Note: A more detailed version of the same slides, more suitable for **reading** starts on slide 59.

Overcoming limitations to ALP parameter inference with Neural Ratio Estimation

Gert Kluge

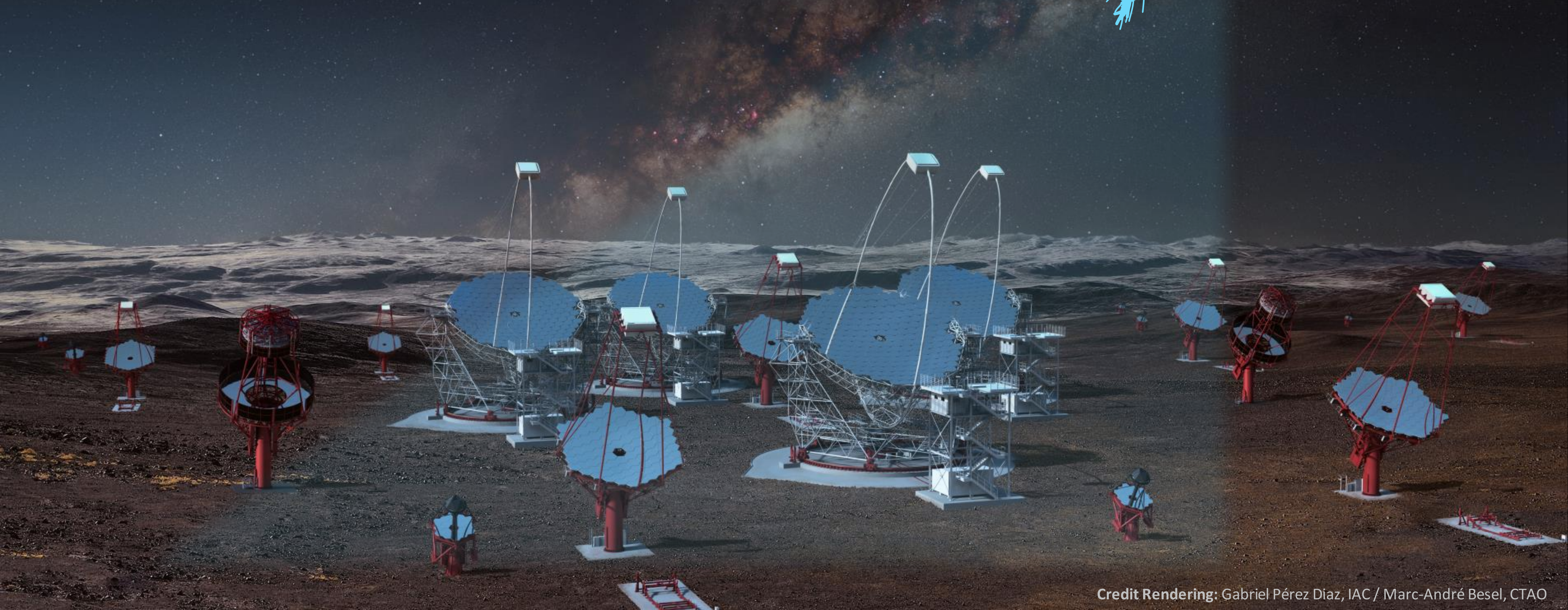
PhD candidate at the University of Oslo
contact: gertwk@uio.no

In collaboration with
Giacomo D'Amico, Julia Djuvsland, and Heidi Sandaker

Overcoming limitations to ALP parameter inference with Neural Ratio Estimation

the Cherenkov Telescope Array (CTA)

Set to become the world's leading very high-energy γ-ray telescope

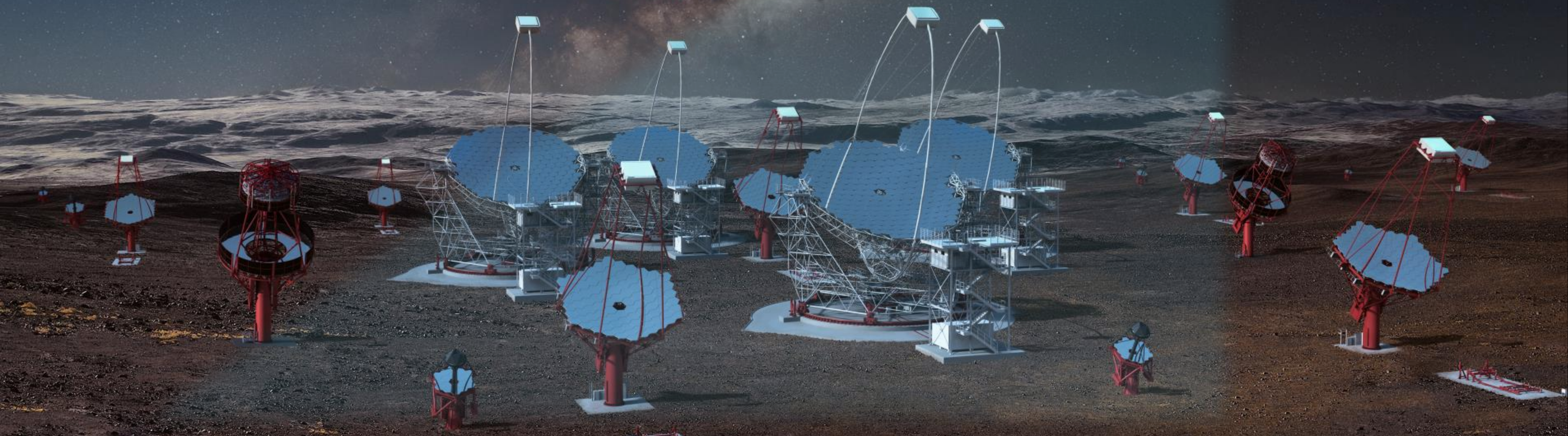


Credit Rendering: Gabriel Pérez Diaz, IAC / Marc-André Besel, CTAO

the Cherenkov Telescope Array (CTA)

Set to become the world's leading very high-energy γ-ray telescope

- Order of magnitude more sensitive than current Cherenkov telescopes at TeV-level
- Energy resolution better than 10 percent at TeV-level



Credit Rendering: Gabriel Pérez Diaz, IAC / Marc-André Besel, CTAO

We want to search for ALPs with CTA

Credit NGC1275: A. Fabian (Institute of Astronomy, University of Cambridge). NASA, ESA, and the Hubble Heritage.

We want to search for ALPs with CTA

Axion-like particles

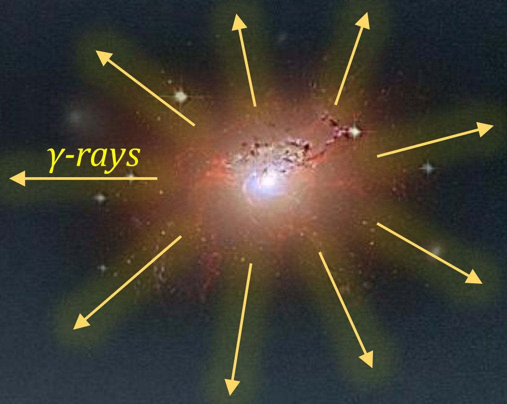
- Extension of the QCD-Axion [1]
- Popular candidates for dark matter and physics beyond the standard model
- **Can oscillate into photons in the presence of magnetic fields**

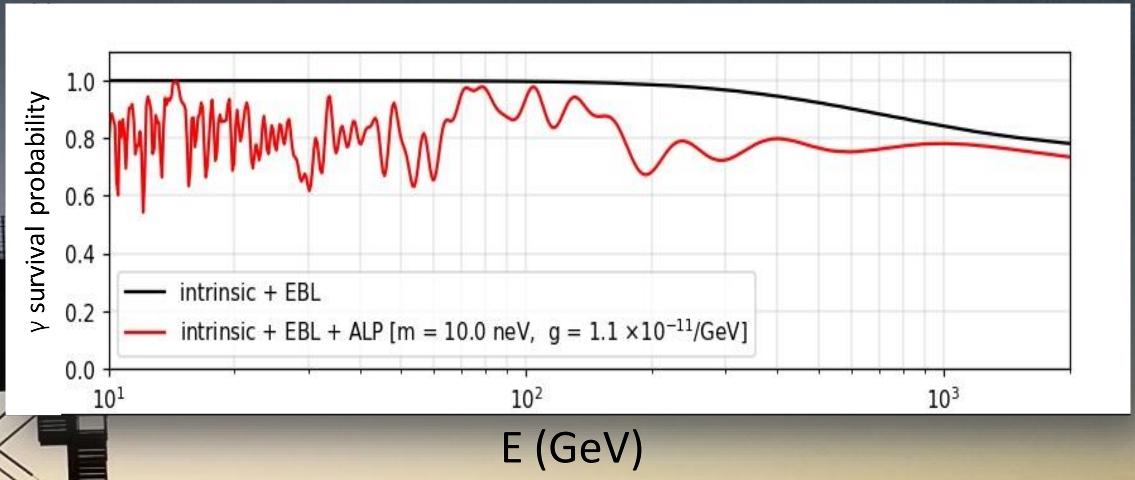
Credit NGC1275: A. Fabian (Institute of Astronomy, University of Cambridge). NASA, ESA, and the Hubble Heritage.

We want to search for ALPs with CTA

Credit NGC1275: A. Fabian (Institute of Astronomy, University of Cambridge). NASA, ESA, and the Hubble Heritage.

We want to search for ALPs with CTA





Credit NGC1275: A. Fabian (Institute of Astronomy, University of Cambridge). NASA, ESA, and the Hubble Heritage.

Problem:

How can we do reliable parameter inference
for this complicated model?

Parameters in our ALP-model*

- ALP mass, m
- ALP coupling to photons, g

Parameters of interest

- NGC1275 intrinsic spectrum amplitude
- NGC1275 intrinsic spectral index
- NGC1275 intrinsic cut-off energy

- Magnetic field strength of NGC1275
- Magnetic field configuration
- Extension of Perseus cluster
- 7 electron density-related parameters
- 3 turbulence-related parameters

$\sim$15 Nuisance parameters

* Our physical model and simulations are based on gammaALPs by Manuel Meyer [2]

Parameters in our ALP-model*

- ALP mass, m
- ALP coupling to photons, g

Parameters of interest

- NGC1275 intrinsic spectrum amplitude
- NGC1275 intrinsic spectral index
- NGC1275 intrinsic cut-off energy

- Magnetic field strength of NGC1275
- Magnetic field configuration
- Extension of Perseus cluster
- 7 electron density-related parameters
- 3 turbulence-related parameters

~15 Nuisance parameters

→ Too many; cannot do inference without neglecting uncertainties

* Our physical model and simulations are based on gammaALPs by Manuel Meyer [2]

Parameters in our ALP-model*

- ALP mass, m
- ALP coupling to photons, g

} Parameters of interest

- NGC1275 intrinsic spectrum amplitude
- NGC1275 intrinsic spectral index
- NGC1275 intrinsic cut-off energy

- Magnetic field strength of NGC1275
- Magnetic field configuration
- Extension of Perseus cluster
- 7 electron density-related parameters
- 3 turbulence-related parameters

} $\sim$15 Nuisance parameters

→ Astrophysical $\Rightarrow$ large uncertainties

→ Too many; cannot do inference without neglecting uncertainties

* Our physical model and simulations are based on **gammaALPs** by Manuel Meyer [2]

Parameters in our ALP-model*

- ALP mass, m
- ALP coupling to photons, g

} Parameters of interest

- NGC1275 intrinsic spectrum amplitude
- NGC1275 intrinsic spectral index
- NGC1275 intrinsic cut-off energy

- Magnetic field strength of NGC1275
- Magnetic field configuration
- Extension of Perseus cluster
- 7 electron density-related parameters
- 3 turbulence-related parameters

} ~15 Nuisance parameters

→ Astrophysical ⇒ large uncertainties

→ Too many; cannot do inference without neglecting uncertainties

⇓

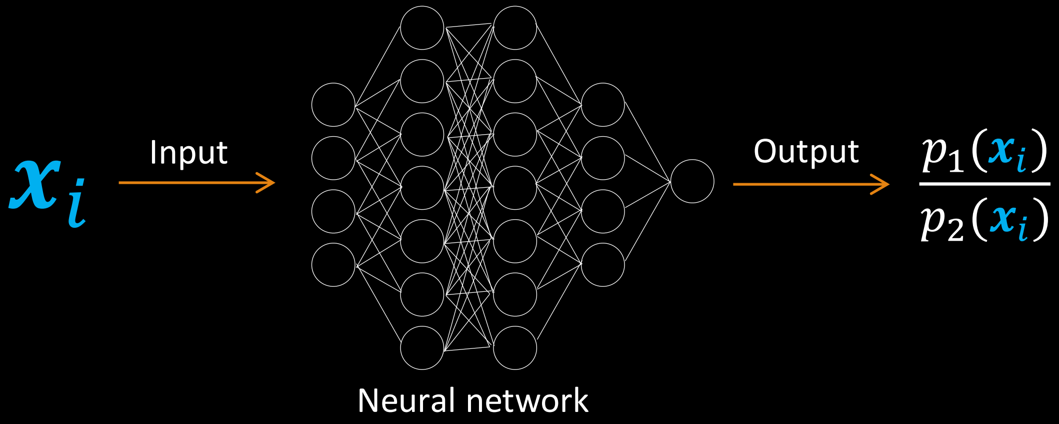
Risk of overconfident limits!

* Our physical model and simulations are based on gammaALPs by Manuel Meyer [2]

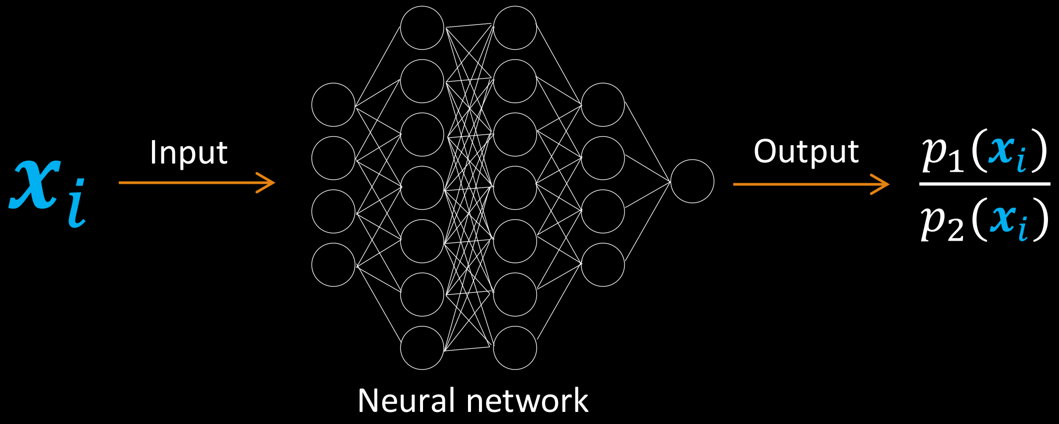
Simulation-based inference is gaining traction as an alternative approach, particularly to do Bayesian inference

The Likelihood Ratio Trick

The Likelihood Ratio Trick



The Likelihood Ratio Trick



This will allow us to estimate the likelihood ratio in Bayes' theorem!

$$p(\boldsymbol{\vartheta}|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\boldsymbol{\vartheta})}{p(\boldsymbol{x})} p(\boldsymbol{\vartheta})$$

The Likelihood Ratio Trick – How to

The Likelihood Ratio Trick – How to

1. Draw many samples:

$$\boldsymbol{x}_0, \boldsymbol{x}_2, \boldsymbol{x}_4, \boldsymbol{x}_6, \dots \sim p_1(\boldsymbol{x})$$

$$\boldsymbol{x}_1, \boldsymbol{x}_3, \boldsymbol{x}_5, \boldsymbol{x}_7, \dots \sim p_2(\boldsymbol{x})$$

The Likelihood Ratio Trick – How to

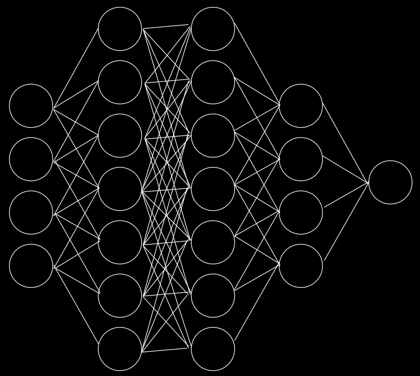
1. Draw many samples:

$$\boldsymbol{x}_0, \boldsymbol{x}_2, \boldsymbol{x}_4, \boldsymbol{x}_6, \ldots \sim \; p_1(\boldsymbol{x})$$

$$\boldsymbol{x}_1, \boldsymbol{x}_3, \boldsymbol{x}_5, \boldsymbol{x}_7, \ldots \sim \; p_2(\boldsymbol{x})$$

2. Train the network to classify the samples:

$\boldsymbol{x}_i$ Input →



Trained neural network

→ Output $d \in (0\,,1)$

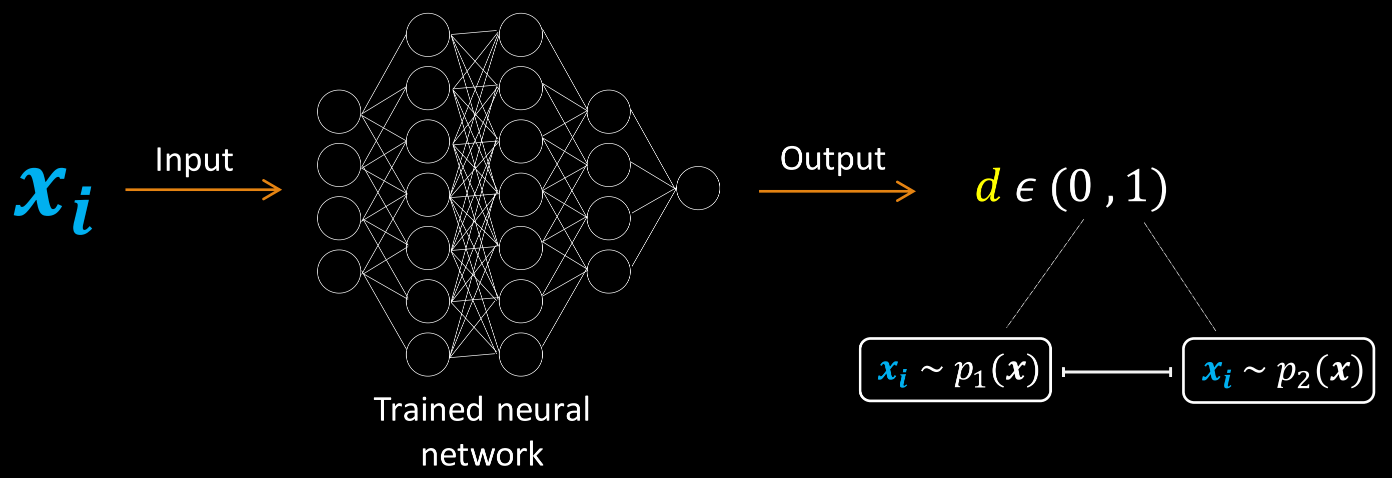
The Likelihood Ratio Trick – How to

1. Draw many samples:

$$\boldsymbol{x}_0, \boldsymbol{x}_2, \boldsymbol{x}_4, \boldsymbol{x}_6, \ldots \sim p_1(\boldsymbol{x})$$

$$\boldsymbol{x}_1, \boldsymbol{x}_3, \boldsymbol{x}_5, \boldsymbol{x}_7, \ldots \sim p_2(\boldsymbol{x})$$

