An implementation of Neural Simulation-Based Inference for Parameter Estimation in ATLAS

CHEP

23 October 2024

Aishik Ghosh, on behalf of the ATLAS Collaboration

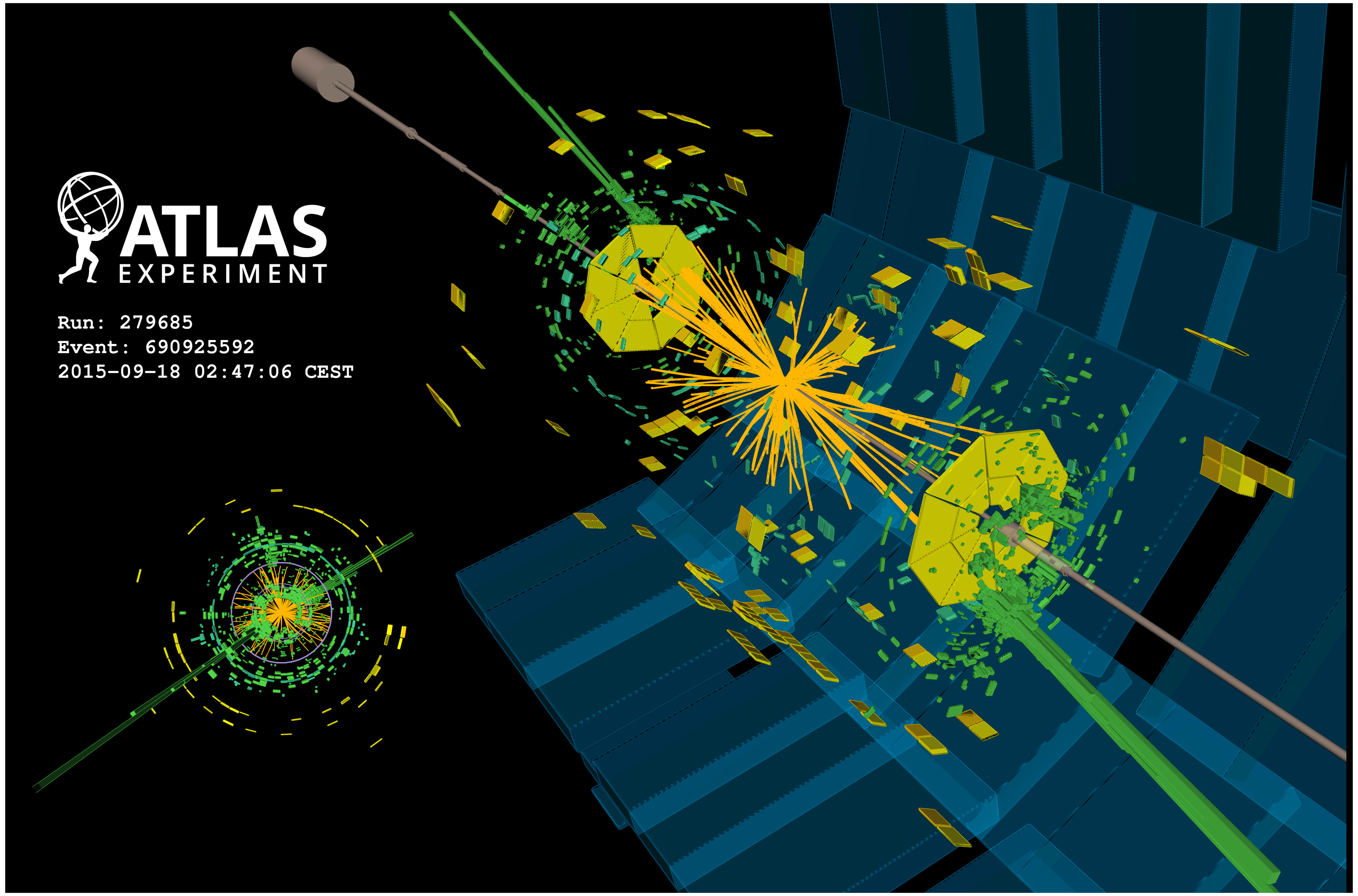
The motivation for high dimensional statistical inference

(Rather than using a low-dimensional histogram for statistical analysis)

Typical LHC Workflow

- Detector has O(100 million) sensors
- Can't build 100M dimensional histogram

- Reconstruction pipeline, event selection
- Design sensitive one-dimensional observable

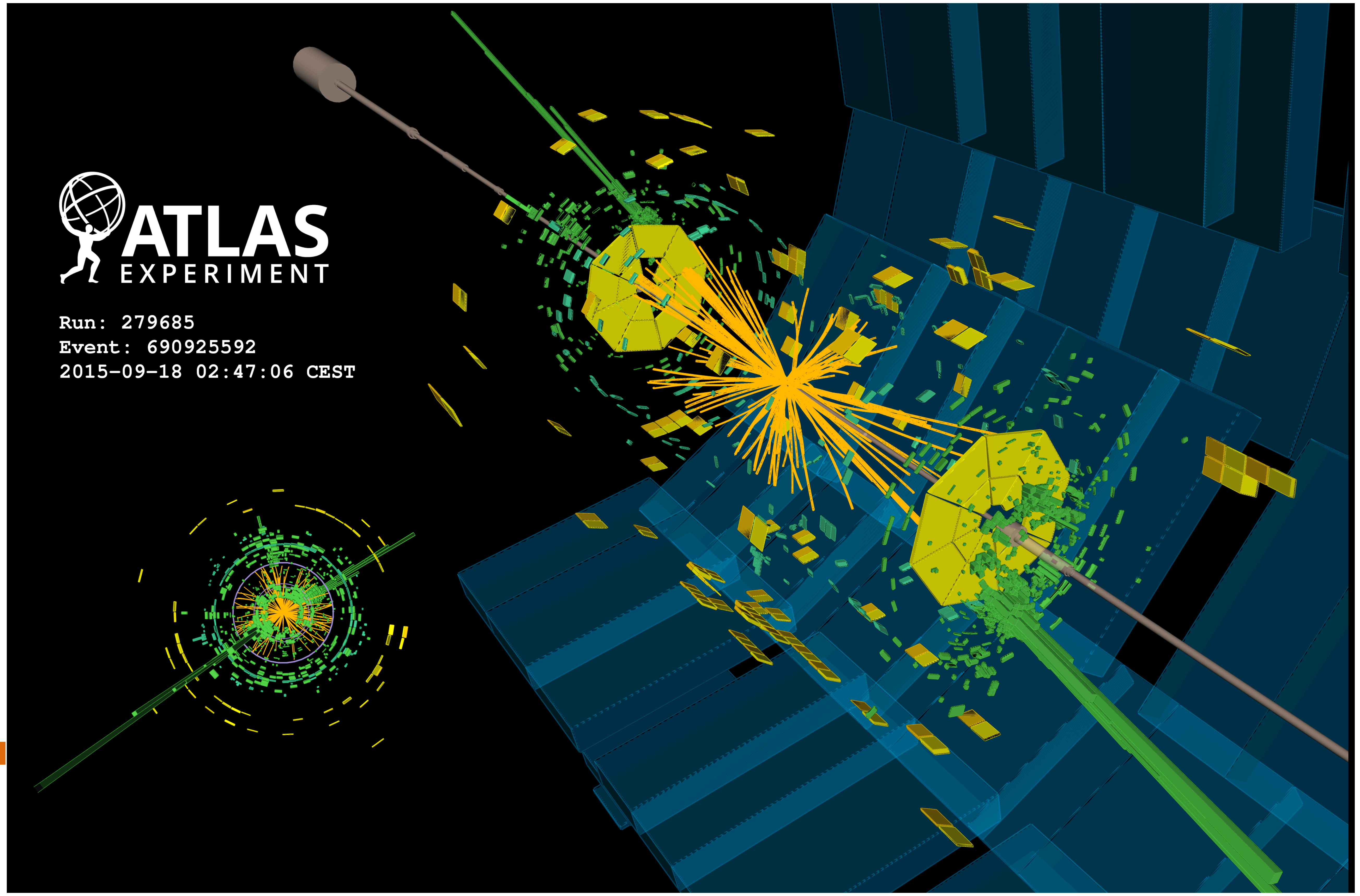


Typical LHC Workflow

- Detector has O(100 million) sensors
- Can't build 100M dimensional histogram

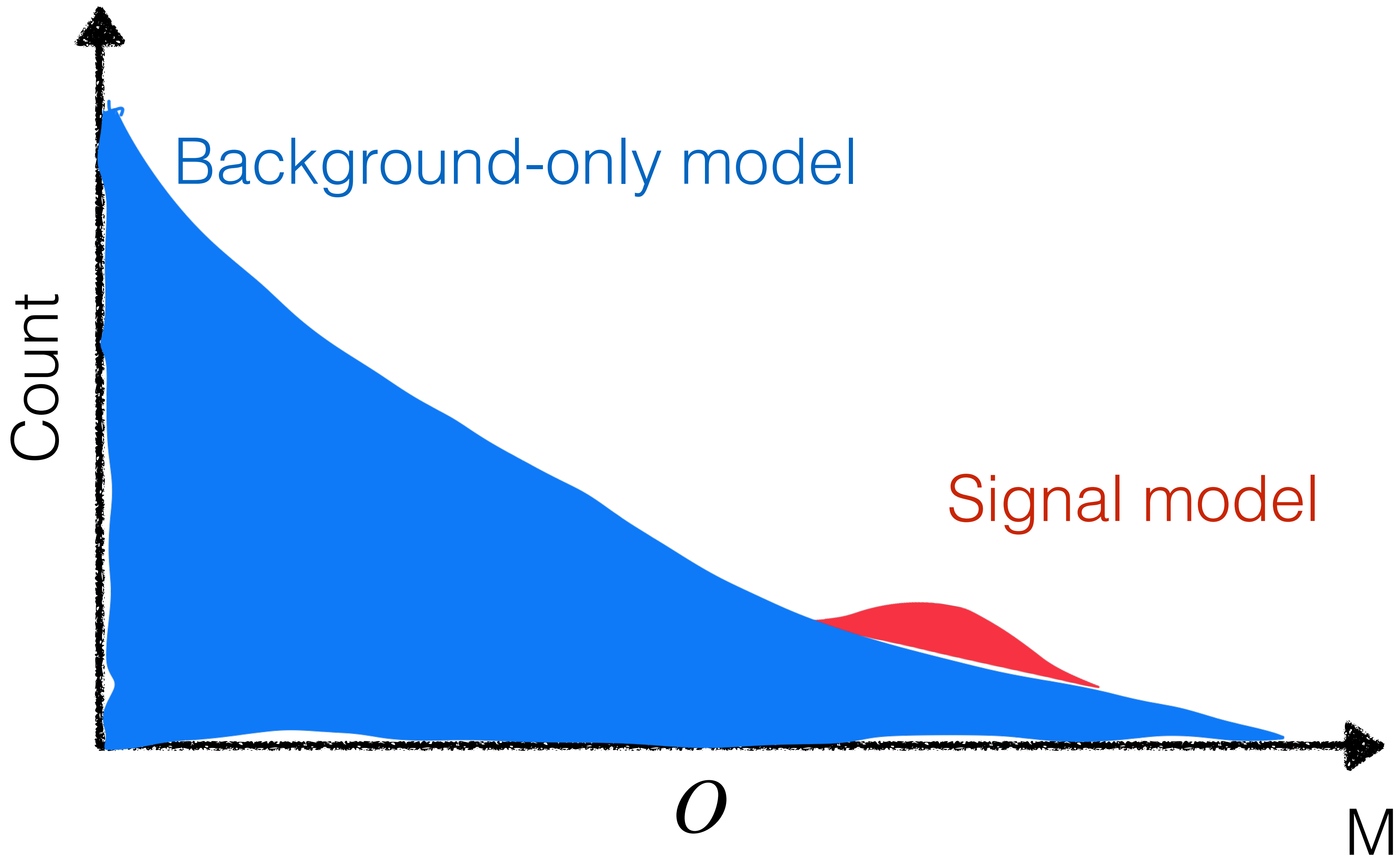
- Reconstruction pipeline, event selection
- Design sensitive one-dimensional observable

1 number

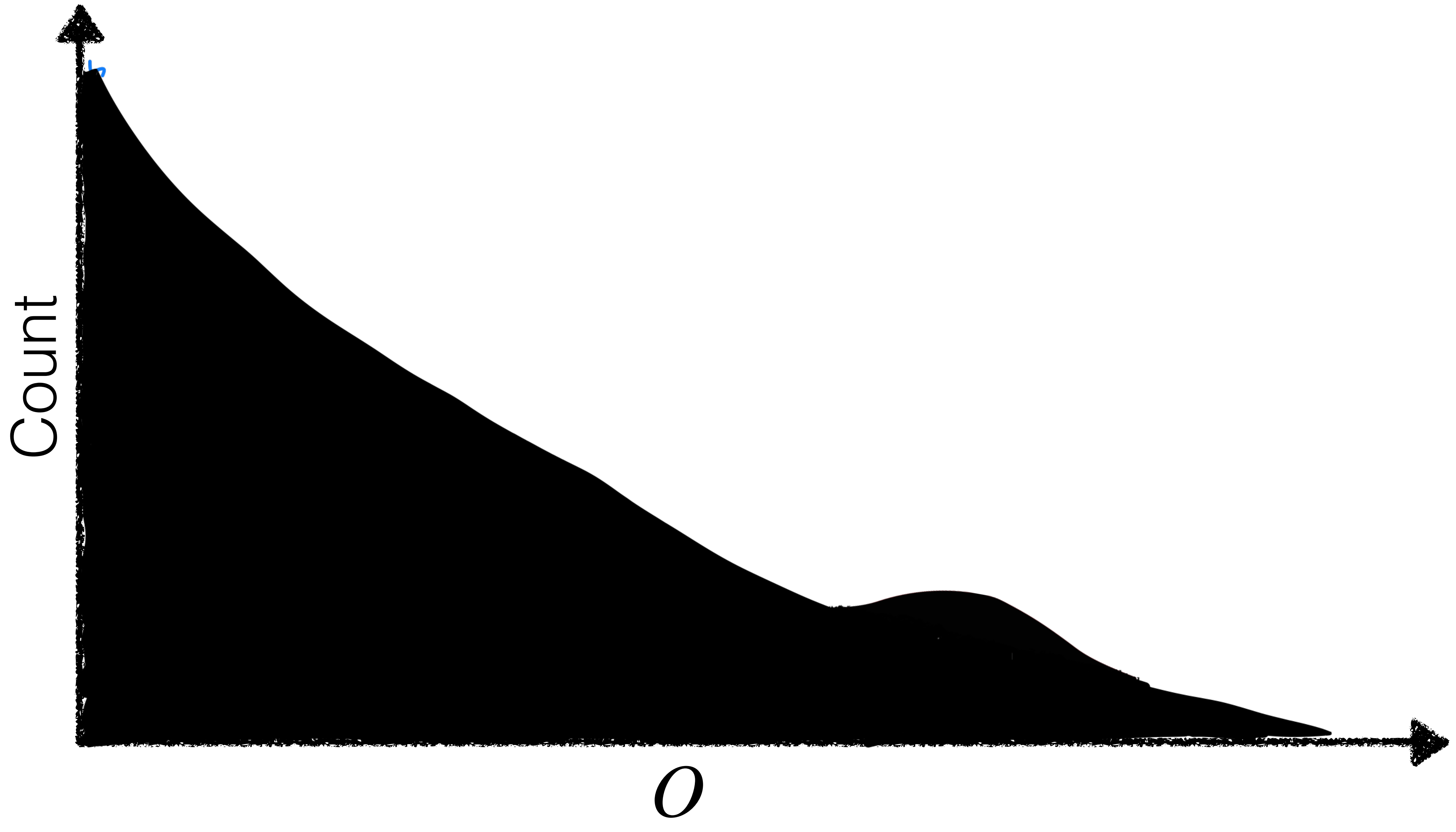


Probability Density Estimation: What we're used to doing..

Theory Predictions



Data

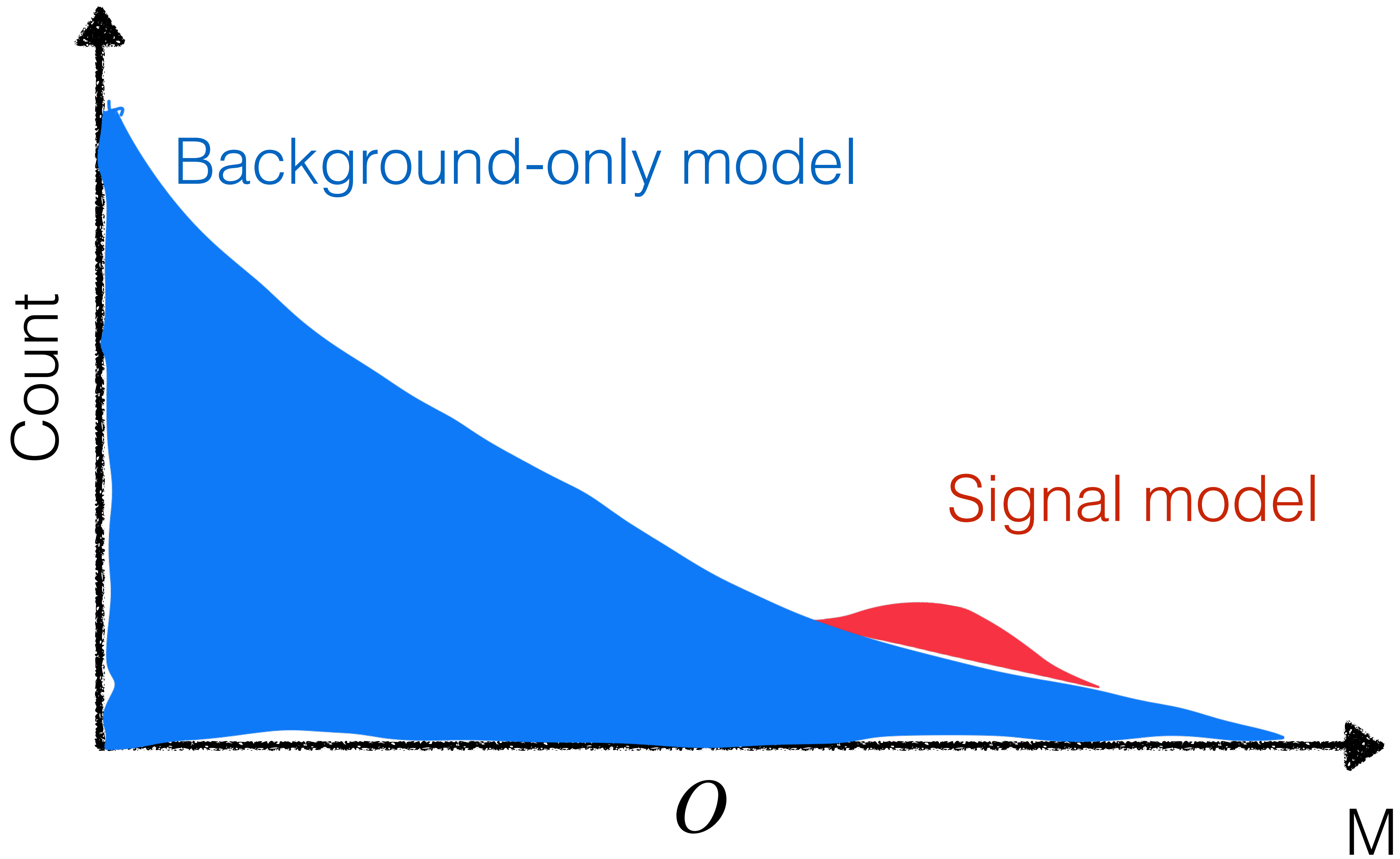


Measure signal strength μ

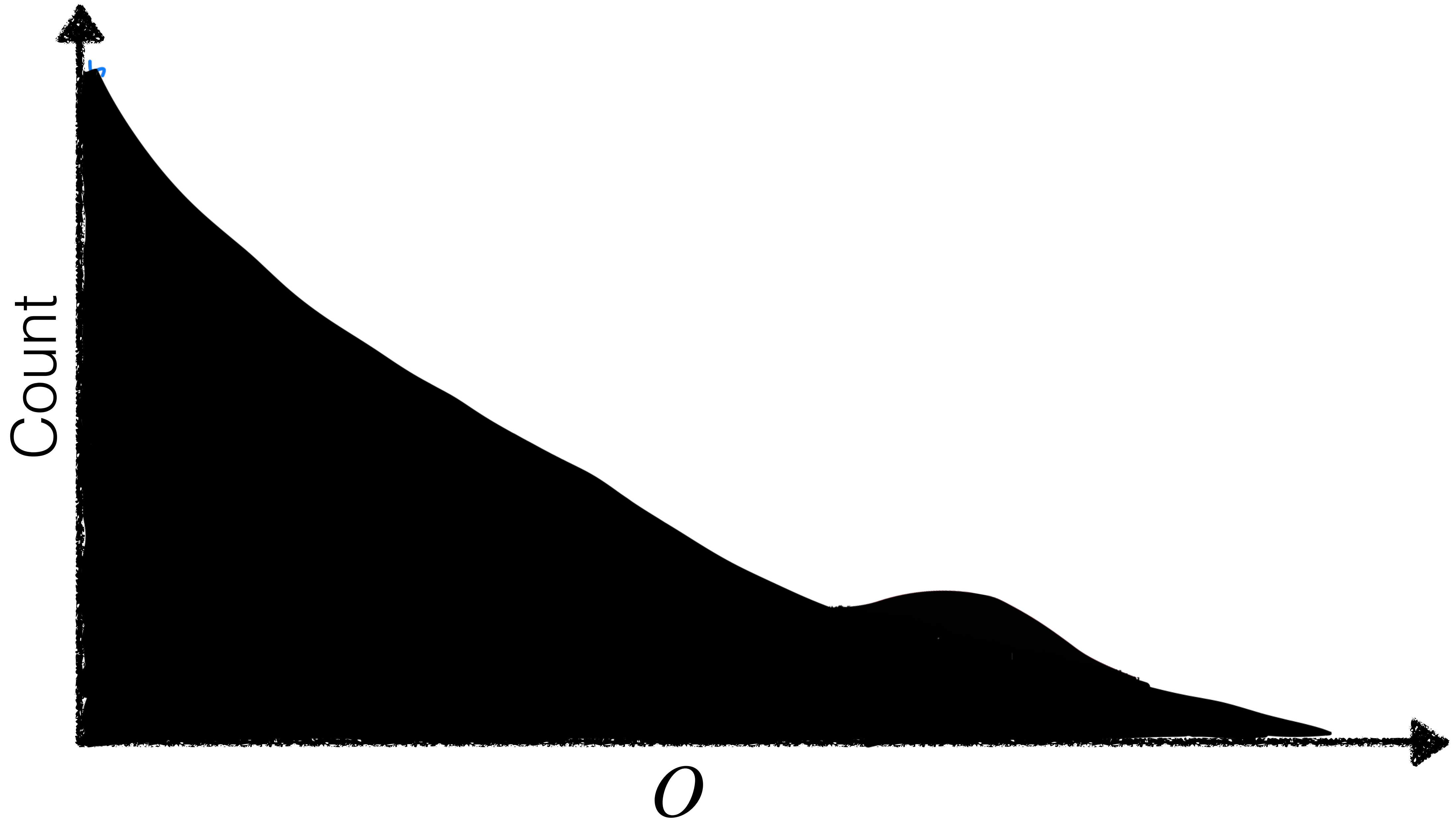
With histograms we can ask "Given the data, what is the likelihood of $\mu = 1$ hypothesis vs $\mu = 2$ hypothesis?"

Probability Density Estimation: What we're used to doing..

Theory Predictions



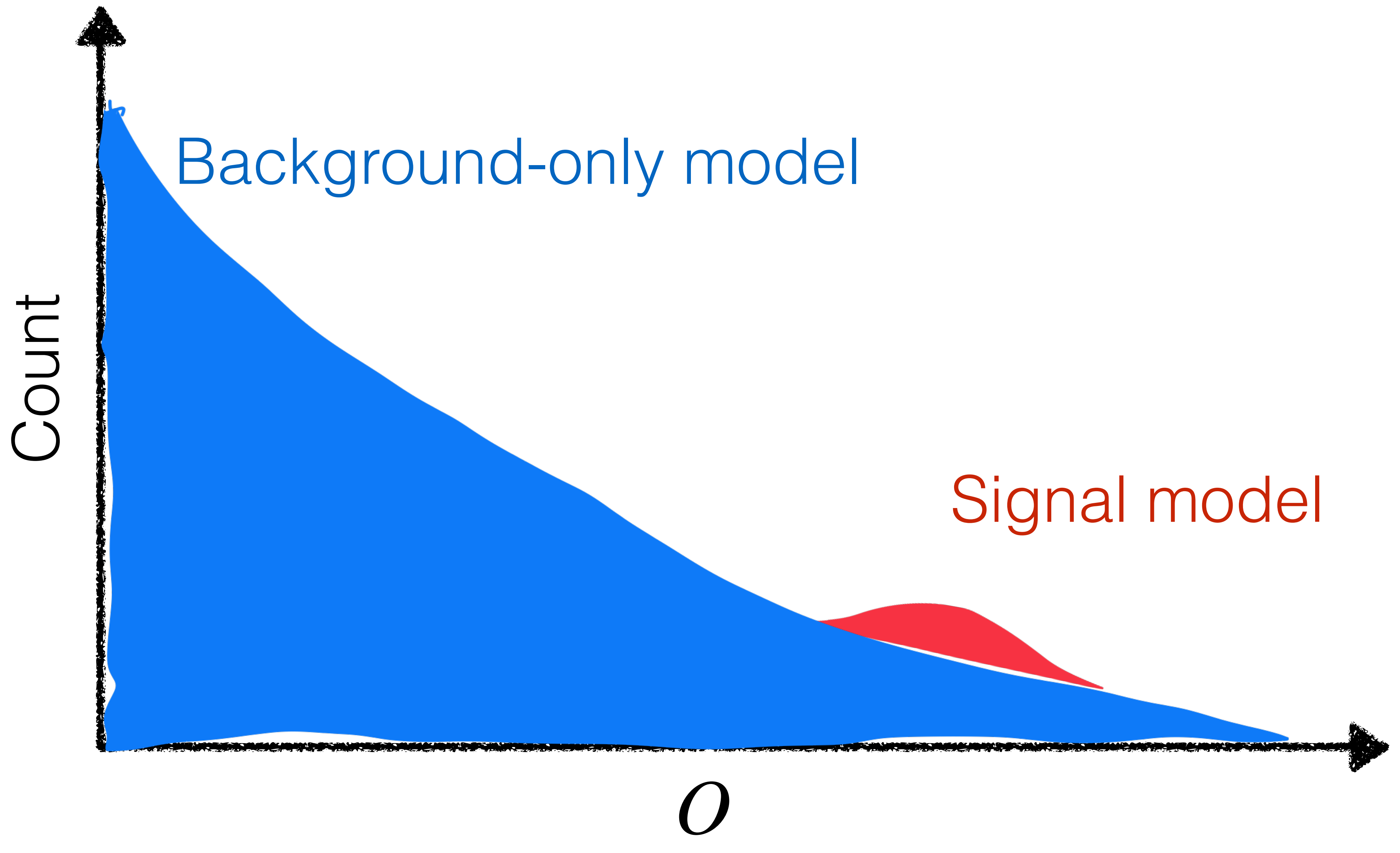
Data



Measure signal strength μ

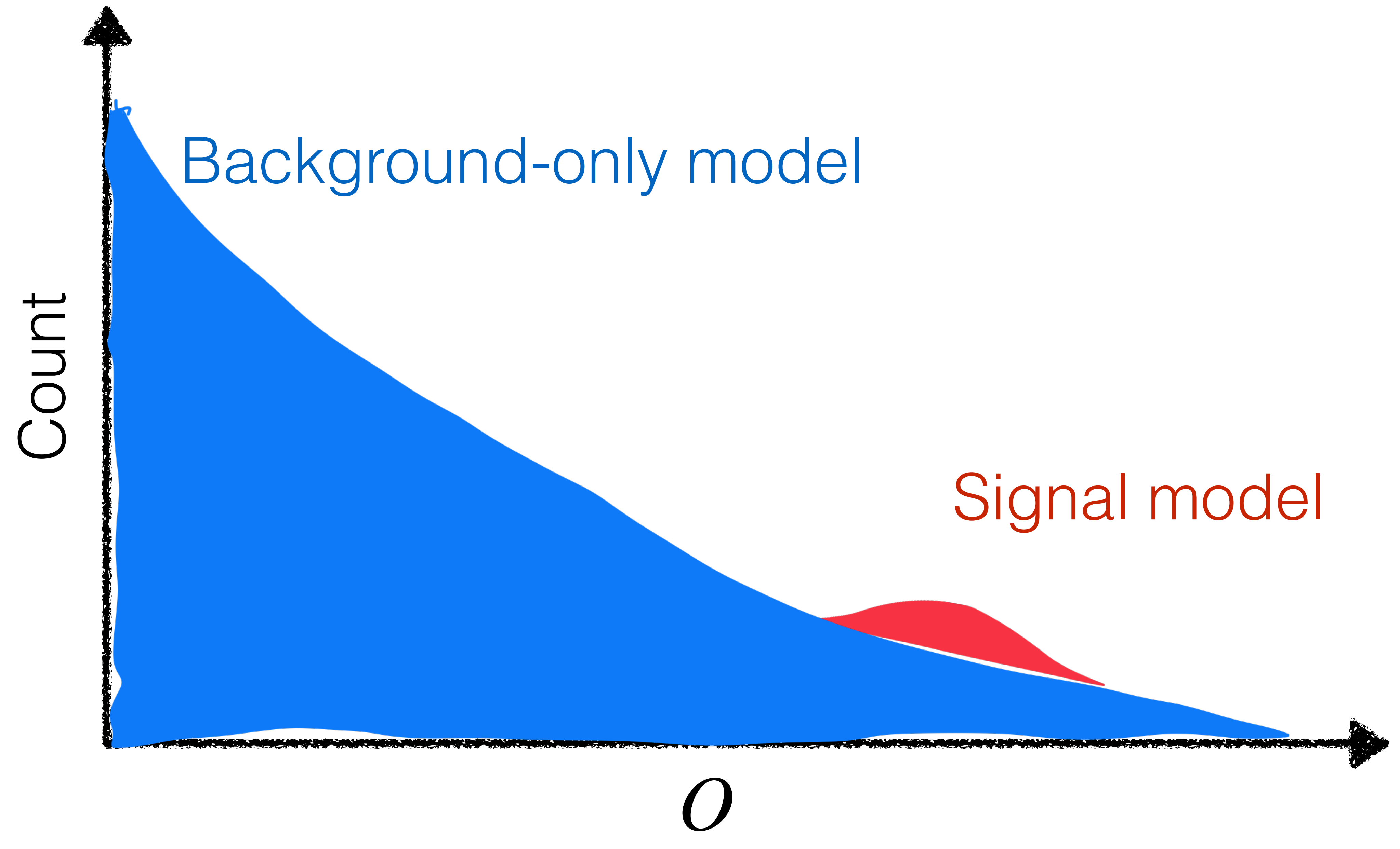
With histograms we can ask "Given the data, what is the likelihood of $\mu = 1$ hypothesis vs $\mu = 2$ hypothesis?"

