Multilevel neural simulation-based inference

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

Yuga Hikida
(Aalto)

Ayush Bharti
(Aalto)

Niall Jeffrey
(UCL)

Simulation-based inference (SBI)

Bayesian inference:

$$\pi(\theta \mid y_1, \ldots, y_m) \propto \prod_{i=1}^{m} p(y_i \mid \theta)\pi(\theta)$$

Simulation-based inference (SBI)

Bayesian inference:

$$\pi(\theta \,|\, y_1, \ldots, y_m) \propto \prod_{i=1}^{m} p(y_i \,|\, \theta) \pi(\theta)$$

Simulation-based inference (SBI)

Bayesian inference:

$$\pi(\theta \mid y_1, \ldots, y_m) \propto \prod_{i=1}^{m} p(y_i \mid \theta) \pi(\theta)$$

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

Deterministic map
(generator)

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

Simulation-based inference (SBI)

Bayesian inference:

$$\pi(\theta \,|\, y_1, \ldots, y_m) \propto \prod_{i=1}^{m} p(y_i \,|\, \theta)\pi(\theta)$$

Two main approaches:

- Approximate Bayesian computation (ABC).

$$\underbrace{x_i = G_\theta}_{\text{Deterministic map (generator)}}(\underbrace{u_i}_{u_i \sim \mathbb{U} \text{ (randomness)}})$$

Deterministic map (generator)

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

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

Simulation-based inference (SBI)

Bayesian inference:

$$\pi(\theta \,|\, y_1, \ldots, y_m) \propto \prod_{i=1}^{m} p(y_i \,|\, \theta) \pi(\theta)$$

Two main approaches:

- Approximate Bayesian computation (ABC).
- Neural-based simulation-based inference.

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

Deterministic map (generator)

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

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

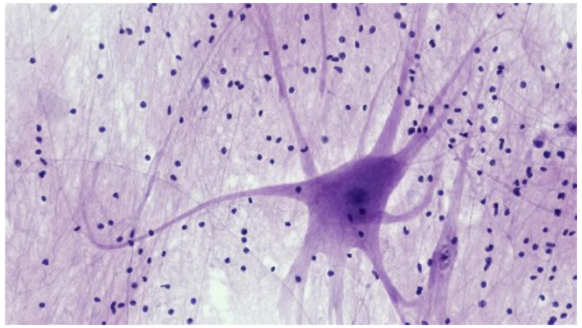
Cranmer, K., Brehmer, J., & Louppe, G. (2020). The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences of the United States of America*, *117*(48).

Simulators in the sciences and beyond

Particle Physics (CERN)

Simulators in the sciences and beyond

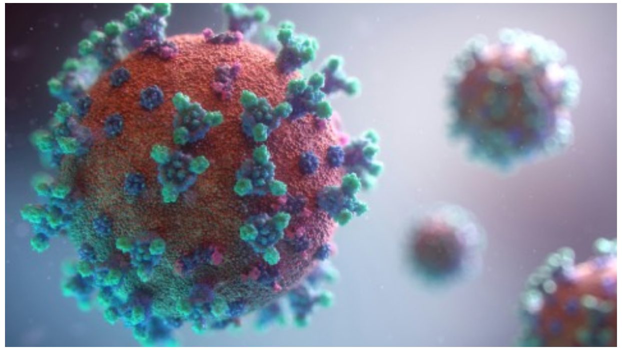
Particle Physics (CERN)



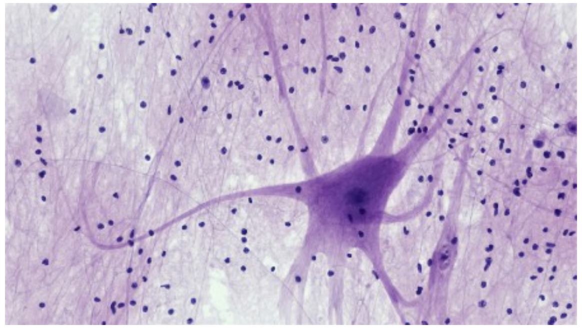
Neuroscience

Simulators in the sciences and beyond

Particle Physics (CERN)



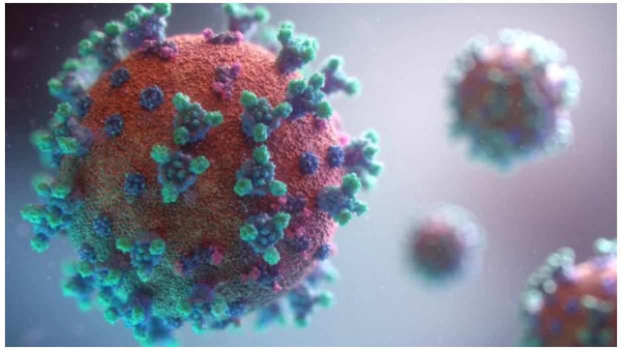
Epidemiology



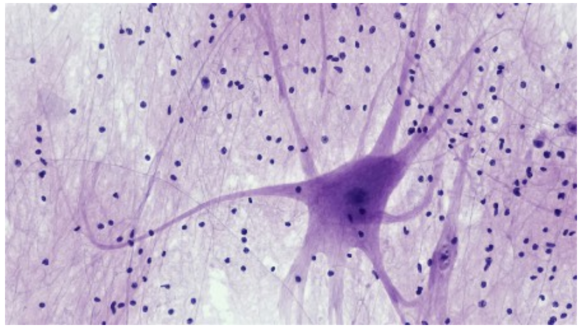
Neuroscience

Simulators in the sciences and beyond

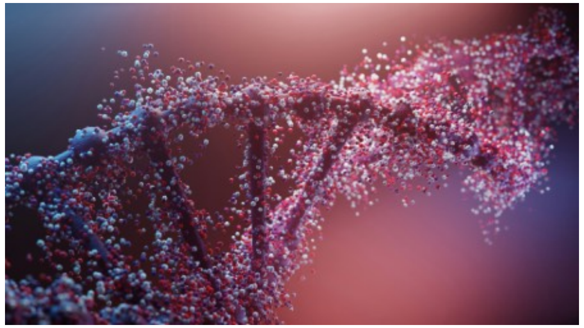
Particle Physics (CERN)



Epidemiology



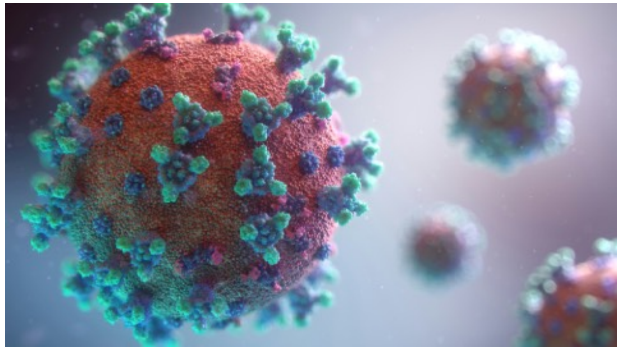
Neuroscience



Genomics

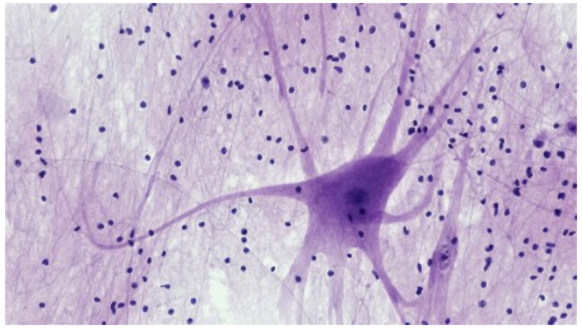
Simulators in the sciences and beyond

Particle Physics (CERN)

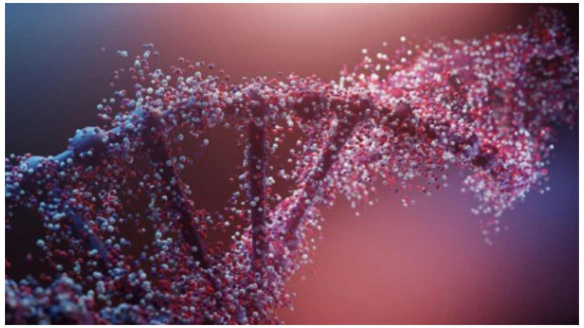


Epidemiology

Health monitoring (Apple)



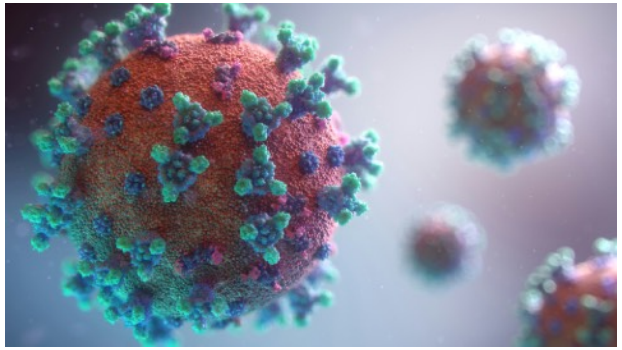
Neuroscience



Genomics

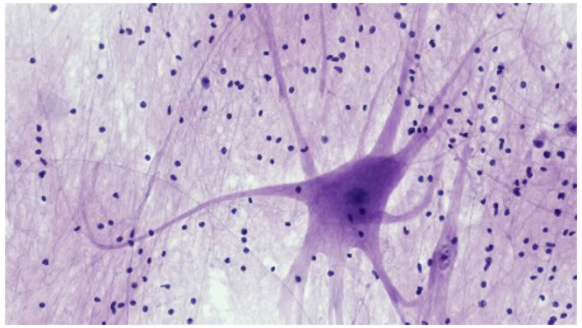
Simulators in the sciences and beyond

Particle Physics (CERN)

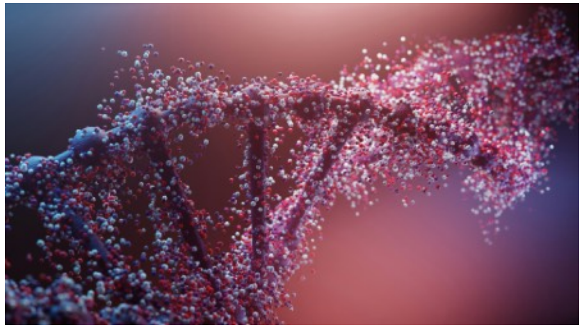


Epidemiology

Health monitoring (Apple)



Neuroscience



Genomics

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

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.

→ Many large-scale scientific applications are still out of reach!

Challenge for SBI

Simulators can be really computationally expensive!

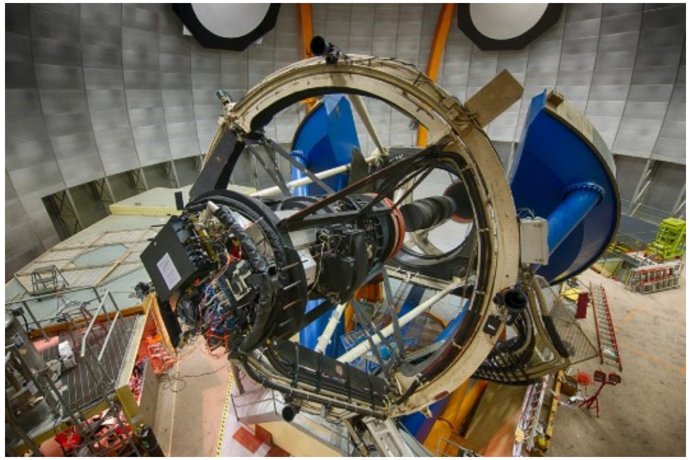
- Most simulators used in SBI papers take only a few seconds (or less) to run.
- Even if a simulator takes only a few minutes, we typically need thousands of simulations!
- Simulators that take more time are currently out of reach of existing methods.

→ Many large-scale scientific applications are still out of reach!

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

A motivating application

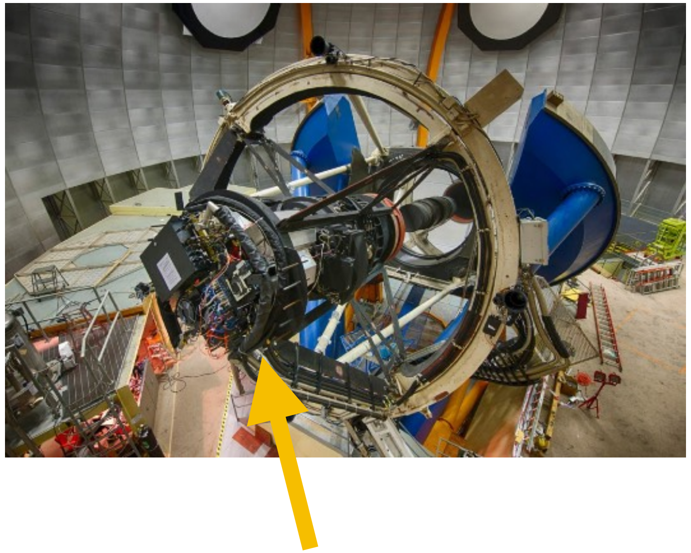
(+ ≈ 400 scientists from 25 institutions in 7 countries)



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.