2. Train the network to classify the samples:



The Likelihood Ratio Trick – How to

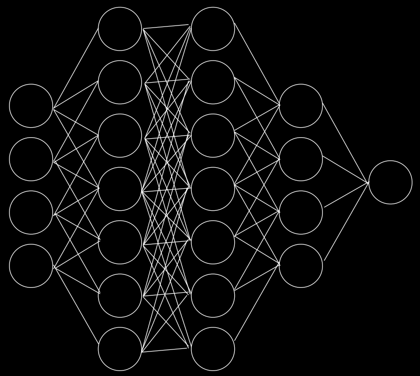
1. Draw many samples:

$$\boldsymbol{x}_0, \boldsymbol{x}_2, \boldsymbol{x}_4, \boldsymbol{x}_6, \ldots \sim p_1(\boldsymbol{x})$$

$$\boldsymbol{x}_1, \boldsymbol{x}_3, \boldsymbol{x}_5, \boldsymbol{x}_7, \ldots \sim p_2(\boldsymbol{x})$$

2. Train the network to classify the samples:

$\boldsymbol{x}_i$ Input →



Trained neural network

Output → $d \in (0\,, 1)$

$\boldsymbol{x}_i \sim p_1(\boldsymbol{x})$ ⟼ $\boldsymbol{x}_i \sim p_2(\boldsymbol{x})$

Mathematical magic ⇒

$$\frac{p_1(\boldsymbol{x}_i)}{p_2(\boldsymbol{x}_i)} = f(d)$$

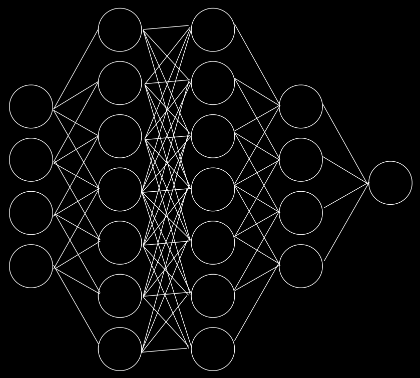
The Likelihood Ratio Trick – How to

1. Draw many samples:

$$\boldsymbol{x}_0, \boldsymbol{x}_2, \boldsymbol{x}_4, \boldsymbol{x}_6, \dots \sim p_1(\boldsymbol{x})$$
$$\boldsymbol{x}_1, \boldsymbol{x}_3, \boldsymbol{x}_5, \boldsymbol{x}_7, \dots \sim p_2(\boldsymbol{x})$$

2. Train the network to classify the samples:

$\boldsymbol{x}_i$ → Input



Trained neural network

Output → $d \in (0\,,1)$

$\boldsymbol{x}_i \sim p_1(\boldsymbol{x})$ ⟝ $\boldsymbol{x}_i \sim p_2(\boldsymbol{x})$

Mathematical magic ⇒

$$\frac{p_1(\boldsymbol{x}_i)}{p_2(\boldsymbol{x}_i)} = \frac{d}{1-d}$$

Heuristic derivation of the likelihood ratio trick

When the neural network is trained, it is actually trying to minimize a loss function. Different loss functions are possible, but for clarity we will assume that we are using the *Binary Cross Entropy* in this derivation:

$$\text{LOSS} = -\sum_{\boldsymbol{x}_i} \left[\; y \ln d + (1-y)\ln(1-d) \; \right]$$

- The sum is over all training samples
- $y = 0$ if a training sample was drawn from $p_1(\boldsymbol{x})$, 1 if from $p_2(\boldsymbol{x})$.
- d = output from neural network for given sample input

Notice that the loss becomes smaller when the network manages to categorize more samples correctly. In the limit of infinite training samples, the loss becomes

$$\text{LOSS} \rightarrow \iint \left[\; p_1(\boldsymbol{x}) \ln d + p_2(\boldsymbol{x}) \ln(1-d) \; \right] \mathrm{d}\boldsymbol{x}$$

We **assume that the network manages to optimize the loss function perfectly**. If this is the case, the derivative of the loss with respect to the neural network's hyperparameters (weights and biases, which are adjusted during training), which we denote by $\boldsymbol{\varphi}$, is 0. Notice that in the integrand, only d is dependent on $\boldsymbol{\varphi}$.

The likelihood ratio trick follows from setting the expression in the square brackets to zero.

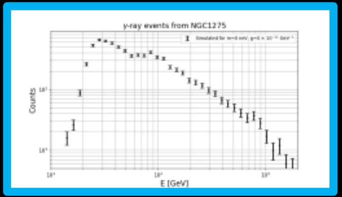
$$\frac{\partial}{\partial \boldsymbol{\varphi}} LOSS = \iint \left[\frac{p_1(\boldsymbol{x})}{d} - \frac{p_2(\boldsymbol{x})}{1-d} \right] \frac{\partial d}{\partial \boldsymbol{\varphi}} = 0 \quad \Longrightarrow \quad \frac{d}{1-d} = \frac{p_1(\boldsymbol{x})}{p_2(\boldsymbol{x})}$$

Neural Ratio Estimation: posteriors from the LR trick

Neural Ratio Estimation: posteriors from the LR trick

1. Draw training samples $(\boldsymbol{x}_i, m_i, g_i)$

$\boldsymbol{x}_i =$



$m_i = 2 \text{ neV}$

$g_i = 3 \times \frac{10^{-11}}{\text{GeV}}$

example samples

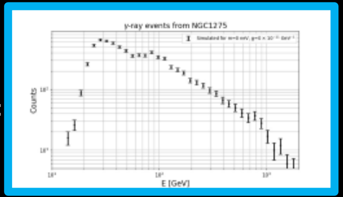
$(\boldsymbol{x}_i, m_i, g_i)$

Neural Ratio Estimation: posteriors from the LR trick

1. Draw training samples $(\boldsymbol{x}_i, m_i, g_i)$

from $p(\boldsymbol{x}|m, g)\, p(m, g)$

and from $p(\boldsymbol{x})\, p(m, g)$

$\boldsymbol{x}_i =$ 

$m_i = 2$ neV

$g_i = 3 \times \frac{10^{-11}}{\text{GeV}}$

example samples $(\boldsymbol{x}_i, m_i, g_i)$

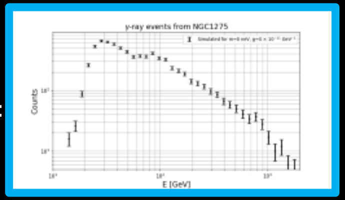
Neural Ratio Estimation: posteriors from the LR trick

1. Draw training samples $(\boldsymbol{x}_i, m_i, g_i)$

from $p(\boldsymbol{x}|m, g)\, p(m, g)$

and from $p(\boldsymbol{x})\, p(m, g)$

2. Train a NN to distinguish the samples

$\boldsymbol{x}_i =$ 

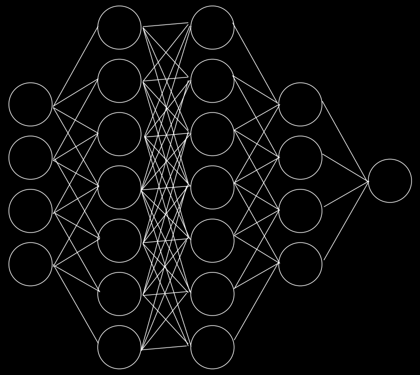
$m_i = 2$ neV

$g_i = 3 \times \frac{10^{-11}}{\text{GeV}}$

example samples

$(\boldsymbol{x}_i, m_i, g_i)$

Input



Neural network

Modified Output

$$\frac{p(\boldsymbol{x}_i|m_i, g_i)}{p(\boldsymbol{x}_i)}$$

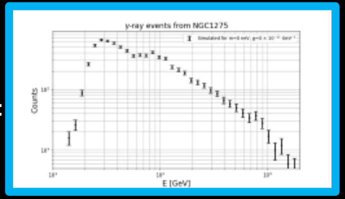
Neural Ratio Estimation: posteriors from the LR trick

1. Draw training samples $(\boldsymbol{x}_i, m_i, g_i)$

from $p(\boldsymbol{x}|m,g)\,p(m,g)$

and from $p(\boldsymbol{x})\,p(m,g)$

2. Train a NN to distinguish the samples

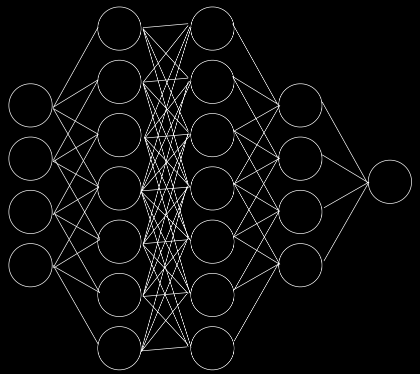
$\boldsymbol{x}_i =$ 

$m_i = 2$ neV

$g_i = 3 \times \frac{10^{-11}}{\text{GeV}}$

example samples $(\boldsymbol{x}_i, m_i, g_i)$

Input



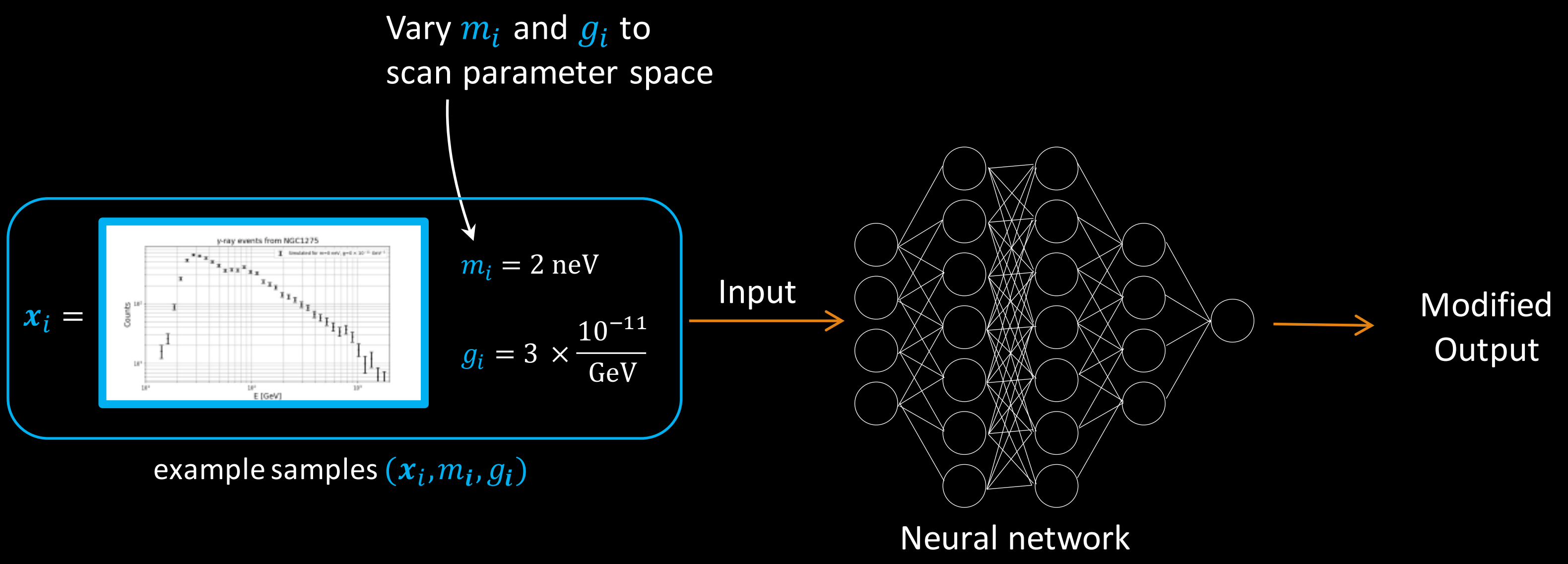
Neural network

Modified Output

$$\frac{p(\boldsymbol{x}_i|m_i,g_i)}{p(\boldsymbol{x}_i)}$$

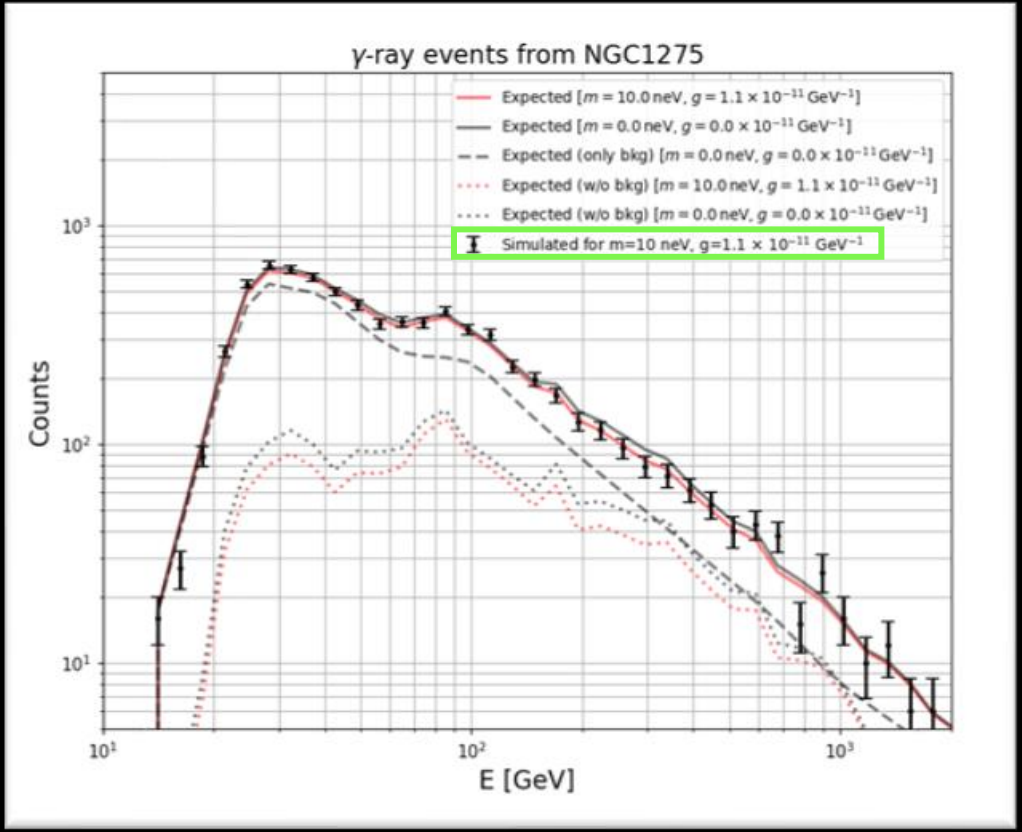
$\times$ Prior = Posterior!

Summary so far



Posterior = Modified Output × Prior

How our simulations are made



Simulated data

=

γ-rays ~ cut-off power law

$$\varphi(E) = \varphi_0 \left(\frac{E}{E_0}\right)^{\gamma} e^{-E/E_{cut}}$$

×

Instrument response

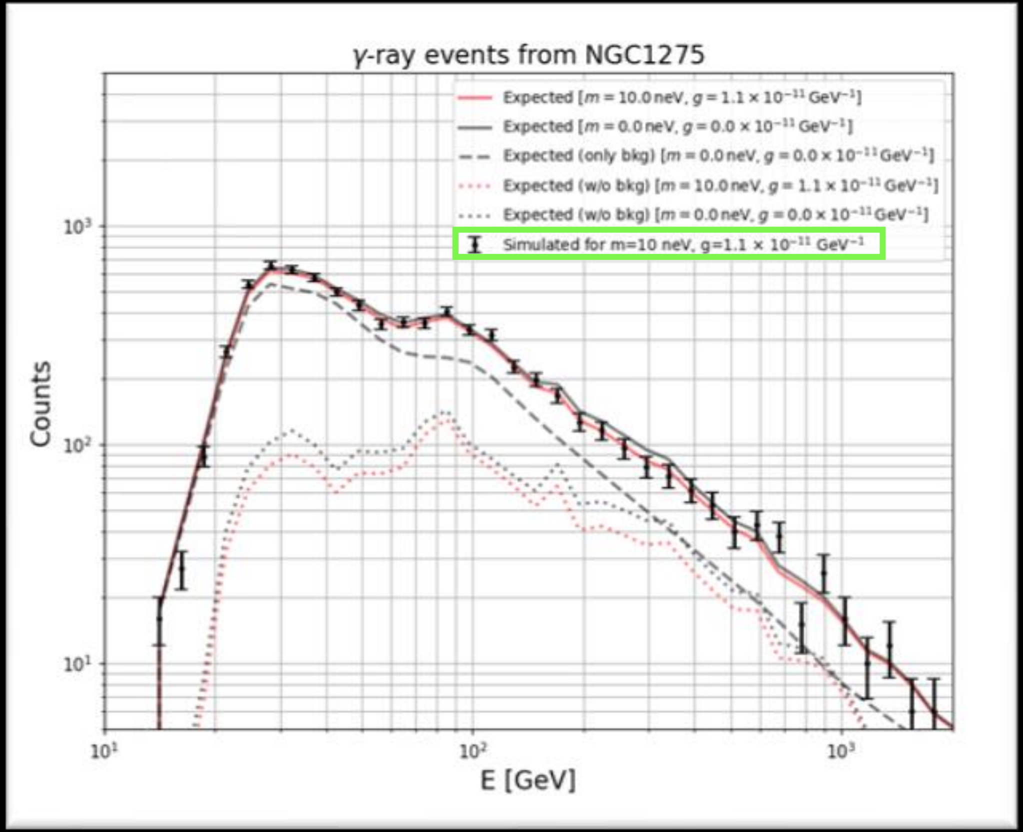
Using the IRF (for CTA) prod3: South_z20_50h, together with gammapy 0.19 [3]

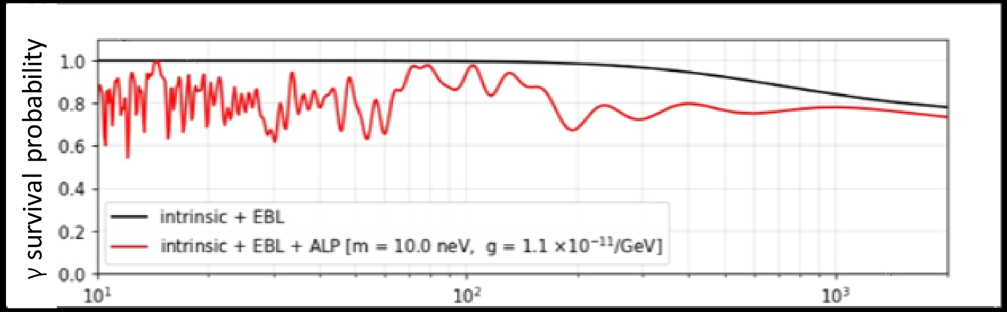
+

Cosmic ray background

Using the IRF (for CTA) prod3: South_z20_50h, together with gammapy 0.19 [3]

How our simulations are made





Simulated data

=

γ-rays ~ cut-off power law

$$\varphi(E) = \varphi_0 \left(\frac{E}{E_0}\right)^{\gamma} e^{-E/E_{cut}}$$

×

Instrument response

Using the IRF (for CTA) prod3: South_z20_50h, together with gammapy 0.19 [3]

×

Absorption from EBL

Using gammaALPs [2]

×

"Wiggles" from photon-ALP-oscillations

+

Cosmic ray background

Using the IRF (for CTA) prod3: South_z20_50h, together with gammapy 0.19 [3]

Let's start with a simplified case study...

