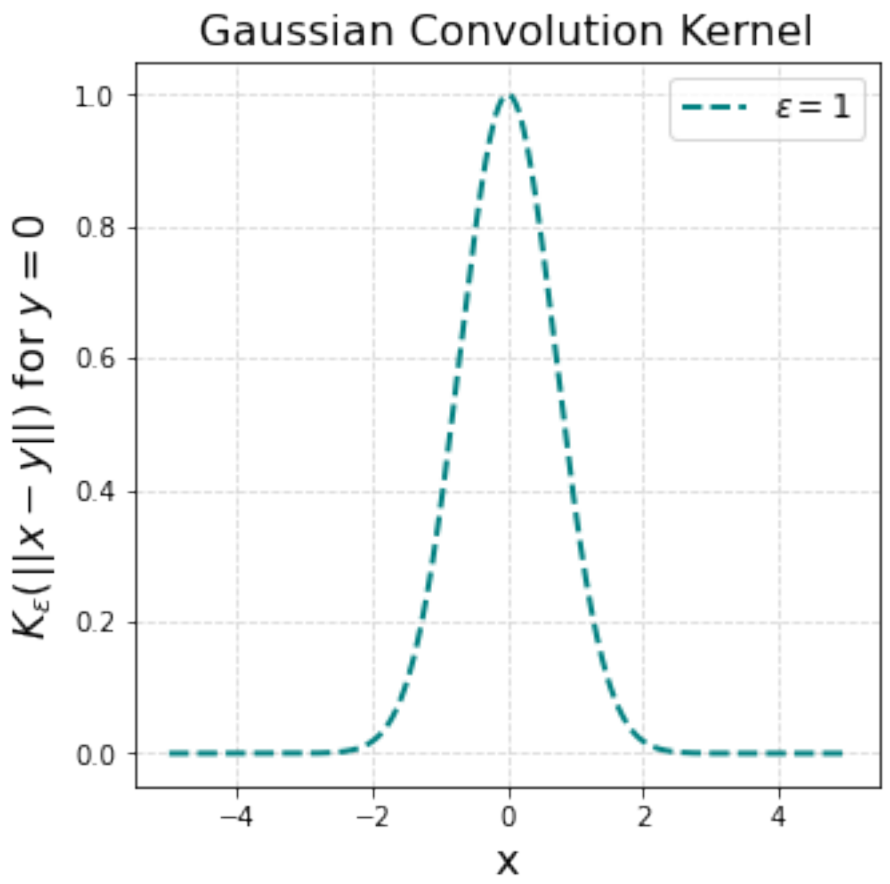
We use the simulator:
$x_{ti} = G_{\theta_t}(u_i), u_i \sim \mathbb{U}$

This is just a simple example; there are many more advanced sampling methods (e.g. SMC)

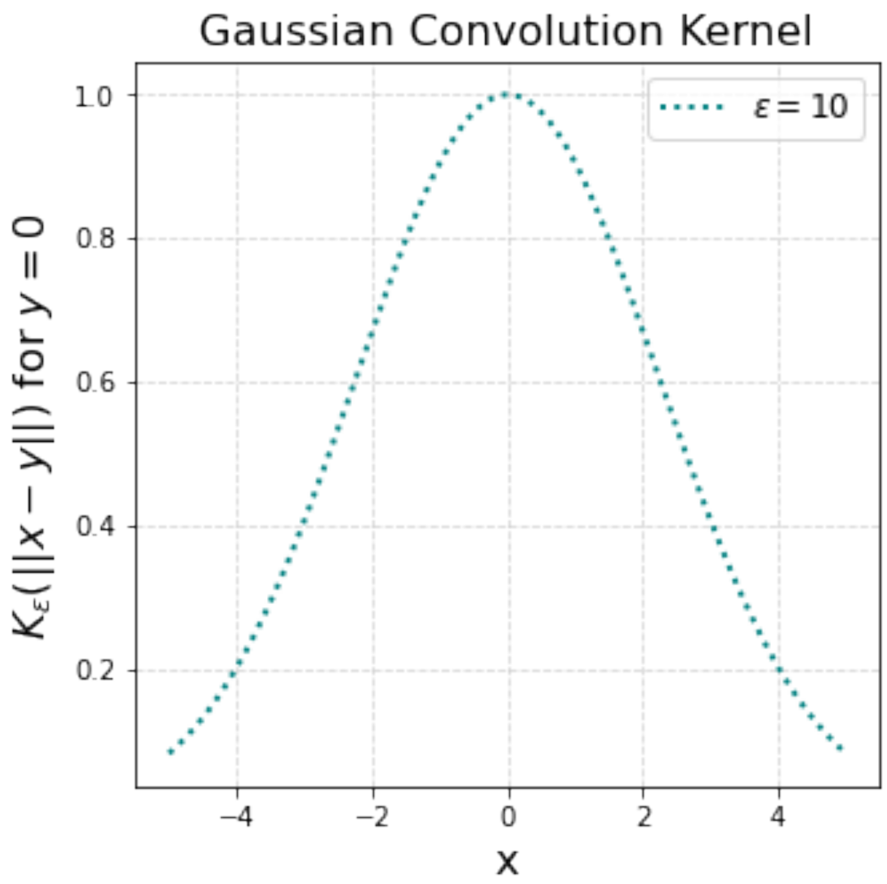
The impact of ϵ



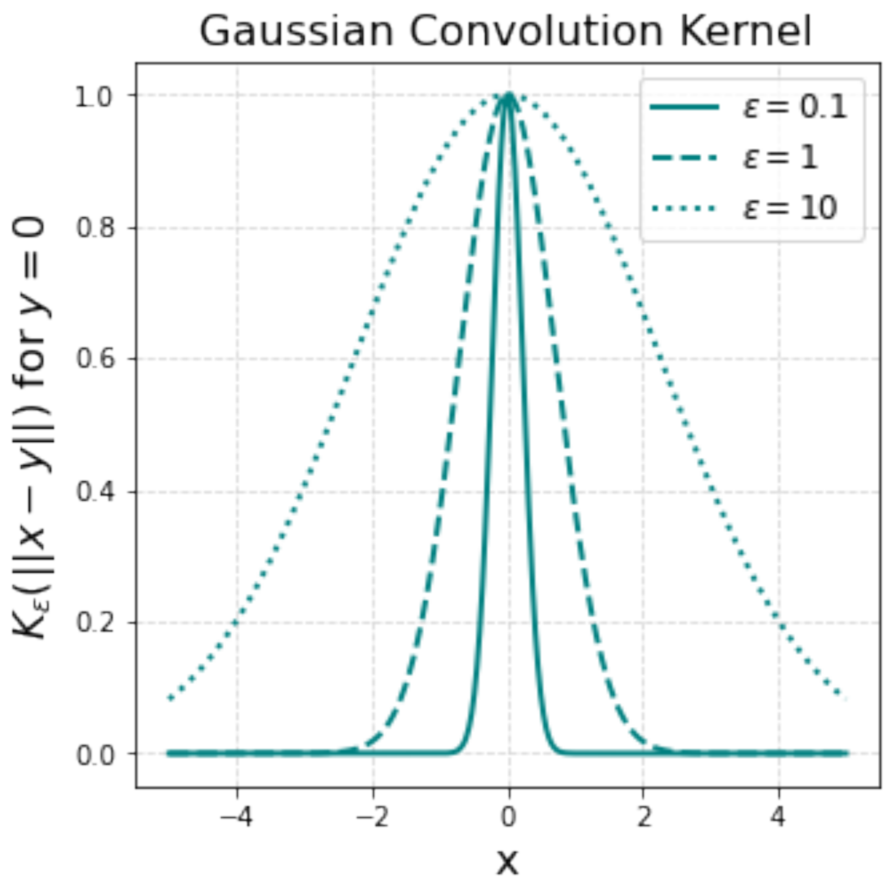
The impact of ϵ



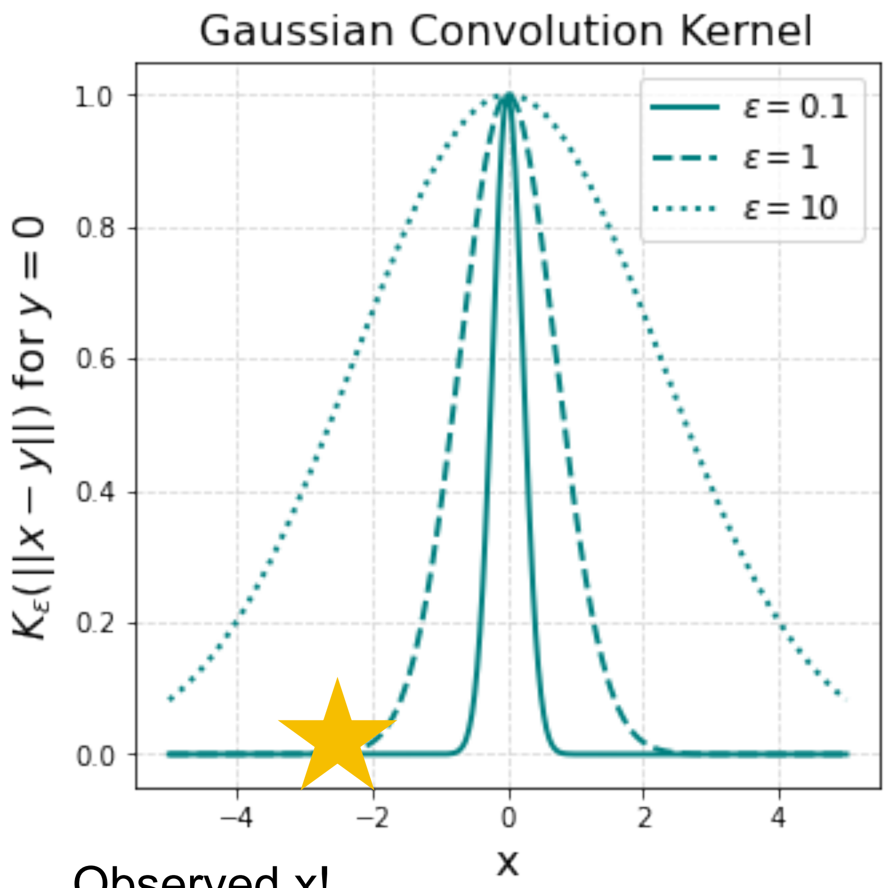
The impact of ϵ



The impact of ϵ



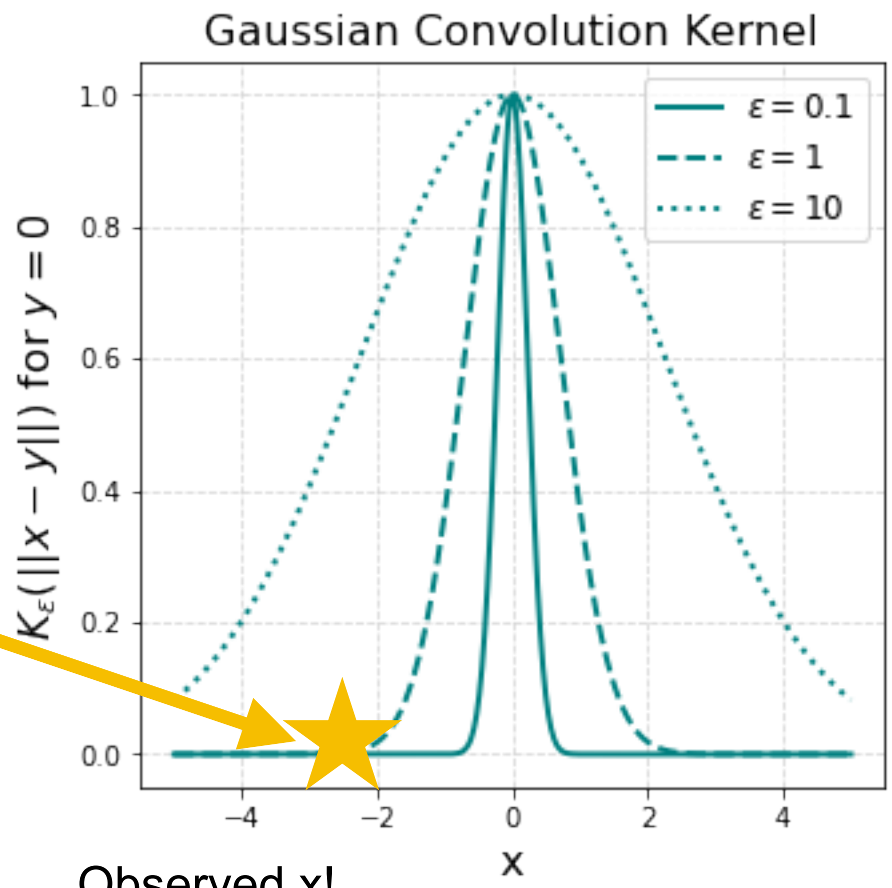
The impact of ϵ



Observed x!

The impact of ϵ

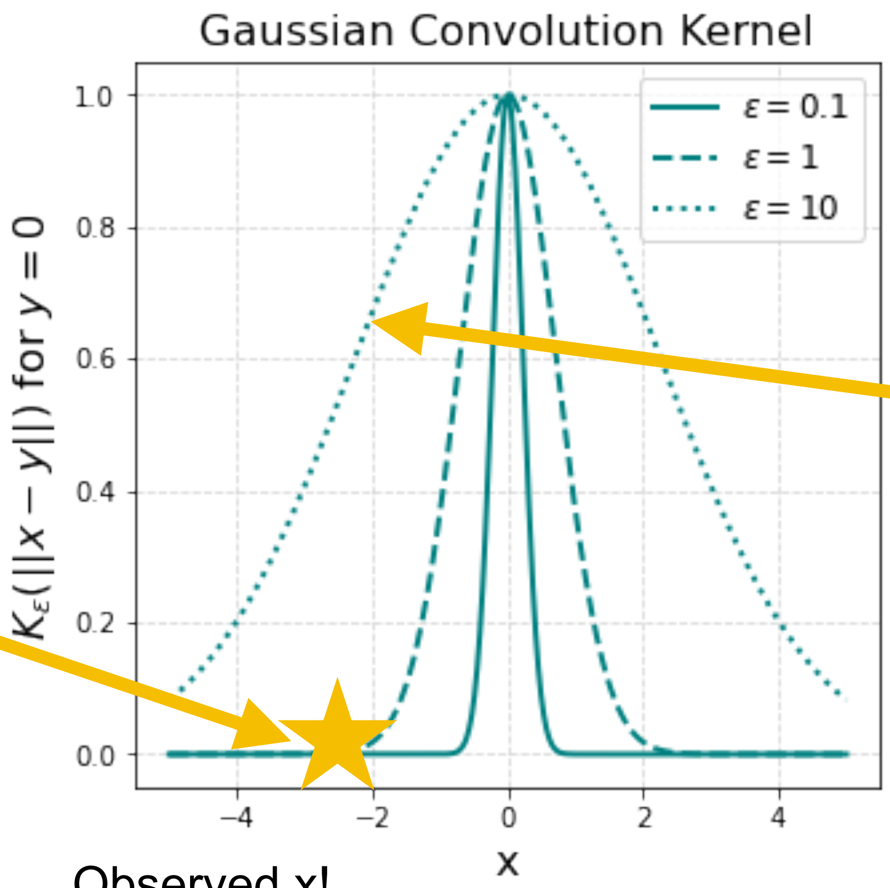
Essentially ignored
when $\epsilon = 0.1$ or $\epsilon = 1$



Observed x!

The impact of ϵ

Essentially ignored when $\epsilon = 0.1$ or $\epsilon = 1$



Observed x!

Considered close if $\epsilon = 10$

Discrepancies-based ABC

$$q_{\text{ABC}}(\theta \mid y_1, \ldots, y_n) \propto \int_{\mathcal{X}} \cdots \int_{\mathcal{X}} \cancel{K_\epsilon(\|x - y\|)} \prod_{i=1}^{n} p(x_i \mid \theta) p(\theta) dx_1 \ldots dx_n$$

$$K_\epsilon\left(D\left(\frac{1}{n}\sum_{i=1}^{n}\delta_{x_i}, \frac{1}{n}\sum_{i=1}^{n}\delta_{y_i}\right)\right) = K_\epsilon\left(D\left((\mathbb{P}_\theta)_n, \mathbb{Q}_n\right)\right)$$

Park, M., Jitkrittum, W., & Sejdinovic, D. (2016). K2-ABC: Approximate bayesian computation with kernel embeddings. *AISTATS*, *51*, 398–407.

Bernton, E., Jacob, P. E., Gerber, M., & Robert, C. P. (2019). Approximate Bayesian computation with the Wasserstein distance. *JRSSB*, *81*(2), 235–269.

Legramanti, S., Durante, D., & Alquier, P. (2025). Concentration and robustness of discrepancy-based ABC via Rademacher complexity. *The Annals of Statistics*, *53*(1), 37–60.

Any Questions?

ML approaches to SBI

We have now already covered the state-of-the-art until 2020-ish!

SBI with conditional density estimators

- I probably don't need to convince you that machine learning methods are very good at emulation…. How can we use this for Bayes?

$$p(\theta \mid y_1, \ldots, y_n) \propto \prod_{i=1}^{n} p(y_i \mid \theta) p(\theta)$$

Zammit-mangion, A., Sainsbury-Dale, M., & Huser, R. (2025). Neural methods for amortized parameter inference. *Annual Review of Statistics and Its Application*, *12*, 311–335.

Deistler, M., Boelts, J., Steinbach, P., Moss, G., Moreau, T., Gloeckler, M., Rodrigues, P. L. C., Linhart, J., Lappalainen, J. K., Miller, B. K., Gonçalves, P. J., Lueckmann, J.-M., Schröder, C., & Macke, J. H. (2025). Simulation-based inference: A practical guide. *arXiv:2508.12939*.

SBI with conditional density estimators

- I probably don't need to convince you that machine learning methods are very good at emulation…. How can we use this for Bayes?

$$p(\theta \mid y_1, \ldots, y_n) \propto \prod_{i=1}^{n} p(y_i \mid \theta) p(\theta)$$

Could emulate this?

Zammit-mangion, A., Sainsbury-Dale, M., & Huser, R. (2025). Neural methods for amortized parameter inference. *Annual Review of Statistics and Its Application*, *12*, 311–335.

Deistler, M., Boelts, J., Steinbach, P., Moss, G., Moreau, T., Gloeckler, M., Rodrigues, P. L. C., Linhart, J., Lappalainen, J. K., Miller, B. K., Gonçalves, P. J., Lueckmann, J.-M., Schröder, C., & Macke, J. H. (2025). Simulation-based inference: A practical guide. *arXiv:2508.12939*.

SBI with conditional density estimators

- I probably don't need to convince you that machine learning methods are very good at emulation…. How can we use this for Bayes?

$$p(\theta \mid y_1, \ldots, y_n) \propto \prod_{i=1}^{n} p(y_i \mid \theta) p(\theta)$$

Could emulate this?

Could emulate this?

Zammit-mangion, A., Sainsbury-Dale, M., & Huser, R. (2025). Neural methods for amortized parameter inference. *Annual Review of Statistics and Its Application*, *12*, 311–335.

Deistler, M., Boelts, J., Steinbach, P., Moss, G., Moreau, T., Gloeckler, M., Rodrigues, P. L. C., Linhart, J., Lappalainen, J. K., Miller, B. K., Gonçalves, P. J., Lueckmann, J.-M., Schröder, C., & Macke, J. H. (2025). Simulation-based inference: A practical guide. *arXiv:2508.12939*.

SBI with conditional density estimators

- I probably don't need to convince you that machine learning methods are very good at emulation…. How can we use this for Bayes?

$$p(\theta \mid y_1, \ldots, y_n) \propto \prod_{i=1}^{n} p(y_i \mid \theta) p(\theta)$$

- Both are conditional densities, and so we need to think about how we can use the 'power' of machine learning to emulate this type of quantity.

Zammit-mangion, A., Sainsbury-Dale, M., & Huser, R. (2025). Neural methods for amortized parameter inference. *Annual Review of Statistics and Its Application*, *12*, 311–335.

Deistler, M., Boelts, J., Steinbach, P., Moss, G., Moreau, T., Gloeckler, M., Rodrigues, P. L. C., Linhart, J., Lappalainen, J. K., Miller, B. K., Gonçalves, P. J., Lueckmann, J.-M., Schröder, C., & Macke, J. H. (2025). Simulation-based inference: A practical guide. *arXiv:2508.12939*.

SBI with conditional density estimators

- I probably don't need to convince you that machine learning methods are very good at emulation…. How can we use this for Bayes?

$$p(\theta \mid y_1, \ldots, y_n) \propto \prod_{i=1}^{n} p(y_i \mid \theta) p(\theta)$$

- Both are conditional densities, and so we need to think about how we can use the 'power' of machine learning to emulate this type of quantity.

- We will start by emulating the likelihood; i.e. we want a flexible class: $\{q_\phi(x \mid \theta)\}_{\Phi \in \Phi}$

Zammit-mangion, A., Sainsbury-Dale, M., & Huser, R. (2025). Neural methods for amortized parameter inference. *Annual Review of Statistics and Its Application*, *12*, 311–335.

Deistler, M., Boelts, J., Steinbach, P., Moss, G., Moreau, T., Gloeckler, M., Rodrigues, P. L. C., Linhart, J., Lappalainen, J. K., Miller, B. K., Gonçalves, P. J., Lueckmann, J.-M., Schröder, C., & Macke, J. H. (2025). Simulation-based inference: A practical guide. *arXiv:2508.12939*.

Some simpler models…

$$\{q_\phi(x \mid \theta)\}_{\Phi \in \Phi}$$

Some simpler models…

$$\{q_\phi(x \mid \theta)\}_{\Phi \in \Phi}$$

- We could start with the statistician's favourite model:

$$q_\phi(x \mid \theta) = \mathcal{N}(x \mid \mu(\phi; \theta), \Sigma(\phi; \theta))$$

Some simpler models…

$$\{q_\phi(x\,|\,\theta)\}_{\Phi\in\Phi}$$

Depends on the conditioning variable

- We could start with the statistician's favourite model:

$$q_\phi(x\,|\,\theta) = \mathcal{N}(x\,|\,\mu(\phi;\theta), \Sigma(\phi;\theta))$$

Some simpler models…

Depends on the parameters of the model

Depends on the conditioning variable

$$\{q_\phi(x \mid \theta)\}_{\Phi \in \Phi}$$

- We could start with the statistician's favourite model:

$$q_\phi(x \mid \theta) = \mathcal{N}(x \mid \mu(\phi; \theta), \Sigma(\phi; \theta))$$

Some simpler models…

$$\{q_\phi(x\,|\,\theta)\}_{\Phi\in\Phi}$$

- We could start with the statistician's favourite model:

$$q_\phi(x\,|\,\theta) = \mathcal{N}(x\,|\,\mu(\phi;\theta), \Sigma(\phi;\theta))$$

- We can increase the flexibility:

$$q_\phi(x\,|\,\theta) = \sum_{c=1}^{C} w_c(\phi;\theta)\mathcal{N}(x\,|\,\mu_c(\phi;\theta), \Sigma_c(\phi;\theta))$$

Bishop, C. M. (1994). *Mixture density networks* (NCRG/94/004).

Transformations and densities

- Consider some base distribution p_v and some transformation T such that

$$x = T(v), \quad v \sim p_v(v)$$

Transformations and densities

- Consider some base distribution p_v and some transformation T such that

$$x = T(v), \quad v \sim p_v(v)$$

- Suppose T is invertible and both T and T^{-1} are differentiable. Then:

$$p_x(x) = p_v(v) \left| \text{det} J_T(x) \right|^{-1}$$

Transformations and densities

- Consider some base distribution p_v and some transformation T such that

$$x = T(v), \quad v \sim p_v(v)$$

- Suppose T is invertible and both T and T^{-1} are differentiable. Then:

$$p_x(x) = p_v(v) \left| \det J_T(x) \right|^{-1}$$

$$J_T(v) := \begin{bmatrix} \frac{\partial T_1}{\partial v_1} & \cdots & \frac{\partial T_1}{\partial v_d} \\ \vdots & \ddots & \vdots \\ \frac{\partial T_d}{\partial v_1} & \cdots & \frac{\partial T_d}{\partial v_d} \end{bmatrix}$$

Transformations and densities

- Consider some base distribution p_v and some transformation T such that

$$x = T(v), \quad v \sim p_v(v)$$

- Suppose T is invertible and both T and T^{-1} are differentiable. Then:

$$p_x(x) = p_v(v)\left|\det J_T(x)\right|^{-1} = p_v(T^{-1}(x))\left|\det J_{T^{-1}}(x)\right|$$

$$J_T(v) := \begin{bmatrix} \frac{\partial T_1}{\partial v_1} & \cdots & \frac{\partial T_1}{\partial v_d} \\ \vdots & \ddots & \vdots \\ \frac{\partial T_d}{\partial v_1} & \cdots & \frac{\partial T_d}{\partial v_d} \end{bmatrix}$$

Transformations and densities

- Consider some base distribution p_v and some transformation T such that

$$x = T(v), \quad v \sim p_v(v)$$

- Suppose T is invertible and both T and T^{-1} are differentiable. Then:

$$p_x(x) = p_v(v) \left| \det J_T(x) \right|^{-1} = p_v(T^{-1}(x)) \left| \det J_{T^{-1}}(x) \right|$$

- How do we design T if we want the density model to be very flexible?

Transformations and densities

- Consider some base distribution p_v and some transformation T such that

$$x = T(v), \quad v \sim p_v(v)$$

- Suppose T is invertible and both T and T^{-1} are differentiable. Then:

$$p_x(x) = p_v(v)\left|\det J_T(x)\right|^{-1} = p_v(T^{-1}(x))\left|\det J_{T^{-1}}(x)\right|$$

- How do we design T if we want the density model to be very flexible?

→ Use neural networks!!

Normalising flows (I)

- Note that we can compose such maps and keep their desirable properties:

$$T = T^K \circ \ldots \circ T^2 \circ T^1$$

Normalising flows (I)

- Note that we can compose such maps and keep their desirable properties:

$$T_\phi = T_\phi^K \circ \ldots \circ T_\phi^2 \circ T_\phi^1$$

We can also parametrise them!

Normalising flows (I)

- Note that we can compose such maps and keep their desirable properties:

$$T_\phi = T_\phi^K \circ \ldots \circ T_\phi^2 \circ T_\phi^1$$

- We end up with a normalising flow:

$$q_\phi(x) = p_v(v) \left| \mathrm{det} J_{T_\phi}(x) \right|^{-1}$$

Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., & Lakshminarayanan, B. (2021). Normalizing flows for probabilistic modeling and inference. *JMLR*, *22*, 1–64.

Kobyzev, I., Prince, S. J. D., & Brubaker, M. A. (2021). Normalizing flows: An introduction and review of current methods. *IEEE TPAMI*, *43*(11), 3964–3979.

Normalising flows (I)

- Note that we can compose such maps and keep their desirable properties:

$$T_{\phi,\theta} = T^K_{\phi,\theta} \circ \ldots \circ T^2_{\phi,\theta} \circ T^1_{\phi,\theta}$$

Straightforward to create conditional density!

- We end up with a normalising flow:

$$q_\phi(x\,|\,\theta) = p_v(v)\left|\mathrm{det} J_{T_{\phi,\theta}}(x)\right|^{-1}$$

Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., & Lakshminarayanan, B. (2021). Normalizing flows for probabilistic modeling and inference. *JMLR*, *22*, 1–64.

Kobyzev, I., Prince, S. J. D., & Brubaker, M. A. (2021). Normalizing flows: An introduction and review of current methods. *IEEE TPAMI*, *43*(11), 3964–3979.

Normalising flows (II)

$$q_\phi(x \mid \theta) = p_v(v) \left| \mathrm{det} J_{T_{\phi,\theta}}(x) \right|^{-1} \qquad T_{\phi,\theta} = T^K_{\phi,\theta} \circ \ldots \circ T^2_{\phi,\theta} \circ T^1_{\phi,\theta}$$

Normalising flows (II)

$$q_\phi(x\,|\,\theta) = p_v(v)\left|\text{det}J_{T_{\phi,\theta}}(x)\right|^{-1} \qquad T_{\phi,\theta} = T^K_{\phi,\theta} \circ \ldots \circ T^2_{\phi,\theta} \circ T^1_{\phi,\theta}$$

- $T^1_{\phi,\theta}, \ldots, T^K_{\phi,\theta}$ are selected to make $q_\phi(x\,|\,\theta)$ **tractable**, and for det$J_{T_{\phi,\theta}}(x)$ to be computed efficiently.

Normalising flows (II)

$$q_\phi(x\,|\,\theta) = p_v(v)\left|\text{det}J_{T_{\phi,\theta}}(x)\right|^{-1} \qquad T_{\phi,\theta} = T^K_{\phi,\theta} \circ \ldots \circ T^2_{\phi,\theta} \circ T^1_{\phi,\theta}$$

- $T^1_{\phi,\theta}, \ldots, T^K_{\phi,\theta}$ are selected to make $q_\phi(x\,|\,\theta)$ **tractable**, and for $\text{det}J_{T_{\phi,\theta}}(x)$ to be computed efficiently.

- We typically train the network (i.e. find a good ϕ) by **minimising the forward KL divergence**.

Normalising flows (II)

$$q_\phi(x\,|\,\theta) = p_v(v)\left|\mathrm{det}J_{T_{\phi,\theta}}(x)\right|^{-1} \qquad T_{\phi,\theta} = T^K_{\phi,\theta} \circ \ldots \circ T^2_{\phi,\theta} \circ T^1_{\phi,\theta}$$

- $T^1_{\phi,\theta}, \ldots, T^K_{\phi,\theta}$ are selected to make $q_\phi(x\,|\,\theta)$ **tractable**, and for $\mathrm{det}J_{T_{\phi,\theta}}(x)$ to be computed efficiently.

- We typically train the network (i.e. find a good ϕ) by **minimising the forward KL divergence**.

- Terminology: Are normalising flows simulators?

Normalising flows (II)

$$q_\phi(x \mid \theta) = p_v(v) \left| \det J_{T_{\phi,\theta}}(x) \right|^{-1} \qquad T_{\phi,\theta} = T^K_{\phi,\theta} \circ \ldots \circ T^2_{\phi,\theta} \circ T^1_{\phi,\theta}$$

- $T^1_{\phi,\theta}, \ldots, T^K_{\phi,\theta}$ are selected to make $q_\phi(x \mid \theta)$ **tractable**, and for $\det J_{T_{\phi,\theta}}(x)$ to be computed efficiently.

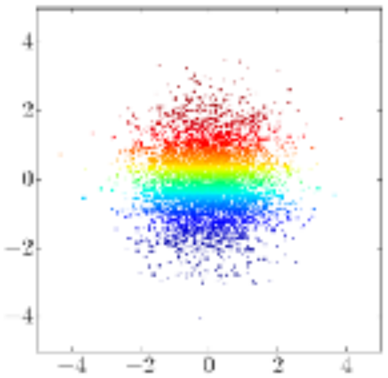
- We typically train the network (i.e. find a good ϕ) by **minimising the forward KL divergence**.

- Terminology: Are normalising flows simulators?

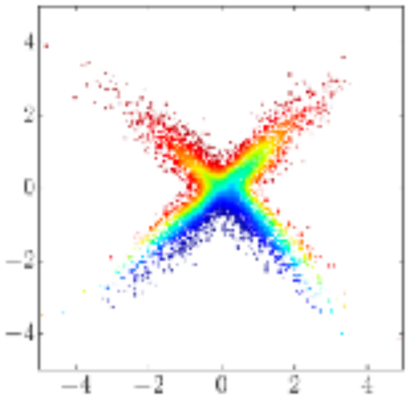
They can be, but (similarly to diffusion models) they do not typically encode any science, they are just constructed to be very flexible models!

Normalising flows (III)

$p_v(v)$



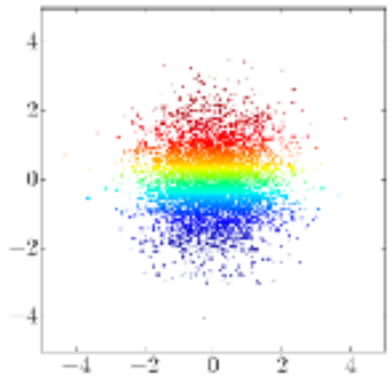
$q_\phi(x)$



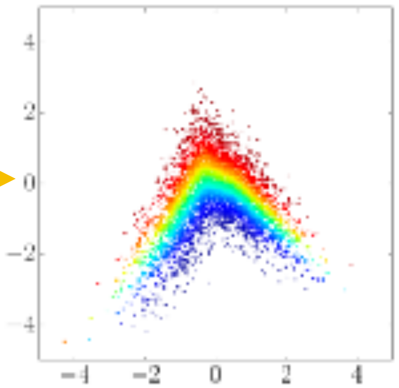
Plots borrowed from: Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., & Lakshminarayanan, B. (2021). Normalizing flows for probabilistic modeling and inference. *JMLR*, *22*, 1–64.

Normalising flows (III)

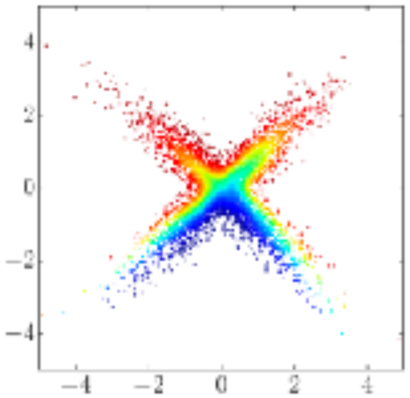
$p_v(v)$



T_ϕ^1

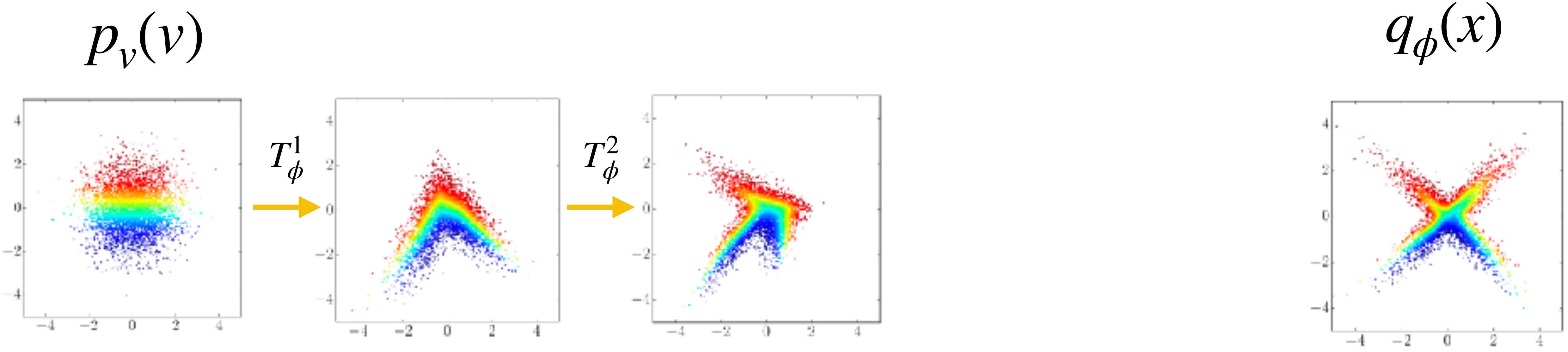


$q_\phi(x)$



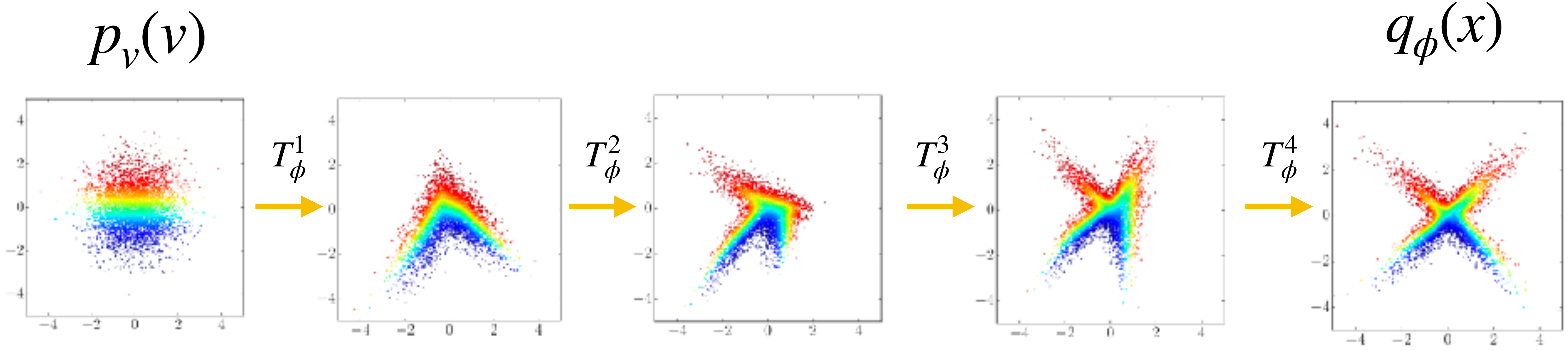
Plots borrowed from: Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., & Lakshminarayanan, B. (2021). Normalizing flows for probabilistic modeling and inference. *JMLR*, *22*, 1–64.

Normalising flows (III)



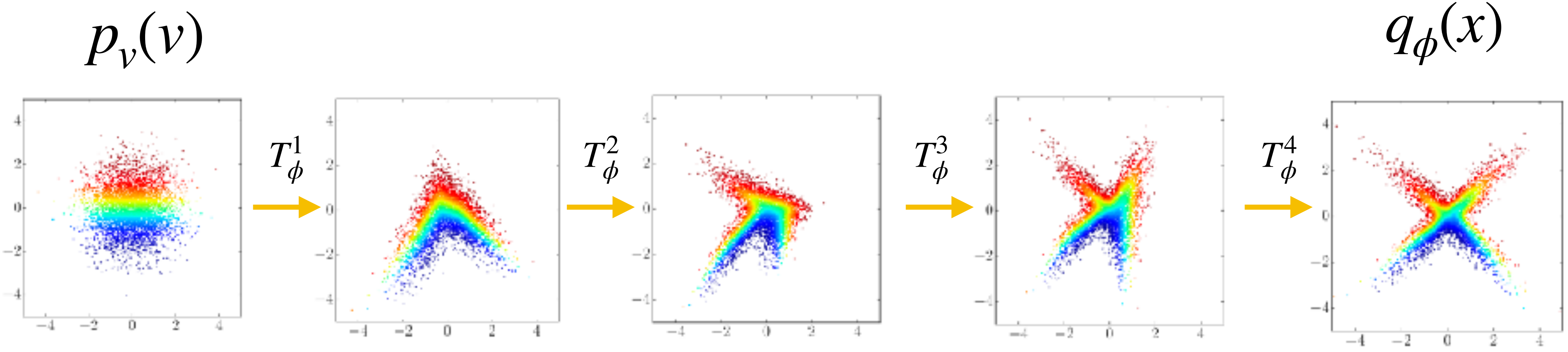
Plots borrowed from: Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., & Lakshminarayanan, B. (2021). Normalizing flows for probabilistic modeling and inference. *JMLR*, *22*, 1–64.

Normalising flows (III)



Plots borrowed from: Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., & Lakshminarayanan, B. (2021). Normalizing flows for probabilistic modeling and inference. *JMLR*, *22*, 1–64.

Normalising flows (III)



The composition of relatively simple transformations can give fairly complex maps!

Plots borrowed from: Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., & Lakshminarayanan, B. (2021). Normalizing flows for probabilistic modeling and inference. *JMLR*, *22*, 1–64.

Neural likelihood estimation (NLE)

- **Step 1:** train $q_\phi(x \mid \theta)$ to approximate the likelihood using samples from the prior ($\theta_1, \ldots, \theta_n \sim p(\theta)$) and simulator ($x_i \sim p(\cdot \mid \theta_i)$):

$$\hat{\phi}_n := \arg\min_{\phi \in \Phi} \ell_{\mathrm{NLE}}(\phi), \quad \ell_{\mathrm{NLE}}(\phi) = -\frac{1}{n}\sum_{i=1}^{n} \log q_\phi(x_i \mid \theta_i) \approx -\mathbb{E}_{\theta \sim p(\theta)}[\mathbb{E}_{x \sim p(\cdot \mid \theta)}[\log q_\phi(x \mid \theta)]]$$

Papamakarios, G., Sterratt, D. C., & Murray, I. (2019). Sequential neural likelihood: Fast likelihood-free inference with autoregressive flows. *AISTATS*, 837–848.

Neural likelihood estimation (NLE)

- **Step 1:** train $q_\phi(x \mid \theta)$ to approximate the likelihood using samples from the prior ($\theta_1, \ldots, \theta_n \sim p(\theta)$) and simulator ($x_i \sim p(\cdot \mid \theta_i)$):

$$\hat{\phi}_n := \arg\min_{\phi \in \Phi} \ell_{\text{NLE}}(\phi), \quad \ell_{\text{NLE}}(\phi) = -\frac{1}{n}\sum_{i=1}^{n} \log q_\phi(x_i \mid \theta_i) \approx -\mathbb{E}_{\theta \sim p(\theta)}[\mathbb{E}_{x \sim p(\cdot \mid \theta)}[\log q_\phi(x \mid \theta)]]$$

- **Step 2:** Approximate posterior (MCMC, VI) constructed with surrogate likelihood!

$$p_{\text{NLE}}(\theta \mid y_1, \ldots, y_n) \propto \prod_{i=1}^{n} q_{\hat{\phi}_n}(y_i \mid \theta) p(\theta)$$

Papamakarios, G., Sterratt, D. C., & Murray, I. (2019). Sequential neural likelihood: Fast likelihood-free inference with autoregressive flows. *AISTATS*, 837–848.

Amortisation for NLE

- Recall the NLE posterior:

$$p_{\text{NLE}}(\theta \mid y_1, \ldots, y_n) \propto \prod_{i=1}^{n} q_{\hat{\phi}_n}(y_i \mid \theta) p(\theta)$$

Amortisation for NLE

- Recall the NLE posterior:

$$p_{\mathrm{NLE}}(\theta \mid y_1, \ldots, y_n) \propto \prod_{i=1}^{n} q_{\hat{\phi}_n}(y_i \mid \theta) p(\theta)$$

- What if we get some new observations $\tilde{y}_1, \ldots, \tilde{y}_n$?

→ We already have an emulator of the likelihood, so we just need to use it!

Amortisation for NLE

- Recall the NLE posterior:

$$p_{\text{NLE}}(\theta \,|\, y_1, \ldots, y_n) \propto \prod_{i=1}^{n} q_{\hat{\phi}_n}(y_i \,|\, \theta) p(\theta)$$

- What if we get some new observations $\tilde{y}_1, \ldots, \tilde{y}_n$?

→ We already have an emulator of the likelihood, so we just need to use it!

$$p_{\text{NLE}}(\theta \,|\, \tilde{y}_1, \ldots, \tilde{y}_n) \propto \prod_{i=1}^{n} q_{\hat{\phi}_n}(\tilde{y}_i \,|\, \theta) p(\theta)$$

Amortisation for NLE

- Recall the NLE posterior:

$$p_{\mathrm{NLE}}(\theta \mid y_1, \ldots, y_n) \propto \prod_{i=1}^{n} q_{\hat{\phi}_n}(y_i \mid \theta) p(\theta)$$

- What if we get some new observations $\tilde{y}_1, \ldots, \tilde{y}_n$?

→ We already have an emulator of the likelihood, so we just need to use it!

$$p_{\mathrm{NLE}}(\theta \mid \tilde{y}_1, \ldots, \tilde{y}_n) \propto \prod_{i=1}^{n} q_{\hat{\phi}_n}(\tilde{y}_i \mid \theta) p(\theta)$$

We still need to re-run MCMC/VI though… We are **partially amortised**.

Neural posterior estimation (NPE)

- **Step 1:** train $q_\phi(\theta \mid x)$ to approximate the posterior using samples from the prior ($\theta_1, \ldots, \theta_n \sim p(\theta)$) and simulator ($x_i \sim p(\cdot \mid \theta_i)$):

$$\hat{\phi}_n := \arg\min_{\phi \in \Phi} \ell_{\mathrm{NPE}}(\phi), \quad \ell_{\mathrm{NPE}}(\phi) = -\frac{1}{n}\sum_{i=1}^{n} \log q_\phi(\theta_i \mid x_i) \approx -\mathbb{E}_{\theta \sim p(\theta)}[\mathbb{E}_{x \sim p(\cdot \mid \theta)}[\log q_\phi(\theta \mid x)]]$$

Papamakarios, G., & Murray, I. (2016). Fast e-free inference of simulation models with Bayesian conditional density estimation. *NeurIPS*, 1036–1044.