A motivating application

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

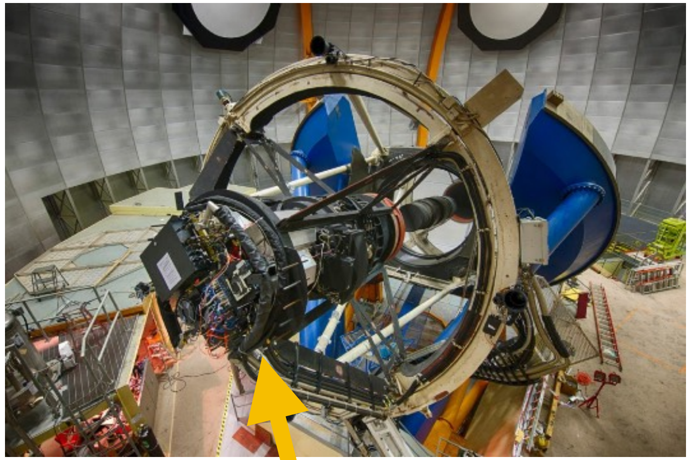


The Dark energy survey camera!

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.

A motivating application

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

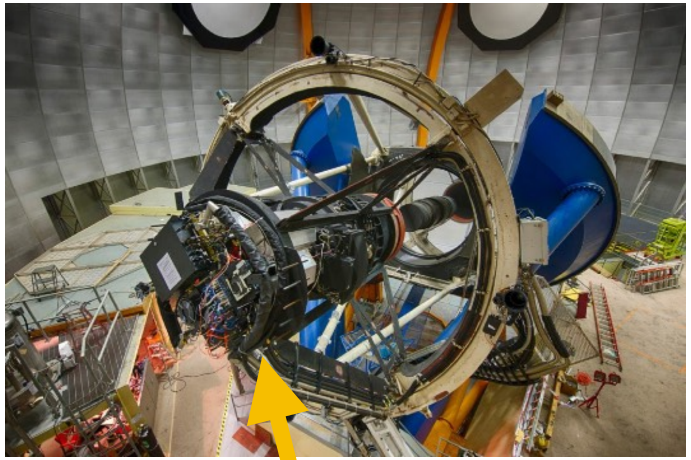


The Dark energy survey camera!

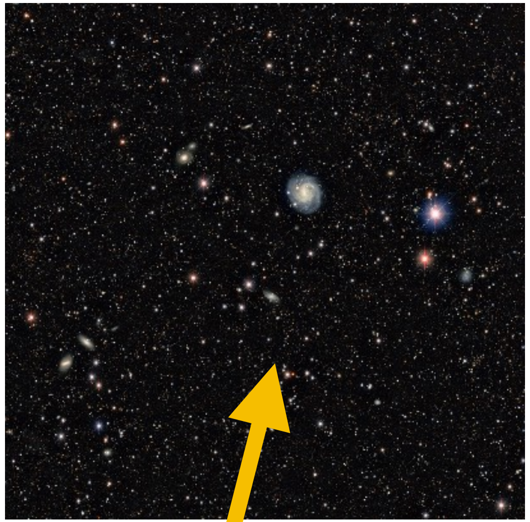
Jeffrey, N., et al. (2025). Dark energy survey year 3 results: likelihood-free, simulation-based wCDM inference with neural compression of weak-lensing map statistics. *Monthly Notices of the Royal Astronomical Society*, *536*(2), 1303–1322.

A motivating application

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



The Dark energy survey camera!

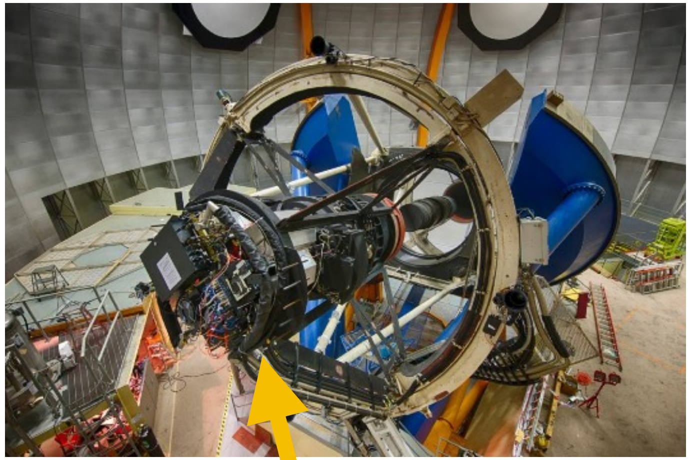


Data collected through the Dark energy survey camera

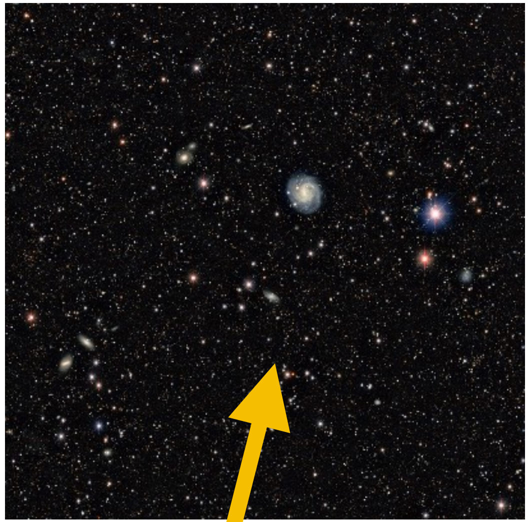
Jeffrey, N., et al. (2025). Dark energy survey year 3 results: likelihood-free, simulation-based wCDM inference with neural compression of weak-lensing map statistics. *Monthly Notices of the Royal Astronomical Society*, *536*(2), 1303–1322.

A motivating application

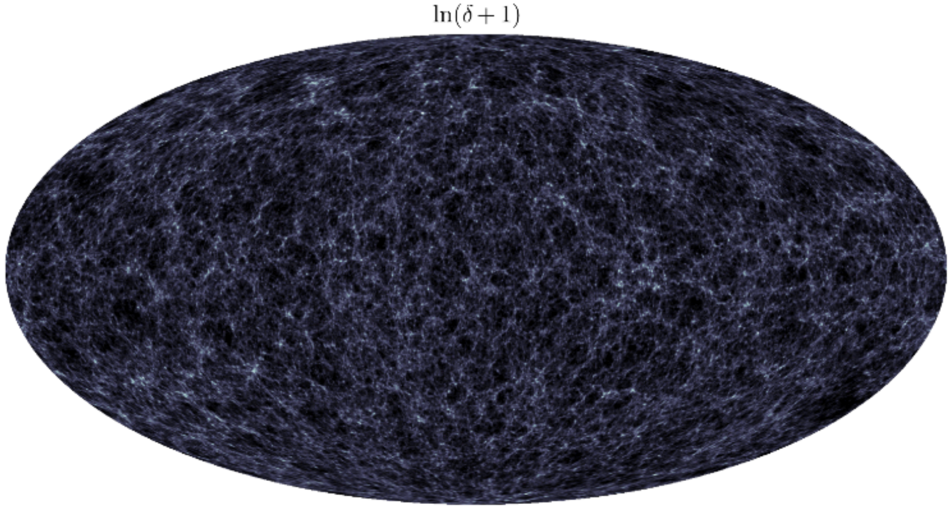
(+ $\approx$ 400 scientists from 25 institutions in 7 countries)



The Dark energy survey camera!



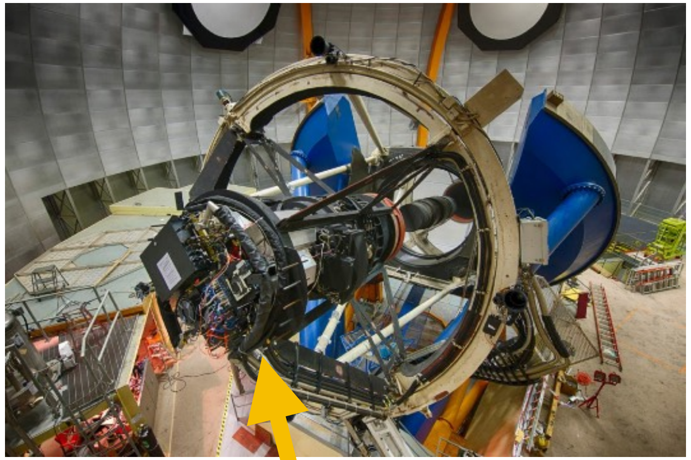
Data collected through the Dark energy survey camera



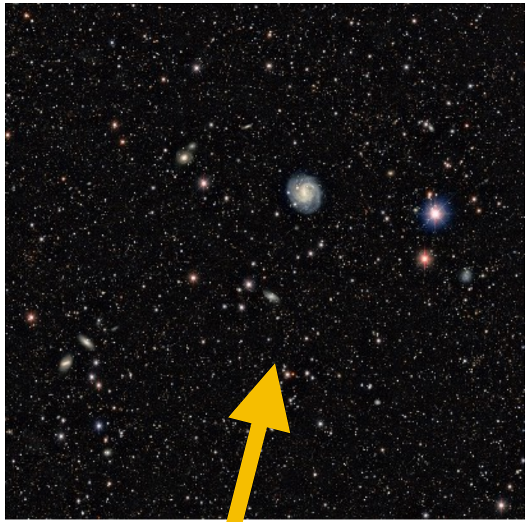
Jeffrey, N., et al. (2025). Dark energy survey year 3 results: likelihood-free, simulation-based wCDM inference with neural compression of weak-lensing map statistics. *Monthly Notices of the Royal Astronomical Society*, *536*(2), 1303–1322.

A motivating application

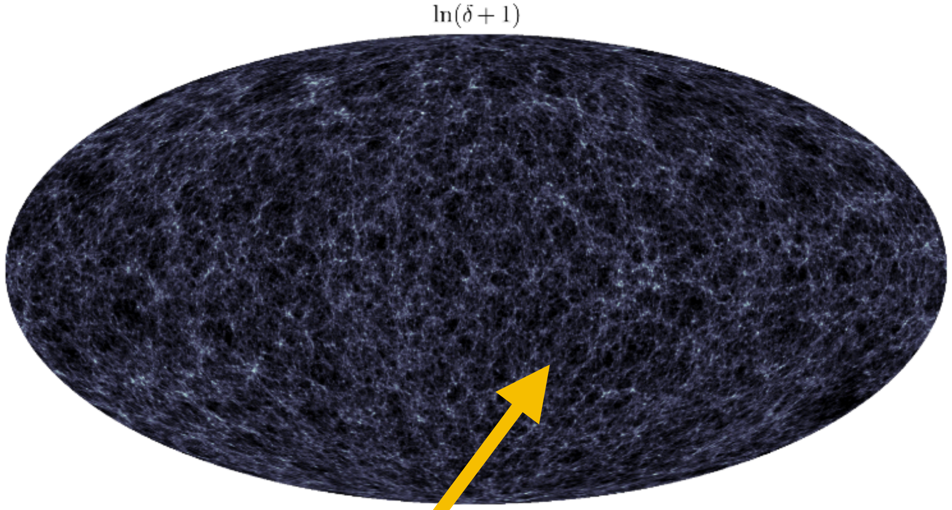
(+ $\approx$ 400 scientists from 25 institutions in 7 countries)



The Dark energy survey camera!