<u>Model Parameters</u>

- ALP mass, m
- ALP coupling to photons, g

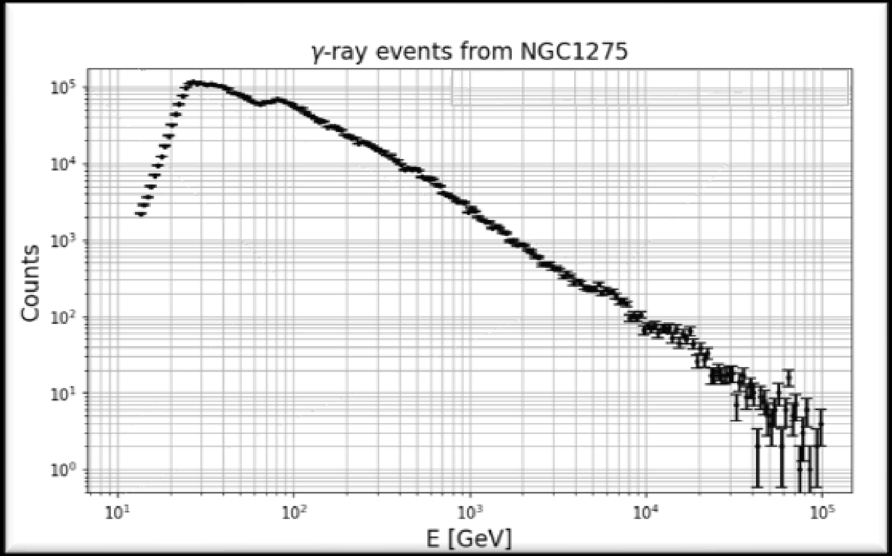
} Parameters of interest

- NGC1275 intrinsic spectrum amplitude
- NGC1275 intrinsic spectral index
- NGC1275 intrinsic cut-off energy

} Nuisance parameters

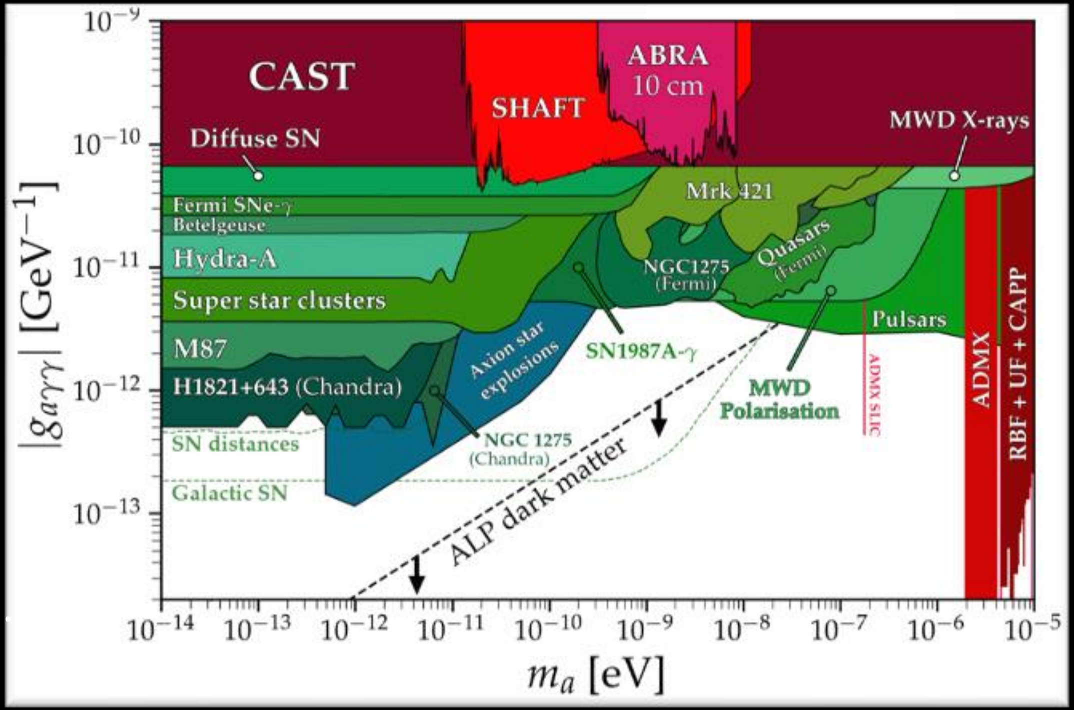
<u>Modelling specifications</u>

- ~ 1 000 000 simulations used in training
- Prior: uniform on log scale
- Assumed observation time = 50 hr
- 200 energy bins in range between 10 GeV and 100 TeV



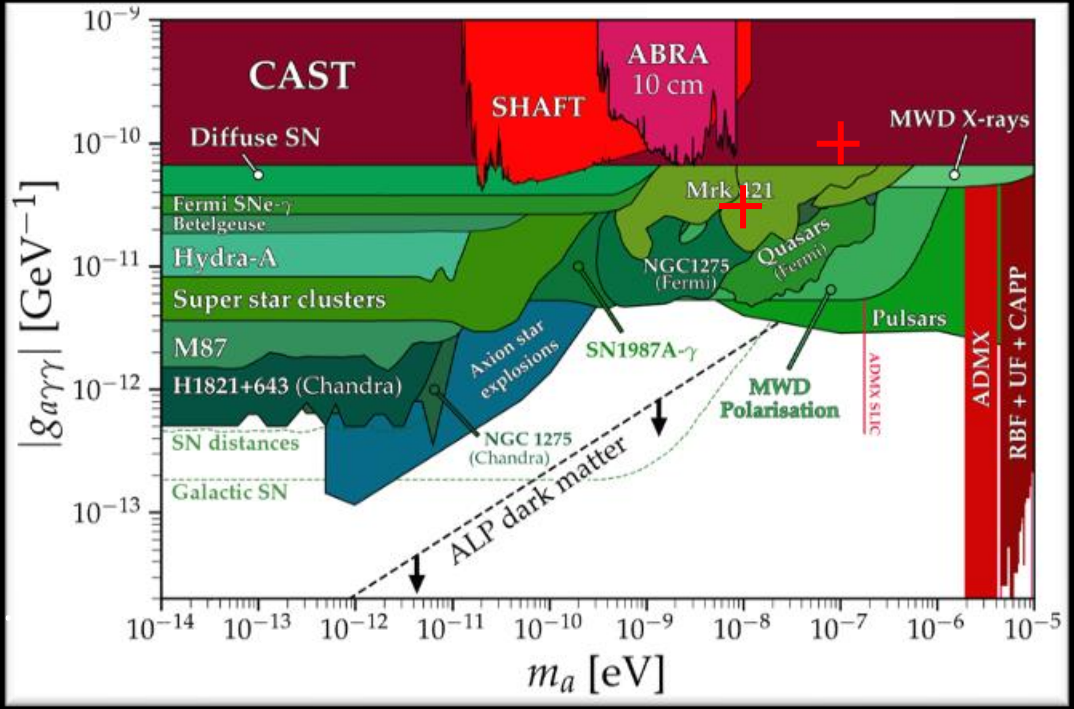
Simulated observation

First attempts at ALP-inference show promise



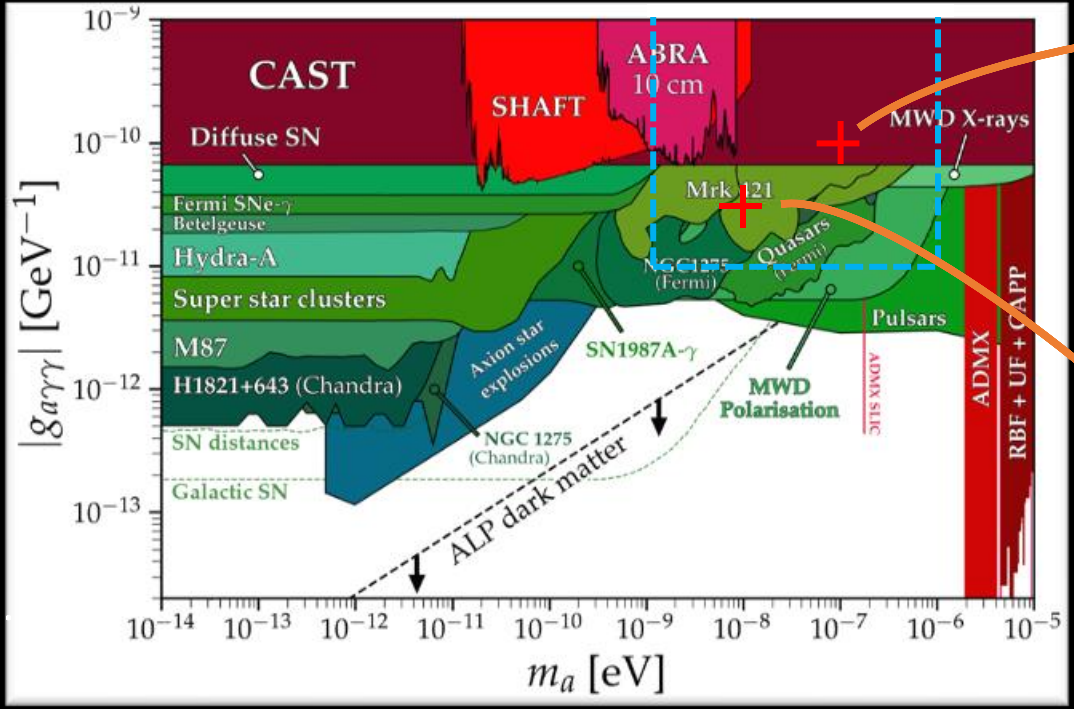
Source: [6]

First attempts at ALP-inference show promise

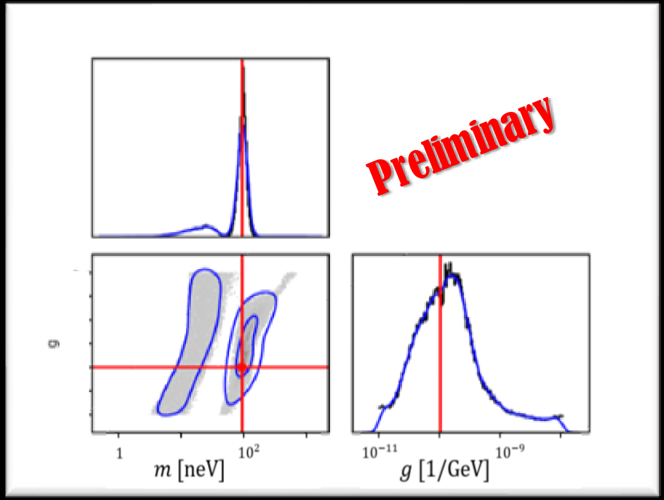


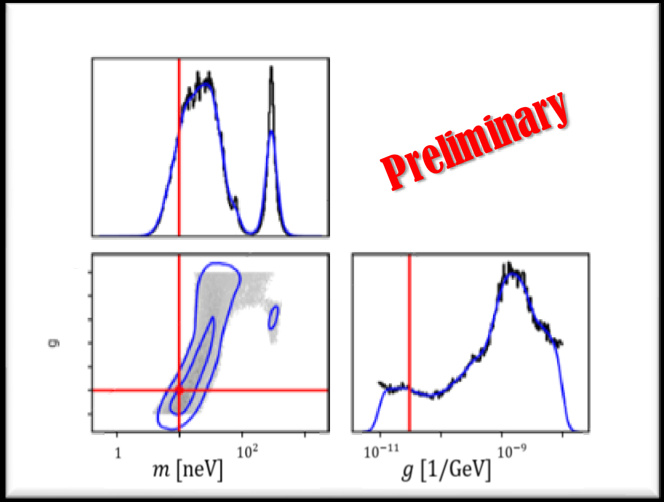
Source: [6]

First attempts at ALP-inference* show promise



Source: [6]

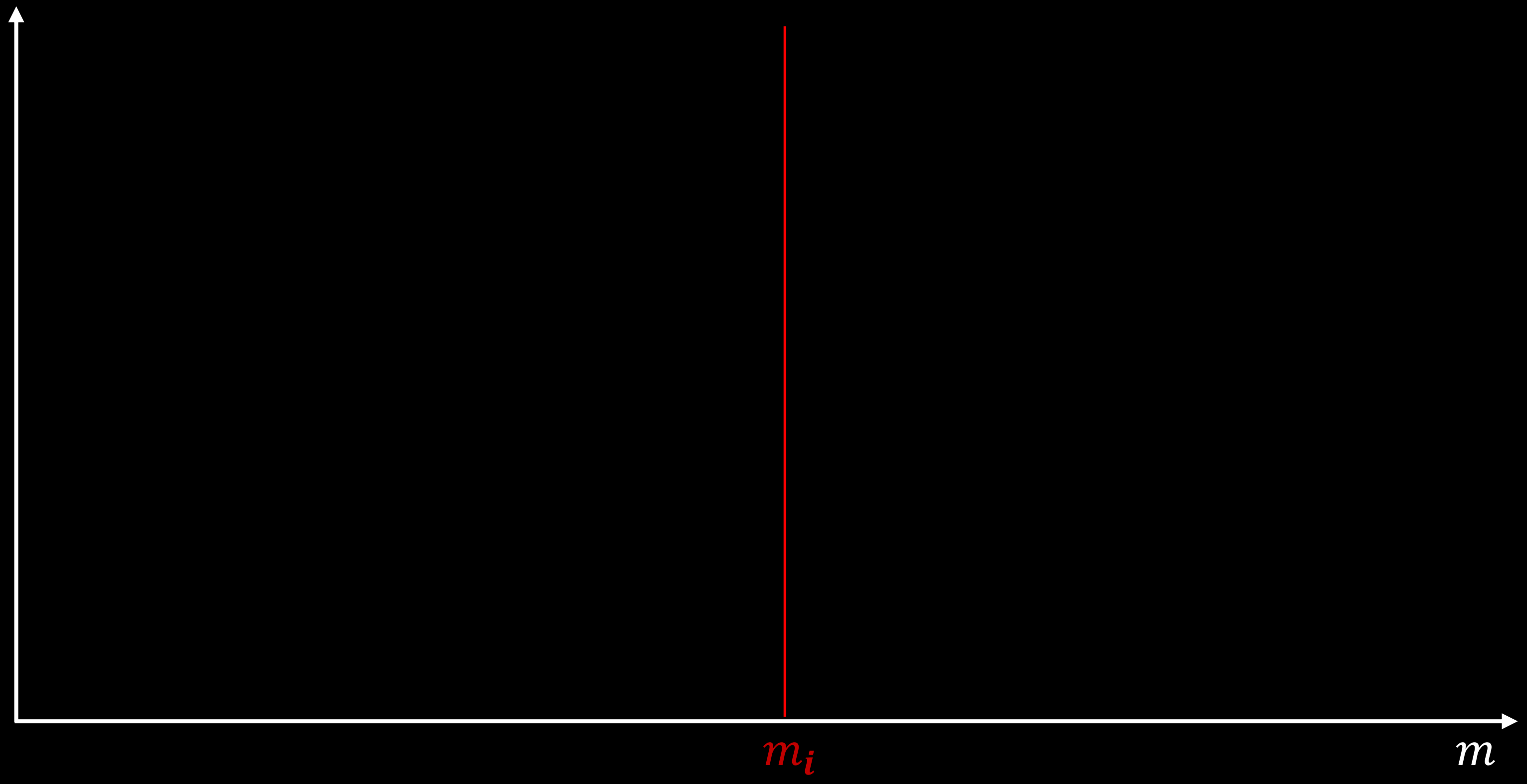




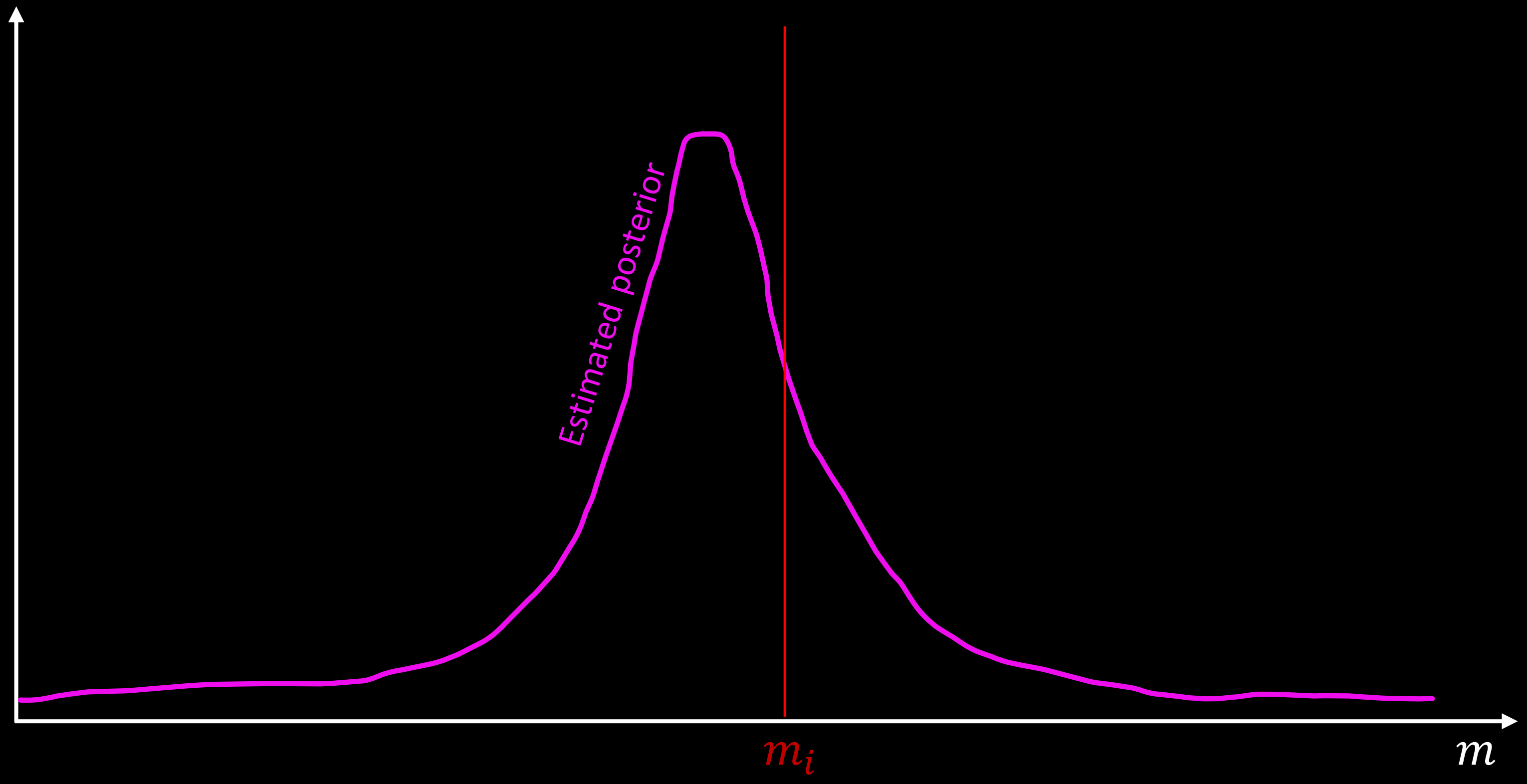
* We perform the NRE posterior estimation using the open-source python package SWYFT [4]

How can we trust the estimated posterior to accurately represent the true posterior?