Lueckmann, J. M., Gonçalves, P. J., Bassetto, G., Öcal, K., Nonnenmacher, M., & Macke, J. H. (2017). Flexible statistical inference for mechanistic models of neural dynamics. *NeurIPS*, 1290–1300.

Greenberg, D. S., Nonnenmacher, M., & Macke, J. H. (2019). Automatic posterior transformation for likelihood-free inference. *ICML*, 4288–4304.

Neural posterior estimation (NPE)

- **Step 1:** train $q_\phi(\theta\,|\,x)$ to approximate the posterior using samples from the prior ($\theta_1, \ldots, \theta_n \sim p(\theta)$) and simulator ($x_i \sim p(\,\cdot\,|\,\theta_i)$):

$$\hat{\phi}_n := \arg\min_{\phi \in \Phi} \ell_{\text{NPE}}(\phi), \quad \ell_{\text{NPE}}(\phi) = -\frac{1}{n}\sum_{i=1}^{n} \log q_\phi(\theta_i\,|\,x_i) \approx -\mathbb{E}_{\theta\sim p(\theta)}[\mathbb{E}_{x\sim p(\cdot|\theta)}[\log q_\phi(\theta\,|\,x)]]$$

- **Step 2:** Condition on the observed data:

$$p_{\text{NPE}}(\theta\,|\,y_1, \ldots, y_n) = q_{\hat{\phi}_n}(\theta\,|\,y_1, \ldots, y_n)$$

Papamakarios, G., & Murray, I. (2016). Fast e-free inference of simulation models with Bayesian conditional density estimation. *NeurIPS*, 1036–1044.

Lueckmann, J. M., Gonçalves, P. J., Bassetto, G., Öcal, K., Nonnenmacher, M., & Macke, J. H. (2017). Flexible statistical inference for mechanistic models of neural dynamics. *NeurIPS*, 1290–1300.

Greenberg, D. S., Nonnenmacher, M., & Macke, J. H. (2019). Automatic posterior transformation for likelihood-free inference. *ICML*, 4288–4304.

Amortisation of NPE

$$p_{\mathrm{NPE}}(\theta \,|\, y_1, \ldots, y_n) = q_{\hat{\phi}_n}(\theta \,|\, y_1, \ldots, y_n)$$

Amortisation of NPE

$$p_{\text{NPE}}(\theta \mid y_1, \ldots, y_n) = q_{\hat{\phi}_n}(\theta \mid y_1, \ldots, y_n)$$

- What if we get some new observations $\tilde{y}_1, \ldots, \tilde{y}_n$?

Amortisation of NPE

$$p_{\text{NPE}}(\theta \mid y_1, \ldots, y_n) = q_{\hat{\phi}_n}(\theta \mid y_1, \ldots, y_n)$$

- What if we get some new observations $\tilde{y}_1, \ldots, \tilde{y}_n$?

$$p_{\text{NPE}}(\theta \mid \tilde{y}_1, \ldots, \tilde{y}_n) = q_{\hat{\phi}_n}(\theta \mid \tilde{y}_1, \ldots, \tilde{y}_n)$$

Amortisation of NPE

$$p_{\text{NPE}}(\theta \,|\, y_1, \ldots, y_n) = q_{\hat{\phi}_n}(\theta \,|\, y_1, \ldots, y_n)$$

- What if we get some new observations $\tilde{y}_1, \ldots, \tilde{y}_n$?

$$p_{\text{NPE}}(\theta \,|\, \tilde{y}_1, \ldots, \tilde{y}_n) = q_{\hat{\phi}_n}(\theta \,|\, \tilde{y}_1, \ldots, \tilde{y}_n)$$

- We have a direct handle on the new posterior; no need for MCMC/VI!

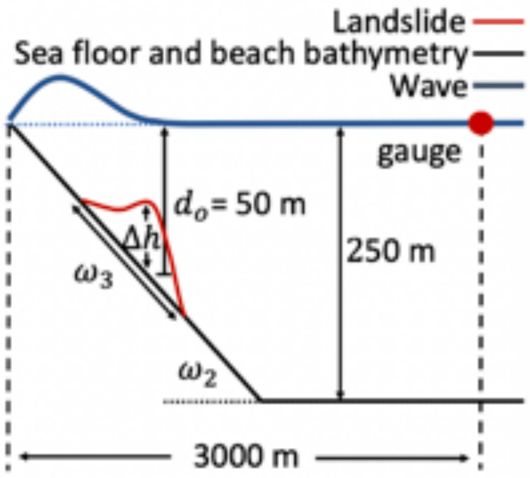
→ We are **fully amortised**.

Any Questions?

Challenges with existing SBI methods

Challenge 1: Expensive simulators

Example 1:

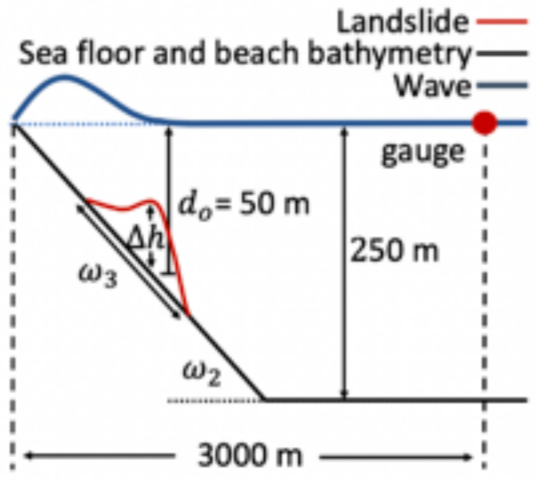


≈ 2 hours per sim on laptop

Li, K., Giles, D., Karvonen, T., Guillas, S., & **Briol, F.-X**. (2023). Multilevel Bayesian quadrature. *AISTATS*, 1845–1868.

Challenge 1: Expensive simulators

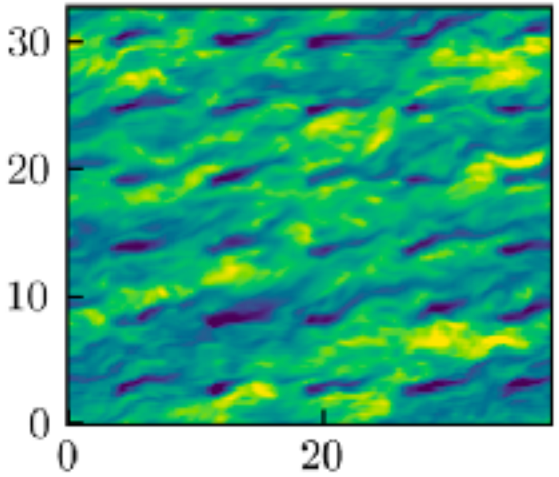
Example 1:



≈ 2 hours per sim on laptop

Li, K., Giles, D., Karvonen, T., Guillas, S., & **Briol, F.-X**. (2023). Multilevel Bayesian quadrature. *AISTATS*, 1845–1868.

Example 2:

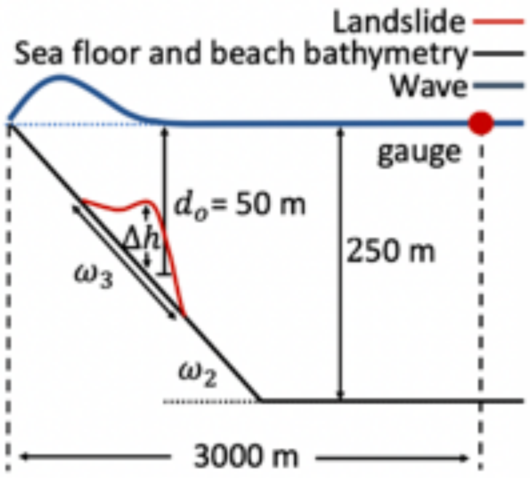


≈ 100 hours per sim on Met Office cluster

Kirby, A., **Briol, F.-X.**, Dunstan, T. D., & Nishino, T. (2023). Data-driven modelling of turbine wake interactions and flow resistance in large wind farms. *Wind Energy*, *26*(9), 875–1011.

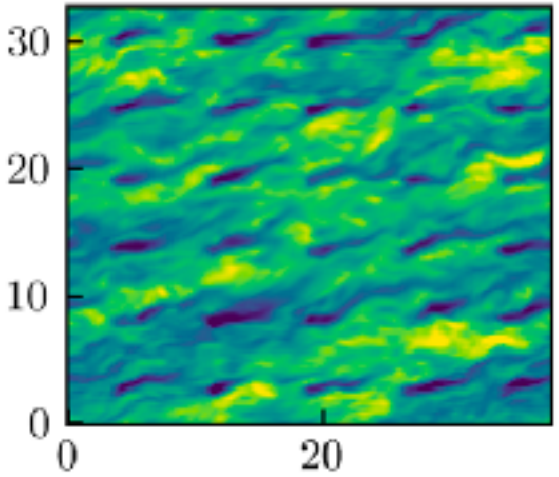
Challenge 1: Expensive simulators

Example 1:



≈ 2 hours per sim on laptop

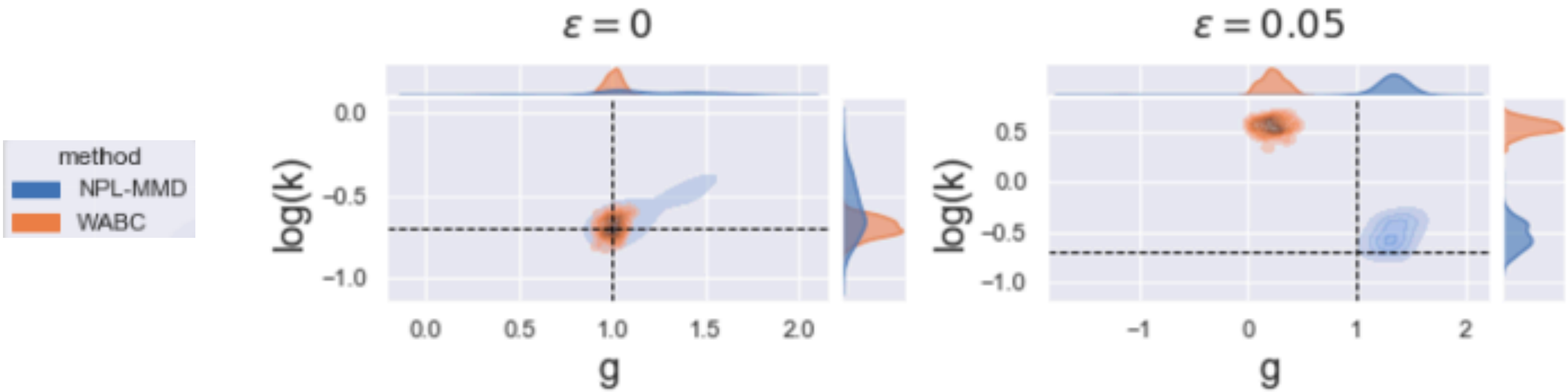
Example 2:



≈ 100 hours per sim on Met Office cluster

→ Currently out of reach of modern SBI methods!

Challenge 2: Model misspecification

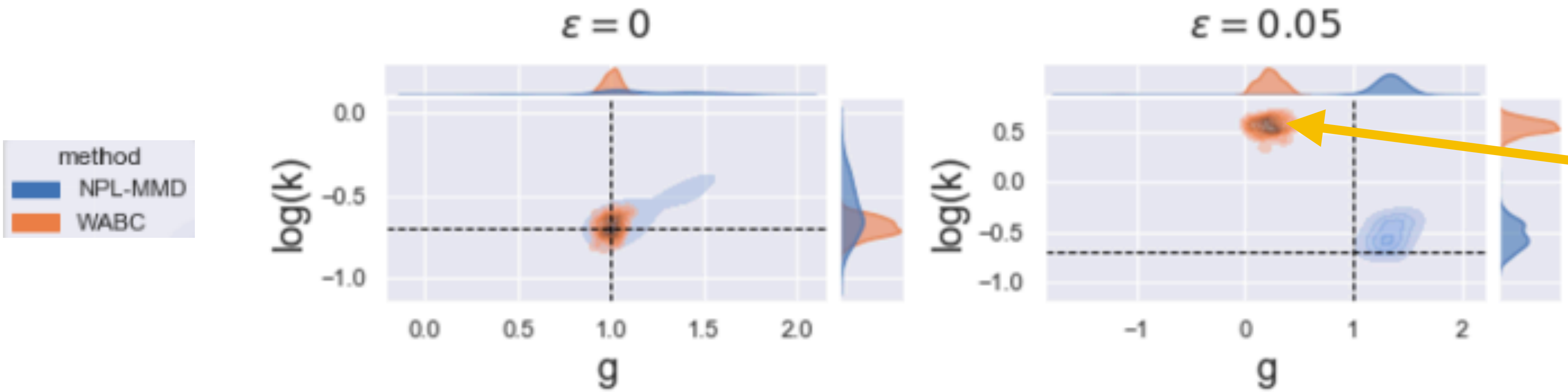


G-and-k distribution

Dellaporta, C., Knoblauch, J., Damoulas, T. & **Briol, F-X** (2022). Robust Bayesian inference for simulator-based models via the MMD posterior bootstrap. AISTATS, 943-970. Best paper award.

Kelly, R. P., Warne, D. J., Frazier, D. T., Nott, D. J., Gutmann, M. U., & Drovandi, C. (2025). Simulation-based Bayesian inference under model misspecification. *arXiv:2503.12315*.

Challenge 2: Model misspecification



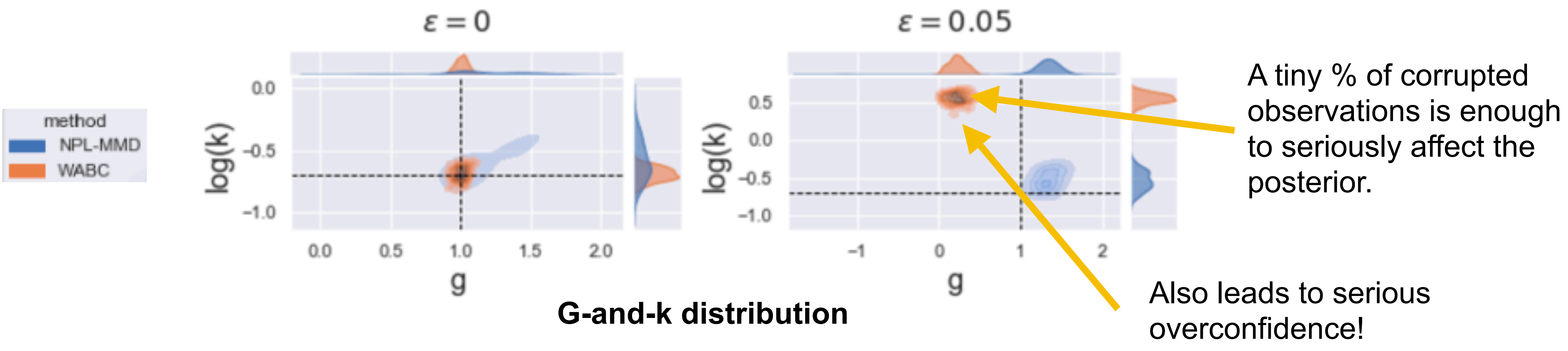
A tiny % of corrupted observations is enough to seriously affect the posterior.

G-and-k distribution

Dellaporta, C., Knoblauch, J., Damoulas, T. & **Briol, F-X** (2022). Robust Bayesian inference for simulator-based models via the MMD posterior bootstrap. AISTATS, 943-970. Best paper award.

Kelly, R. P., Warne, D. J., Frazier, D. T., Nott, D. J., Gutmann, M. U., & Drovandi, C. (2025). Simulation-based Bayesian inference under model misspecification. *arXiv:2503.12315*.

Challenge 2: Model misspecification

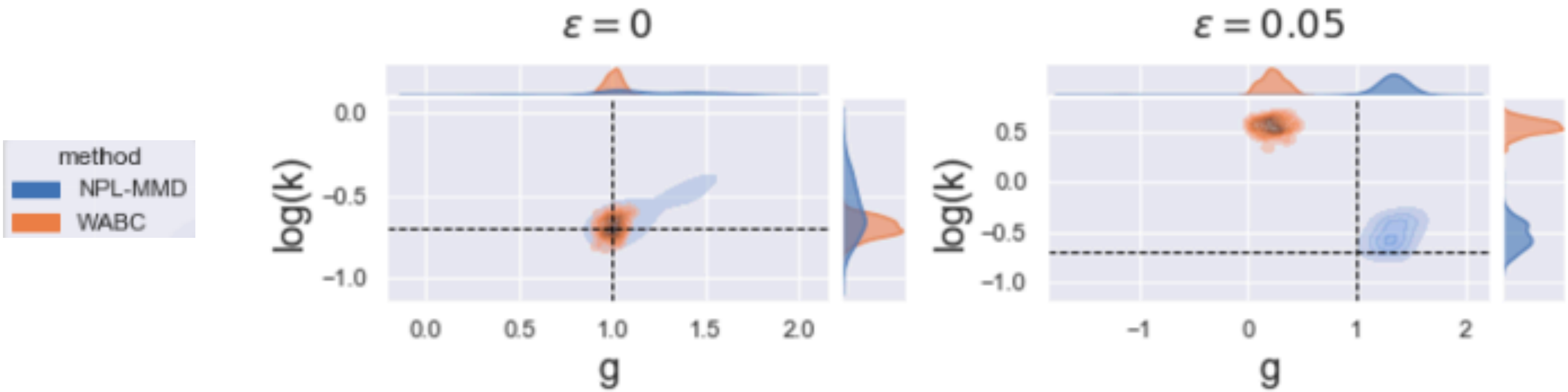


G-and-k distribution

Dellaporta, C., Knoblauch, J., Damoulas, T. & **Briol, F-X** (2022). Robust Bayesian inference for simulator-based models via the MMD posterior bootstrap. AISTATS, 943-970. Best paper award.

Kelly, R. P., Warne, D. J., Frazier, D. T., Nott, D. J., Gutmann, M. U., & Drovandi, C. (2025). Simulation-based Bayesian inference under model misspecification. *arXiv:2503.12315*.

Challenge 2: Model misspecification



G-and-k distribution

Currently very few robust methods with theoretical guarantees

Challenge 3: Over-confidence

Published in Transactions on Machine Learning Research (11/2022)

A Trust Crisis In Simulation-Based Inference? Your Posterior Approximations Can Be Unfaithful

Joeri Hermans* joeri@peinser.com
Unaffiliated

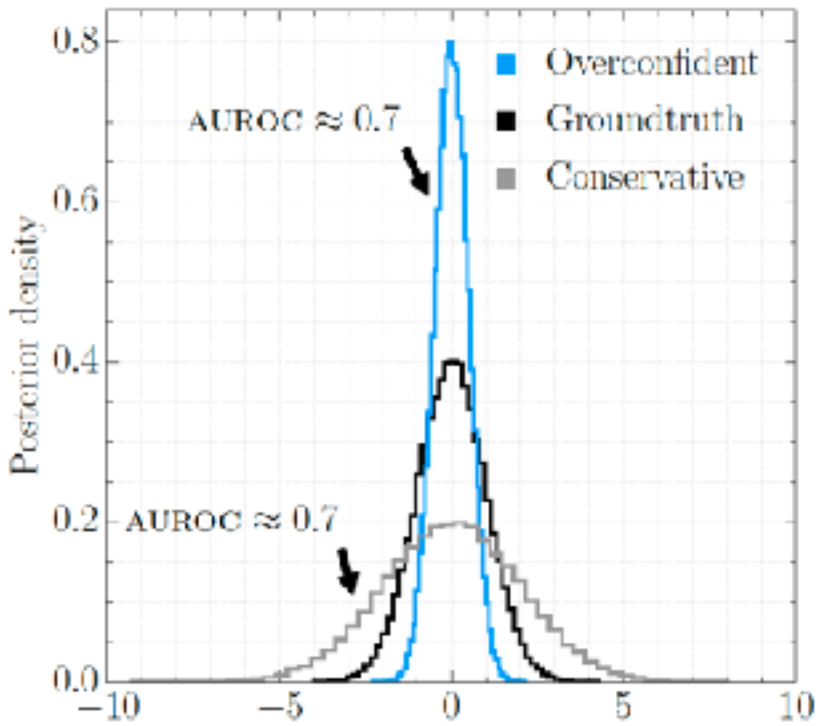
Arnaud Delaunoy* a.delaunoy@uliege.be
University of Liège

François Rozet francois.rozet@uliege.be
University of Liège

Antoine Wehenkel antoine.wehenkel@uliege.be
University of Liège

Volodimir Begy volodimir.begy@univie.ac.at
University of Vienna

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



Challenge 3: Over-confidence

Published in Transactions on Machine Learning Research (11/2022)

A Trust Crisis In Simulation-Based Inference? Your Posterior Approximations Can Be Unfaithful

Joeri Hermans* joeri@peinser.com
Unaffiliated

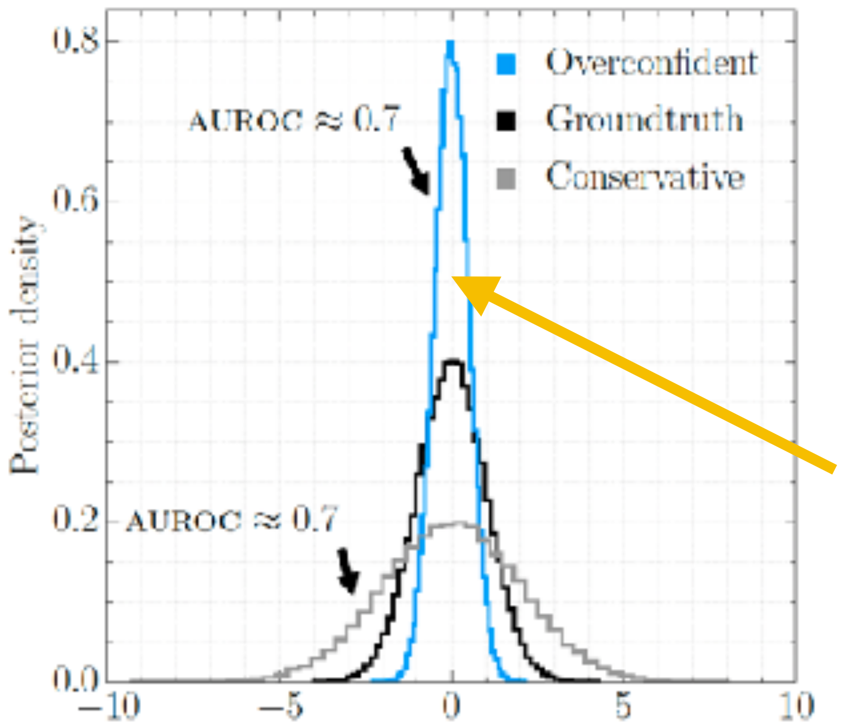
Arnaud Delaunoy* a.delaunoy@uliege.be
University of Liège

François Rozet francois.rozet@uliege.be
University of Liège

Antoine Wehenkel antoine.wehenkel@uliege.be
University of Liège

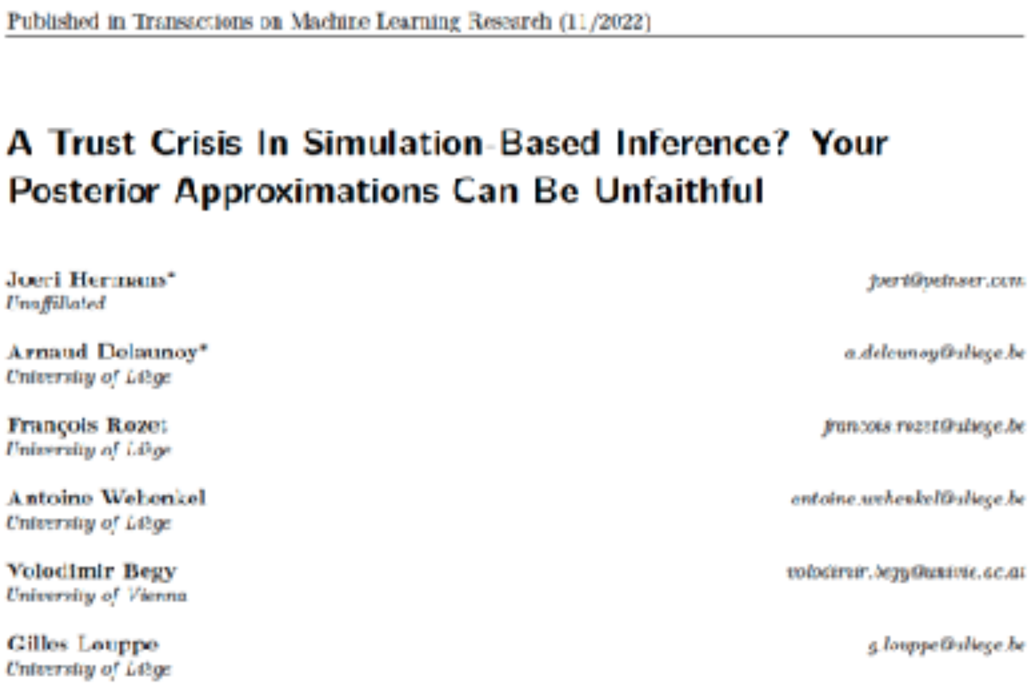
Volodimir Begy volodimir.begy@univie.ac.at
University of Vienna

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



Overconfident =
Too narrow!

Challenge 3: Over-confidence


Published in Transactions on Machine Learning Research (11/2022)

A Trust Crisis In Simulation-Based Inference? Your Posterior Approximations Can Be Unfaithful

Joeri Hermans* joeri@peinser.com
Unaffiliated

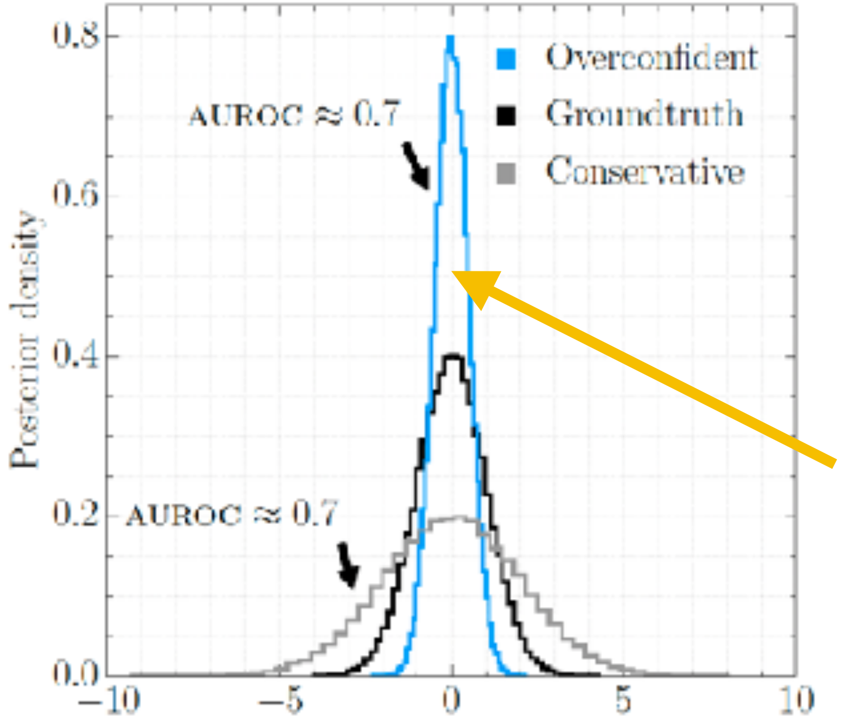
Arnaud Delaunoy* a.delaunoy@uliege.be
University of Liège

François Rozet francois.rozet@uliege.be
University of Liège

Antoine Wehenkel antoine.wehenkel@uliege.be
University of Liège

Volodimir Begy volodimir.begy@univie.ac.at
University of Vienna

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



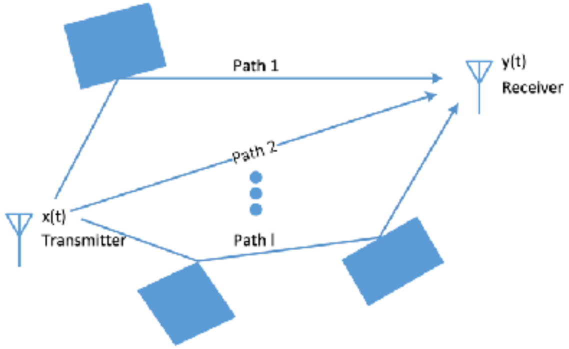
Overconfident =
Too narrow!

Observation 1 All benchmarked algorithms may produce non-conservative posterior approximations.

Challenge 4: High-dimensionality

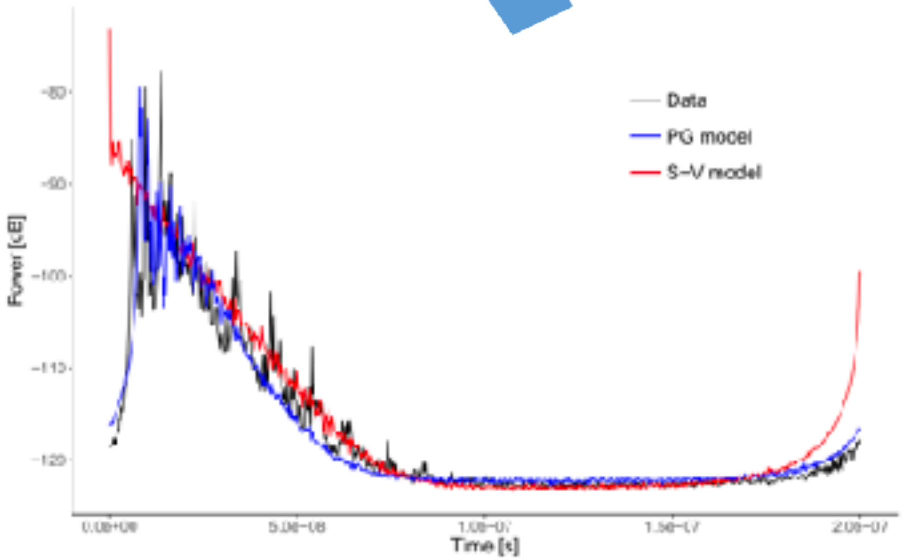
- As with everything in stats/ML, the curse of dimensionality hurts us…. Computing distances or estimating densities is very tough!

Challenge 4: High-dimensionality

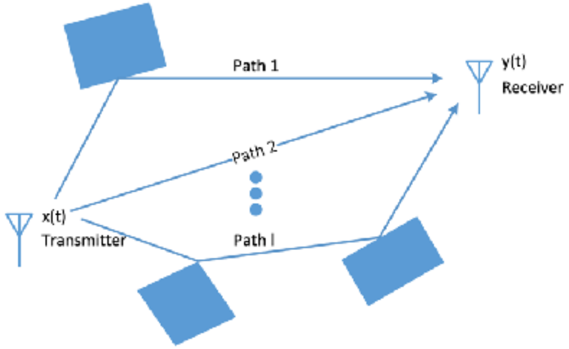


- As with everything in stats/ML, the curse of dimensionality hurts us…. Computing distances or estimating densities is very tough!

- Remember the radio-propagation example. The dimension is typically around 800….

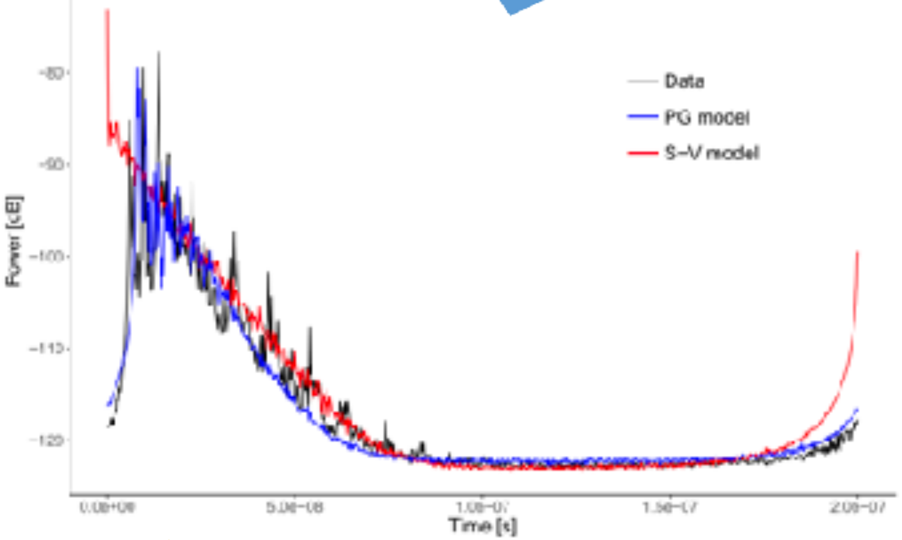


Challenge 4: High-dimensionality



- As with everything in stats/ML, the curse of dimensionality hurts us…. Computing distances or estimating densities is very tough!

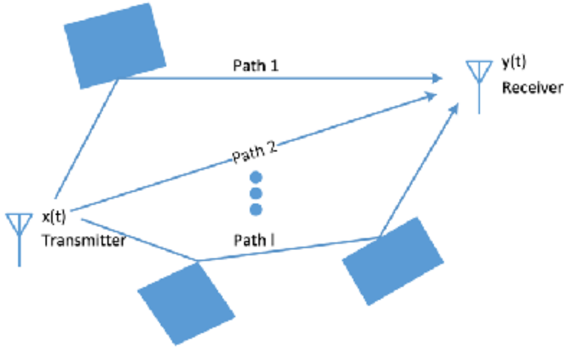
- Remember the radio-propagation example. The dimension is typically around 800….



Turns out 4 dimensional summary is practically sufficient here!

Bharti, A., **Briol, F.-X.**, & Pedersen, T. (2021). A general method for calibrating stochastic radio channel models with kernels. *IEEE Transactions on Antennas and Propagation*, *70*(6), 3986–4001.

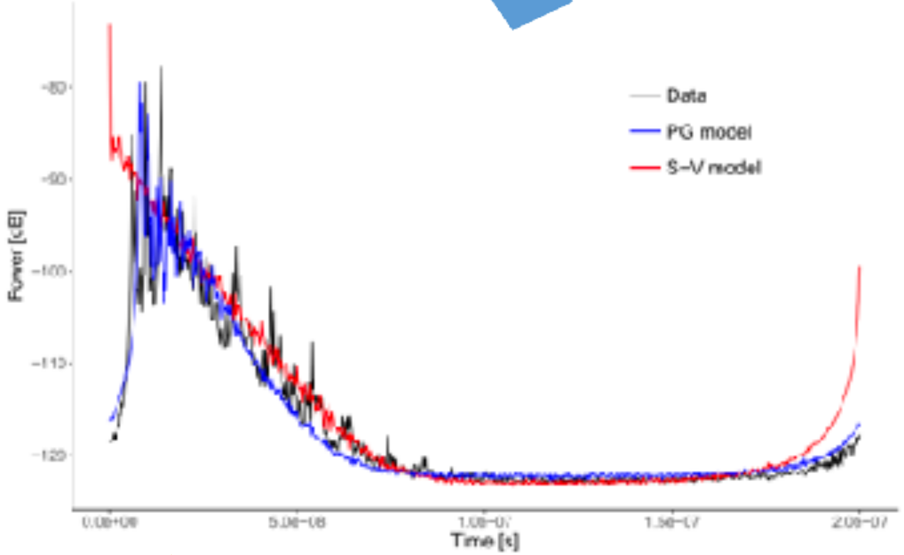
Challenge 4: High-dimensionality



- As with everything in stats/ML, the curse of dimensionality hurts us…. Computing distances or estimating densities is very tough!

- Remember the radio-propagation example. The dimension is typically around 800….

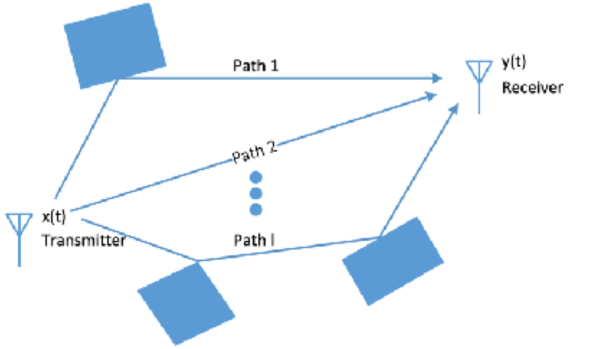
- We therefore end up working with **summary statistics,** either hand-crafted or learnt via a neural network (i.e. a ‘summary network’).



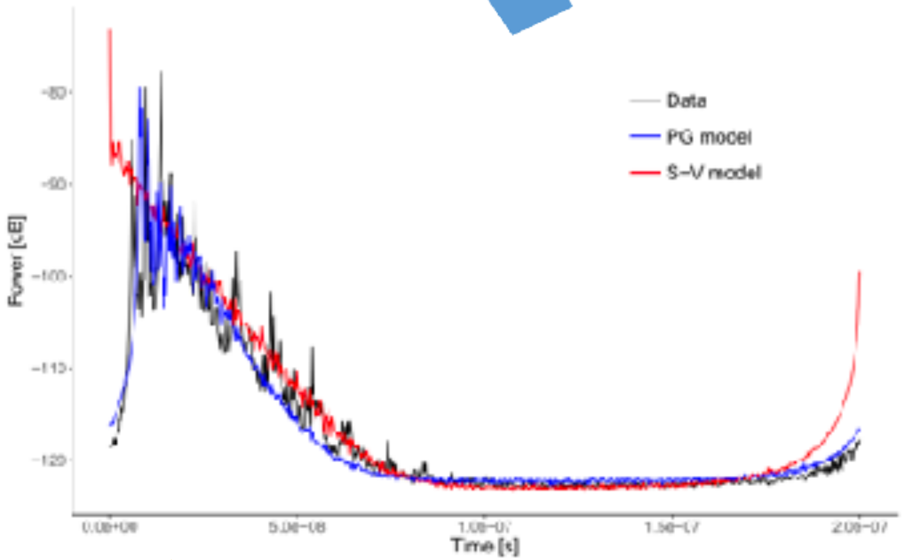
Turns out 4 dimensional summary is practically sufficient here!

Bharti, A., **Briol, F.-X.**, & Pedersen, T. (2021). A general method for calibrating stochastic radio channel models with kernels. *IEEE Transactions on Antennas and Propagation*, *70*(6), 3986–4001.

Challenge 4: High-dimensionality



- As with everything in stats/ML, the curse of dimensionality hurts us…. Computing distances or estimating densities is very tough!
- Remember the radio-propagation example. The dimension is typically around 800….
- We therefore end up working with **summary statistics,** either hand-crafted or learnt via a neural network (i.e. a 'summary network').
- Dimensionality of parameter space also a problem…



Turns out 4 dimensional summary is practically sufficient here!

Bharti, A., **Briol, F.-X.**, & Pedersen, T. (2021). A general method for calibrating stochastic radio channel models with kernels. *IEEE Transactions on Antennas and Propagation*, *70*(6), 3986–4001.

Roadmap going ahead…

Background + challenges for SBI

Roadmap going ahead…

Background + challenges for SBI

Snapshot 1:
Multi-fidelity methods for simulation-based inference (NeurIPS?, 2025)

Roadmap going ahead…

Background + challenges for SBI

Snapshot 1:
Multi-fidelity methods for simulation-based inference (NeurIPS?, 2025)

Snapshot 2:
Cost-aware methods for simulation-based inference (AISTATS, 2025)

Roadmap going ahead…

Background + challenges for SBI

Snapshot 1:
Multi-fidelity methods for simulation-based inference (NeurIPS?, 2025)

Snapshot 2:
Cost-aware methods for simulation-based inference (AISTATS, 2025)

Snapshot 3:
Provably robust generalisation of Bayes for simulation-based inference (AISTATS Best Paper Award, 2022)

Any Questions?

Multilevel neural simulation-based inference

Paper: Hikida, Y., Bharti, A., Jeffrey, N. & **Briol, F-X** (2025). Multilevel neural simulation-based inference. arXiv:2506.06087. (to appear at NeurIPS?)

Code: https://github.com/yugahikida/multilevel-sbi

Challenge for SBI

Simulators can be really computationally expensive!

Challenge for SBI

Simulators can be really computationally expensive!

- Most simulators used in SBI papers take only a few seconds (or less) to run.
- Even if a simulator takes only a few minutes, we typically need thousands of simulations!
- Simulators that take more time are currently out of reach of existing methods.

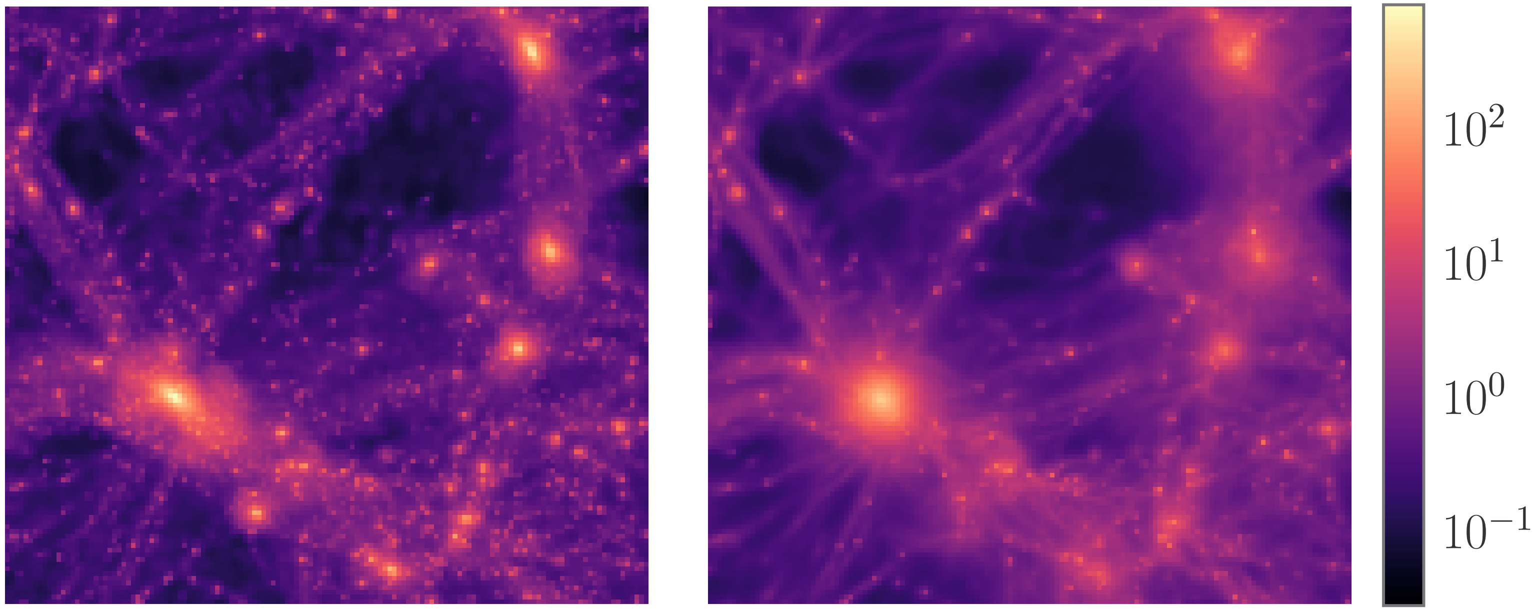
Challenge for SBI

Simulators can be really computationally expensive!

- Most simulators used in SBI papers take only a few seconds (or less) to run.
- Even if a simulator takes only a few minutes, we typically need thousands of simulations!
- Simulators that take more time are currently out of reach of existing methods.

→ This leads to a form of model misspecification by design!