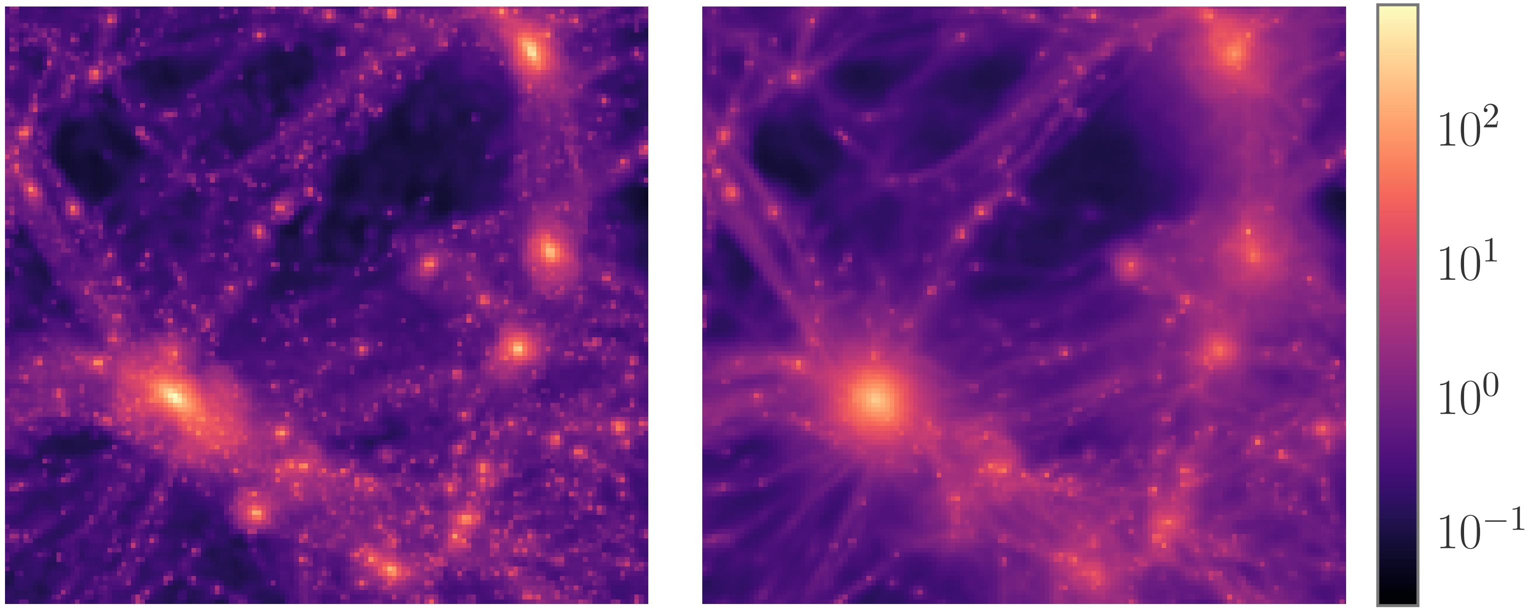
Data collected through the Dark energy survey camera



'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.

SBI for cosmology



Low-fidelity High-fidelity

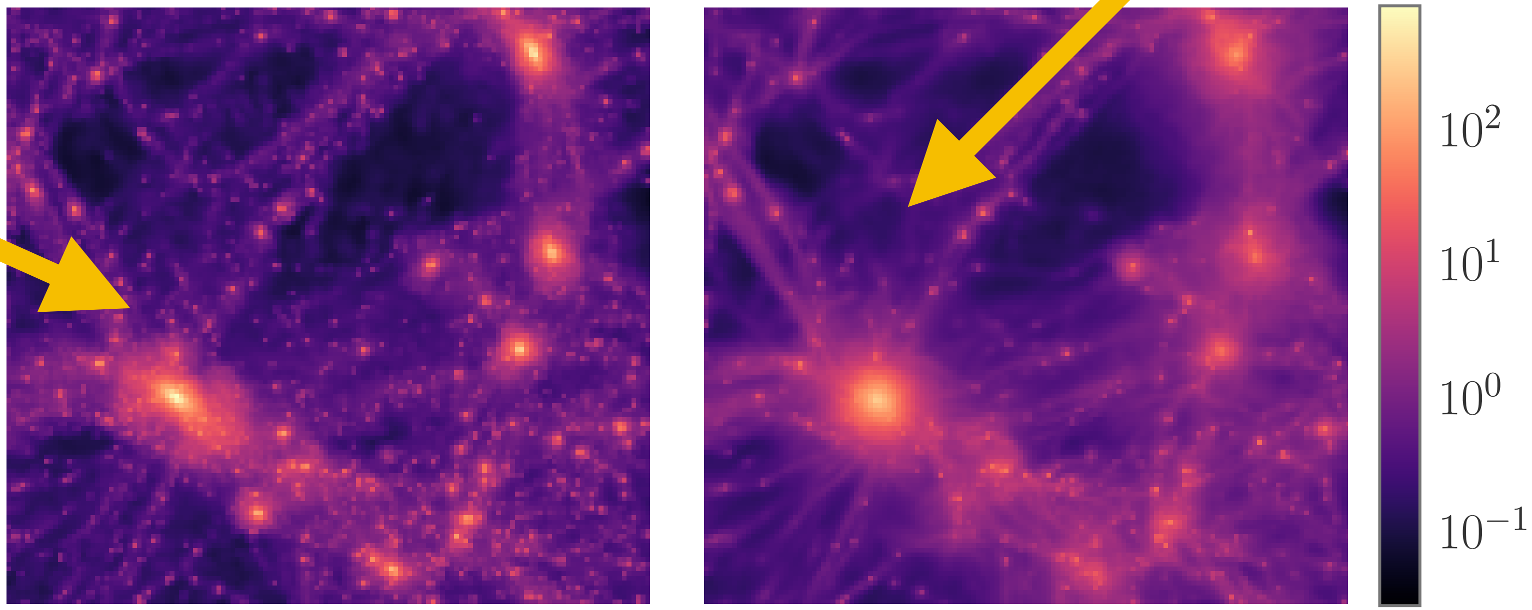
Jeffrey, N., et al. (2025). Dark energy survey year 3 results: likelihood-free, simulation-based wCDM inference with neural compression of weak-lensing map statistics. *Monthly Notices of the Royal Astronomical Society*, *536*(2), 1303–1322.

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

Hydrodynamic simulations

Gravity-only N-body simulations



Low-fidelity High-fidelity

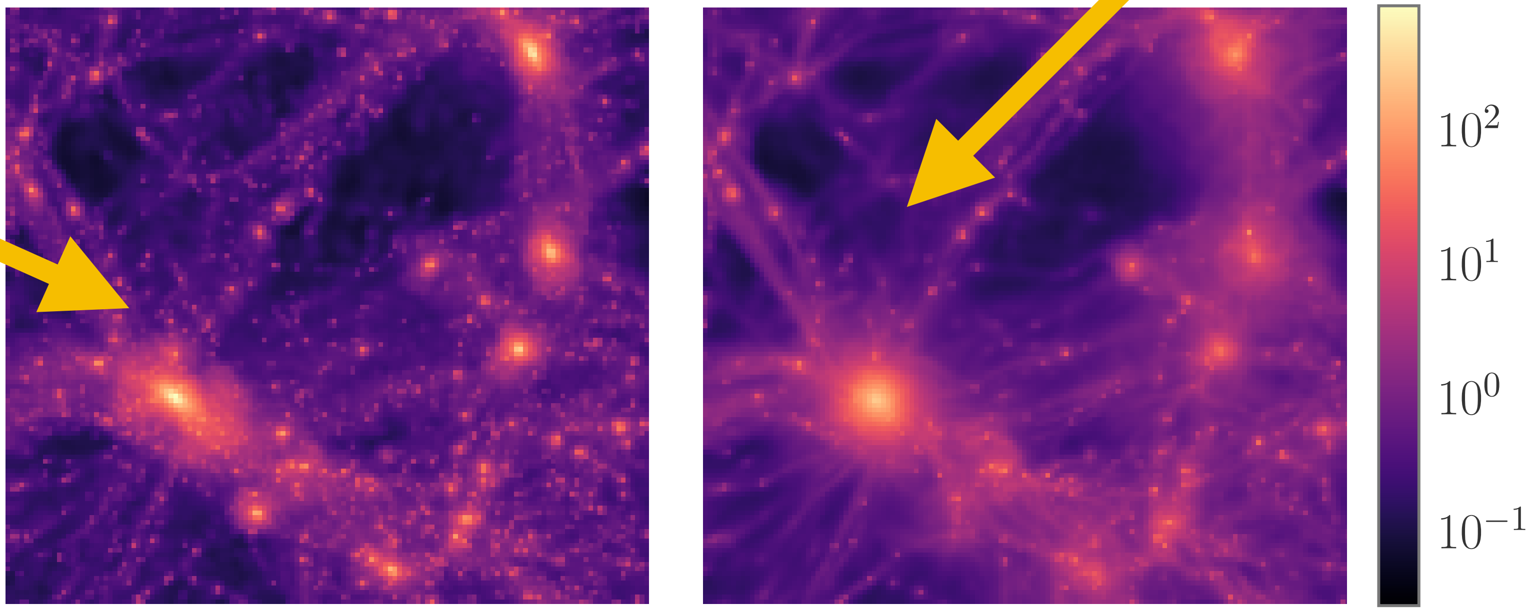
Jeffrey, N., et al. (2025). Dark energy survey year 3 results: likelihood-free, simulation-based wCDM inference with neural compression of weak-lensing map statistics. *Monthly Notices of the Royal Astronomical Society*, *536*(2), 1303–1322.

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:

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

Existing work on multi-fidelity in SBI

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

Neural SBI

- Can think of this as a two-step procedure:

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. a*rXiv:2508.12939*.

Neural SBI

- Can think of this as a two-step procedure:

  Step 1: Simulate synthetic data, then use this to fit a **neural surrogate** for the likelihood/posterior (often a normalising flow).

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. a*rXiv:2508.12939*.

Neural SBI

- Can think of this as a two-step procedure:

  Step 1: Simulate synthetic data, then use this to fit a **neural surrogate** for the likelihood/posterior (often a normalising flow).

  Step 2: Use our surrogate to do inference with the observed data.

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. a*rXiv:2508.12939*.

Intro to normalising flows: transformations and densities

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

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

Intro to normalising flows: 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:

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

Intro to normalising flows: 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:

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

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

Intro to normalising flows: 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:

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

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

Intro to normalising flows: 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:

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

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

Intro to normalising flows: 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

- 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

- 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

- 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:

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

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

- 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:

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

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

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

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

- We end up with a normalising flow:

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

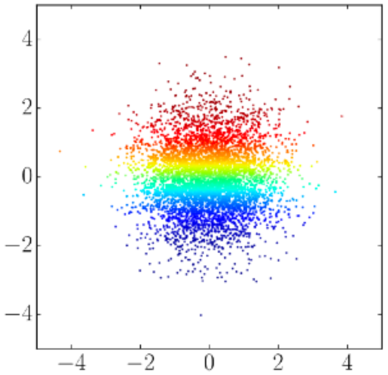
$\longrightarrow$ $T^1_{\phi,\theta}, \ldots, T^K_{\phi,\theta}$ are selected to make $q_\phi(x \mid \theta)$ **tractable**

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

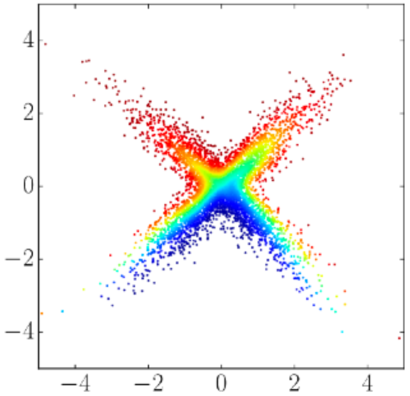
Kobyzev, I., Prince, S. J. D., & Brubaker, M. A. (2021). Normalizing flows: An introduction and review of current methods. *IEEE TPAMI, 43*(11), 3964–3979.

Illustration of normalising flows

$p_v(v)$



$q_\phi(x)$



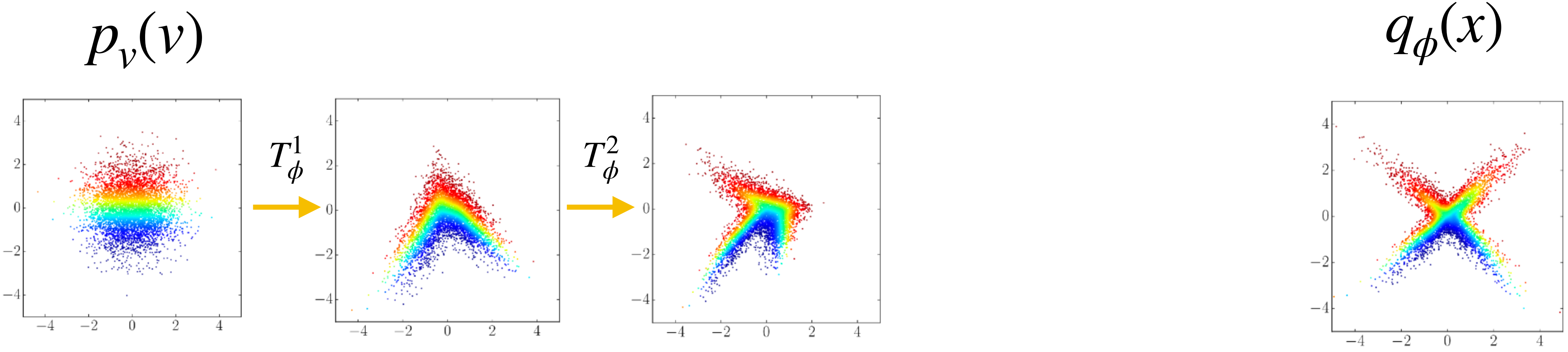
Plots borrowed from: Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., & Lakshminarayanan, B. (2021). Normalizing flows for probabilistic modeling and inference. *JMLR*, *22*, 1–64.