New challenge: Non-linear changes in kinematics (w.r.t. parameter of interest)

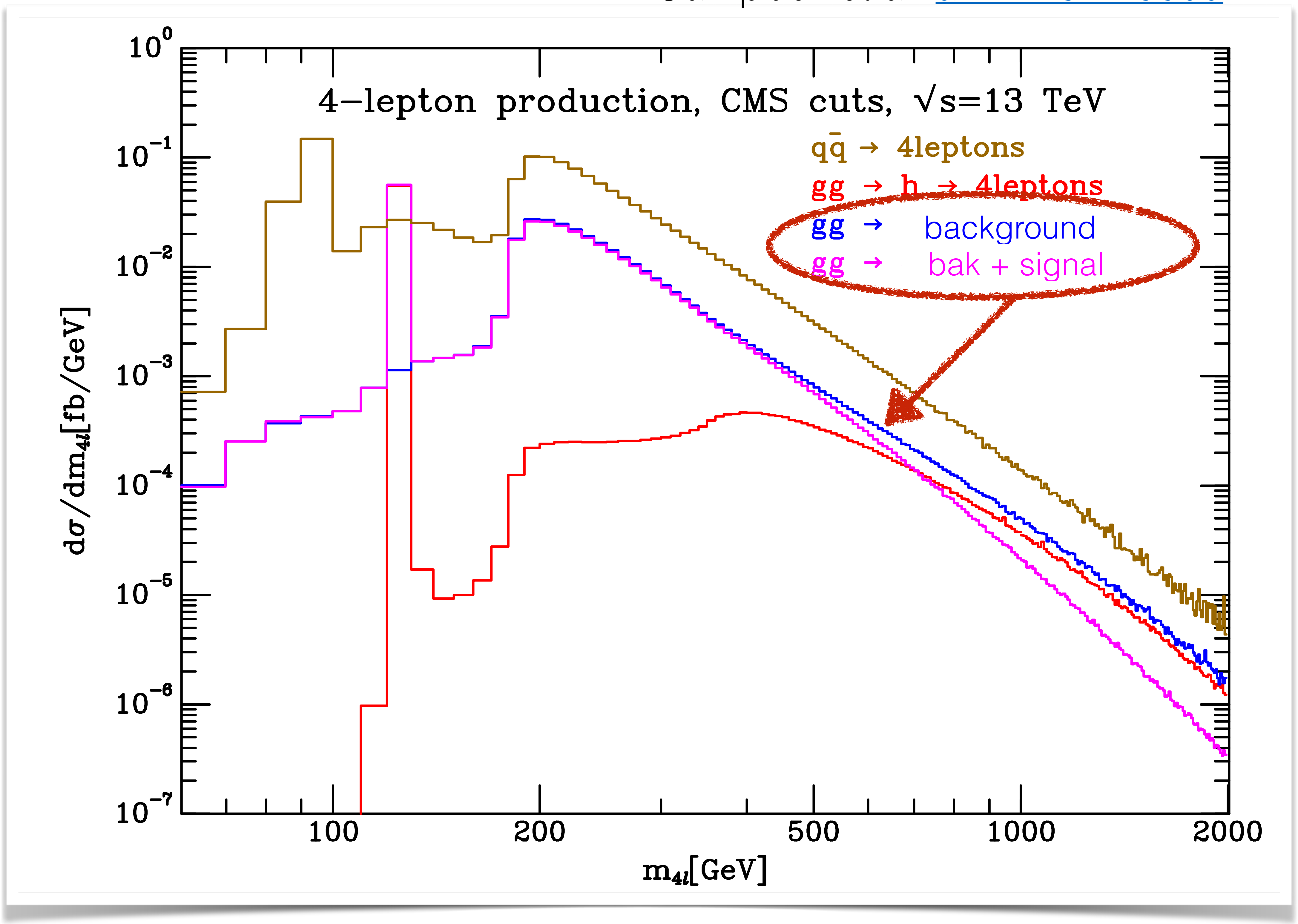


A histogram of any single observable is no longer optimal (see Ghosh et al: hal-02971995(p172)), but neural networks estimate high-dimensional likelihood ratios (see Cranmer et al: arXiv:1506.02169) !

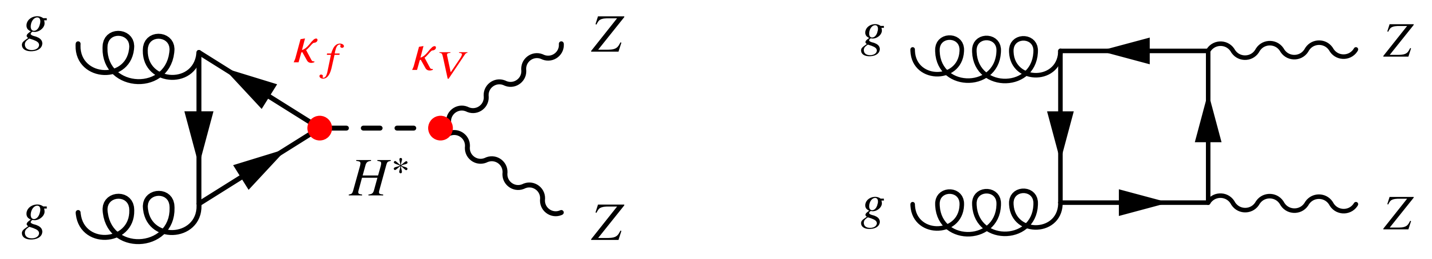
New challenge: Non-linear changes in kinematics (w.r.t. parameter of interest)



Campbell et al: [arXiv:1311.3589]



Quantum interference:

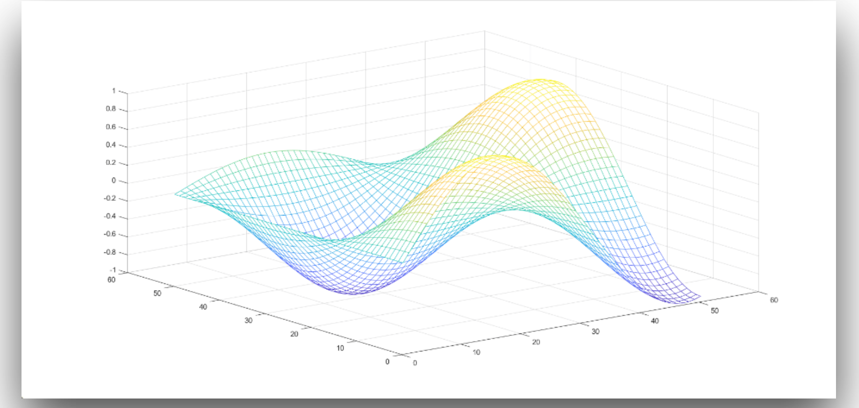


A histogram of any single observable is no longer optimal (see Ghosh et al: [hal-02971995(p172)]), but neural networks estimate high-dimensional likelihood ratios (see Cranmer et al: [arXiv:1506.02169]) !

Cranmer et al: arXiv:1506.02169

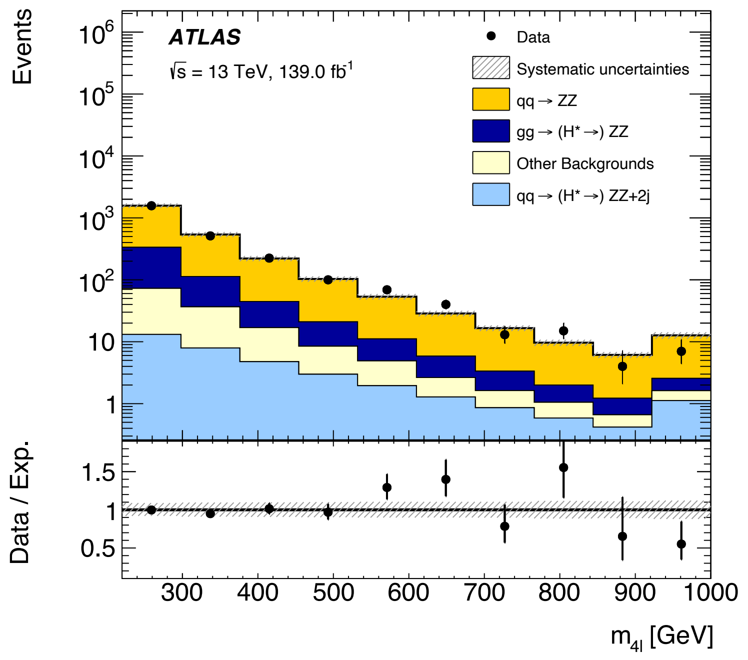
"Neural Simulation-Based Inference"

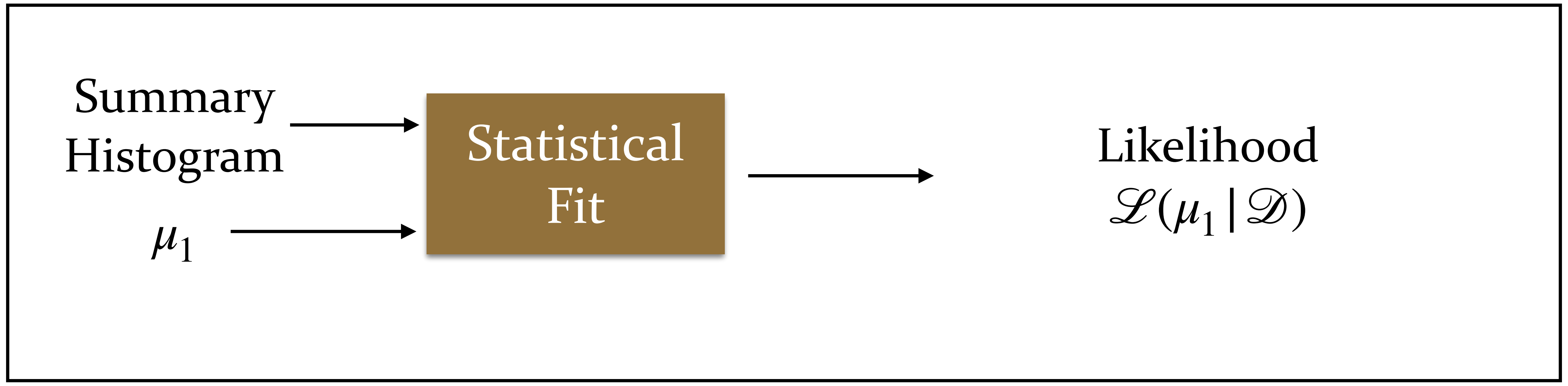
Traditional framework:



High-dim data

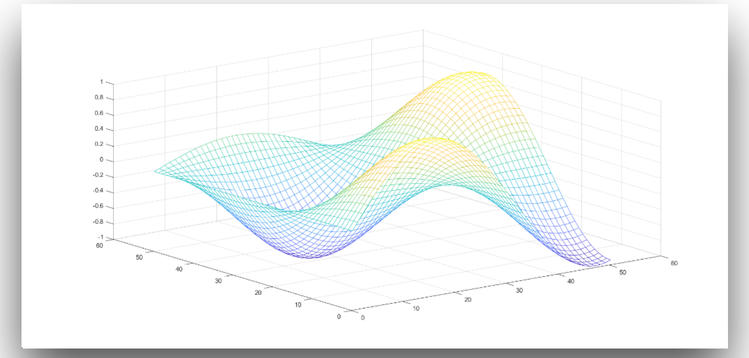
Summarisation to histogram



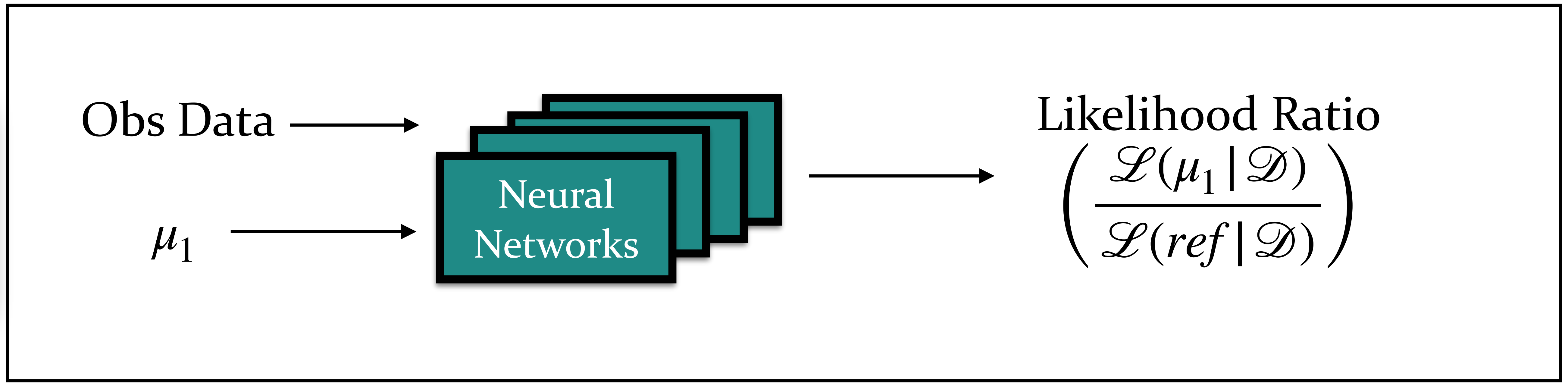


μ is now arbitrary parameter of interest(s)

The neural inference framework:

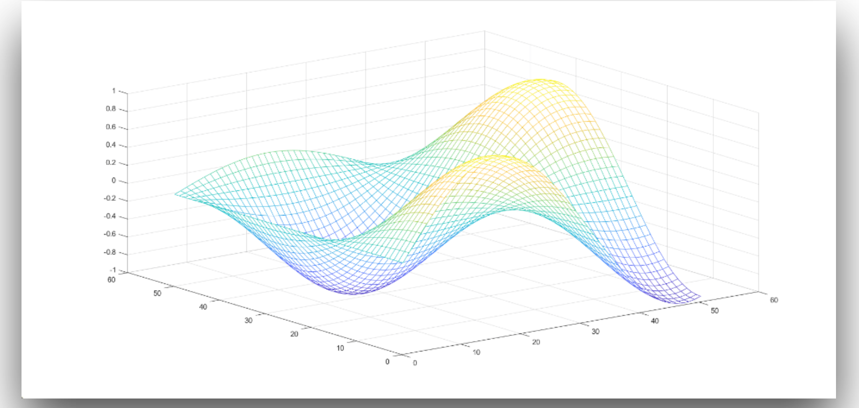


High-dim data



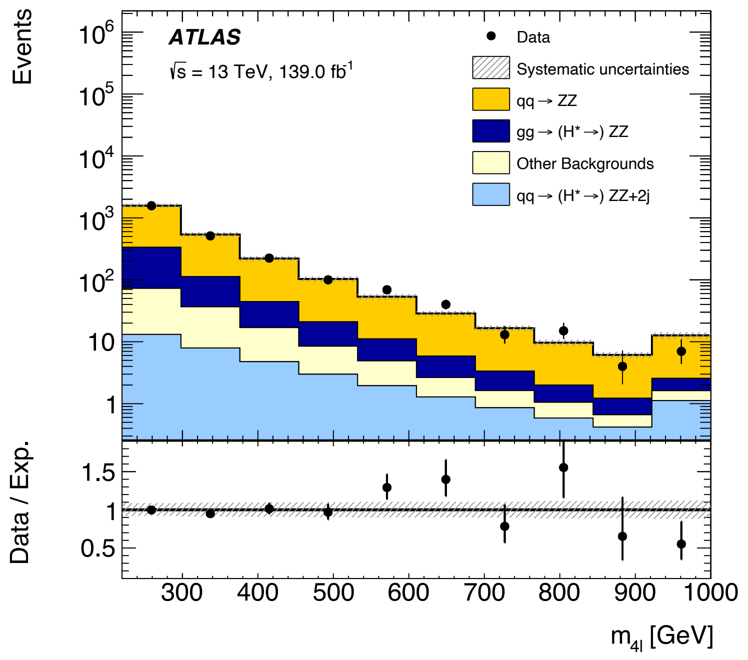
"Neural Simulation-Based Inference"

Traditional framework:



High-dim data

Summarisation to histogram



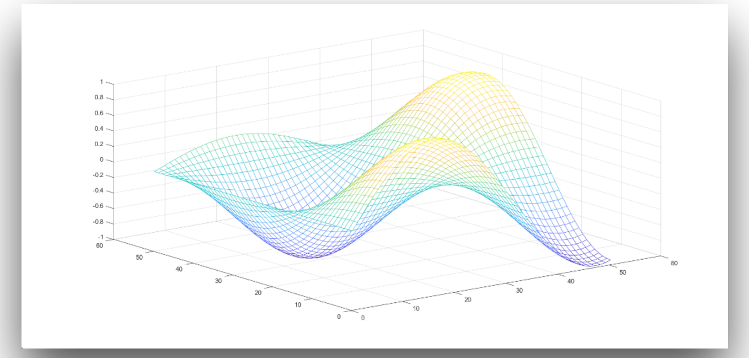
Summary Histogram → Statistical Fit → Likelihood $\mathscr{L}(\mu_1 | \mathscr{D})$

μ_1 → Statistical Fit

Hypothesis μ_1

μ is now arbitrary parameter of interest(s)

The neural inference framework:



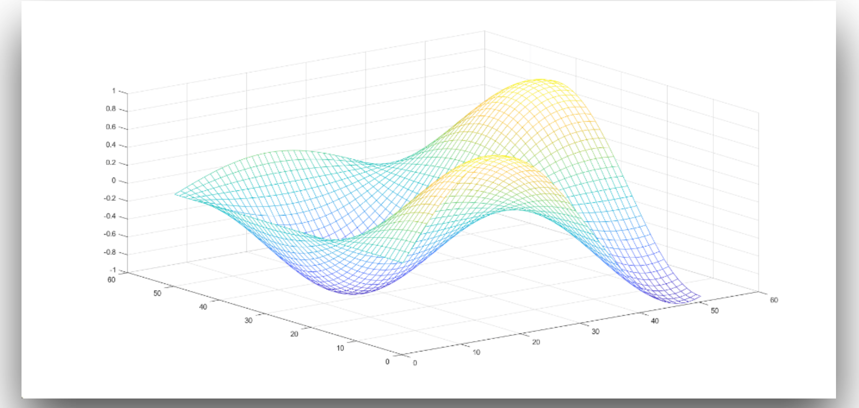
High-dim data

Obs Data → Neural Networks → Likelihood Ratio $\left(\frac{\mathscr{L}(\mu_1 | \mathscr{D})}{\mathscr{L}(ref | \mathscr{D})}\right)$

μ_1 → Neural Networks

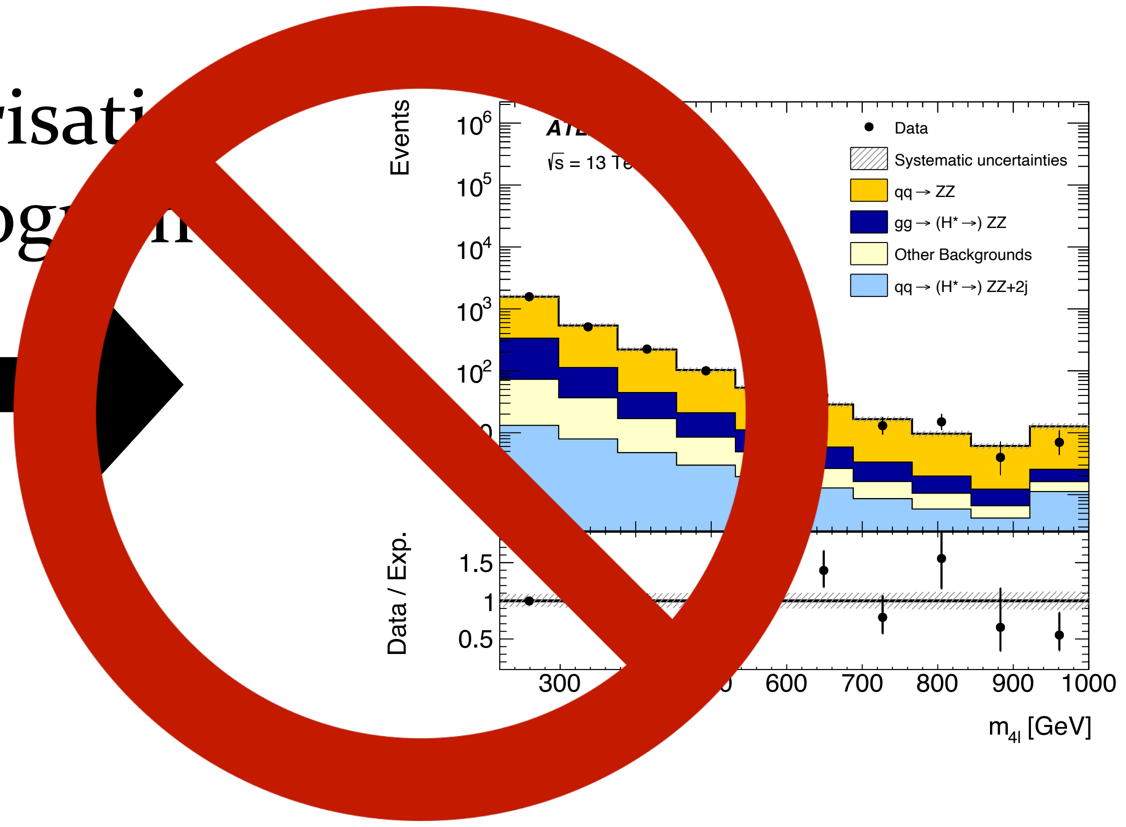
"Neural Simulation-Based Inference"

Traditional framework:



High-dim data

Summarisati
to histog



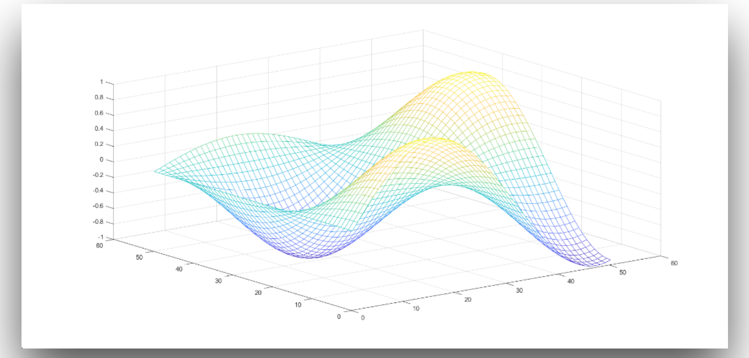
Summary Histogram → Statistical Fit → Likelihood $\mathscr{L}(\mu_1 | \mathscr{D})$

μ_1 → Statistical Fit

Hypothesis μ_1

μ is now arbitrary parameter of interest(s)

The neural inference framework:



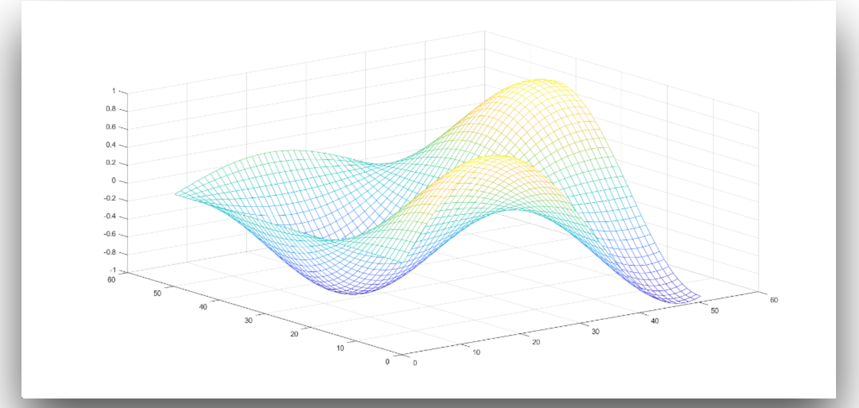
High-dim data

Obs Data → Neural Networks → Likelihood Ratio $\left(\frac{\mathscr{L}(\mu_1 | \mathscr{D})}{\mathscr{L}(ref | \mathscr{D})}\right)$

μ_1 → Neural Networks

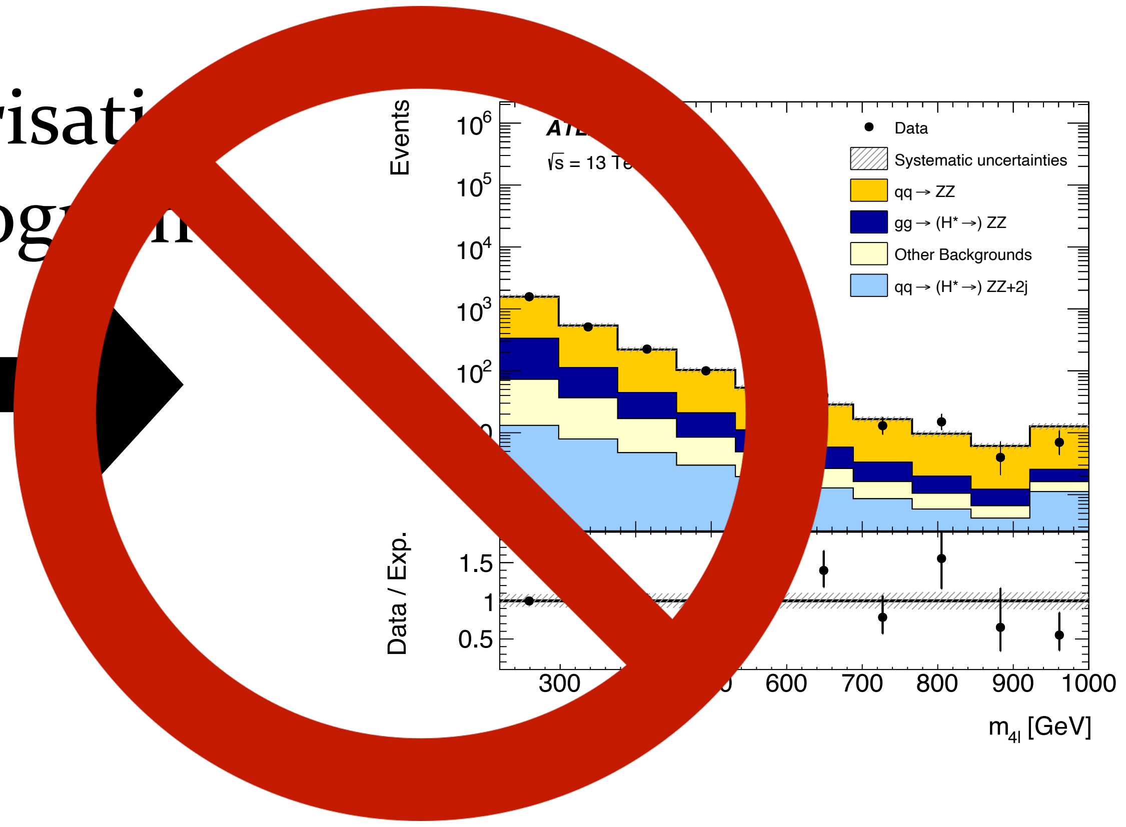
"Neural Simulation-Based Inference"

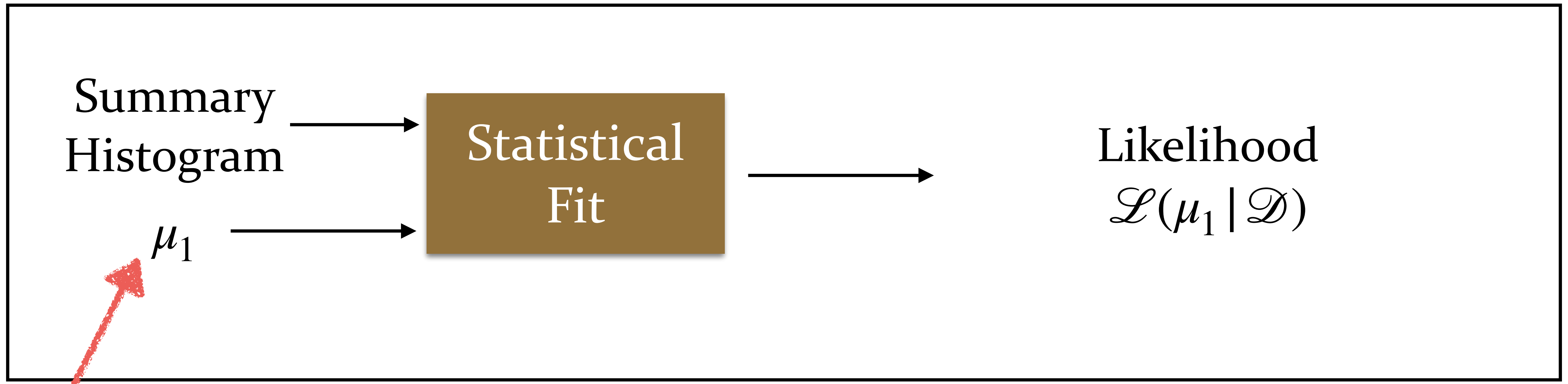
Traditional framework:



High-dim data

Summarisation to histogram

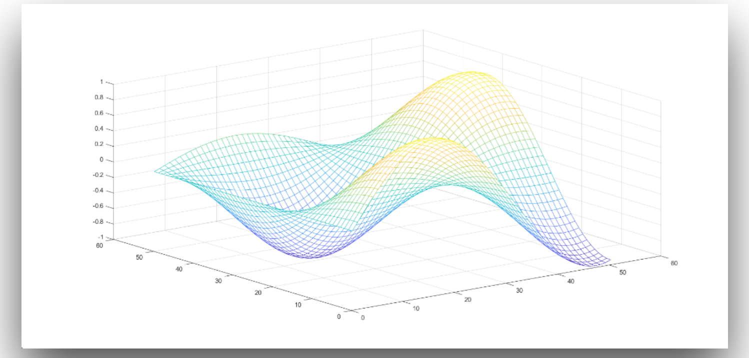




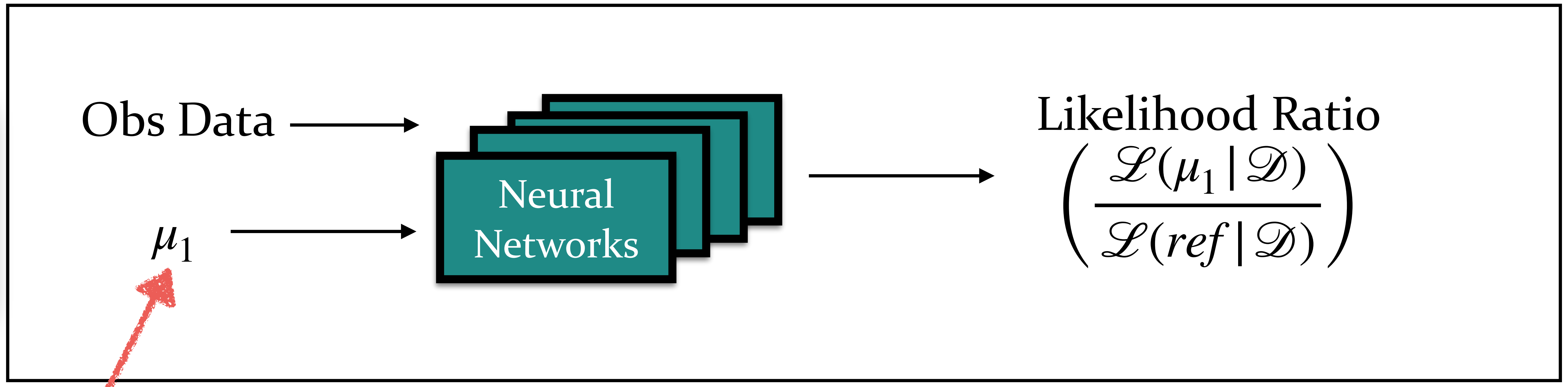
Hypothesis μ_1

μ is now arbitrary parameter of interest(s)

The neural inference framework:



High-dim data

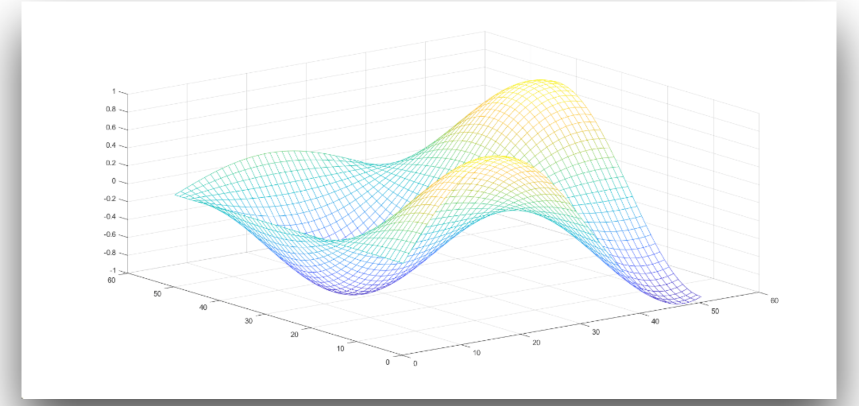


Hypothesis μ_1

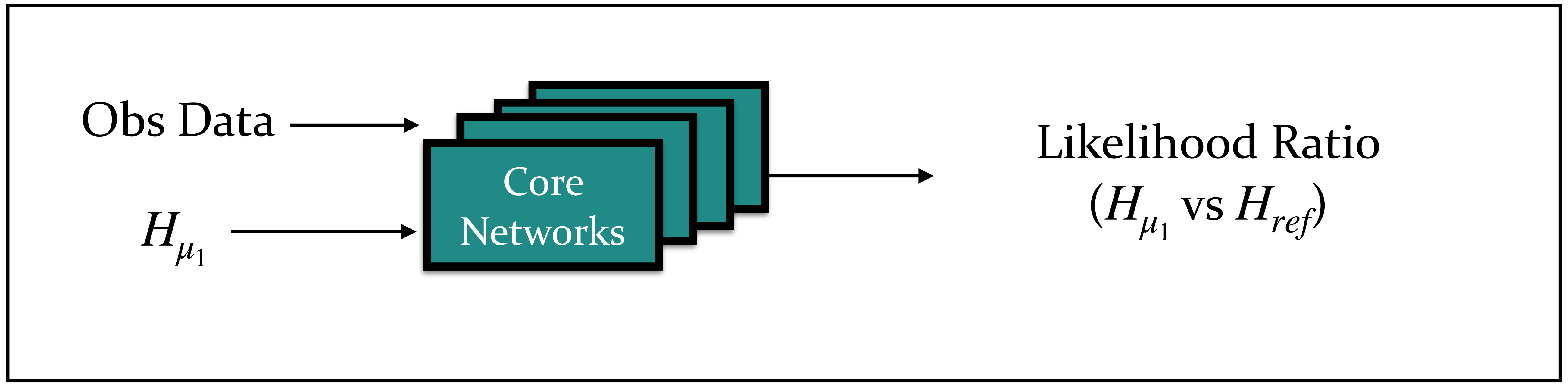
Open problems to extend to full ATLAS analysis:

- Robustness: Design and validation
- Systematic Uncertainties: Incorporate them in likelihood (ratio) model
- Neyman Construction: Throwing toys in a per-event analysis

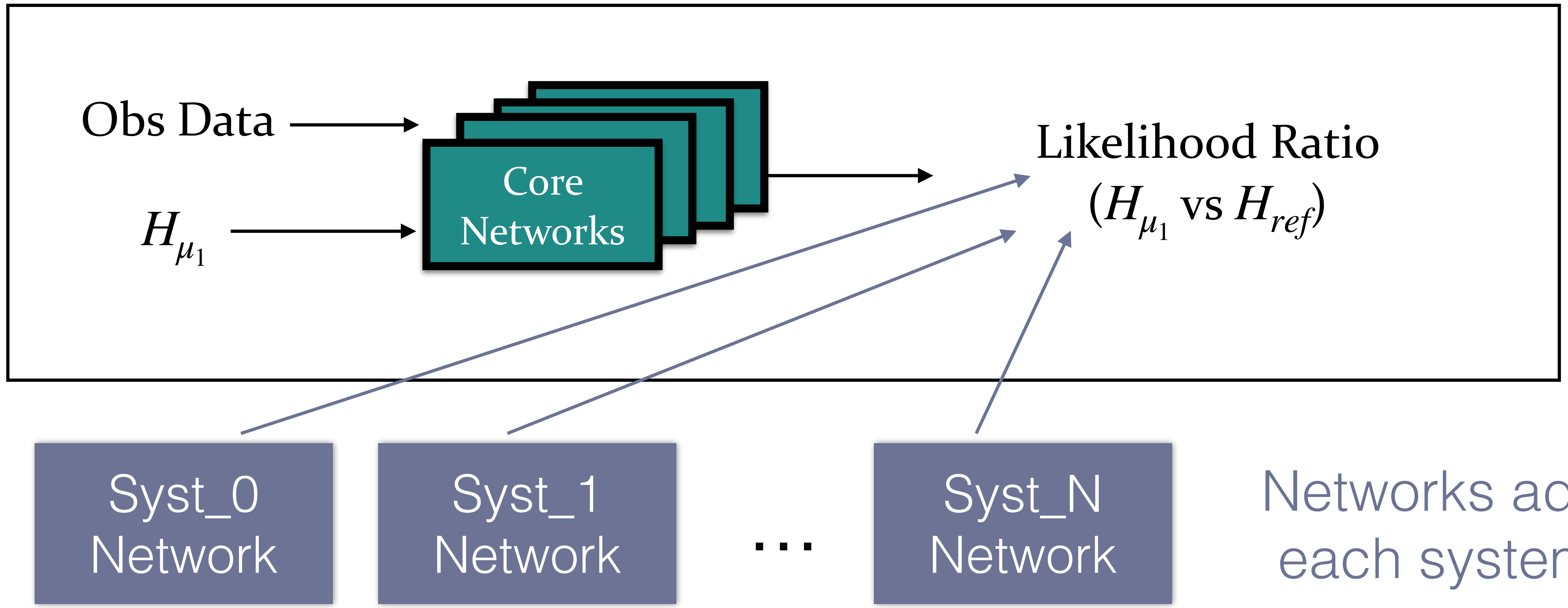
Big picture of full solution developed in ATLAS



$O(16)$ observables

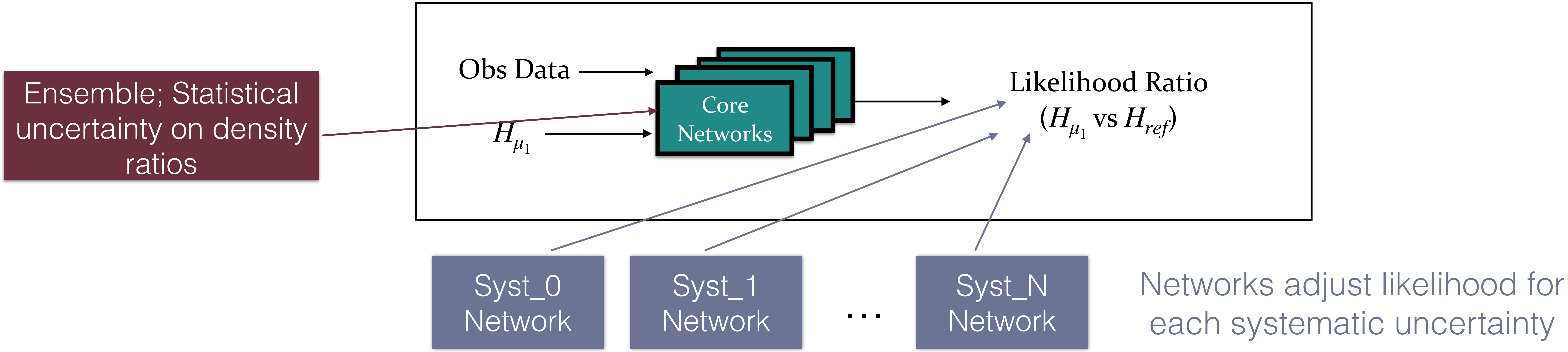


Big picture of full solution developed in ATLAS

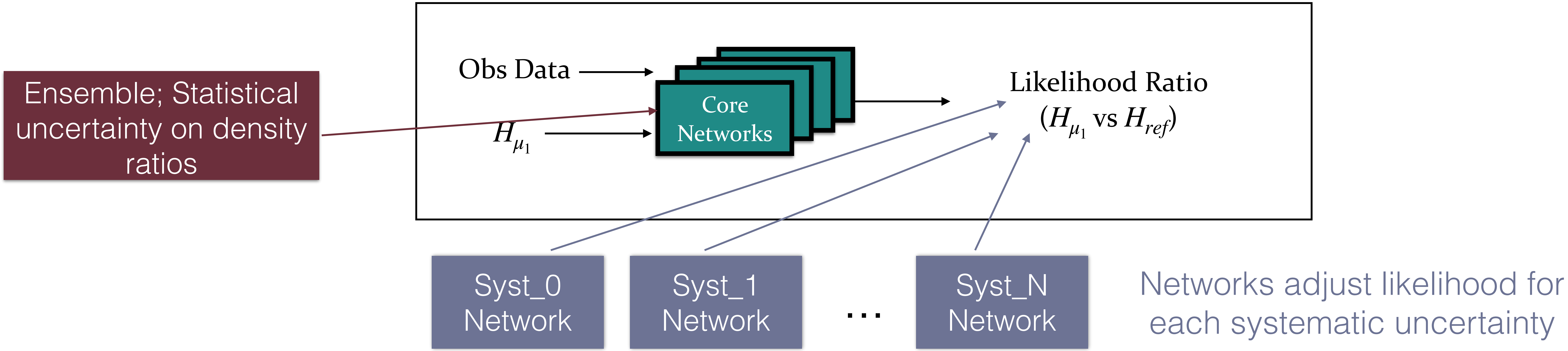


Networks adjust likelihood for each systematic uncertainty

Big picture of full solution developed in ATLAS



Big picture of full solution developed in ATLAS



- ✦ Train $O(10^3)$ networks on TensorFlow
- ✦ **Computing resources provided by Google, SMU,** other HPC clusters
- ✦ Fits with JAX

Open problems to extend to full ATLAS analysis:

- Robustness: Design and validation
- Systematic Uncertainties: Incorporate them in likelihood (ratio) model
- Neyman Construction: Throwing toys in a per-event analysis

Search-Oriented Mixture Model

x_i is one individual event

General Formula

$$p(x_i|\mu) = \frac{1}{\nu(\mu)} \sum_{j}^{C} f_j(\mu) \cdot \nu_j \; p_j(x_i)$$

j runs over different physics process
(Eg. $gg \to H^* \to 4l$, $gg \to ZZ \to 4l$)

Example use case

Search-Oriented Mixture Model

x_i is one individual event

General Formula

$$p(x_i|\mu) = \frac{1}{\nu(\mu)} \sum_{j}^{C} f_j(\mu) \cdot \nu_j \; p_j(x_i)$$

j runs over different physics process
(Eg. $gg \rightarrow H^* \rightarrow 4l$, $gg \rightarrow ZZ \rightarrow 4l$)

Example use case

$$p_{\mathrm{ggF}}(x|\mu) = \frac{1}{\nu_{\mathrm{ggF}}(\mu)} \left[(\mu - \sqrt{\mu}) \, \nu_S \, p_{\mathrm{S}}(x) + \sqrt{\mu} \, \nu_{\mathrm{SBI}_1} \, p_{\mathrm{SBI}_1}(x) + (1 - \sqrt{\mu}) \nu_{\mathrm{B}} \, p_{\mathrm{B}}(x) \right]$$

Search-Oriented Mixture Model

x_i is one individual event

General Formula

$$p(x_i|\mu) = \frac{1}{\nu(\mu)} \sum_j^C f_j(\mu) \cdot \nu_j \; p_j(x_i)$$

j runs over different physics process
(Eg. $gg \rightarrow H^* \rightarrow 4l$, $gg \rightarrow ZZ \rightarrow 4l$)

Comes from theory model chosen to interpret data

Example use case

$$p_{\mathrm{ggF}}(x|\mu) = \frac{1}{\nu_{\mathrm{ggF}}(\mu)} \left[(\mu - \sqrt{\mu})\, \nu_S \; p_S(x) + \sqrt{\mu}\, \nu_{\mathrm{SBI}_1} \; p_{\mathrm{SBI}_1}(x) + (1 - \sqrt{\mu}) \nu_{\mathrm{B}} \; p_{\mathrm{B}}(x) \right]$$

Search-Oriented Mixture Model

x_i is one individual event

General Formula

$$p(x_i|\mu) = \frac{1}{\nu(\mu)} \sum_{j}^{C} f_j(\mu) \cdot \nu_j \, p_j(x_i)$$

j runs over different physics process
(Eg. $gg \to H^* \to 4l$, $gg \to ZZ \to 4l$)

Event rates estimated from simulations

Comes from theory model chosen to interpret data

Example use case

$$p_{\mathrm{ggF}}(x|\mu) = \frac{1}{\nu_{\mathrm{ggF}}(\mu)} \left[(\mu - \sqrt{\mu})\, \nu_S \, p_S(x) + \sqrt{\mu}\, \nu_{\mathrm{SBI}_1} \, p_{\mathrm{SBI}_1}(x) + (1 - \sqrt{\mu}) \nu_B \, p_B(x) \right]$$

Search-Oriented Mixture Model

x_i is one individual event

General Formula

$$p(x_i|\mu) = \frac{1}{\nu(\mu)} \sum_{j}^{C} f_j(\mu) \cdot \nu_j \, p_j(x_i)$$

?

j runs over different physics process
(Eg. $gg \to H^* \to 4l$, $gg \to ZZ \to 4l$)

Event rates estimated from simulations

Comes from theory model chosen to interpret data

Example use case

$$p_{\mathrm{ggF}}(x|\mu) = \frac{1}{\nu_{\mathrm{ggF}}(\mu)} \left[(\mu - \sqrt{\mu})\, \nu_S \, p_S(x) + \sqrt{\mu}\, \nu_{\mathrm{SBI}_1}\, p_{\mathrm{SBI}_1}(x) + (1 - \sqrt{\mu}) \nu_B \, p_B(x) \right]$$

Search-Oriented Mixture Model

x_i is one individual event

General Formula

$$p(x_i|\mu) = \frac{1}{\nu(\mu)} \sum_j^C f_j(\mu) \cdot \nu_j \, p_j(x_i) \quad ? \quad \Rightarrow \quad \frac{p(x_i|\mu)}{p_{\text{ref}}(x_i)} = \frac{1}{\nu(\mu)} \sum_j^C f_j(\mu) \cdot \nu_j \, \frac{p_j(x_i)}{p_{\text{ref}}(x_i)}$$

Reference hypothesis

j runs over different physics process (Eg. $gg \to H^* \to 4l$, $gg \to ZZ \to 4l$)

Event rates estimated from simulations

Comes from theory model chosen to interpret data

Example use case

$$p_{\text{ggF}}(x|\mu) = \frac{1}{\nu_{\text{ggF}}(\mu)} \left[(\mu - \sqrt{\mu})\, \nu_S \, p_S(x) + \sqrt{\mu}\, \nu_{\text{SBI}_1} \, p_{\text{SBI}_1}(x) + (1 - \sqrt{\mu})\, \nu_B \, p_B(x) \right]$$

Search-Oriented Mixture Model

x_i is one individual event

General Formula

$$p(x_i|\mu) = \frac{1}{\nu(\mu)} \sum_j^C f_j(\mu) \cdot \nu_j \, p_j(x_i) \quad \Rightarrow \quad \frac{p(x_i|\mu)}{p_{\text{ref}}(x_i)} = \frac{1}{\nu(\mu)} \sum_j^C f_j(\mu) \cdot \nu_j \, \frac{p_j(x_i)}{p_{\text{ref}}(x_i)}$$

?

Reference hypothesis

j runs over different physics process (Eg. $gg \to H^* \to 4l$, $gg \to ZZ \to 4l$)

Event rates estimated from simulations

Comes from theory model chosen to interpret data

Example use case

$$p_{\text{ggF}}(x|\mu) = \frac{1}{\nu_{\text{ggF}}(\mu)} \left[(\mu - \sqrt{\mu})\, \nu_S \, p_S(x) + \sqrt{\mu}\, \nu_{\text{SBI}_1} \, p_{\text{SBI}_1}(x) + (1 - \sqrt{\mu}) \nu_B \, p_B(x) \right]$$

$$\Rightarrow \quad \frac{p(x|\mu)}{p_S(x)} = \frac{1}{\nu(\mu)} \left[(\mu - \sqrt{\mu})\, \nu_S + \sqrt{\mu}\, \nu_{\text{SBI}_1} \frac{p_{\text{SBI}_1}(x)}{p_S(x)} + (1 - \sqrt{\mu}) \nu_B \frac{p_B(x)}{p_S(x)} \right]$$

Search-Oriented Mixture Model

x_i is one individual event

General Formula

$$p(x_i|\mu) = \frac{1}{\nu(\mu)} \sum_j^C f_j(\mu) \cdot \nu_j \, p_j(x_i)$$

?

Estimated using an ensemble of networks

$$\frac{p(x_i|\mu)}{p_{\text{ref}}(x_i)} = \frac{1}{\nu(\mu)} \sum_j^C f_j(\mu) \cdot \nu_j \frac{p_j(x_i)}{p_{\text{ref}}(x_i)}$$

Reference hypothesis

j runs over different physics process (Eg. $gg \to H^* \to 4l$, $gg \to ZZ \to 4l$)

Event rates estimated from simulations

Comes from theory model chosen to interpret data

Example use case

$$p_{\text{ggF}}(x|\mu) = \frac{1}{\nu_{\text{ggF}}(\mu)} \left[(\mu - \sqrt{\mu})\, \nu_S \, p_S(x) + \sqrt{\mu}\, \nu_{\text{SBI}_1} \, p_{\text{SBI}_1}(x) + (1 - \sqrt{\mu}) \nu_B \, p_B(x) \right]$$

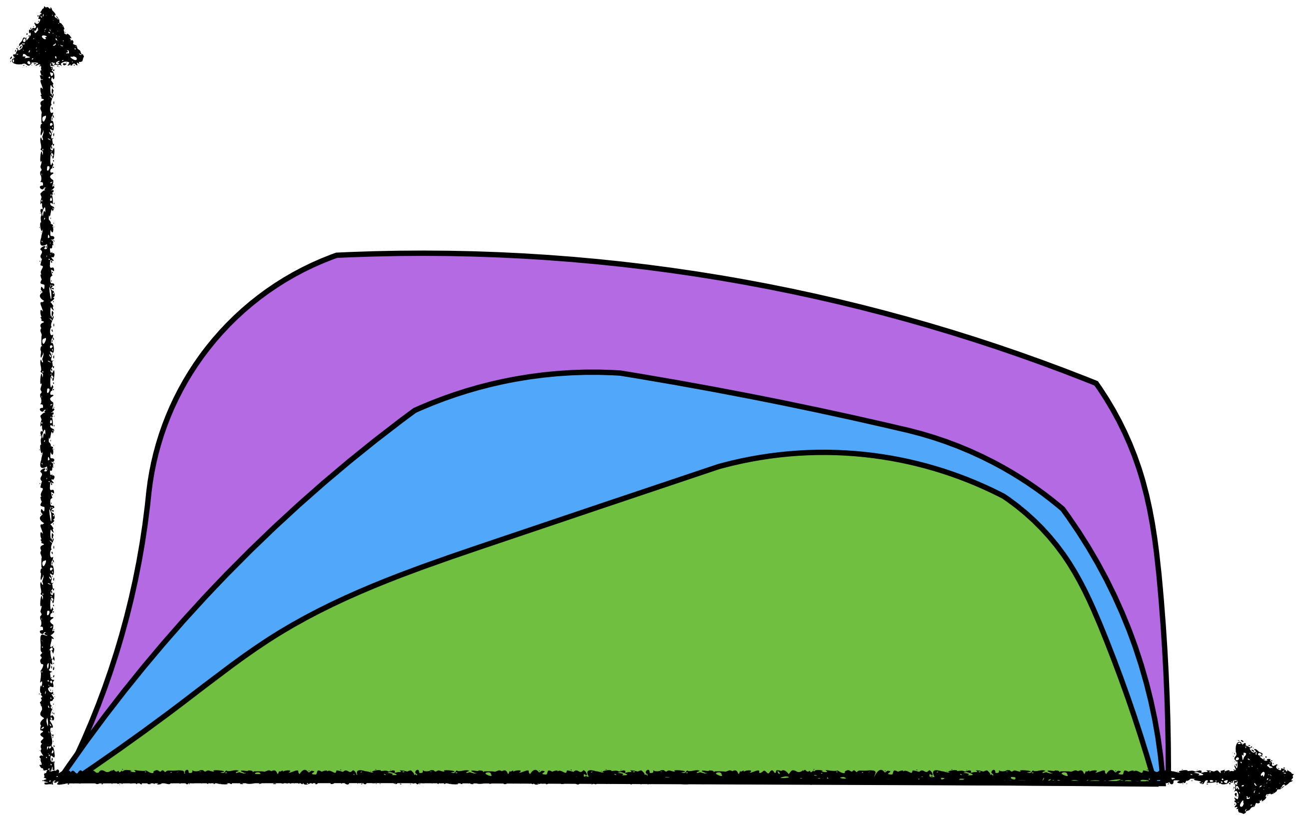
$$\frac{p(x|\mu)}{p_S(x)} = \frac{1}{\nu(\mu)} \left[(\mu - \sqrt{\mu})\, \nu_S + \sqrt{\mu}\, \nu_{\text{SBI}_1} \frac{p_{\text{SBI}_1}(x)}{p_S(x)} + (1 - \sqrt{\mu}) \nu_B \frac{p_B(x)}{p_S(x)} \right]$$

Robust, parameterised classifier without parameterising

H_{ref} : Reference hypothesis

H_μ

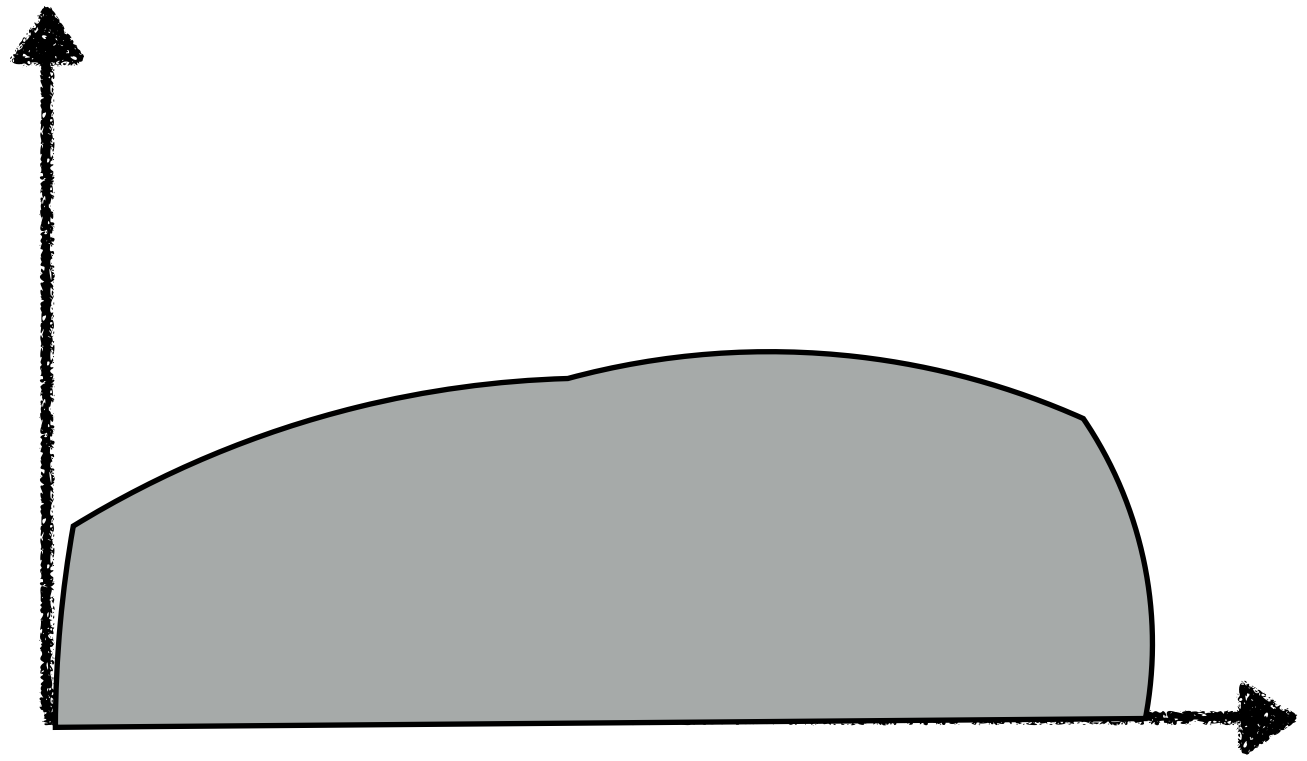
Hypothesis



VS

H_{ref}

Reference hypothesis



$$\frac{p(x_i|\mu)}{p_{\text{ref}}(x_i)} = \frac{1}{\nu(\mu)} \sum_j^C f_j(\mu) \cdot \nu_j \, \frac{p_j(x_i)}{p_{\text{ref}}(x_i)}$$

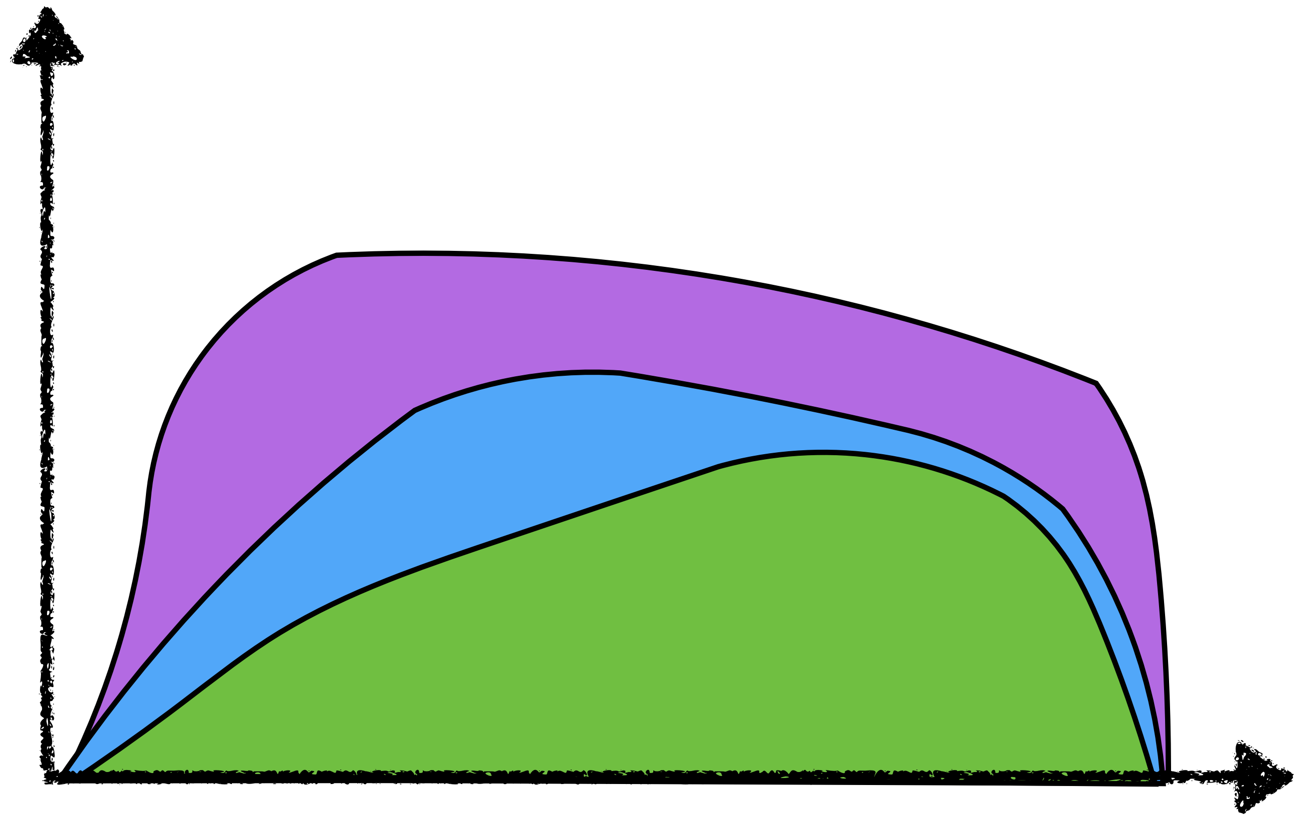
A separate classifier per physics process j
(Eg. $gg \to H^* \to 4l$, $gg \to ZZ \to 4l$)