SBI for cosmology



Low-fidelity High-fidelity

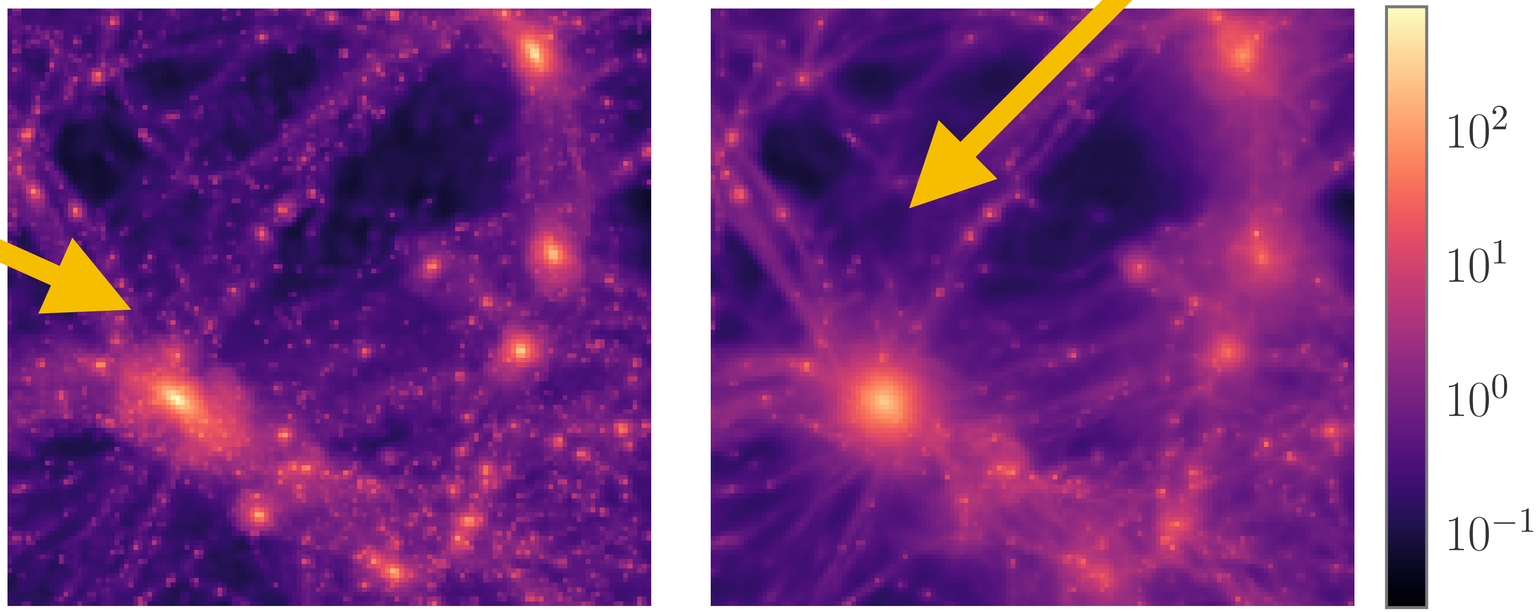
Jeffrey, N., et al. (2025). Dark energy survey year 3 results: likelihood-free, simulation-based wCDM inference with neural compression of weak-lensing map statistics. *Monthly Notices of the Royal Astronomical Society*, *536*(2), 1303–1322.

Villaescusa-Navarro, F., et al. (2021). The CAMELS project: Cosmology and astrophysics with machine-learning simulations. *The Astrophysical Journal*, *915*(1), 71.

SBI for cosmology

Gravity-only N-body simulations

Hydrodynamic simulations



Low-fidelity High-fidelity

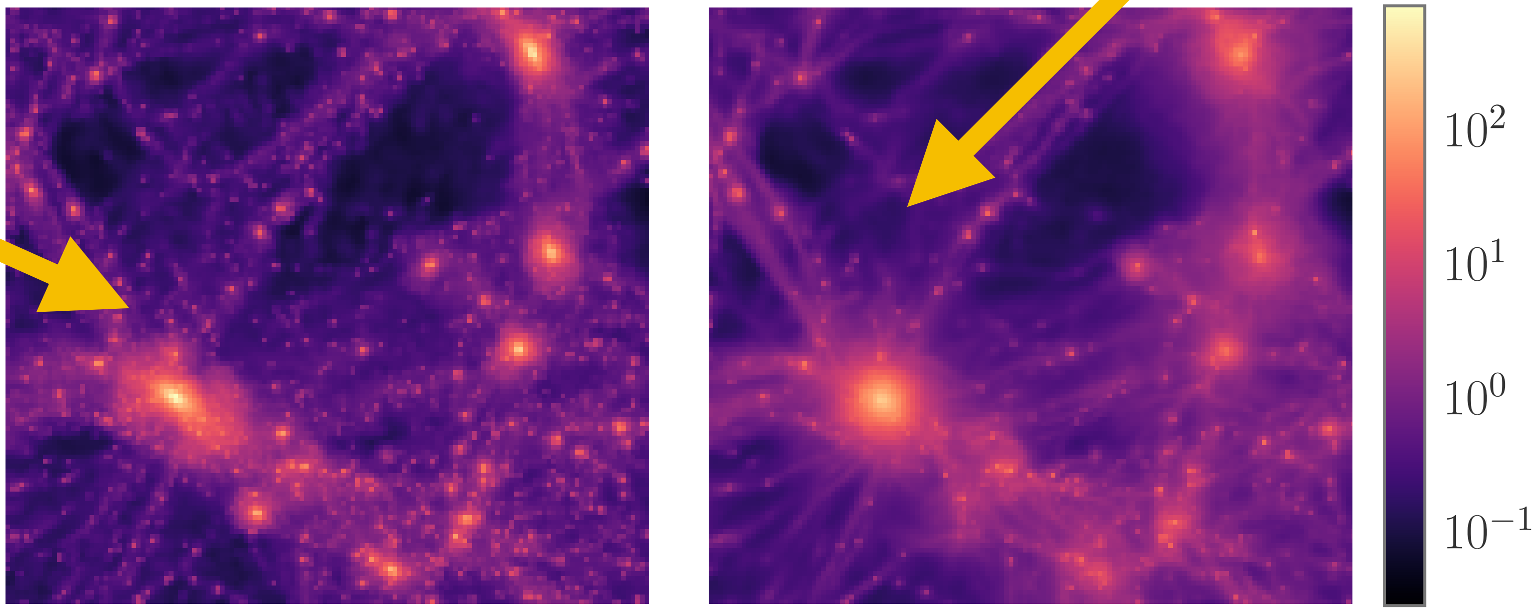
Jeffrey, N., et al. (2025). Dark energy survey year 3 results: likelihood-free, simulation-based wCDM inference with neural compression of weak-lensing map statistics. *Monthly Notices of the Royal Astronomical Society*, *536*(2), 1303–1322.

Villaescusa-Navarro, F., et al. (2021). The CAMELS project: Cosmology and astrophysics with machine-learning simulations. *The Astrophysical Journal*, *915*(1), 71.

SBI for cosmology

Hydrodynamic simulations

Gravity-only N-body simulations



Low-fidelity

High-fidelity

$\approx$ 100x more expensive!!

Jeffrey, N., et al. (2025). Dark energy survey year 3 results: likelihood-free, simulation-based wCDM inference with neural compression of weak-lensing map statistics. *Monthly Notices of the Royal Astronomical Society*, *536*(2), 1303–1322.

Villaescusa-Navarro, F., et al. (2021). The CAMELS project: Cosmology and astrophysics with machine-learning simulations. *The Astrophysical Journal*, *915*(1), 71.

Existing work on multi-fidelity in SBI

Many great works, but which are not specialised for neural-SBI:

Jasra, A., Jo, S., Nott, D., Shoemaker, C., & Tempone, R. (2019). Multilevel Monte Carlo in approximate Bayesian computation. *Stochastic Analysis and Applications*, *37*(3), 346–360.

Prescott, T. P., & Baker, R. E. (2020). Multifidelity approximate Bayesian computation. *SIAM-ASA Journal on Uncertainty Quantification*, *8*(1), 114–138.

Warne, D. J., Prescott, T. P., Baker, R. E., & Simpson, M. J. (2022). Multifidelity multilevel Monte Carlo to accelerate approximate Bayesian parameter inference for partially observed stochastic processes. *Journal of Computational Physics*, *469*, 111543.

Existing work on multi-fidelity in SBI

One very recent attempt, but no theory and critical issue with hyper parameter selection:

Krouglova, A. N., Johnson, H. R., Confavreux, B., Deistler, M., & Gonçalves, P. J. (2025). Multifidelity simulation-based inference for computationally expensive simulators. a*rXiv:2502.08416*.

Existing work on multi-fidelity in SBI

→ **Open problem:** Rigorous and theoretically-grounded multi-fidelity for neural SBI!

Neural likelihood estimation (NLE)

- **Step 1:** train $q_\phi(\cdot \mid \theta)$ to approximate the likelihood using samples from the prior ($\theta_1, \ldots, \theta_n \sim p(\theta)$) and simulator ($x_i \sim p(\cdot \mid \theta_i)$):

$$\hat{\phi}_n := \arg\min_{\phi \in \Phi} \ell_{\text{NLE}}(\phi), \quad \ell_{\text{NLE}}(\phi) = -\frac{1}{n}\sum_{i=1}^{n} \log q_\phi(x_i \mid \theta_i) \approx -\mathbb{E}_{\theta \sim p(\theta)}[\mathbb{E}_{x \sim p(\cdot \mid \theta)}[\log q_\phi(x \mid \theta)]]$$

Neural likelihood estimation (NLE)

- **Step 1:** train $q_\phi(\cdot\,|\,\theta)$ to approximate the likelihood using samples from the prior ($\theta_1, \ldots, \theta_n \sim p(\theta)$) and simulator ($x_i \sim p(\cdot\,|\,\theta_i)$):

$$\hat{\phi}_n := \arg\min_{\phi\in\Phi} \ell_{\text{NLE}}(\phi), \quad \ell_{\text{NLE}}(\phi) = -\frac{1}{n}\sum_{i=1}^{n} \log q_\phi(x_i\,|\,\theta_i) \approx -\mathbb{E}_{\theta\sim p(\theta)}[\mathbb{E}_{x\sim p(\cdot|\theta)}[\log q_\phi(x\,|\,\theta)]]$$

- **Step 2:** Do Bayes with approximate likelihood!

$$p_{\text{NLE}}(\theta\,|\,y_1, \ldots, y_n) \propto \prod_{i=1}^{n} q_{\hat{\phi}_n}(y_i\,|\,\theta)p(\theta)$$

Neural likelihood estimation (NLE)

- **Step 1:** train $q_\phi(\cdot \mid \theta)$ to approximate the likelihood using samples from the prior ($\theta_1, \ldots, \theta_n \sim p(\theta)$) and simulator ($x_i \sim p(\cdot \mid \theta_i)$):

$$\hat{\phi}_n := \arg\min_{\phi \in \Phi} \ell_{\text{NLE}}(\phi), \quad \ell_{\text{NLE}}(\phi) = -\frac{1}{n}\sum_{i=1}^{n} \log q_\phi(x_i \mid \theta_i) \approx -\mathbb{E}_{\theta \sim p(\theta)}[\mathbb{E}_{x \sim p(\cdot \mid \theta)}[\log q_\phi(x \mid \theta)]]$$

Typically the most **computationally expensive** step!!

- **Step 2:** Do Bayes with approximate likelihood!

$$p_{\text{NLE}}(\theta \mid y_1, \ldots, y_n) \propto \prod_{i=1}^{n} q_{\hat{\phi}_n}(y_i \mid \theta) p(\theta)$$

A better step 1?

$$\ell_{\mathrm{NLE}}(\phi) = -\frac{1}{n}\sum_{i=1}^{n} \log q_\phi(x_i \,|\, \theta_i) \approx -\mathbb{E}_{\theta\sim p(\theta)}[\mathbb{E}_{x\sim p(\cdot|\theta)}[\log q_\phi(x \,|\, \theta)]]$$

Can we do this better/cheaper?!

A better step 1?

$$\ell_{\mathrm{NLE}}(\phi) = -\frac{1}{n}\sum_{i=1}^{n} \log q_\phi(x_i \,|\, \theta_i) \approx -\mathbb{E}_{\theta \sim p(\theta)}[\mathbb{E}_{x \sim p(\cdot|\theta)}[\log q_\phi(x \,|\, \theta)]]$$

Can we do this better/cheaper?!

→ Yes, Multilevel Monte Carlo!

Giles, M. B. (2015). Multilevel Monte Carlo methods. *Acta Numerica*, *24*, 259–328.

Jasra, A., Law, K., & Suciu, C. (2020). Advanced Multilevel Monte Carlo Methods. *International Statistical Review*, *88*(3), 548–579.

Multilevel Monte Carlo

Suppose we have a $f_0, f_1, \ldots, f_L = f$ of increasing cost but also increasing accuracy. Then:

$$\mathbb{E}_{z\sim\mu}[f(z)]$$

Multilevel Monte Carlo

Suppose we have a $f_0, f_1, \ldots, f_L = f$ of increasing cost but also increasing accuracy. Then:

$$\mathbb{E}_{z\sim\mu}[f(z)] \; = \mathbb{E}_{z\sim\mu}[f_L(z)]$$

Multilevel Monte Carlo

Suppose we have a $f_0, f_1, \ldots, f_L = f$ of increasing cost but also increasing accuracy. Then:

$$\mathbb{E}_{z\sim\mu}[f(z)] = \mathbb{E}_{z\sim\mu}[f_L(z)] = \mathbb{E}_{z\sim\mu}\left[f_{L-1}(z)\right] + \mathbb{E}_{z\sim\mu}\left[f_L(z) - f_{L-1}(z)\right]$$

Multilevel Monte Carlo

Suppose we have a $f_0, f_1, \ldots, f_L = f$ of increasing cost but also increasing accuracy. Then:

$$\mathbb{E}_{z\sim\mu}[f(z)] = \mathbb{E}_{z\sim\mu}[f_L(z)] = \mathbb{E}_{z\sim\mu}\left[f_{\boxed{L-1}}(z)\right] + \mathbb{E}_{z\sim\mu}\left[f_{\boxed{L}}(z) - f_{\boxed{L-1}}(z)\right]$$

Multilevel Monte Carlo

Suppose we have a $f_0, f_1, \ldots, f_L = f$ of increasing cost but also increasing accuracy. Then:

$$\mathbb{E}_{z\sim\mu}[f(z)] = \mathbb{E}_{z\sim\mu}[f_L(z)] = \mathbb{E}_{z\sim\mu}\left[f_{L-1}(z)\right] + \mathbb{E}_{z\sim\mu}\left[f_L(z) - f_{L-1}(z)\right]$$

$$= \mathbb{E}_{z\sim\mu}\left[f_0(z)\right] + \sum_{l=1}^{L} \mathbb{E}_{z\sim\mu}\left[f_l(z) - f_{l-1}(z)\right]$$

Multilevel Monte Carlo

Suppose we have a $f_0, f_1, \ldots, f_L = f$ of increasing cost but also increasing accuracy. Then:

$$
\begin{aligned}
\mathbb{E}_{z\sim\mu}[f(z)] &= \mathbb{E}_{z\sim\mu}[f_L(z)] = \mathbb{E}_{z\sim\mu}\left[f_{L-1}(z)\right] + \mathbb{E}_{z\sim\mu}\left[f_L(z) - f_{L-1}(z)\right] \\
&= \mathbb{E}_{z\sim\mu}\left[f_0(z)\right] + \sum_{l=1}^{L} \mathbb{E}_{z\sim\mu}\left[f_l(z) - f_{l-1}(z)\right] \\
&\approx \frac{1}{n_0}\sum_{i=1}^{n_0} f_0(z_i^0) + \sum_{l=1}^{L}\left(\frac{1}{n_l}\sum_{i=1}^{n_l}\left(f_l\left(z_i^l\right) - f_{l-1}\left(z_i^l\right)\right)\right)
\end{aligned}
$$

Multilevel Monte Carlo

Suppose we have a $f_0, f_1, \dots, f_L = f$ of increasing cost but also increasing accuracy. Then:

$$
\begin{aligned}
\mathbb{E}_{z\sim\mu}[f(z)] &= \mathbb{E}_{z\sim\mu}[f_L(z)] = \mathbb{E}_{z\sim\mu}\left[f_{L-1}(z)\right] + \mathbb{E}_{z\sim\mu}\left[f_L(z) - f_{L-1}(z)\right] \\
&= \mathbb{E}_{z\sim\mu}\left[f_0(z)\right] + \sum_{l=1}^{L} \mathbb{E}_{z\sim\mu}\left[f_l(z) - f_{l-1}(z)\right] \\
&\approx \frac{1}{n_0}\sum_{i=1}^{n_0} f_0(z_i^0) + \sum_{l=1}^{L}\left(\frac{1}{n_l}\sum_{i=1}^{n_l}\left(f_l\left(z_i^l\right) - f_{l-1}\left(z_i^l\right)\right)\right)
\end{aligned}
$$

Very cheap - can take n_0 large.

Multilevel Monte Carlo

Suppose we have a $f_0, f_1, \ldots, f_L = f$ of increasing cost but also increasing accuracy. Then:

$$
\begin{aligned}
\mathbb{E}_{z\sim\mu}[f(z)] &= \mathbb{E}_{z\sim\mu}[f_L(z)] = \mathbb{E}_{z\sim\mu}\left[f_{L-1}(z)\right] + \mathbb{E}_{z\sim\mu}\left[f_L(z) - f_{L-1}(z)\right] \\
&= \mathbb{E}_{z\sim\mu}\left[f_0(z)\right] + \sum_{l=1}^{L} \mathbb{E}_{z\sim\mu}\left[f_l(z) - f_{l-1}(z)\right] \\
&\approx \frac{1}{n_0}\sum_{i=1}^{n_0} f_0(z_i^0) + \sum_{l=1}^{L}\left(\frac{1}{n_l}\sum_{i=1}^{n_l}\left(f_l\left(z_i^l\right) - f_{l-1}\left(z_i^l\right)\right)\right)
\end{aligned}
$$

Very cheap - can take n_0 large.

Very expensive - **cannot** take n_l large…. But low variance!

Multilevel NLE

$$-\mathbb{E}_{\theta\sim\pi}\left[\mathbb{E}_{x\sim\mathbb{P}_\theta}\left[\log q_\phi(x\,|\,\theta)\right]\right]$$

Multilevel NLE

Change of measure

$$-\mathbb{E}_{\theta\sim\pi}\left[\mathbb{E}_{x\sim\mathbb{P}_\theta}\left[\log q_\phi(x\,|\,\theta)\right]\right] = \mathbb{E}_{\theta\sim\pi,u\sim\mathbb{U}}\left[-\log q_\phi\left(G_\theta(u)\,|\,\theta\right)\right]$$

Multilevel NLE

$$-\mathbb{E}_{\theta\sim\pi}\left[\mathbb{E}_{x\sim\mathbb{P}_\theta}\left[\log q_\phi(x\,|\,\theta)\right]\right] = \mathbb{E}_{\theta\sim\pi,u\sim\mathbb{U}}\left[-\log q_\phi\left(G_\theta(u)\,|\,\theta\right)\right]$$

$$= \mathbb{E}_{\theta\sim\pi,u\sim\mathbb{U}}\left[-\log q_\phi\left(G_\theta^L(u)\,|\,\theta\right)\right]$$

Multilevel NLE

$$-\mathbb{E}_{\theta\sim\pi}\left[\mathbb{E}_{x\sim\mathbb{P}_\theta}\left[\log q_\phi(x\,|\,\theta)\right]\right] = \mathbb{E}_{\theta\sim\pi,u\sim\mathbb{U}}\left[-\log q_\phi\left(G_\theta(u)\,|\,\theta\right)\right]$$

$$= \mathbb{E}_{\theta\sim\pi,u\sim\mathbb{U}}\left[-\log q_\phi\left(G_\theta^L(u)\,|\,\theta\right)\right]$$

$$= \mathbb{E}_{\theta\sim\pi,u\sim\mathbb{U}}\left[f_\phi^L(\theta,u)\right]$$

→ This is now a joint expectation in the prior and $\mathbb{U}$!

Multilevel NLE

$$
\begin{aligned}
-\mathbb{E}_{\theta\sim\pi}\left[\mathbb{E}_{x\sim\mathbb{P}_\theta}\left[\log q_\phi(x\,|\,\theta)\right]\right] &= \mathbb{E}_{\theta\sim\pi,u\sim\mathbb{U}}\left[-\log q_\phi\left(G_\theta(u)\,|\,\theta\right)\right] \\
&= \mathbb{E}_{\theta\sim\pi,u\sim\mathbb{U}}\left[-\log q_\phi\left(G_\theta^L(u)\,|\,\theta\right)\right] \\
&= \mathbb{E}_{\theta\sim\pi,u\sim\mathbb{U}}\left[f_\phi^L(\theta,u)\right]
\end{aligned}
$$

→ This is now a joint expectation in the prior and $\mathbb{U}$!

We can directly apply MLMC to it, where intermediate integrands are of the form:

$$
f_\phi^l(\theta,u) = -\log q_\phi\left(G_\theta^l(u)\,|\,\theta\right)
$$

Multilevel NLE

$$-\mathbb{E}_{\theta\sim\pi}\left[\mathbb{E}_{x\sim\mathbb{P}_\theta}\left[\log q_\phi(x\,|\,\theta)\right]\right] = \mathbb{E}_{\theta\sim\pi,u\sim\mathbb{U}}\left[-\log q_\phi\left(G_\theta(u)\,|\,\theta\right)\right]$$

$$= \mathbb{E}_{\theta\sim\pi,u\sim\mathbb{U}}\left[-\log q_\phi\left(G_\theta^L(u)\,|\,\theta\right)\right]$$

$$= \mathbb{E}_{\theta\sim\pi,u\sim\mathbb{U}}\left[f_\phi^L(\theta,u)\right]$$

→ This is now a joint expectation in the prior and $\mathbb{U}$!

We can directly apply MLMC to it, where intermediate integrands are of the form:

$$f_\phi^{\boxed{l}}(\theta,u) = -\log q_\phi\left(G_\theta^{\boxed{l}}(u)\,|\,\theta\right)$$

Multilevel neural SBI

Our ‘data’ is therefore:

$$\left\{ \theta_i^l, u_i^l, G_{\theta_i^l}^l\left(u_i^l\right), G_{\theta_i^l}^{l-1}\left(u_i^l\right) \right\} \quad \text{where} \quad \theta_i^l \sim \pi, u_i^l \sim \mathbb{U},$$

Multilevel neural SBI

Seed-matched!

Our ‘data’ is therefore:

$$\left\{ \theta_i^l, u_i^l, G_{\theta_i^l}^l\left(u_i^l\right), G_{\theta_i^l}^{l-1}\left(u_i^l\right) \right\} \quad \text{where} \quad \theta_i^l \sim \pi, u_i^l \sim \mathbb{U},$$

Multilevel neural SBI

Our ‘data’ is therefore:

$$\left\{\theta_i^l, u_i^l, G_{\theta_i^l}^l\left(u_i^l\right), G_{\theta_i^l}^{l-1}\left(u_i^l\right)\right\} \quad \text{where} \quad \theta_i^l \sim \pi, u_i^l \sim \mathbb{U},$$

Our objective for step 1 is:

$$\ell_{\text{ML-NLE}}(\phi) := \frac{1}{n_0}\sum_{i=1}^{n_0} f_\phi^0(u_i^0, \theta_i^0) + \sum_{l=1}^{L}\frac{1}{n_l}\sum_{i=1}^{n_l}\left(f_\phi^l(u_i^l, \theta_i^l) - f_\phi^{l-1}(u_i^l, \theta_i^l)\right)$$

Multilevel neural SBI

Our 'data' is therefore:

$$\left\{ \theta_i^l, u_i^l, G_{\theta_i^l}^l\left(u_i^l\right), G_{\theta_i^l}^{l-1}\left(u_i^l\right) \right\} \quad \text{where} \quad \theta_i^l \sim \pi, u_i^l \sim \mathbb{U},$$

Our objective for step 1 is:

$$\ell_{\mathrm{ML-NLE}}(\phi) := \frac{1}{n_0} \sum_{i=1}^{n_0} f_\phi^0(u_i^0, \theta_i^0) + \sum_{l=1}^{L} \frac{1}{n_l} \sum_{i=1}^{n_l} \left(f_\phi^l(u_i^l, \theta_i^l) - f_\phi^{l-1}(u_i^l, \theta_i^l) \right)$$

Note that we presented this for NLE, but the same could work for NPE, other scoring rules, etc…!

Challenges with training

$$\ell_{\mathrm{ML-NLE}}(\phi) := \frac{1}{n_0}\sum_{i=1}^{n_0} f_\phi^0(u_i^0, \theta_i^0) + \frac{1}{n_1}\sum_{i=1}^{n_1}\left(f_\phi^1(u_i^1, \theta_i^1) - f_\phi^0(u_i^1, \theta_i^1)\right)$$

Challenges with training

$$\ell_{\mathrm{ML-NLE}}(\phi) := \frac{1}{n_0}\sum_{i=1}^{n_0} f_\phi^0(u_i^0,\theta_i^0) + \frac{1}{n_1}\sum_{i=1}^{n_1}\left(f_\phi^1(u_i^1,\theta_i^1) - f_\phi^0(u_i^1,\theta_i^1)\right)$$

$$\frac{1}{n_0}\sum_{i=1}^{n_0} \nabla f_\phi^0(u_i^0,\theta_i^0) \approx \mathbb{E}[\nabla f_\phi^0] \qquad -\mathbb{E}[\nabla f_\phi^0] \approx -\frac{1}{n_1}\sum_{i=1}^{n_1} \nabla f_\phi^0(u_i^1,\theta_i^1)$$

Contradictory gradients! This is a problem when we are close to stationarity and n_0/n_1 are small… The variance of the negative term is always large!!

Challenges with training

$$\ell_{\mathrm{ML-NLE}}(\phi) := \frac{1}{n_0}\sum_{i=1}^{n_0} f_\phi^0(u_i^0, \theta_i^0) + \frac{1}{n_1}\sum_{i=1}^{n_1}\left(f_\phi^1(u_i^1, \theta_i^1) - f_\phi^0(u_i^1, \theta_i^1)\right)$$

$$\frac{1}{n_0}\sum_{i=1}^{n_0} \nabla f_\phi^0(u_i^0, \theta_i^0) \approx \mathbb{E}[\nabla f_\phi^0] \qquad -\mathbb{E}[\nabla f_\phi^0] \approx -\frac{1}{n_1}\sum_{i=1}^{n_1} \nabla f_\phi^0(u_i^1, \theta_i^1)$$

Contradictory gradients! This is a problem when we are close to stationarity and n_0/n_1 are small… The variance of the negative term is always large!!

We fix the issue by normalising gradients so that these two terms have the same magnitude, and by projecting onto each other's normal planes, which stabilises training.

Bound on the variance

Under some mild assumptions, we get:

$$\text{Var}\left[\ell_{\text{ML-NLE}}(\phi)\right] \leq \frac{K_0(\phi)}{n_0}\left(\|G^0\|^4_{W^{1,4}(\pi\times\mathbb{U})}+1\right)+\sum_{l=1}^{L}\frac{K_l(\phi)}{n_l}\left(\|G^l-G^{l-1}\|^2_{W^{1,4}(\pi\times\mathbb{U})}+1\right)$$

Bound on the variance

Under some mild assumptions, we get:

$$\text{Var}\left[\ell_{\text{ML-NLE}}(\phi)\right] \leq \frac{K_0(\phi)}{n_0}\left(\|G^0\|^4_{W^{1,4}(\pi\times\mathbb{U})}+1\right)+\sum_{l=1}^{L}\frac{K_l(\phi)}{n_l}\left(\|G^l-G^{l-1}\|^2_{W^{1,4}(\pi\times\mathbb{U})}+1\right)$$

Large!

Small!

Bound on the variance

Under some mild assumptions, we get:

$$\text{Var}\left[\ell_{\text{ML-NLE}}(\phi)\right] \leq \frac{K_0(\phi)}{n_0}\left(\|G^0\|^4_{W^{1,4}(\pi\times\mathbb{U})}+1\right)+\sum_{l=1}^{L}\frac{K_l(\phi)}{n_l}\left(\|G^l-G^{l-1}\|^2_{W^{1,4}(\pi\times\mathbb{U})}+1\right)$$

Large!

Complexity of low-fidelity generator - large!

Small!

Complexity of other integrands - small!

Bound on the variance

Under some mild assumptions, we get:

$$\text{Var}\left[\ell_{\text{ML-NLE}}(\phi)\right] \leq \frac{K_0(\phi)}{n_0}\left(\|G^0\|^4_{W^{1,4}(\pi\times\mathbb{U})} + 1\right) + \sum_{l=1}^{L}\frac{K_l(\phi)}{n_l}\left(\|G^l - G^{l-1}\|^2_{W^{1,4}(\pi\times\mathbb{U})} + 1\right)$$

Large!

Complexity of low-fidelity generator - large!

Small!

Complexity of other integrands - small!

Assumptions:

1) We need the generators to have at least one derivative and four moments! ($W^{1,4}(\pi \times \mathbb{U})$)

Bound on the variance

Under some mild assumptions, we get:

$$\text{Var}\left[\ell_{\text{ML-NLE}}(\phi)\right] \leq \frac{K_0(\phi)}{n_0}\left(\|G^0\|^4_{W^{1,4}(\pi\times\mathbb{U})}+1\right)+\sum_{l=1}^{L}\frac{K_l(\phi)}{n_l}\left(\|G^l-G^{l-1}\|^2_{W^{1,4}(\pi\times\mathbb{U})}+1\right)$$

Large!

Complexity of low-fidelity generator - large!

Small!

Complexity of other integrands - small!

Assumptions:

1) We need the generators to have at least one derivative and four moments! ($W^{1,4}(\pi \times \mathbb{U})$)
2) We need π and $\mathbb{U}$ to satisfy a Poincaré inequality (ok for Gaussian, uniform, etc..)

Bound on the variance

Under some mild assumptions, we get:

$$\text{Var}\left[\ell_{\text{ML-NLE}}(\phi)\right] \leq \frac{K_0(\phi)}{n_0}\left(\|G^0\|^4_{W^{1,4}(\pi\times\mathbb{U})}+1\right)+\sum_{l=1}^{L}\frac{K_l(\phi)}{n_l}\left(\|G^l-G^{l-1}\|^2_{W^{1,4}(\pi\times\mathbb{U})}+1\right)$$

Large!

Complexity of low-fidelity generator - large!

Small!

Complexity of other integrands - small!

Assumptions:

1) We need the generators to have at least one derivative and four moments! ($W^{1,4}(\pi\times\mathbb{U})$)
2) We need π and $\mathbb{U}$ to satisfy a Poincaré inequality (ok for Gaussian, uniform, etc..)
3) The surrogate $q_\phi(\cdot\,|\,\theta)$ has a Lipschitz gradient locally, and does not blow up too fast.

Bound on the variance

Can use this to determine optimal samples per level!

Under some mild assumptions, we get:

$$\text{Var}\left[\ell_{\text{ML-NLE}}(\phi)\right] \leq \frac{K_0(\phi)}{n_0}\left(\|G^0\|^4_{W^{1,4}(\pi\times\mathbb{U})} + 1\right) + \sum_{l=1}^{L}\frac{K_l(\phi)}{n_l}\left(\|G^l - G^{l-1}\|^2_{W^{1,4}(\pi\times\mathbb{U})} + 1\right)$$

Large!

Complexity of low-fidelity generator - large!

Small!

Complexity of other integrands - small!

Assumptions:

1) We need the generators to have at least one derivative and four moments! $(W^{1,4}(\pi \times \mathbb{U}))$

2) We need π and $\mathbb{U}$ to satisfy a Poincaré inequality (ok for Gaussian, uniform, etc..)

3) The surrogate $q_\phi(\cdot \mid \theta)$ has a Lipschitz gradient locally, and does not blow up too fast.

Simulations per level

Under some mild regularity conditions, we can find the optimal number of simulations per level assuming we have a maximum computational budget of C_{budget}:

$$n_0^\star \propto \frac{C_{\text{budget}}}{\sqrt{C_0}}\sqrt{\|G^0\|^4_{W^{1,4}(\pi\times\mathbb{U})} + 1}, \qquad n_l^\star \propto \frac{C_{\text{budget}}}{\sqrt{C_l + C_{l+1}}}\sqrt{\|G^l - G^{l-1}\|^2_{W^{1,4}(\pi\times\mathbb{U})} + 1}\,.$$

Simulations per level

Under some mild regularity conditions, we can find the optimal number of simulations per level assuming we have a maximum computational budget of C_{budget}:

$$n_0^\star \propto \frac{C_{\text{budget}}}{\sqrt{C_0}} \sqrt{\|G^0\|^4_{W^{1,4}(\pi \times \mathbb{U})} + 1}, \qquad n_l^\star \propto \frac{C_{\text{budget}}}{\sqrt{C_l + C_{l+1}}} \sqrt{\|G^l - G^{l-1}\|^2_{W^{1,4}(\pi \times \mathbb{U})} + 1}.$$

The more ‘complex’ the generator
(or the difference in generators),
the more simulations we need.

Simulations per level

Under some mild regularity conditions, we can find the optimal number of simulations per level assuming we have a maximum computational budget of C_{budget}:

$$n_0^\star \propto \frac{C_{\text{budget}}}{\sqrt{C_0}} \sqrt{\|G^0\|^4_{W^{1,4}(\pi \times \mathbb{U})} + 1}, \qquad n_l^\star \propto \frac{C_{\text{budget}}}{\sqrt{C_l + C_{l+1}}} \sqrt{\|G^l - G^{l-1}\|^2_{W^{1,4}(\pi \times \mathbb{U})} + 1}.$$

The larger the cost of simulations at this level, the less simulations we can afford.

Simulations per level

Under some mild regularity conditions, we can find the optimal number of simulations per level assuming we have a maximum computational budget of C_{budget}:

$$n_0^\star \propto \frac{C_{\text{budget}}}{\sqrt{C_0}} \sqrt{\|G^0\|^4_{W^{1,4}(\pi\times\mathbb{U})} + 1}\,, \qquad n_l^\star \propto \frac{C_{\text{budget}}}{\sqrt{C_l + C_{l+1}}} \sqrt{\|G^l - G^{l-1}\|^2_{W^{1,4}(\pi\times\mathbb{U})} + 1}\,.$$

The larger the budget, the more simulations we can afford.

Simulations per level

Under some mild regularity conditions, we can find the optimal number of simulations per level assuming we have a maximum computational budget of C_{budget}:

$$n_0^\star \propto \frac{C_{\text{budget}}}{\sqrt{C_0}} \sqrt{\|G^0\|^4_{W^{1,4}(\pi \times \mathbb{U})} + 1}, \qquad n_l^\star \propto \frac{C_{\text{budget}}}{\sqrt{C_l + C_{l+1}}} \sqrt{\|G^l - G^{l-1}\|^2_{W^{1,4}(\pi \times \mathbb{U})} + 1}\,.$$

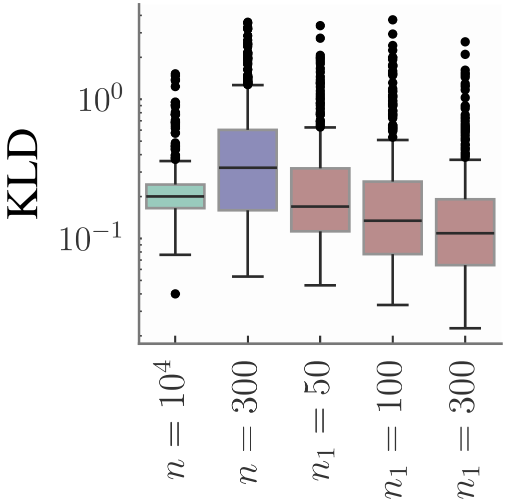
Note that these expressions contain a lot of quantities we may not know a-priori, but it is still indicative and helpful for selecting which simulations to run in practice.

G-and-k distribution

$$G^l_\theta(u) = \theta_1 + \theta_2\left(1 + 0.8\left(\frac{1-\exp(-\theta_3 z_l(u))}{1+\exp(-\theta_3 z_l(u))}\right)\right)\left(1+z_l(u)^2\right)^{\log(\theta_4)} z_l(u),$$

$$z_1(u) = \Phi^{-1}(u) = \sqrt{2}\,\mathrm{erf}^{-1}(2u-1), \qquad u \sim \mathrm{Unif}([0,1]),$$

$$z_0(u) := \sqrt{2}\,\mathrm{erf}^{-1}_{\mathrm{low}}(2u-1), \quad \mathrm{erf}^{-1}_{\mathrm{low}}(v) := \frac{\pi}{2}\left(u + \frac{\pi}{12}u^3\right).$$



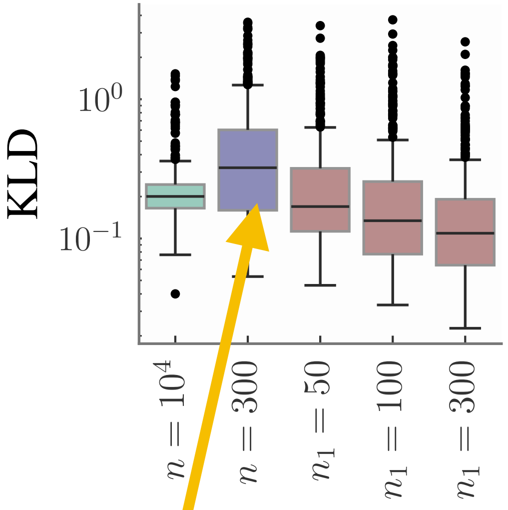
Prangle, D. (2020). gk: An R Package for the g-and-k and generalised g-and-h distributions. The R Journal, 12(1):7.

G-and-k distribution

$$G_\theta^l(u) = \theta_1 + \theta_2 \left(1 + 0.8\left(\frac{1-\exp(-\theta_3 z_l(u))}{1+\exp(-\theta_3 z_l(u))}\right)\right)\left(1 + z_l(u)^2\right)^{\log(\theta_4)} z_l(u),$$

$$z_1(u) = \Phi^{-1}(u) = \sqrt{2}\,\mathrm{erf}^{-1}(2u-1), \qquad u \sim \mathrm{Unif}([0,1]),$$

$$z_0(u) := \sqrt{2}\,\mathrm{erf}_{\mathrm{low}}^{-1}(2u-1), \quad \mathrm{erf}_{\mathrm{low}}^{-1}(v) := \frac{\pi}{2}\left(u + \frac{\pi}{12}u^3\right).$$



High-fidelity only:
too few simulations!

Prangle, D. (2020). gk: An R Package for the g-and-k and generalised g-and-h distributions. The R Journal, 12(1):7.

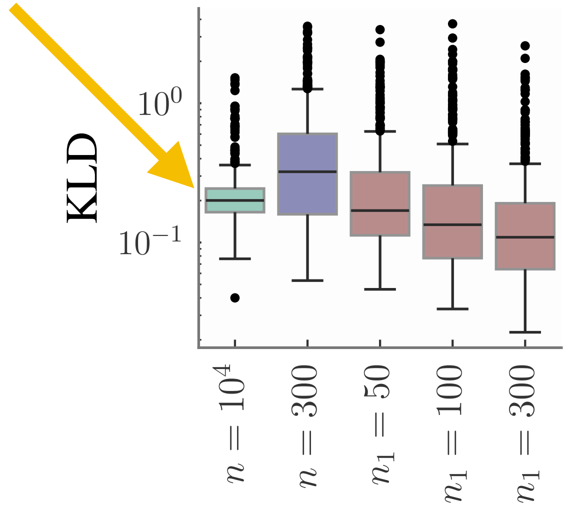
G-and-k distribution

$$G^l_\theta(u) = \theta_1 + \theta_2 \left(1 + 0.8\left(\frac{1-\exp(-\theta_3 z_l(u))}{1+\exp(-\theta_3 z_l(u))}\right)\right)\left(1+z_l(u)^2\right)^{\log(\theta_4)} z_l(u),$$

$$z_1(u) = \Phi^{-1}(u) = \sqrt{2}\,\mathrm{erf}^{-1}(2u-1), \qquad u \sim \mathrm{Unif}([0,1]),$$

$$z_0(u) := \sqrt{2}\,\mathrm{erf}^{-1}_{\mathrm{low}}(2u-1), \quad \mathrm{erf}^{-1}_{\mathrm{low}}(v) := \frac{\pi}{2}\left(u + \frac{\pi}{12}u^3\right).$$

Low-fidelity only:
Many simulations,
but low quality!



Prangle, D. (2020). gk: An R Package for the g-and-k and generalised g-and-h distributions. The R Journal, 12(1):7.

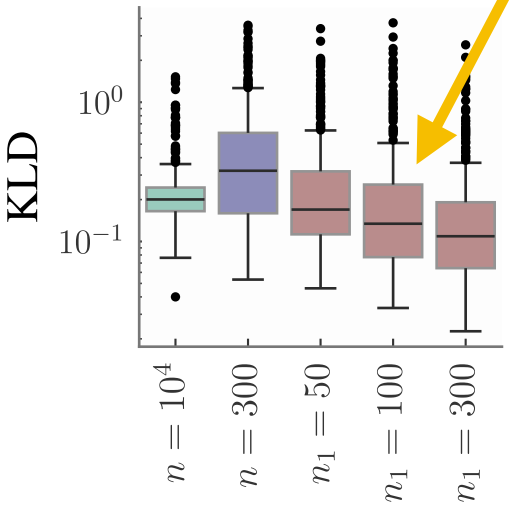
G-and-k distribution

$$G_\theta^l(u) = \theta_1 + \theta_2 \left(1 + 0.8\left(\frac{1-\exp(-\theta_3 z_l(u))}{1+\exp(-\theta_3 z_l(u))}\right)\right)\left(1+z_l(u)^2\right)^{\log(\theta_4)} z_l(u),$$

$$z_1(u) = \Phi^{-1}(u) = \sqrt{2}\mathrm{erf}^{-1}(2u-1), \qquad u \sim \mathrm{Unif}([0,1]),$$

$$z_0(u) := \sqrt{2}\mathrm{erf}_{\mathrm{low}}^{-1}(2u-1), \quad \mathrm{erf}_{\mathrm{low}}^{-1}(v) := \frac{\pi}{2}\left(u + \frac{\pi}{12}u^3\right).$$

ML-NLE: both many simulations and high quality!



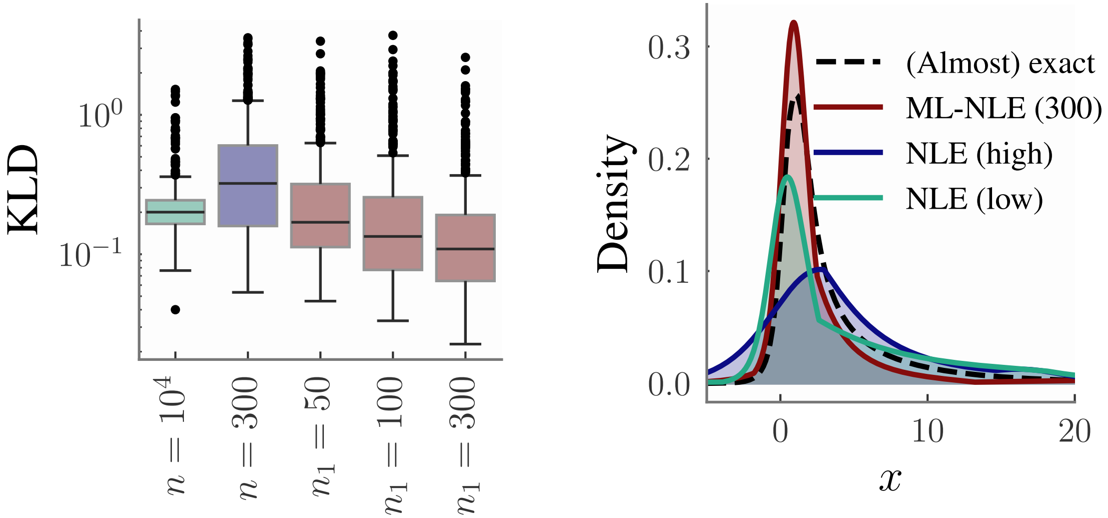
Prangle, D. (2020). gk: An R Package for the g-and-k and generalised g-and-h distributions. The R Journal, 12(1):7.

G-and-k distribution

$$G_\theta^l(u) = \theta_1 + \theta_2\left(1 + 0.8\left(\frac{1-\exp(-\theta_3 z_l(u))}{1+\exp(-\theta_3 z_l(u))}\right)\right)\left(1+z_l(u)^2\right)^{\log(\theta_4)} z_l(u),$$

$$z_1(u) = \Phi^{-1}(u) = \sqrt{2}\,\mathrm{erf}^{-1}(2u-1), \qquad u \sim \mathrm{Unif}([0,1]),$$

$$z_0(u) := \sqrt{2}\,\mathrm{erf}^{-1}_{\mathrm{low}}(2u-1), \quad \mathrm{erf}^{-1}_{\mathrm{low}}(v) := \frac{\pi}{2}\left(u + \frac{\pi}{12}u^3\right).$$



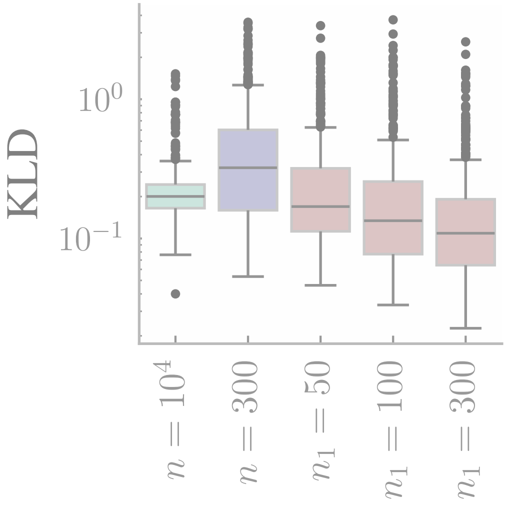
Prangle, D. (2020). gk: An R Package for the g-and-k and generalised g-and-h distributions. The R Journal, 12(1):7.