Illustration of normalising flows

$p_v(v)$ $q_\phi(x)$

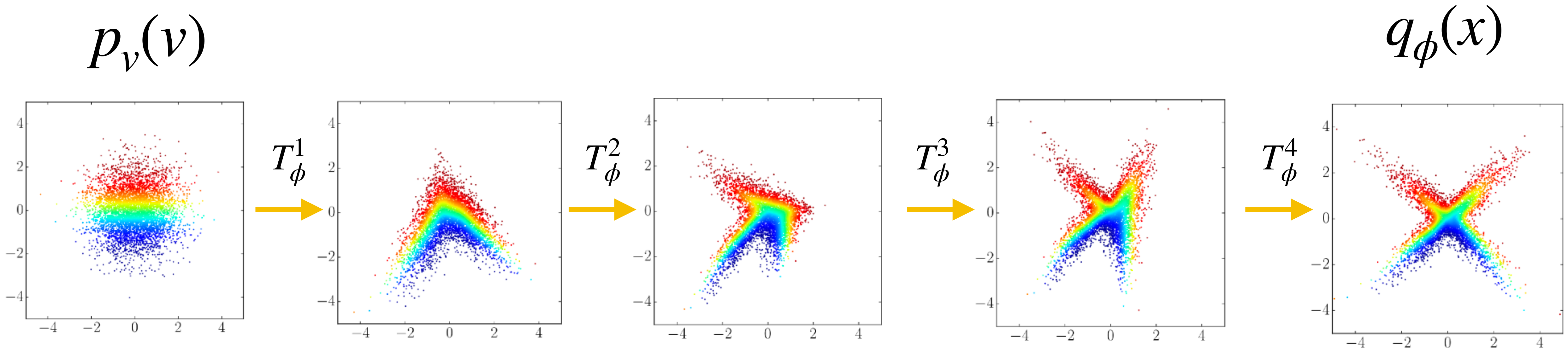
Plots borrowed from: Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., & Lakshminarayanan, B. (2021). Normalizing flows for probabilistic modeling and inference. *JMLR*, *22*, 1–64.

Illustration of normalising flows



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.

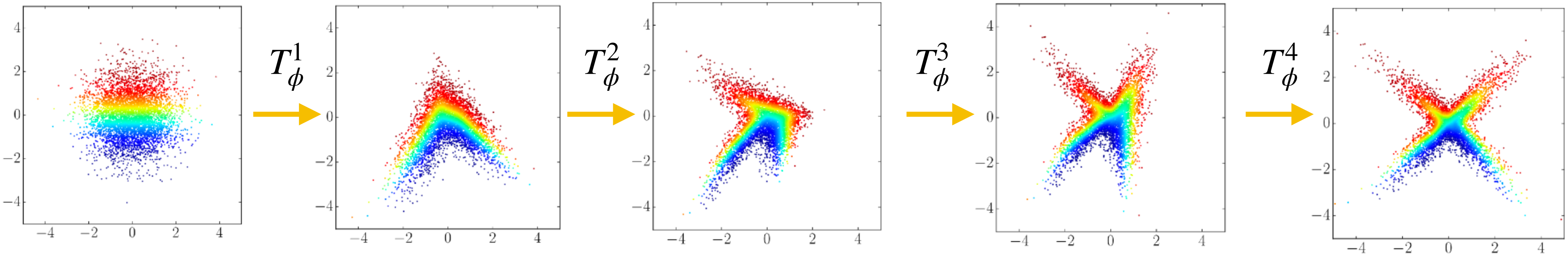
Illustration of normalising flows



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.

Illustration of normalising flows

$p_v(v)$ $q_\phi(x)$



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

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

Neural likelihood estimation (NLE)

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

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

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

$$\pi_{\text{NLE}}(\theta\,|\,y_1, \ldots, y_m) \propto \prod_{i=1}^{m} q_{\hat{\phi}_n}(y_i\,|\,\theta)\pi(\theta)$$

Neural likelihood estimation (NLE)

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

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

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

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

$$\pi_{\text{NLE}}(\theta\,|\,y_1, \ldots, y_m) \propto \prod_{i=1}^{m} q_{\hat{\phi}_n}(y_i\,|\,\theta)\pi(\theta)$$

Neural likelihood estimation (NLE)

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

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

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

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

$$\pi_{\text{NLE}}(\theta \mid y_1, \ldots, y_m) \propto \prod_{i=1}^{m} q_{\hat{\phi}_n}(y_i \mid \theta)\pi(\theta)$$

- Can do similarly and approximate a posterior….. Neural posterior estimation (NPE).

A better step 1?

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

Can we do this better/cheaper?!

A better step 1?

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

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 $f_0, f_1, \ldots, f_L = f$ 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.

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

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

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

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:

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

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:

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

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

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

Seed-matched!

Our 'data' is therefore:

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

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

Seed-matched!

Our ‘data’ is therefore:

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

Our objective for step 1 is:

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

$$\text{Var}\left[f_\phi^l(u_i^l, \theta_i^l) - f_\phi^{l-1}(u_i^l, \theta_i^l) \right] = \text{Var}[f_\phi^l(u_i^l, \theta_i^l)] + \text{Var}[f_\phi^{l-1}(u_i^l, \theta_i^l)] - 2\text{Cov}\left[f_\phi^l(u_i^l, \theta_i^l), f_\phi^{l-1}(u_i^l, \theta_i^l) \right]$$

Multilevel neural SBI

Seed-matched!

Our 'data' is therefore:

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

Our objective for step 1 is:

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

Small!!

Large!!

$$\text{Var}\left[f_\phi^l(u_i^l, \theta_i^l) - f_\phi^{l-1}(u_i^l, \theta_i^l)\right] = \text{Var}[f_\phi^l(u_i^l, \theta_i^l)] + \text{Var}[f_\phi^{l-1}(u_i^l, \theta_i^l)] - 2\text{Cov}\left[f_\phi^l(u_i^l, \theta_i^l), f_\phi^{l-1}(u_i^l, \theta_i^l)\right]$$

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:

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

$$\text{Var}\left[f_\phi^l(u_i^l, \theta_i^l) - f_\phi^{l-1}(u_i^l, \theta_i^l) \right] = \text{Var}[f_\phi^l(u_i^l, \theta_i^l)] + \text{Var}[f_\phi^{l-1}(u_i^l, \theta_i^l)] - 2\text{Cov}\left[f_\phi^l(u_i^l, \theta_i^l), f_\phi^{l-1}(u_i^l, \theta_i^l) \right]$$

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

Challenges with training

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

Challenges with training

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

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

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

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

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

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

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, which stabilises training.

Bound on the variance

Under some mild assumptions, we get:

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

Bound on the variance

Under some mild assumptions, we get:

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

Large!

Small!

Bound on the variance

Under some mild assumptions, we get:

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

Large!

Complexity of low-fidelity generator - large!

Small!

Complexity of other integrands - small!

Bound on the variance

Under some mild assumptions, we get:

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

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:

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

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:

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

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

We can find the optimal number of simulations per level given a maximum computational budget of C_{budget}:

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

Simulations per level

We can find the optimal number of simulations per level given a maximum computational budget of C_{budget}:

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

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

Simulations per level

We can find the optimal number of simulations per level given a maximum computational budget of C_{budget}:

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

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

Simulations per level

We can find the optimal number of simulations per level given a maximum computational budget of C_{budget}:

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

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

Simulations per level

We can find the optimal number of simulations per level given a maximum computational budget of C_{budget}:

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

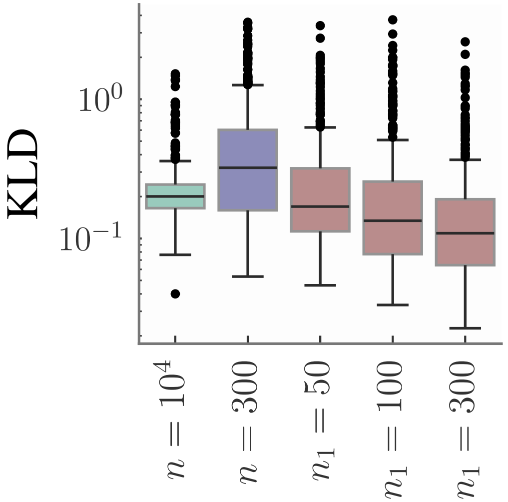
Note that these expressions contain a lot of quantities **we may not know a-priori,** but it is still indicative and helpful for selecting which simulations to run in practice.

G-and-k distribution

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

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

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



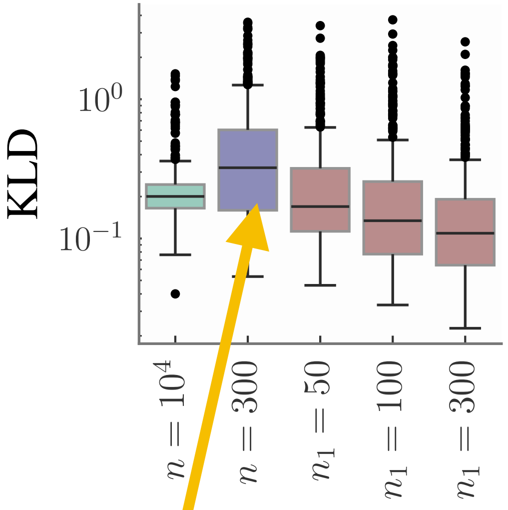
Prangle, D. (2020). gk: An R Package for the g-and-k and generalised g-and-h distributions. The R Journal, 12(1):7.

G-and-k distribution

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

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

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



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.

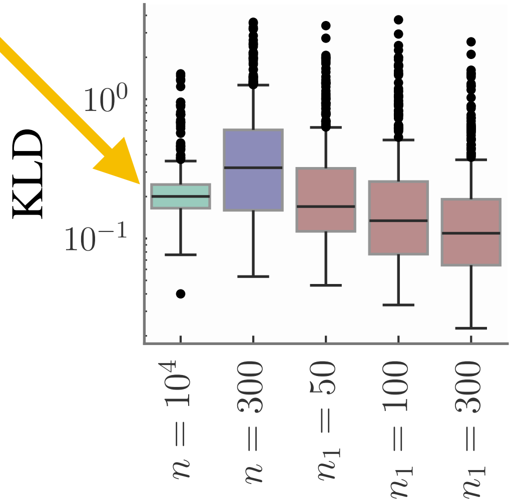
G-and-k distribution

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

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

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

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.

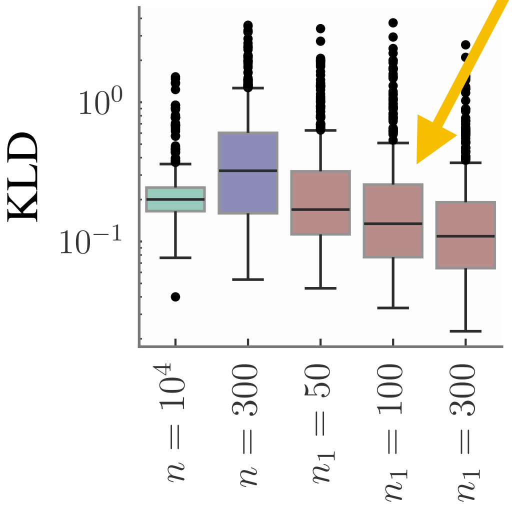
G-and-k distribution

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

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

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

ML-NLE: both many simulations and high quality!



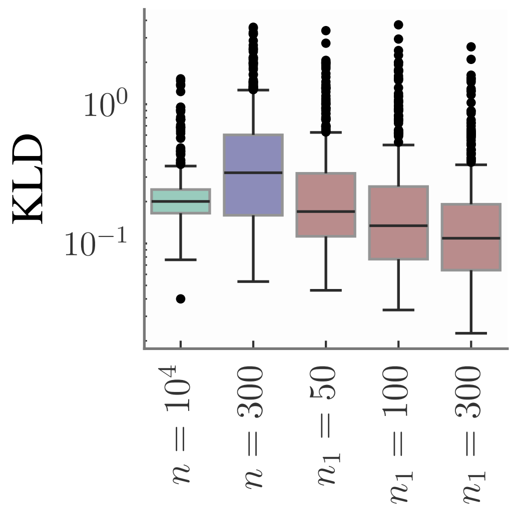
Prangle, D. (2020). gk: An R Package for the g-and-k and generalised g-and-h distributions. The R Journal, 12(1):7.