Robust, parameterised classifier without parameterising

H_{ref} : Reference hypothesis

H_μ

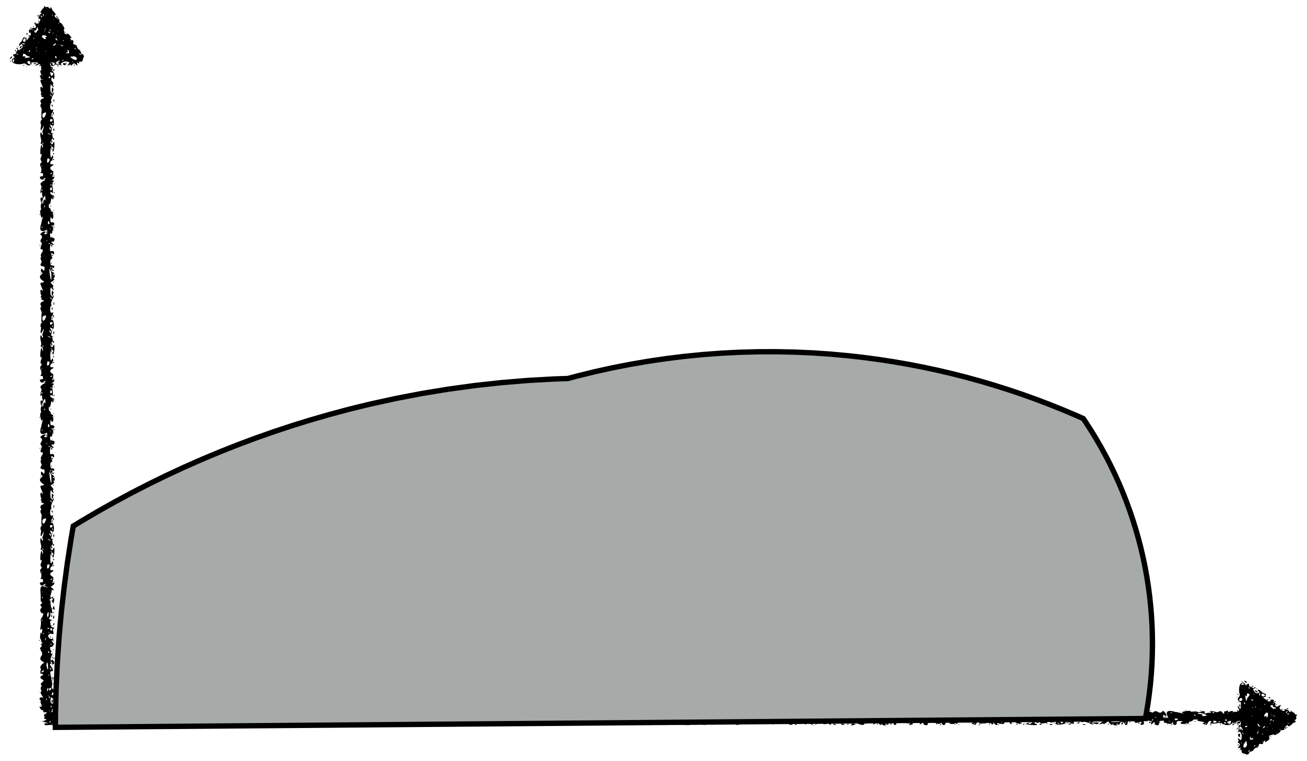
Hypothesis



VS

H_{ref}

Reference hypothesis



$$\frac{p(x_i|\mu)}{p_{\text{ref}}(x_i)} = \frac{1}{\nu(\mu)} \sum_j^C f_j(\mu) \cdot \nu_j \; \frac{p_j(x_i)}{p_{\text{ref}}(x_i)}$$

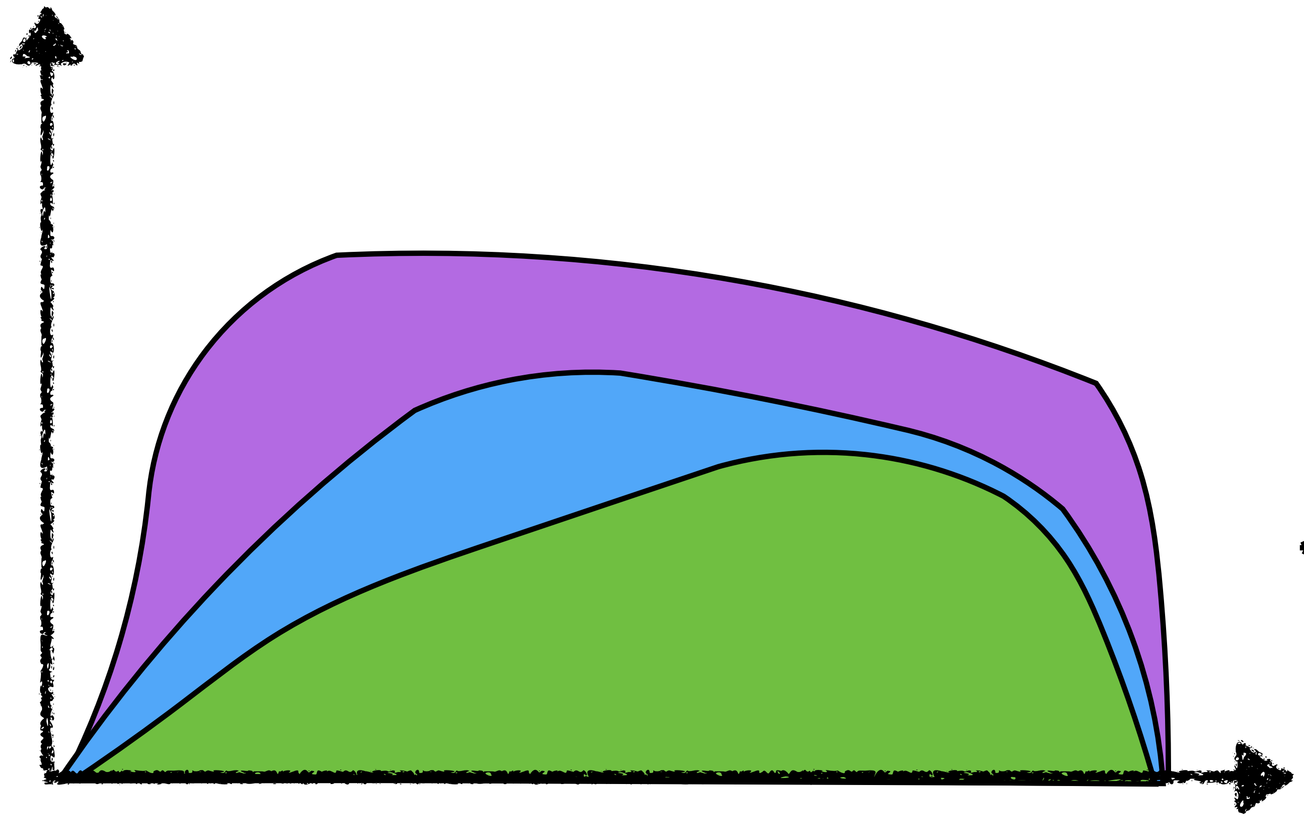
A separate classifier per physics process j
(Eg. $gg \rightarrow H^* \rightarrow 4l$, $gg \rightarrow ZZ \rightarrow 4l$)

Robust, parameterised classifier without parameterising

H_{ref}: Reference hypothesis

H_μ

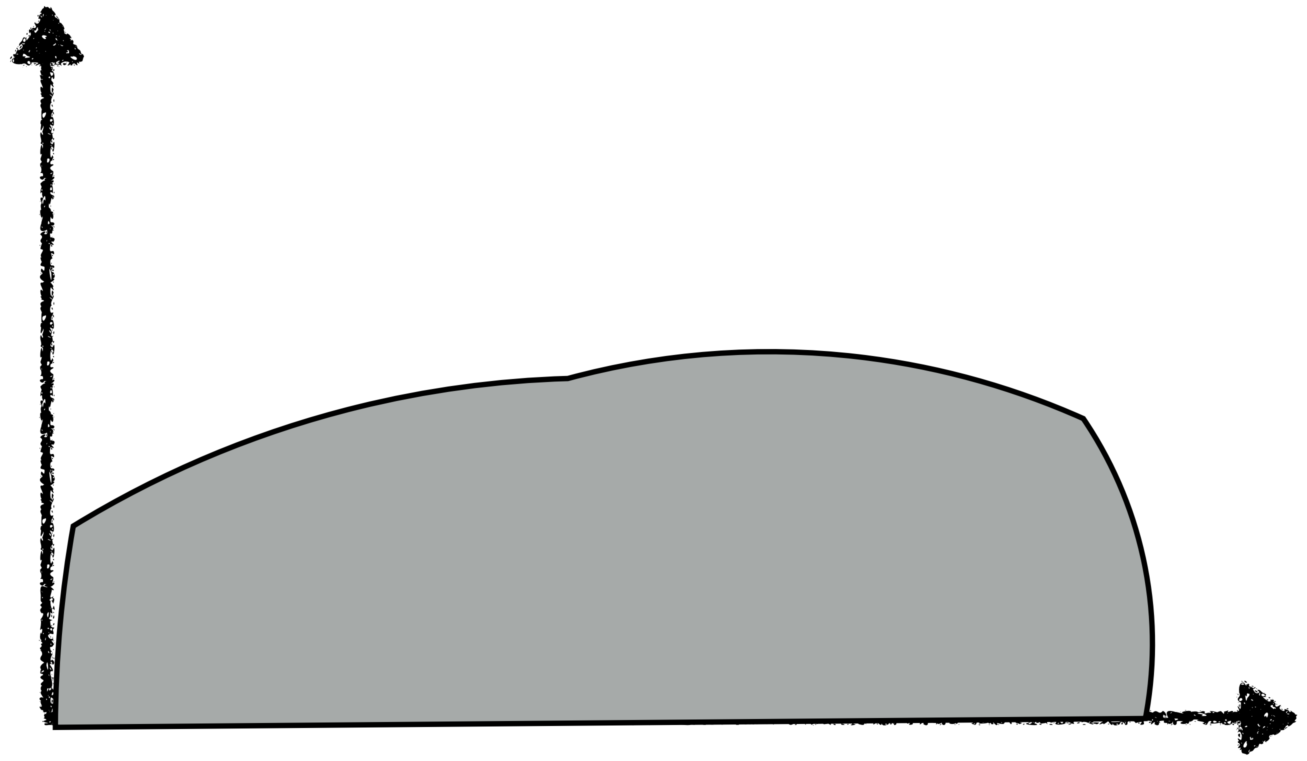
Hypothesis

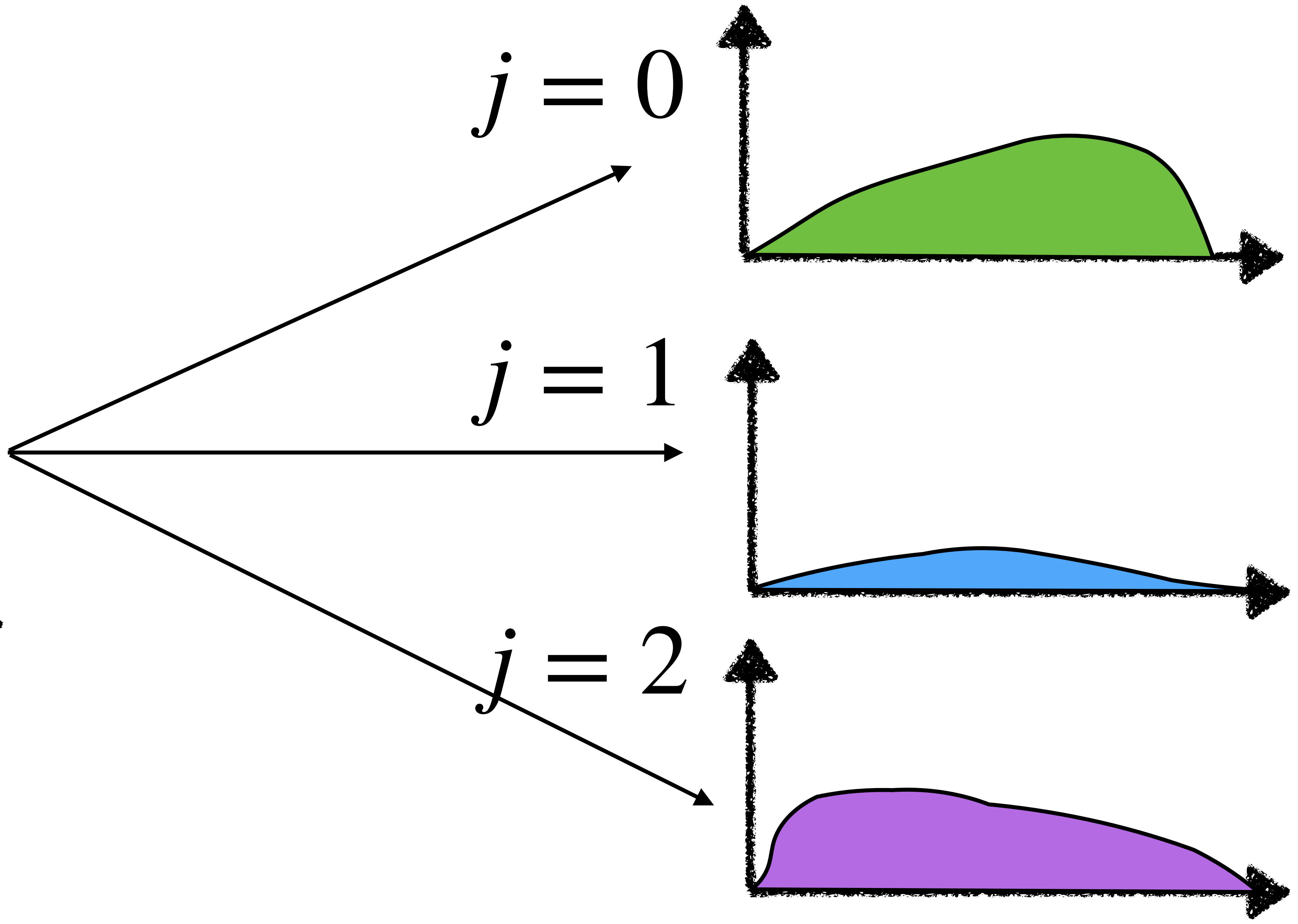


VS

H_{ref}

Reference hypothesis

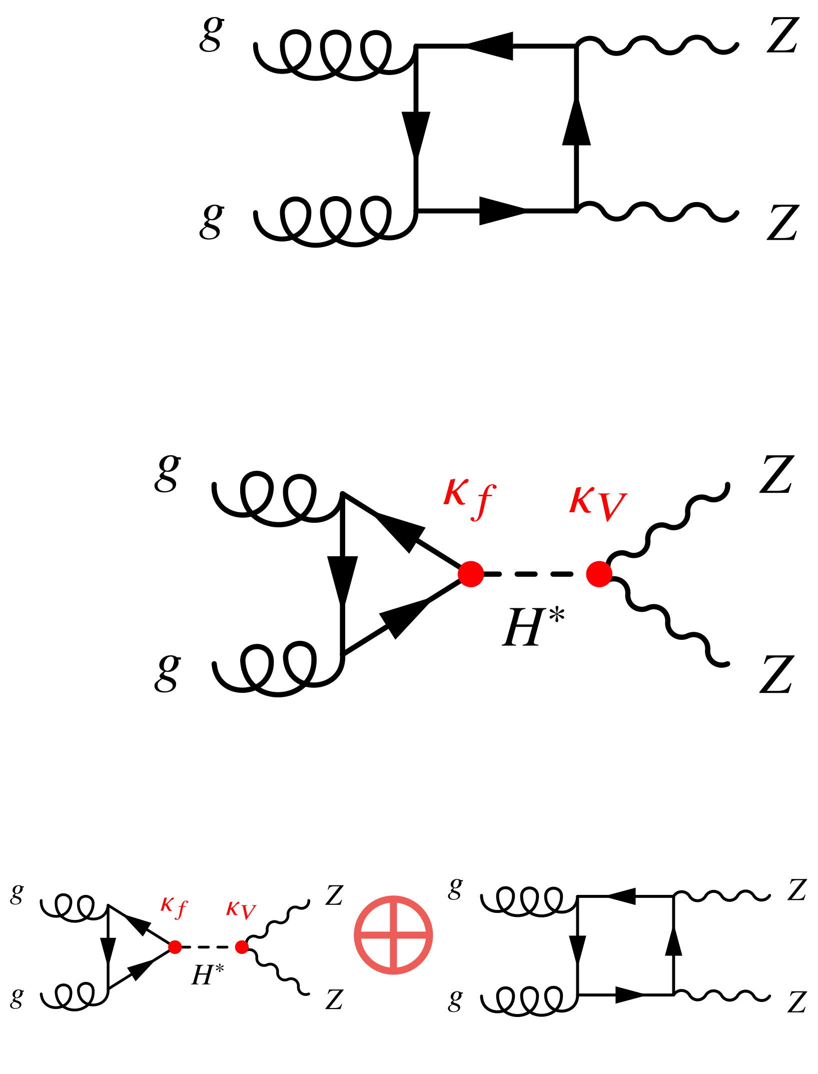




$f_0(\mu) = 1$

$f_1(\mu) = \mu$

$f_2(\mu) = \sqrt{\mu}$



$$\frac{p(x_i|\mu)}{p_{\text{ref}}(x_i)} = \frac{1}{\nu(\mu)} \sum_j^C f_j(\mu) \cdot \nu_j \, \frac{p_j(x_i)}{p_{\text{ref}}(x_i)}$$

A separate classifier per physics process j
(Eg. $gg \to H^* \to 4l$, $gg \to ZZ \to 4l$)

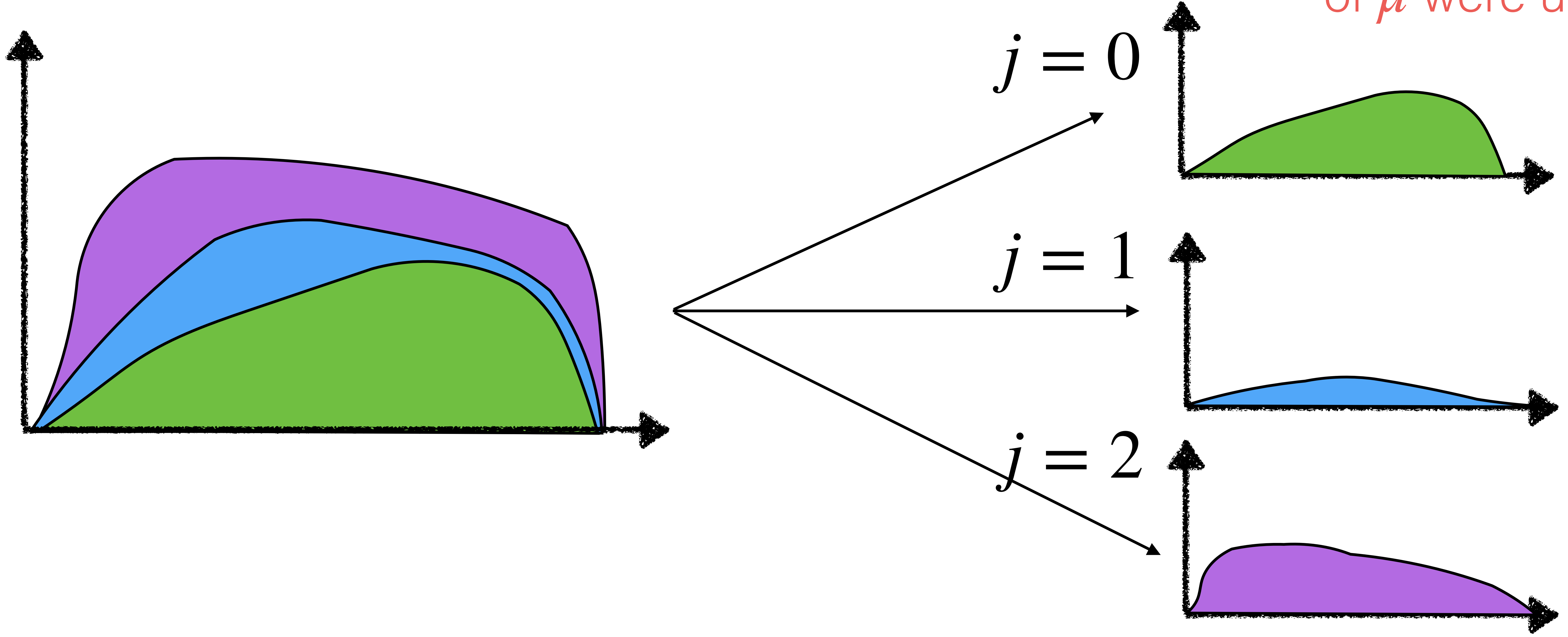
Robust, parameterised classifier without parameterising

H_{ref}: Reference hypothesis

$f_i(\mu)$ will depend on morphing bases points (which values of μ were used to simulate samples)

H_μ

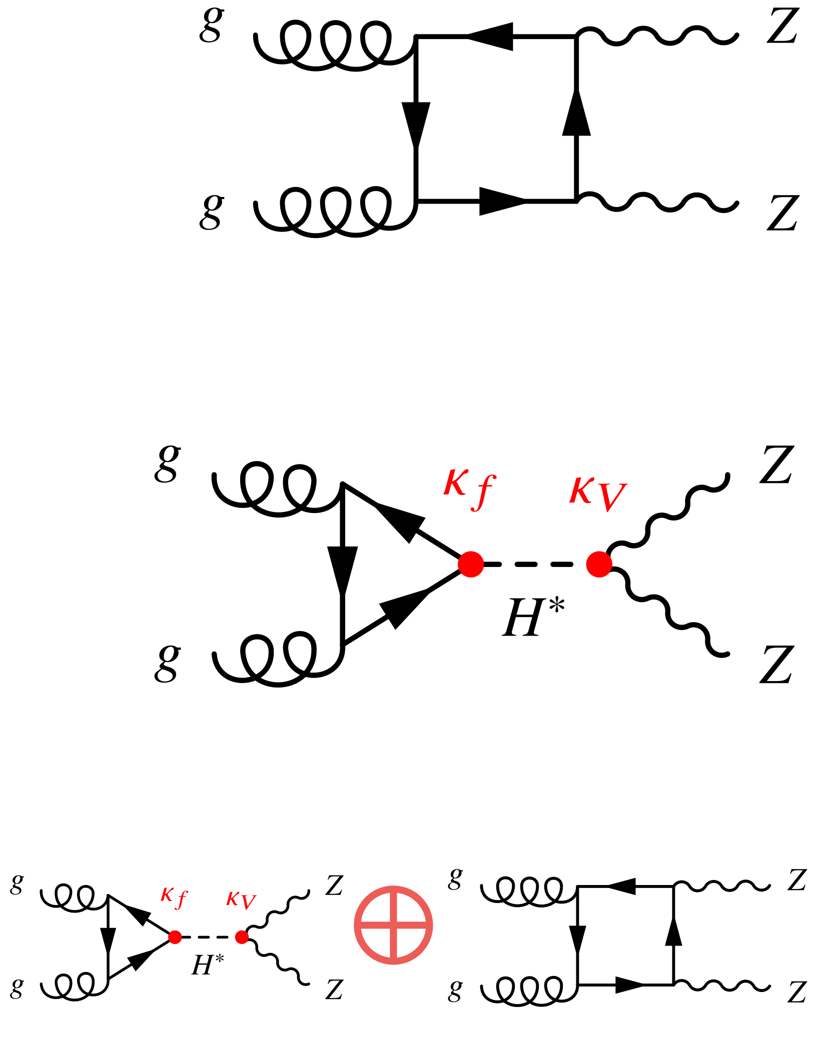
Hypothesis



$f_0(\mu) = 1$

$f_1(\mu) = \mu$

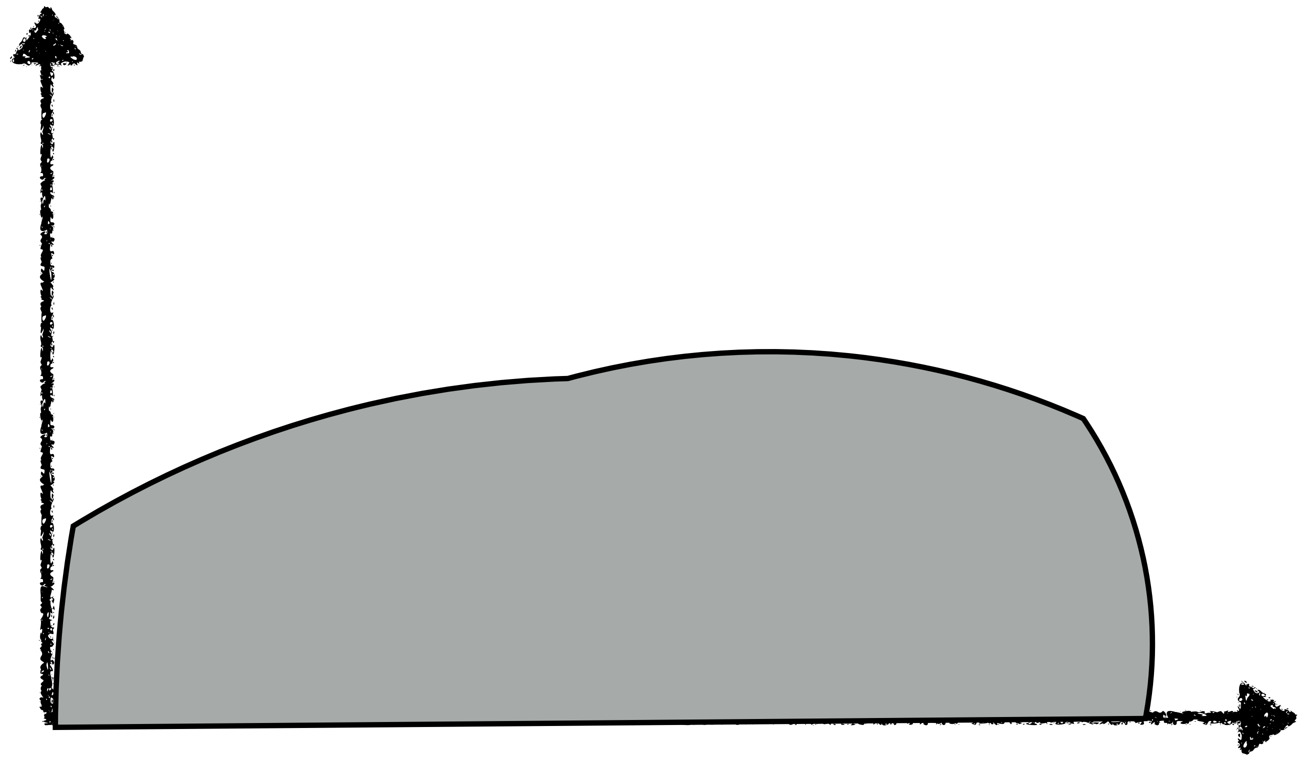
$f_2(\mu) = \sqrt{\mu}$



VS

H_{ref}

Reference hypothesis



$$\frac{p(x_i|\mu)}{p_{\mathrm{ref}}(x_i)} = \frac{1}{\nu(\mu)} \sum_j^C f_j(\mu) \cdot \nu_j \frac{p_j(x_i)}{p_{\mathrm{ref}}(x_i)}$$

A separate classifier per physics process j

(Eg. $gg \rightarrow H^* \rightarrow 4l$, $gg \rightarrow ZZ \rightarrow 4l$)