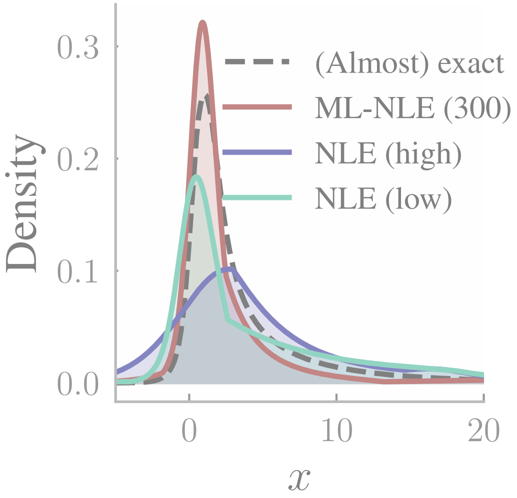
G-and-k distribution

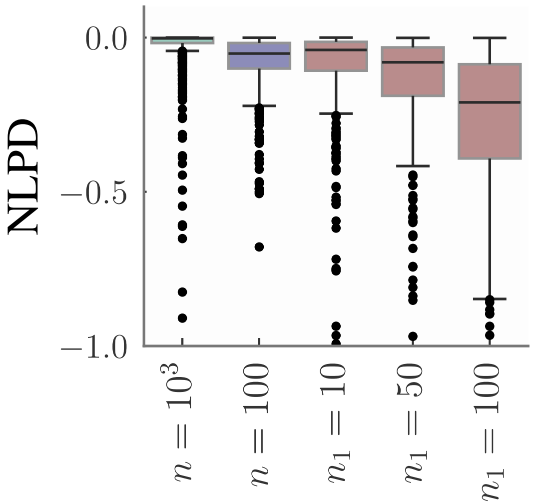
$$G_\theta^l(u) = \theta_1 + \theta_2 \left(1 + 0.8\left(\frac{1-\exp(-\theta_3 z_l(u))}{1+\exp(-\theta_3 z_l(u))}\right)\right)\left(1 + z_l(u)^2\right)^{\log(\theta_4)} z_l(u),$$

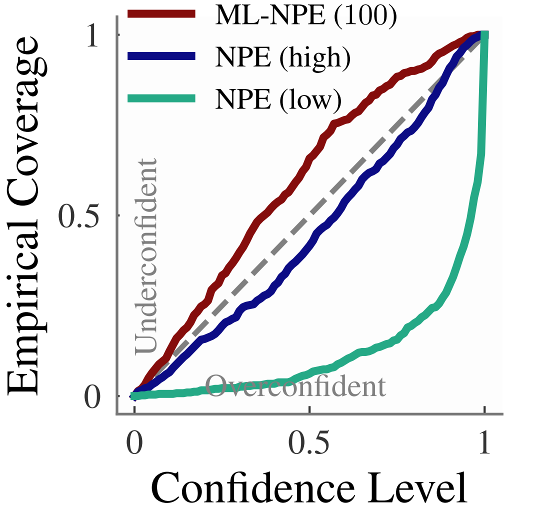
$$z_1(u) = \Phi^{-1}(u) = \sqrt{2}\,\mathrm{erf}^{-1}(2u-1), \qquad u \sim \mathrm{Unif}([0,1]),$$

$$z_0(u) := \sqrt{2}\,\mathrm{erf}_{\mathrm{low}}^{-1}(2u-1), \quad \mathrm{erf}_{\mathrm{low}}^{-1}(v) := \frac{\pi}{2}\left(u + \frac{\pi}{12}u^3\right).$$









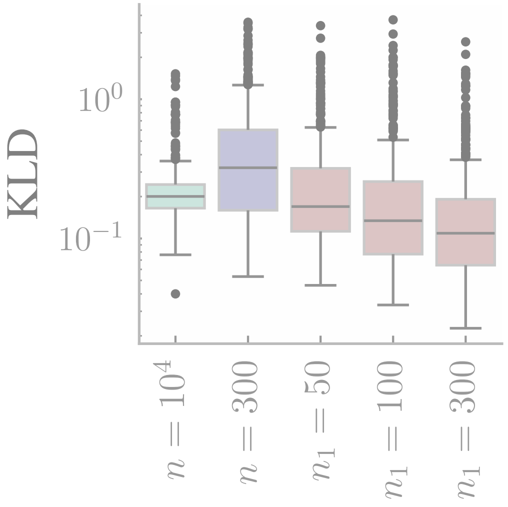
Prangle, D. (2020). gk: An R Package for the g-and-k and generalised g-and-h distributions. The R Journal, 12(1):7.

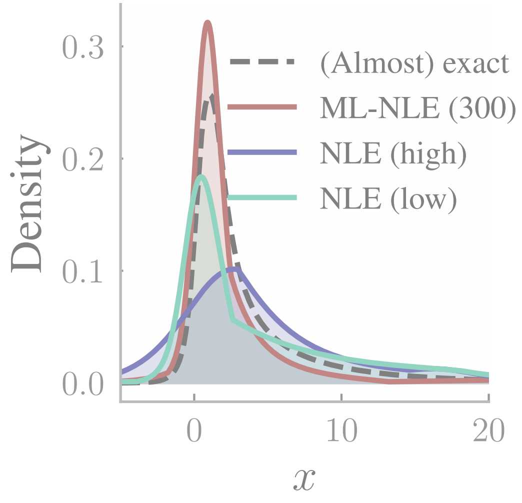
G-and-k distribution

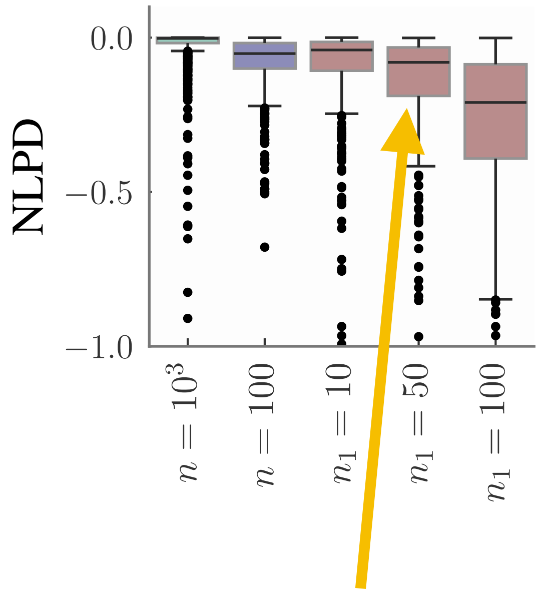
$$G_\theta^l(u) = \theta_1 + \theta_2 \left(1 + 0.8\left(\frac{1-\exp(-\theta_3 z_l(u))}{1+\exp(-\theta_3 z_l(u))}\right)\right)\left(1+z_l(u)^2\right)^{\log(\theta_4)} z_l(u),$$

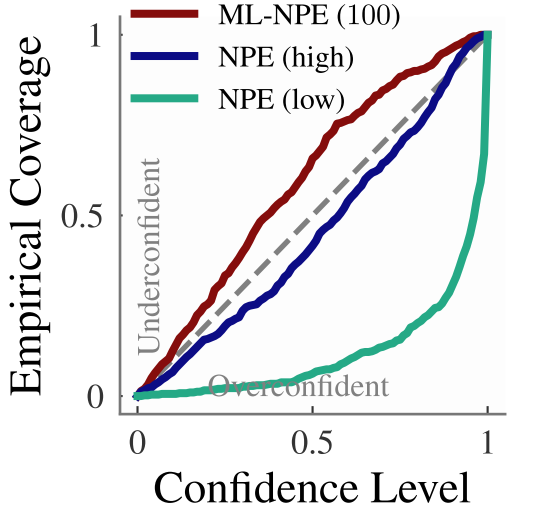
$$z_1(u) = \Phi^{-1}(u) = \sqrt{2}\,\mathrm{erf}^{-1}(2u-1), \qquad u \sim \mathrm{Unif}([0,1]),$$

$$z_0(u) := \sqrt{2}\,\mathrm{erf}^{-1}_{\mathrm{low}}(2u-1), \quad \mathrm{erf}^{-1}_{\mathrm{low}}(v) := \frac{\pi}{2}\left(u + \frac{\pi}{12}u^3\right).$$









ML-NPE: Similar conclusion!

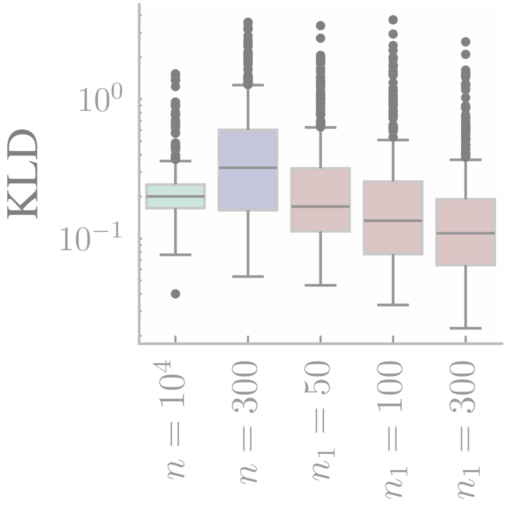
Prangle, D. (2020). gk: An R Package for the g-and-k and generalised g-and-h distributions. The R Journal, 12(1):7.

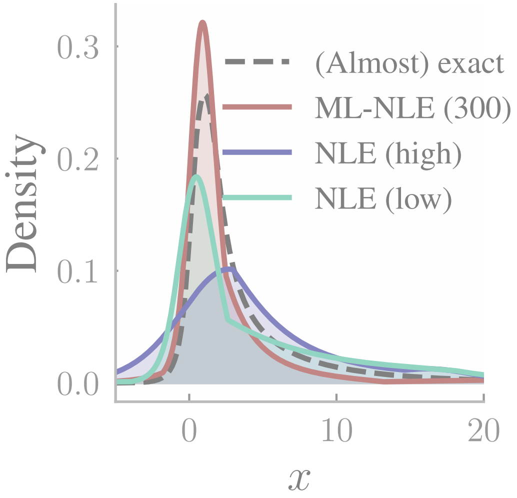
G-and-k distribution

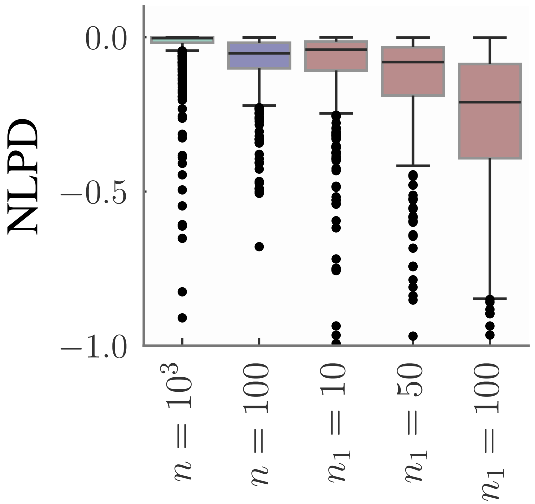
$$G_\theta^l(u) = \theta_1 + \theta_2\left(1 + 0.8\left(\frac{1-\exp(-\theta_3 z_l(u))}{1+\exp(-\theta_3 z_l(u))}\right)\right)\left(1+z_l(u)^2\right)^{\log(\theta_4)} z_l(u),$$

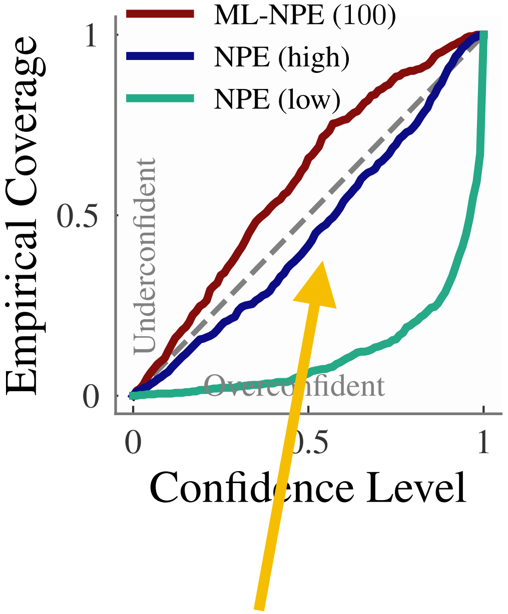
$$z_1(u) = \Phi^{-1}(u) = \sqrt{2}\,\mathrm{erf}^{-1}(2u-1), \qquad u \sim \mathrm{Unif}([0,1]),$$

$$z_0(u) := \sqrt{2}\,\mathrm{erf}^{-1}_{\mathrm{low}}(2u-1), \quad \mathrm{erf}^{-1}_{\mathrm{low}}(v) := \frac{\pi}{2}\left(u+\frac{\pi}{12}u^3\right).$$





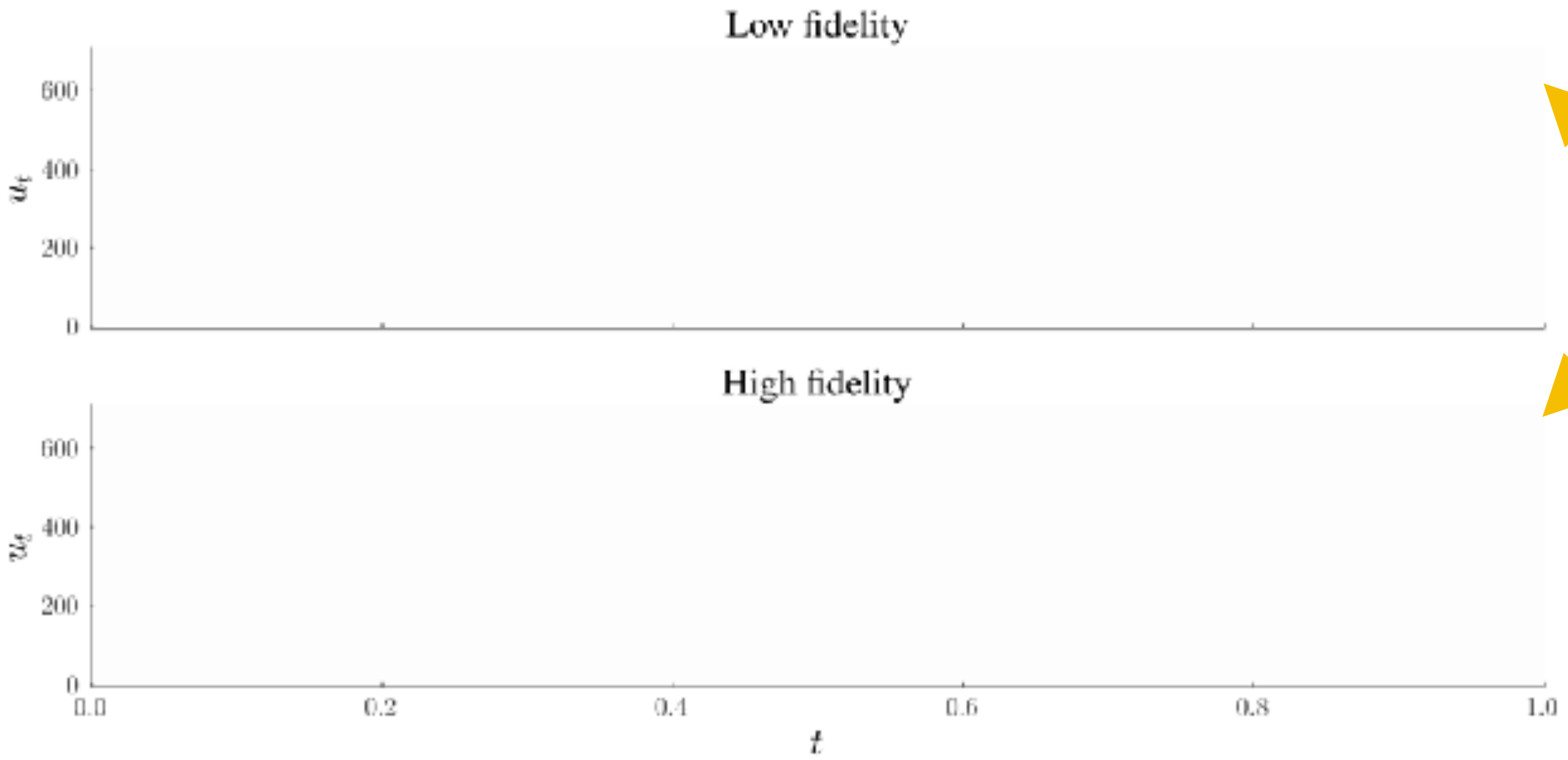




Coverage slightly cautious

Prangle, D. (2020). gk: An R Package for the g-and-k and generalised g-and-h distributions. The R Journal, 12(1):7.

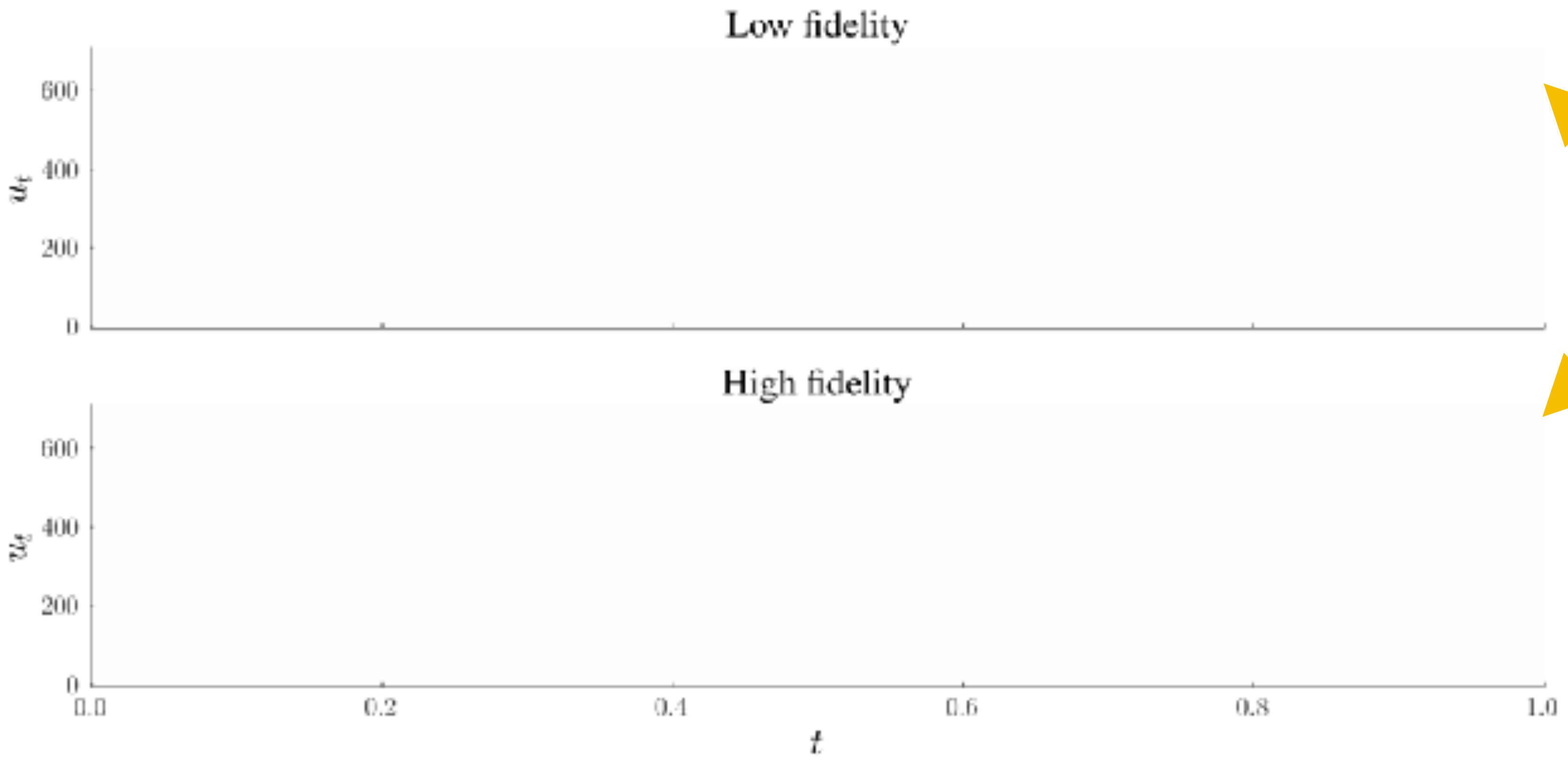
Toggle-switch models for genes (d=1, p=7)



You only observe the last time step

Bonassi, F. V., You, L., & West, M. (2011). Bayesian learning from marginal data in bionetwork models. *Statistical Applications in Genetics and Molecular Biology*, *10*(1).

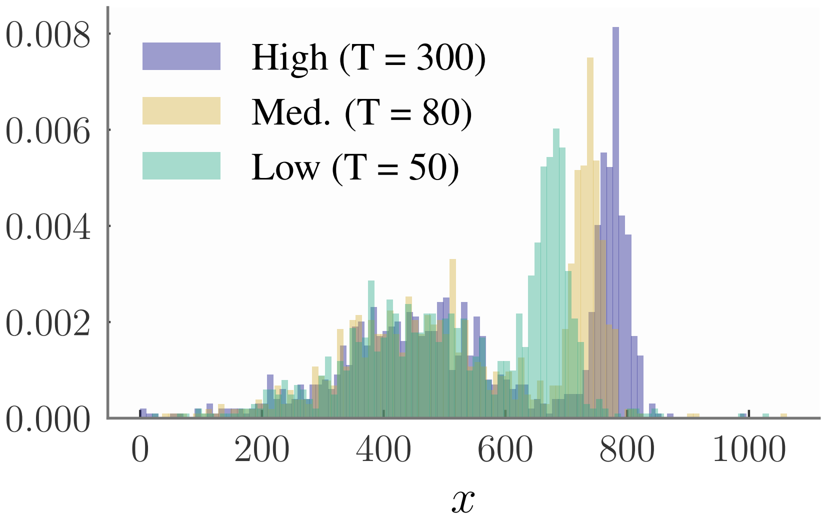
Toggle-switch models for genes (d=1, p=7)



You only observe the last time step

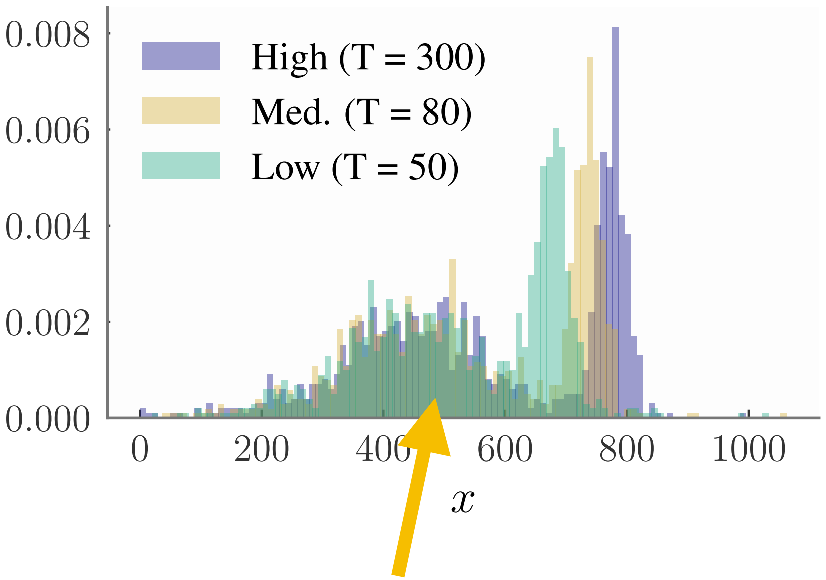
Bonassi, F. V., You, L., & West, M. (2011). Bayesian learning from marginal data in bionetwork models. *Statistical Applications in Genetics and Molecular Biology*, *10*(1).

Toggle-switch models for genes (d=1, p=7)



Bonassi, F. V., You, L., & West, M. (2011). Bayesian learning from marginal data in bionetwork models. *Statistical Applications in Genetics and Molecular Biology*, *10*(1).

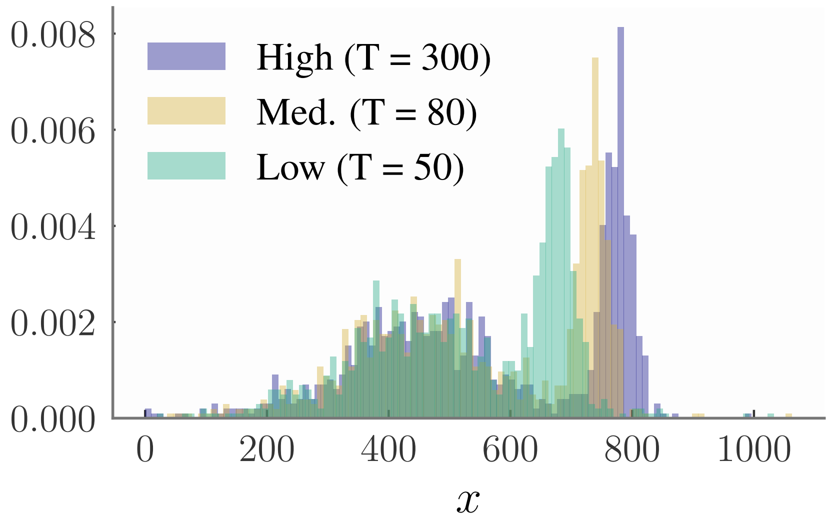
Toggle-switch models for genes (d=1, p=7)

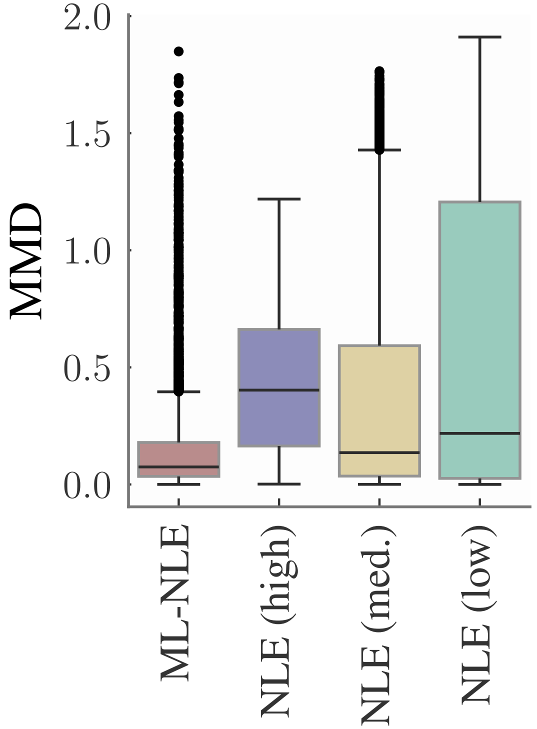


Observations bi-modal, with second mode only well approximated for high-fidelity levels

Bonassi, F. V., You, L., & West, M. (2011). Bayesian learning from marginal data in bionetwork models. *Statistical Applications in Genetics and Molecular Biology*, *10*(1).

Toggle-switch models for genes (d=1, p=7)

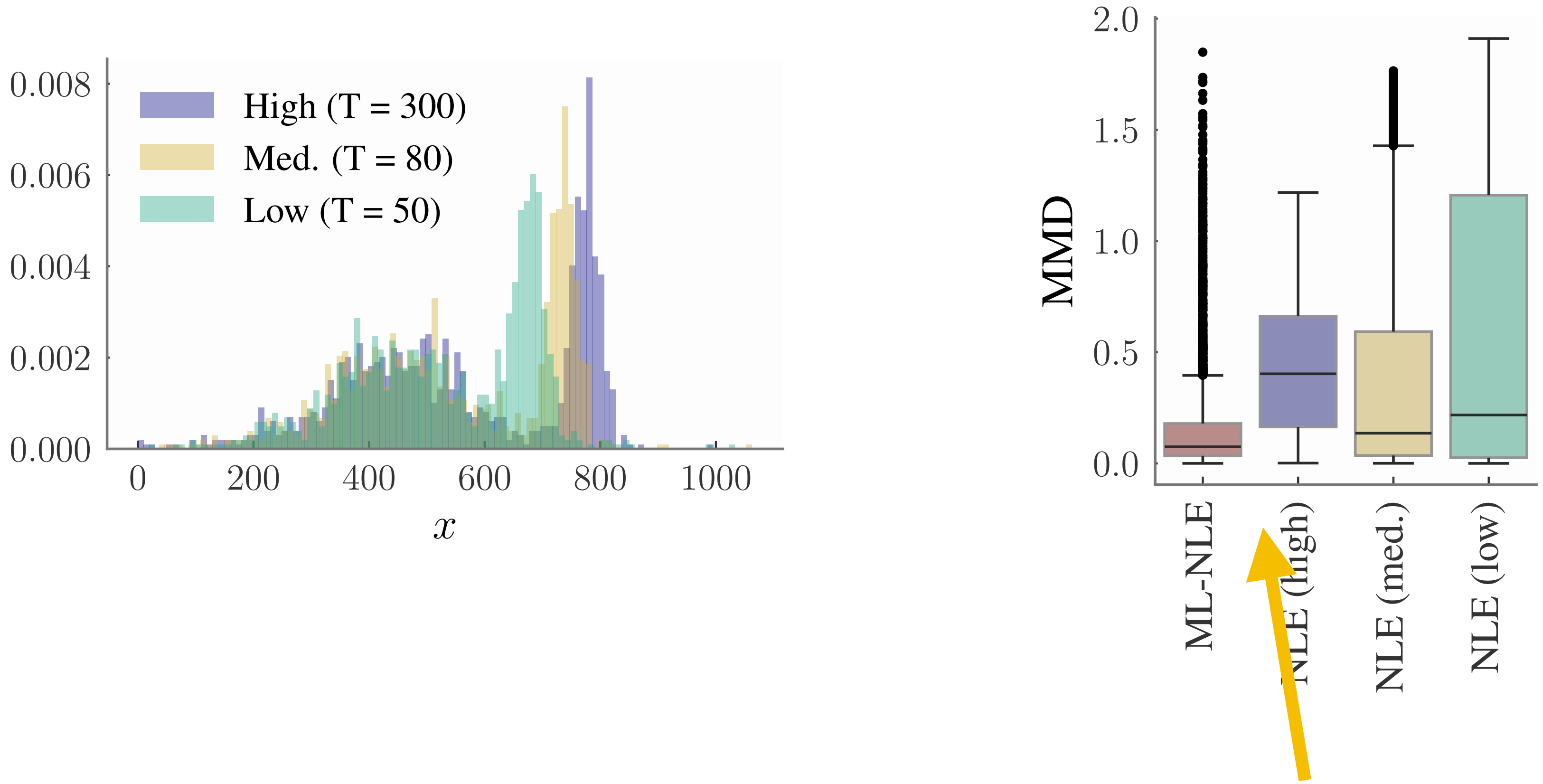




$n_0 = 10000$
$n_1 = 500$
$n_2 = 300$

Bonassi, F. V., You, L., & West, M. (2011). Bayesian learning from marginal data in bionetwork models. *Statistical Applications in Genetics and Molecular Biology*, *10*(1).

Toggle-switch models for genes (d=1, p=7)

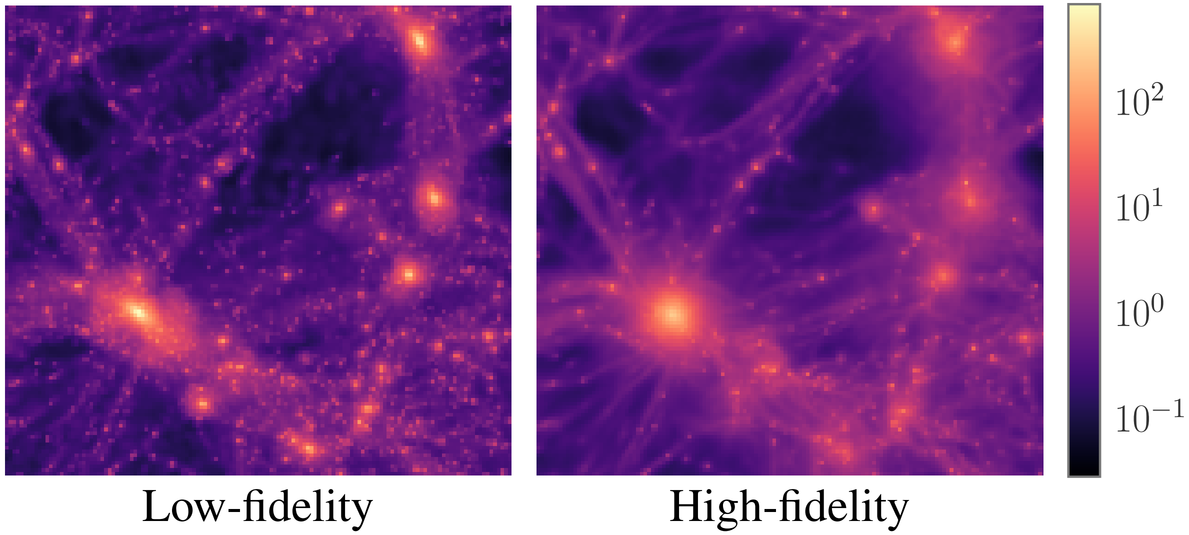


$n_0 = 10000$
$n_1 = 500$
$n_2 = 300$

ML-NLE benefits from low-fidelity simulations for first mode but also from high-fidelity simulations for second mode

Bonassi, F. V., You, L., & West, M. (2011). Bayesian learning from marginal data in bionetwork models. *Statistical Applications in Genetics and Molecular Biology*, *10*(1).

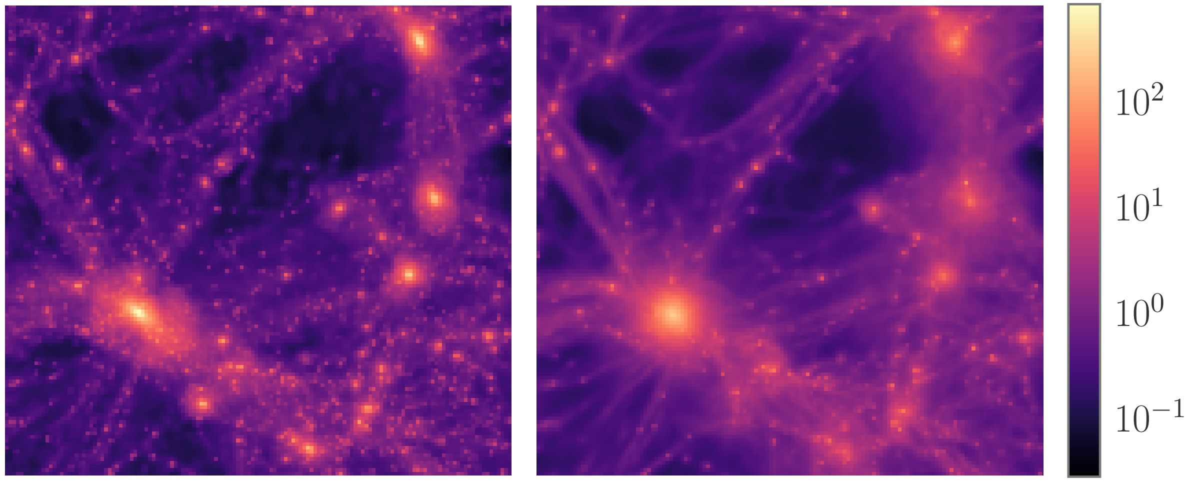
Back to cosmology…. (d=39, p=1)



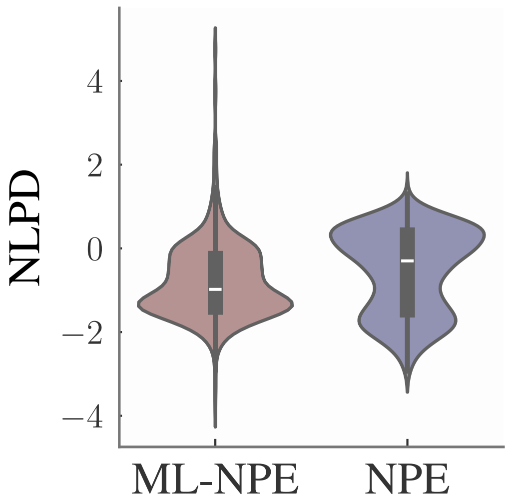
NPE: $n = 20$ (all high fidelity!)

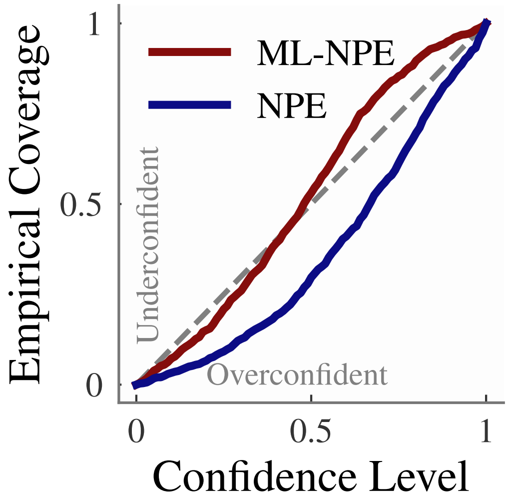
ML-NPE: $n_0 = 20, \quad n_1 = 980$

Back to cosmology…. (d=39, p=1)



Low-fidelity High-fidelity

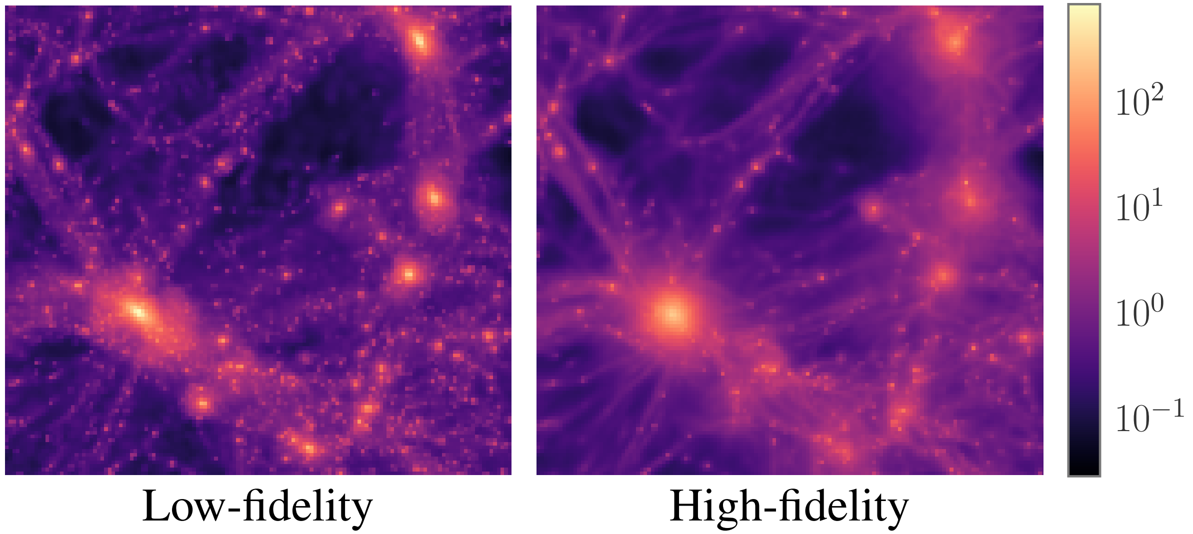


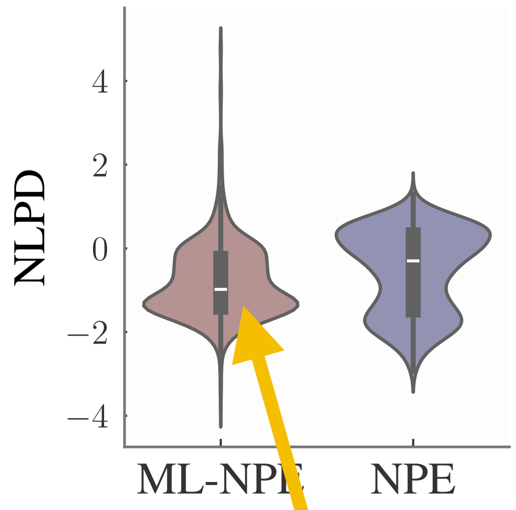


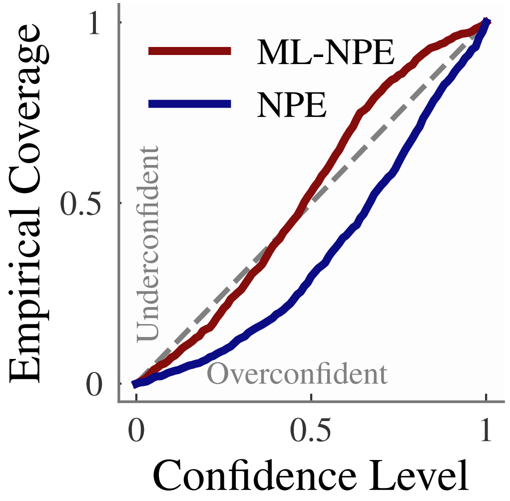
NPE: $n = 20$ (all high fidelity!)

ML-NPE: $n_0 = 20, \quad n_1 = 980$

Back to cosmology.... (d=39, p=1)





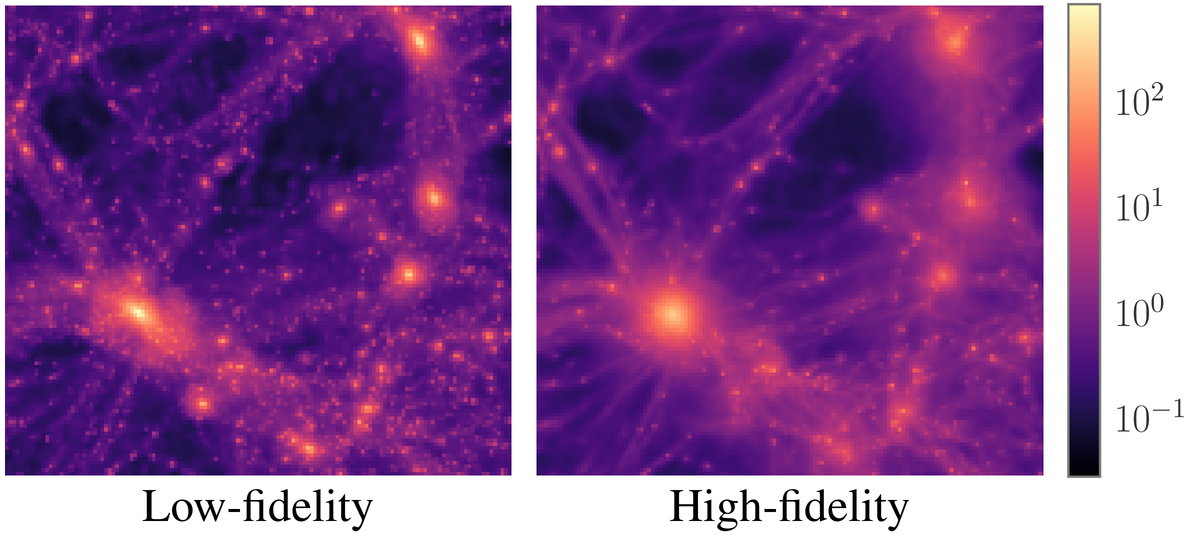


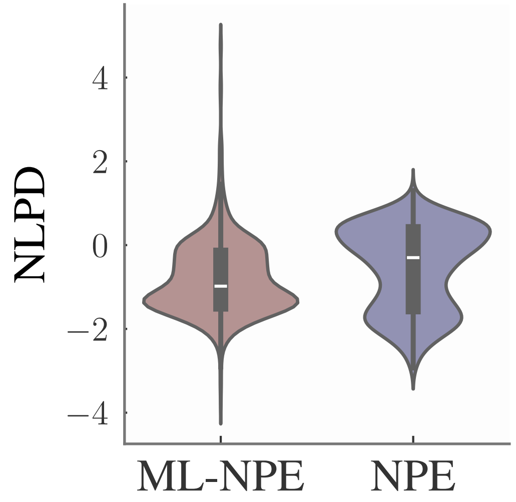
Improve fit of the surrogate posterior!