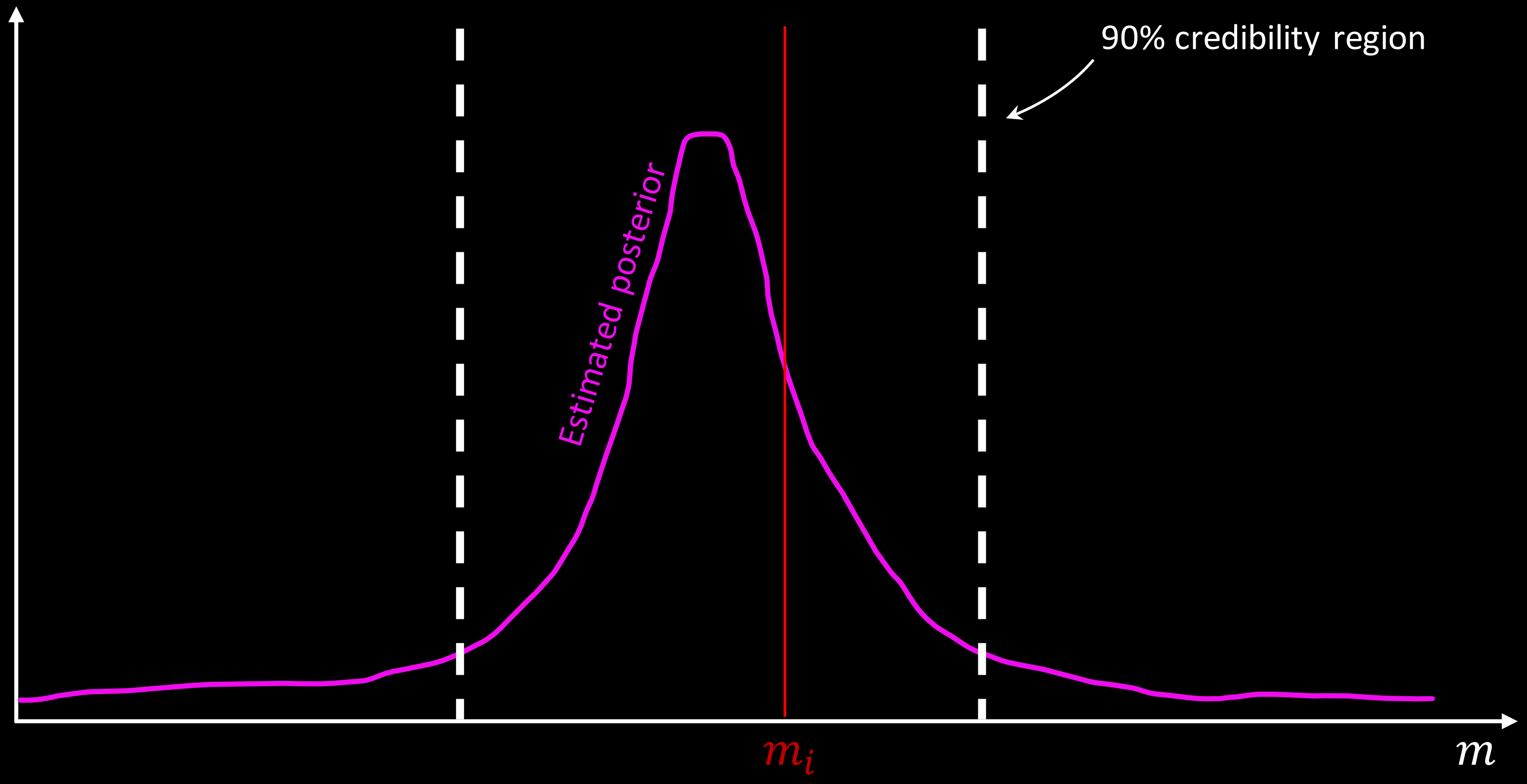
Validation of estimated posteriors



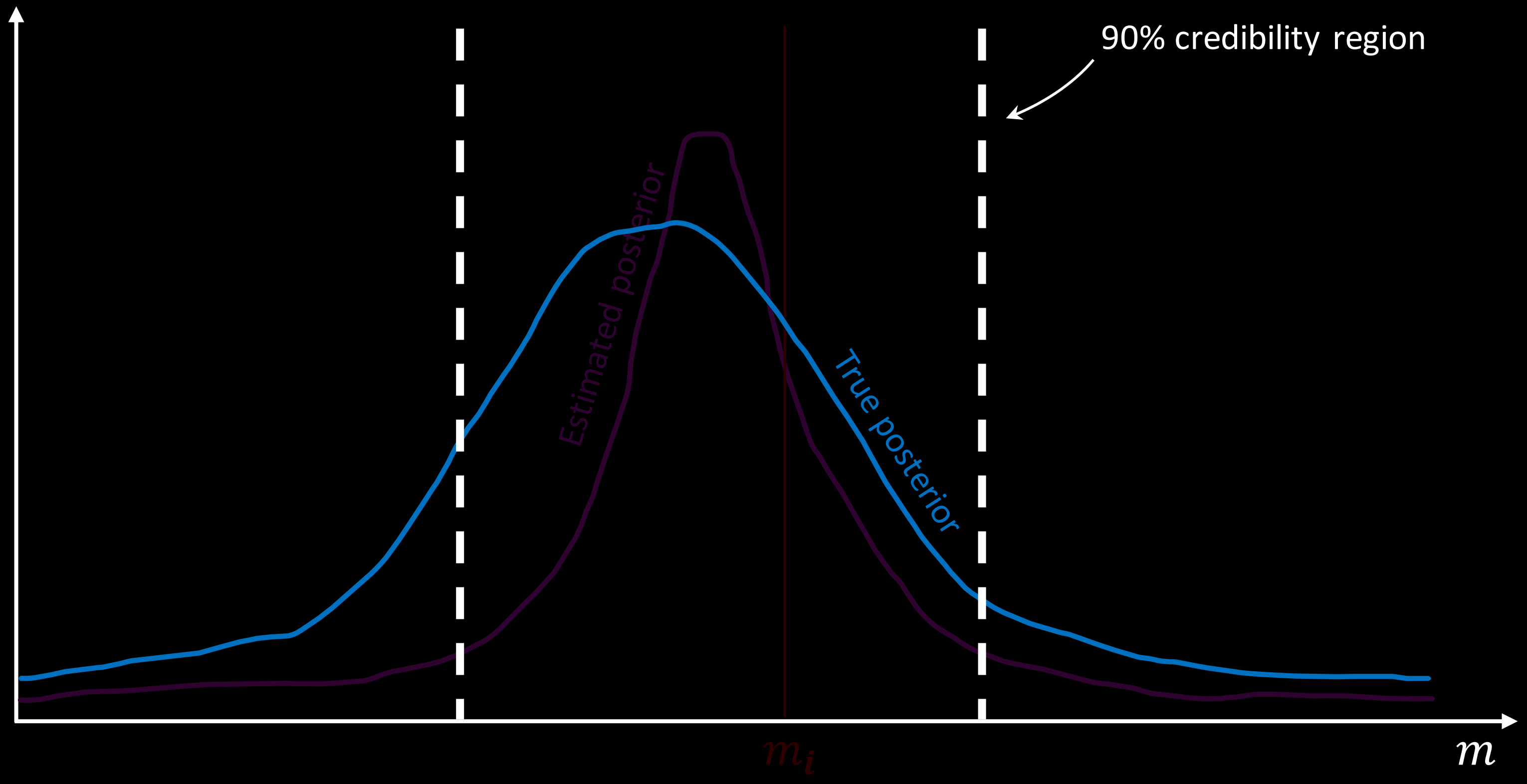
Validation of estimated posteriors



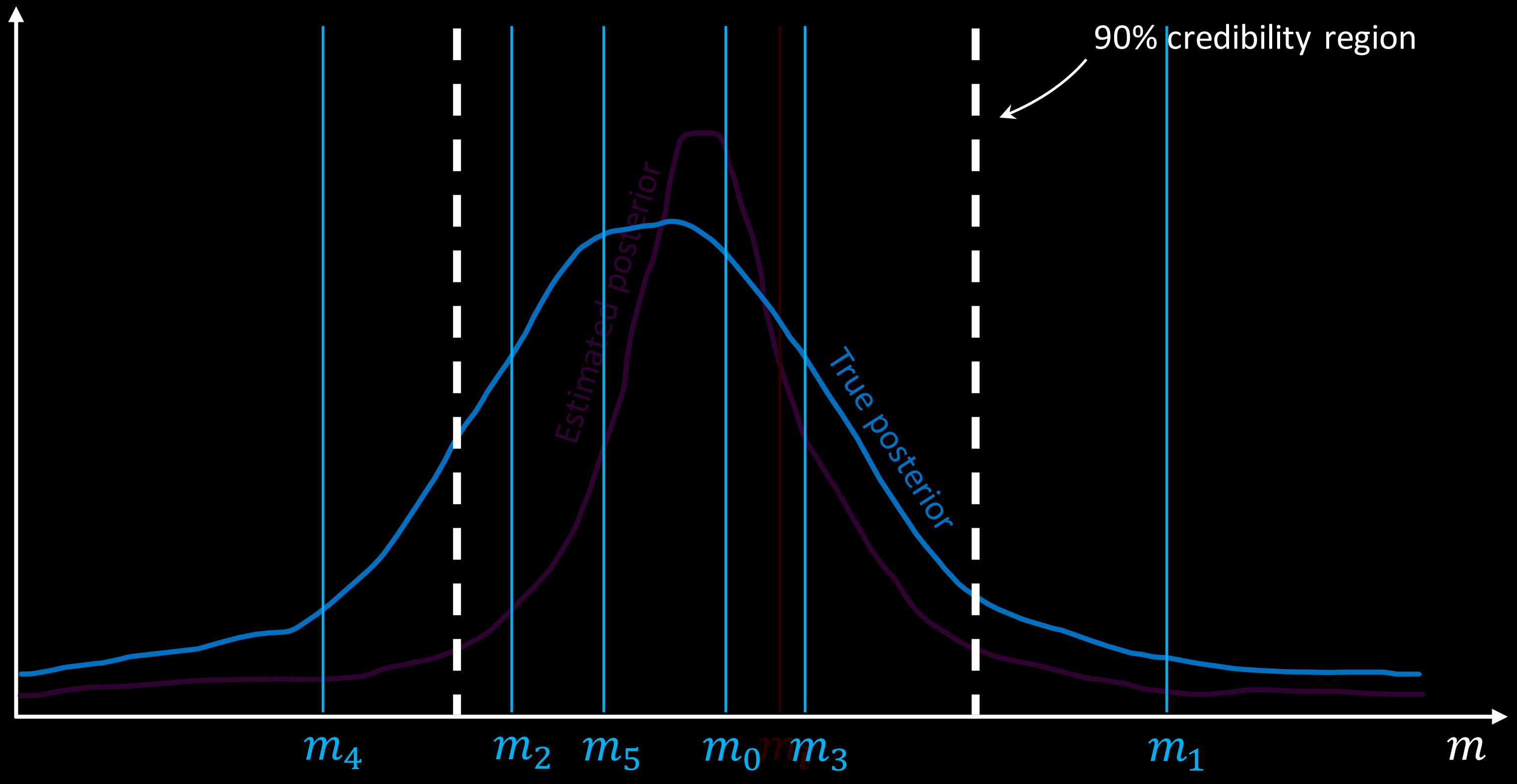
Validation of estimated posteriors (for one parameter)