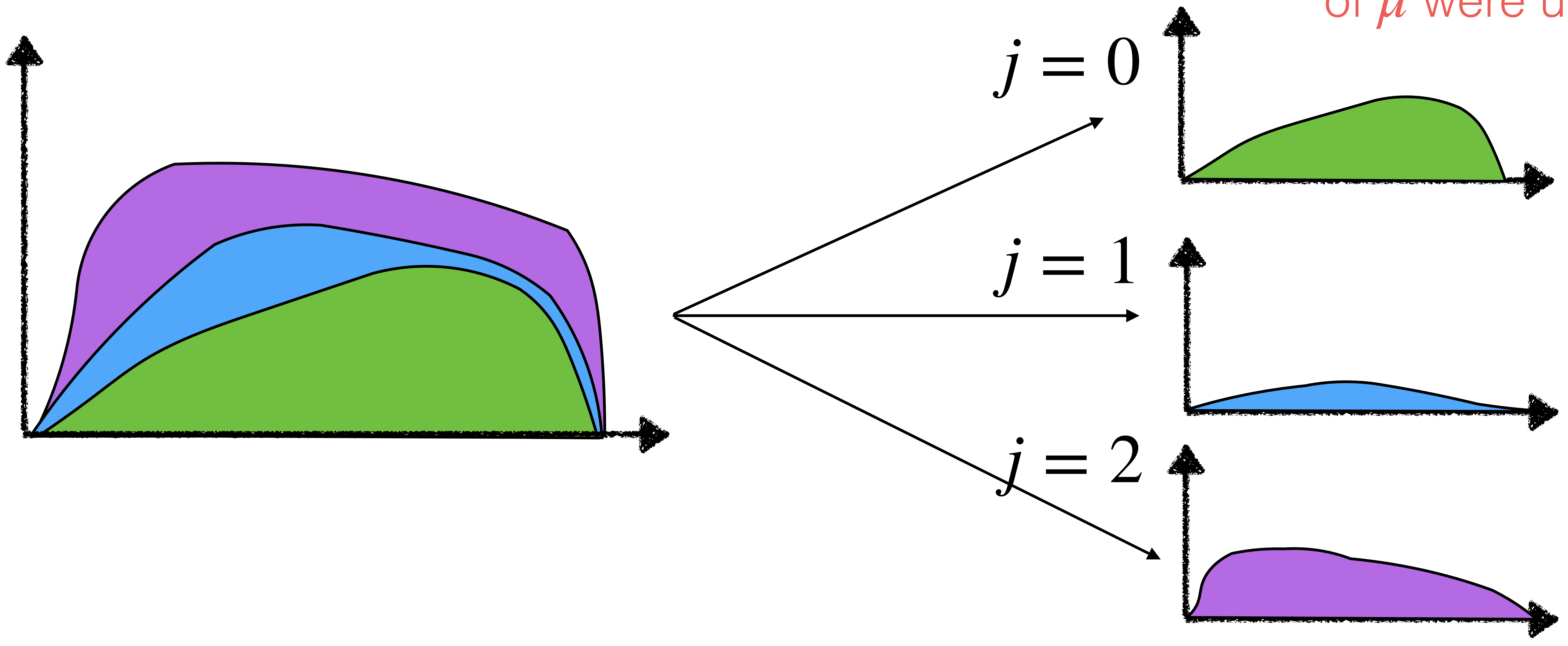
Robust, parameterised classifier without parameterising

H_{ref} : Reference hypothesis

$f_i(\mu)$ will depend on morphing bases points (which values of μ were used to simulate samples)

H_μ

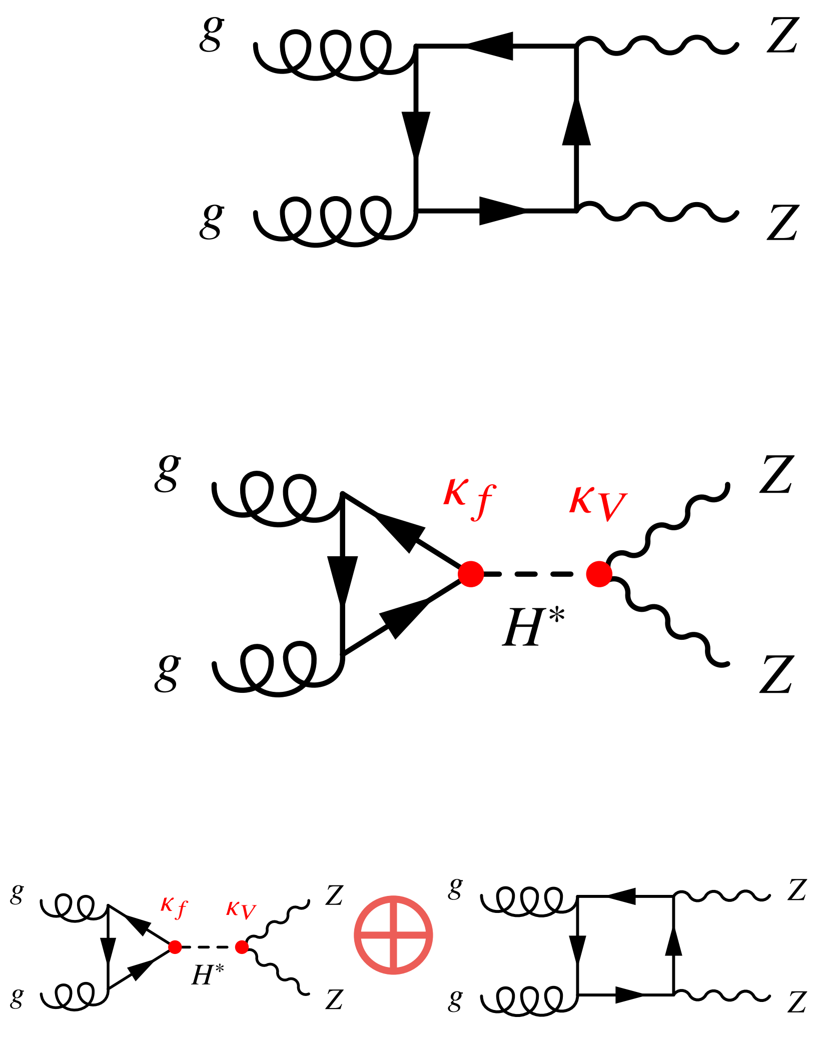
Hypothesis



$f_0(\mu) = 1$

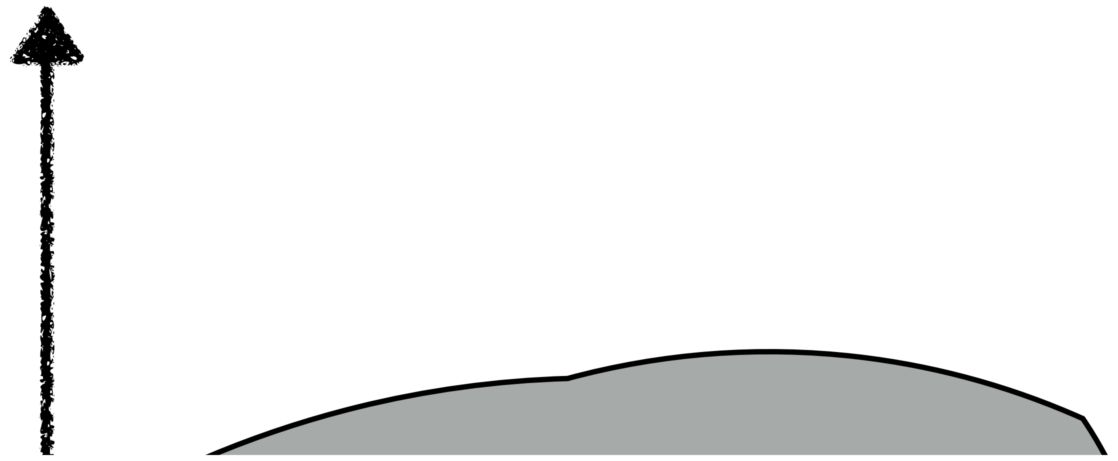
$f_1(\mu) = \mu$

$f_2(\mu) = \sqrt{\mu}$



vs

H_{ref}



$$\frac{p(x_i|\mu)}{p_{\text{ref}}(x_i)} = \frac{1}{\nu(\mu)} \sum_j^C f_j(\mu) \cdot \nu_j \frac{p_j(x_i)}{p_{\text{ref}}(x_i)}$$

Analytically parameterised in μ, allows to get LR for any hypothesis μ without training parameterised networks !

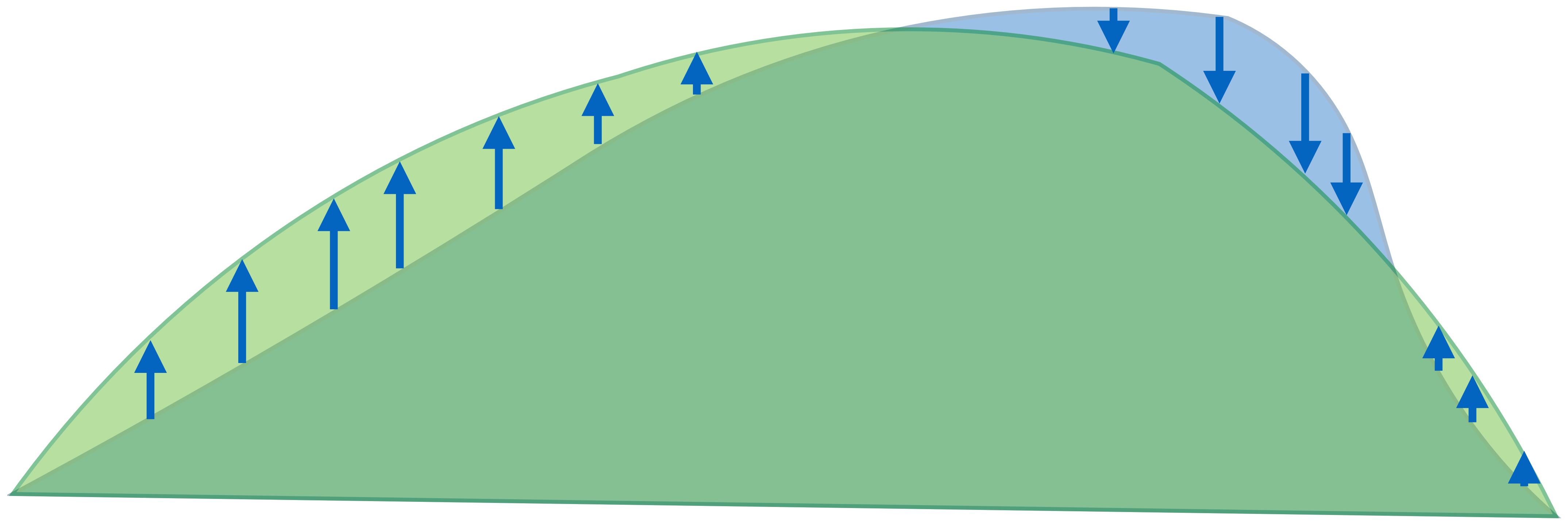
A separate classifier per physics process j (Eg. $gg \to H^* \to 4l$, $gg \to ZZ \to 4l$)

Open problems to extend to full ATLAS analysis:

- Robustness: Design and validation
- Systematic Uncertainties: Incorporate them in likelihood (ratio) model
- Neyman Construction: Throwing toys in a per-event analysis

Validate quality of LR estimation with re-weighting task

Reweighting: Calculate weights w_i for events x_i in blue sample to match green sample

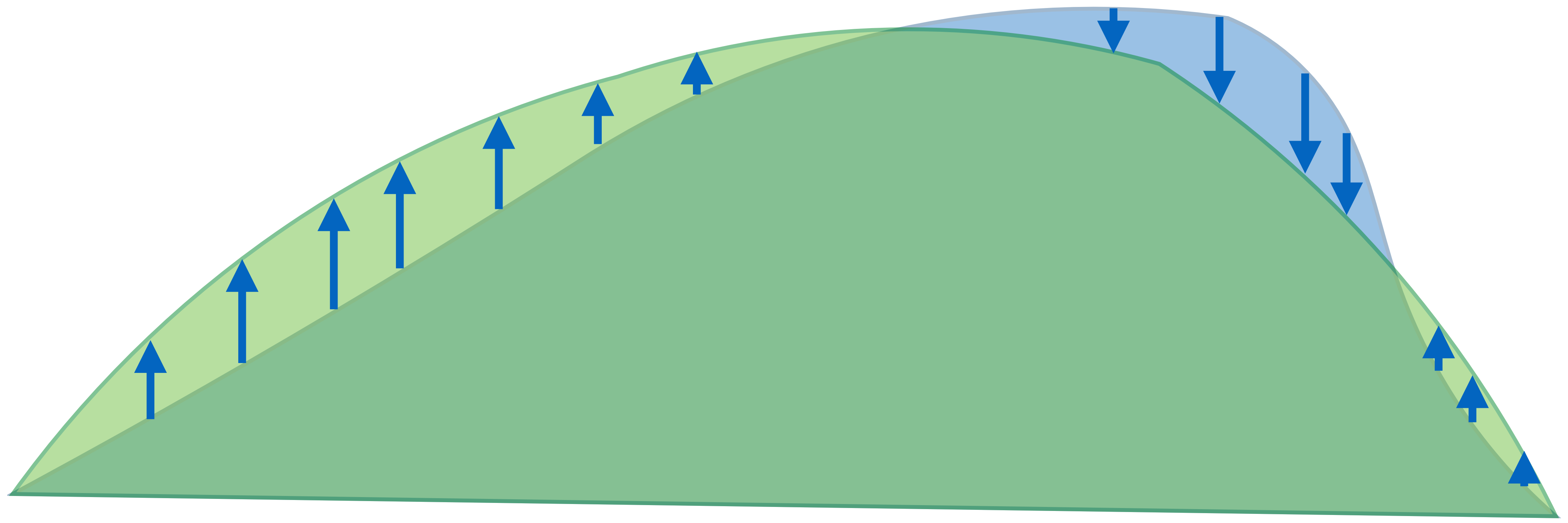


Validate quality of LR estimation with re-weighting task

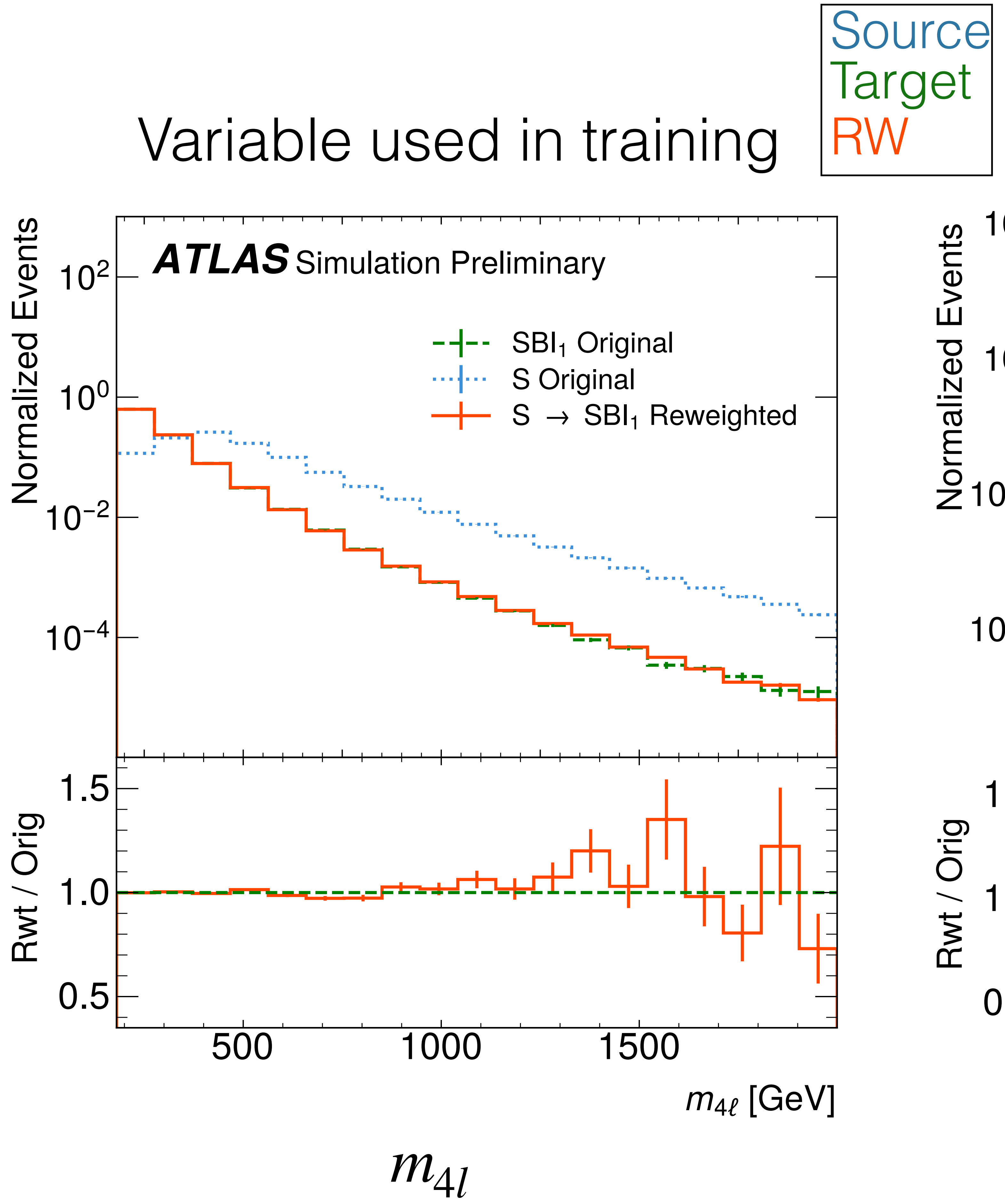
Reweighting: Calculate weights w_i for events x_i in blue sample to match green sample

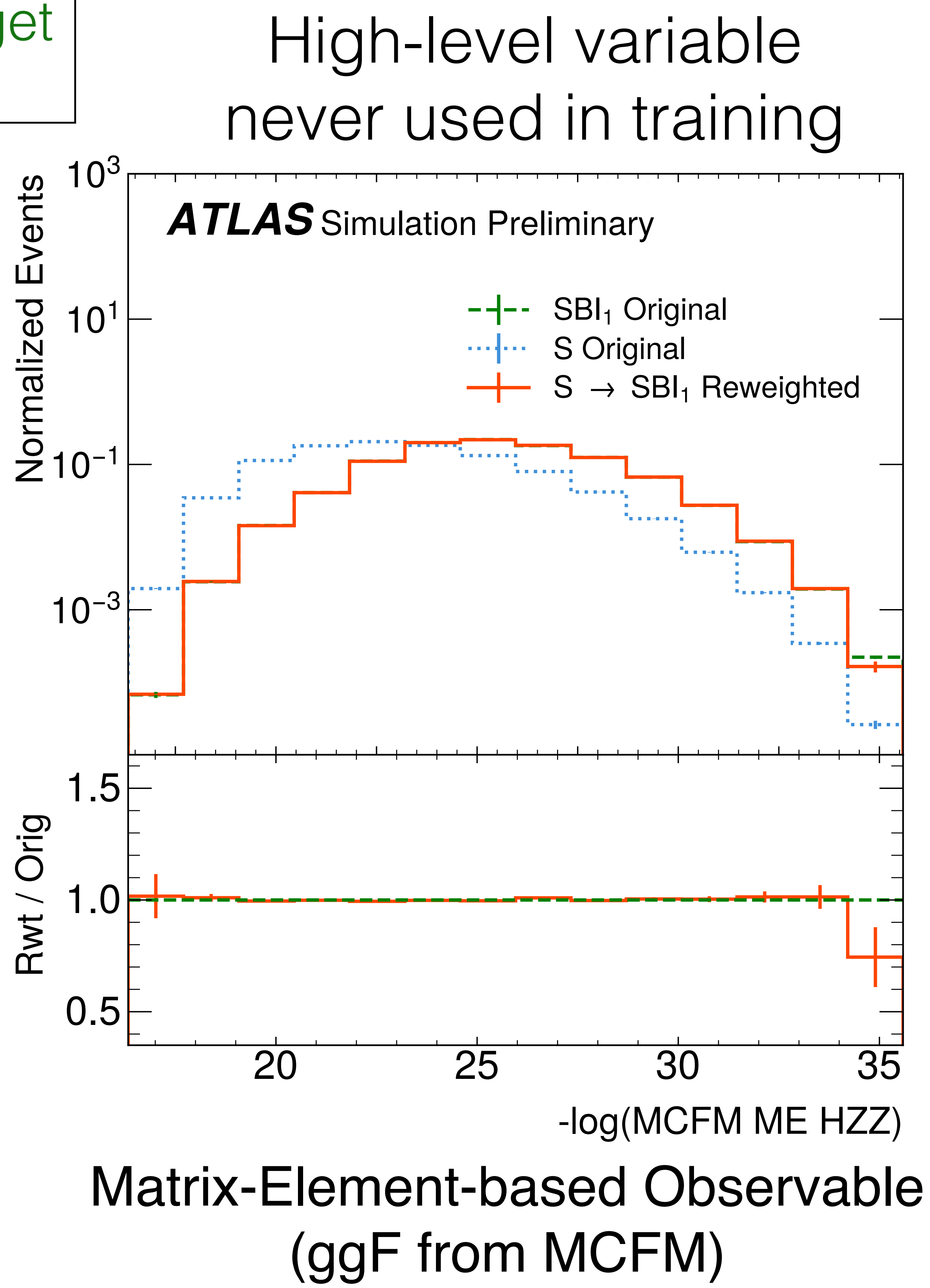
$$w_i = r(x_i, \mu_0, \mu_1) = \frac{p(x_i | \mu_0)}{p(x_i | \mu_1)}$$

Already estimated using an ensemble of networks



Re-weight closures





High-Dim Classifier Test:

Train independent classifier on RW vs Target, AUC=0.5 ⇒ LRs well estimated

See additional closure tests

Open problems to extend to full ATLAS analysis:

✓ Robustness: Design and validation

‣ Systematic Uncertainties: Incorporate them in likelihood (ratio) model

- Neyman Construction: Throwing toys in a per-event analysis

Systematic uncertainties

Experimental uncertainties:

Eg. Inaccuracies in the calibration of our detector

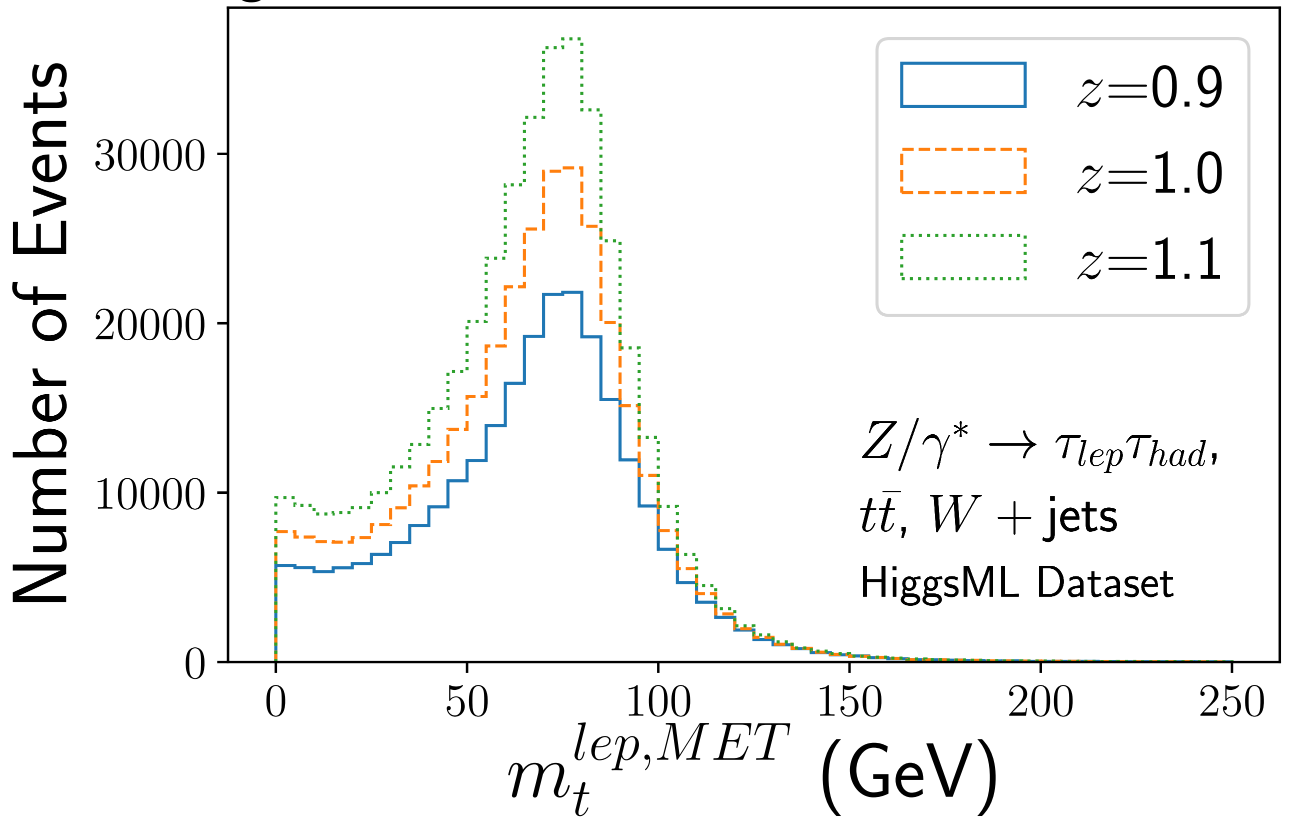


Image: arXiv:2105.08742

Theory uncertainties:

Eg. Inability to compute QFT to infinite order

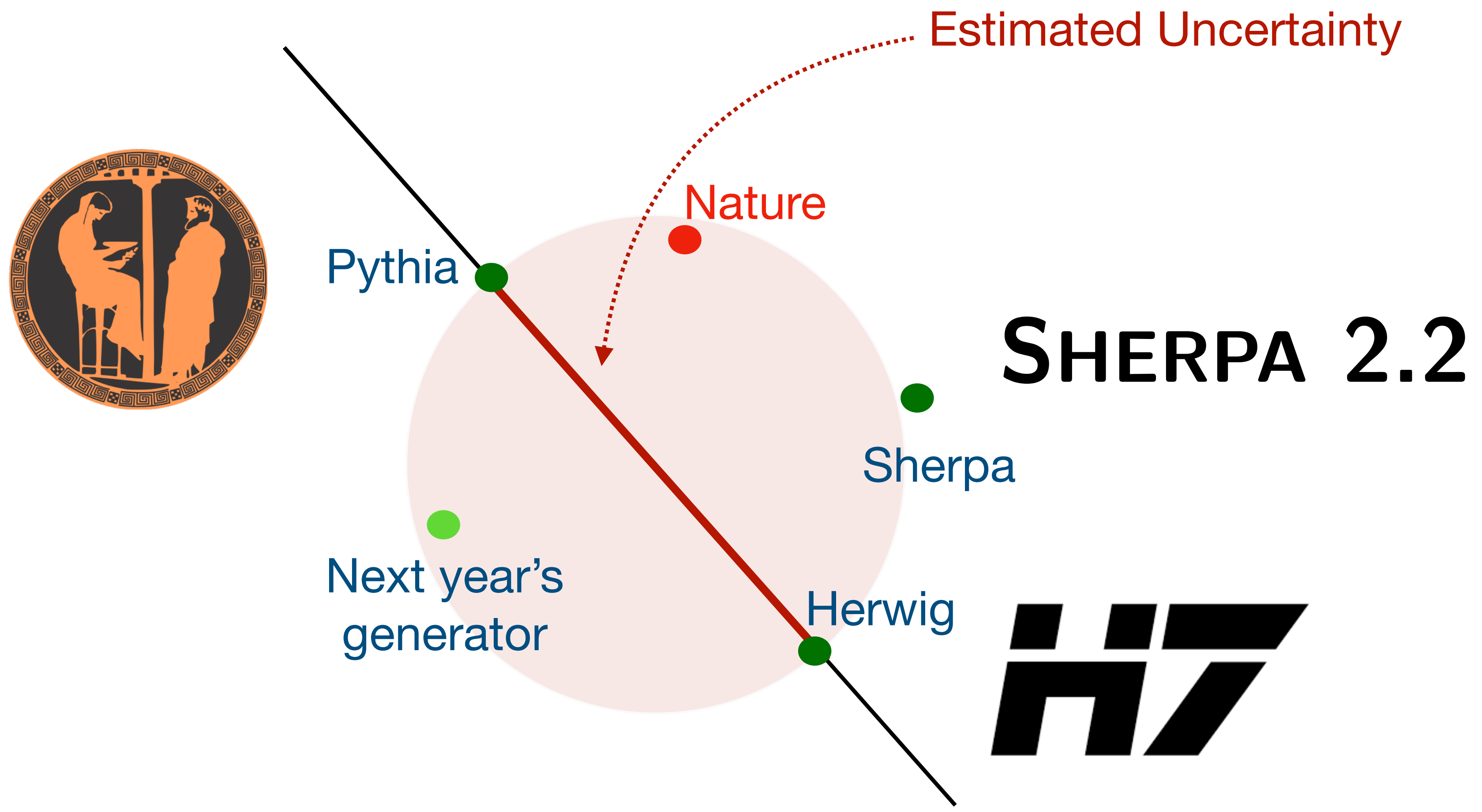


Image: arXiv:2109.08159

Systematic uncertainties

Experimental uncertainties:

Eg. Inaccuracies in the calibration of our detector

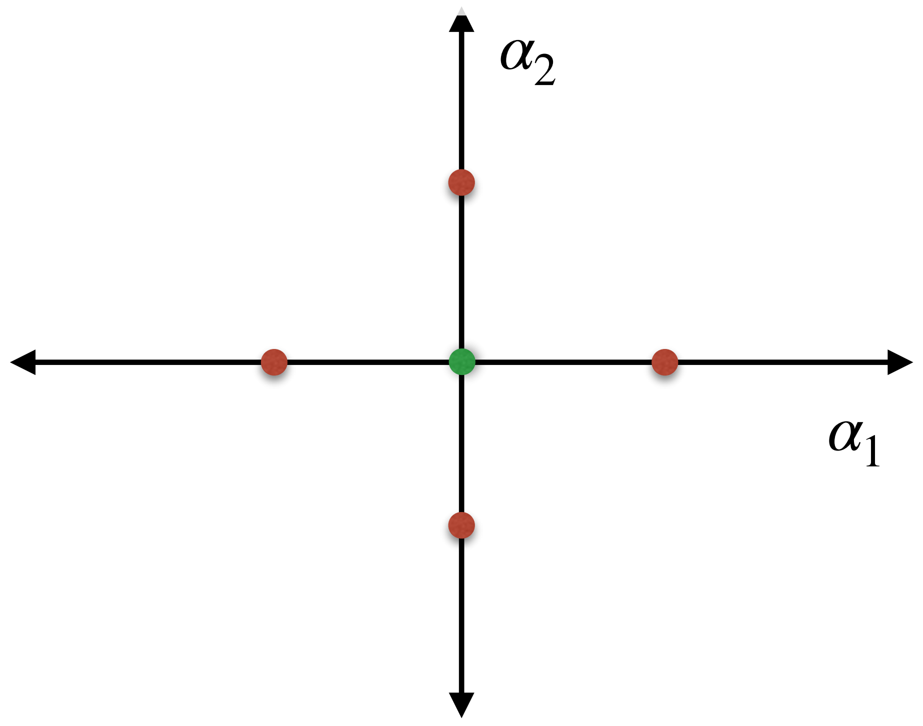
Image: arXiv:2105.08742

Theory uncertainties:

Eg. Inability to compute QFT to infinite order

Image: arXiv:2109.08159

- We only have simulations at 3 variations of each nuisance parameter α_k



Known interpolation strategies

See formula used

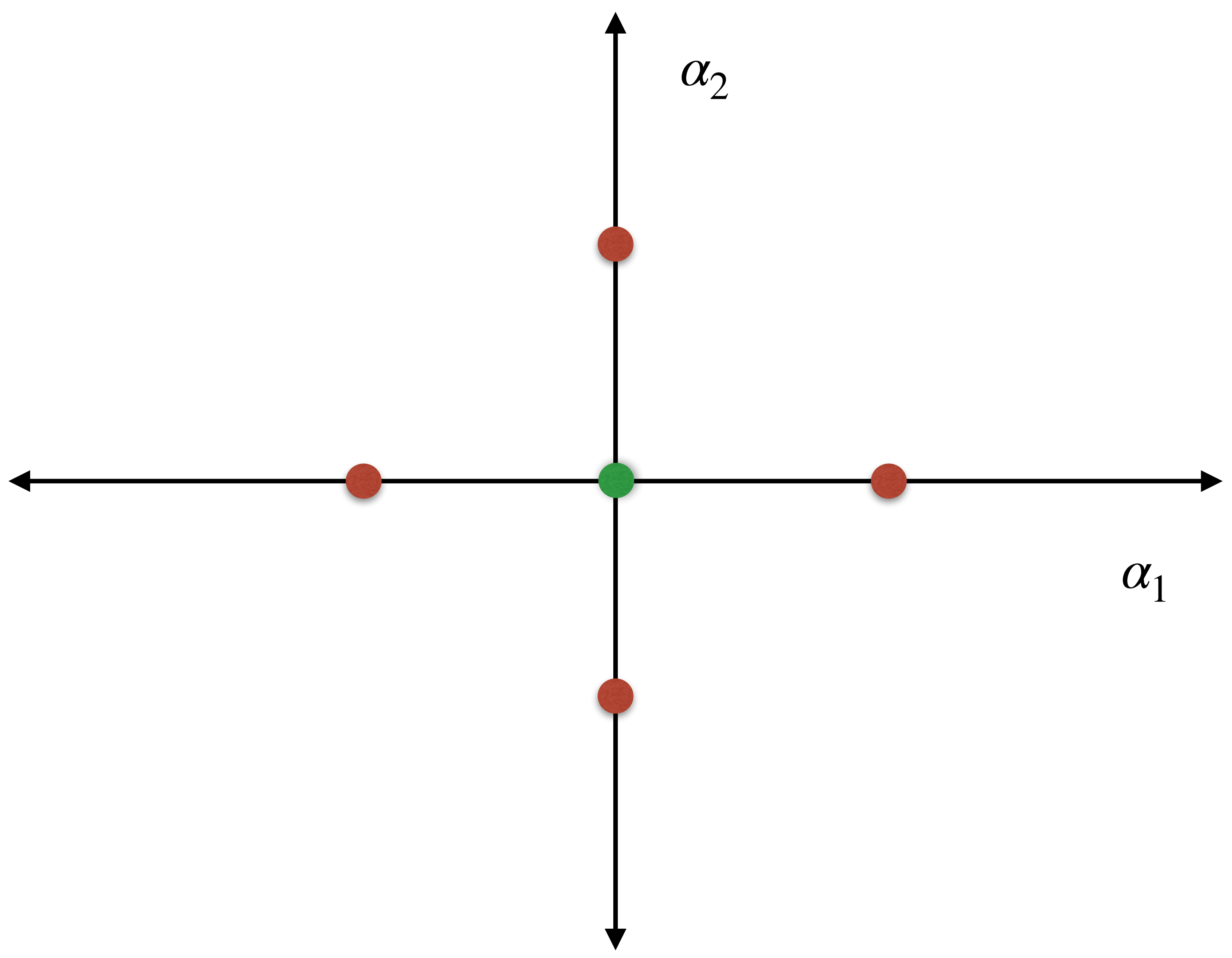
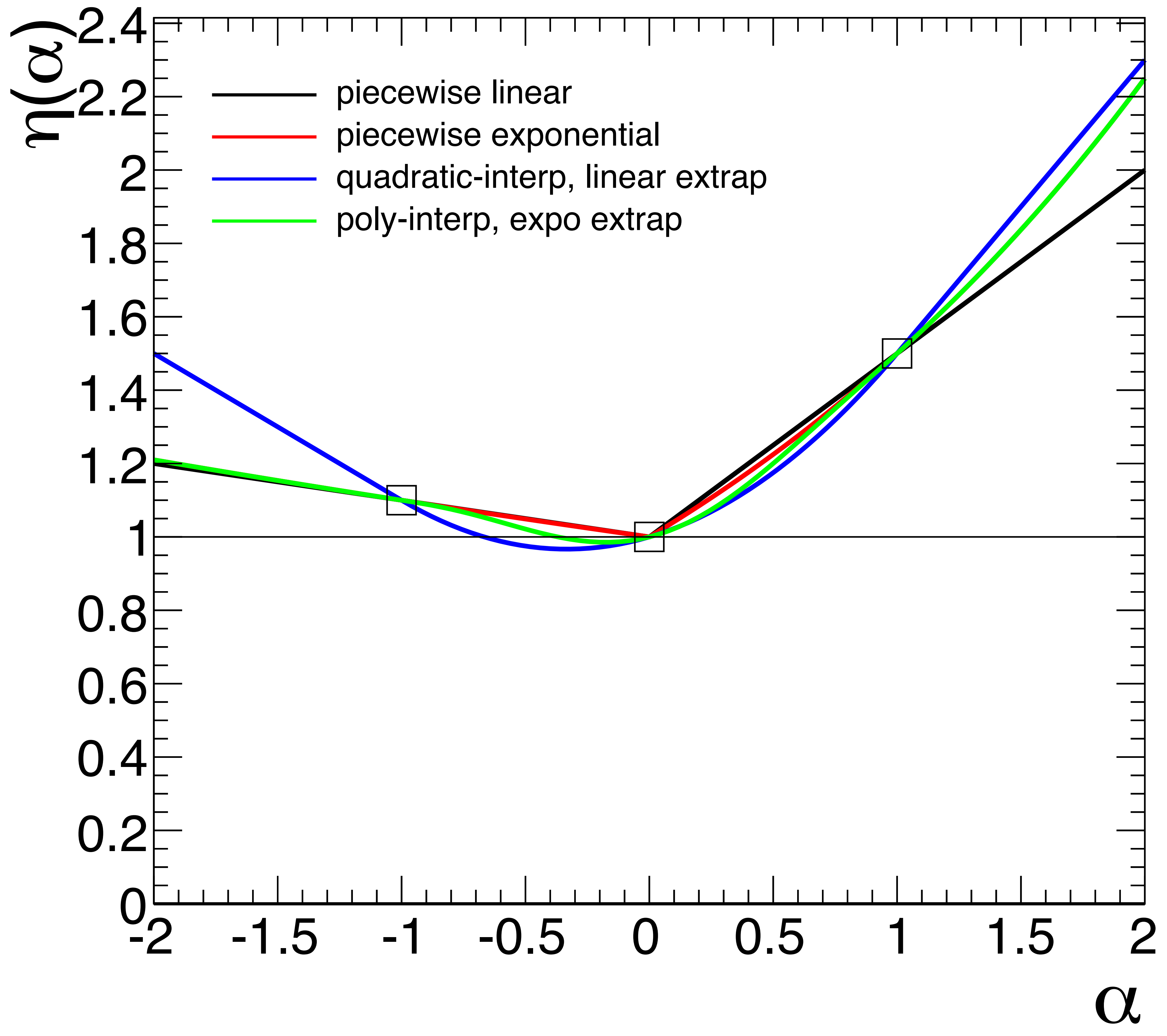


Image: arXiv:1503.07622

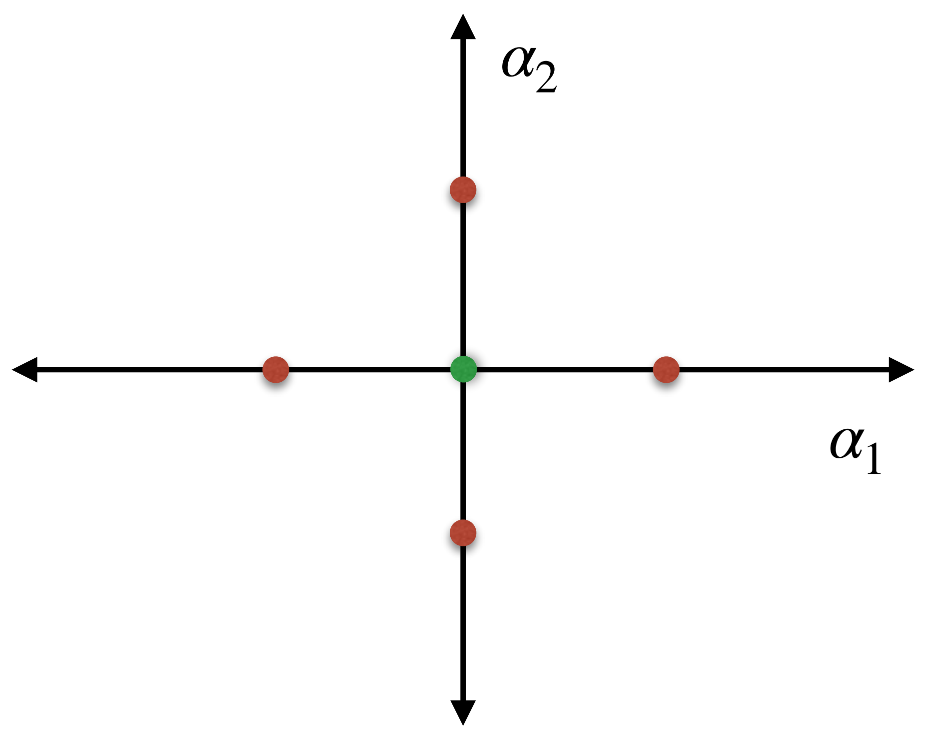


⇒ Combine these traditional interpolation with neural network estimation of per-event likelihood ratios

Probability density ratio including nuisance parameters (α)

x_i is one individual event

$$\frac{p(x_i \mid \mu, \alpha)}{p_{ref}(x_i)} =$$

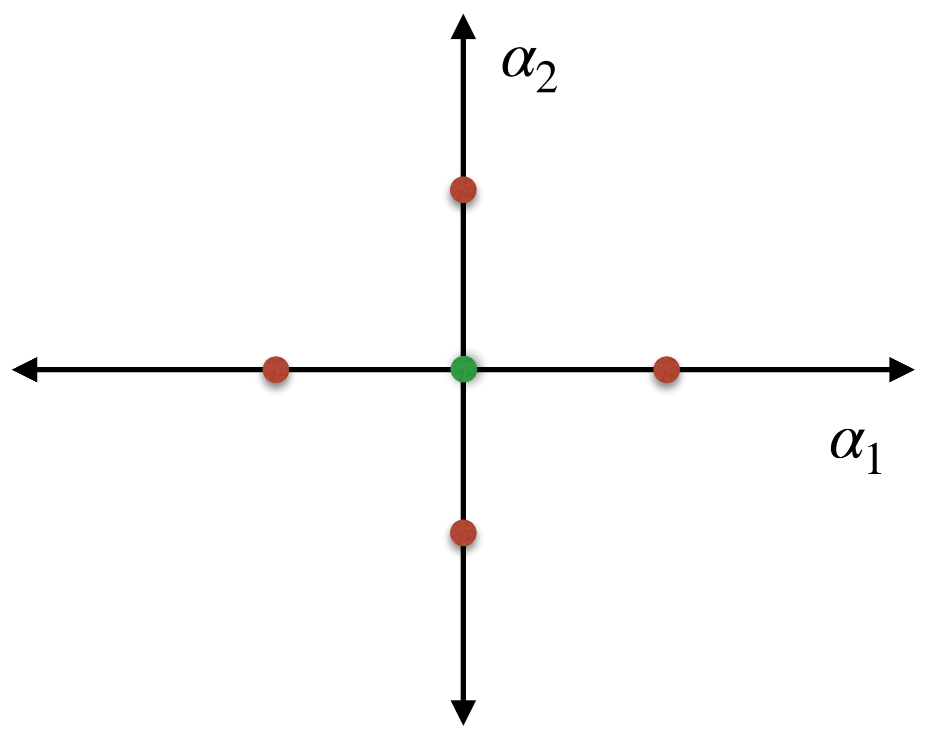


See details of vertical interpolation for $G_j(\alpha_k)$, $g_j(x_i, \alpha_k)$

Probability density ratio including nuisance parameters (α)

x_i is one individual event

$$\frac{p(x_i\,|\,\mu,\alpha)}{p_{ref}(x_i)} = \frac{1}{\nu(\mu,\alpha)} \sum_j^C f_j(\mu)\cdot\nu_j\cdot\frac{p_j(x_i)}{p_{ref}(x_i)}\cdot\prod_k^{N_{syst}} G_j(\alpha_k)\cdot g_j(x_i,\alpha_k)$$



$$g_j(x_i,\alpha_k) = \frac{p_j(x_i,\alpha_k)}{p_j(x_i)}$$

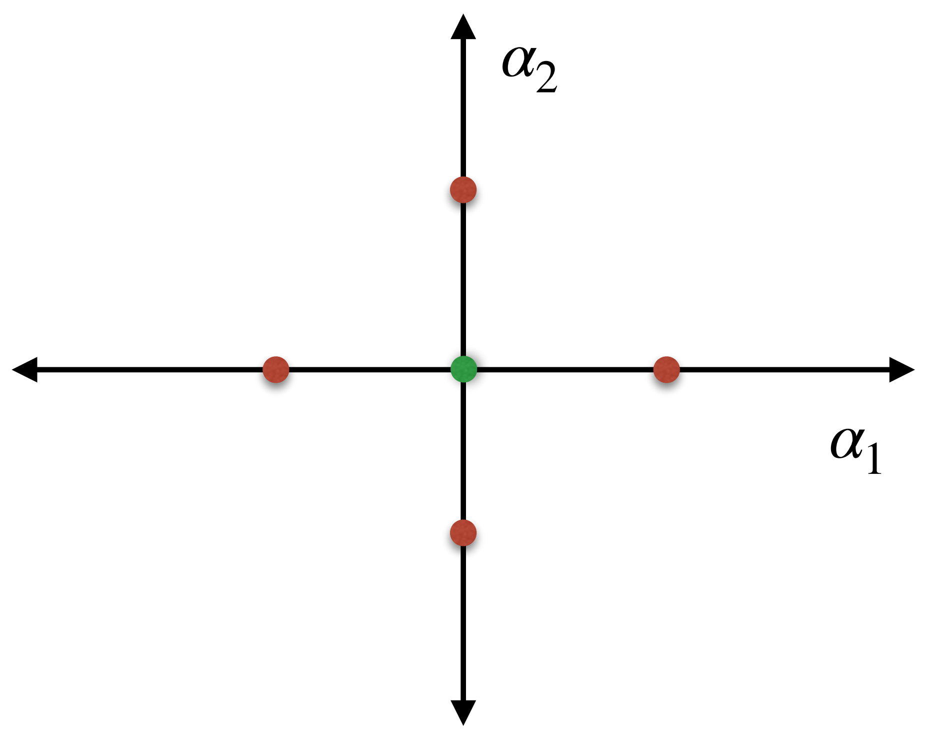
See details of vertical interpolation for $G_j(\alpha_k)$, $g_j(x_i,\alpha_k)$

Probability density ratio including nuisance parameters (α)

x_i is one individual event

$$\frac{p(x_i \mid \mu, \alpha)}{p_{ref}(x_i)} = \frac{1}{\nu(\mu, \alpha)} \sum_j^C f_j(\mu) \cdot \nu_j \cdot \frac{p_j(x_i)}{p_{ref}(x_i)} \cdot \prod_k^{N_{syst}} G_j(\alpha_k) \cdot g_j(x_i, \alpha_k)$$

We have this already



$$g_j(x_i, \alpha_k) = \frac{p_j(x_i, \alpha_k)}{p_j(x_i)}$$

See details of vertical interpolation for $G_j(\alpha_k), g_j(x_i, \alpha_k)$

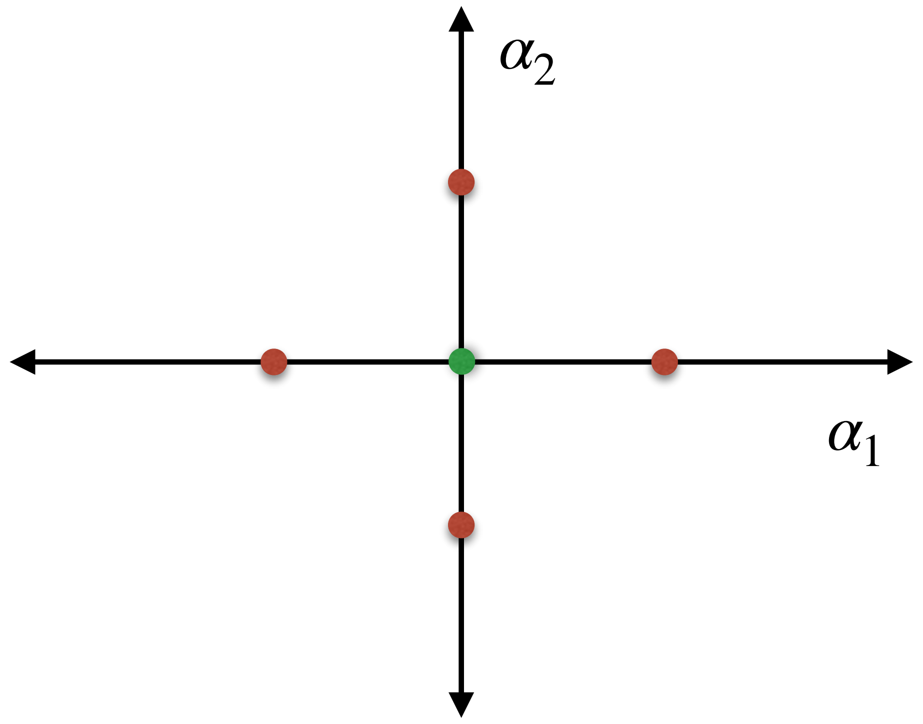
Probability density ratio including nuisance parameters (α)

x_i is one individual event

$$\frac{p(x_i \mid \mu, \alpha)}{p_{ref}(x_i)} = \frac{1}{\nu(\mu, \alpha)} \sum_j^C f_j(\mu) \cdot \nu_j \cdot \frac{p_j(x_i)}{p_{ref}(x_i)} \cdot \prod_k^{N_{syst}} G_j(\alpha_k) \cdot g_j(x_i, \alpha_k)$$

We have this already

Estimate from simulations and existing interpolation methods



$$g_j(x_i, \alpha_k) = \frac{p_j(x_i, \alpha_k)}{p_j(x_i)}$$

See details of vertical interpolation for $G_j(\alpha_k), g_j(x_i, \alpha_k)$

Probability density ratio including nuisance parameters (α)

x_i is one individual event

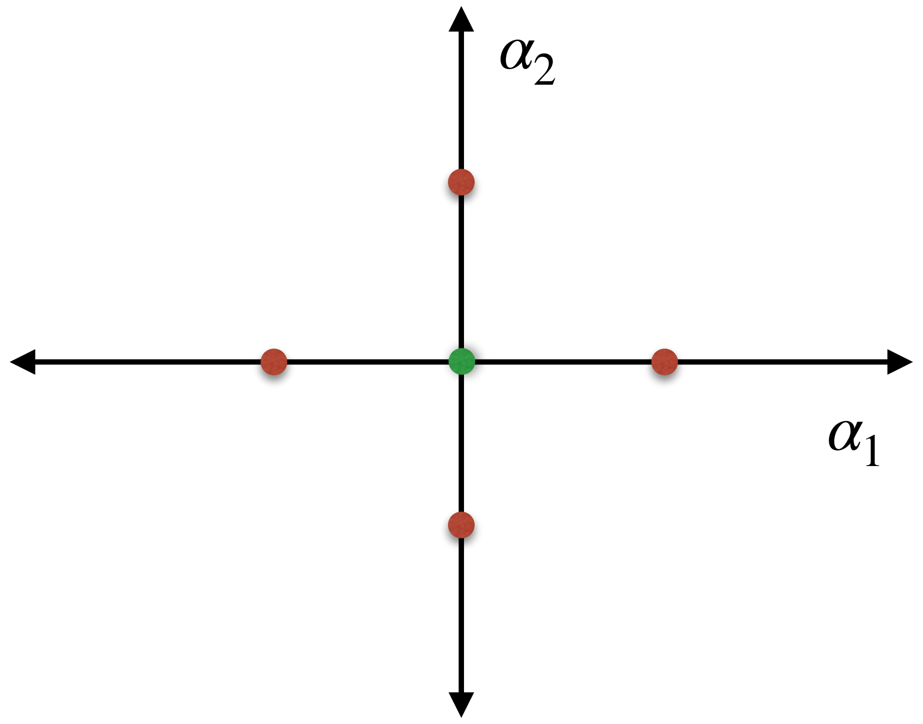
$$\frac{p(x_i\,|\,\mu,\alpha)}{p_{ref}(x_i)} = \frac{1}{\nu(\mu,\alpha)} \sum_{j}^{C} f_j(\mu)\cdot\nu_j\cdot\frac{p_j(x_i)}{p_{ref}(x_i)}\cdot\prod_{k}^{N_{syst}} G_j(\alpha_k)\cdot g_j(x_i,\alpha_k)$$

We have this already

Per-event terms estimated using another ensemble of networks and interpolation methods

$$g_j(x_i,\alpha_k) = \frac{p_j(x_i,\alpha_k)}{p_j(x_i)}$$

Estimate from simulations and existing interpolation methods



See details of vertical interpolation for $G_j(\alpha_k)$, $g_j(x_i,\alpha_k)$

Final test statistic

x_i is one individual event

$$\frac{L_{\text{full}}(\mu, \alpha|\mathcal{D})}{L_{\text{ref}}(\mathcal{D})} = \text{Pois}(N_{\text{data}}|\nu(\mu, \alpha)) \prod_{i}^{N_{\text{data}}} \frac{p(x_i|\mu, \alpha)}{p_{\text{ref}}(x_i)} \prod_{k} \text{Gaus}(a_k|\alpha_k, \delta_k)$$

Final test statistic

x_i is one individual event

$$\frac{L_{\text{full}}(\mu, \alpha|\mathcal{D})}{L_{\text{ref}}(\mathcal{D})} = \text{Pois}(N_{\text{data}}|\nu(\mu, \alpha)) \prod_{i}^{N_{\text{data}}} \frac{p(x_i|\mu, \alpha)}{p_{\text{ref}}(x_i)} \prod_{k} \text{Gaus}(a_k|\alpha_k, \delta_k)$$

From previous slide

Final test statistic

x_i is one individual event

$$\frac{L_{\text{full}}(\mu, \alpha|\mathcal{D})}{L_{\text{ref}}(\mathcal{D})} = \text{Pois}(N_{\text{data}}|\nu(\mu, \alpha)) \prod_{i}^{N_{\text{data}}} \frac{p(x_i|\mu, \alpha)}{p_{\text{ref}}(x_i)} \prod_{k} \text{Gaus}(a_k|\alpha_k, \delta_k)$$

From previous slide

Prod over events

Final test statistic

x_i is one individual event

$$\frac{L_{\text{full}}(\mu, \alpha | \mathcal{D})}{L_{\text{ref}}(\mathcal{D})} = \text{Pois}(N_{\text{data}} | \nu(\mu, \alpha)) \prod_i^{N_{\text{data}}} \frac{p(x_i | \mu, \alpha)}{p_{\text{ref}}(x_i)} \prod_k \text{Gaus}(a_k | \alpha_k, \delta_k)$$

From previous slide

Rate term

Prod over events

Final test statistic

x_i is one individual event

$$\frac{L_{\text{full}}(\mu,\alpha|\mathcal{D})}{L_{\text{ref}}(\mathcal{D})} = \text{Pois}(N_{\text{data}}|\nu(\mu,\alpha)) \prod_{i}^{N_{\text{data}}} \frac{p(x_i|\mu,\alpha)}{p_{\text{ref}}(x_i)} \prod_{k} \text{Gaus}(a_k|\alpha_k,\delta_k)$$

From previous slide

Rate term

Prod over events

Constrain term

Final test statistic

x_i is one individual event

$$\frac{L_{\text{full}}(\mu,\alpha|\mathcal{D})}{L_{\text{ref}}(\mathcal{D})} = \text{Pois}(N_{\text{data}}|\nu(\mu,\alpha)) \prod_{i}^{N_{\text{data}}} \frac{p(x_i|\mu,\alpha)}{p_{\text{ref}}(x_i)} \prod_k \text{Gaus}(a_k|\alpha_k,\delta_k)$$

From previous slide

Rate term

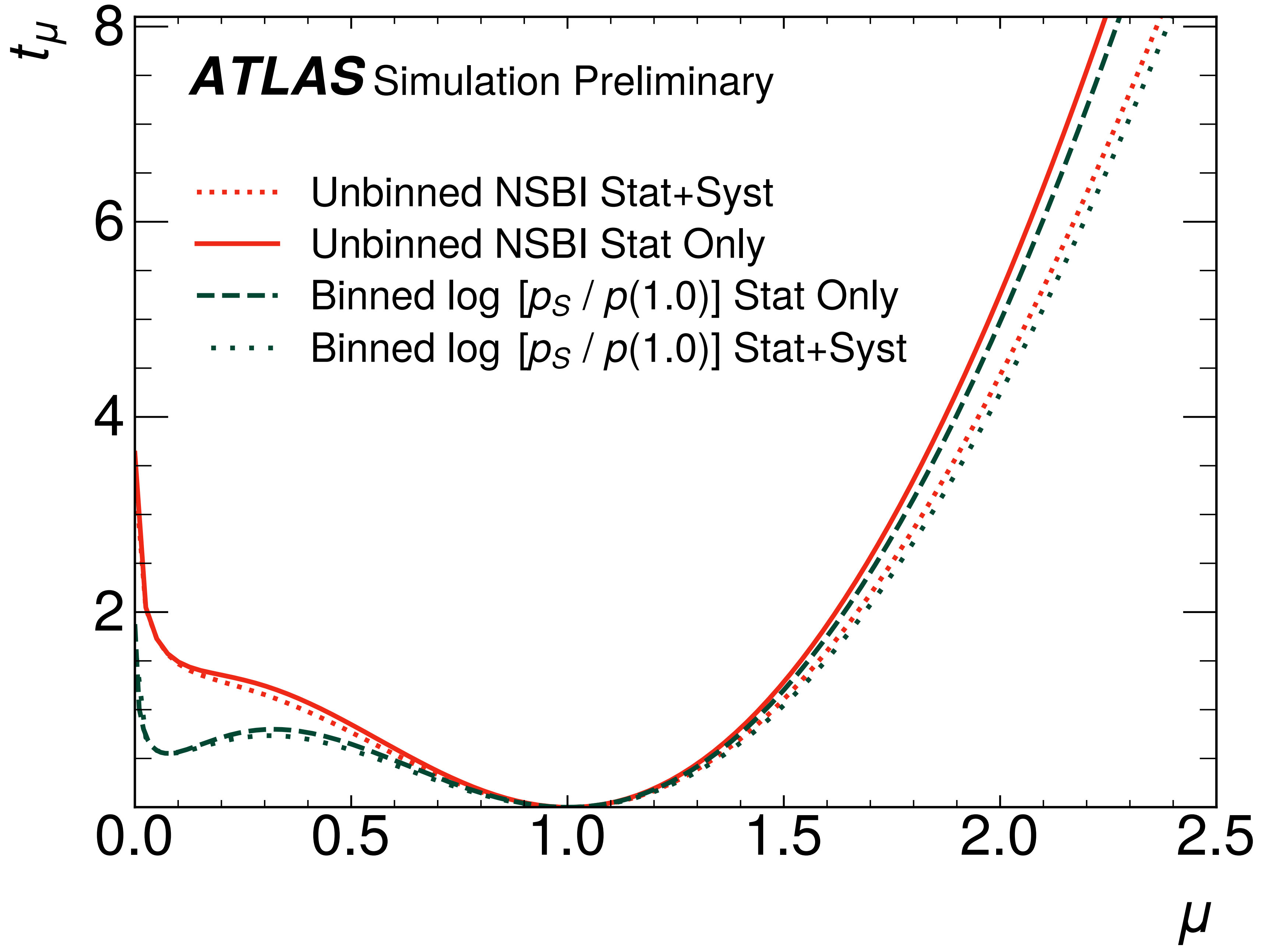
Prod over events

Constrain term

Profiling:

$$t_\mu = -2\ln\left(\frac{L_{\text{full}}(\mu,\hat{\hat{\alpha}})/\cancel{L_{\text{ref}}}}{L_{\text{full}}(\hat{\mu},\hat{\alpha})/\cancel{L_{\text{ref}}}}\right)$$

This is why we define p_{ref} to be independent of μ



Final test statistic

x_i is one individual event

$$\frac{L_{\text{full}}(\mu,\alpha|\mathcal{D})}{L_{\text{ref}}(\mathcal{D})} = \text{Pois}(N_{\text{data}}|\nu(\mu,\alpha)) \prod_i^{N_{\text{data}}} \frac{p(x_i|\mu,\alpha)}{p_{\text{ref}}(x_i)} \prod_k \text{Gaus}(a_k|\alpha_k,\delta_k)$$