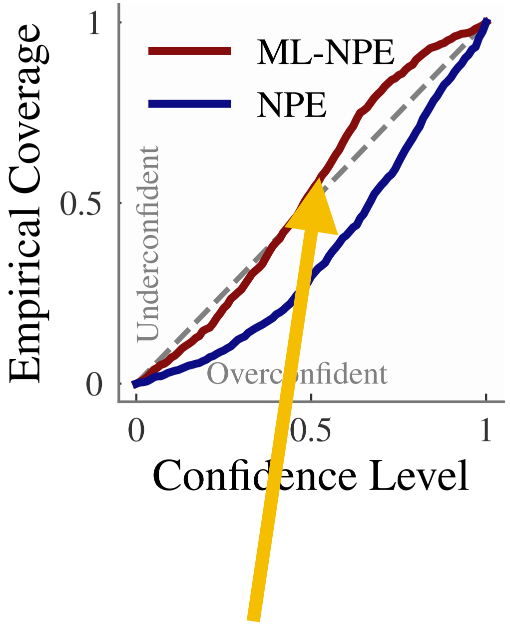
NPE: $n = 20$ (all high fidelity!)

ML-NPE: $n_0 = 20, \quad n_1 = 980$

Back to cosmology…. (d=39, p=1)







Improved calibration!

NPE: $n = 20$ (all high fidelity!)

ML-NPE: $n_0 = 20, \quad n_1 = 980$

Any Questions?

Paper: Hikida, Y., Bharti, A., Jeffrey, N. & Briol, F-X (2025). Multilevel neural simulation-based inference. arXiv:2506.06087 (to appear at NeurIPS?).

Code: https://github.com/yugahikida/multilevel-sbi

UCL

Cost-aware simulation-based inference

Paper: Bharti, A., Huang, D., Kaski, S., & **Briol, F.-X.** (2025). Cost-aware simulation-based inference. International Conference on Artificial Intelligence and Statistics, 28–36.

Code: https://github.com/huangdaolang/cost-aware-sbi

Challenge for SBI

Simulators can be really computationally expensive!

Challenge for SBI

Simulators can be really computationally expensive!

However we may not have an easy way to obtain low-fidelity simulators….

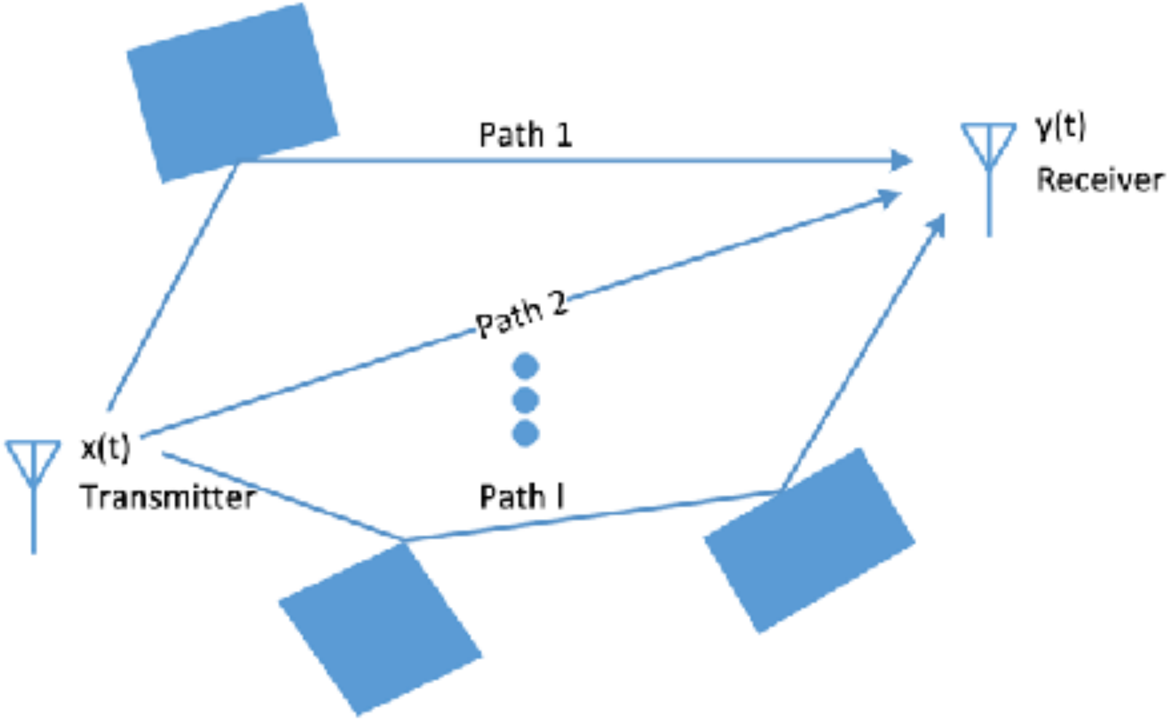
Challenge for SBI

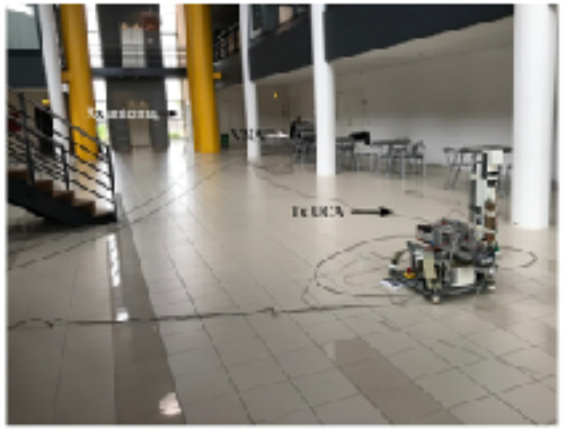
Simulators can be really computationally expensive!

However we may not have an easy way to obtain low-fidelity simulators….

→ We can adjust our sampling to sample less often from expensive parameterisations!

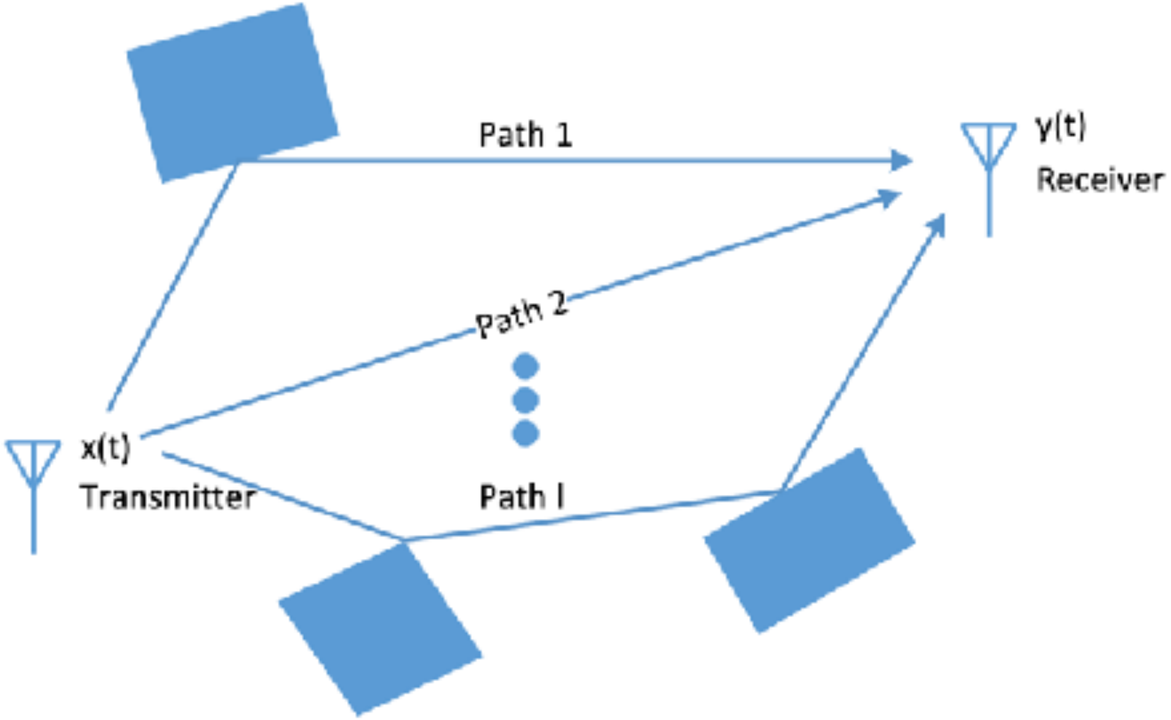
SBI for radio-propagation

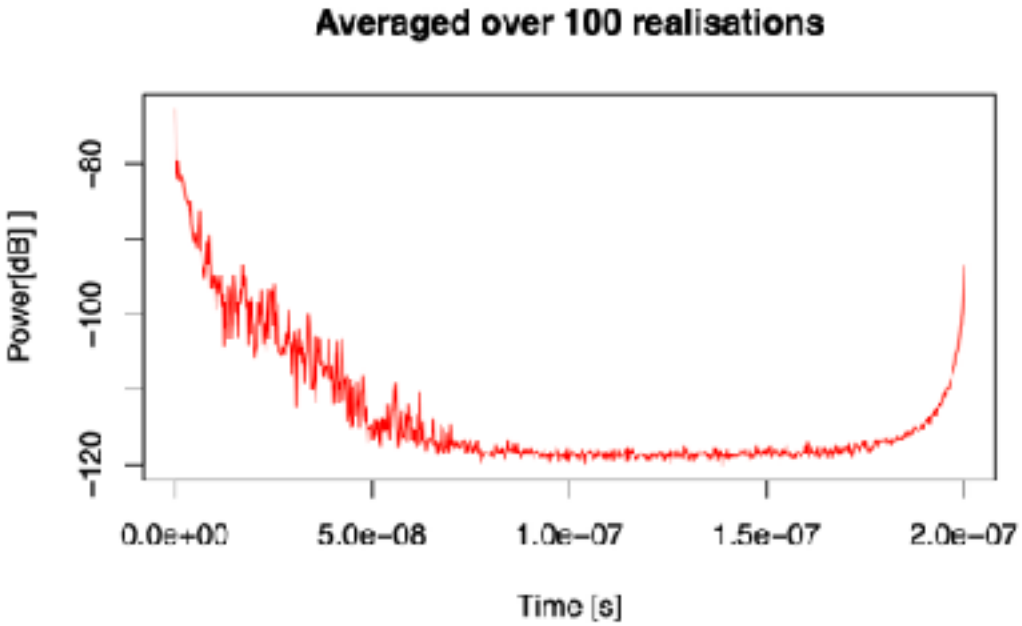




Bharti, A., **Briol, F-X.**, Pedersen, T. (2022). A general method for calibrating stochastic radio channel models with kernels. IEEE Transactions on Antennas and Propagation, vol. 70, no. 6, pp. 3986-4001, June 2022.

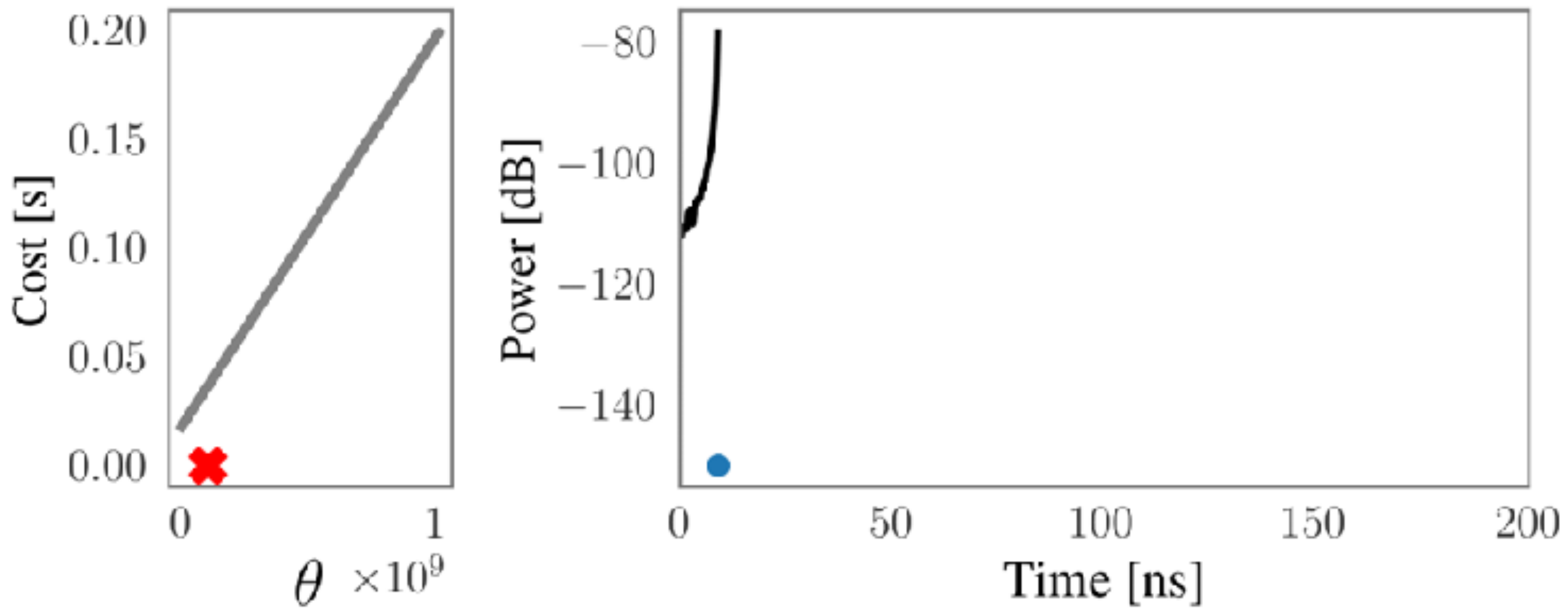
SBI for radio-propagation





Bharti, A., **Briol, F-X.**, Pedersen, T. (2022). A general method for calibrating stochastic radio channel models with kernels. IEEE Transactions on Antennas and Propagation, vol. 70, no. 6, pp. 3986-4001, June 2022.

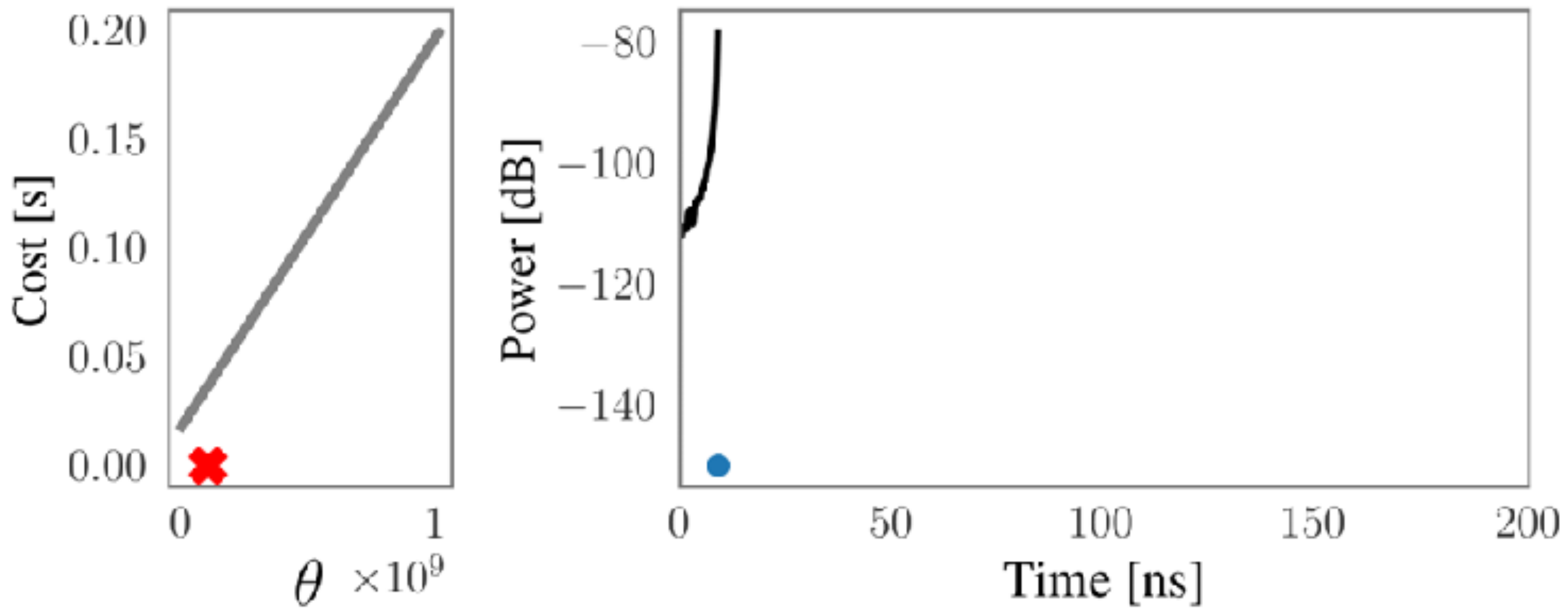
The cost of simulations is not constant…



Rate parameter of a Poisson process!

[https://fxbriol.github.io/images/ca-SBI.mp4]

The cost of simulations is not constant…



Rate parameter of a Poisson process!

[https://fxbriol.github.io/images/ca-SBI.mp4]

Neural likelihood estimation (NLE)

- **Step 1:** train a conditional density model $q_\phi(\cdot \mid \theta)$ to approximate the likelihood using samples from the prior ($\theta_1, \ldots, \theta_n \sim p(\theta)$) and simulator ($x_i \sim p(\cdot \mid \theta_i)$):

$$\hat{\phi}_n := \arg\min_{\phi \in \Phi} \ell_{\mathrm{NLE}}(\phi), \quad \ell_{\mathrm{NLE}}(\phi) = -\frac{1}{n}\sum_{i=1}^{n} \log q_\phi(x_i \mid \theta_i) \approx -\mathbb{E}_{\theta\sim p(\theta)}[\mathbb{E}_{x\sim p(x\mid\theta)}[\log q_\phi(x \mid \theta)]]$$

Neural likelihood estimation (NLE)

- **Step 1:** train a conditional density model $q_\phi(\cdot\,|\,\theta)$ to approximate the likelihood using samples from the prior ($\theta_1, \ldots, \theta_n \sim p(\theta)$) and simulator ($x_i \sim p(\cdot\,|\,\theta_i)$):

$$\hat{\phi}_n := \arg\min_{\phi\in\Phi} \ell_{\mathrm{NLE}}(\phi), \quad \ell_{\mathrm{NLE}}(\phi) = -\frac{1}{n}\sum_{i=1}^{n} \log q_\phi(x_i\,|\,\theta_i) \approx -\mathbb{E}_{\theta\sim p(\theta)}[\mathbb{E}_{x\sim p(x|\theta)}[\log q_\phi(x\,|\,\theta)]]$$

- **Step 2:** Do Bayes with approximate likelihood!

$$p_{\mathrm{NLE}}(\theta\,|\,y_1, \ldots, y_n) \propto \prod_{i=1}^{n} q_{\hat{\phi}_n}(y_i\,|\,\theta)p(\theta)$$

A cheaper step 1?

$$\ell_{\mathrm{NLE}}(\phi) = -\frac{1}{n}\sum_{i=1}^{n} \log q_\phi(x_i \,|\, \theta_i) \approx -\mathbb{E}_{\theta\sim p(\theta)}[\mathbb{E}_{x\sim p(\cdot|\theta)}[\log q_\phi(x \,|\, \theta)]]$$

Can we do this better/cheaper?!

A cheaper step 1?

$$\ell_{\mathrm{NLE}}(\phi) = -\frac{1}{n}\sum_{i=1}^{n}\log q_\phi(x_i\,|\,\theta_i) \approx -\mathbb{E}_{\theta\sim p(\theta)}[\mathbb{E}_{x\sim p(\cdot|\theta)}[\log q_\phi(x\,|\,\theta)]]$$

Can we do this better/cheaper?!

Idea:

- Let's make use of the cost function $c:\Theta \to \mathbb{R}$.

A cheaper step 1?

$$\ell_{\text{NLE}}(\phi) = -\frac{1}{n}\sum_{i=1}^{n} \log q_\phi(x_i \,|\, \theta_i) \approx -\mathbb{E}_{\theta \sim p(\theta)}[\mathbb{E}_{x \sim p(\cdot|\theta)}[\log q_\phi(x \,|\, \theta)]]$$

Can we do this better/cheaper?!

Idea:

- Let's make use of the cost function $c : \Theta \to \mathbb{R}$.
- We can try to sample less often in expensive regions …… but we still want to target the right objective.

Importance sampling

$$\mu = \int_{\Theta} f(\theta)\pi(\theta)d\theta$$

Importance sampling

$$\mu = \int_\Theta f(\theta)\pi(\theta)d\theta \quad = \int_\Theta f(\theta)\frac{\pi(\theta)}{\tilde{\pi}(\theta)}\tilde{\pi}(\theta)d\theta$$

Importance sampling

$$\mu = \int_{\Theta} f(\theta)\pi(\theta)d\theta \quad = \int_{\Theta} f(\theta)\frac{\pi(\theta)}{\tilde{\pi}(\theta)}\tilde{\pi}(\theta)d\theta$$

$$\approx \sum_{i=1}^{N} w(\theta_i)f(\theta_i) \qquad \theta_1, \ldots, \theta_N \sim \tilde{\pi}$$

Importance sampling

$$\mu = \int_\Theta f(\theta)\pi(\theta)d\theta \quad = \int_\Theta f(\theta)\frac{\pi(\theta)}{\tilde{\pi}(\theta)}\tilde{\pi}(\theta)d\theta$$

$$\approx \sum_{i=1}^{N} w(\theta_i)f(\theta_i) \qquad \theta_1, \ldots, \theta_N \sim \tilde{\pi}$$

$$w_{\mathrm{IS}}(\theta_i) = \frac{1}{N}\frac{\pi(\theta_i)}{\tilde{\pi}(\theta_i)}$$

Importance sampling

$$\mu = \int_\Theta f(\theta)\pi(\theta)d\theta \quad = \int_\Theta f(\theta)\frac{\pi(\theta)}{\tilde{\pi}(\theta)}\tilde{\pi}(\theta)d\theta$$

$$\approx \sum_{i=1}^{N} w(\theta_i)f(\theta_i) \qquad \theta_1, \ldots, \theta_N \sim \tilde{\pi}$$

$$w_{\mathrm{IS}}(\theta_i) = \frac{1}{N}\frac{\pi(\theta_i)}{\tilde{\pi}(\theta_i)} \qquad w_{\mathrm{SNIS}}(\theta_i) = \frac{w_{\mathrm{IS}}(\theta_i)}{\sum_{j=1}^{N} w_{\mathrm{IS}}(\theta_j)}$$

Importance sampling

$$\mu = \int_\Theta f(\theta)\pi(\theta)d\theta \quad = \int_\Theta f(\theta)\frac{\pi(\theta)}{\tilde{\pi}(\theta)}\tilde{\pi}(\theta)d\theta$$

$$\approx \sum_{i=1}^{N} w(\theta_i)f(\theta_i) \qquad \theta_1,\ldots,\theta_N \sim \tilde{\pi}$$

$$w_{\mathrm{IS}}(\theta_i) = \frac{1}{N}\frac{\pi(\theta_i)}{\tilde{\pi}(\theta_i)} \qquad w_{\mathrm{SNIS}}(\theta_i) = \frac{w_{\mathrm{IS}}(\theta_i)}{\sum_{j=1}^{N} w_{\mathrm{IS}}(\theta_j)}$$

Question: How do you pick the importance distribution?

Cost-aware importance sampling

$$\tilde{\pi}_g(\theta) \propto \frac{\pi(\theta)}{g(c(\theta))},$$

Cost-aware importance sampling

$$\tilde{\pi}_g(\theta) \propto \frac{\boxed{\pi(\theta)}}{g(c(\theta))},$$

We want a distribution similar to our target π

Cost-aware importance sampling

$$\tilde{\pi}_g(\theta) \propto \frac{\pi(\theta)}{g(\boxed{c(\theta)})},$$

We do not want to sample often where the cost is large!

Cost-aware importance sampling

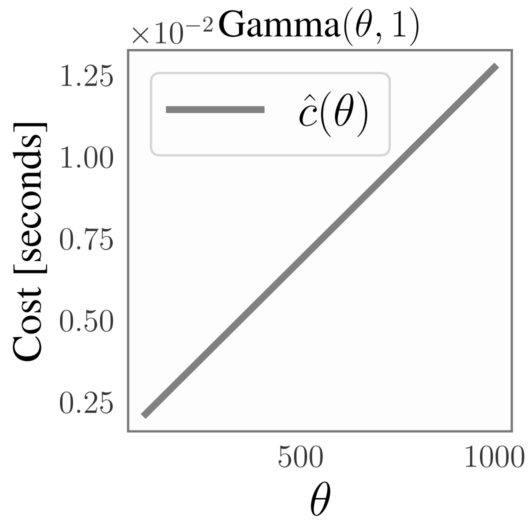
$$\tilde{\pi}_g(\theta) \propto \frac{\pi(\theta)}{\boxed{g}(c(\theta))},$$

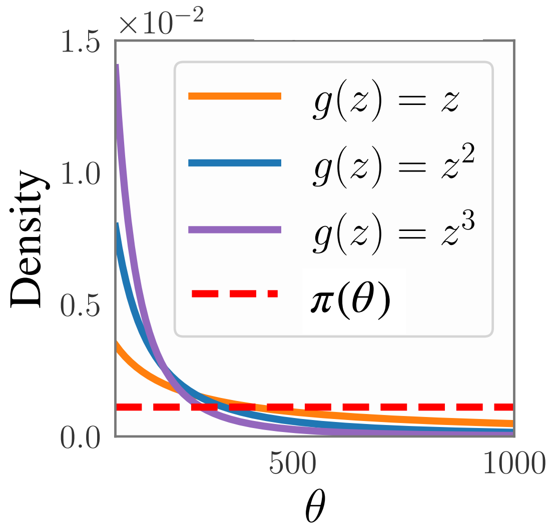
$g : (0,\infty) \to (0,\infty)$ taken to be non-decreasing.

Represents how much we dislike 'expensive' parameters!

Cost-aware importance sampling

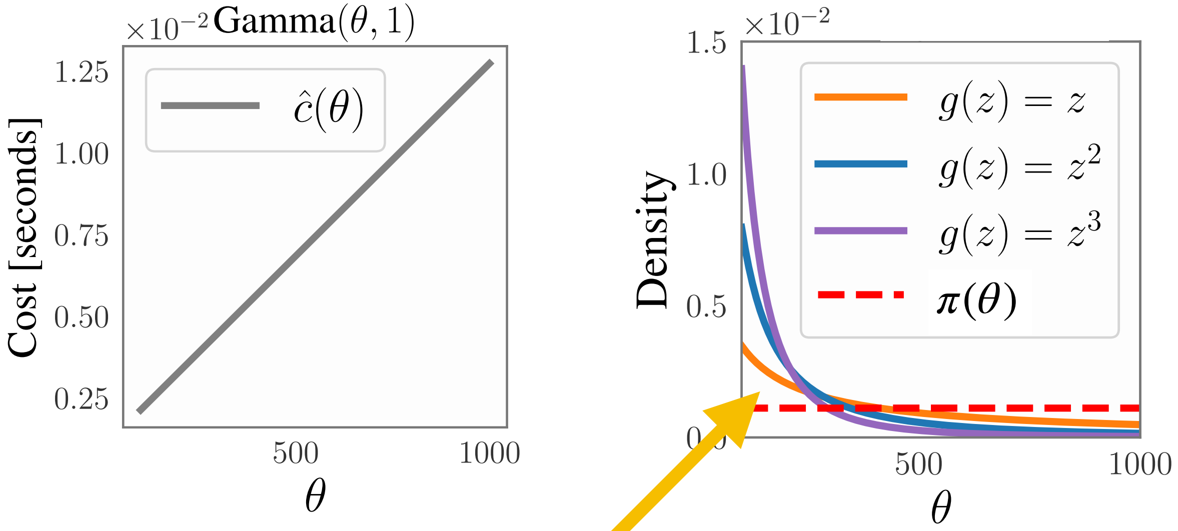
$$\tilde{\pi}_g(\theta) \propto \frac{\pi(\theta)}{g(c(\theta))},$$





Cost-aware importance sampling

$$\tilde{\pi}_g(\theta) \propto \frac{\pi(\theta)}{g(c(\theta))},$$



Upweight cheap region!

Cost-aware importance sampling

$$w(\theta) = \frac{1}{N}\frac{\pi(\theta)}{\tilde{\pi}_g(\theta)} = \frac{B\pi(\theta)g(c(\theta))}{N\pi(\theta)} \propto g(c(\theta))$$

Through $\tilde{\pi}_g$, we sample less often from expensive regions, so we need to up-weight expensive samples.

Cost-aware importance sampling

$$w(\theta) = \frac{1}{N}\frac{\pi(\theta)}{\tilde{\pi}_g(\theta)} = \frac{B\pi(\theta)g(c(\theta))}{N\pi(\theta)} \propto g(c(\theta))$$

$$w_{\mathrm{Ca}}(\theta_i) = \frac{w(\theta_i)}{\sum_{j=1}^{n} w(\theta_j)} = \frac{g(c(\theta_i))}{\sum_{j=1}^{n} g(c(\theta_j))}$$

We use SNIS weights

$$\mu = \int_{\Theta} f(\theta)\pi(\theta)d\theta \approx \sum_{i=1}^{n} w_{\mathrm{Ca}}(\theta_i)f(\theta_i) = \hat{\mu}_n^{\mathrm{Ca}}$$

Sampling from the cost-aware proposal

- We can use rejection sampling!

Sampling from the cost-aware proposal

- We can use rejection sampling!

Repeat until n samples are accepted:

1. Sample $\theta^\star \sim \pi(\theta)$.
2. Accept $\theta^\star$ as a sample from $\tilde{\pi}_g$ with probability $A(\theta)$.

Sampling from the cost-aware proposal

- We can use rejection sampling!

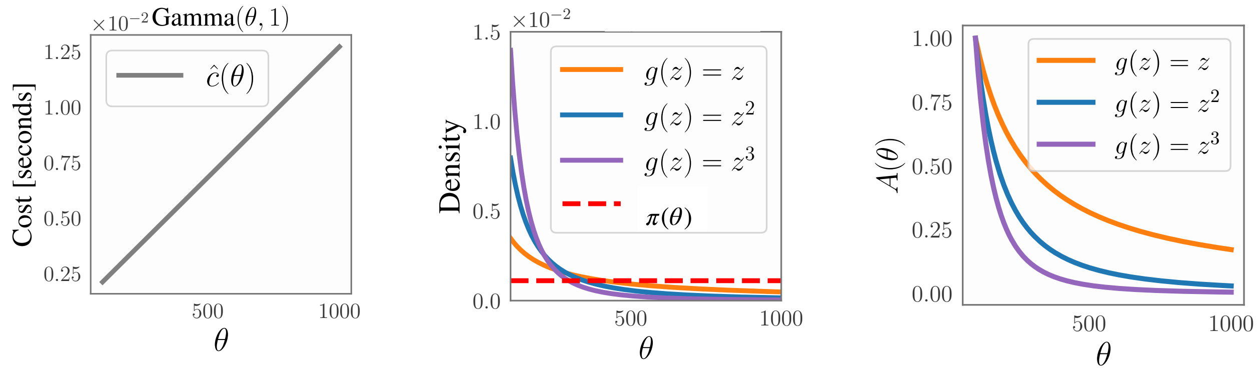
Repeat until n samples are accepted:

1. Sample $\theta^\star \sim \pi(\theta)$.
2. Accept $\theta^\star$ as a sample from $\tilde{\pi}_g$ with probability $A(\theta)$.

Proposition: Assume $g_{\min} := \inf_{\theta \in \Theta} g(c(\theta)) > 0$. Then

- $\tilde{\pi}_g$ is a density.
- The correct acceptance probability is $A(\theta) = \dfrac{g_{\min}}{g(c(\theta))}$

Sampling from the cost-aware proposal



- We can use rejection sampling!

Repeat until n samples are accepted:

1. Sample $\theta^\star \sim \pi(\theta)$.
2. Accept $\theta^\star$ as a sample from $\tilde{\pi}_g$ with probability $A(\theta)$.

Proposition: Assume $g_{\min} := \inf_{\theta \in \Theta} g(c(\theta)) > 0$. Then

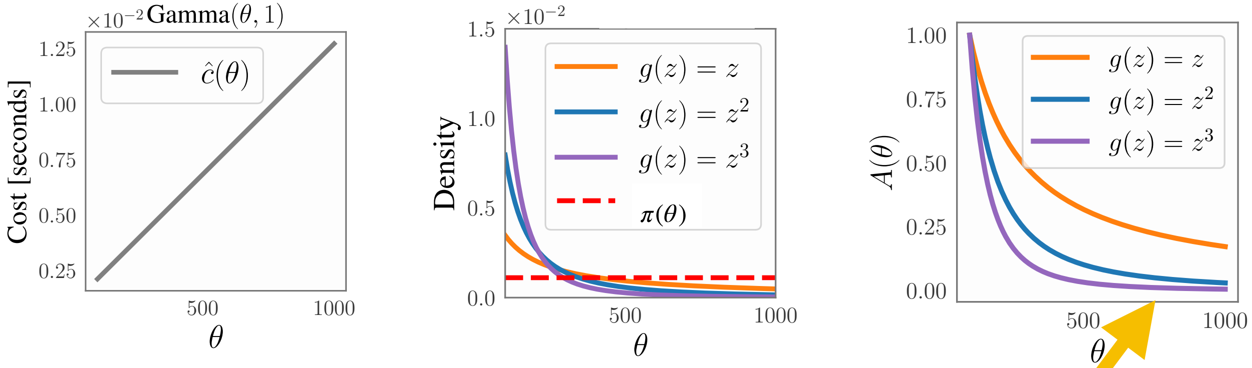
- $\tilde{\pi}_g$ is a density.
- The correct acceptance probability is $A(\theta) = \dfrac{g_{\min}}{g(c(\theta))}$

Sampling from the cost-aware proposal

- We can use rejection sampling!

Repeat until n samples are accepted:

1. Sample $\theta^\star \sim \pi(\theta)$.
2. Accept $\theta^\star$ as a sample from $\tilde{\pi}_g$ with probability $A(\theta)$.



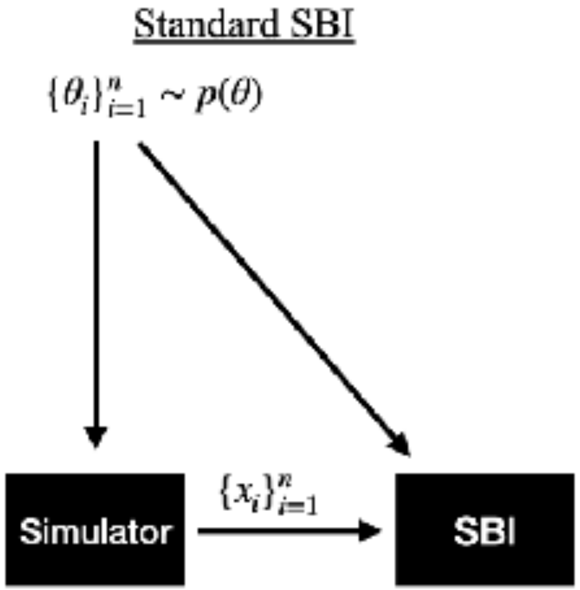
Being cost-averse decreases acceptance prob!

Proposition: Assume $g_{\min} := \inf_{\theta \in \Theta} g(c(\theta)) > 0$. Then

- $\tilde{\pi}_g$ is a density.
- The correct acceptance probability is $A(\theta) = \dfrac{g_{\min}}{g(c(\theta))}$

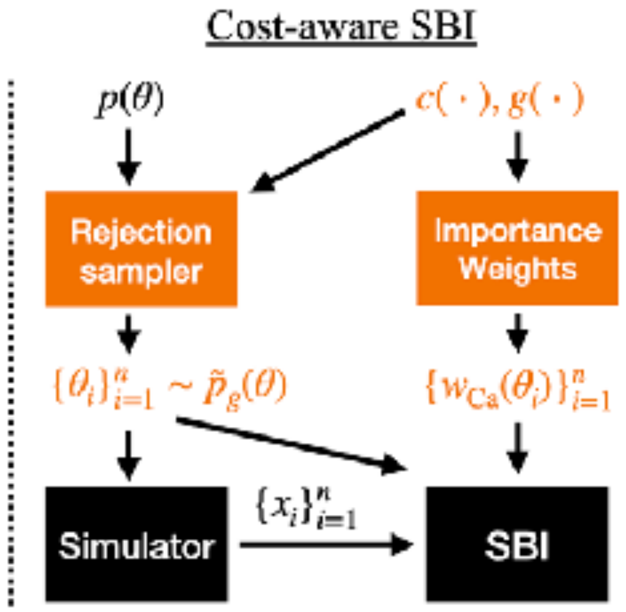
Putting it all together!

$$\ell_{\mathrm{NLE}}(\phi) = -\frac{1}{n}\sum_{i=1}^{n} \log q_\phi(\mathbf{x}_i \mid \theta_i), \qquad \theta_i \sim p(\theta), \mathbf{x}_i \sim p(\,\cdot \mid \theta)$$



Putting it all together!

$$\ell_{\mathrm{Ca-NLE}}(\phi) = -\frac{1}{n}\sum_{i=1}^{n} w_{\mathrm{Ca}}(\theta_i) \log q_\phi(x_i \mid \theta_i), \qquad \theta_i \sim \tilde{p}_g(\theta), x_i \sim p(\cdot \mid \theta)$$



Some reassuring results

Importance sampling can have infinite variance!!!

Some reassuring results

- Suppose that $g_{\max} = \sup_{\theta \in \Theta} g(c(\theta)) < \infty$. Then:

Some reassuring results

- Suppose that $g_{\max} = \sup_{\theta \in \Theta} g(c(\theta)) < \infty$. Then:
    1. The weights are bounded: $\frac{g_{\min}}{n g_{\max}} \leq w_{\mathrm{Ca}}(\theta_i) \leq \frac{g_{\max}}{n g_{\min}} \qquad \forall i \in \{1, \ldots, n\}$,

Some reassuring results

- Suppose that $g_{\max} = \sup_{\theta \in \Theta} g(c(\theta)) < \infty$. Then:

2. If f is square-integrable; i.e. $\int_\Theta f(\theta)^2 \pi(\theta) d\theta < \infty$, then $\mathrm{Var}(\hat{\mu}_{\mathrm{Ca}}) = \sigma^2_{\mathrm{Ca}}$ where:

$$\frac{g_{\min}}{g_{\max}} \left(\sigma^2_{\mathrm{MC}} - \frac{\mu^2}{n} \right) \le \sigma^2_{\mathrm{Ca}} \le \frac{g_{\max}}{g_{\min}} \left(\sigma^2_{\mathrm{MC}} - \frac{\mu^2}{n} \right).$$

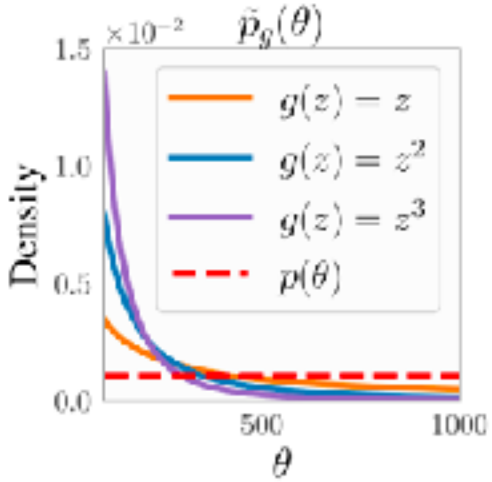
Some reassuring results

- Suppose that $g_{\max} = \sup_{\theta \in \Theta} g(c(\theta)) < \infty$. Then:

3. The ESS is bounded: $\left(\frac{g_{\min}}{g_{\max}}\right)^2 \leq \text{ESS} \leq \left(\frac{g_{\max}}{g_{\min}}\right)^2$.

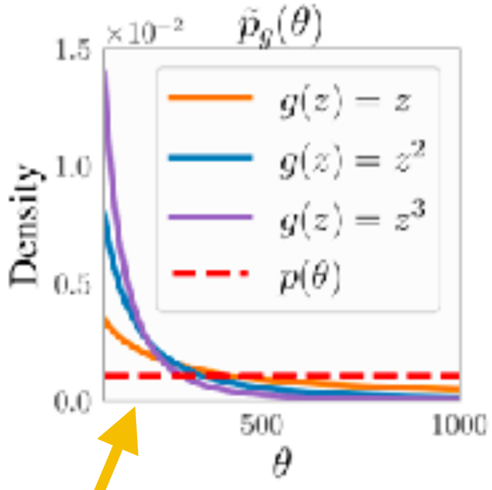
A Gamma simulator

- $\mathbb{P}_\theta = \text{Gamma}(\theta, 1)$,
- Simulator: Ahrens-Dieter acceptance-rejection method.
- Method: ABC!



A Gamma simulator

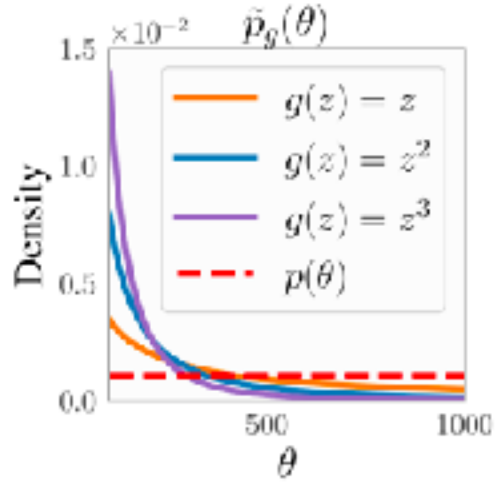
- $\mathbb{P}_\theta = \text{Gamma}(\theta, 1)$,
- Simulator: Ahrens-Dieter acceptance-rejection method.
- Method: ABC!

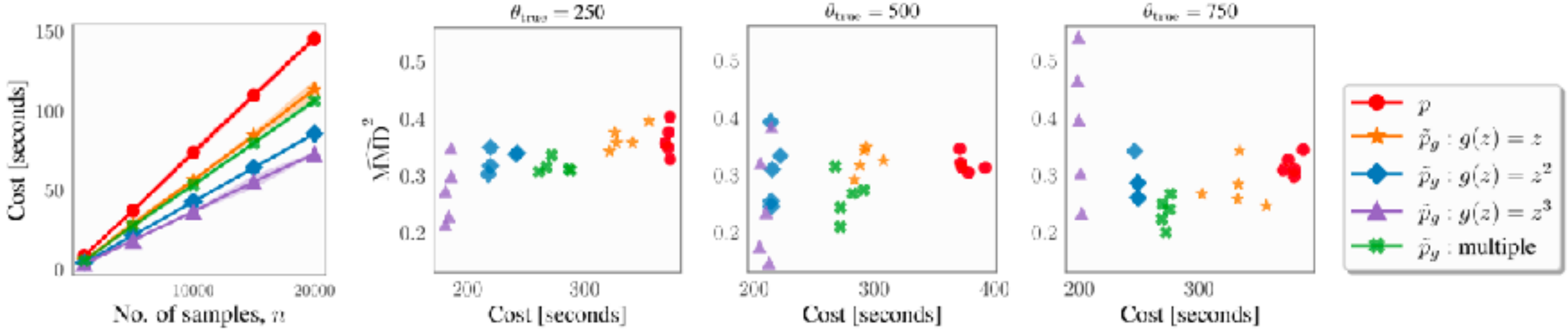


Cost-aware pushes us to sample from small θ values!