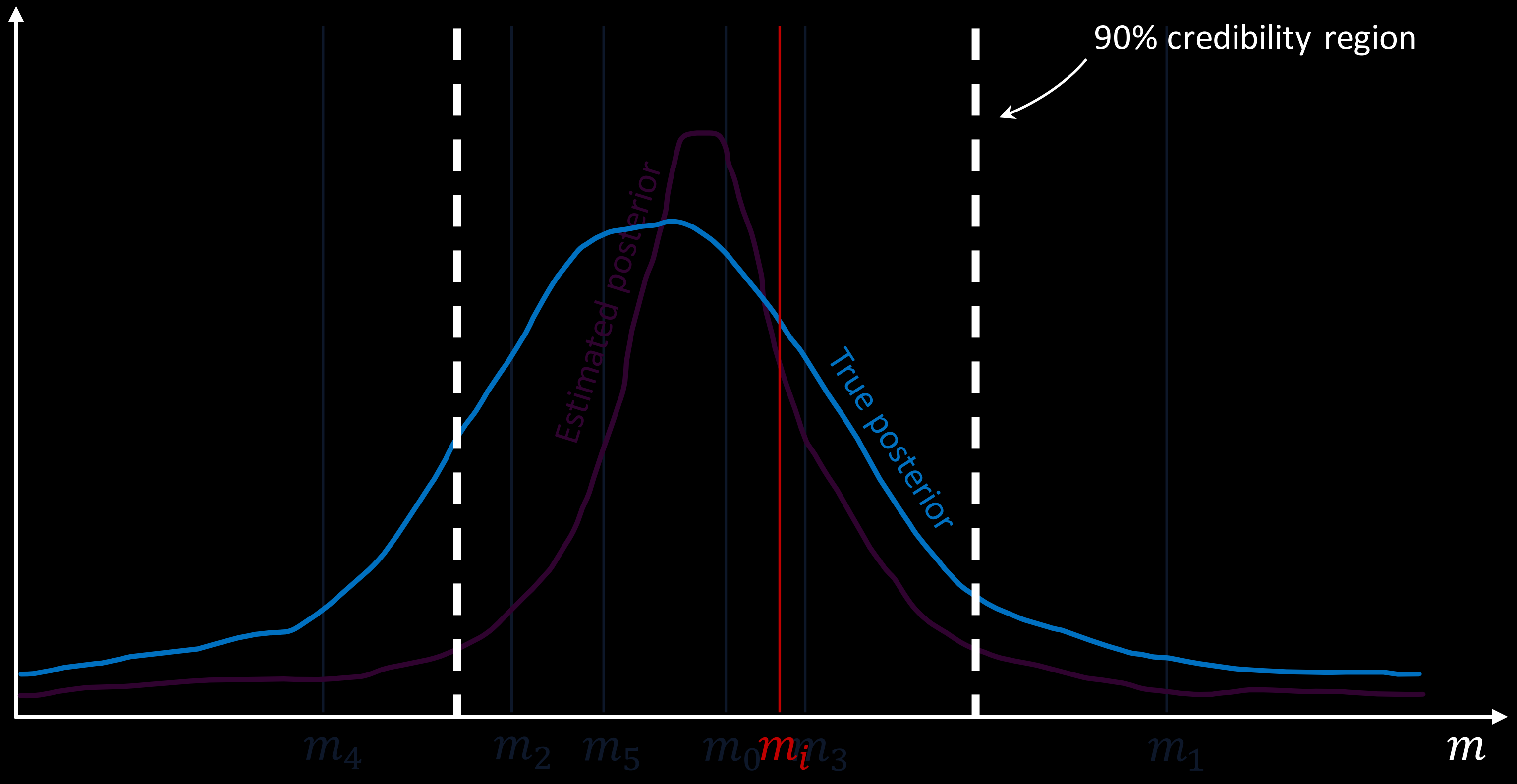
Validation of estimated posteriors



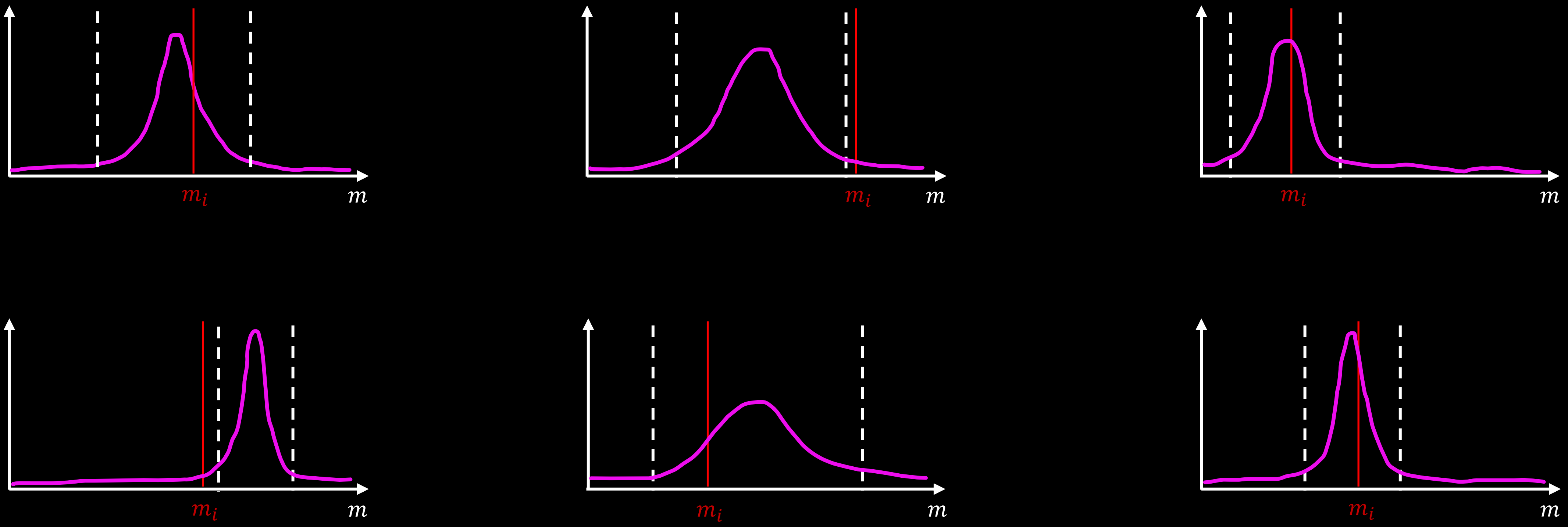
Validation of estimated posteriors



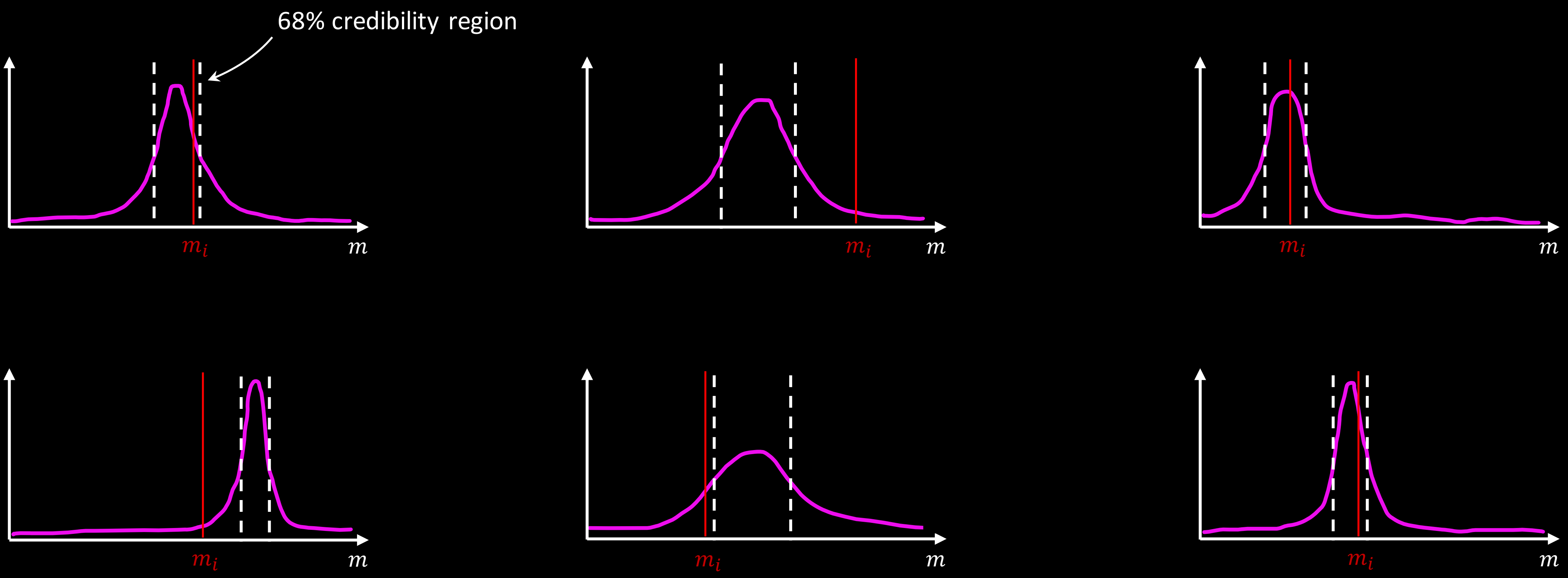
Validation of estimated posteriors



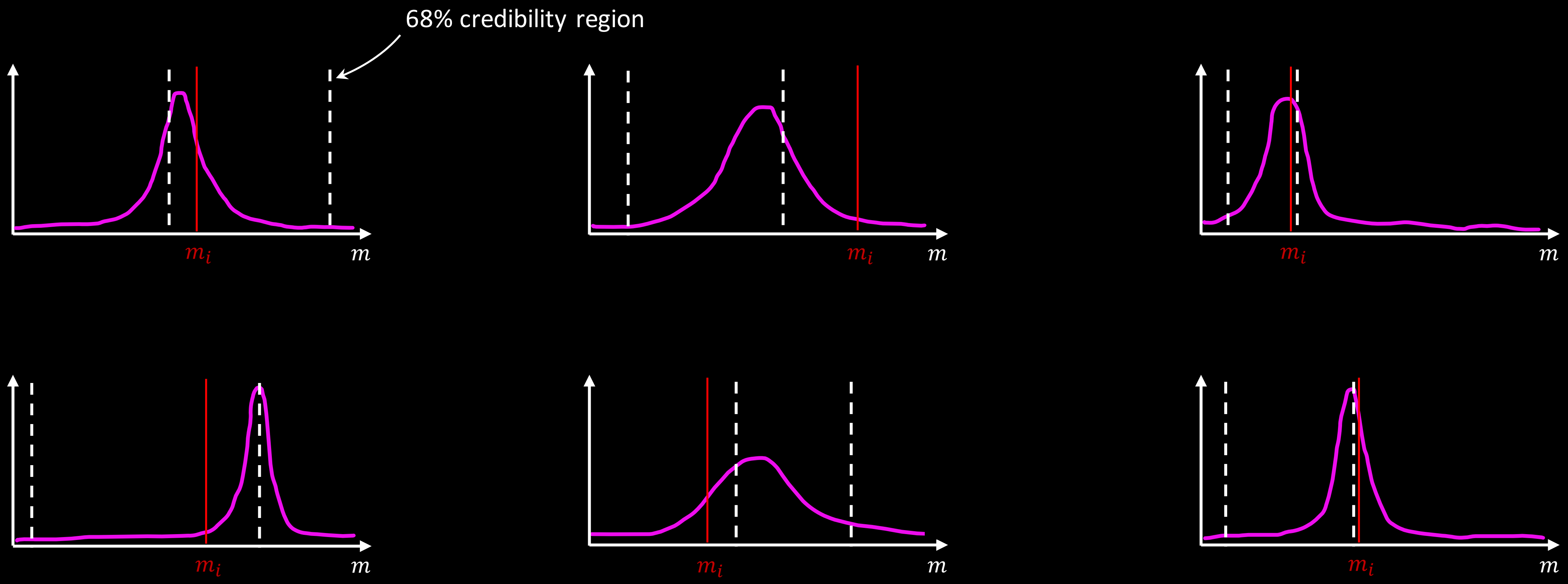
Validation of estimated posteriors



Validation of estimated posteriors



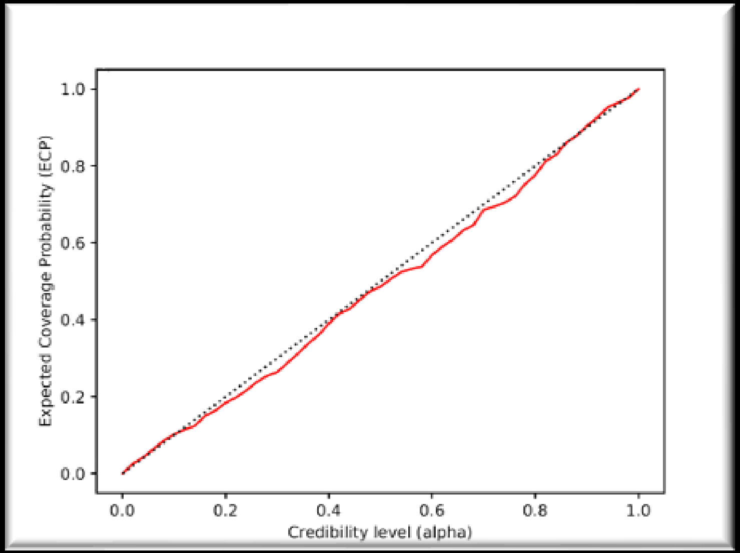
Validation of estimated posteriors



Expected coverage testing indicates we are on the right track

Validation of *(mass, coupling)*

ECP



α

ECP $\approx \alpha$ for all α

… but we still got a (little) ways to go

1. We may still not be training our NNs to their full potential

… but we still got a (little) ways to go

1. We may still not be training our NNs to their full potential

   - Related to technical (hardware) limitations → hopefully soon to be solved.
   - But validation indicates we are on the right track!

… but we still got a (little) ways to go

1. We may still not be training our NNs to their full potential

    - Related to technical (hardware) limitations → hopefully soon to be solved.
    - But validation indicates we are on the right track!

… but we still got a (little) ways to go

1. We may still not be training our NNs to their full potential

    - Related to technical (hardware) limitations → hopefully soon to be solved.
    - But validation indicates we are on the right track!

2. We want to take into account several more nuisance parameters

… but we still got a (little) ways to go

1. We may still not be training our NNs to their full potential
   - Related to technical (hardware) limitations → hopefully soon to be solved.
   - But validation indicates we are on the right track!
2. We want to take into account several more nuisance parameters
3. We need to switch to more state-of-the-art instrument response functions

Good news:
Steps 2 and 4 don't require any new implementations!

… but we still got a (little) ways to go

1. We may still not be training our NNs to their full potential
   - Related to technical (hardware) limitations → hopefully soon to be solved.
   - But validation indicates we are on the right track!

2. We want to take into account several more nuisance parameters

3. We need to switch to more state-of-the-art instrument response functions

4. We need to explore a larger region of parameter space

… but we still got a (little) ways to go

1. We may still not be training our NNs to their full potential
   - Related to technical (hardware) limitations → hopefully soon to be solved.
   - But validation indicates we are on the right track!
2. We want to take into account several more nuisance parameters
3. We need to switch to more state-of-the-art instrument response functions
4. We need to explore a larger region of parameter space

Good news:

Steps 2 and 4 don't require any new implementations!

Summary

- Axion-like particles (ALP) are popular beyond-SM and DM candidates, with potential – but difficult – detectability in cosmic gamma-rays.

- Neural Ratio Estimation (NRE) may allow us to produce accurate ALP-limits that would otherwise be overconfident.

- Recent developments are making it possible to assess the reliability of NRE, thus making it a serious contender to conventional inference techniques.

- Our preliminary results indicate that NRE is a viable method for our physics case

- Our analysis will likely improve significantly soon, given more machine learning-friendly technical resources.

Summary

- Axion-like particles (ALP) are popular beyond-SM and DM candidates, with potential – but difficult – detectability in cosmic gamma-rays.
- Neural Ratio Estimation (NRE) may allow us to produce accurate ALP-limits that would otherwise be overconfident.
- Recent developments are making it possible to assess the reliability of NRE, thus making it a serious contender to conventional inference techniques.
- Our preliminary results indicate that NRE is a viable method for our physics case
- Our analysis will likely improve significantly soon, given more machine learning-friendly technical resources.

Summary

- Axion-like particles (ALP) are popular beyond-SM and DM candidates, with potential – but difficult – detectability in cosmic gamma-rays.
- Neural Ratio Estimation (NRE) may allow us to produce accurate ALP-limits that would otherwise be overconfident.
- Recent developments are making it possible to assess the reliability of NRE, thus making it a serious contender to conventional inference techniques.
- Our preliminary results indicate that NRE is a viable method for our physics case
- Our analysis will likely improve significantly soon, given more machine learning-friendly technical resources.

References

[1] G. Irastorza and J. Redondo, "New experimental approaches in the search for axion-like particles," Progress in Particle and Nuclear Physics, vol. 102, pp. 89–159, 2018.

[2] https://github.com/me-manu/gammaALPs

[3] https://docs.gammapy.org/0.19/

[4] https://github.com/undark-lab/swyft

[5] P. Lemos, A. Coogan, Y. Hezaveh, and L. Perreault-Levasseur, "Sampling-based accuracy testing of posterior estimators for general inference," in Proceedings of the 40th International Conference on Machine Learning (A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, eds.), vol. 202 of Proceedings of Machine Learning Research, pp. 19256–19273, PMLR, 23–29 Jul 2023

[6] https://github.com/cajohare/AxionLimits/blob/master/docs/ap.md

Readable Version

Overcoming limitations to ALP parameter inference with Neural Ratio Estimation

Gert Kluge

PhD candidate at the University of Oslo

contact: gertwk@uio.no

In collaboration with

Giacomo D'Amico, Julia Djuvsland, and Heidi Sandaker

Axion-like particles

What we all do!

Overcoming limitations to ALP parameter inference with *Neural Ratio Estimation*

Not limited to ALP searches!

Inference using **machine learning**

A method of
"Simulation-based inference (SBI)"
or
"Likelihood-free inference"

the Cherenkov Telescope Array (CTA)

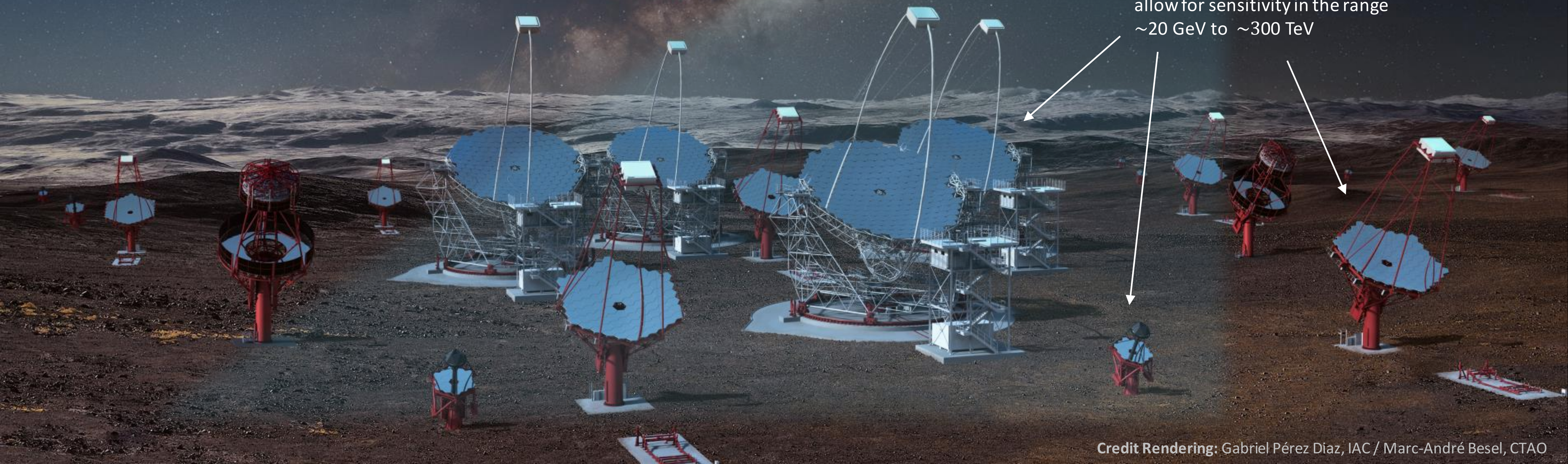
Set to become the world's leading very high-energy γ-ray telescope

- Order of magnitude more sensitive than current Cherenkov telescopes at TeV-level
- Energy resolution better than 10 percent at TeV-level

Photon directions and energies are reconstructed from extensive air showers

Cherenkov light

3 different sizes of telescopes allow for sensitivity in the range ~20 GeV to ~300 TeV



Credit Rendering: Gabriel Pérez Diaz, IAC / Marc-André Besel, CTAO

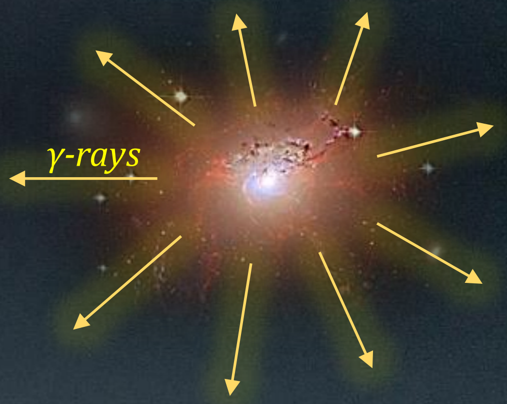
We want to search for ALPs with CTA

Axion-like particles

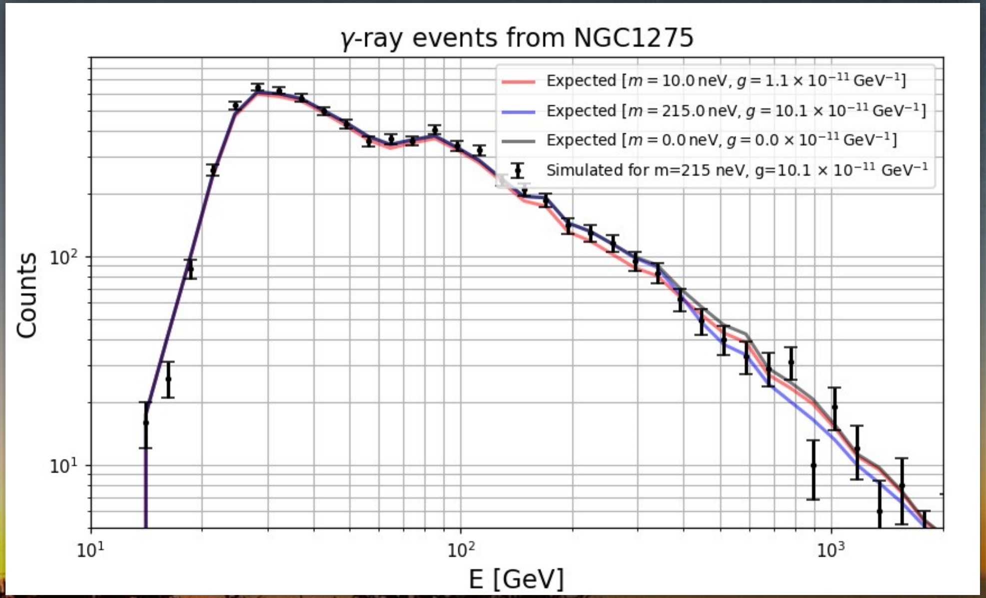
- Extension of the QCD-Axion [1]
- Popular candidates for dark matter and physics beyond the standard model
- **Can oscillate into photons in the presence of magnetic fields**

Credit NGC1275: A. Fabian (Institute of Astronomy, University of Cambridge). NASA, ESA, and the Hubble Heritage.

We want to search for ALPs with CTA



γ-ray spectra from bright sources with strong magnetic fields may show signs of ALP-photon oscillations.



Simulated γ-ray spectrum of NGC1275. See slide 16 for details.

Credit NGC1275: A. Fabian (Institute of Astronomy, University of Cambridge). NASA, ESA, and the Hubble Heritage.

Problem:

How can we do reliable parameter inference for this complicated model?