From previous slide

Rate term

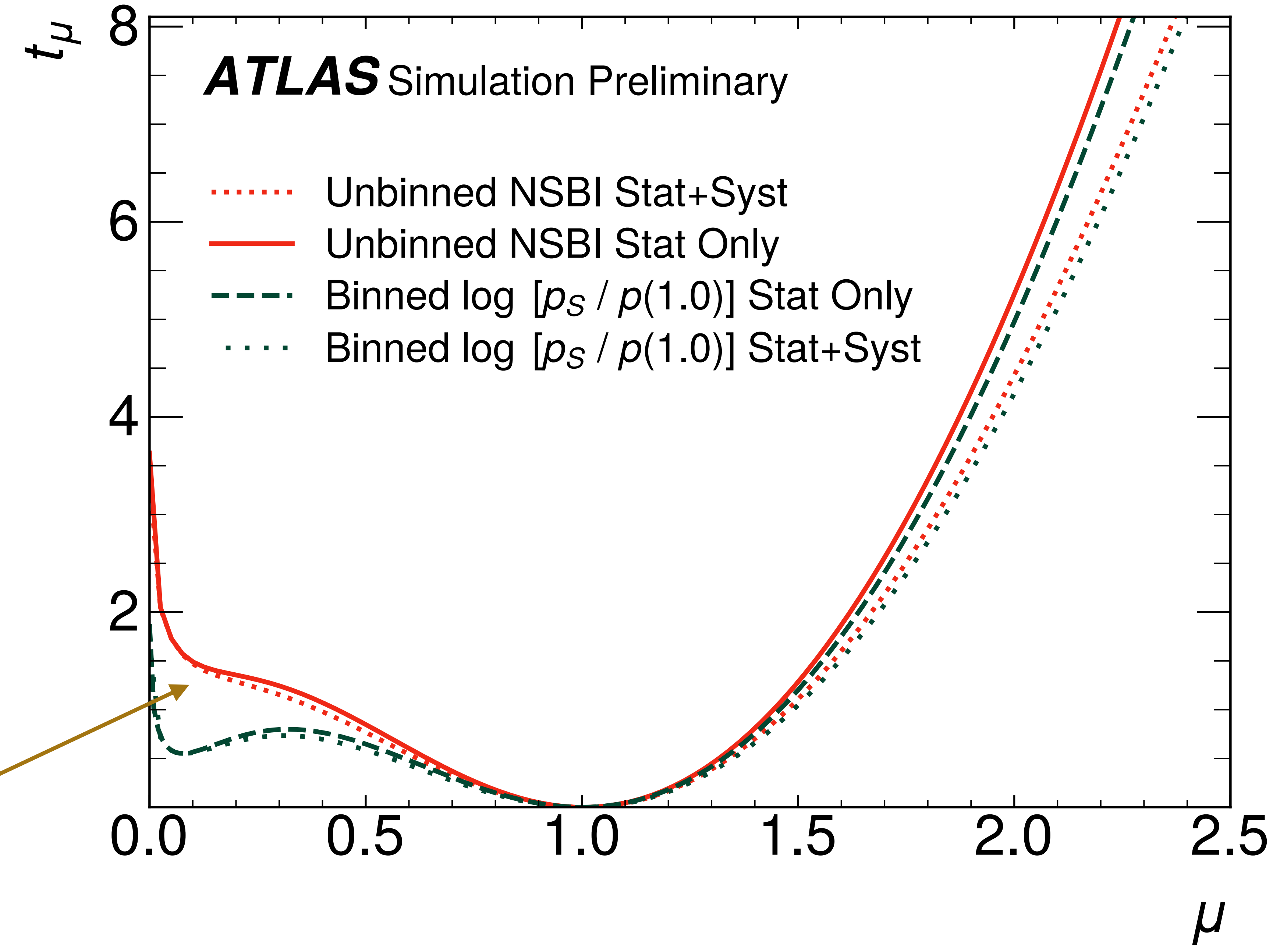
Prod over events

Constrain term

Profiling:

$$t_\mu = -2\ln\left(\frac{L_{\text{full}}(\mu,\hat{\hat{\alpha}})/\cancel{L_{\text{ref}}}}{L_{\text{full}}(\hat{\mu},\hat{\alpha})/\cancel{L_{\text{ref}}}}\right)$$

This is why we define p_{ref} to be independent of μ



Non-parabolic shape due to non-linear effects from quantum interference

Reference Sample

A combination of signal samples, to ensure there's non-vanishing support entire region of analysis
Does not have to be physical!

$$p_{\text{ref}}(x_i) = \frac{1}{\sum_k \nu_k} \sum_{k}^{C_{\text{signals}}} \nu_k \cdot p_k(x_i)$$

⇒ In our dataset, $p_{ref}(\cdot) = p_S(\cdot)$

Choice of $p_{ref}(\cdot)$ can be made purely on numerical stability of training, as it drops out in profile step

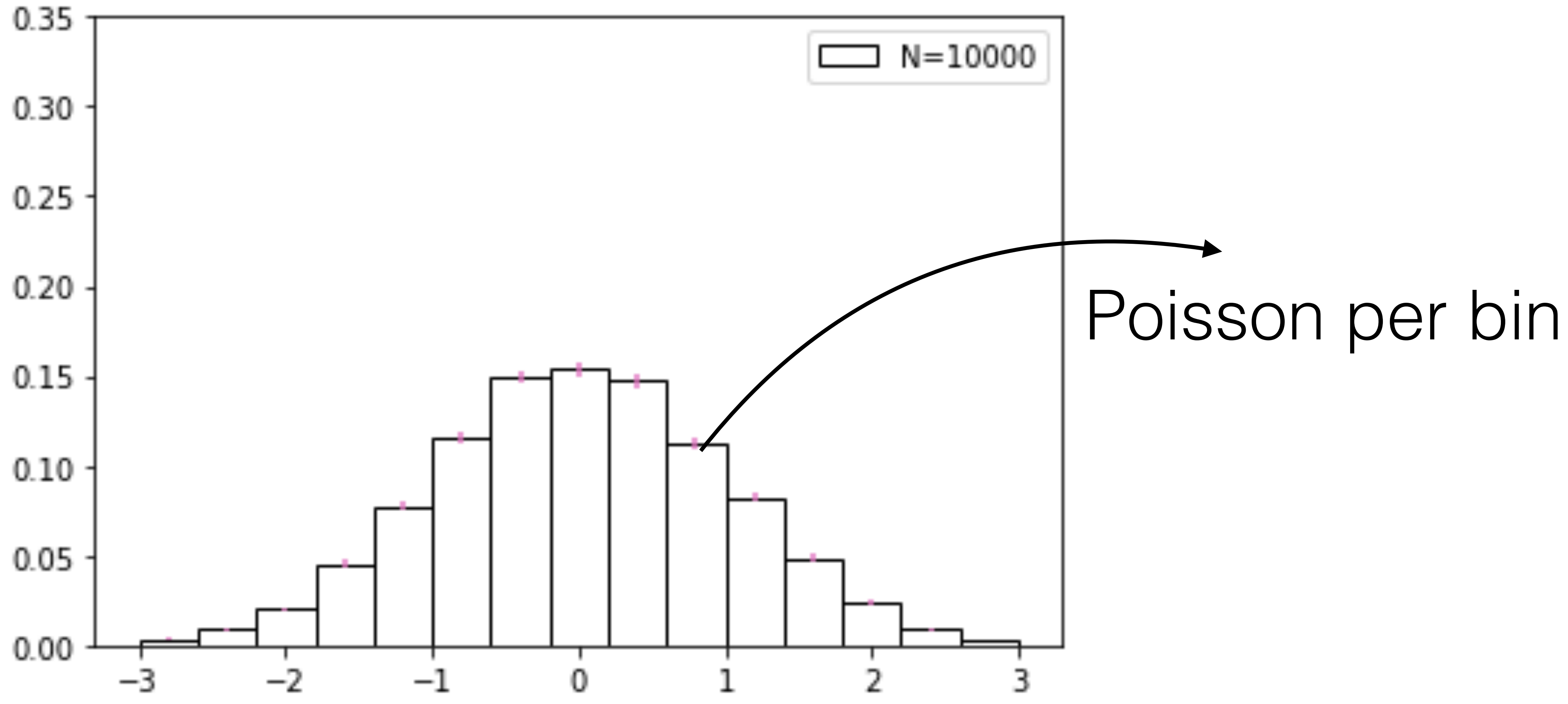
$$t_\mu = -2 \ln \left(\frac{L_{\text{full}}(\mu, \widehat{\widehat{\alpha}}) / \cancel{L_{\text{ref}}}}{L_{\text{full}}(\widehat{\mu}, \widehat{\alpha}) / \cancel{L_{\text{ref}}}} \right)$$

Open problems to extend to full ATLAS analysis:

- ✓ Robustness: Design and validation
- ✓ Systematic Uncertainties: Incorporate them in likelihood (ratio) model
- ‣ Neyman Construction: Throwing toys in a per-event analysis

Throwing event-level toys

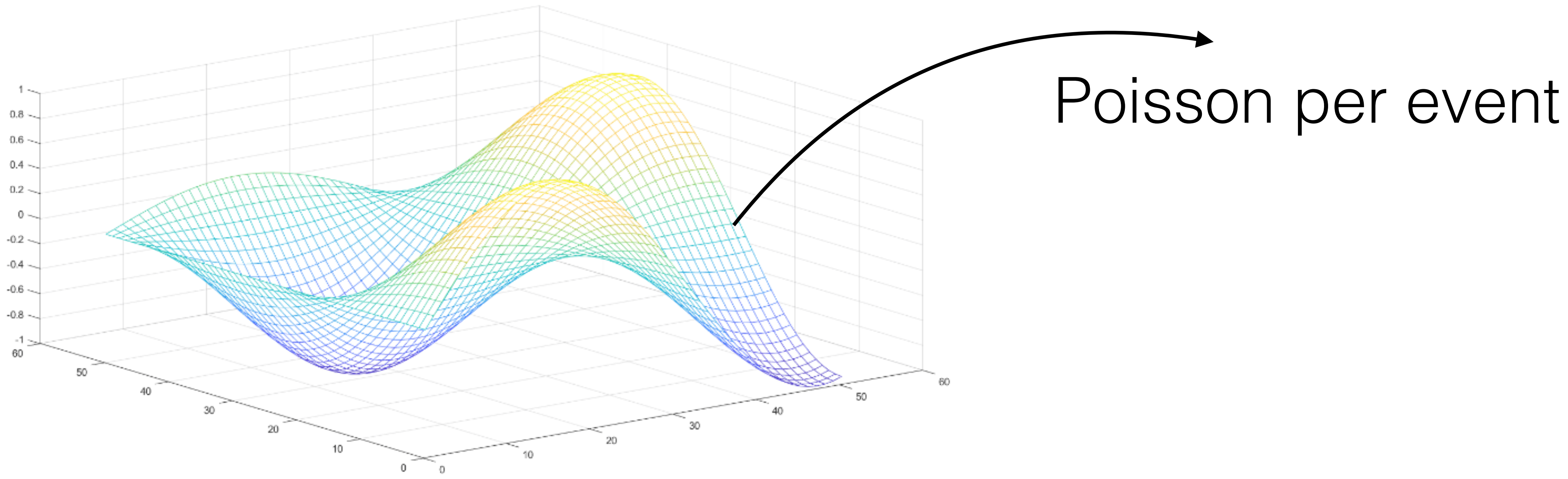
Traditionally:



Asimov Histogram

$$N_i^{toy} = Poisson(N_i^{Asimov})$$

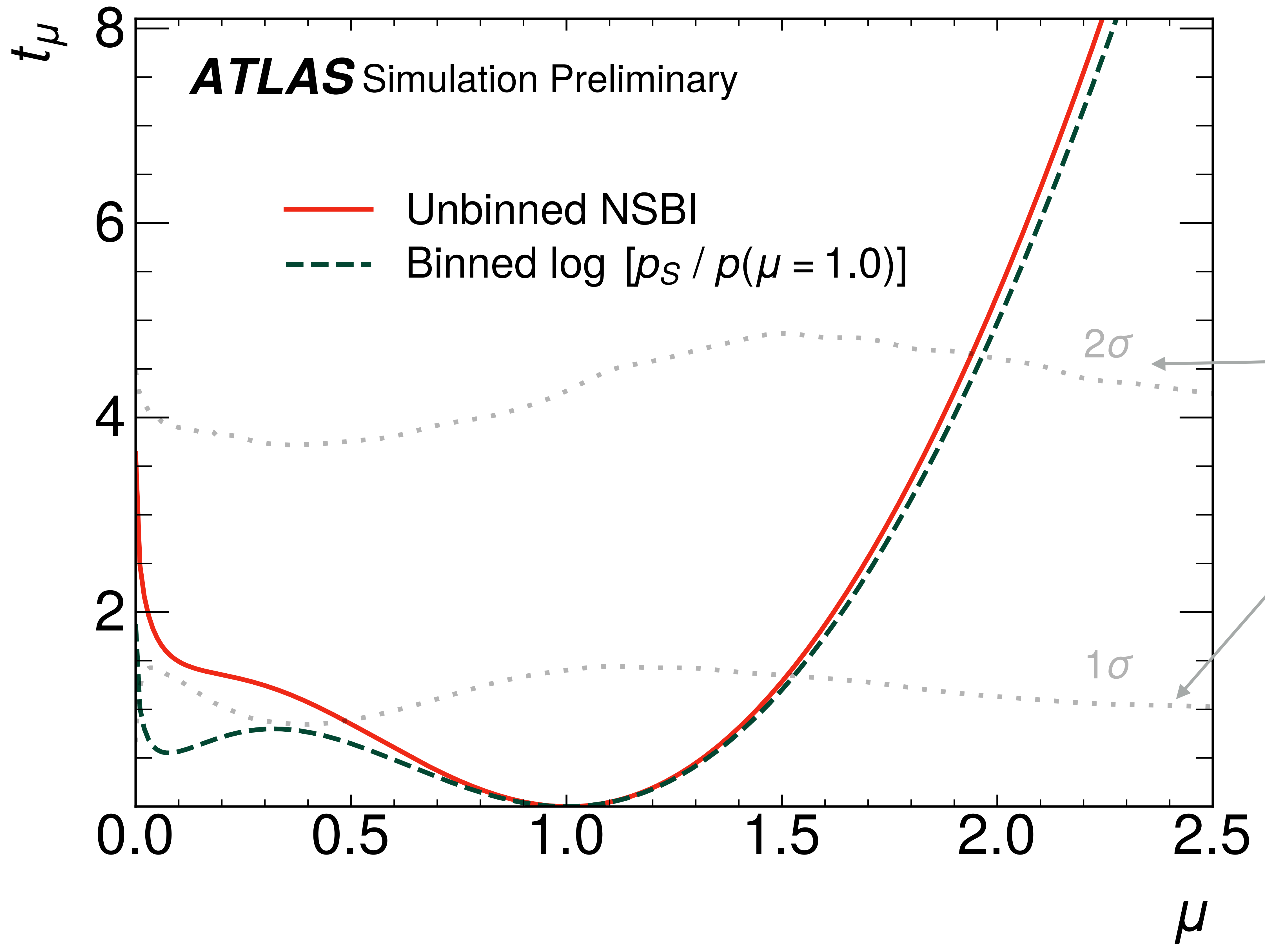
NSBI:



$$w_i^{toy} = Poisson(w_i^{Asimov})$$

('Unweighted' events, i.e. integer weights)

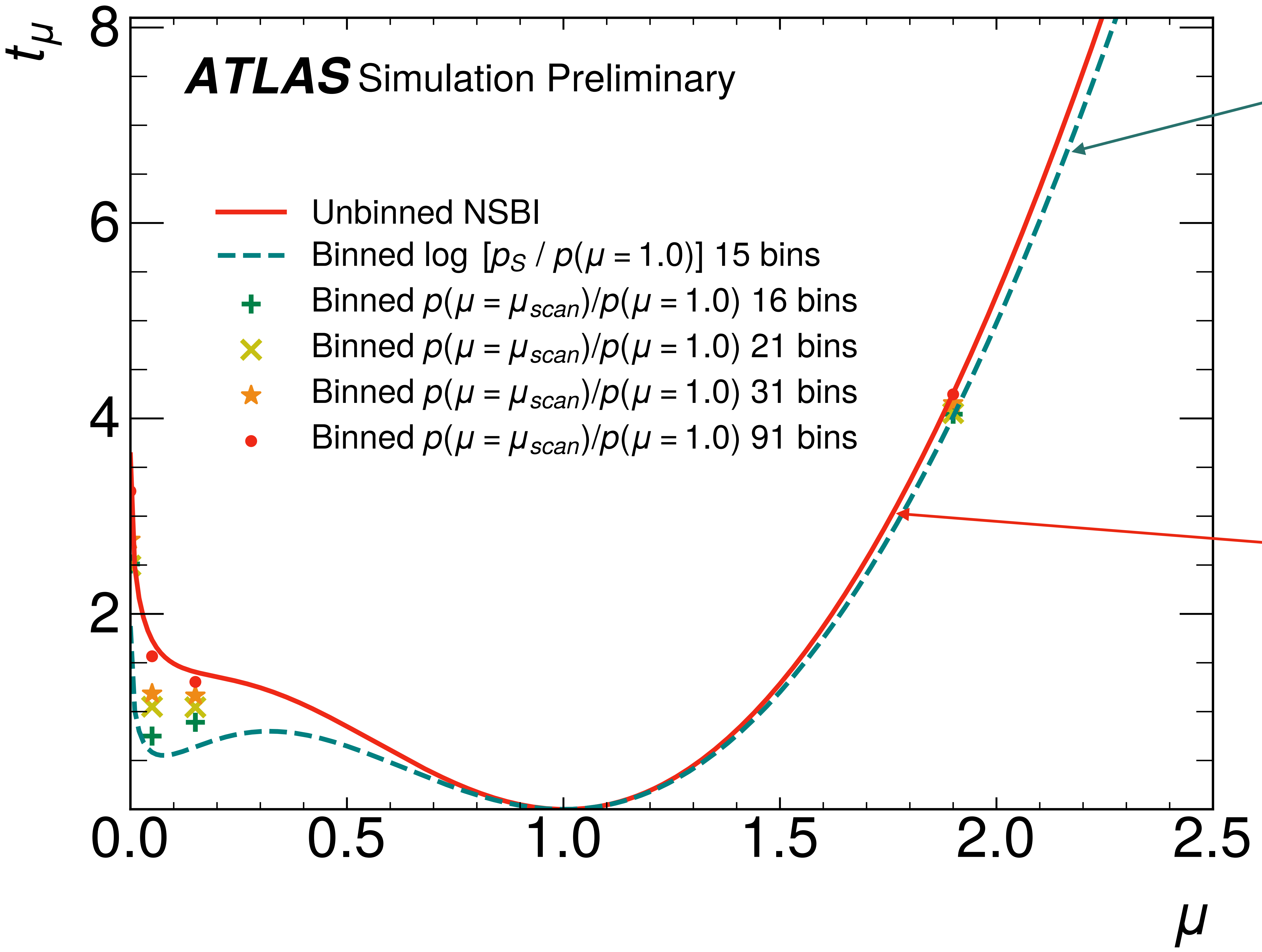
Confidence belts



Similar to structure seen in histogram analysis

Why does NSBI work better than traditional analyses?

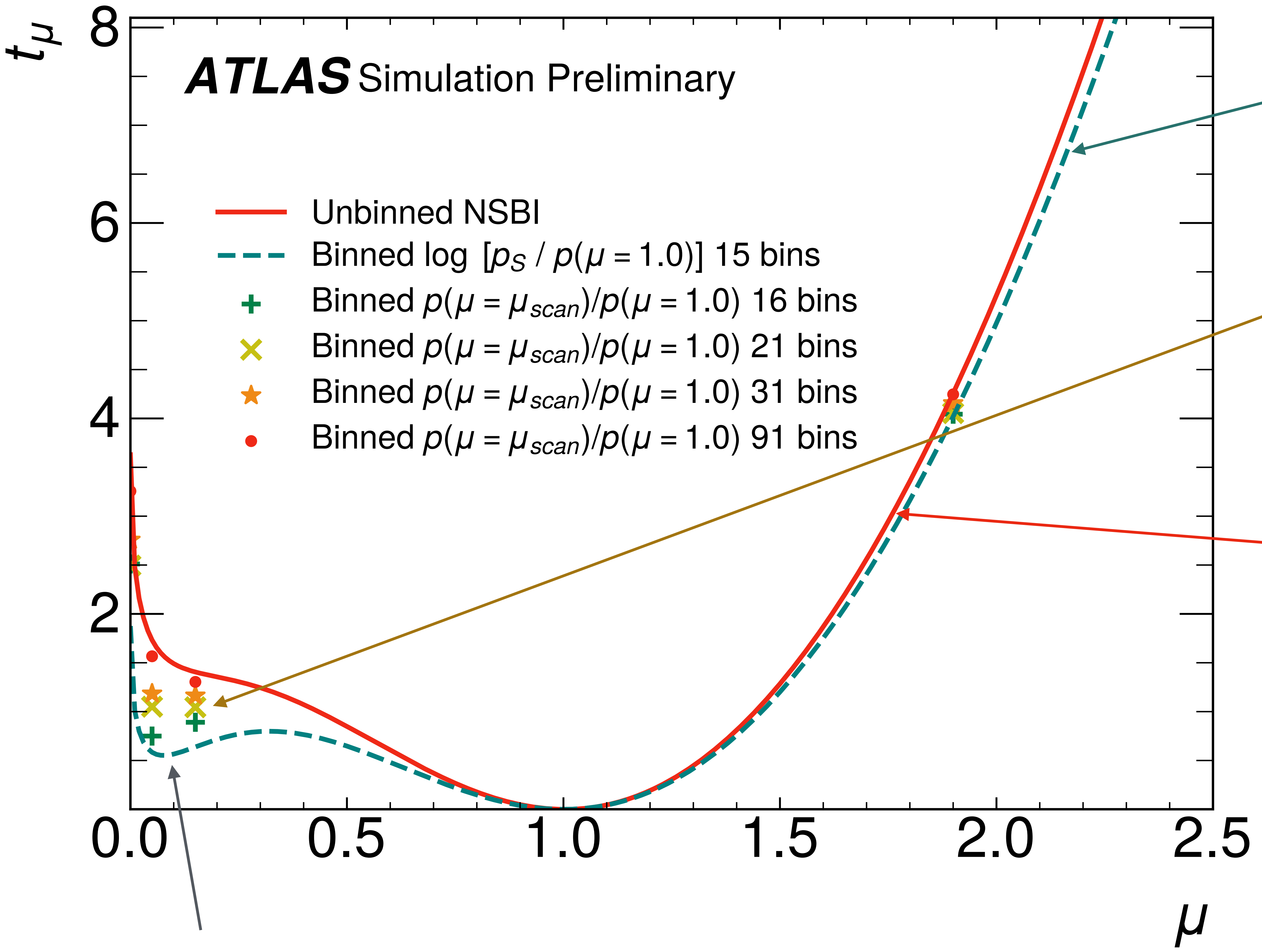
Why does it work better than traditional analyses?



$O_{fixed} = \log \frac{p_S(x_i)}{p_{SBI}(x_i)}$: Similar to histogram analysis

NSBI: Parameterised, unbinned

Why does it work better than traditional analyses?



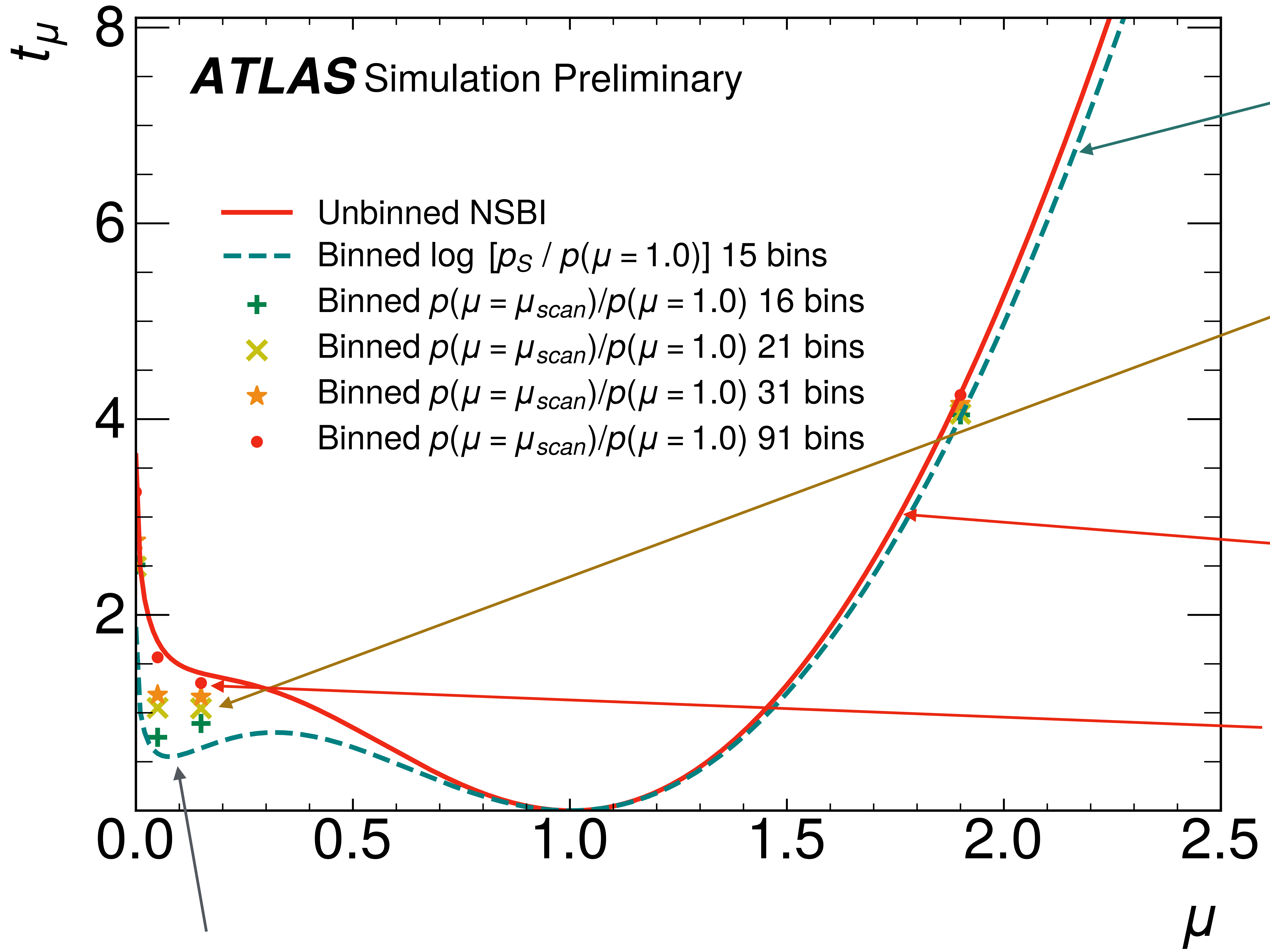
$O_{\text{fixed}} = \log \frac{p_S(x_i)}{p_{SBI}(x_i)}$: Similar to histogram analysis

$O_\mu = \frac{p(x_i \mid \mu)}{p(x_i \mid \mu = 1)}$: Parameterised observable, histogram fit

NSBI: Parameterised, unbinned

Significant improvement in QI impacted region

Why does it work better than traditional analyses?