A Gamma simulator

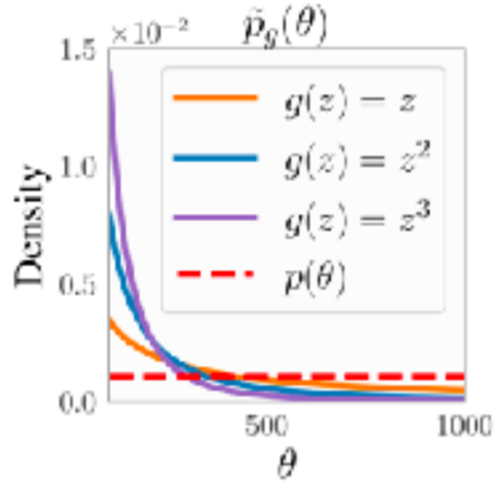
- $\mathbb{P}_\theta = \text{Gamma}(\theta,1)$,
- Simulator: Ahrens-Dieter acceptance-rejection method.
- Method: ABC!

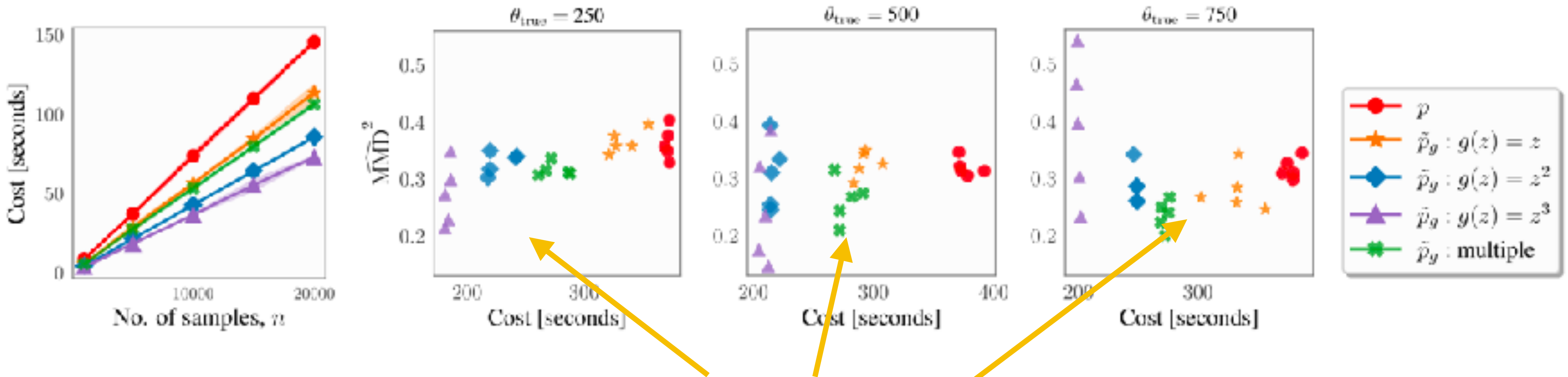




A Gamma simulator

- $\mathbb{P}_\theta = \text{Gamma}(\theta, 1)$,
- Simulator: Ahrens-Dieter acceptance-rejection method.
- Method: ABC!

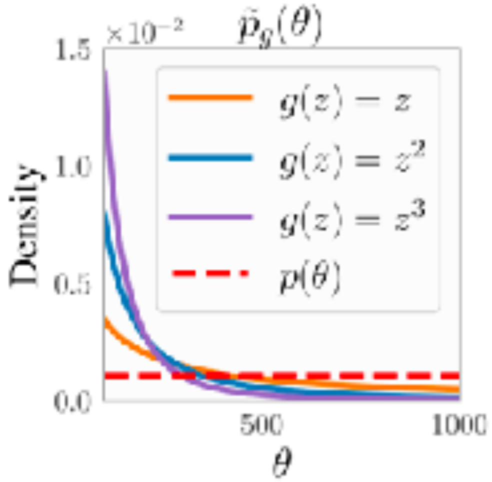




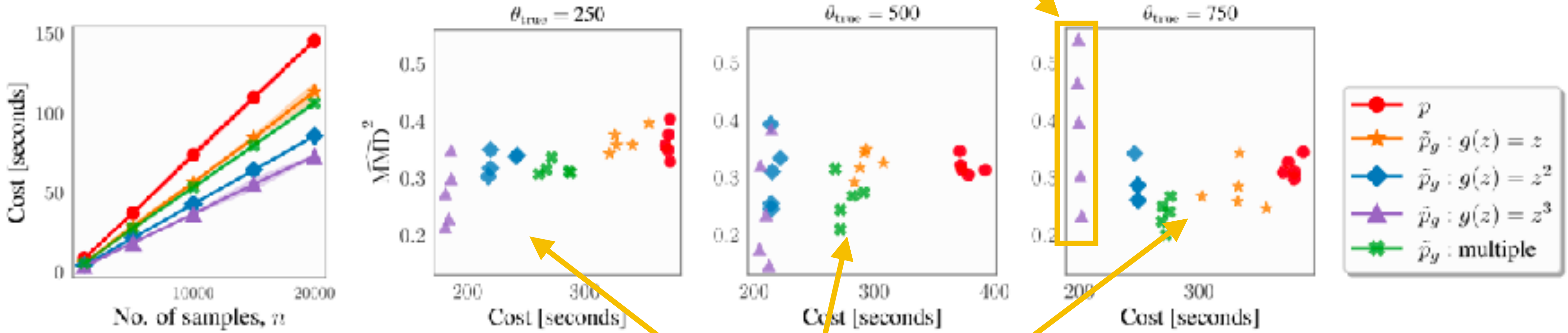
Being cost-aware tends to reduce your cost without a loss of accuracy!

A Gamma simulator

- $\mathbb{P}_\theta = \text{Gamma}(\theta, 1)$,
- Simulator: Ahrens-Dieter acceptance-rejection method.
- Method: ABC!



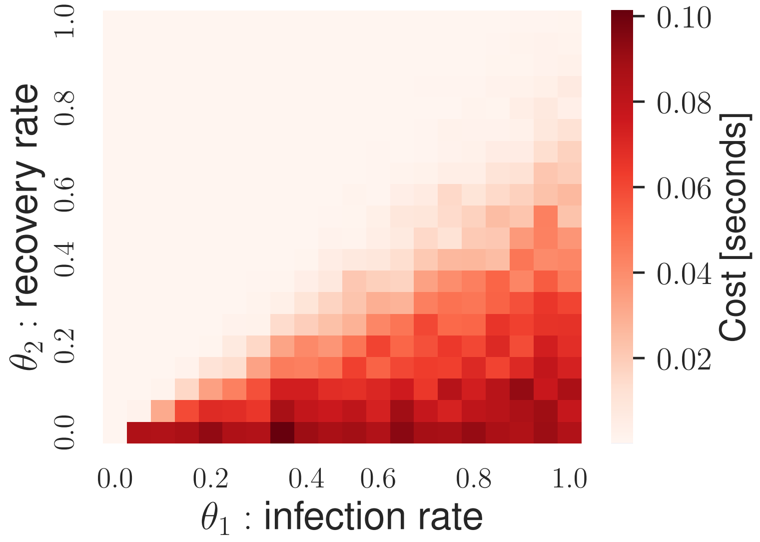
If truth in expensive region, being
'too' cost-aware won't be great!



Being cost-aware tends to reduce your cost without a loss of accuracy!

Some epidemiological models

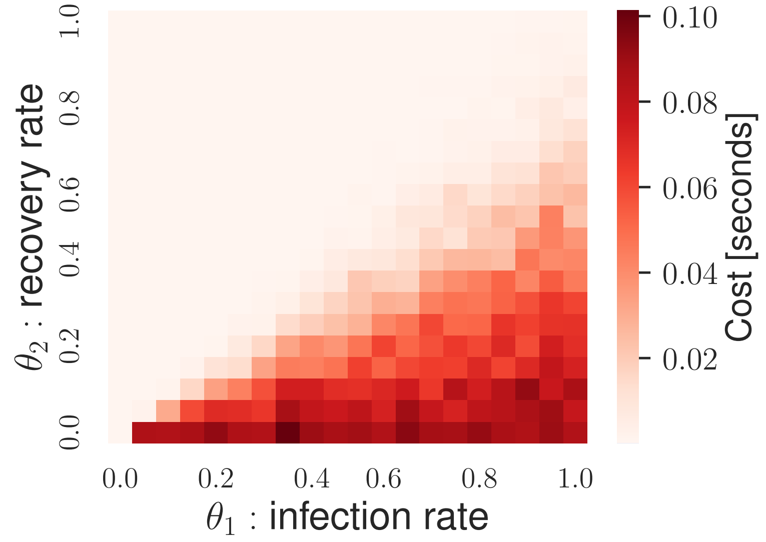
- We consider three different models with 1,2 and 3 parameters respectively, and use NPE.



Kypraios, T., Neal, P., and Prangle, D. (2017). A tutorial introduction to Bayesian inference for stochastic epidemic models using approximate Bayesian computation. Mathematical Biosciences, 287:42–53.

Some epidemiological models

- We consider three different models with 1,2 and 3 parameters respectively, and use NPE.

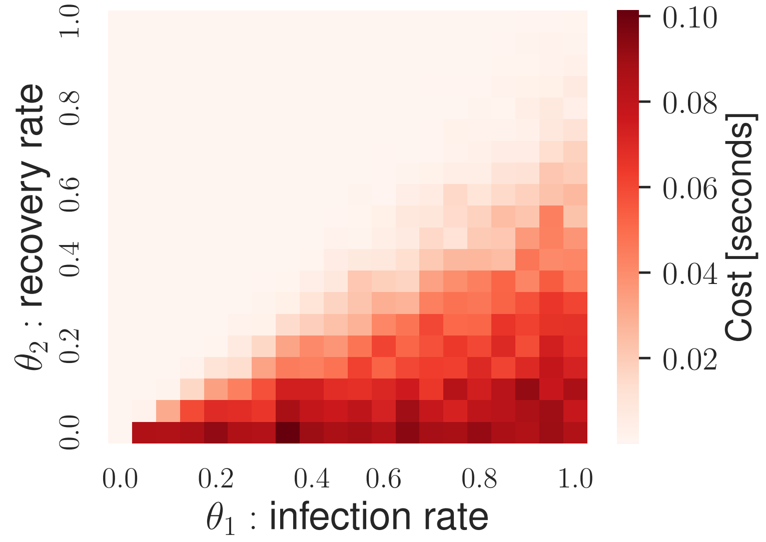


	$\widehat{\mathrm{MMD}}^2$ (↓)					Time saved (↑)			
	NPE	Ca-NPE $g(z)=z^{0.5}$	Ca-NPE $g(z)=z$	Ca-NPE $g(z)=z^2$	Ca-NPE multiple	Ca-NPE $g(z)=z^{0.5}$	Ca-NPE $g(z)=z$	Ca-NPE $g(z)=z^2$	Ca-NPE multiple
Homogen.	0.02(0.02)	0.02(0.01)	0.02(0.02)	0.23(0.08)	0.05(0.04)	16%(2)	38%(2)	70%(2)	30%(5)
Temporal	0.03(0.03)	0.06(0.03)	0.07(0.03)	0.07(0.03)	0.05(0.04)	36%(4)	65%(2)	85%(1)	24%(5)
Bernoulli	0.02(0.00)	0.02(0.00)	0.02(0.01)	0.04(0.01)	0.02(0.00)	23%(4)	37%(4)	47%(3)	25%(6)

Kypraios, T., Neal, P., and Prangle, D. (2017). A tutorial introduction to Bayesian inference for stochastic epidemic models using approximate Bayesian computation. Mathematical Biosciences, 287:42–53.

Some epidemiological models

- We consider three different models with $1{,}2$ and 3 parameters respectively, and use NPE.



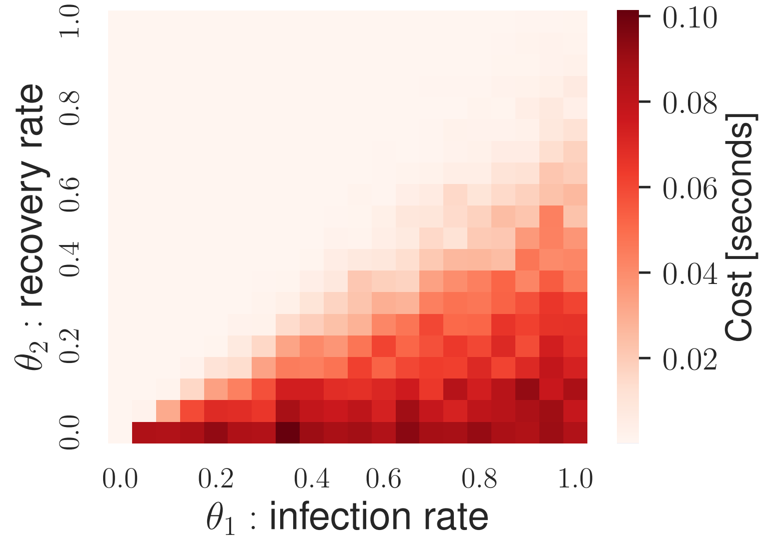
$g(z) = z^{0.5}$: Same accuracy but modest improvement!

	$\widehat{\mathrm{MMD}}^2$ (↓)					Time saved (↑)			
	NPE	Ca-NPE $g(z) = z^{0.5}$	Ca-NPE $g(z) = z$	Ca-NPE $g(z) = z^2$	Ca-NPE multiple	Ca-NPE $g(z) = z^{0.5}$	Ca-NPE $g(z) = z$	Ca-NPE $g(z) = z^2$	Ca-NPE multiple
Homogen.	0.02(0.02)	0.02(0.01)	0.02(0.02)	0.23(0.08)	0.05(0.04)	16%(2)	38%(2)	70%(2)	30%(5)
Temporal	0.03(0.03)	0.06(0.03)	0.07(0.03)	0.07(0.03)	0.05(0.04)	36%(4)	65%(2)	85%(1)	24%(5)
Bernoulli	0.02(0.00)	0.02(0.00)	0.02(0.01)	0.04(0.01)	0.02(0.00)	23%(4)	37%(4)	47%(3)	25%(6)

Kypraios, T., Neal, P., and Prangle, D. (2017). A tutorial introduction to Bayesian inference for stochastic epidemic models using approximate Bayesian computation. Mathematical Biosciences, 287:42–53.

Some epidemiological models

- We consider three different models with $1,2$ and 3 parameters respectively, and use NPE.



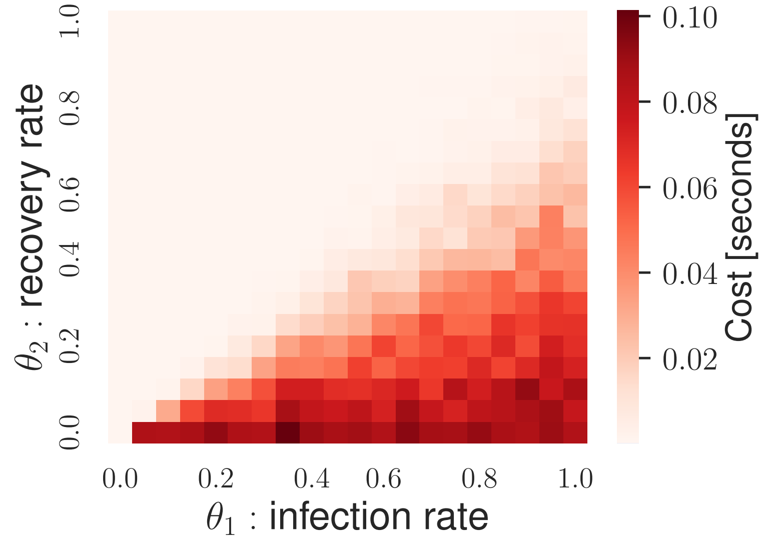
$g(z) = z$: Still same accuracy but slightly better improvement!

	$\widehat{\text{MMD}}^2$ (↓)					Time saved (↑)			
	NPE	Ca-NPE $g(z) = z^{0.5}$	Ca-NPE $g(z) = z$	Ca-NPE $g(z) = z^2$	Ca-NPE multiple	Ca-NPE $g(z) = z^{0.5}$	Ca-NPE $g(z) = z$	Ca-NPE $g(z) = z^2$	Ca-NPE multiple
Homogen.	0.02(0.02)	0.02(0.01)	$\boxed{0.02(0.02)}$	0.23(0.08)	0.05(0.04)	16%(2)	$\boxed{38\%(2)}$	70%(2)	30%(5)
Temporal	0.03(0.03)	0.06(0.03)	0.07(0.03)	0.07(0.03)	0.05(0.04)	36%(4)	65%(2)	85%(1)	24%(5)
Bernoulli	0.02(0.00)	0.02(0.00)	0.02(0.01)	0.04(0.01)	0.02(0.00)	23%(4)	37%(4)	47%(3)	25%(6)

Kypraios, T., Neal, P., and Prangle, D. (2017). A tutorial introduction to Bayesian inference for stochastic epidemic models using approximate Bayesian computation. Mathematical Biosciences, 287:42–53.

Some epidemiological models

- We consider three different models with 1,2 and 3 parameters respectively, and use NPE.



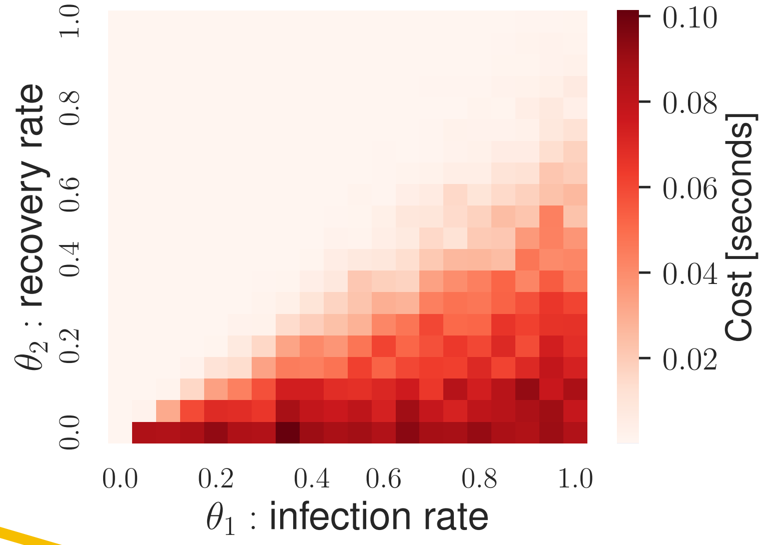
$g(z) = z^2$: Worse accuracy but much cheaper

	$\widehat{\mathrm{MMD}}^2$ (↓)					Time saved (↑)			
	NPE	Ca-NPE $g(z) = z^{0.5}$	Ca-NPE $g(z) = z$	Ca-NPE $g(z) = z^2$	Ca-NPE multiple	Ca-NPE $g(z) = z^{0.5}$	Ca-NPE $g(z) = z$	Ca-NPE $g(z) = z^2$	Ca-NPE multiple
Homogen.	0.02(0.02)	0.02(0.01)	0.02(0.02)	$\boxed{0.23(0.08)}$	0.05(0.04)	16%(2)	38%(2)	$\boxed{70\%(2)}$	30%(5)
Temporal	0.03(0.03)	0.06(0.03)	0.07(0.03)	0.07(0.03)	0.05(0.04)	36%(4)	65%(2)	85%(1)	24%(5)
Bernoulli	0.02(0.00)	0.02(0.00)	0.02(0.01)	0.04(0.01)	0.02(0.00)	23%(4)	37%(4)	47%(3)	25%(6)

Kypraios, T., Neal, P., and Prangle, D. (2017). A tutorial introduction to Bayesian inference for stochastic epidemic models using approximate Bayesian computation. Mathematical Biosciences, 287:42–53.

Some epidemiological models

- We consider three different models with 1,2 and 3 parameters respectively, and use NPE.



Typically slight loss of accuracy but decent reduction in cost!

	$\widehat{\mathrm{MMD}}^2$ (↓)					Time saved (↑)			
	NPE	Ca-NPE $g(z)=z^{0.5}$	Ca-NPE $g(z)=z$	Ca-NPE $g(z)=z^2$	Ca-NPE multiple	Ca-NPE $g(z)=z^{0.5}$	Ca-NPE $g(z)=z$	Ca-NPE $g(z)=z^2$	Ca-NPE multiple
Homogen.	0.02(0.02)	0.02(0.01)	0.02(0.02)	0.23(0.08)	0.05(0.04)	16%(2)	38%(2)	70%(2)	30%(5)
Temporal	0.03(0.03)	0.06(0.03)	0.07(0.03)	0.07(0.03)	0.05(0.04)	36%(4)	65%(2)	85%(1)	24%(5)
Bernoulli	0.02(0.00)	0.02(0.00)	0.02(0.01)	0.04(0.01)	0.02(0.00)	23%(4)	37%(4)	47%(3)	25%(6)

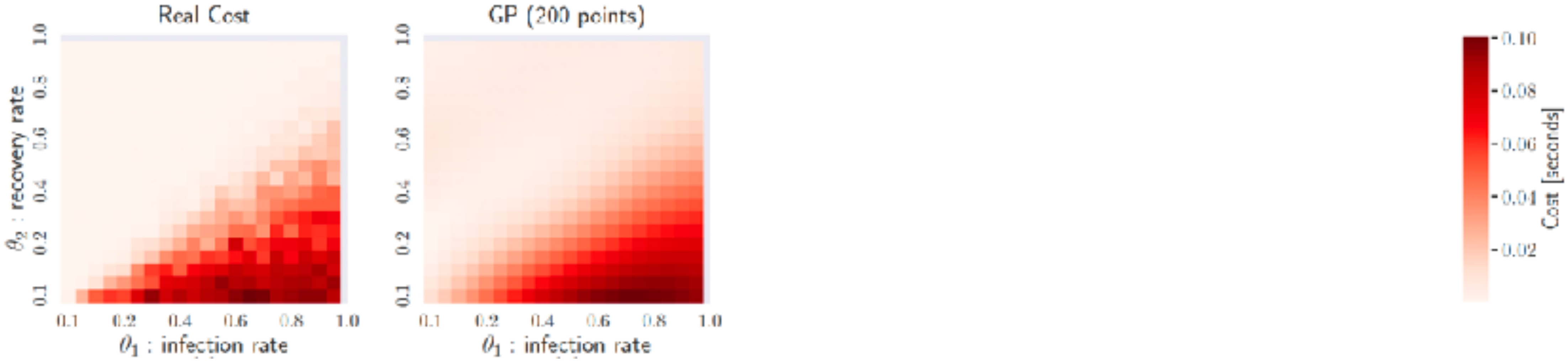
Kypraios, T., Neal, P., and Prangle, D. (2017). A tutorial introduction to Bayesian inference for stochastic epidemic models using approximate Bayesian computation. Mathematical Biosciences, 287:42–53.

Estimating the cost function

When the cost function is unknown, it can be estimated through simulations+regression.
This is typically very cheap, and simulations can be re-used for inference!

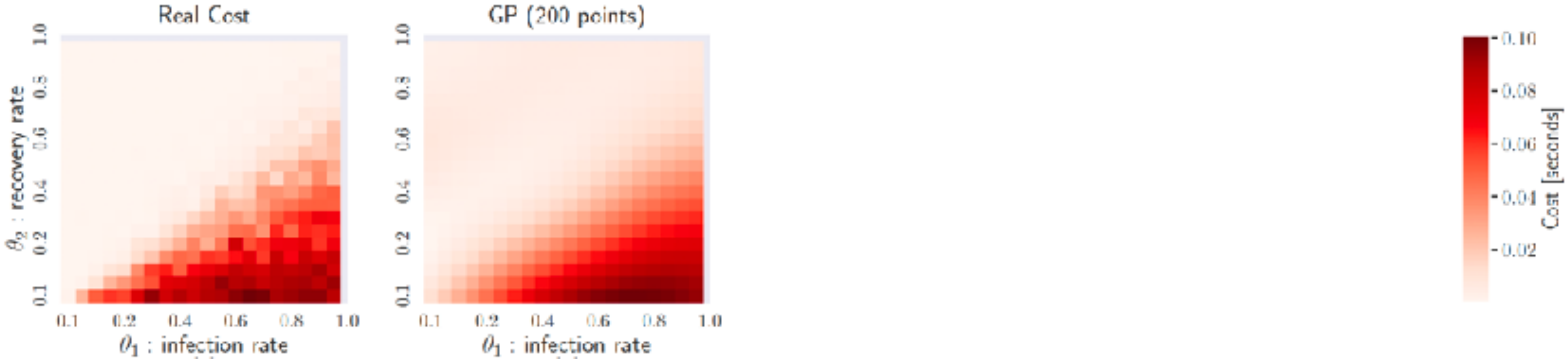
Estimating the cost function

When the cost function is unknown, it can be estimated through simulations+regression.
This is typically very cheap, and simulations can be re-used for inference!



Estimating the cost function

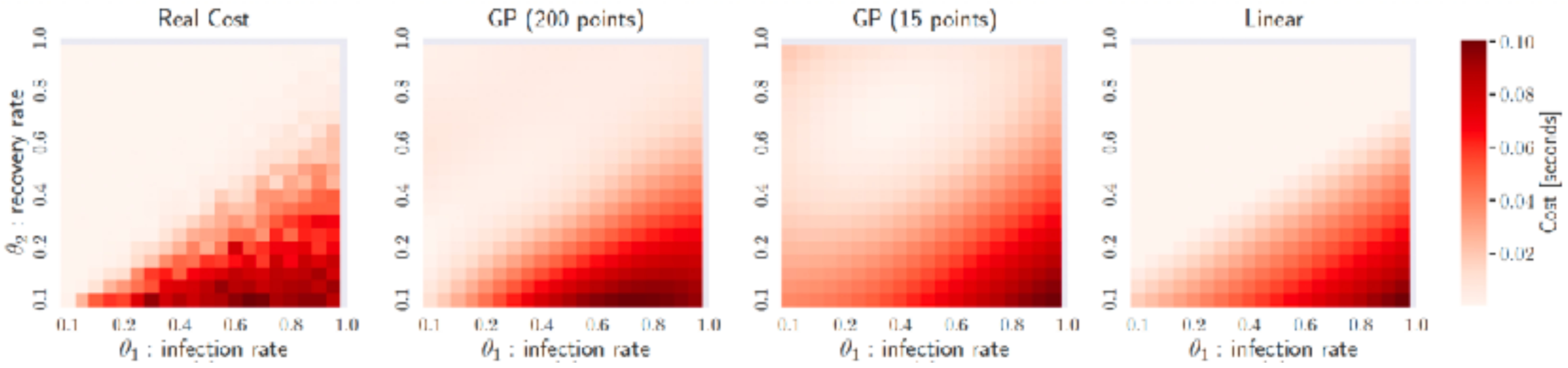
When the cost function is unknown, it can be estimated through simulations+regression. This is typically very cheap, and simulations can be re-used for inference!



Very accurate!

Estimating the cost function

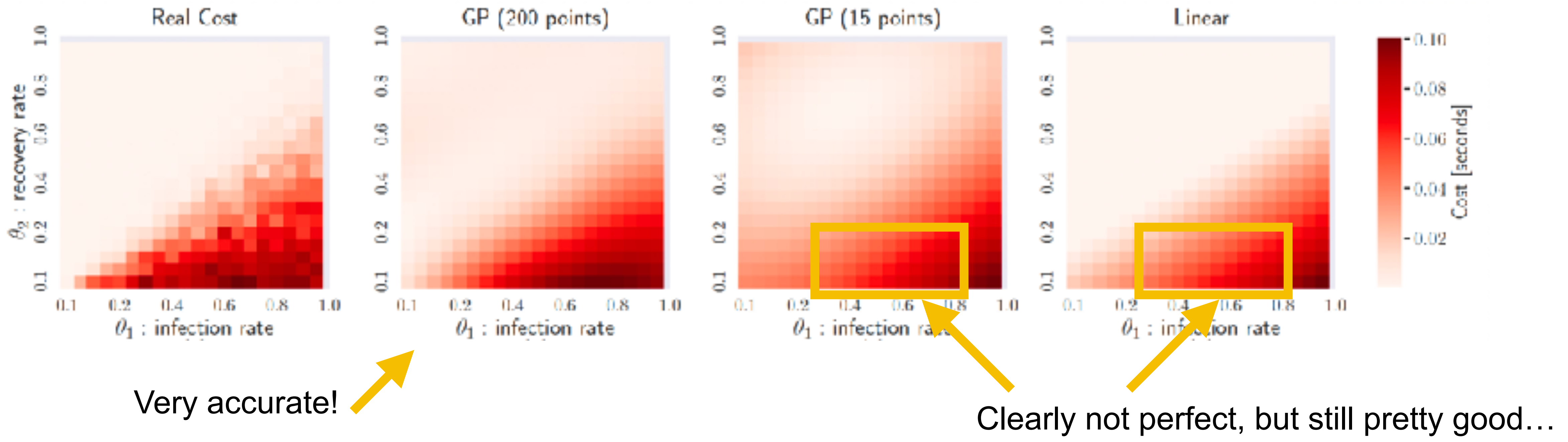
When the cost function is unknown, it can be estimated through simulations+regression.
This is typically very cheap, and simulations can be re-used for inference!



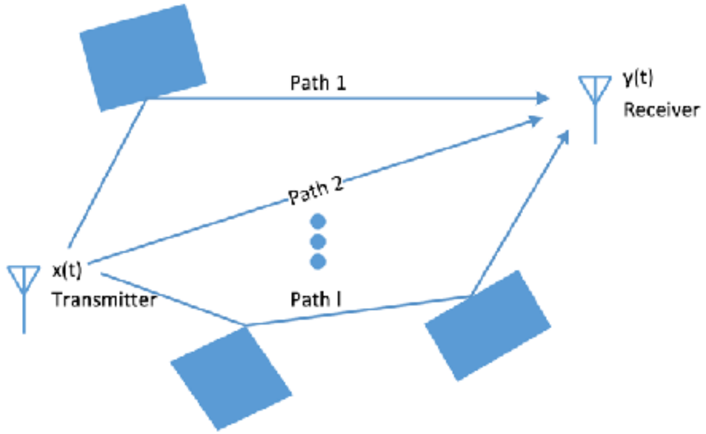
Very accurate!

Estimating the cost function

When the cost function is unknown, it can be estimated through simulations+regression.
This is typically very cheap, and simulations can be re-used for inference!

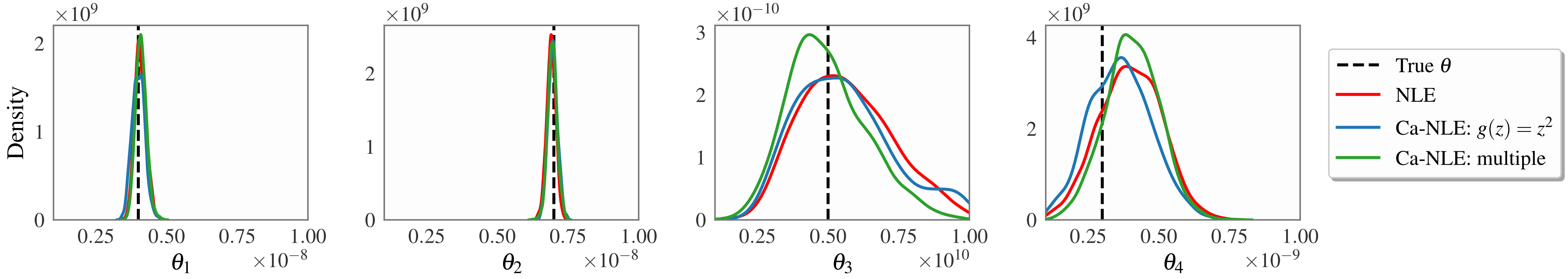


Back to radio-propagation



<u>Computational Cost</u>

- **Standard NLE:** 15.6h,
- **Cost-aware NLE:** 8.8h!!



Bharti, A., **Briol, F-X.**, Pedersen, T. (2022). A general method for calibrating stochastic radio channel models with kernels. IEEE Transactions on Antennas and Propagation, vol. 70, no. 6, pp. 3986-4001, June 2022.

Conclusion

Conclusion

- We proposed a novel importance sampling algorithm which focuses on **down weighting sampling** in regions with a **large downstream cost.**

Conclusion

- We proposed a novel importance sampling algorithm which focuses on **down weighting sampling** in regions with a **large downstream cost.**
- Although I presented this for NLE/NPE, we also have experiments for ABC and it could be applied to any other sampling-based SBI method.

Conclusion

- We proposed a novel importance sampling algorithm which focuses on **down weighting sampling** in regions with a **large downstream cost.**
- Although I presented this for NLE/NPE, we also have experiments for ABC and it could be applied to any other sampling-based SBI method.
- Need more computational statisticians engaging with neural-based simulation inference!

Any Questions?

Paper: Bharti, A., Huang, D., Kaski, S., & Briol, F.-X. (2025). Cost-aware simulation-based inference. International Conference on Artificial Intelligence and Statistics, 28–36.

Code: https://github.com/huangdaolang/cost-aware-sbi

Robust Bayesian simulation-based inference

Paper: Dellaporta, C., Knoblauch, J., Damoulas, T. & **Briol, F-X** (2022). Robust Bayesian inference for simulator-based models via the MMD posterior bootstrap. AISTATS, 943-970. Best paper award.

Code: https://github.com/haritadell/npl_mmd_project

Connections with Jeremias' course

Optimisation-centric posteriors / Generalised Variational Inference

$$q_n^*(\theta) = \arg\min_{q \in \mathcal{Q}} \left\{ \mathcal{L}(q, x_{1:n}) \quad + \quad \mathrm{D}\big(q, \pi\big) \right\}$$

Gibbs/Generalised/

$$\pi_n^{\mathsf{L}}(\theta \mid x_{1:n}) = \frac{\exp\{-\mathsf{L}(x_{1:n}, p_\theta)\} \cdot \pi(\theta)}{\int \exp\{-\mathsf{L}(x_{1:n}, p_\theta)\} \cdot \pi(\theta) d\theta}$$

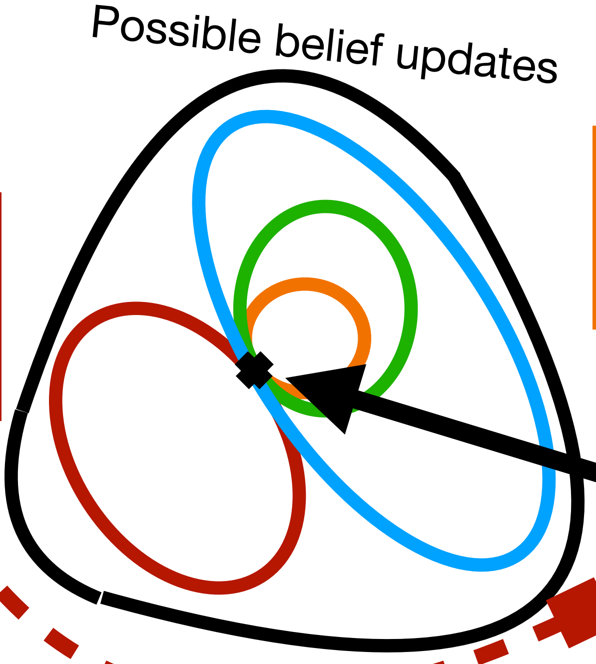
Martingale posteriors & resampling-based approaches

For $i = 1,2,\ldots$

$$X_{n+i+1} \sim p(X_{n+i} \mid x_{1:n}, X_{n+1:n+i})$$

$$\theta^\infty = \mathrm{argmin}_{\theta \in \Theta} \mathsf{L}\left([x_{1:n}, X_{n+1:\infty}], \theta\right)$$

[See Fong, Holmes, & Walker (2023)]



Power/Fractional/

$$\pi_n^{(\lambda)}(\theta \mid x_{1:n}) = \frac{p(x_{1:n} \mid \theta)^\lambda \cdot \pi(\theta)}{\int p(x_{1:n} \mid \theta)^\lambda \cdot \pi(\theta) d\theta}$$

Bayes'

$$\pi_n(\theta \mid x_{1:n}) = \frac{p(x_{1:n} \mid \theta) \cdot \pi(\theta)}{\int p(x_{1:n} \mid \theta) \cdot \pi(\theta) d\theta}$$

Connections with Jeremias' course

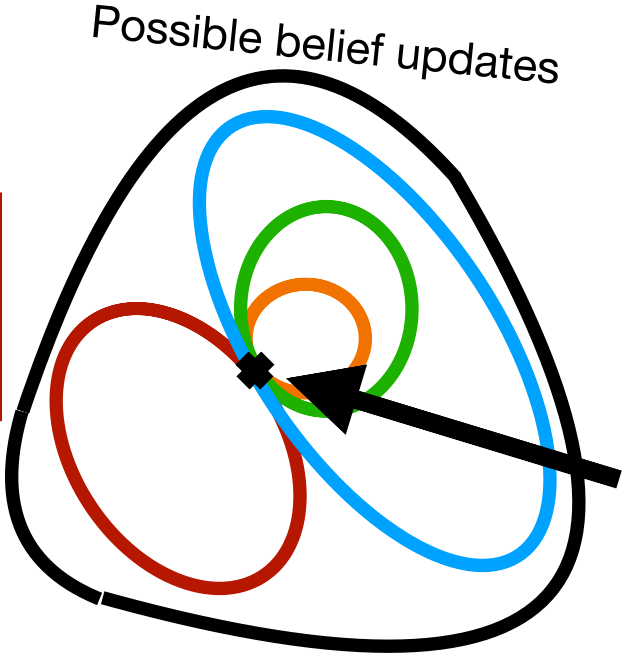
Martingale posteriors & resampling-based approaches

For $i = 1,2,\ldots$

$$X_{n+i+1} \sim p(X_{n+i} \mid x_{1:n}, X_{n+1:n+i})$$

$$\theta^{\infty} = \text{argmin}_{\theta \in \Theta} \mathsf{L}\left([x_{1:n}, X_{n+1:\infty}], \theta\right)$$

[See Fong, Holmes, & Walker (2023)]



Non-parametric Learning

- Place a Dirichlet process $\mathrm{DP}(\alpha; \mathbb{F})$ prior on $\mathbb{Q}$

Lyddon, S., Walker, S., & Holmes, C. (2018). Nonparametric learning from Bayesian models with randomized objective functions. *NeurIPS*, 2071–2081.

Fong, E., Lyddon, S., & Holmes, C. (2019). Scalable nonparametric sampling from multimodal posteriors with the posterior bootstrap. ICML, 3443–3464.

Non-parametric Learning

- Place a Dirichlet process $\mathrm{DP}(\alpha; \mathbb{F})$ prior on $\mathbb{Q}$

- Condition this prior on the observed data $y_1, \ldots, y_n \sim \mathbb{Q}$ to get a posterior:

$$\mathrm{DP}(\alpha'; \mathbb{F}') \qquad \alpha' = \alpha + n \qquad \mathbb{F}' = \frac{\alpha}{\alpha + n}\mathbb{F} + \frac{n}{\alpha + n}\mathbb{Q}_n$$

Lyddon, S., Walker, S., & Holmes, C. (2018). Nonparametric learning from Bayesian models with randomized objective functions. *NeurIPS*, 2071–2081.

Fong, E., Lyddon, S., & Holmes, C. (2019). Scalable nonparametric sampling from multimodal posteriors with the posterior bootstrap. ICML, 3443–3464.

Non-parametric Learning

Rather than doing inference on $\{\mathbb{P}_\theta\}_{\theta\in\Theta}$ (which could be misspecified), we do inference on $\mathbb{Q}$!

- Place a Dirichlet process $\mathrm{DP}(\alpha; \mathbb{F})$ prior on $\mathbb{Q}$

- Condition this prior on the observed data $y_1, \ldots, y_n \sim \mathbb{Q}$ to get a posterior:

$$\mathrm{DP}(\alpha'; \mathbb{F}') \qquad \alpha' = \alpha + n \qquad \mathbb{F}' = \frac{\alpha}{\alpha+n}\mathbb{F} + \frac{n}{\alpha+n}\mathbb{Q}_n$$

Lyddon, S., Walker, S., & Holmes, C. (2018). Nonparametric learning from Bayesian models with randomized objective functions. *NeurIPS*, 2071–2081.

Fong, E., Lyddon, S., & Holmes, C. (2019). Scalable nonparametric sampling from multimodal posteriors with the posterior bootstrap. ICML, 3443–3464.

Non-parametric Learning

Rather than doing inference on $\{\mathbb{P}_\theta\}_{\theta\in\Theta}$ (which could be misspecified), we do inference on $\mathbb{Q}$!

- Place a Dirichlet process $\mathrm{DP}(\alpha; \mathbb{F})$ prior on $\mathbb{Q}$

- Condition this prior on the observed data $y_1, \ldots, y_n \sim \mathbb{Q}$ to get a posterior:

$$\mathrm{DP}(\alpha'; \mathbb{F}') \qquad \alpha' = \alpha + n \qquad \mathbb{F}' = \frac{\alpha}{\alpha+n}\mathbb{F} + \frac{n}{\alpha+n}\mathbb{Q}_n$$

- Map to parameter space

$$\theta^* := \arg\min_{\theta\in\Theta} \mathbb{E}_{X\sim\mathbb{Q}}[l(X,\theta)]$$

Lyddon, S., Walker, S., & Holmes, C. (2018). Nonparametric learning from Bayesian models with randomized objective functions. *NeurIPS*, 2071–2081.

Fong, E., Lyddon, S., & Holmes, C. (2019). Scalable nonparametric sampling from multimodal posteriors with the posterior bootstrap. ICML, 3443–3464.

Non-parametric Learning

Rather than doing inference on $\{\mathbb{P}_\theta\}_{\theta\in\Theta}$ (which could be misspecified), we do inference on $\mathbb{Q}$!

- Place a Dirichlet process $\mathrm{DP}(\alpha; \mathbb{F})$ prior on $\mathbb{Q}$

- Condition this prior on the observed data $y_1, \ldots, y_n \sim \mathbb{Q}$ to get a posterior:

$$\mathrm{DP}(\alpha'; \mathbb{F}') \qquad \alpha' = \alpha + n \qquad \mathbb{F}' = \frac{\alpha}{\alpha+n}\mathbb{F} + \frac{n}{\alpha+n}\mathbb{Q}_n$$

- Map to parameter space

$$\theta^* := \arg\min_{\theta\in\Theta} \mathbb{E}_{X\sim\mathbb{Q}}[l(X,\theta)]$$

We still care about $\{\mathbb{P}_\theta\}_{\theta\in\Theta}$, so we map back to parameter space!

Lyddon, S., Walker, S., & Holmes, C. (2018). Nonparametric learning from Bayesian models with randomized objective functions. *NeurIPS*, 2071–2081.

Fong, E., Lyddon, S., & Holmes, C. (2019). Scalable nonparametric sampling from multimodal posteriors with the posterior bootstrap. ICML, 3443–3464.

The posterior bootstrap (i.e. NPL in practice)

(1) Sample $\mathbb{Q}^{(1)}, \mathbb{Q}^{(2)}, \ldots$ from the DP posterior.

The posterior bootstrap (i.e. NPL in practice)

(1) Sample $\mathbb{Q}^{(1)}, \mathbb{Q}^{(2)}, \ldots$ from the DP posterior.

(2) Compute the corresponding $\theta^{(1)}, \theta^{(2)}, \ldots$ using:

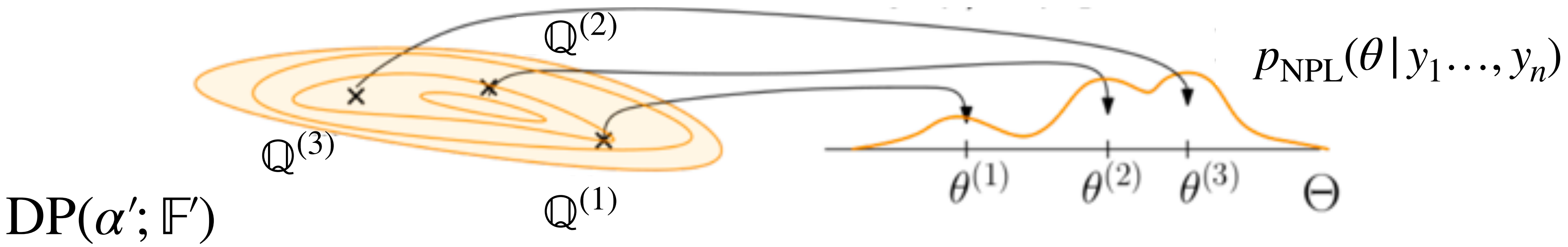
$$\theta^{(j)} := \arg\min_{\theta \in \Theta} \mathbb{E}_{X \sim \mathbb{Q}^{(j)}}[l(X, \theta)]$$

The posterior bootstrap (i.e. NPL in practice)

(1) Sample $\mathbb{Q}^{(1)}, \mathbb{Q}^{(2)}, \ldots$ from the DP posterior.