Parameter inference requires computing likelihood ratios

$\boldsymbol{x}$ = Observation
$\boldsymbol{\vartheta}$ = Parameters of interest
$\boldsymbol{\theta}$ = Nuisance parameters
i = Sample index

Frequentist approach:

$$TS(\boldsymbol{\vartheta}_i|\boldsymbol{x}_i) = -2\ \log \frac{\sup_{\boldsymbol{\theta}} p(\boldsymbol{x}_i|\boldsymbol{\vartheta}_i, \boldsymbol{\theta})}{\sup_{(\boldsymbol{\vartheta},\boldsymbol{\theta})} p(\boldsymbol{x}_i|\boldsymbol{\vartheta}, \boldsymbol{\theta})}$$

⇒ Must optimize over all parameters

Bayesian approach:

$$p(\boldsymbol{\vartheta}_i|\boldsymbol{x}_i) = \frac{p(\boldsymbol{x}_i|\boldsymbol{\vartheta}_i)}{p(\boldsymbol{x}_i)} p(\boldsymbol{\vartheta}_i)$$

⇒ Must integrate over nuisance parameters

Takes too long if there are
too many (nuisance) parameters

Parameters in our ALP-model*

- ALP mass, m
- ALP coupling to photons, g

Parameters of interest

- NGC1275 intrinsic spectrum amplitude
- NGC1275 intrinsic spectral index
- NGC1275 intrinsic cut-off energy

- Magnetic field strength of NGC1275
- Magnetic field configuration
- Extension of Perseus cluster
- 7 electron density-related parameters
- 3 turbulence-related parameters

$\sim$15 Nuisance parameters

Astrophysical ⇒ large uncertainties

Too many; cannot calculate the posterior without neglecting uncertainties

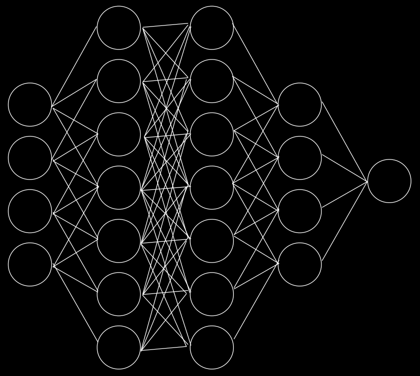
Risk of overconfident posteriors!

* Our physical model and simulations are based on gammaALPs by Manuel Meyer [2]

Simulation based inference is emerging as an alternative approach, particularly to do Bayesian inference

The Likelihood Ratio Trick

Goal: Given two distributions $p_1(\boldsymbol{x})$ and $p_2(\boldsymbol{x})$, train a neural network to do the following:

$\boldsymbol{x}_i$ Input →  → Output $\frac{p_1(\boldsymbol{x}_i)}{p_2(\boldsymbol{x}_i)}$

Neural network

A sample from some distribution

Could be an observation!

This will allow us to estimate the likelihood ratio in Bayes' theorem!

$$p(\boldsymbol{\vartheta}|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\boldsymbol{\vartheta})}{p(\boldsymbol{x})} p(\boldsymbol{\vartheta})$$

How to do the Likelihood Ratio Trick:

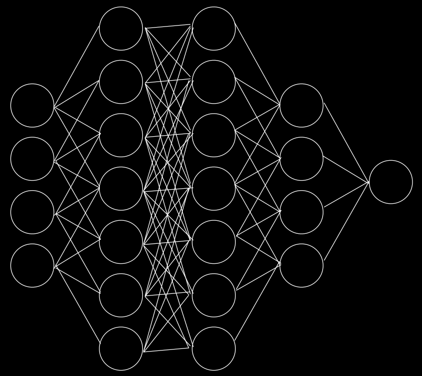
1. Draw many samples from both distributions:

$$\boldsymbol{x}_0, \boldsymbol{x}_2, \boldsymbol{x}_4, \boldsymbol{x}_6, \ldots \sim p_1(\boldsymbol{x})$$

$$\boldsymbol{x}_1, \boldsymbol{x}_3, \boldsymbol{x}_5, \boldsymbol{x}_7, \ldots \sim p_2(\boldsymbol{x})$$

2. Train the network to classify the samples according to which distribution they were likely to have been drawn from:

$\boldsymbol{x}_i$ Input →



Neural network

Output → $d \in (0, 1)$ LR trick → $\frac{p_1(\boldsymbol{x}_i)}{p_2(\boldsymbol{x}_i)} = f(d)$

$\boldsymbol{x}_i \sim p_1(\boldsymbol{x})$ ⟷ $\boldsymbol{x}_i \sim p_2(\boldsymbol{x})$

Known!
(but dependent on loss function)

3. The probability ratio is now a simple known function of the "classification output" d.

For example, if the *binary cross entropy* is used as the NNs loss function, then

$$f(d) = \frac{d}{1-d} = \frac{p_1(\boldsymbol{x}_i)}{p_2(\boldsymbol{x}_i)}$$

Heuristic derivation of the likelihood ratio trick

When the neural network is trained, it is actually trying to minimize a loss function. Different loss functions are possible, but for clarity we will assume that we are using the *Binary Cross Entropy* in this derivation:

$$\text{LOSS} = -\sum_{\boldsymbol{x}_i} \left[\; y \ln d + (1-y)\ln(1-d) \; \right]$$

- The sum is over all training samples
- $y = 0$ if a training sample was drawn from $p_1(\boldsymbol{x})$, 1 if from $p_2(\boldsymbol{x})$.
- d = output from neural network for given sample input

Notice that the loss becomes smaller when the network manages to categorize more samples correctly. In the limit of infinite training samples, the loss becomes

$$\text{LOSS} \rightarrow \iint \left[\; p_1(\boldsymbol{x}) \ln d + p_2(\boldsymbol{x}) \ln(1-d) \; \right] \mathrm{d}\boldsymbol{x}$$

We **assume that the network manages to optimize the loss function perfectly**. If this is the case, the derivative of the loss with respect to the neural network's hyperparameters (weights and biases, which are adjusted during training), which we denote by $\boldsymbol{\varphi}$, is 0. Notice that in the integrand, only d is dependent on $\boldsymbol{\varphi}$.

The likelihood ratio trick follows from setting the expression in the square brackets to zero.

$$\frac{\partial}{\partial \boldsymbol{\varphi}} LOSS = \iint \left[\frac{p_1(\boldsymbol{x})}{d} - \frac{p_2(\boldsymbol{x})}{1-d} \right] \frac{\partial d}{\partial \boldsymbol{\varphi}} = 0 \quad \Longrightarrow \quad \frac{d}{1-d} = \frac{p_1(\boldsymbol{x})}{p_2(\boldsymbol{x})}$$

An intuitive interpretation of the LRT

To have the best possible chance of attributing a sample $\boldsymbol{x_i}$ to the correct distribution, the NN must count what number of training samples similar to $\boldsymbol{x_i}$ came from $p_1(\boldsymbol{x})$ compared to $p_2(\boldsymbol{x})$. The relationship between those two numbers converges to precisely $\frac{p_1(\boldsymbol{x})}{p_2(\boldsymbol{x})}$.

Neural Ratio Estimation: posteriors from the LR trick

1. Draw many samples $(\boldsymbol{x}_i, m_i, g_i)$

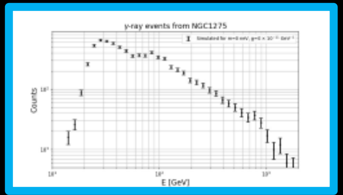
from $p(\boldsymbol{x}|m, g)\, p(m, g)$

i.e. Sample (m_i, g_i) from the prior $p(m, g)$, then simulate $\boldsymbol{x}_i$ from (m_i, g_i).

and from $p(\boldsymbol{x})\, p(m, g)$

i.e. Sample (m_j, g_j) from the prior $p(m, g)$, then simulate $\boldsymbol{x}_i$ from (m_j, g_j).
Then draw (m_i, g_i) from the prior.

2. Train a NN to distinguish the samples, and apply the LR trick

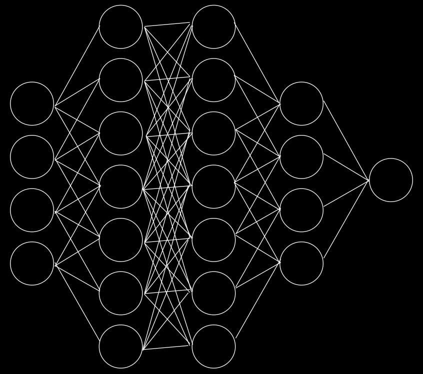
$\boldsymbol{x}_i =$ 

$m_i = 2\ \mathrm{neV}$

$g_i = 3 \times \frac{10^{-11}}{\mathrm{GeV}}$

example samples $(\boldsymbol{x}_i, m_i, g_i)$

Input



Neural network

Modified Output

$$\frac{p(\boldsymbol{x}_i|m_i, g_i)\, \cancel{p(m_i, g_i)}}{p(\boldsymbol{x}_i)\, \cancel{p(m_i, g_i)}} = \frac{p(m_i, g_i|\boldsymbol{x}_i)}{p(m_i, g_i)}$$

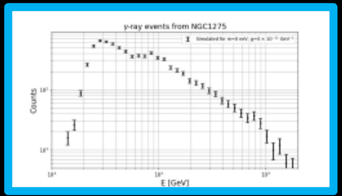
By Bayes' theorem

Posterior!

Prior

Summary so far

Vary m_i and g_i to
scan parameter space

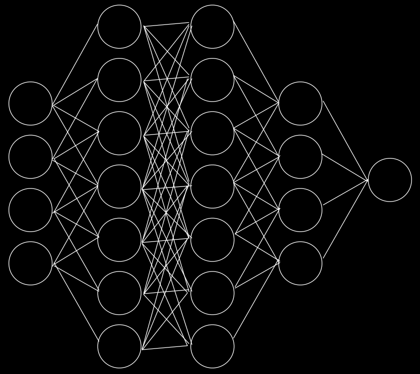
$\boldsymbol{x}_i =$ 

$m_i = 2\ \text{neV}$

$g_i = 3 \times \frac{10^{-11}}{\text{GeV}}$

example samples $(\boldsymbol{x}_i, m_i, g_i)$

Input

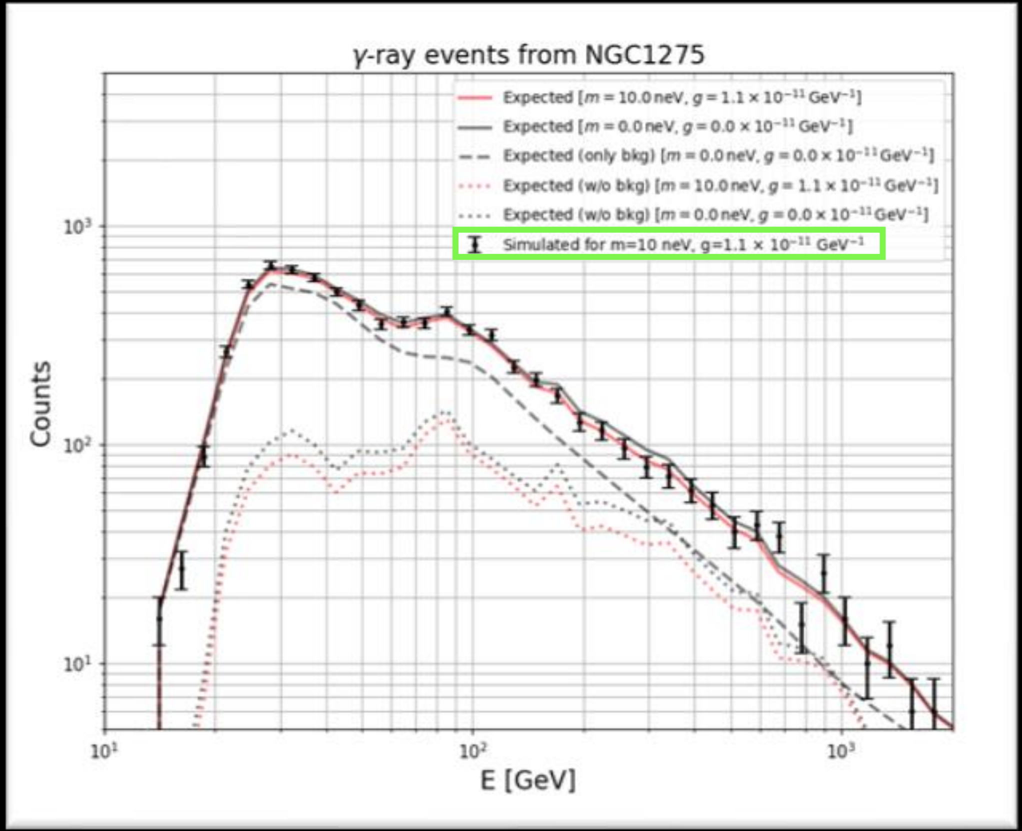


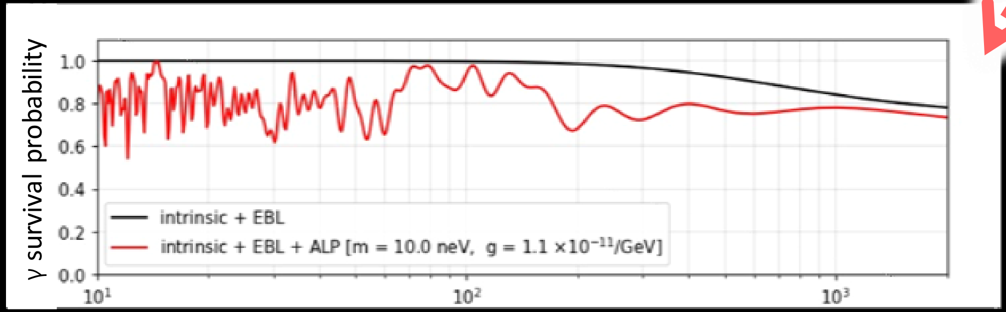
Neural network

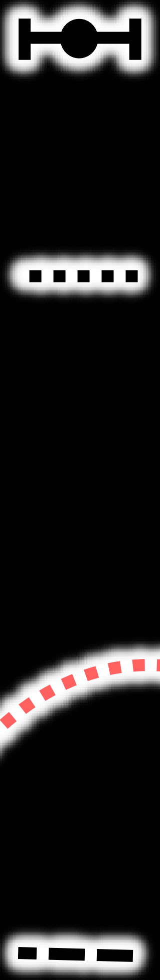
Modified
Output

Posterior = Modified Output × Prior

How our simulations are made







Simulated data

=

γ-rays ~ cut-off power law

$$\varphi(E) = \varphi_0 \left(\frac{E}{E_0}\right)^{\gamma} e^{-E/E_{cut}}$$

×

Instrument response
(sensitivity + energy dispersion)

Using the IRF (for CTA) prod3: South_z20_50h, together with gammapy 0.19 [3]

×

Absorption from EBL
(Extragalactic background light)

×

"Wiggles" from photon-ALP-oscillations

Using gammaALPs [2]

+

Cosmic ray background
(Irreducibly misidentified as γ-rays)

Using the IRF (for CTA) prod3: South_z20_50h, together with gammapy 0.19 [3]

Let's start with a simplified case study...

Model Parameters

- ALP mass, m
- ALP coupling to photons, g

} Parameters of interest

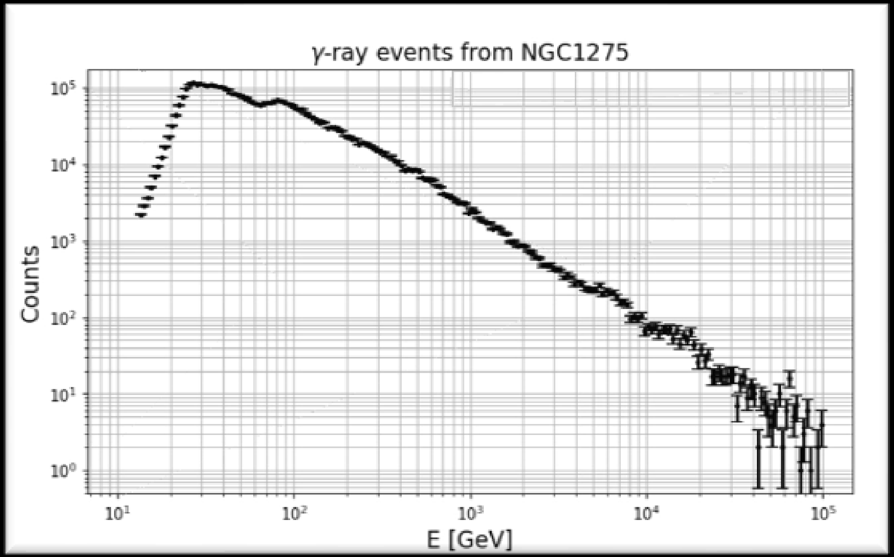
- NGC1275 intrinsic spectrum amplitude
- NGC1275 intrinsic spectral index
- NGC1275 intrinsic cut-off energy

} Nuisance parameters

(Taking into account only a few nuisance parameters while testing viability of the approach)

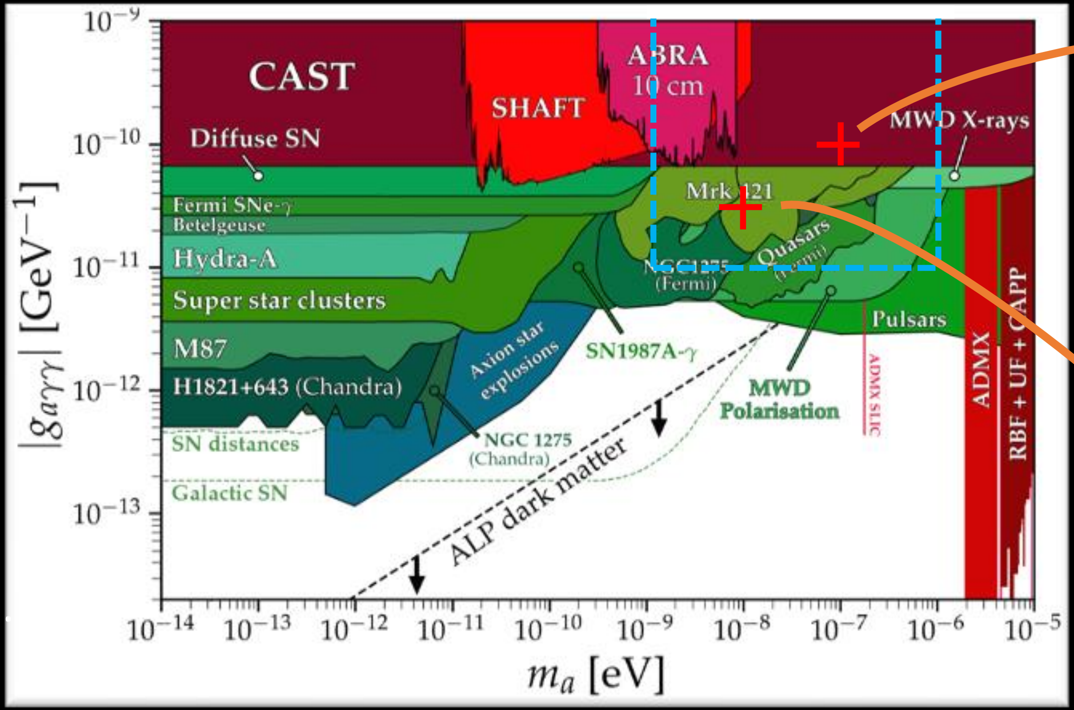
Modelling specifications

- ~ 1 000 000 simulations used in training
- Prior: uniform on log scale
- Assumed observation time = 50 hr
- 200 energy bins in range between 10 GeV and 100 TeV

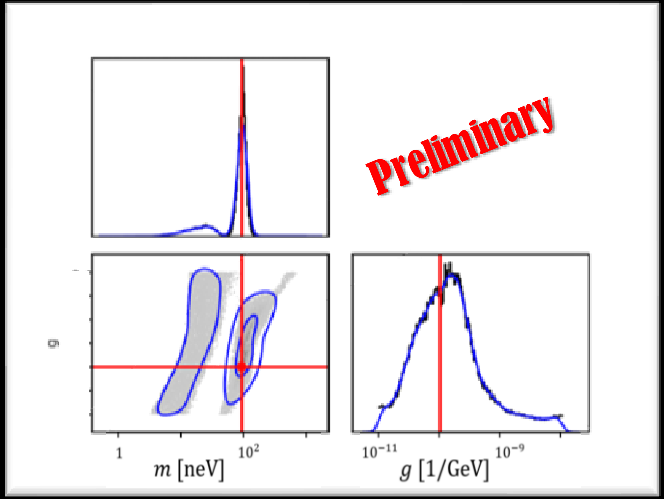


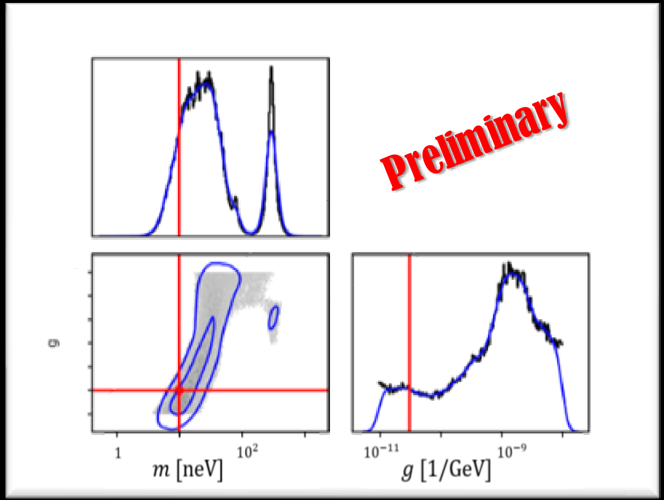
Simulated observation

First attempts at ALP-inference* show promise



Source: [6]





Preliminary indication:

The method seems to be sensitive in areas of parameter space where we expect sensitivity.