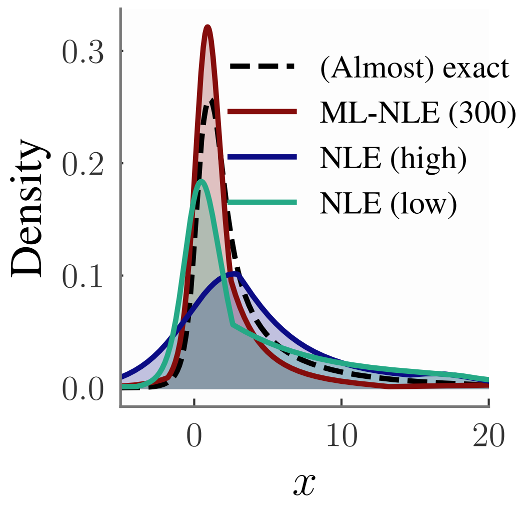
G-and-k distribution

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

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

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





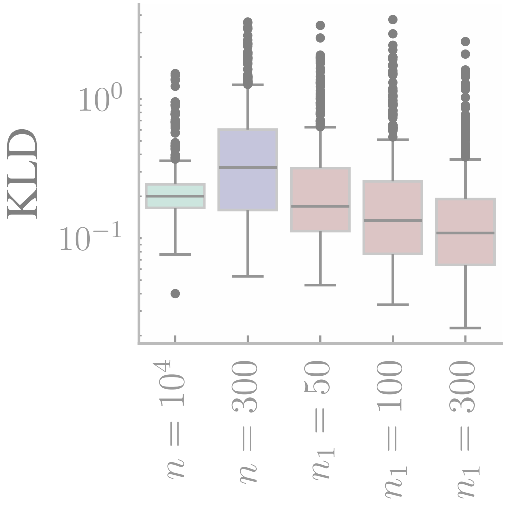
Prangle, D. (2020). gk: An R Package for the g-and-k and generalised g-and-h distributions. The R Journal, 12(1):7.

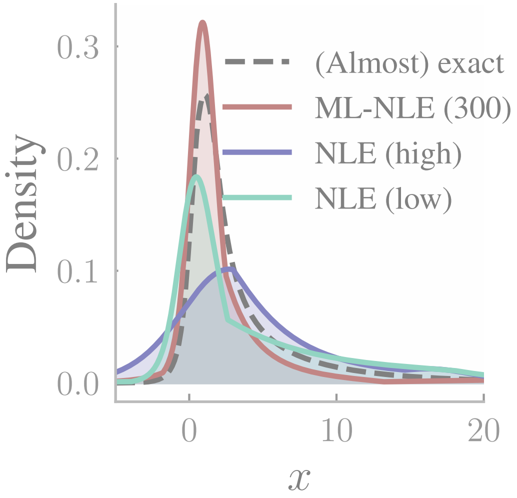
G-and-k distribution

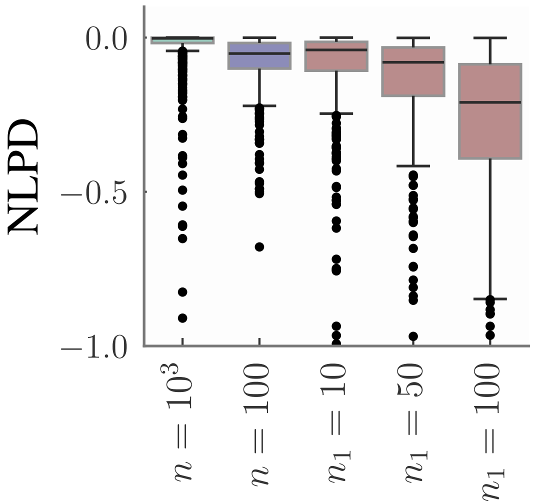
$$G^{l}_{\theta}(u) = \theta_1 + \theta_2 \left(1 + 0.8\left(\frac{1-\exp(-\theta_3 z_l(u))}{1+\exp(-\theta_3 z_l(u))}\right)\right)\left(1+z_l(u)^2\right)^{\log(\theta_4)} z_l(u),$$

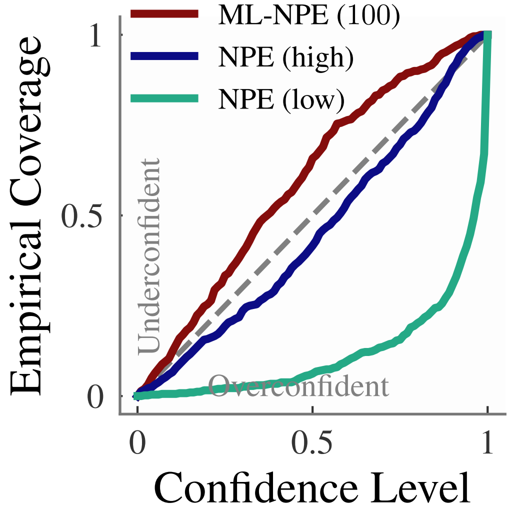
$$z_1(u) = \Phi^{-1}(u) = \sqrt{2}\,\mathrm{erf}^{-1}(2u-1), \qquad u \sim \mathrm{Unif}([0,1]),$$

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









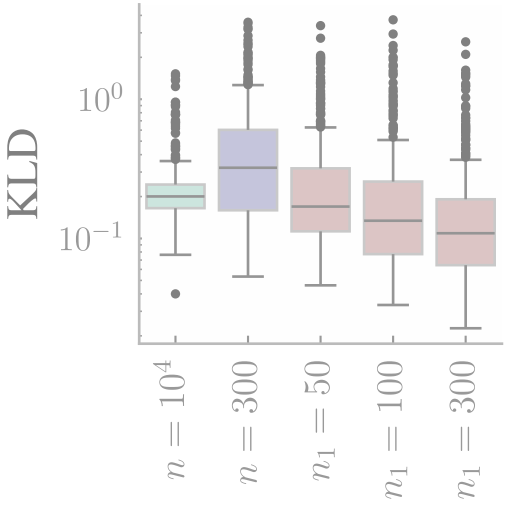
Prangle, D. (2020). gk: An R Package for the g-and-k and generalised g-and-h distributions. The R Journal, 12(1):7.

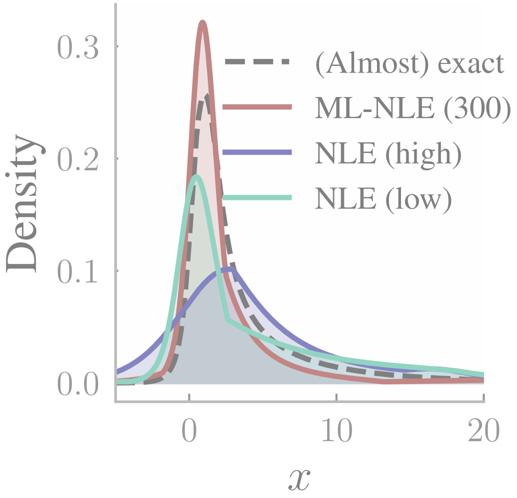
G-and-k distribution

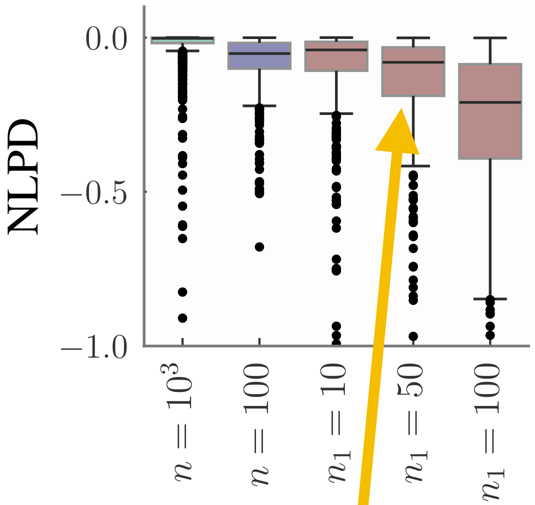
$$G_\theta^l(u) = \theta_1 + \theta_2 \left(1 + 0.8\left(\frac{1-\exp(-\theta_3 z_l(u))}{1+\exp(-\theta_3 z_l(u))}\right)\right)\left(1+z_l(u)^2\right)^{\log(\theta_4)} z_l(u),$$

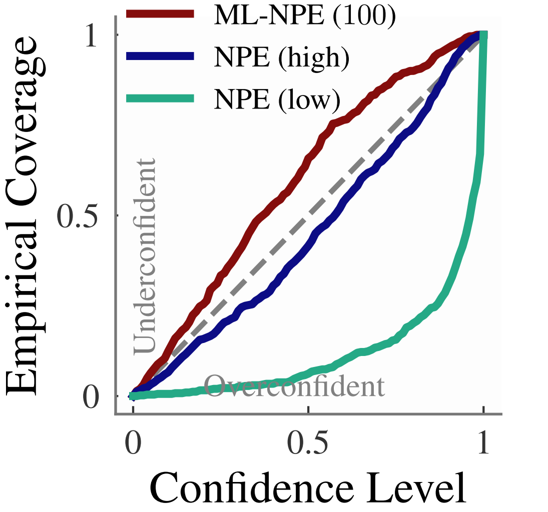
$$z_1(u) = \Phi^{-1}(u) = \sqrt{2}\,\mathrm{erf}^{-1}(2u-1), \qquad u \sim \mathrm{Unif}([0,1]),$$

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









ML-NPE: Similar conclusion!

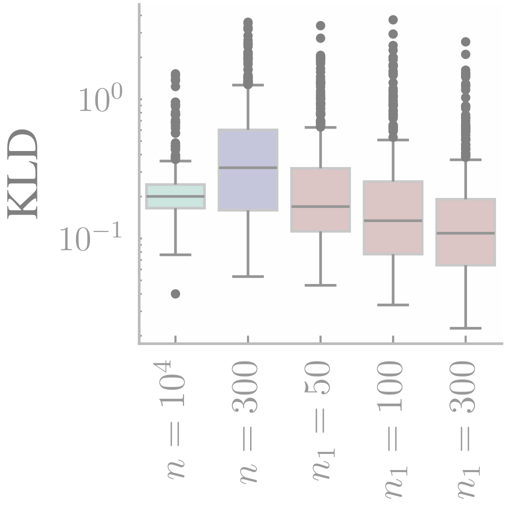
Prangle, D. (2020). gk: An R Package for the g-and-k and generalised g-and-h distributions. The R Journal, 12(1):7.

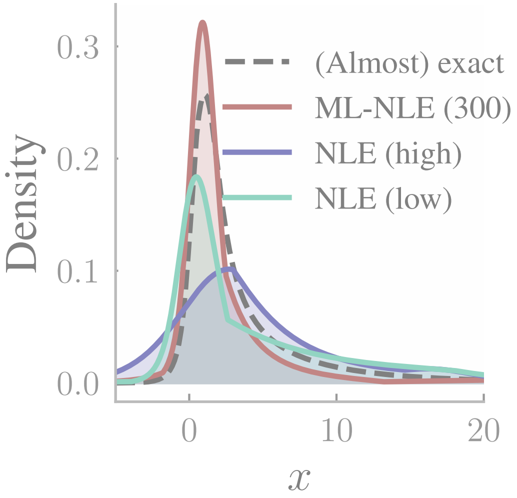
G-and-k distribution

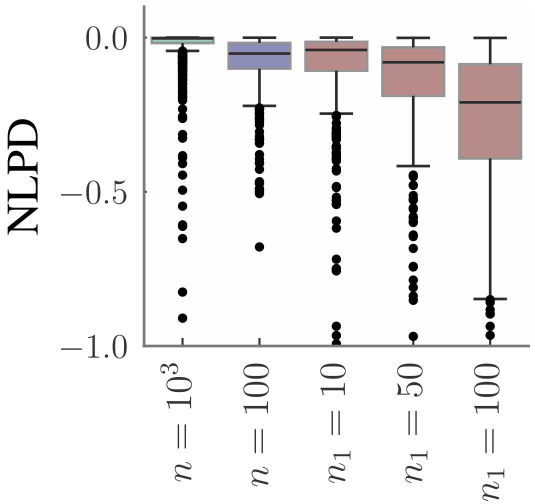
$$G^l_\theta(u) = \theta_1 + \theta_2 \left(1 + 0.8\left(\frac{1-\exp(-\theta_3 z_l(u))}{1+\exp(-\theta_3 z_l(u))}\right)\right)\left(1+z_l(u)^2\right)^{\log(\theta_4)} z_l(u),$$