(2) Compute the corresponding $\theta^{(1)}, \theta^{(2)}, \ldots$ using:

$$\theta^{(j)} := \arg\min_{\theta \in \Theta} \mathbb{E}_{X \sim \mathbb{Q}^{(j)}}[l(X, \theta)]$$



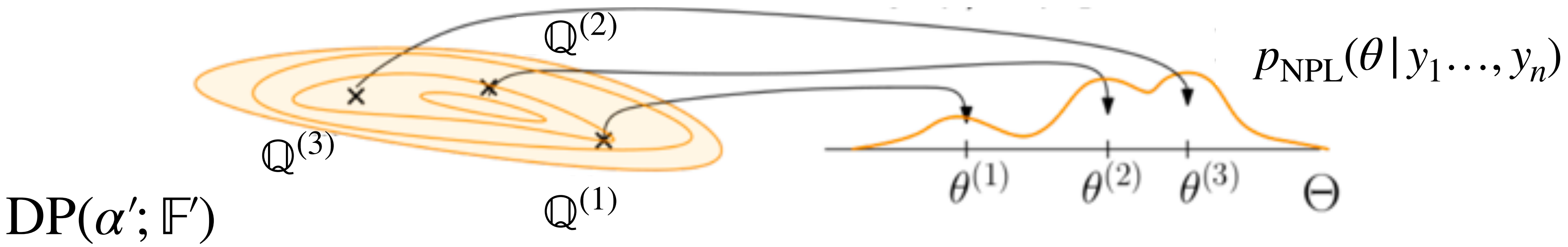
The posterior bootstrap (i.e. NPL in practice)

(1) Sample $\mathbb{Q}^{(1)}, \mathbb{Q}^{(2)}, \ldots$ from the DP posterior.

Approximated with stick-breaking procedure

(2) Compute the corresponding $\theta^{(1)}, \theta^{(2)}, \ldots$ using:

$$\theta^{(j)} := \arg\min_{\theta \in \Theta} \mathbb{E}_{X \sim \mathbb{Q}^{(j)}}[l(X, \theta)]$$



The posterior bootstrap (i.e. NPL in practice)

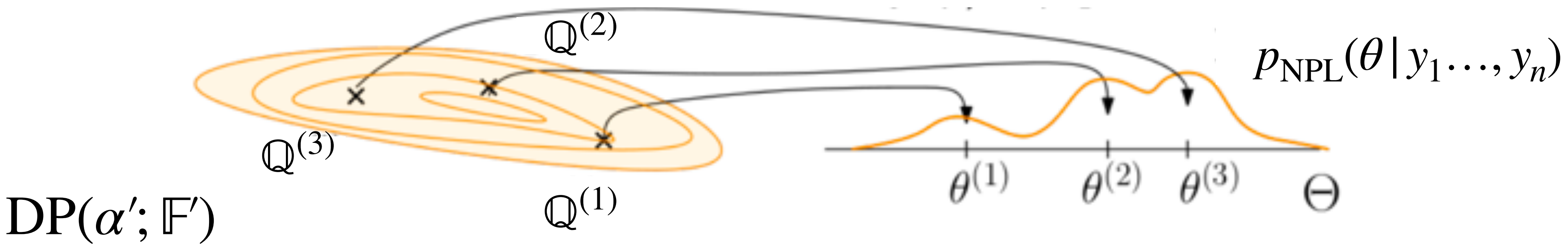
(1) Sample $\mathbb{Q}^{(1)}, \mathbb{Q}^{(2)}, \ldots$ from the DP posterior.

Approximated with stick-breaking procedure

(2) Compute the corresponding $\theta^{(1)}, \theta^{(2)}, \ldots$ using:

$$\theta^{(j)} := \arg\min_{\theta \in \Theta} \mathbb{E}_{X \sim \mathbb{Q}^{(j)}}[l(X, \theta)]$$

Approximated with empirical loss



The MMD posterior bootstrap

(1) Sample $\mathbb{Q}^{(1)}, \mathbb{Q}^{(2)}, \ldots$ using stick-breaking approximation of DP posterior.

(2) Compute the corresponding $\theta^{(1)}, \theta^{(2)}, \ldots$ using:

$$\theta^{(j)} := \arg\min_{\theta \in \Theta} \mathrm{MMD}^2(\mathbb{P}_\theta, \mathbb{Q}_n^{(j)})$$

The MMD posterior bootstrap

(1) Sample $\mathbb{Q}^{(1)}, \mathbb{Q}^{(2)}, \ldots$ using stick-breaking approximation of DP posterior.

(2) Compute the corresponding $\theta^{(1)}, \theta^{(2)}, \ldots$ using:

$$\theta^{(j)} := \arg\min_{\theta \in \Theta} \mathrm{MMD}^2(\mathbb{P}_\theta, \mathbb{Q}_n^{(j)})$$

- The MMD with bounded kernel has been shown to be a robust distance

The MMD posterior bootstrap

(1) Sample $\mathbb{Q}^{(1)}, \mathbb{Q}^{(2)}, \ldots$ using stick-breaking approximation of DP posterior.

(2) Compute the corresponding $\theta^{(1)}, \theta^{(2)}, \ldots$ using:

$$\theta^{(j)} := \arg\min_{\theta \in \Theta} \mathrm{MMD}^2(\mathbb{P}_\theta, \mathbb{Q}_n^{(j)})$$

- The MMD with bounded kernel has been shown to be a robust distance

$$\mathrm{MMD}^2(\mathbb{P}, \mathbb{Q}) = \int_{\mathcal{X}} \int_{\mathcal{X}} k(x, y)\mathbb{P}(dx)\mathbb{P}(dy) - 2 \int_{\mathcal{X}} \int_{\mathcal{X}} k(x, y)\mathbb{P}(dx)\mathbb{Q}(dy) + \int_{\mathcal{X}} \int_{\mathcal{X}} k(x, y)\mathbb{Q}(dx)\mathbb{Q}(dy)$$

Bounded!

The MMD posterior bootstrap

(1) Sample $\mathbb{Q}^{(1)}, \mathbb{Q}^{(2)}, \ldots$ using stick-breaking approximation of DP posterior.

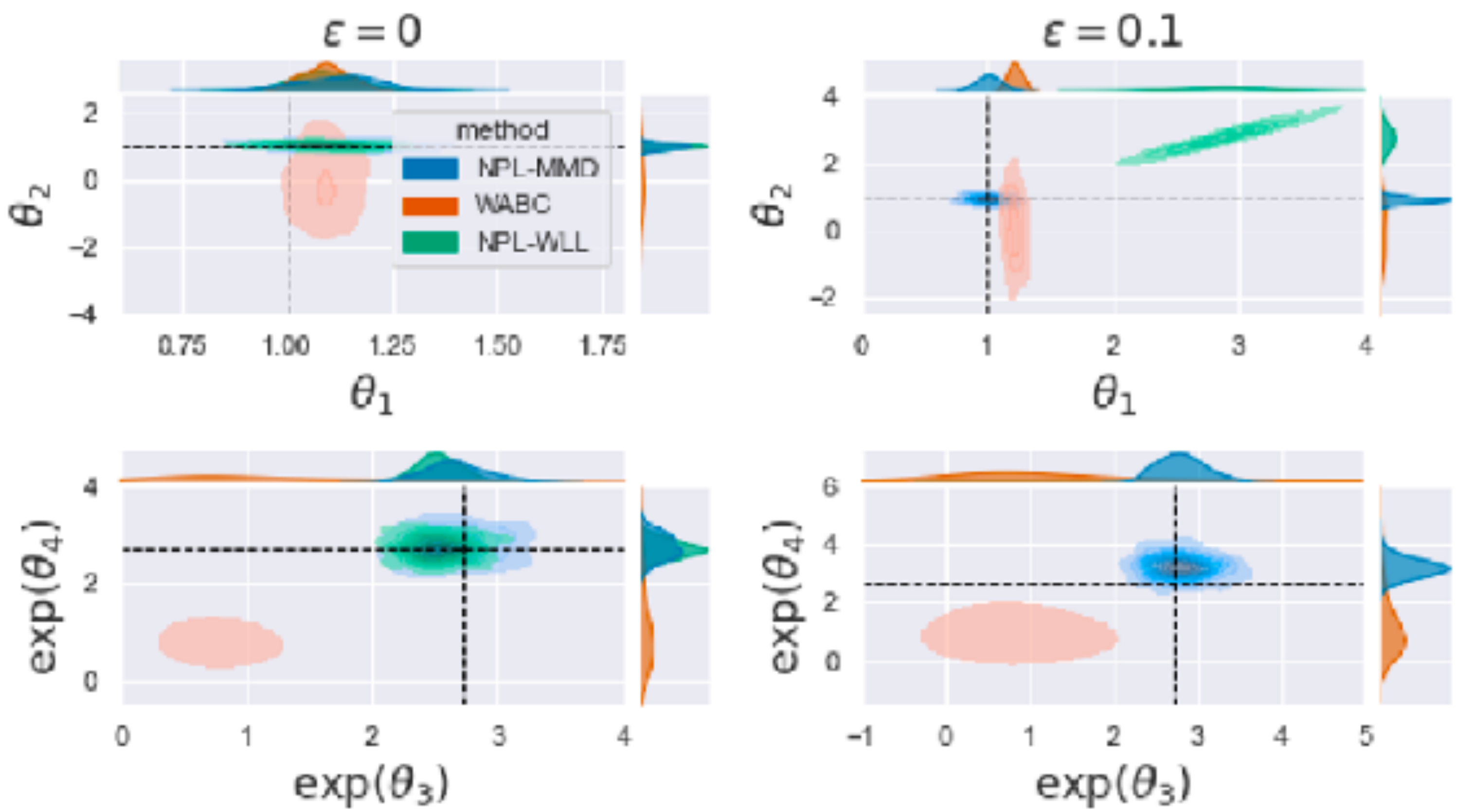
(2) Compute the corresponding $\theta^{(1)}, \theta^{(2)}, \ldots$ using:

$$\theta^{(j)} := \arg\min_{\theta \in \Theta} \mathrm{MMD}^2(\mathbb{P}_\theta, \mathbb{Q}_n^{(j)})$$

- The MMD with bounded kernel has been shown to be a robust distance

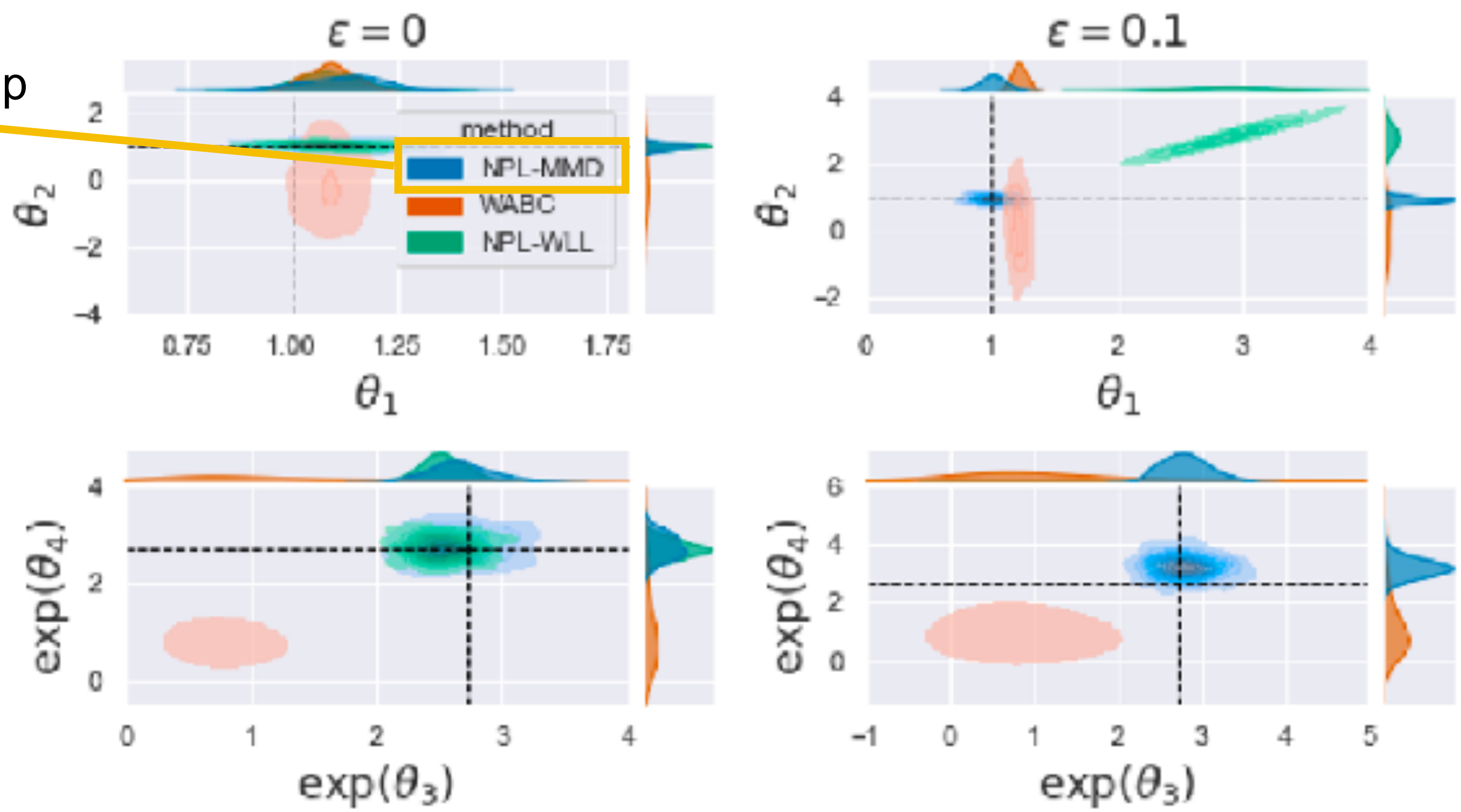
→ **Double robustness** robust inference procedure and robust estimator!

Example 1: Misspecified Gaussian



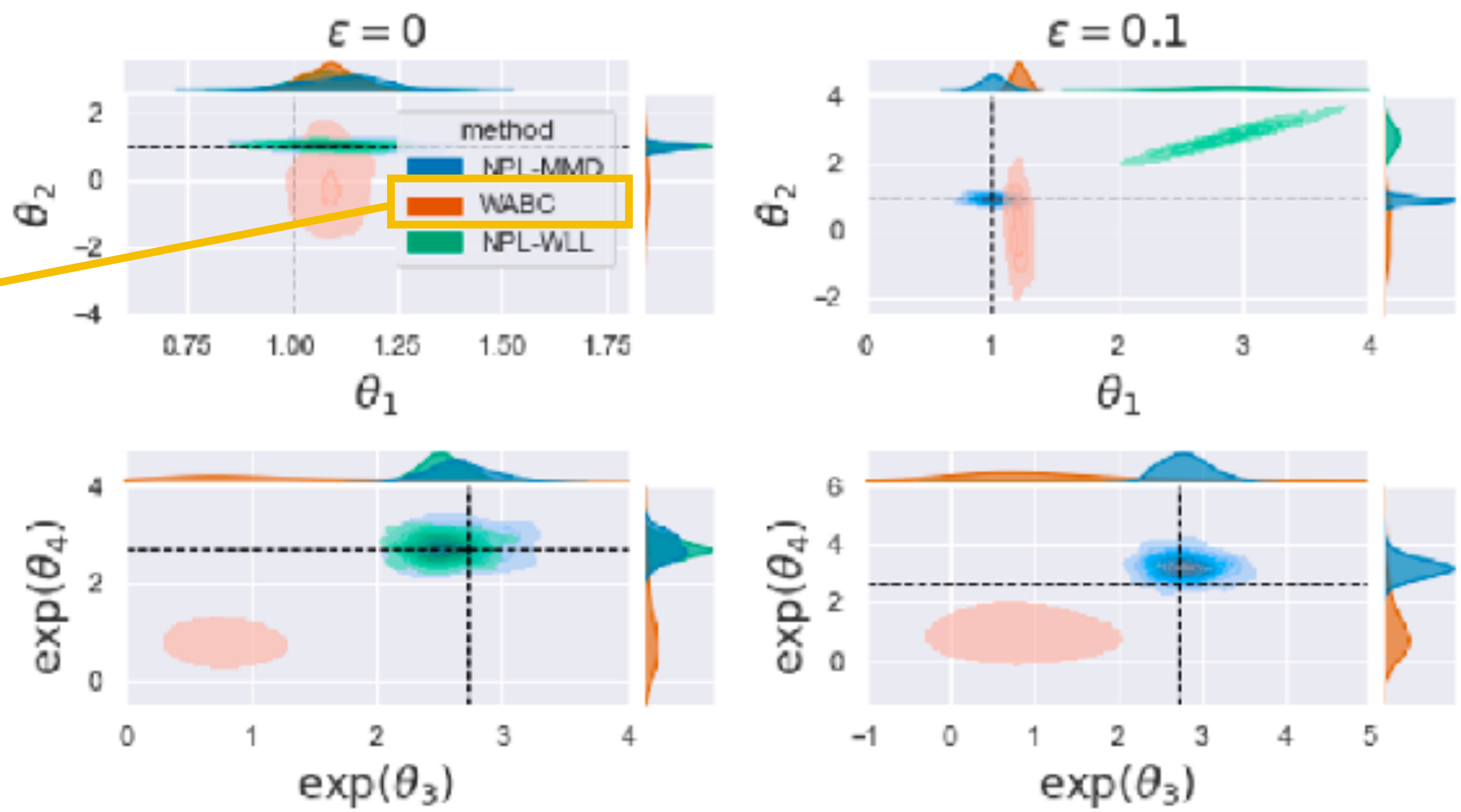
Example 1: Misspecified Gaussian

NPL-MMD:
MMD posterior bootstrap

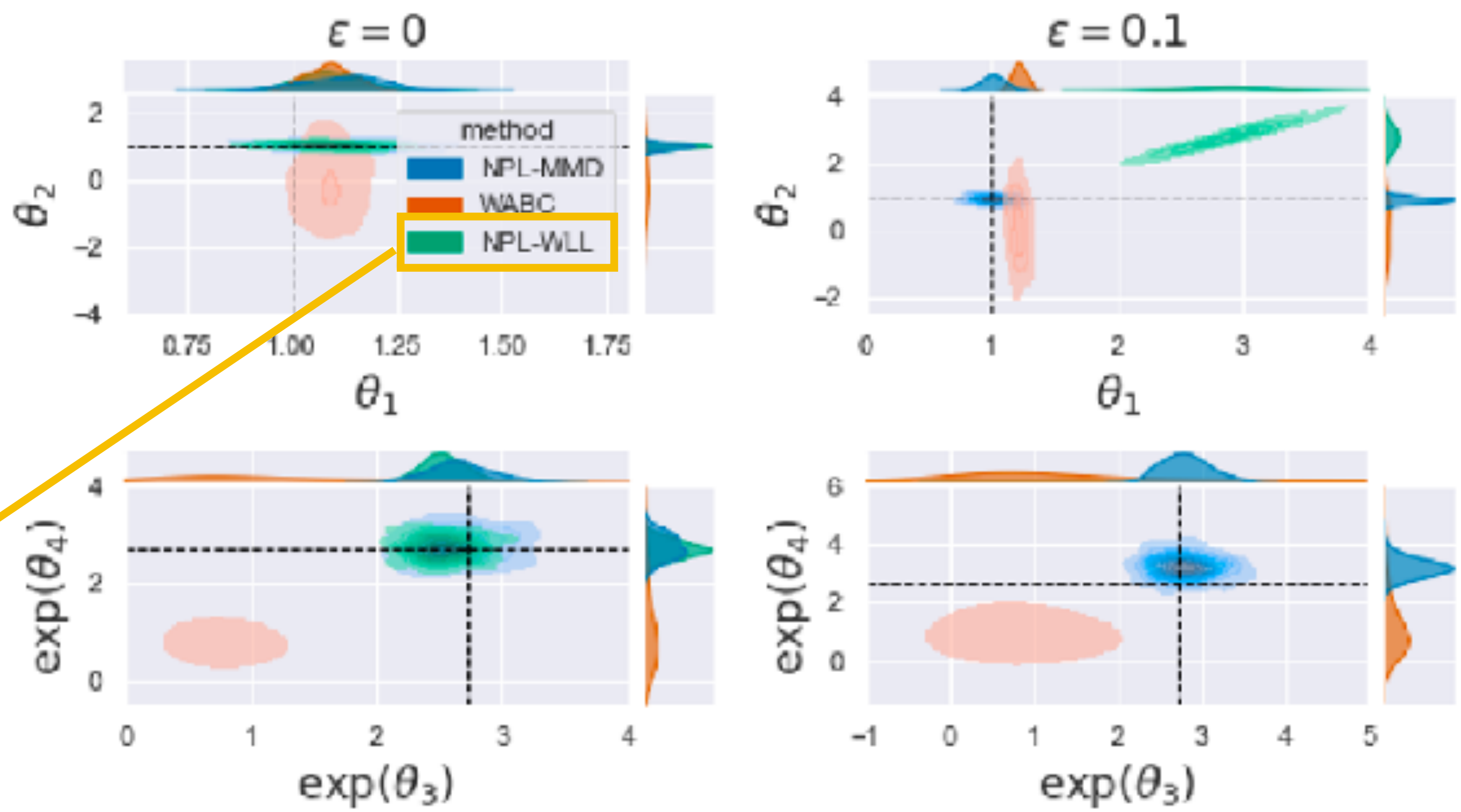


Example 1: Misspecified Gaussian

WABC:
ABC with Wasserstein distance



Example 1: Misspecified Gaussian



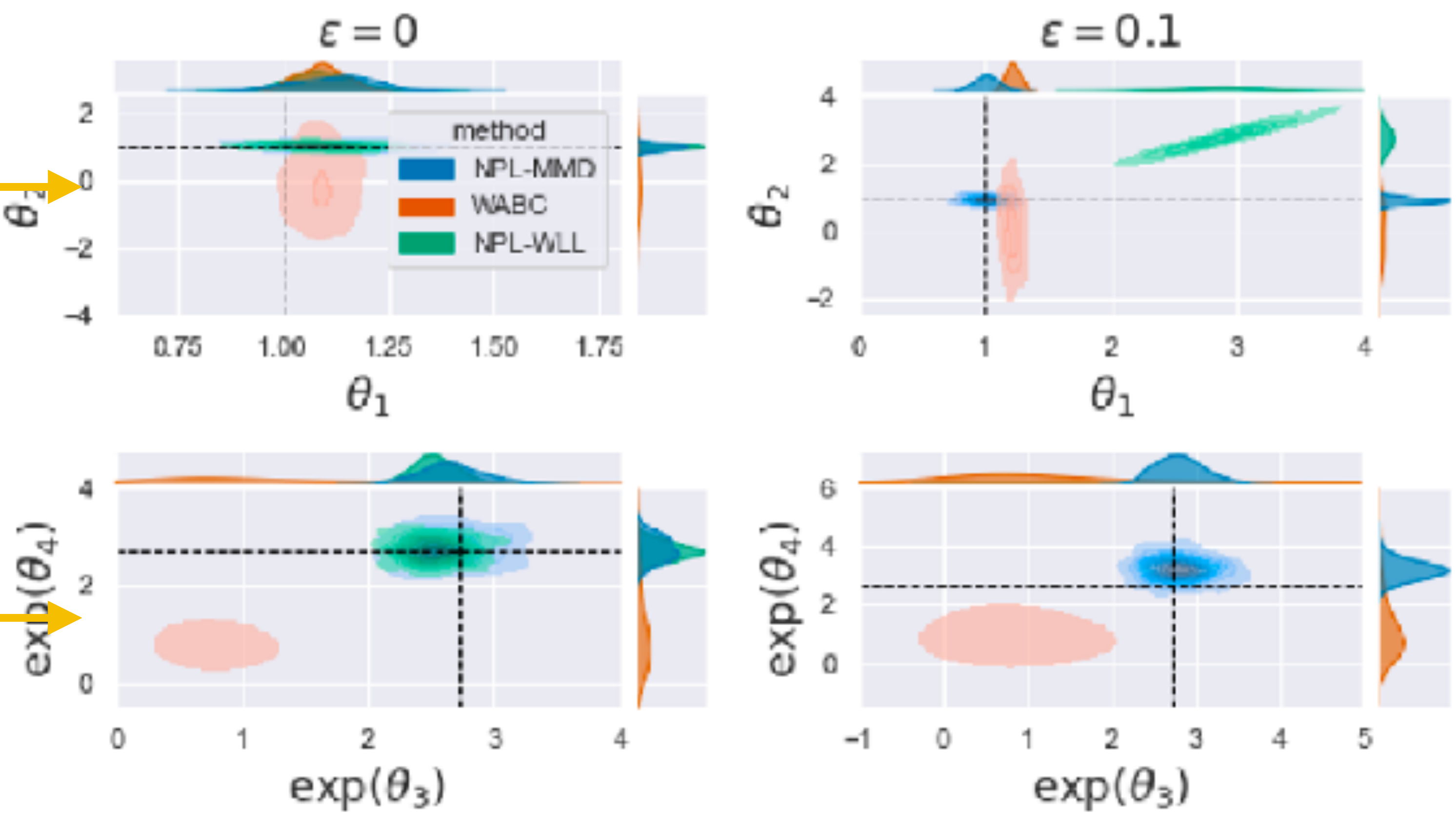
NPL-WLL:

$$l(x, \theta) = -\beta \log p(x \mid \theta)$$

Example 1: Misspecified Gaussian

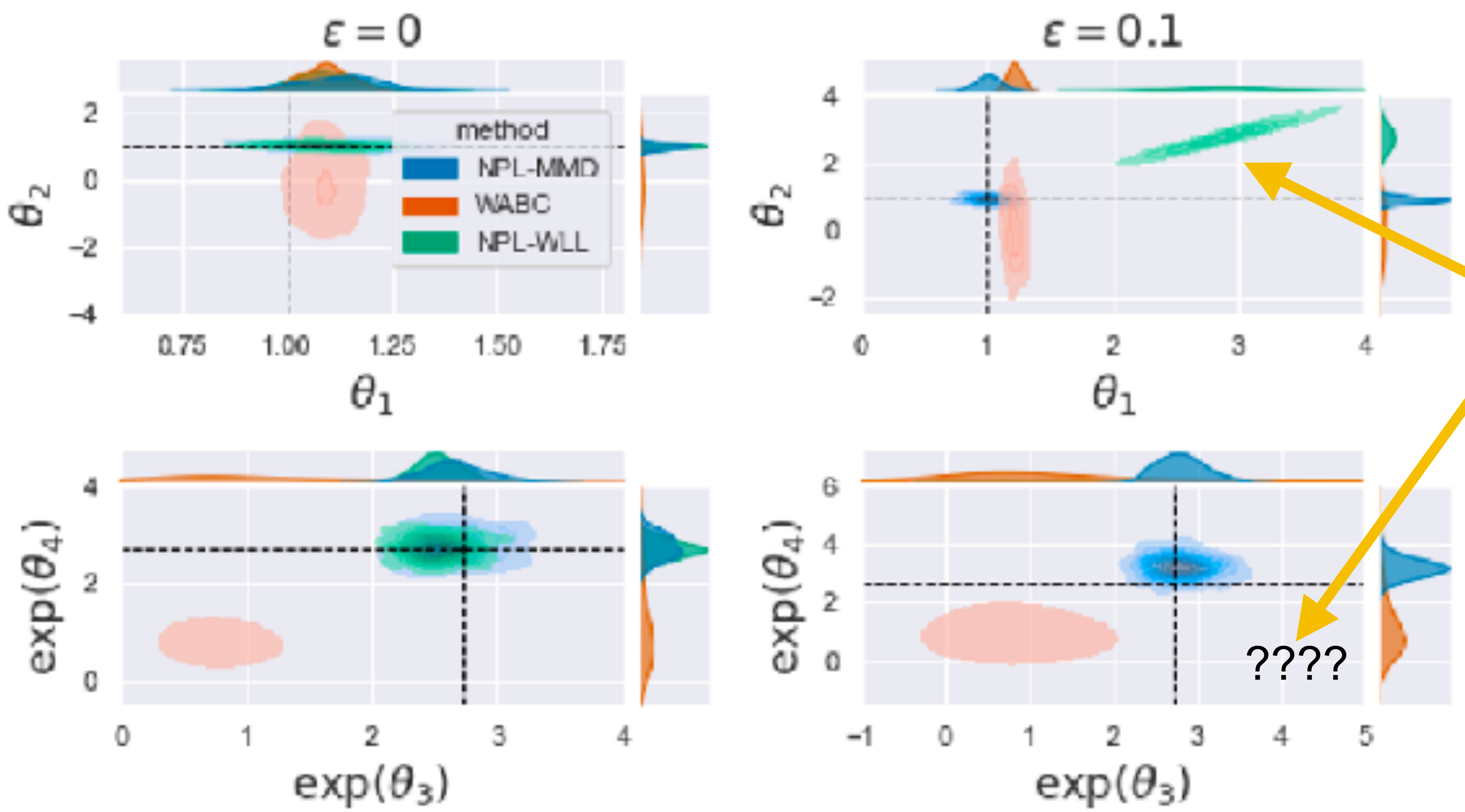
‘Easy’ parameters;
All do ok!

‘Hard’ parameters;
WABC already
struggles a bit



Well-specified case!

Example 1: Misspecified Gaussian

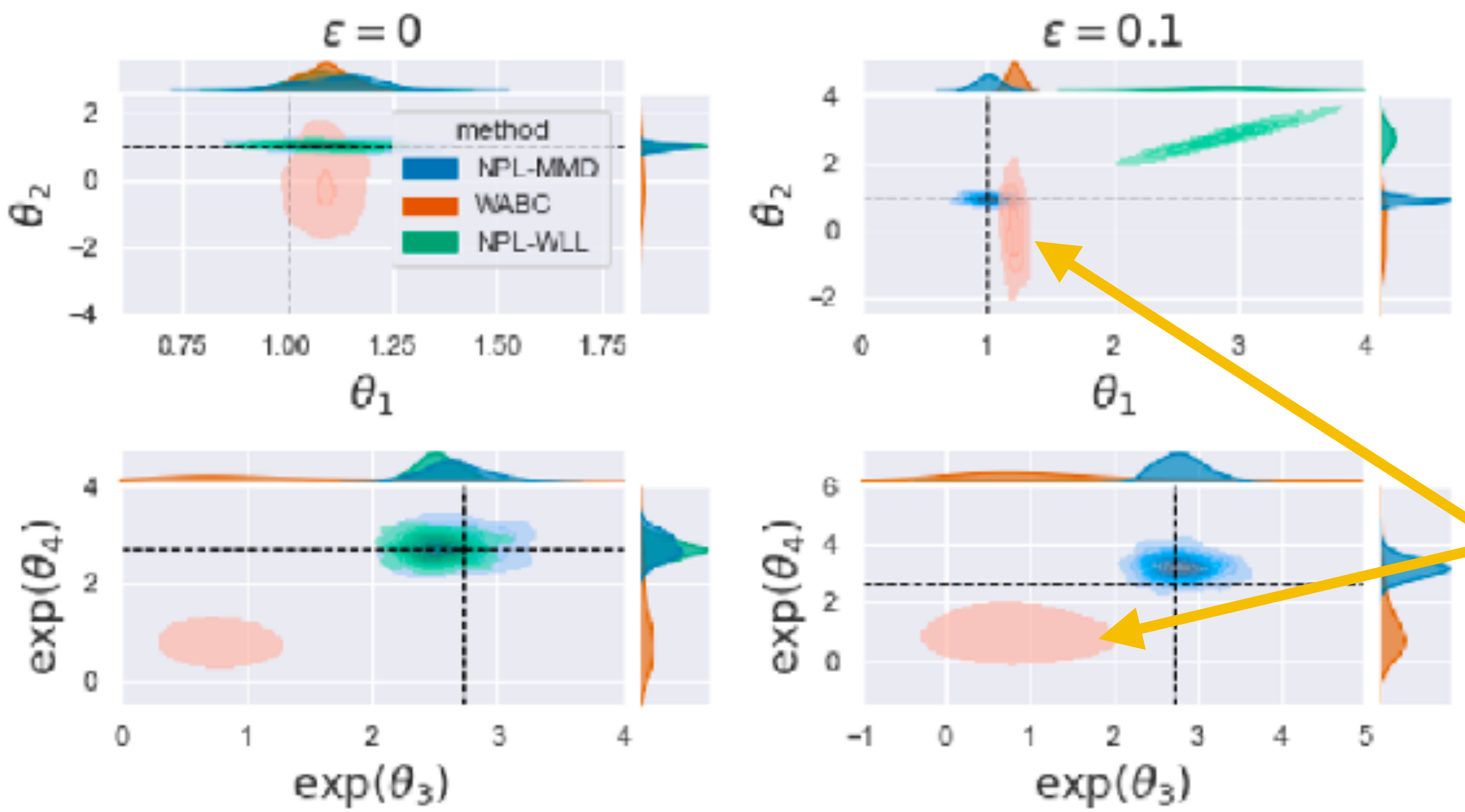


Misspecified case

NPL-WLL really struggles

????

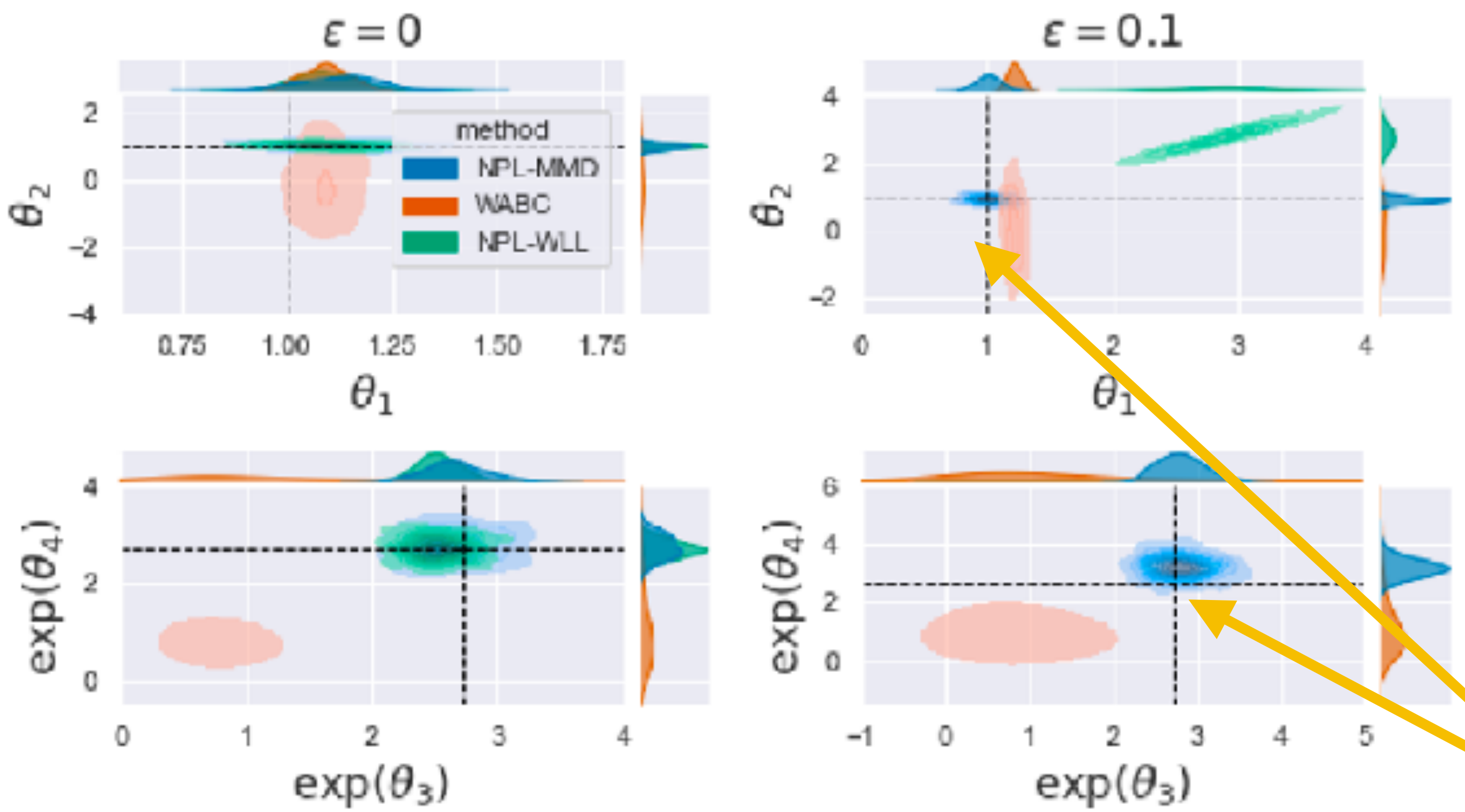
Example 1: Misspecified Gaussian



Misspecified case

WABC impacted!

Example 1: Misspecified Gaussian

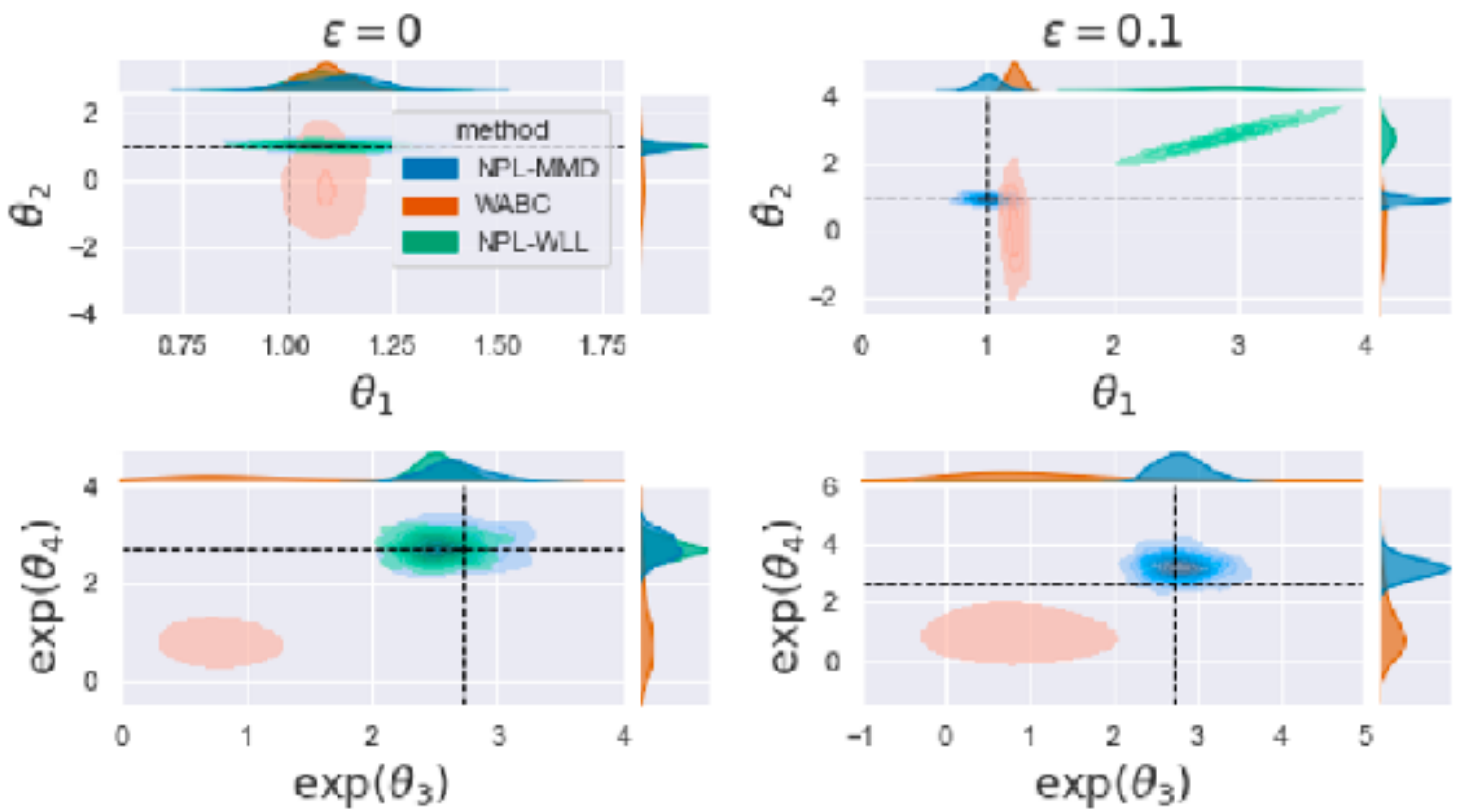


Misspecified case

MMD posterior bootstrap barely impacted!!

Example 1: Misspecified Gaussian

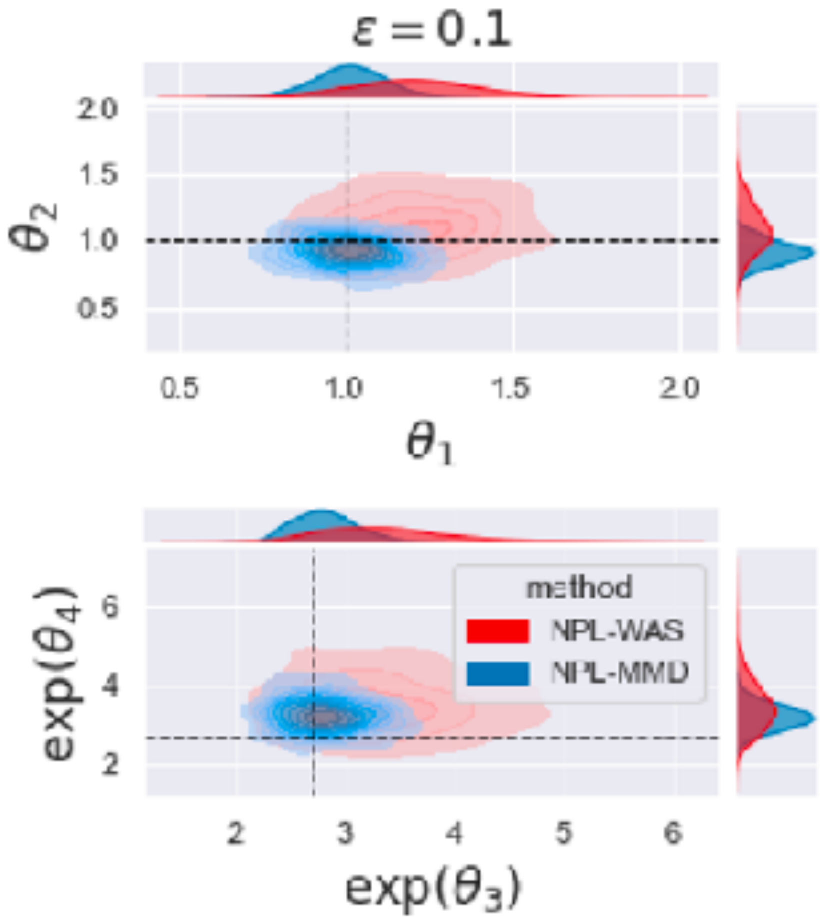
<u>Time to run:</u>
NPL-MMD: ≈ 2 mins
WABC: ≈ 1 hour



Misspecified case

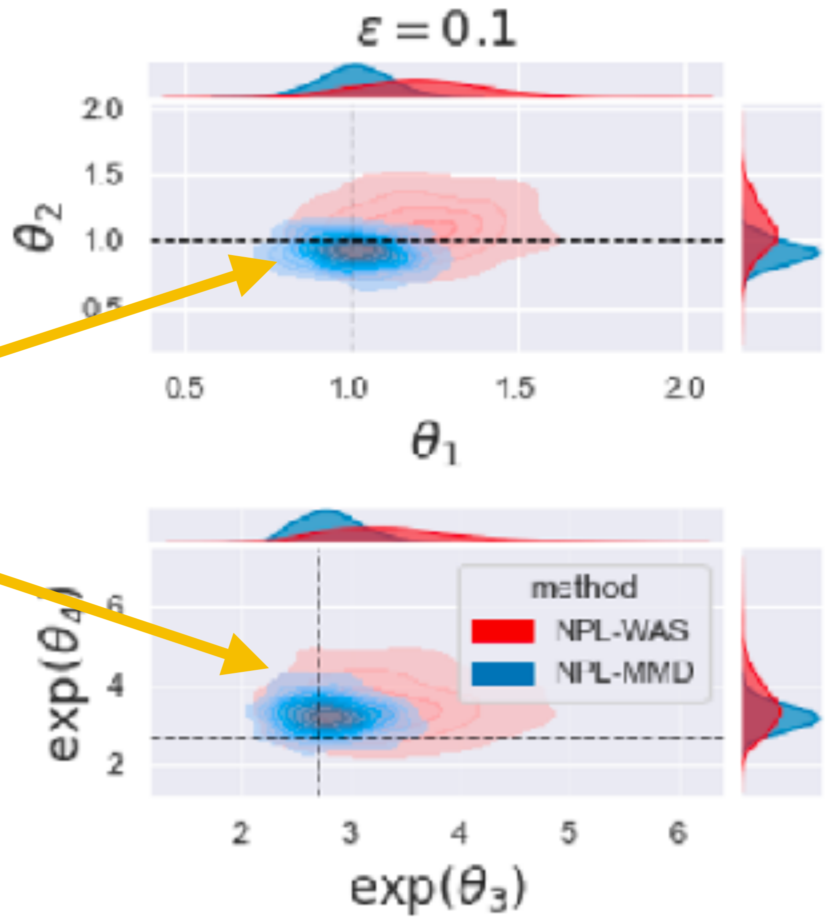
Example 1 continued: Wasserstein NPL

- In principle, nothing stops us from using the Wasserstein instead of MMD.