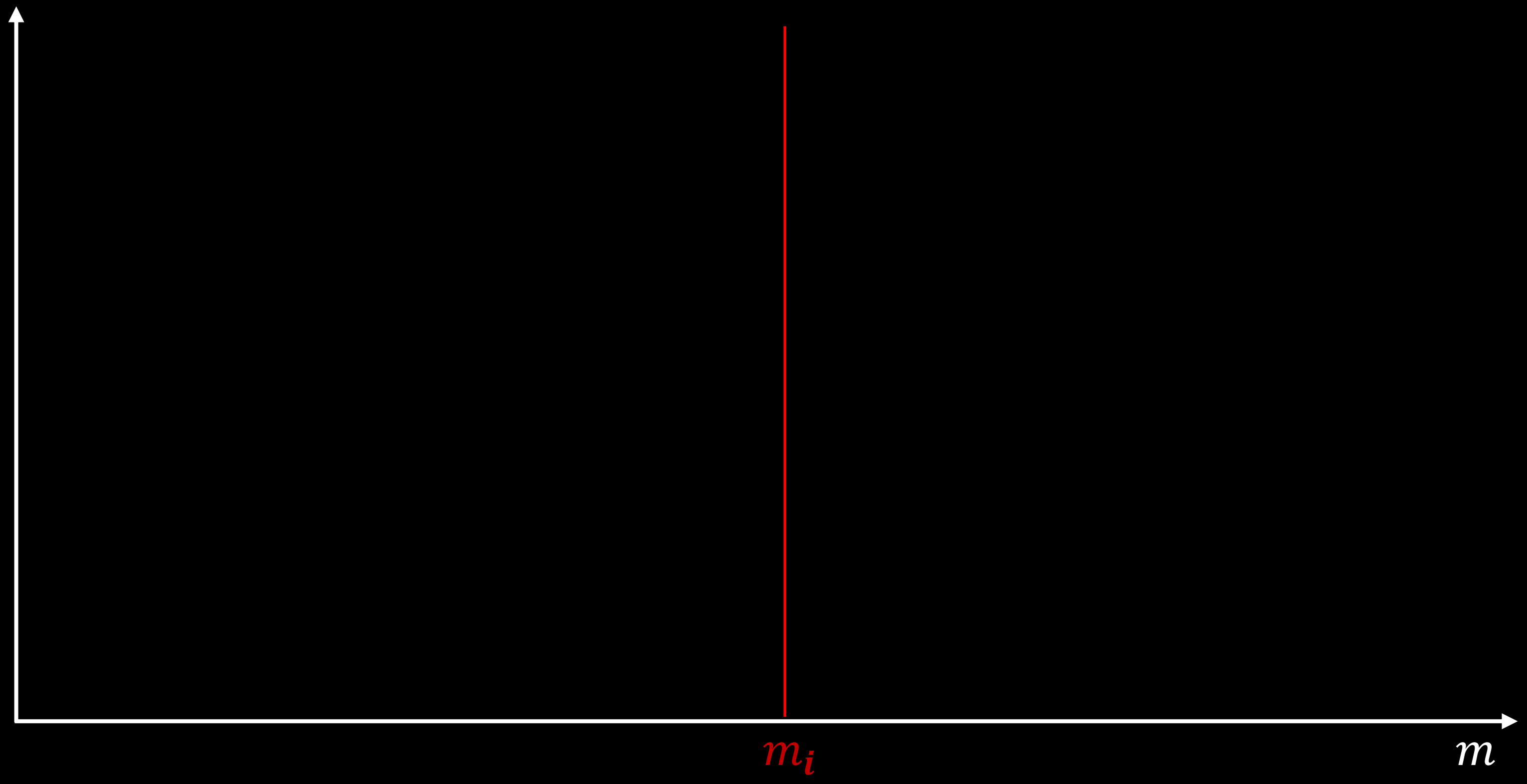
* We perform the NRE posterior estimation using the open-source python package SWYFT [4]

How can we trust the estimated posterior to accurately represent the true posterior?

- We can’t know if the NN minimized the loss function properly
- What number of simulations is close enough to infinity for the LR trick to be valid?

Analogous for multidimensional posteriors

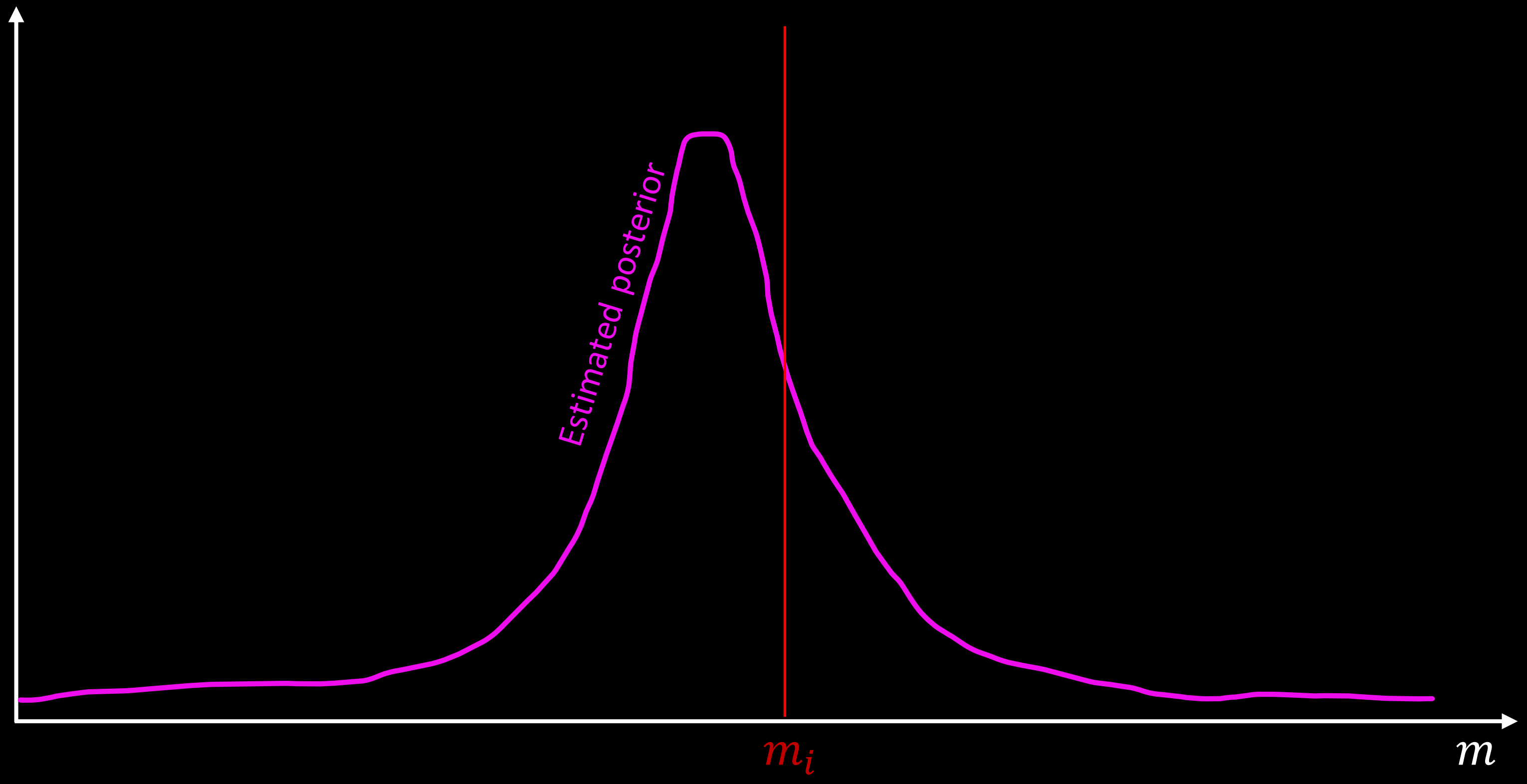
Validation of estimated posteriors (for one parameter)



1. Draw a sample m_i from the prior, and simulate an observation x_i from it.

Analogous for multidimensional posteriors

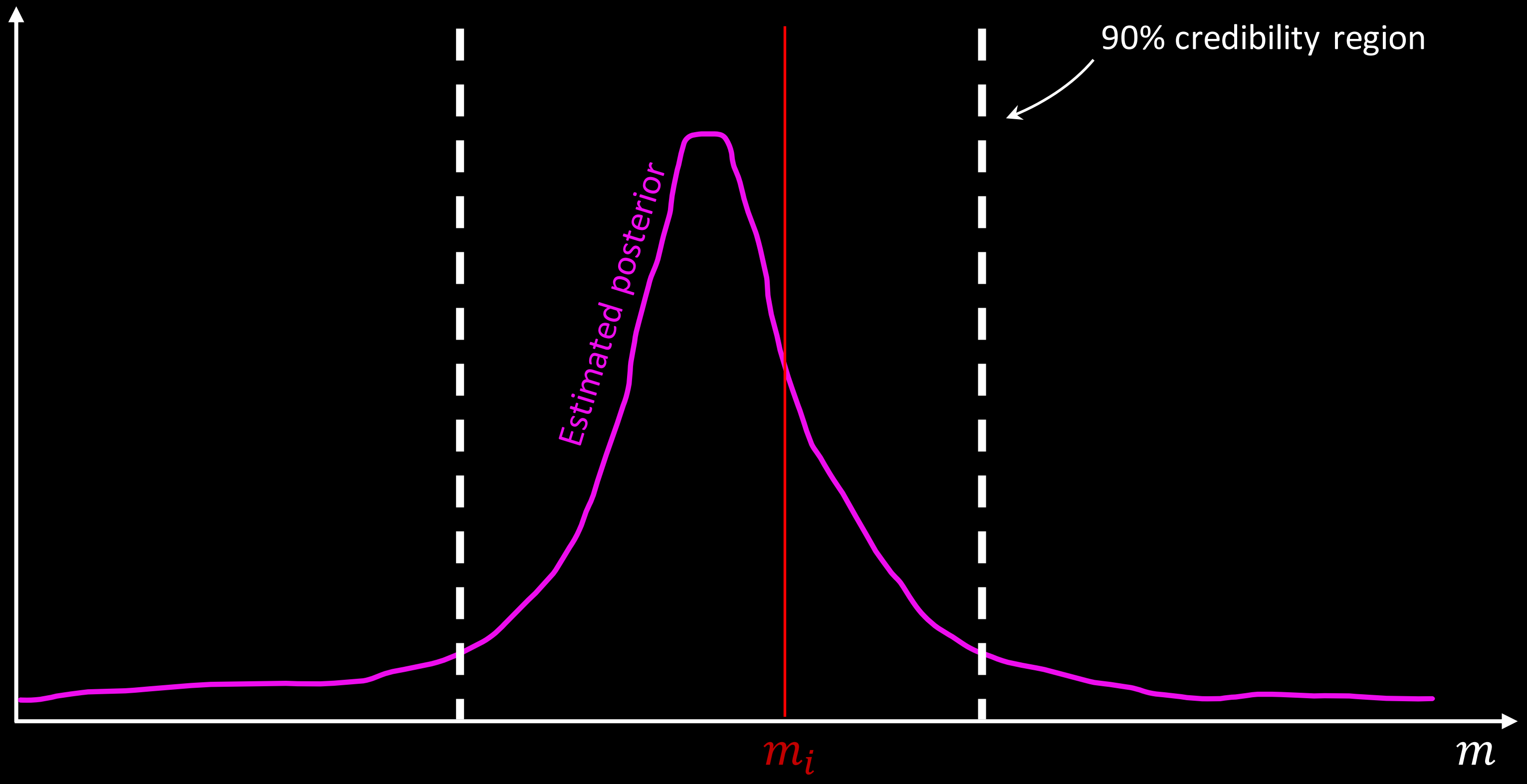
Validation of estimated posteriors (for one parameter)



2. Generate the estimated posterior from $\boldsymbol{x_i}$.

Analogous for multidimensional posteriors

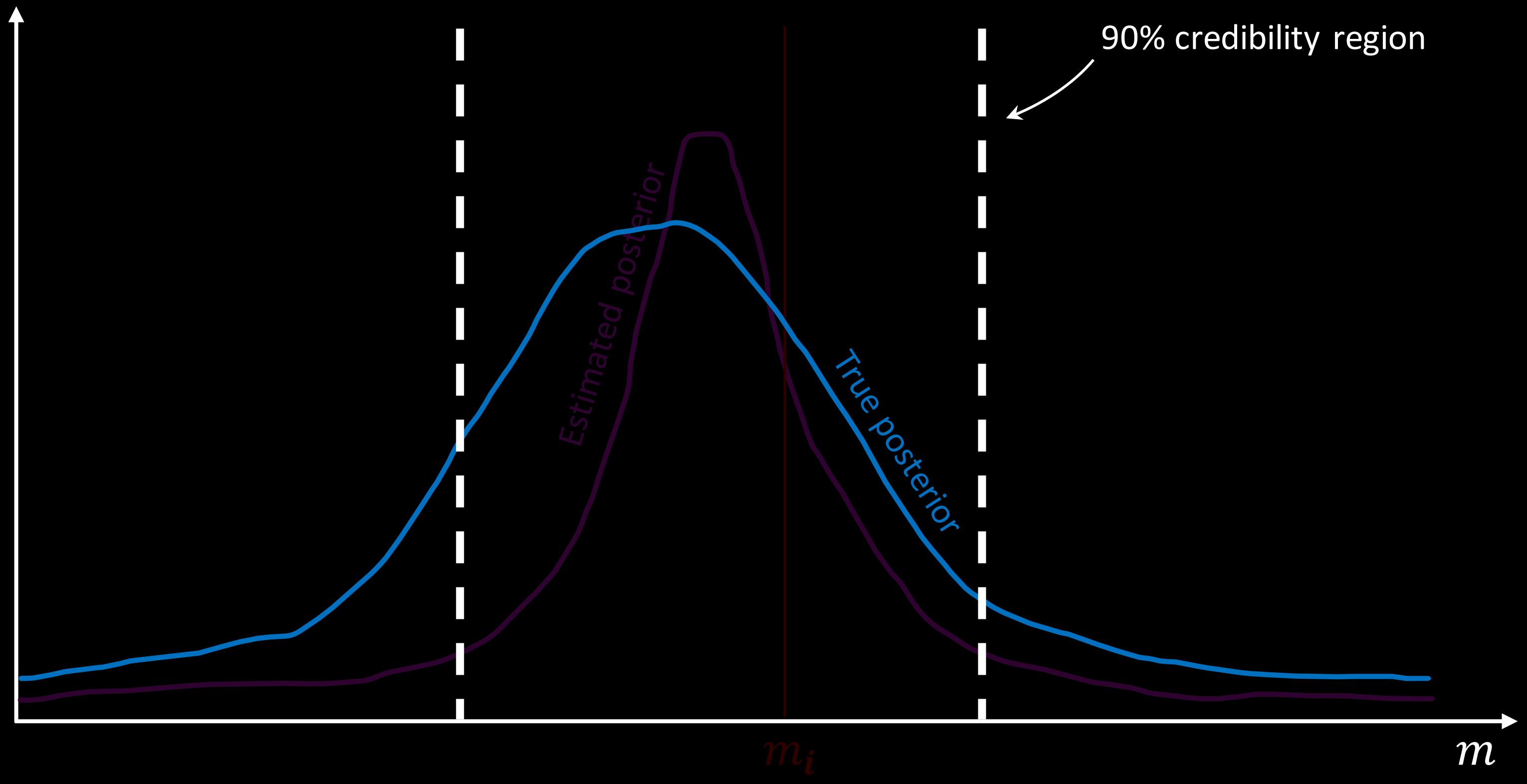
Validation of estimated posteriors (for one parameter)



3. Define a credibility region of chosen credibility level α (0.9 in the example), such that the volume of the estimate posterior in that region is α.