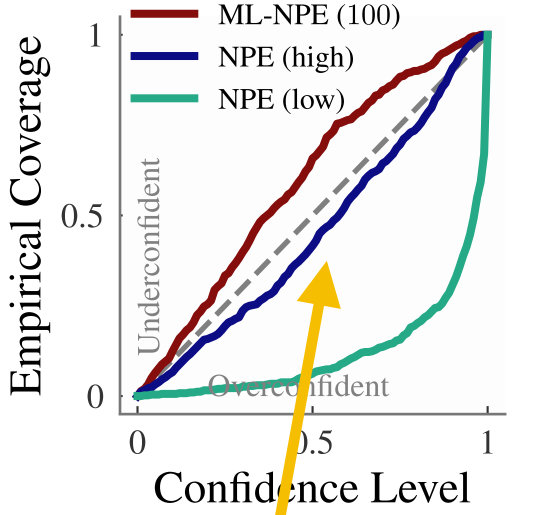
$$z_1(u) = \Phi^{-1}(u) = \sqrt{2}\,\mathrm{erf}^{-1}(2u-1), \qquad u \sim \mathrm{Unif}([0,1]),$$

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





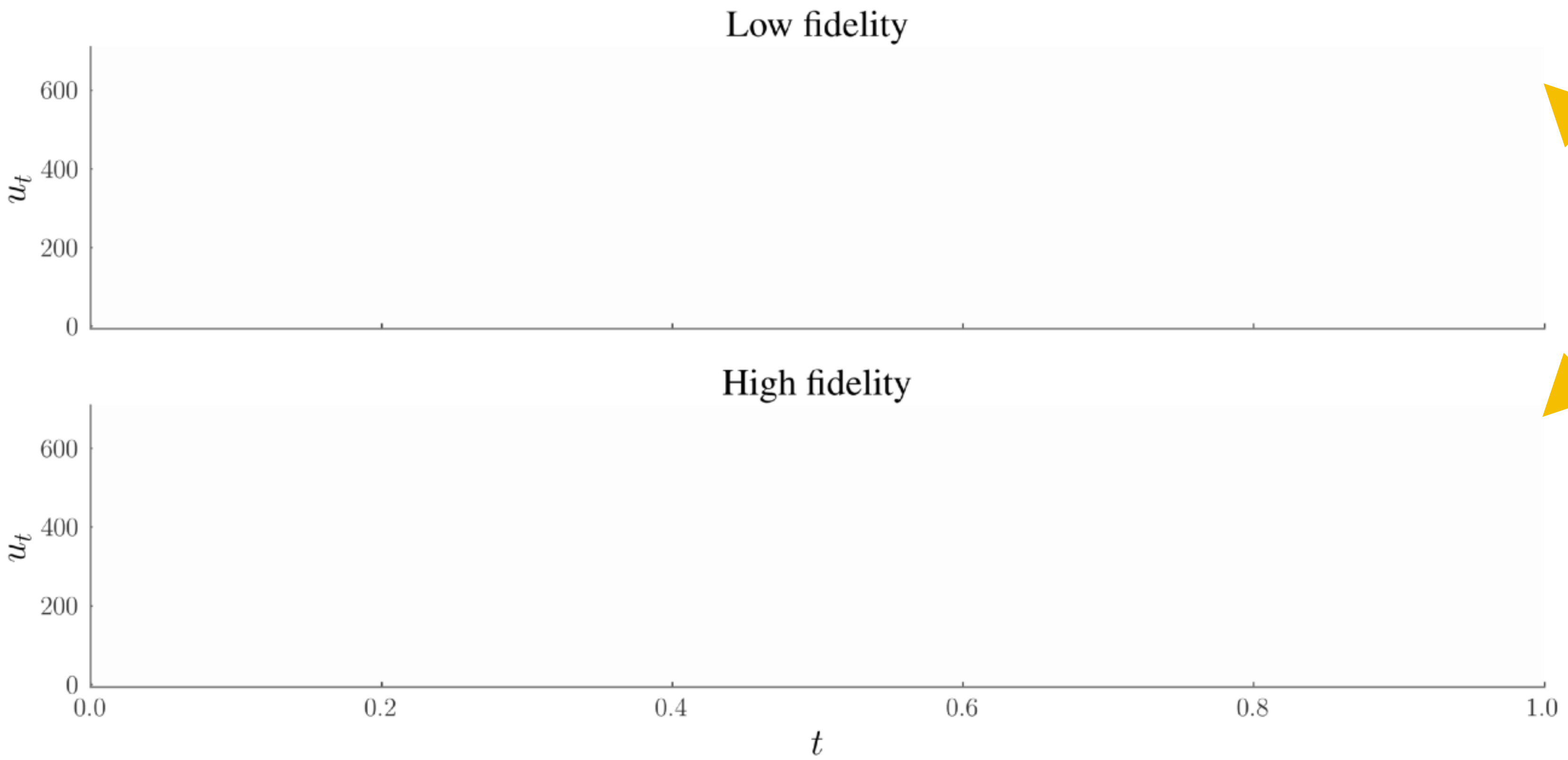




Coverage slightly cautious

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

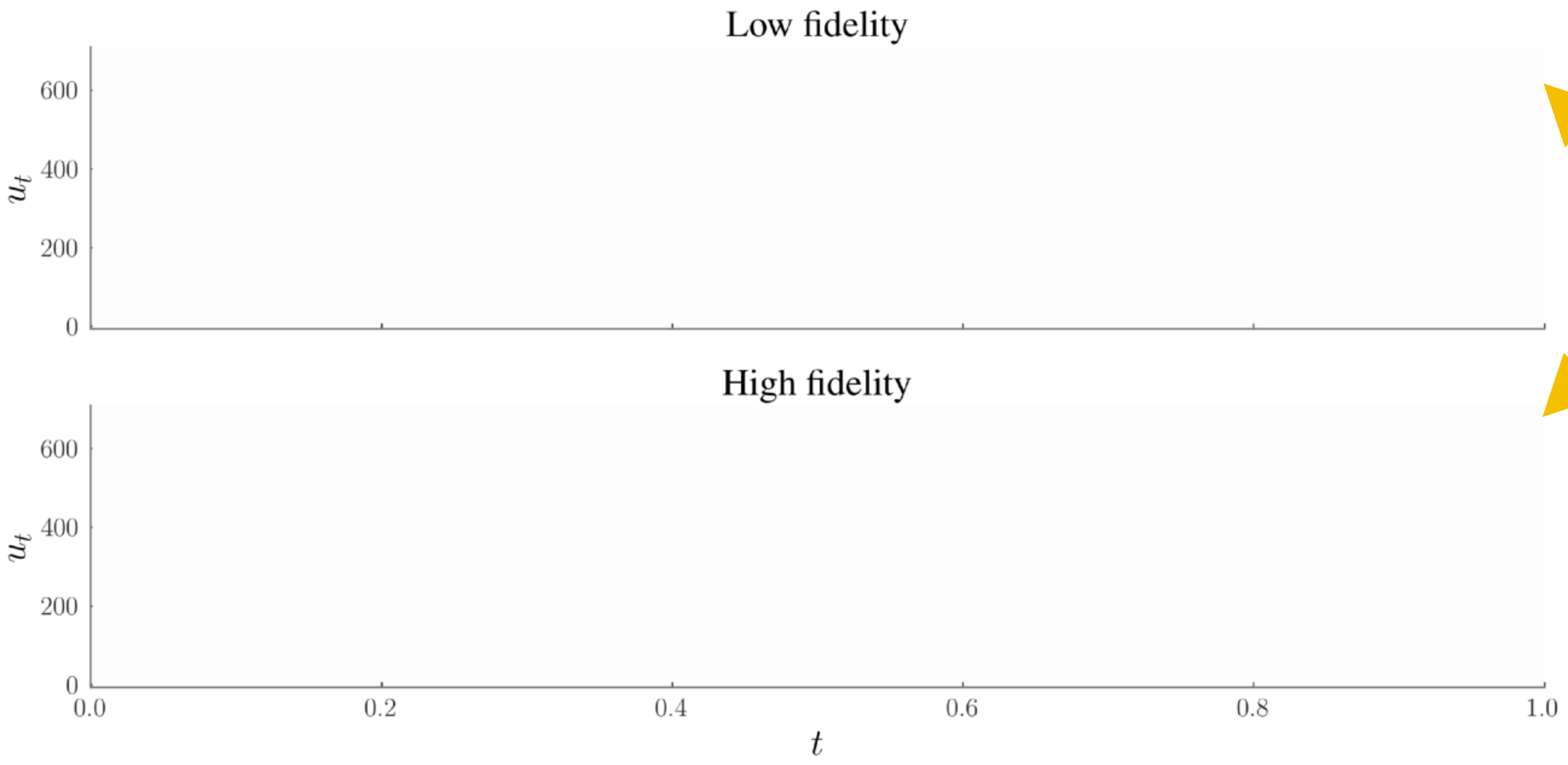
Toggle-switch models for genes (d=1, p=7)



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).

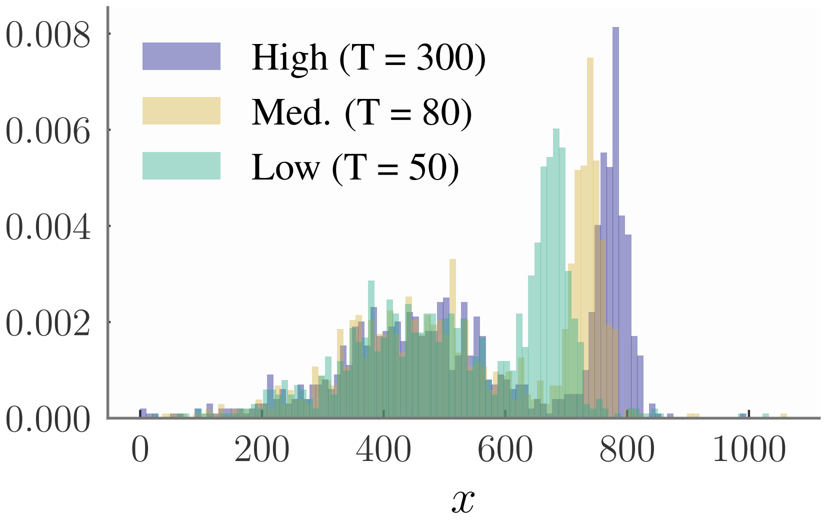
Toggle-switch models for genes (d=1, p=7)



You only observe the last time step

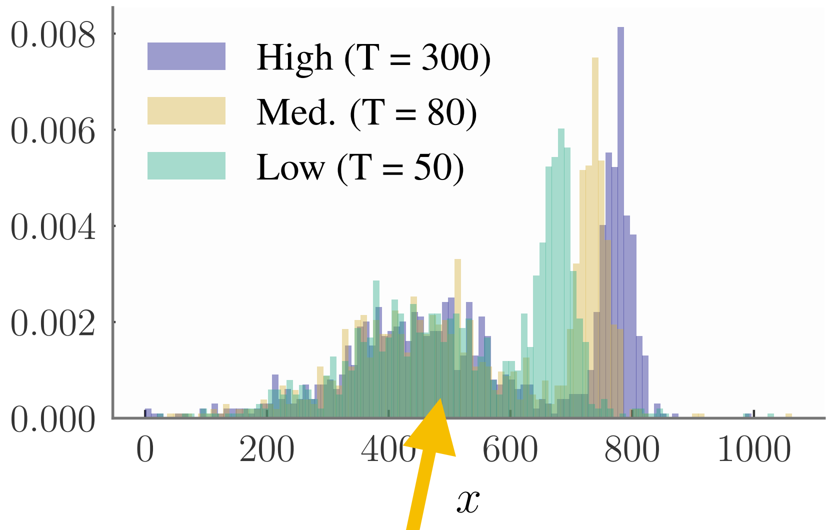
Bonassi, F. V., You, L., & West, M. (2011). Bayesian learning from marginal data in bionetwork models. *Statistical Applications in Genetics and Molecular Biology*, *10*(1).

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).