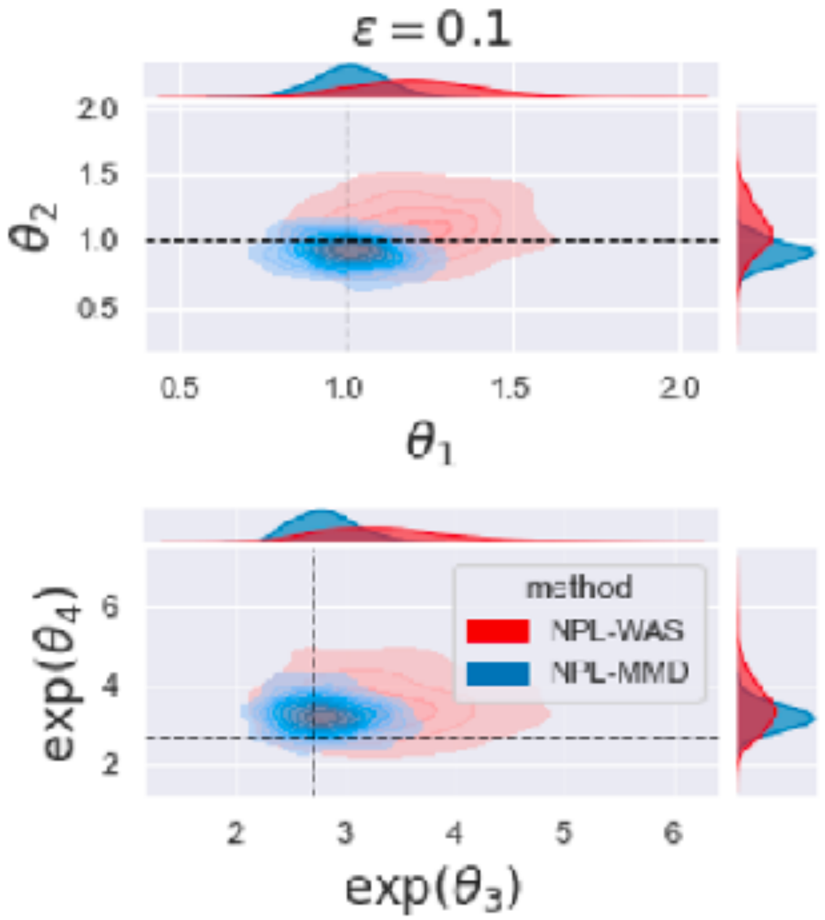
Example 1 continued: Wasserstein NPL

- In principle, nothing stops us from using the Wasserstein instead of MMD.
- The results are still reasonable thanks to NPL framework, but much more diffuse



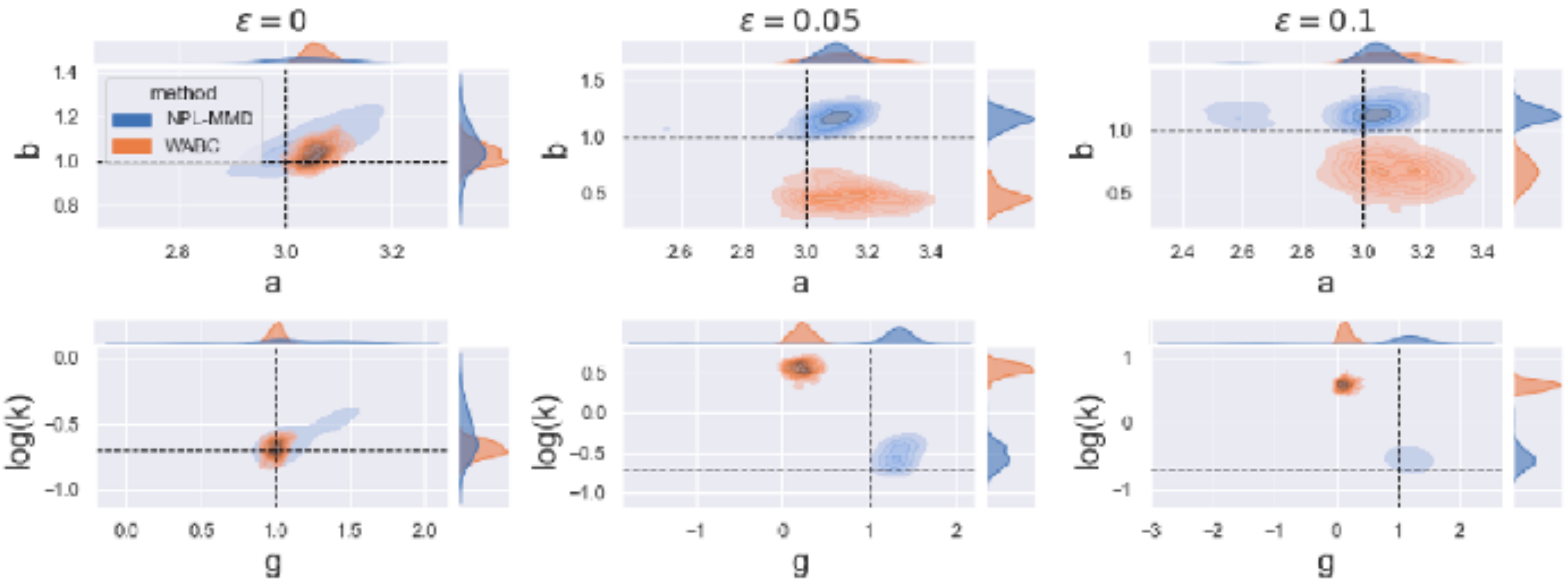
Example 1 continued: Wasserstein NPL

- In principle, nothing stops us from using the Wasserstein instead of MMD.
- The results are still reasonable thanks to NPL framework, but much more diffuse



We really do gain from having both a **robust inference** framework **AND** a **robust estimator**…

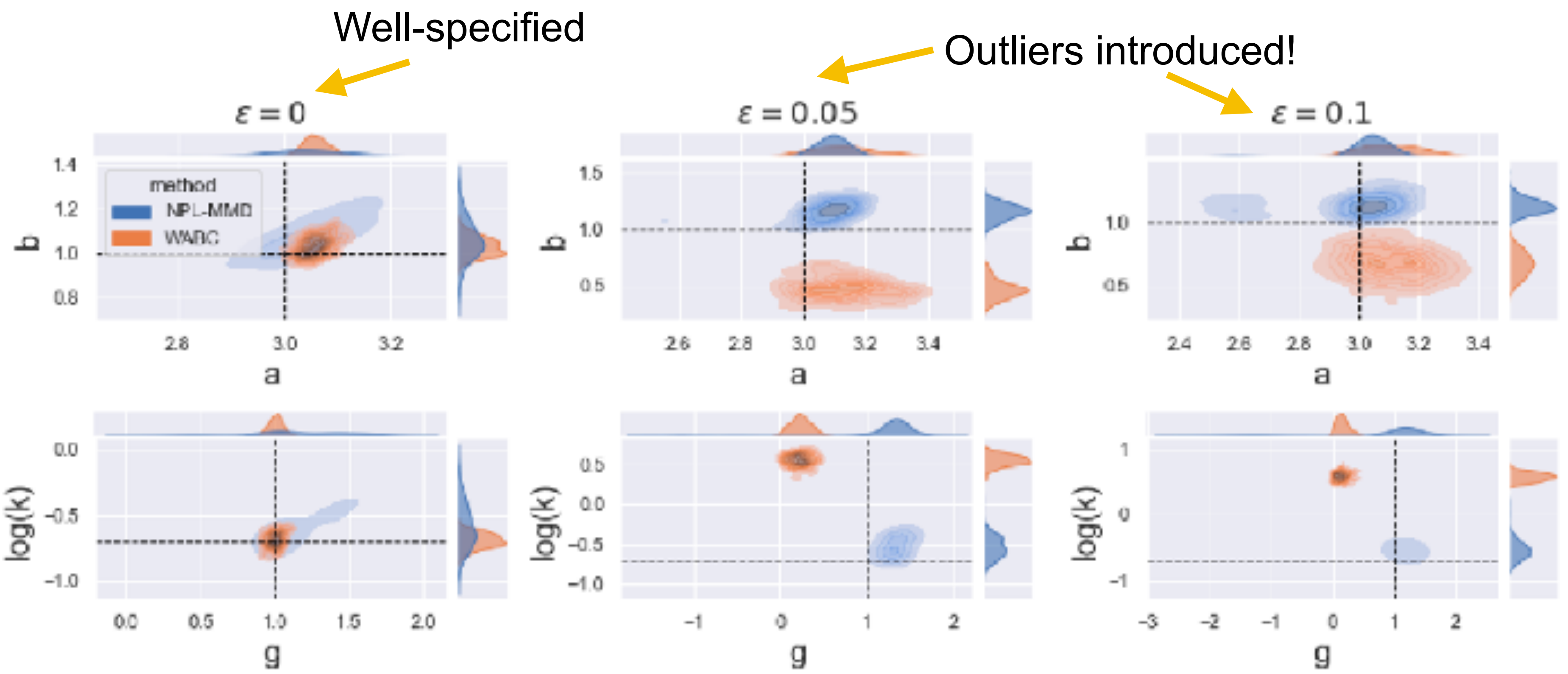
Misspecified g-and-k distribution



$$G_\theta(u) = \theta_1 + \theta_2 \left[1 + 0.8\left(\frac{1-\exp(-\theta_3 z(u))}{1+\exp(-\theta_3 z(u))}\right)\right]\left(1+z(u)^2\right)^{\log(\theta_4)} z(u),$$

$$z(u) = \Phi^{-1}(u) = \sqrt{2}\mathrm{erf}^{-1}(2u-1), \qquad u \sim \text{Unif}([0,1]),$$

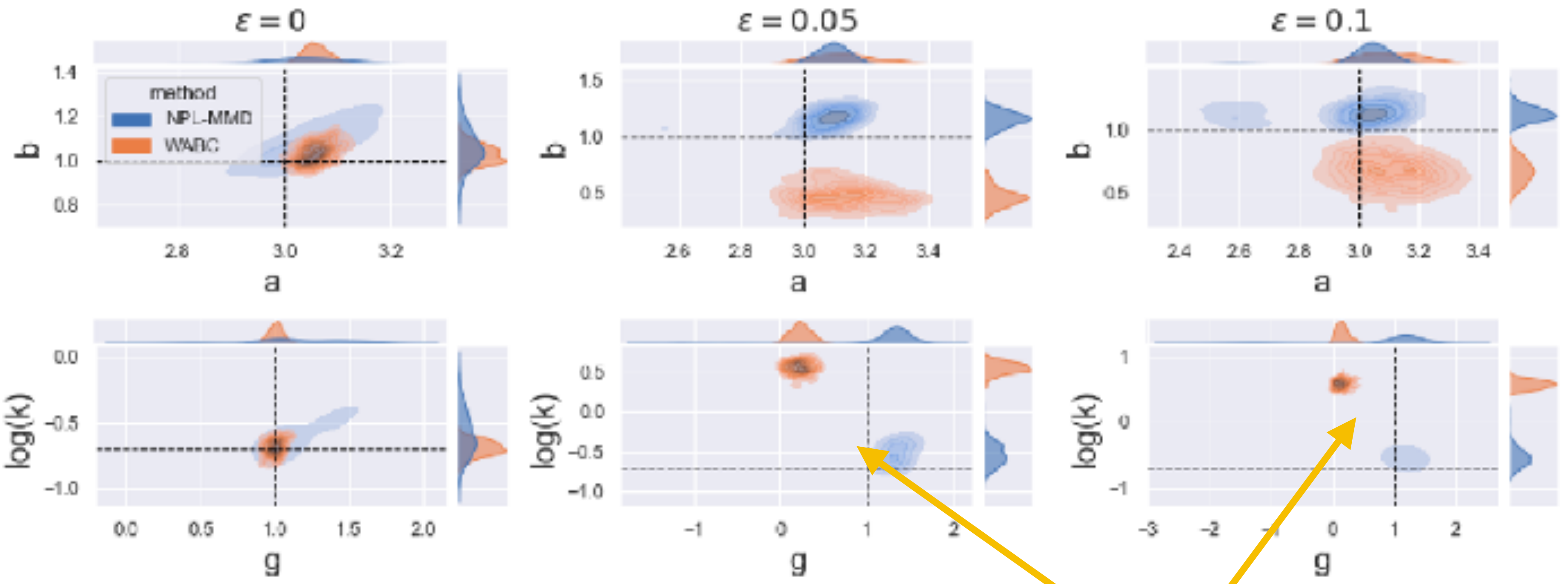
Misspecified g-and-k distribution



$$G_\theta(u) = \theta_1 + \theta_2 \left[1 + 0.8 \left(\frac{1 - \exp(-\theta_3 z(u))}{1 + \exp(-\theta_3 z(u))} \right) \right] \left(1 + z(u)^2\right)^{\log(\theta_4)} z(u),$$

$$z(u) = \Phi^{-1}(u) = \sqrt{2}\mathrm{erf}^{-1}(2u - 1), \qquad u \sim \text{Unif}([0,1]),$$

Misspecified g-and-k distribution



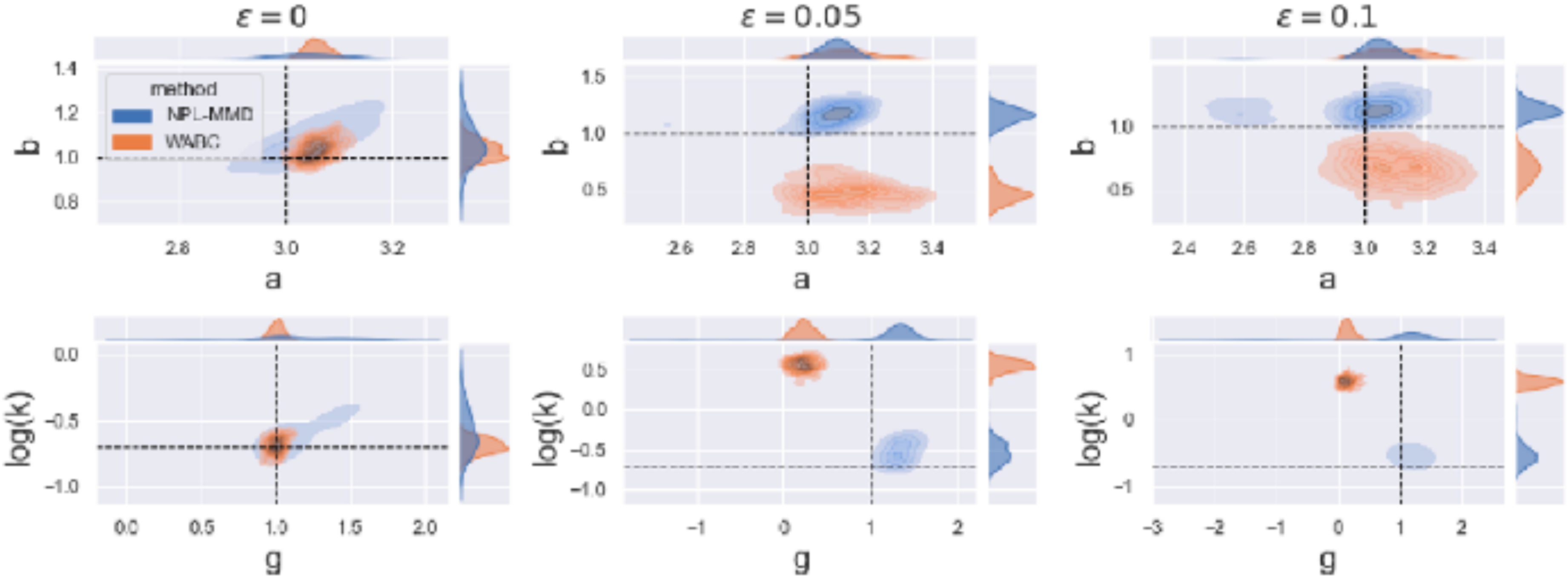
$$G_\theta(u) = \theta_1 + \theta_2 \left[1 + 0.8\left(\frac{1-\exp(-\theta_3 z(u))}{1+\exp(-\theta_3 z(u))}\right)\right]\left(1+z(u)^2\right)^{\log(\theta_4)} z(u),$$

$$z(u) = \Phi^{-1}(u) = \sqrt{2}\mathrm{erf}^{-1}(2u-1), \qquad u \sim \mathrm{Unif}([0,1]),$$

Wasserstein ABC really struggles with outliers, but the MMD posterior bootstrap is not significantly impacted

Misspecified g-and-k distribution

Time to run:
NPL-MMD: ≈ 30 sec
WABC: ≈ 100 sec



$$G_\theta(u) = \theta_1 + \theta_2 \left[1 + 0.8\left(\frac{1-\exp(-\theta_3 z(u))}{1+\exp(-\theta_3 z(u))}\right)\right]\left(1+z(u)^2\right)^{\log(\theta_4)} z(u),$$

$$z(u) = \Phi^{-1}(u) = \sqrt{2}\,\text{erf}^{-1}(2u-1), \qquad u \sim \text{Unif}([0,1]),$$

Going beyond iid…





- So far we have used:

$$(\mathbb{P}_\theta)_n = \frac{1}{n}\sum_{i=1}^{n}\delta_{x_i}, \qquad x_i = G_\theta(u_i), \qquad u_i \sim \text{Unif}[0,1]$$

Going beyond iid…





- So far we have used:

$$(\mathbb{P}_\theta)_n = \frac{1}{n}\sum_{i=1}^{n} \delta_{x_i}, \qquad x_i = G_\theta(u_i), \qquad u_i \sim \text{Unif}[0,1]$$

- Can do better with:

$$(\mathbb{P}_\theta)_n^w := \sum_{i=1}^{n} w_i \, \delta_{\tilde{x}_i}, \qquad \tilde{x}_i = G_\theta(\tilde{u}_i), \qquad \tilde{u}_i$$

Niu, Z., Meier, J., & **Briol, F.-X.** (2023). Discrepancy-based inference for intractable generative models using quasi-Monte Carlo. Electronic Journal of Statistics, 17(1), 1411–1456.

Bharti, A., Naslidnyk, M., Key, O., Kaski, S., & **Briol, F.-X.** (2023). Optimally-weighted estimators of the maximum mean discrepancy for likelihood-free inference. International Conference on Machine Learning, 2289–2312.

Going beyond iid…





- So far we have used:

$$(\mathbb{P}_\theta)_n = \frac{1}{n}\sum_{i=1}^{n} \delta_{x_i}, \qquad x_i = G_\theta(u_i), \qquad u_i \sim \text{Unif}[0,1]$$

- Can do better with:

Non-equal weights

Grids

$$(\mathbb{P}_\theta)_n^w := \sum_{i=1}^{n} \boxed{w_i}\, \delta_{\tilde{x}_i}, \qquad \tilde{x}_i = G_\theta(\tilde{u}_i), \qquad \boxed{\tilde{u}_i}$$

Niu, Z., Meier, J., & **Briol, F.-X.** (2023). Discrepancy-based inference for intractable generative models using quasi-Monte Carlo. Electronic Journal of Statistics, 17(1), 1411–1456.

Bharti, A., Naslidnyk, M., Key, O., Kaski, S., & **Briol, F.-X.** (2023). Optimally-weighted estimators of the maximum mean discrepancy for likelihood-free inference. International Conference on Machine Learning, 2289–2312.

Going beyond iid…



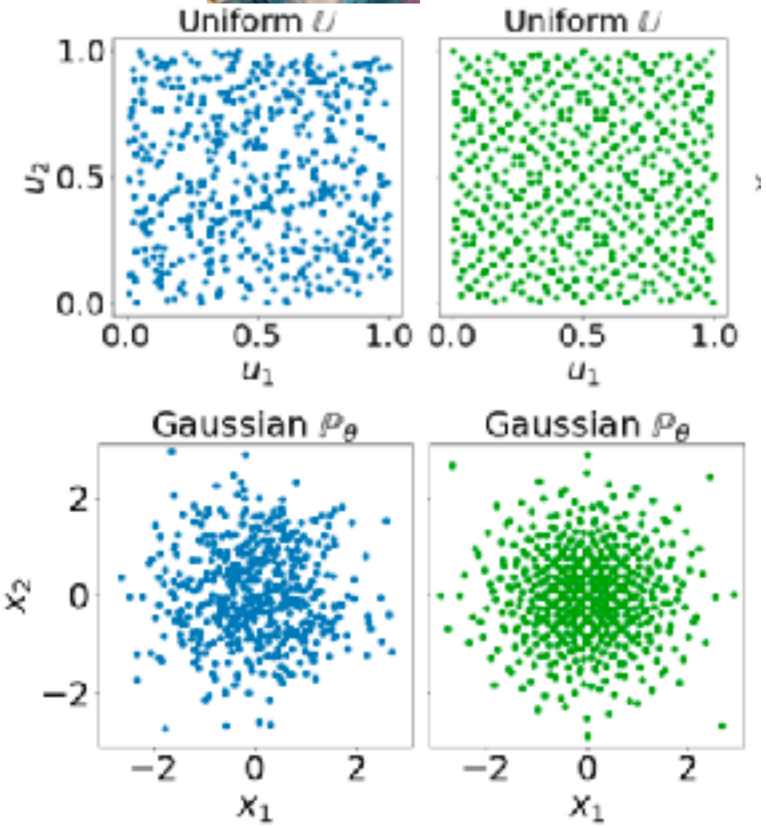


- So far we have used:

$$(\mathbb{P}_\theta)_n = \frac{1}{n}\sum_{i=1}^{n}\delta_{x_i}, \qquad x_i = G_\theta(u_i), \qquad u_i \sim \text{Unif}[0,1]$$

- Can do better with: Non-equal weights Grids

$$(\mathbb{P}_\theta)_n^w := \sum_{i=1}^{n}\boxed{w_i}\,\delta_{\tilde{x}_i}, \qquad \tilde{x}_i = G_\theta(\tilde{u}_i), \qquad \boxed{\tilde{u}_i}$$



Niu, Z., Meier, J., & **Briol, F.-X.** (2023). Discrepancy-based inference for intractable generative models using quasi-Monte Carlo. Electronic Journal of Statistics, 17(1), 1411–1456.

Bharti, A., Naslidnyk, M., Key, O., Kaski, S., & **Briol, F.-X.** (2023). Optimally-weighted estimators of the maximum mean discrepancy for likelihood-free inference. International Conference on Machine Learning, 2289–2312.

Hypothesis testing for simulator misspecification



H_0: Model/simulator is well-specified.

H_1: Model/simulator is misspecified.

Key, O., Gretton, A., **Briol, F-X.**, & Fernandez, T. (2025). Composite goodness-of-fit tests with kernels. *JMLR*, *26*(51), 1–60.

Hypothesis testing for simulator misspecification

H_0: Model/simulator is well-specified.

H_1: Model/simulator is misspecified.

Test statistic: $$\Delta_n = \inf_{\theta \in \Theta} \mathrm{MMD}^2(\mathbb{P}_\theta, \mathbb{Q}_n)$$

Key, O., Gretton, A., **Briol, F-X.**, & Fernandez, T. (2025). Composite goodness-of-fit tests with kernels. *JMLR*, *26*(51), 1–60.

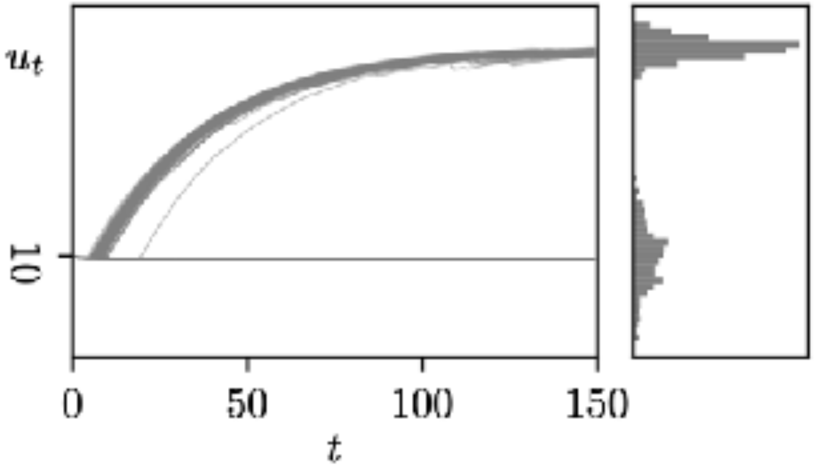
Hypothesis testing for simulator misspecification



H_0: Model/simulator is well-specified.
H_1: Model/simulator is misspecified.

Test statistic: $$\Delta_n = \inf_{\theta \in \Theta} \mathrm{MMD}^2(\mathbb{P}_\theta, \mathbb{Q}_n)$$

Toggle-switch model:



Key, O., Gretton, A., **Briol, F-X.**, & Fernandez, T. (2025). Composite goodness-of-fit tests with kernels. *JMLR*, *26*(51), 1–60.

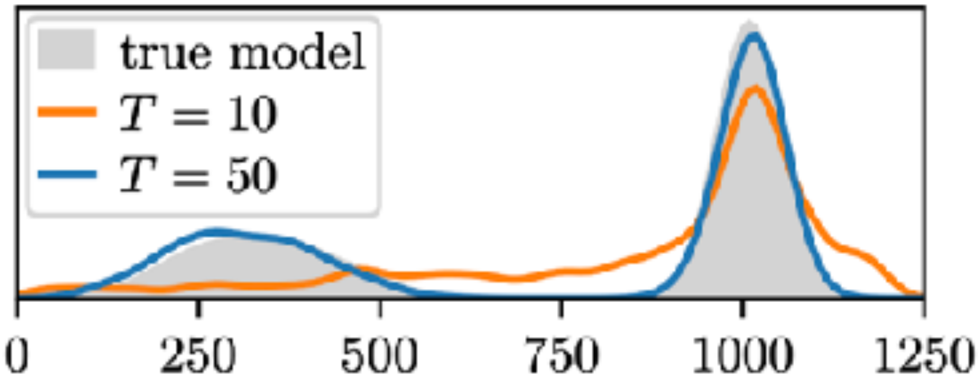
Hypothesis testing for simulator misspecification

H_0: Model/simulator is well-specified.

H_1: Model/simulator is misspecified.

Test statistic: $$\Delta_n = \inf_{\theta \in \Theta} \mathrm{MMD}^2(\mathbb{P}_\theta, \mathbb{Q}_n)$$

Toggle-switch model:



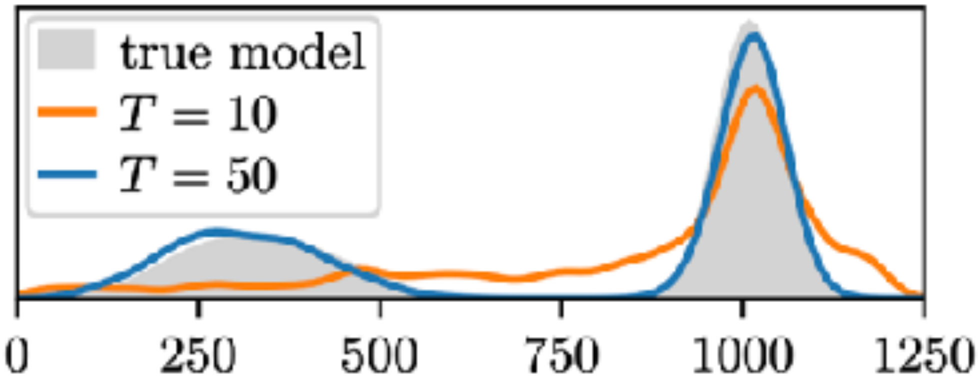
Key, O., Gretton, A., **Briol, F-X.**, & Fernandez, T. (2025). Composite goodness-of-fit tests with kernels. *JMLR*, *26*(51), 1–60.

Hypothesis testing for simulator misspecification

H_0: Model/simulator is well-specified.

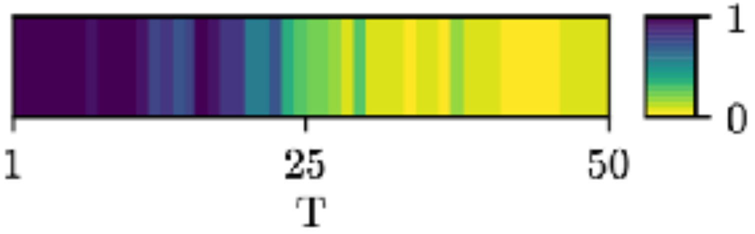
H_1: Model/simulator is misspecified.

Toggle-switch model:



Test statistic: $$\Delta_n = \inf_{\theta \in \Theta} \mathrm{MMD}^2(\mathbb{P}_\theta, \mathbb{Q}_n)$$

% of rejects:



Key, O., Gretton, A., **Briol, F-X.**, & Fernandez, T. (2025). Composite goodness-of-fit tests with kernels. *JMLR*, *26*(51), 1–60.

Hypothesis testing for simulator misspecification

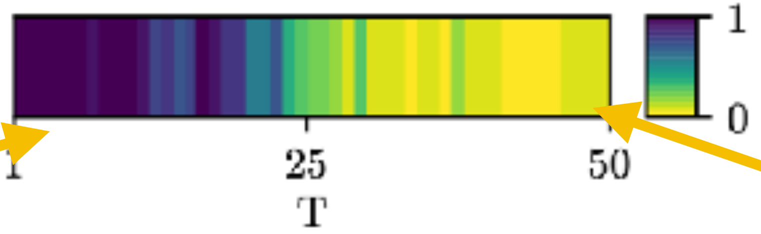


H_0: Model/simulator is well-specified.

H_1: Model/simulator is misspecified.

Test statistic: $$\Delta_n = \inf_{\theta \in \Theta} \mathrm{MMD}^2(\mathbb{P}_\theta, \mathbb{Q}_n)$$

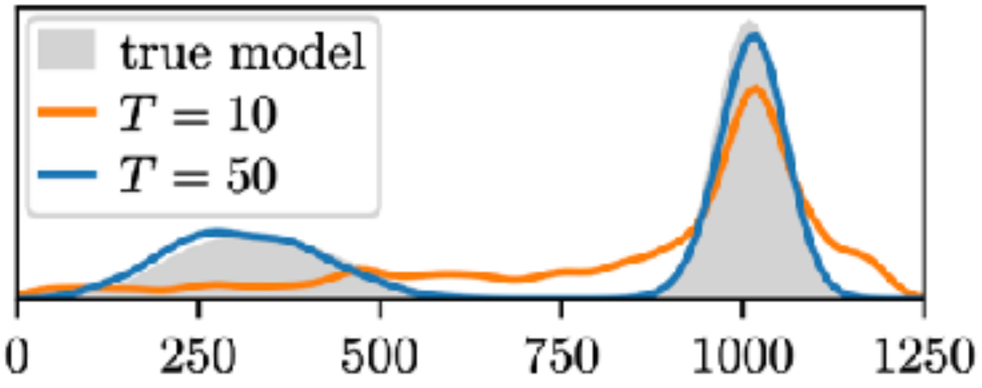
% of rejects:



Easy to distinguish

Hard to distinguish

Toggle-switch model:



Key, O., Gretton, A., **Briol, F-X.**, & Fernandez, T. (2025). Composite goodness-of-fit tests with kernels. *JMLR*, *26*(51), 1–60.

Looking ahead….

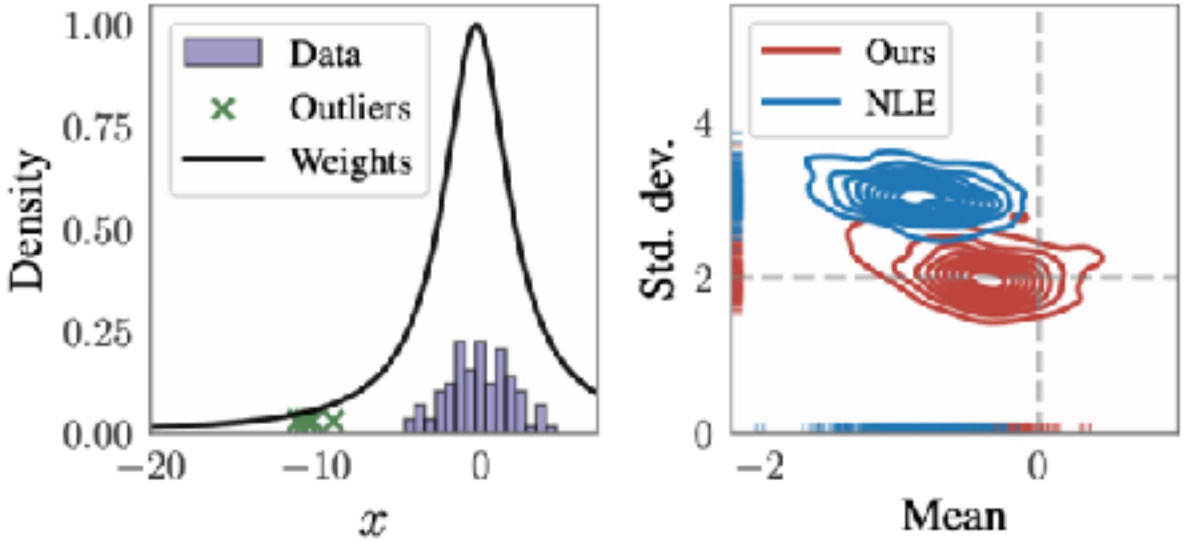


- None of the methods in this section are well-suited for amortisation…

Looking ahead....



- None of the methods in this section are well-suited for amortisation…

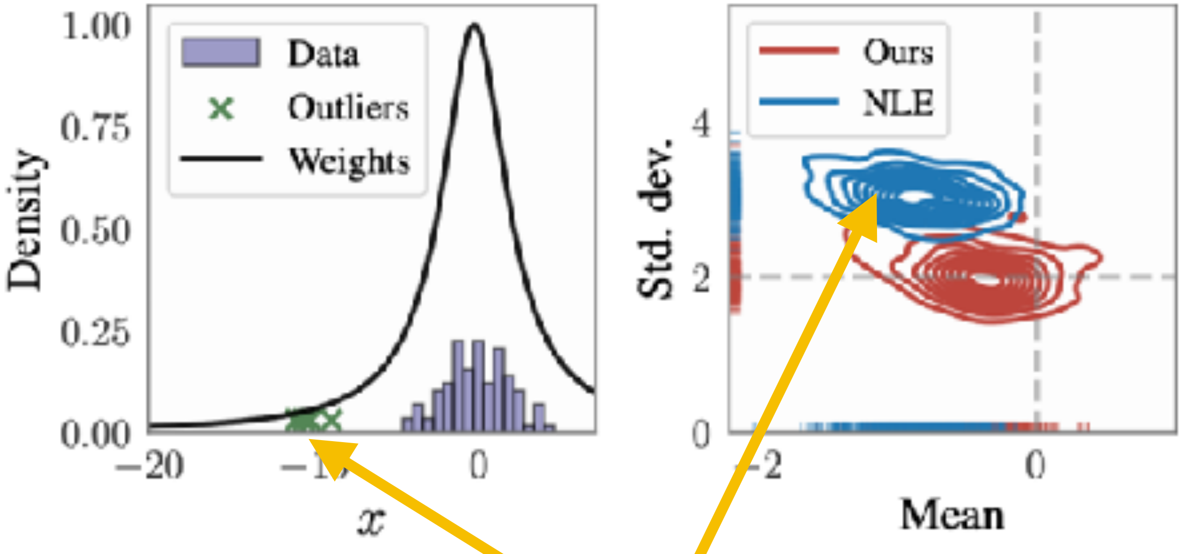


Bharti, A., Dellaporta, H., Hikida, Y., & **Briol, F-X.** (2025+). Amortised and provably-robust simulation-based inference.

Looking ahead….



- None of the methods in this section are well-suited for amortisation…



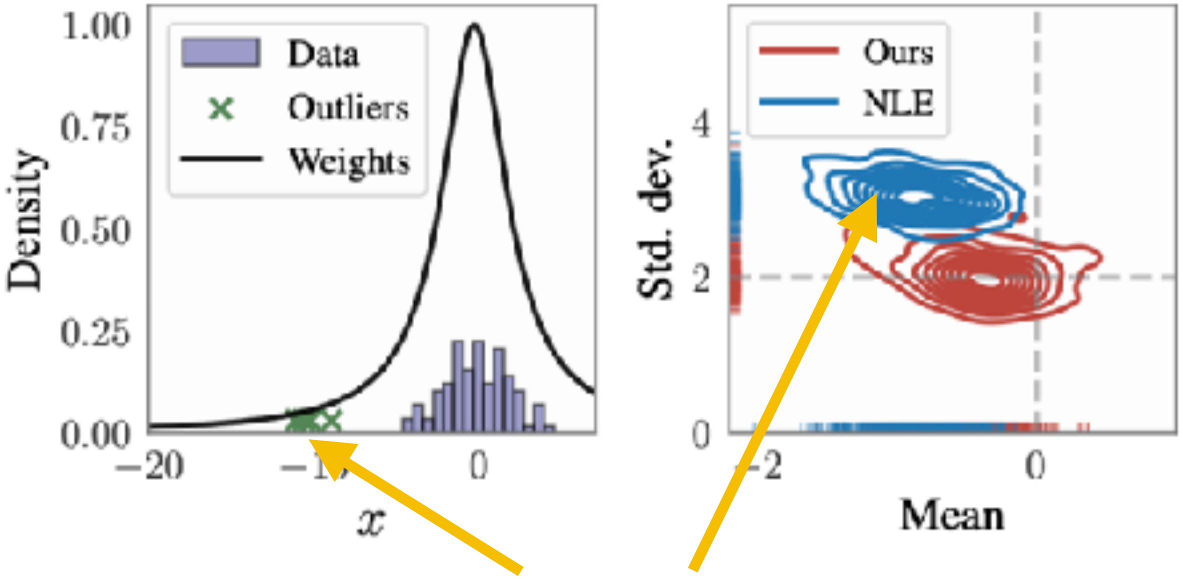
A few outliers can have a drastic impact on NLE!!

Bharti, A., Dellaporta, H., Hikida, Y., & **Briol, F-X.** (2025+). Amortised and provably-robust simulation-based inference.

Looking ahead....

- None of the methods in this section are well-suited for amortisation…

- Existing methods are either provably robust or amortised, but **not both**…!



A few outliers can have a drastic impact on NLE!!

Bharti, A., Dellaporta, H., Hikida, Y., & **Briol, F-X.** (2025+). Amortised and provably-robust simulation-based inference.

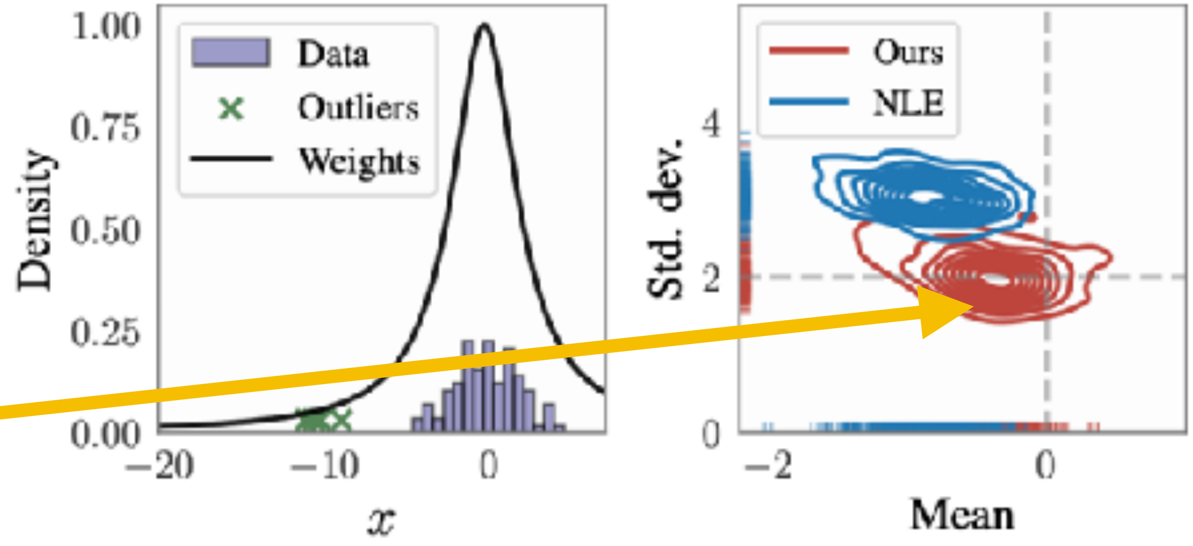
Looking ahead….



- None of the methods in this section are well-suited for amortisation…

- Existing methods are either provably robust or amortised, but **not both**…!

- Currently working on a novel gen-Bayes method to resolve this.



Bharti, A., Dellaporta, H., Hikida, Y., & **Briol, F-X.** (2025+). Amortised and provably-robust simulation-based inference.

Any Questions?

Paper: Dellaporta, C., Knoblauch, J., Damoulas, T. & **Briol, F-X** (2022). Robust Bayesian inference for simulator-based models via the MMD posterior bootstrap. AISTATS, 943-970. Best paper award.

Code: https://github.com/haritadell/npl_mmd_project

Summary of this course

- Basic methods:

Minimum distance estimation

Approximate Bayesian Computation

Neural simulation-based inference

- Modern Challenges for SBI (expensive simulators, misspecification, calibration, high-dimensionality).

- Some illustrations of recent advances:

Hikida, Y., Bharti, A., Jeffrey, N. & **Briol, F-X** (2025). Multilevel neural simulation-based inference. arXiv:2506.06087 (to appear at NeurIPS?).

Bharti, A., Huang, D., Kaski, S., & **Briol, F.-X.** (2025). Cost-aware simulation-based inference. International Conference on Artificial Intelligence and Statistics, 28–36.

Dellaporta, C., Knoblauch, J., Damoulas, T. & **Briol, F-X** (2022). Robust Bayesian inference for simulator-based models via the MMD posterior bootstrap. AISTATS, 943-970. Best paper award.

What I haven't covered…. but probably should have!

- **Alternative methodology:** indirect inference, synthetic likelihoods, doubly-intractable problems, etc…

What I haven’t covered…. but probably should have!

- **Alternative methodology:** indirect inference, synthetic likelihoods, doubly-intractable problems, etc…
- **Advanced emulators:** GANs, flow matching, diffusion models, etc…

What I haven't covered…. but probably should have!

- **Alternative methodology:** indirect inference, synthetic likelihoods, doubly-intractable problems, etc…
- **Advanced emulators:** GANs, flow matching, diffusion models, etc…
- **Theory:** asymptotics, robustness, theory for normalising flows, etc…

What I haven't covered…. but probably should have!

- **Alternative methodology:** indirect inference, synthetic likelihoods, doubly-intractable problems, etc…
- **Advanced emulators:** GANs, flow matching, diffusion models, etc…
- **Theory:** asymptotics, robustness, theory for normalising flows, etc…
- **Software:** sbi, bayesflow, etc…

Some personal take-aways

- Where should we go next?

→ Need to provide **rigour** and **strong theoretical guarantees** so we can use these methods to do serious science…

Some personal take-aways

- Where should we go next?

  → Need to provide **rigour** and **strong theoretical guarantees** so we can use these methods to do serious science…

- Where are the computational statisticians (including me)?!

  → They were sleeping, but are slowly waking up to neural-based methods!