Analogous for multidimensional posteriors

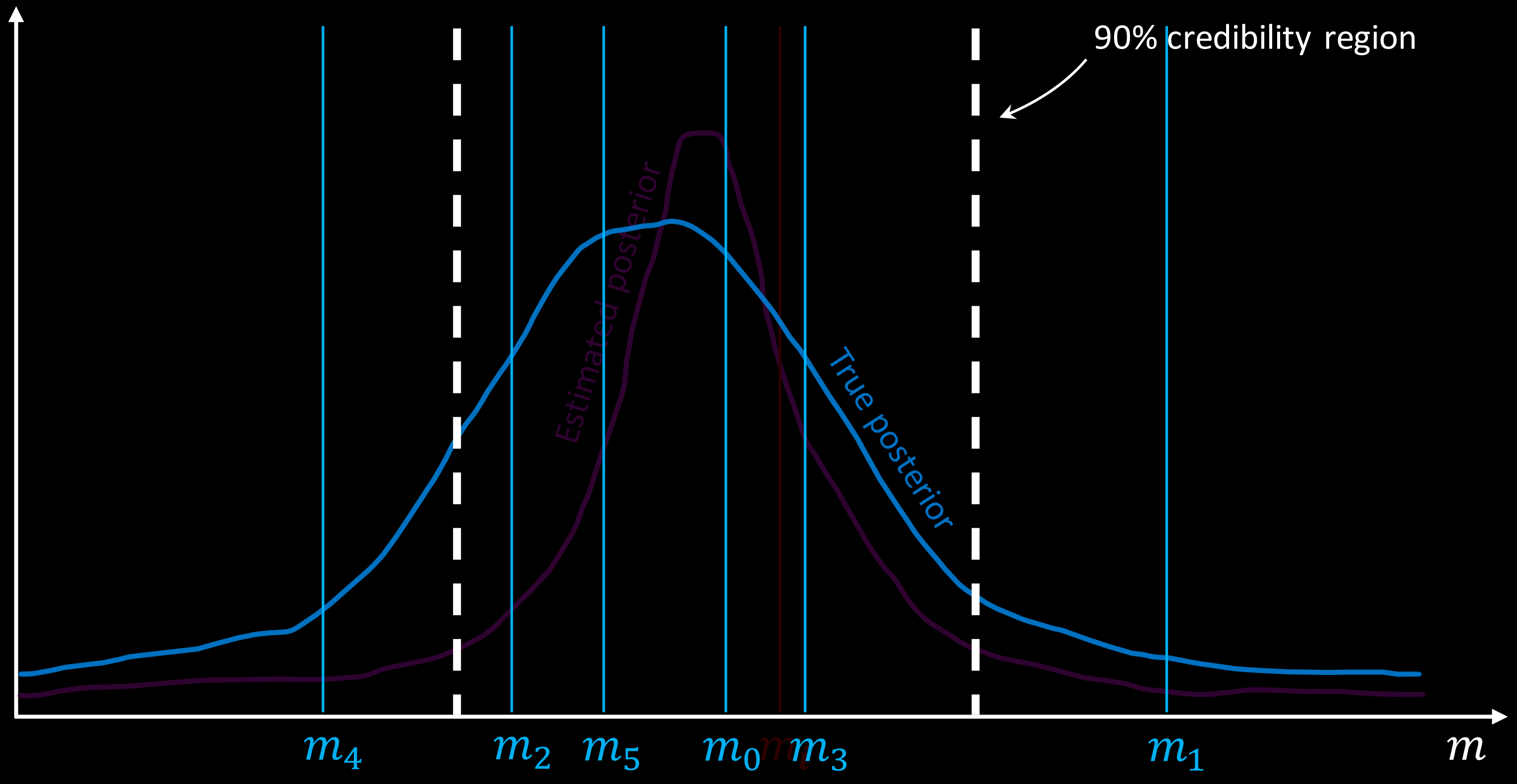
Validation of estimated posteriors (for one parameter)



4. Imagine the true posterior: the one we would get by using Bayes' theorem.

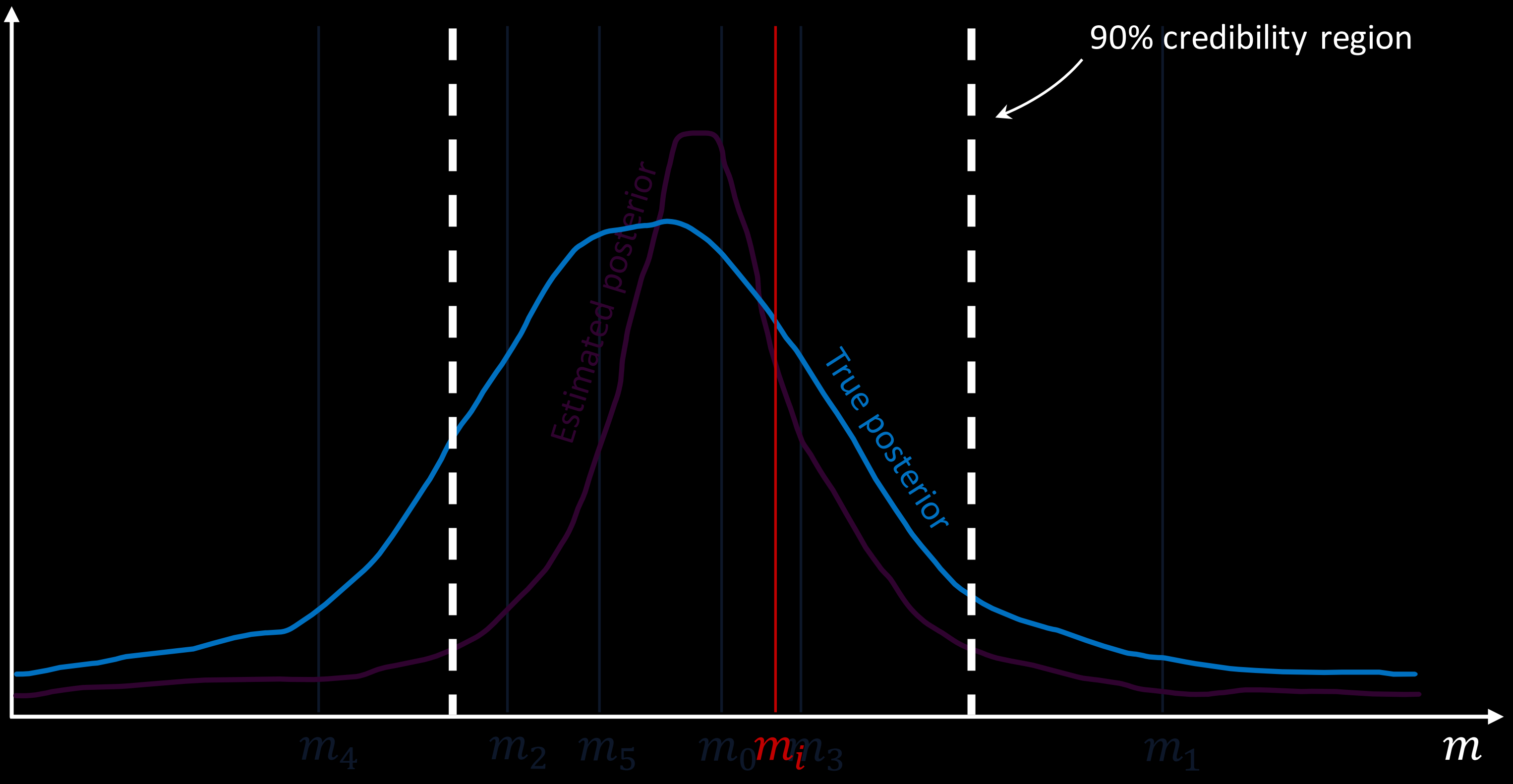
Analogous for multidimensional posteriors

Validation of estimated posteriors (for one parameter)



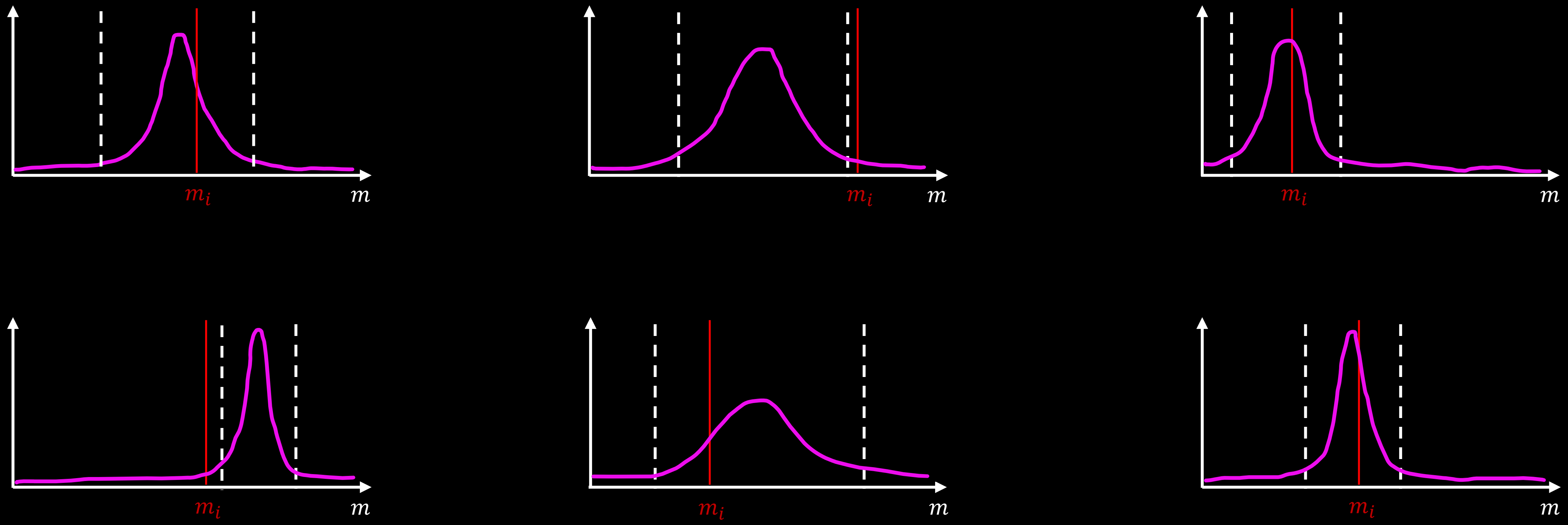
5. If the estimated posterior is valid, then taking samples from the true posterior should result in a proportion α of samples inside the credibility region.

Validation of estimated posteriors (for one parameter)



6. We cannot sample the, true posterior, as we cannot compute it. However, the original sample m_i can be considered to be a single sample of the true posterior.

Validation of estimated posteriors (for one parameter)

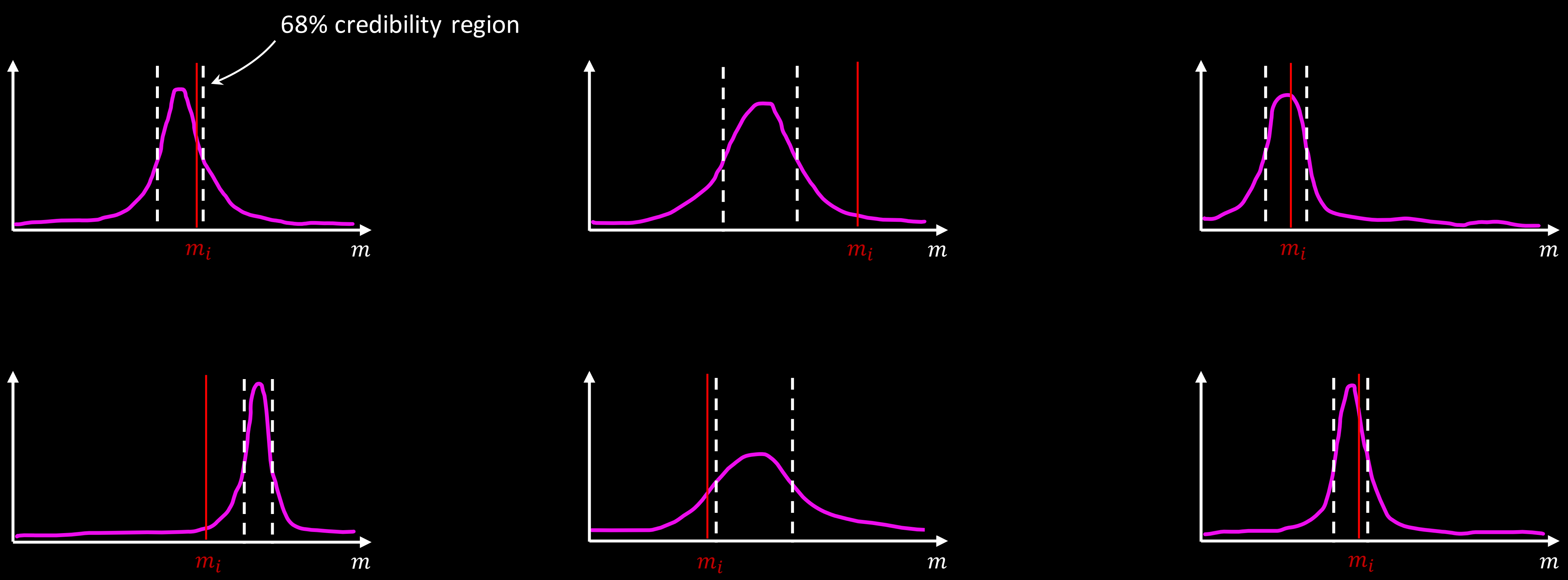


7. We can conduct this experiment several times and count the proportion of cases where m_i happens to be inside the credibility region. This proportion is called the *Expected Coverage Probability (ECP)*.

We expect the ECP to be equal to α.

NOTE that all the posteriors are estimated using the <u>same</u> (previously trained) NN!

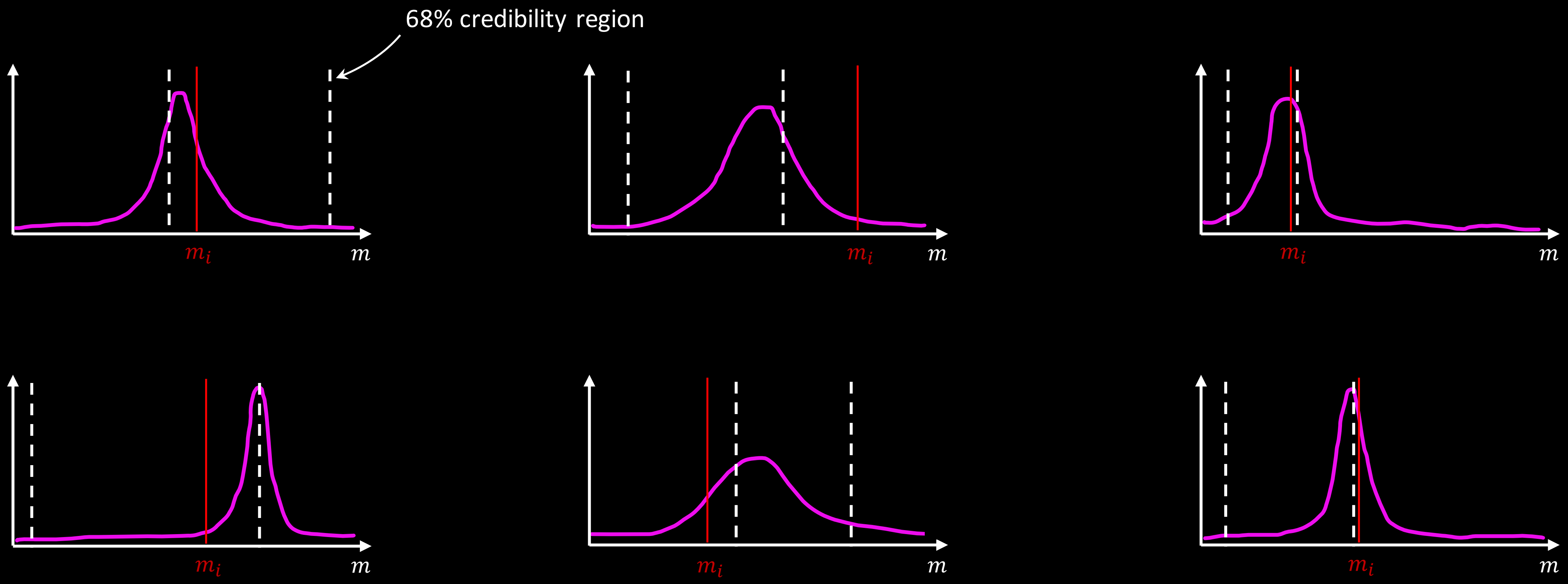
Validation of estimated posteriors (for one parameter)



8. Furthermore, we expect the ECP to be equal to α **for any chosen credibility α.**

NOTE that all the posteriors are estimated using the <u>same</u> (previously trained) NN!

Validation of estimated posteriors (for one parameter)



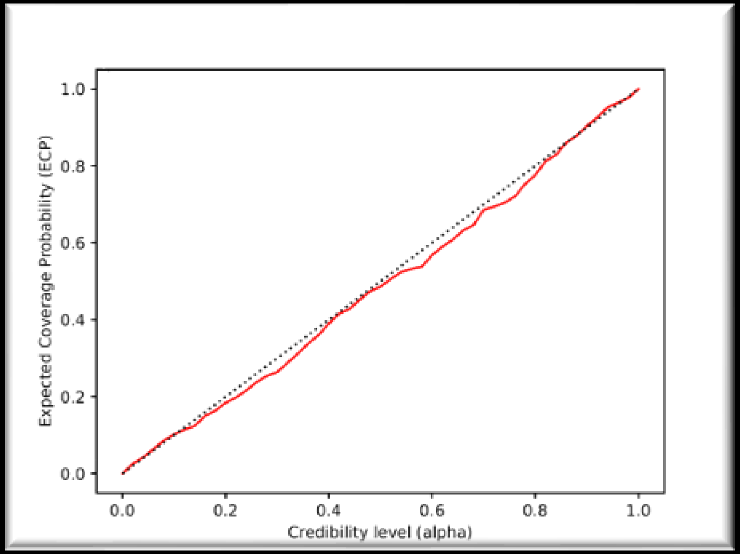
9. According to Lemos et. al [3], if ECP = α for all α, even **when the credibility regions are randomly centered, then the estimated posterior is identical to the true posterior.**

NOTE that all the posteriors are estimated using the <u>same</u> (previously trained) NN!

Expected coverage testing indicates we are on the right track

Validation of *(mass, coupling)*

ECP



α

ECP $\approx \alpha$ for all α

... but maybe a slight bias towards the lower right?

… but we still got a (little) ways to go

1. We may still not be training our NNs to their full potential
   - Related to technical (hardware) limitations → hopefully soon to be solved.
   - But Validation indicates we are on the right track!
2. We want to take into account several more nuisance parameters
3. We need to switch to more state-of-the-art instrument response functions
4. We need to explore a larger (more interesting) region of parameter space

Good news:

Steps 2 and 4 don't require any new implementations!

(just need to "press play")

Summary

- Axion-like particles (ALP) are popular beyond-SM and DM candidates, with potential – but difficult – detectability in cosmic gamma-rays.
- Neural Ratio Estimation (NRE) may allow us to produce accurate ALP-limits that would otherwise be overconfident.
- Recent developments are making it possible to assess the reliability of NRE, thus making it a serious contender to conventional inference techniques.
- Our preliminary results indicate that NRE is a viable method for our physics case
- Our analysis will likely improve significantly soon, given more machine learning-friendly technical resources.

References

[1] G. Irastorza and J. Redondo, "New experimental approaches in the search for axion-like particles," Progress in Particle and Nuclear Physics, vol. 102, pp. 89–159, 2018.

[2] https://github.com/me-manu/gammaALPs

[3] https://docs.gammapy.org/0.19/

[4] https://github.com/undark-lab/swyft

[5] P. Lemos, A. Coogan, Y. Hezaveh, and L. Perreault-Levasseur, "Sampling-based accuracy testing of posterior estimators for general inference," in Proceedings of the 40th International Conference on Machine Learning (A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, eds.), vol. 202 of Proceedings of Machine Learning Research, pp. 19256–19273, PMLR, 23–29 Jul 2023

[6] https://github.com/cajohare/AxionLimits/blob/master/docs/ap.md