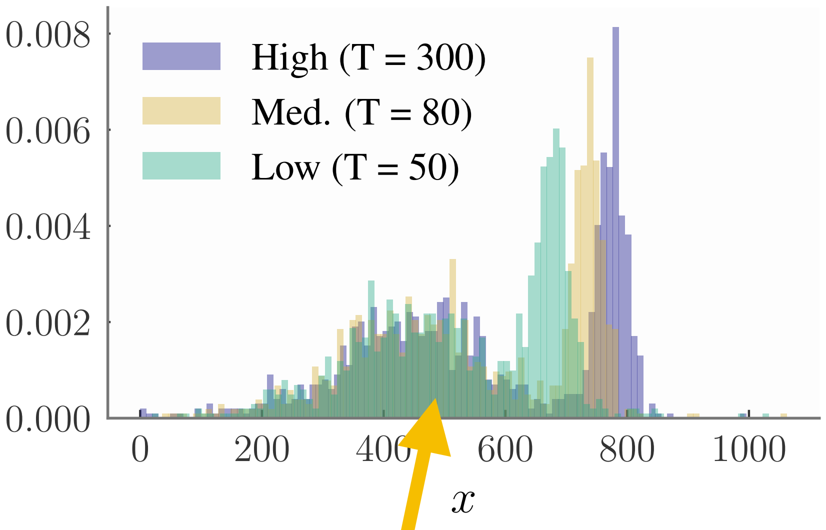
Toggle-switch models for genes (d=1, p=7)

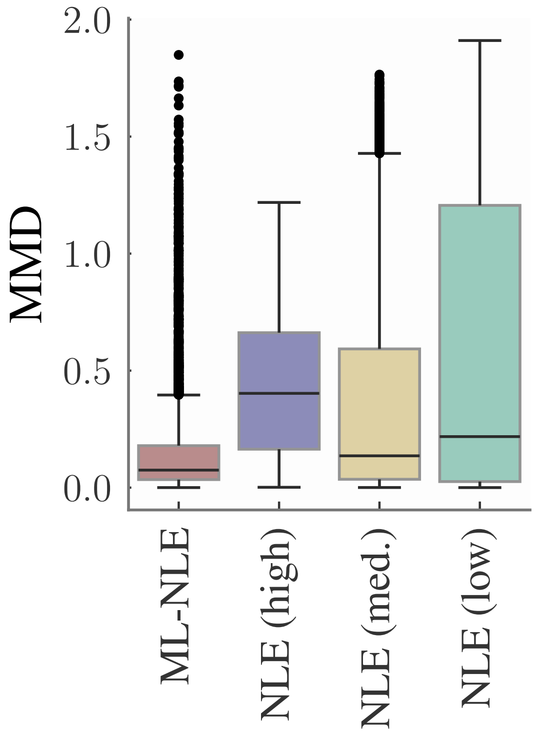


Observations bi-modal, with second mode only well approximated for high-fidelity levels

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)



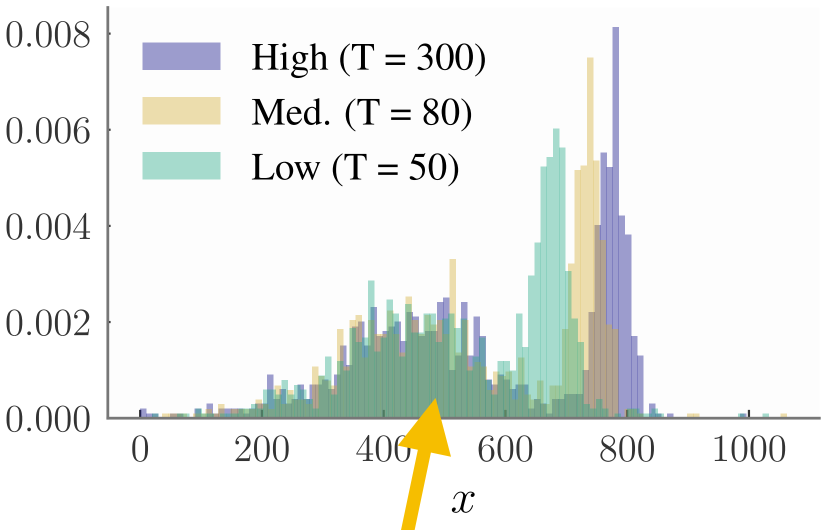


$n_0 = 10000$
$n_1 = 500$
$n_2 = 300$

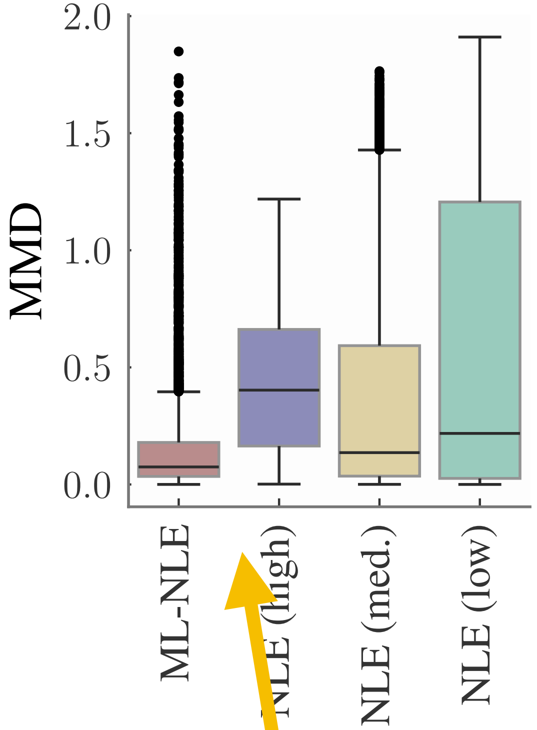
Observations bi-modal, with second mode only well approximated for high-fidelity levels

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)



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



$n_0 = 10000$

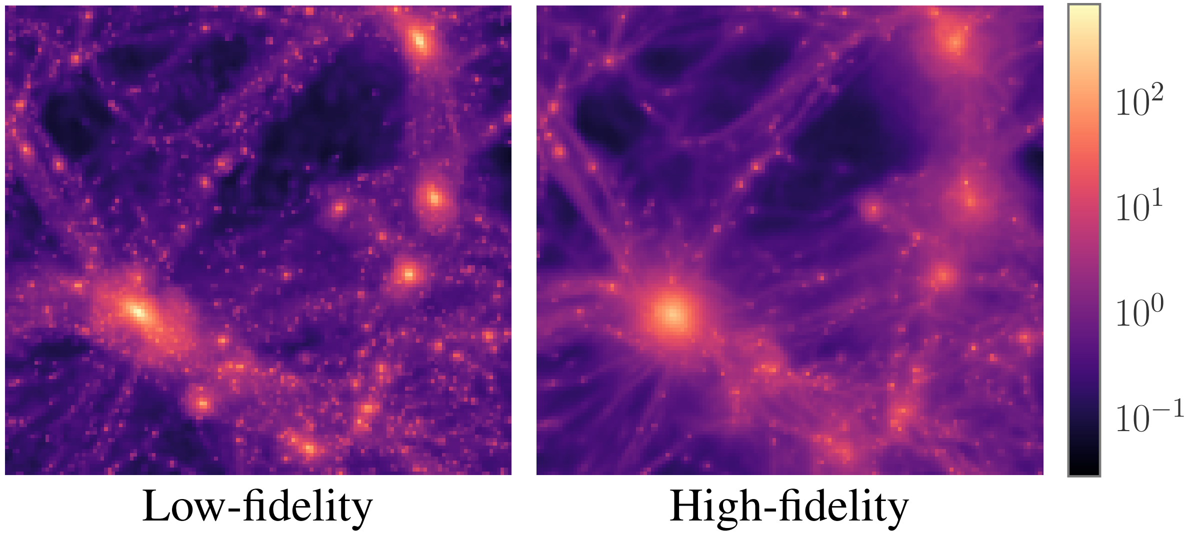
$n_1 = 500$

$n_2 = 300$

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).

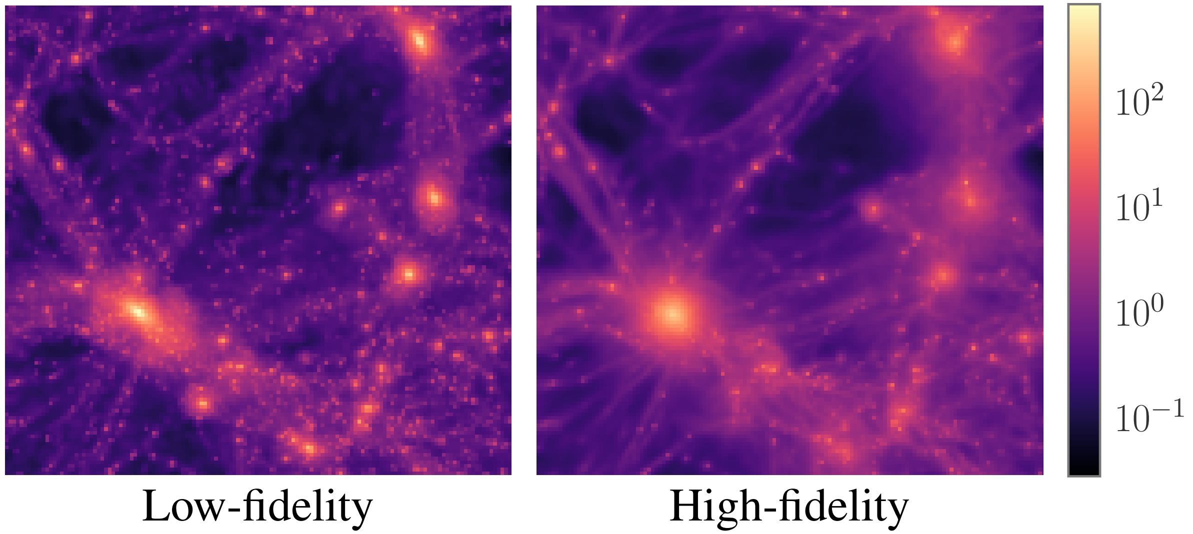
Back to cosmology…. (d=39, p=1)

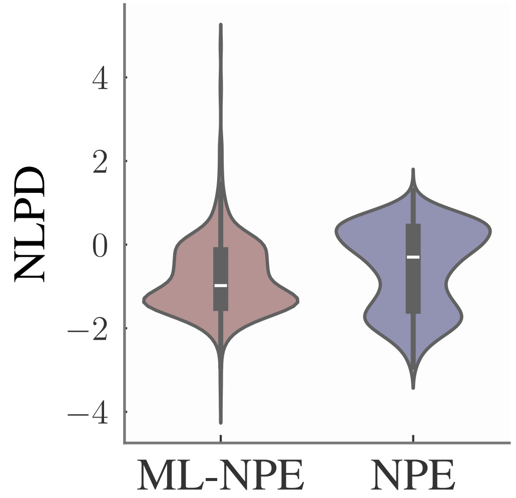


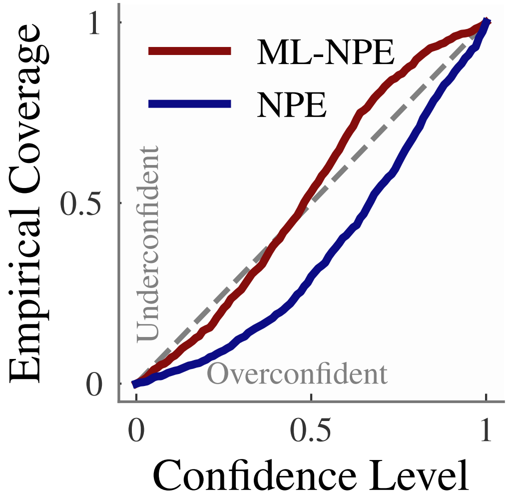
NPE: $n = 20$ (all high fidelity!)

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

Back to cosmology…. (d=39, p=1)



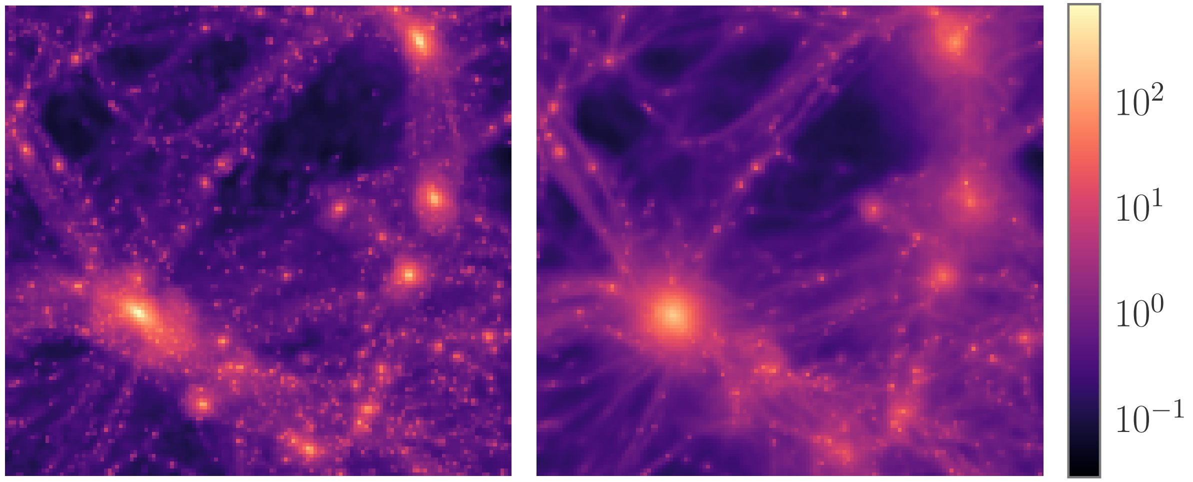




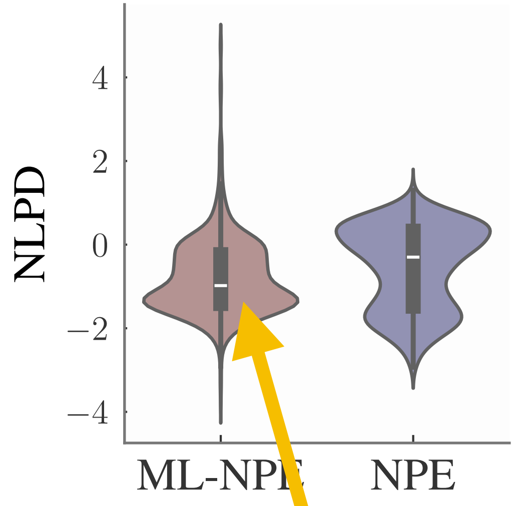
NPE: $n = 20$ (all high fidelity!)

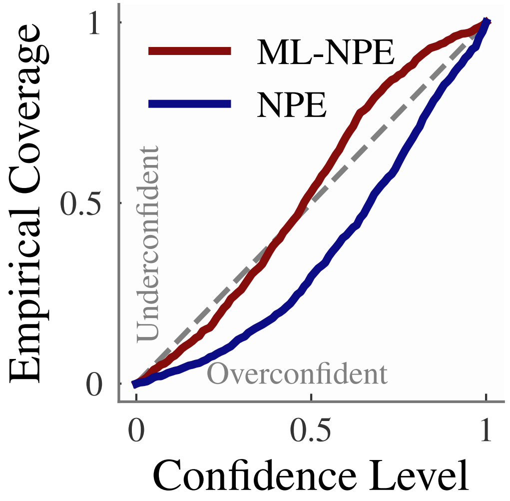
ML-NPE: $n_0 = 20, \quad n_1 = 980$

Back to cosmology…. (d=39, p=1)



Low-fidelity High-fidelity



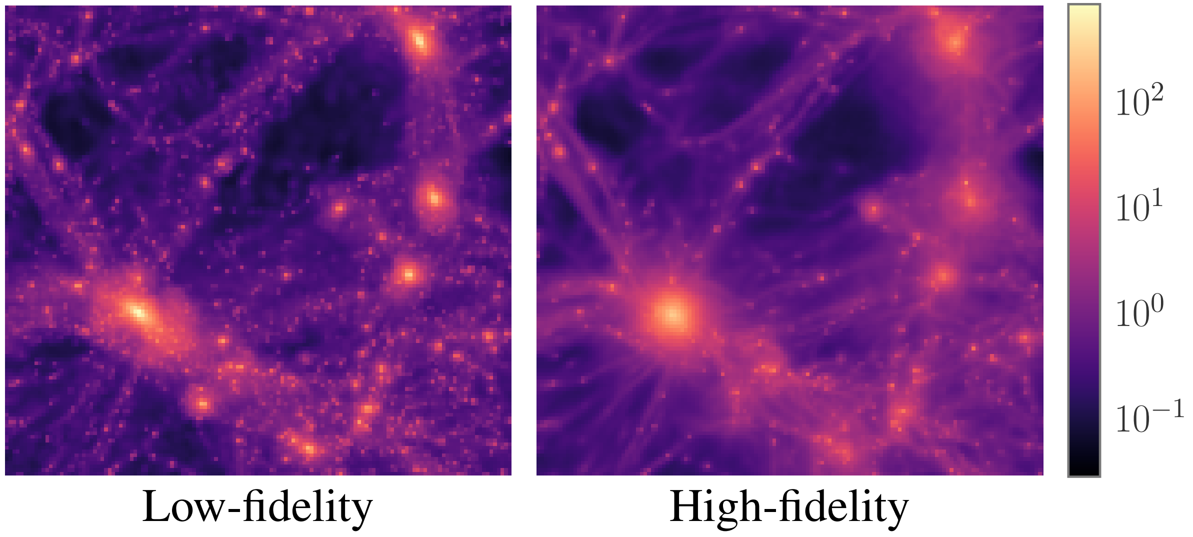


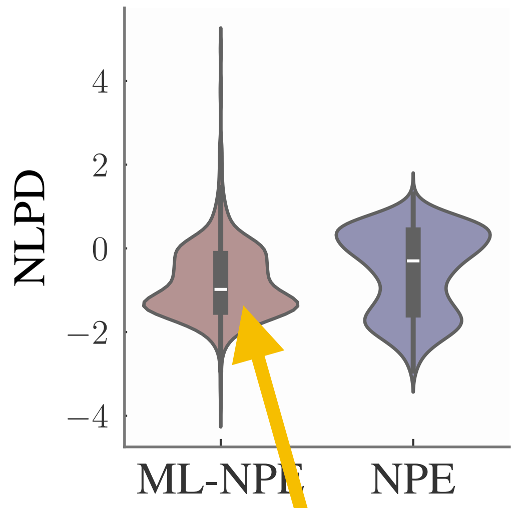
Improve fit of the surrogate posterior!

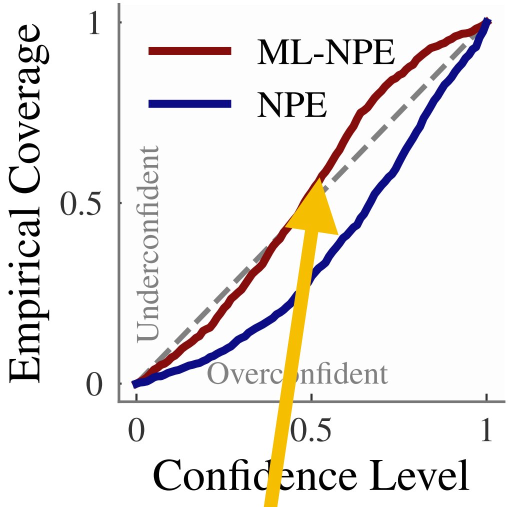
NPE: $n = 20$ (all high fidelity!)

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

Back to cosmology…. (d=39, p=1)







Improve fit of the surrogate posterior!

Improved calibration!

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

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

Conclusion

Conclusion

- We use multilevel Monte Carlo in neural SBI, allowing for a rigorous way of combining low- and high-fidelity simulations!

Conclusion

- We use multilevel Monte Carlo in neural SBI, allowing for a rigorous way of combining low- and high-fidelity simulations!

- Lots of interest from practitioners; two physics papers on this topic:

A. A. Saoulis, D. Piras, N. Jeffrey, A. Spurio-Mancini, A. M. G. Ferreira, and B. Joachimi (2025+). Transfer learning for multifidelity simulation-based inference in cosmology. arXiv:2505.21215.

L. Thiele, A. E. Bayer, and N. Takeishi. Simulation-efficient cosmological inference with multi-fidelity SBI (2025+). arXiv:2507.00514.

Conclusion

- We use multilevel Monte Carlo in neural SBI, allowing for a rigorous way of combining low- and high-fidelity simulations!

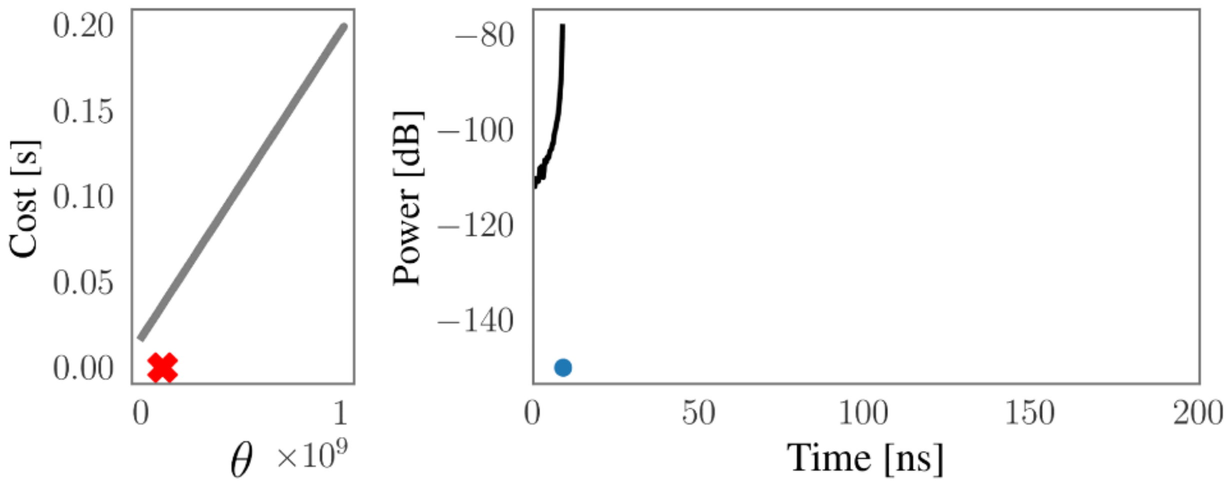
- Lots of of interest from practitioners; two physics papers on this topic:

  A. A. Saoulis, D. Piras, N. Jeffrey, A. Spurio-Mancini, A. M. G. Ferreira, and B. Joachimi (2025+). Transfer learning for multifidelity simulation-based inference in cosmology. arXiv:2505.21215.

  L. Thiele, A. E. Bayer, and N. Takeishi. Simulation-efficient cosmological inference with multi-fidelity SBI (2025+). arXiv:2507.00514.

- Slides from a recent course on SBI at Greek stochastic 2025: https://fxbriol.github.io/pdfs/slides-SBI-course.pdf

Related recent work

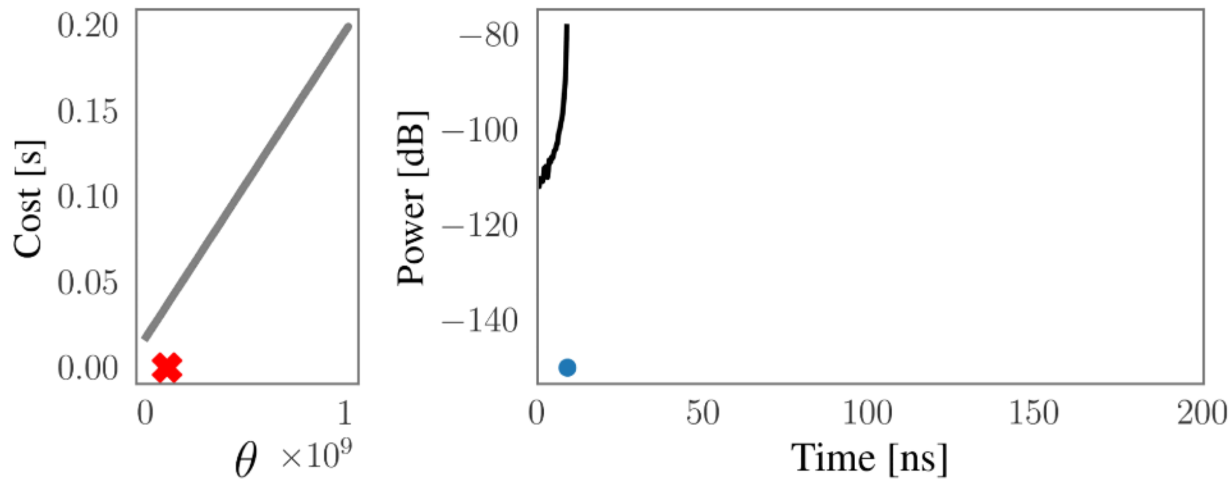


Sometimes the cost of simulation (i.e. of G_θ) depends on θ!

Bharti, A., Huang, D., Kaski, S. & Briol, F-X. (2025). Cost-aware simulation-based inference. Proceedings of The 28th International Conference on Artificial Intelligence and Statistics, PMLR 258:28-36.

OWABI recording (Ayush Bharti; 30th April 2025): https://www.youtube.com/watch?v=9tnp9fbpydY

Related recent work



Sometimes the cost of simulation (i.e. of G_θ) depends on θ!

Bharti, A., Huang, D., Kaski, S. & Briol, F-X. (2025). Cost-aware simulation-based inference. Proceedings of The 28th International Conference on Artificial Intelligence and Statistics, PMLR 258:28-36.

OWABI recording (Ayush Bharti; 30th April 2025): https://www.youtube.com/watch?v=9tnp9fbpydY

Any Questions?

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

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