$O_{\text{fixed}} = \log \frac{p_S(x_i)}{p_{SBI}(x_i)}$: Similar to histogram analysis

$O_\mu = \frac{p(x_i|\mu)}{p(x_i|\mu = 1)}$: Parameterised observable, histogram fit

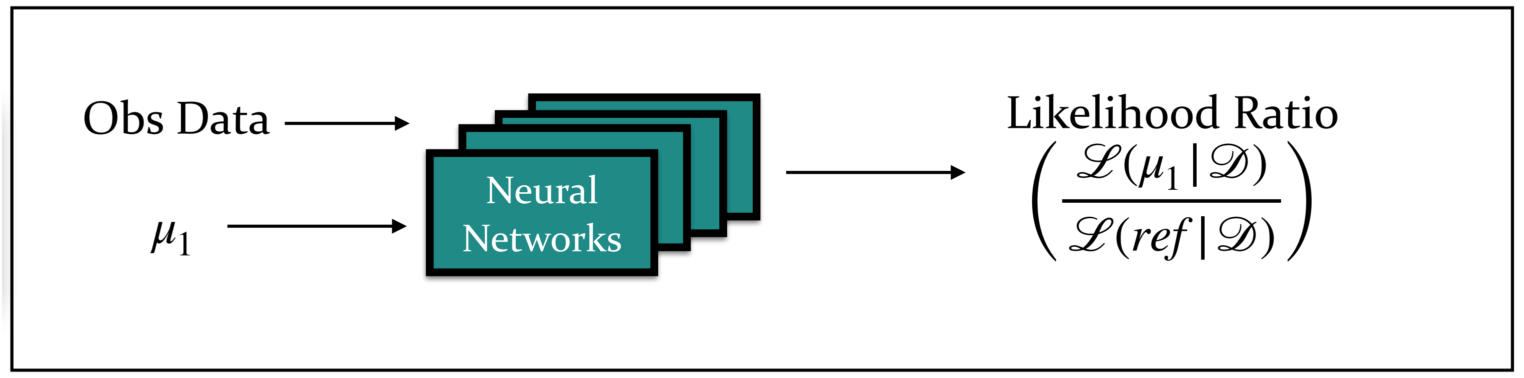
NSBI: Parameterised, unbinned

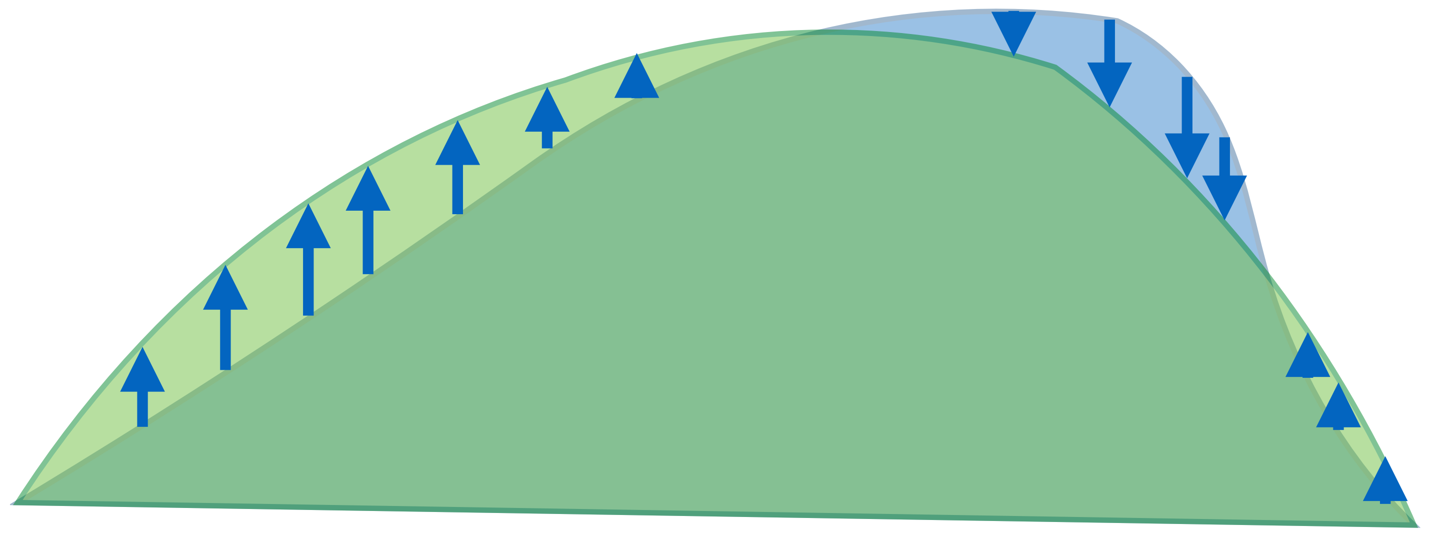
O_μ approaches NSBI as nBins $\to \infty$

Significant improvement in QI impacted region

Conclusion

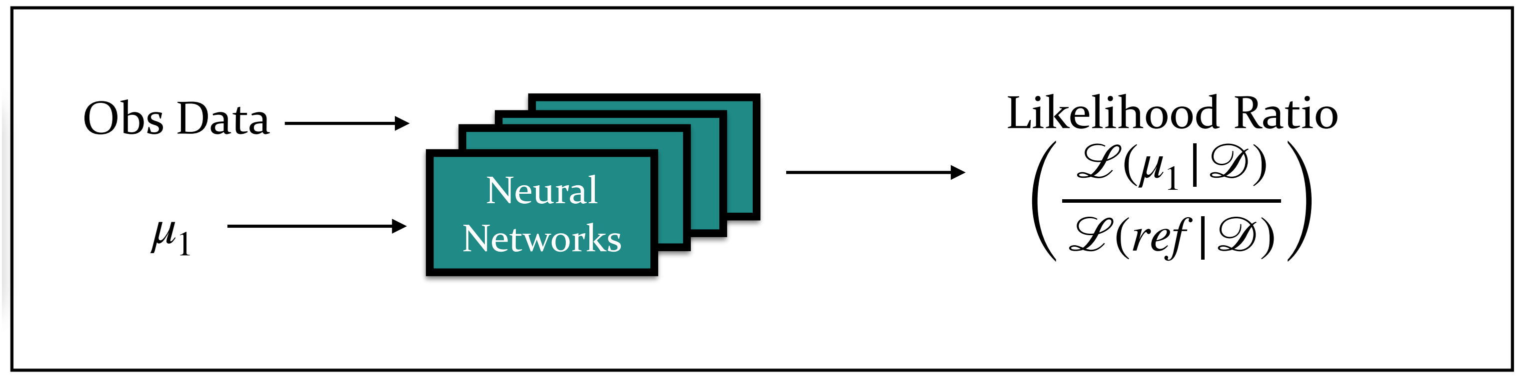
- Developed a complete statistical framework for high-dimensional statistical inference
  - Builds upon traditional methodology in ATLAS
  - Developed diagnostic tools for validation
- Such methods are crucial for analyses where kinematic distributions change non-linearly with the parameter of interest, eg. EFT studies
- Weaknesses: Same as traditional analyses, requires well trained networks

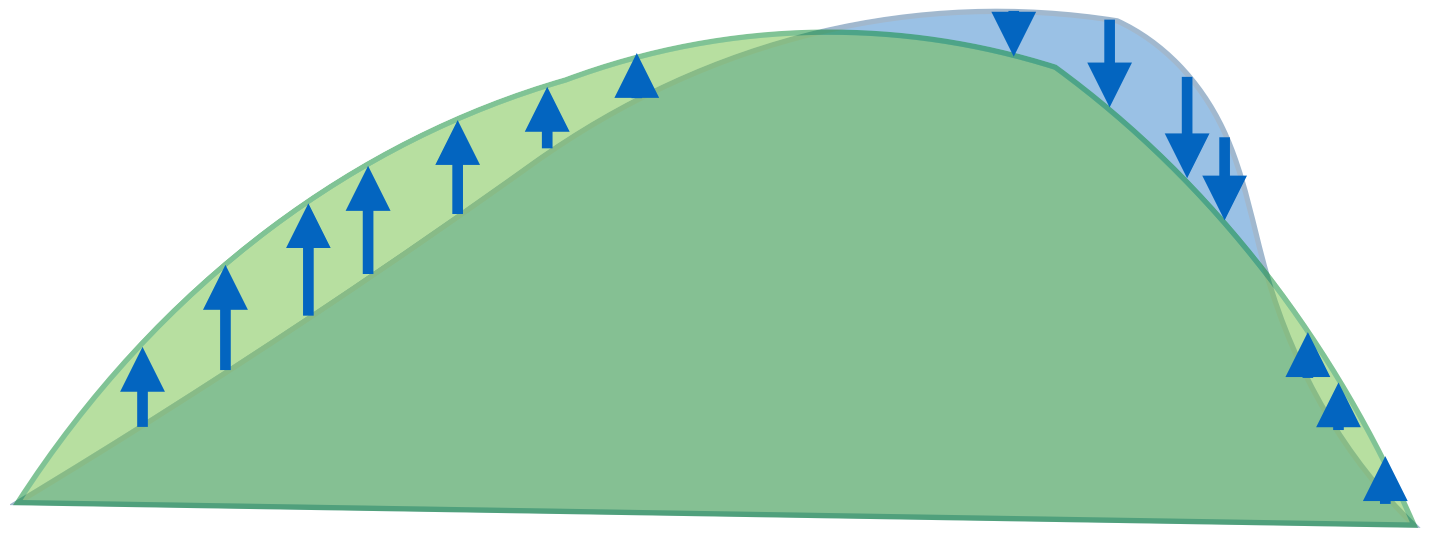




Conclusion

- Developed a complete statistical framework for high-dimensional statistical inference
  - Builds upon traditional methodology in ATLAS
  - Developed diagnostic tools for validation
- Such methods are crucial for analyses where kinematic distributions change non-linearly with the parameter of interest, eg. EFT studies
- Weaknesses: Same as traditional analyses, requires well trained networks





Thanks !

Backup

Choice of observable

Choice of observable

$$\mathscr{L}(\mu \,|\, \mathscr{D}) = p(\mathscr{D} \,|\, \mu)$$

Neyman–Pearson lemma: Likelihood ratio is the most powerful test statistic

We want to compare likelihoods: $$\frac{p(\mathscr{D} \,|\, \mu)}{p(\mathscr{D} \,|\, \mu_0)}$$

Choice of observable

$$\mathscr{L}(\mu \,|\, \mathscr{D}) = p(\mathscr{D} \,|\, \mu)$$

Neyman–Pearson lemma: Likelihood ratio is the most powerful test statistic

We want to compare likelihoods:
$$\frac{p(\mathscr{D} \,|\, \mu)}{p(\mathscr{D} \,|\, \mu_0)}$$

Choice of observable

$$\mathscr{L}(\mu \,|\, \mathscr{D}) = p(\mathscr{D} \,|\, \mu)$$

Neyman–Pearson lemma: Likelihood ratio is the most powerful test statistic

We want to compare likelihoods:

$$\frac{p(\mathscr{D} \,|\, \mu)}{p(\mathscr{D} \,|\, \mu_0)}$$

A neural network classifier trained on S vs B, estimates the decision function*:

$$s(x_i) = \frac{p(x_i \,|\, S)}{p(x_i \,|\, S) + p(x_i \,|\, B)}$$

* Equal class weights

Choice of observable

$$\mathscr{L}(\mu \,|\, \mathscr{D}) = p(\mathscr{D} \,|\, \mu)$$

Neyman–Pearson lemma: **Likelihood ratio is the most powerful test statistic**

We want to compare likelihoods: $$\frac{p(\mathscr{D} \,|\, \mu)}{p(\mathscr{D} \,|\, \mu_0)}$$

A neural network classifier trained on S vs B, estimates the decision function*: $$s(x_i) = \frac{p(x_i \,|\, S)}{p(x_i \,|\, S) + p(x_i \,|\, B)}$$

Which contains all the information required for the likelihood ratio:

$$\frac{p(x_i \,|\, \mu)}{p(x_i \,|\, \mu = 0)} = \frac{\mu \cdot \sigma_S \cdot p(x_i \,|\, S) + \sigma_B \cdot p(x_i \,|\, B)}{\sigma_B \cdot p(x_i \,|\, B)} = \mu \cdot \frac{\sigma_S}{\sigma_B} \cdot \frac{s(x_i)}{1 - s(x_i)} + 1.$$

Same observable s is optimal to test all μ hypotheses!

No need to develop separate analysis per hypothesis μ

* Equal class weights

What breaks down?

$$P(X) = |M_s(X) + M_b(X)|^2 = \underbrace{|M_s(X)|^2}_{P_s(X)} + \underbrace{|M_b(X)|^2}_{P_b(X)} + \underbrace{2\,\mathrm{Re}(\overline{M_s(X)}M_b(X))}_{P_i(X)}$$

$$N_{exp} = \mu \cdot S + B + \sqrt{\mu} \cdot I$$

A neural network classifier trained on S vs B, estimates the decision function: $s(x_i) = \frac{p(x_i|S)}{p(x_i|S) + p(x_i|B)}$

Which contains all the information required for the likelihood ratio:

$$\frac{p(x_i|\mu)}{p(x_i|\mu=0)} = \frac{\mu \cdot \sigma_S \cdot p(x_i|S) + \sigma_B \cdot p(x_i|B)}{\sigma_B \cdot p(x_i|B)} = \mu \cdot \frac{\sigma_S}{\sigma_B} \cdot \frac{s(x_i)}{1 - s(x_i)} + 1.$$

Same observable s is optimal to test all μ hypotheses!
No need to develop separate analysis per hypothesis μ

8

No longer in this convenient spacial case: The same observable **no longer optimal due to non-linear effects** coming from quantum interference

Also **does not generalise to an arbitrary theory parameter** θ, (eg. Effective Field Theory parameters)

Can we modify the LHC analysis methodology to **design near-optimal analyse for the general case?**

Cranmer et al

Estimating high-dimensional density ratios

$$\mathscr{L}(\mu \,|\, \mathscr{D}) = p(\mathscr{D} \,|\, \mu)$$

Neyman–Pearson lemma: **Likelihood ratio is the most powerful test statistic**

We want to compare likelihoods:

$$\frac{p(\mathscr{D} \,|\, \mu)}{p(\mathscr{D} \,|\, ref)}$$

A neural network classifier trained on simulated samples from θ_1 vs simulated samples from ref, estimates the decision function:

$$s(x_i) = \frac{p(x_i \,|\, \mu_1)}{p(x_i \,|\, \mu_1) + p(x_i \,|\, ref)}$$

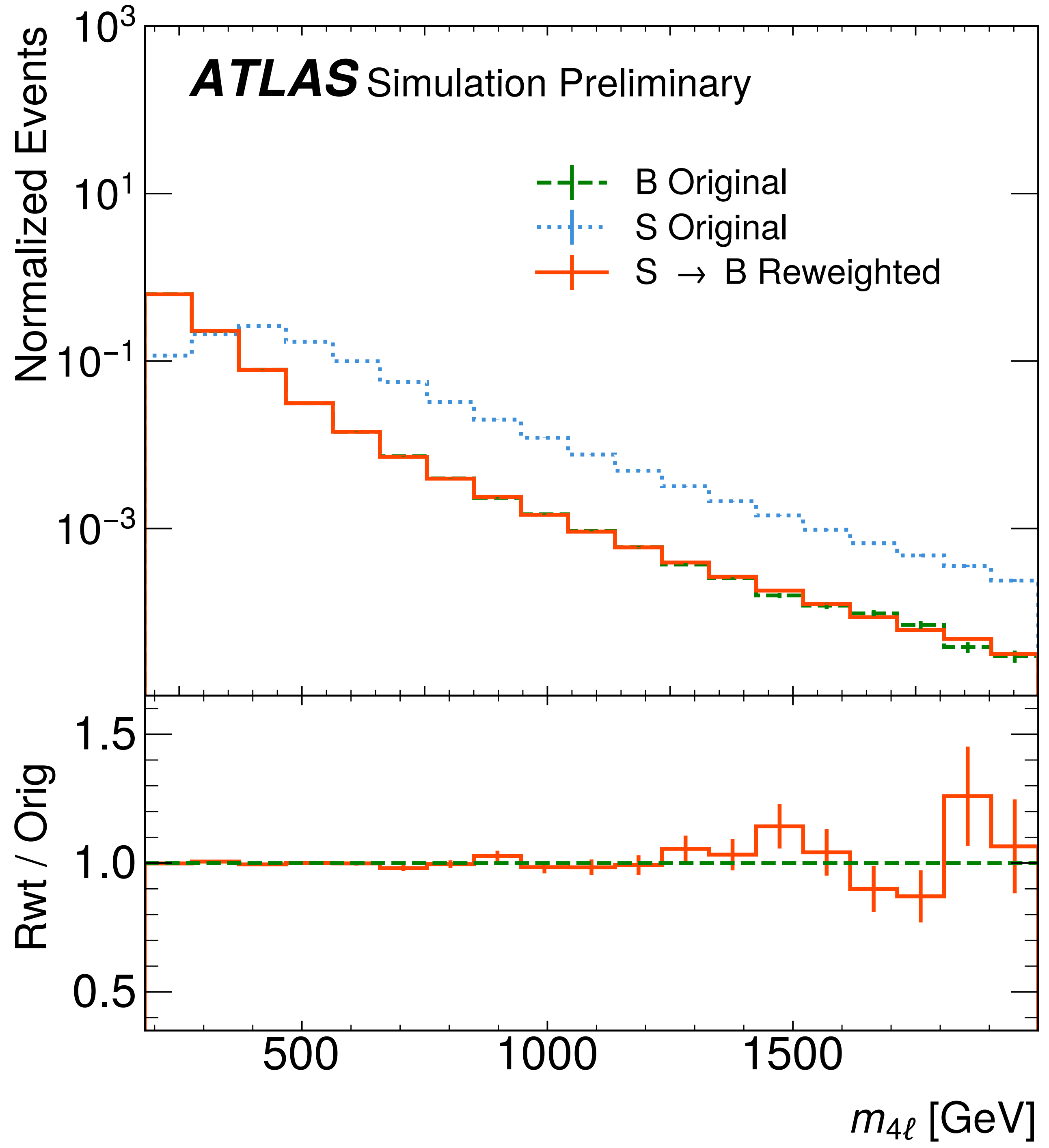
Which contains all the information required for the likelihood ratio:

$$\frac{p(x_i \,|\, \mu_1)}{p(x_i \,|\, ref)} = \frac{s(x_i)}{1 - s(x_i)}$$

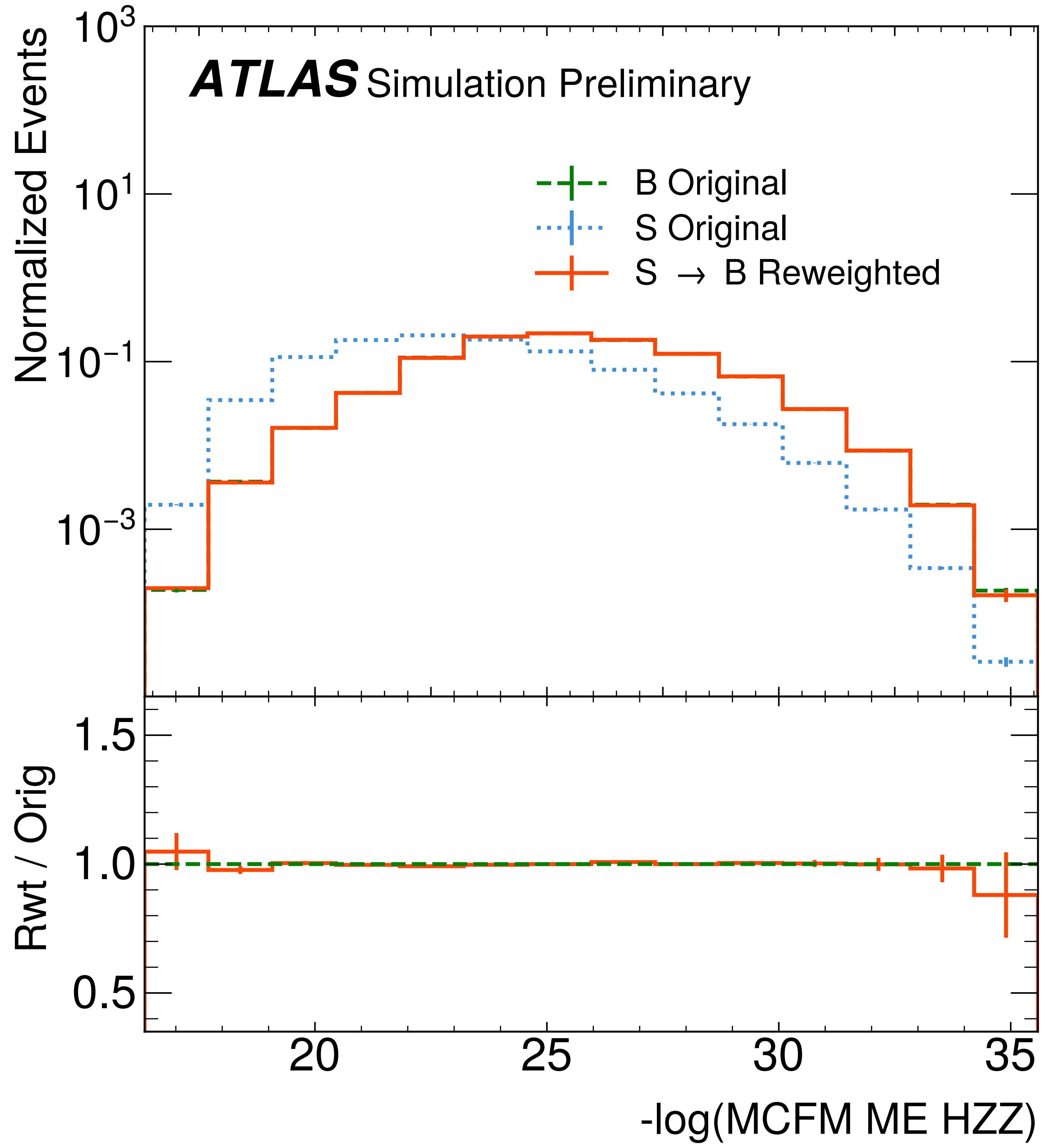
- ✴ Optimal statistic to test each value of μ
- ✴ We get the LR *per event* (unbinned)

Re-weight closures for B

Source
Target
RW



m_{4l}



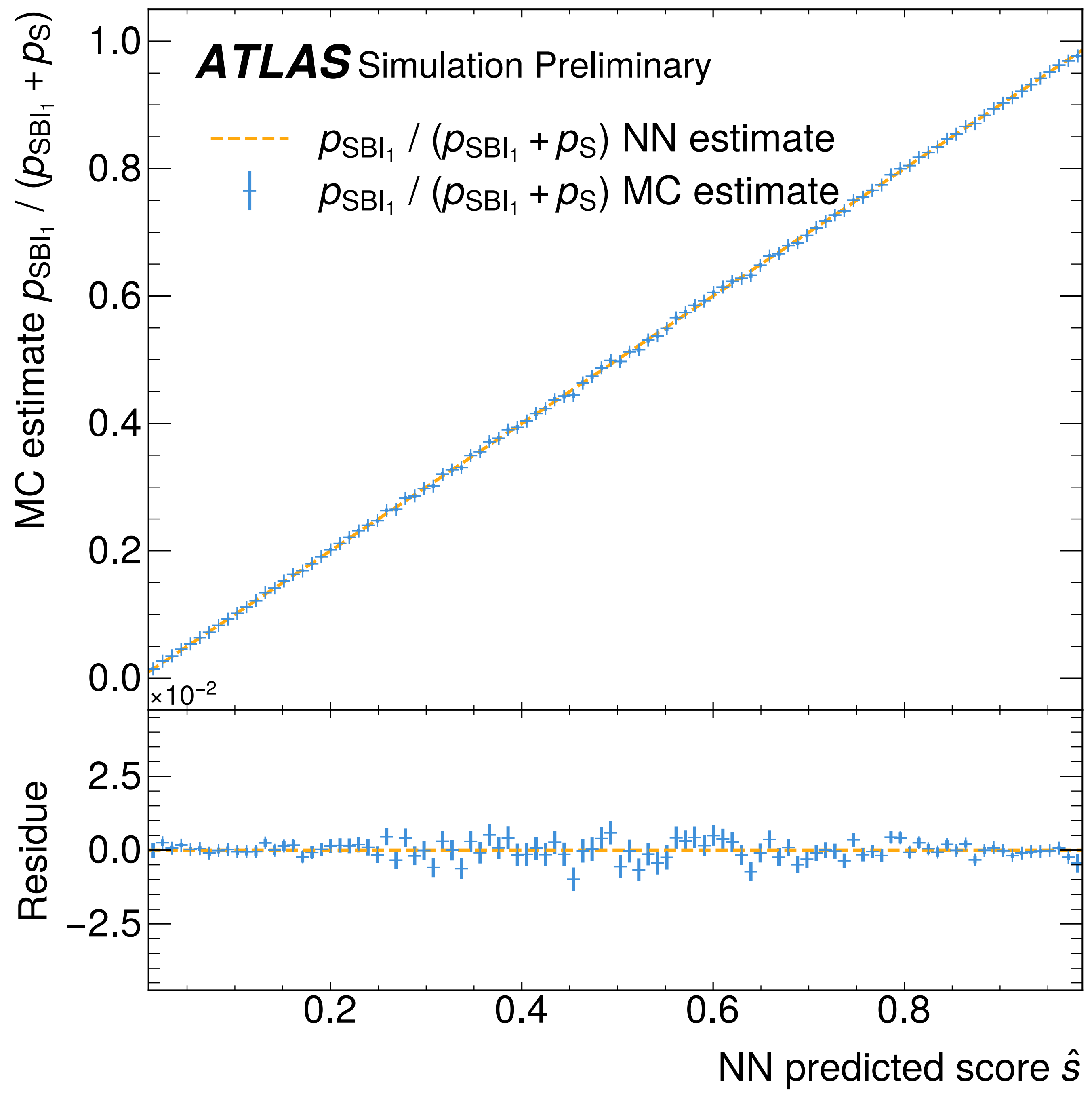
Matrix-Element-based Observable
(ggF from MCFM)

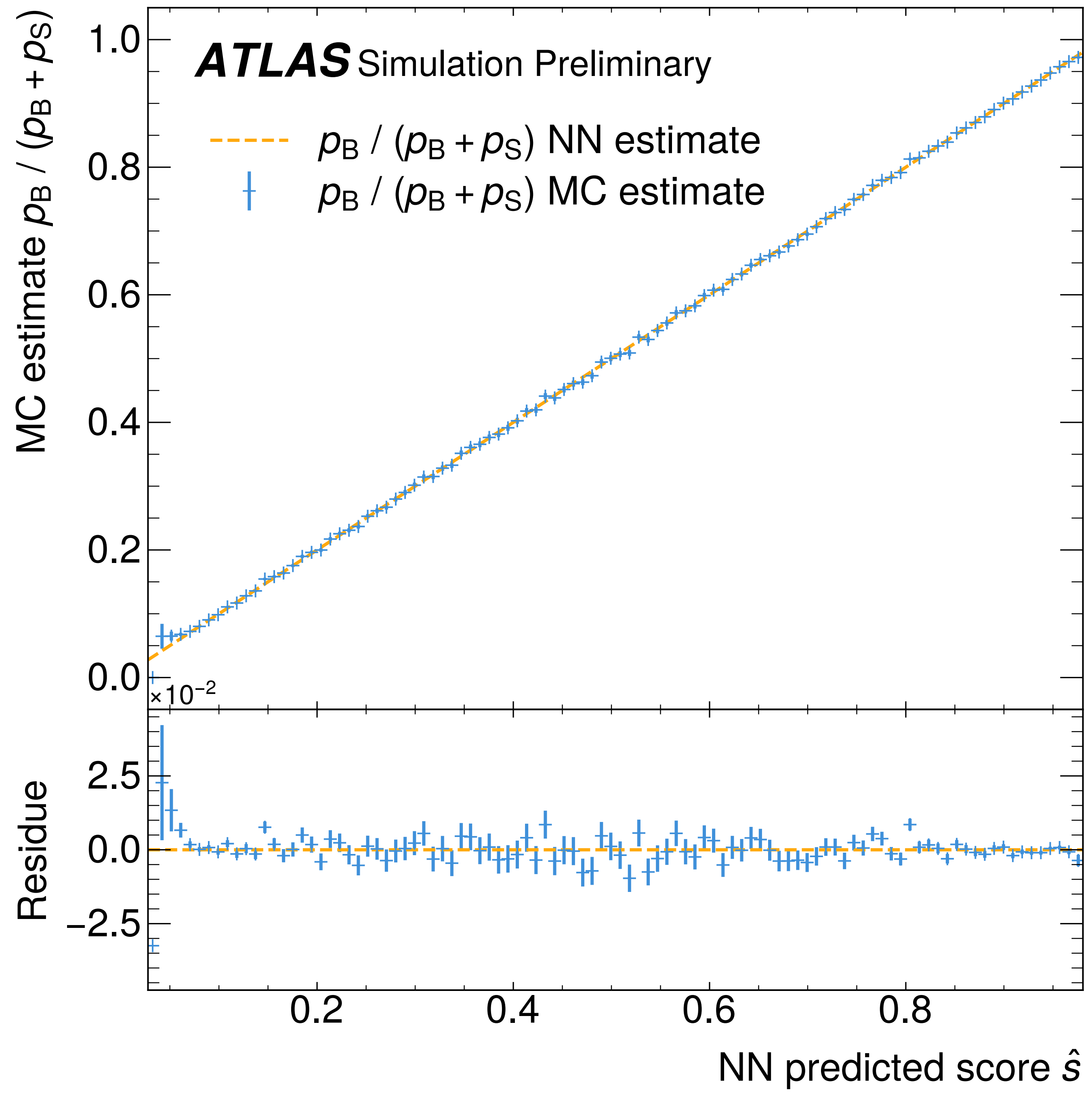
Calibration Curves

$$\frac{P_{SBI}}{P_{SBI} + P_{ref}}$$

$$\frac{P_B}{P_B + P_{ref}}$$

Binned estimate





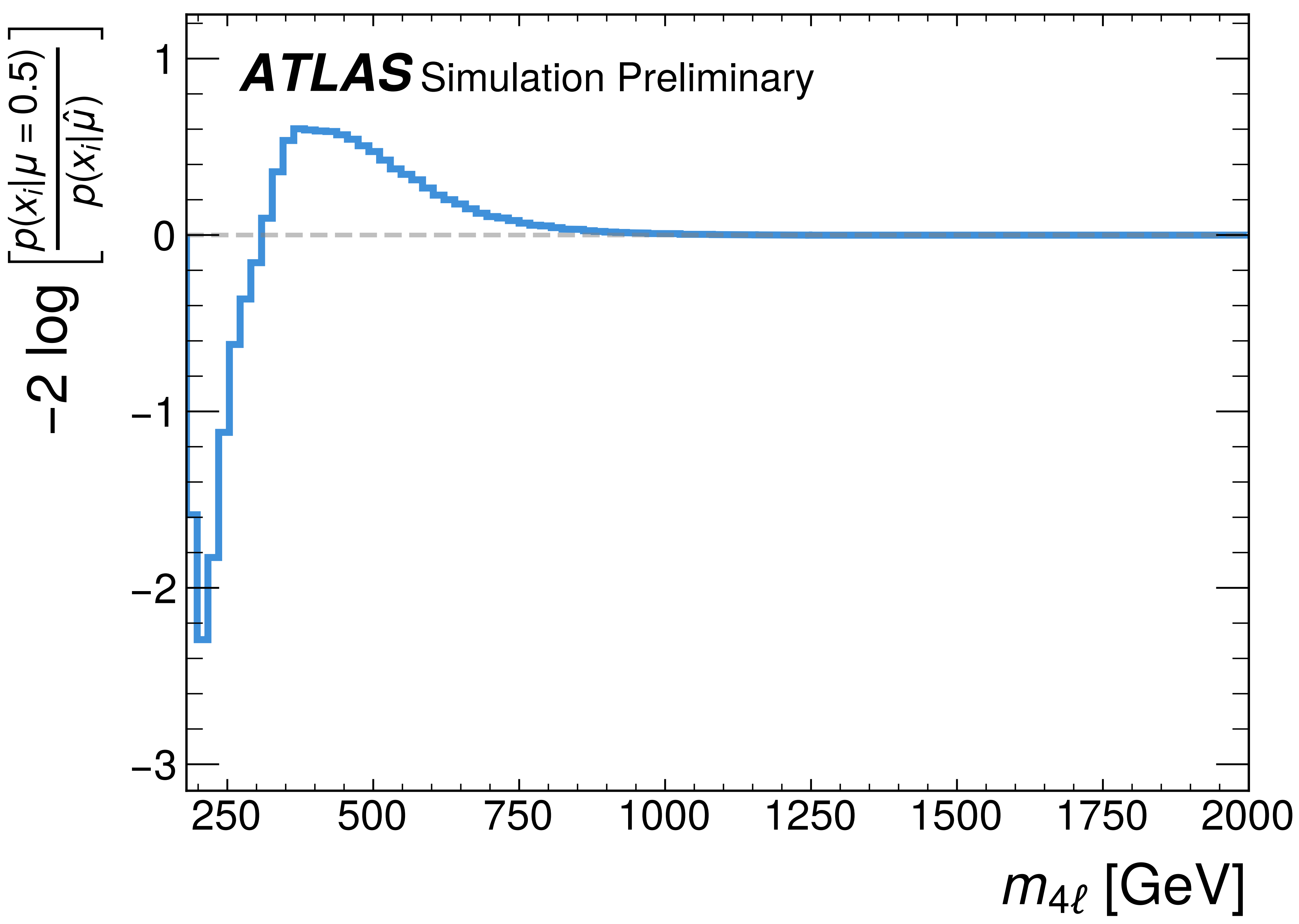
Ensemble prediction

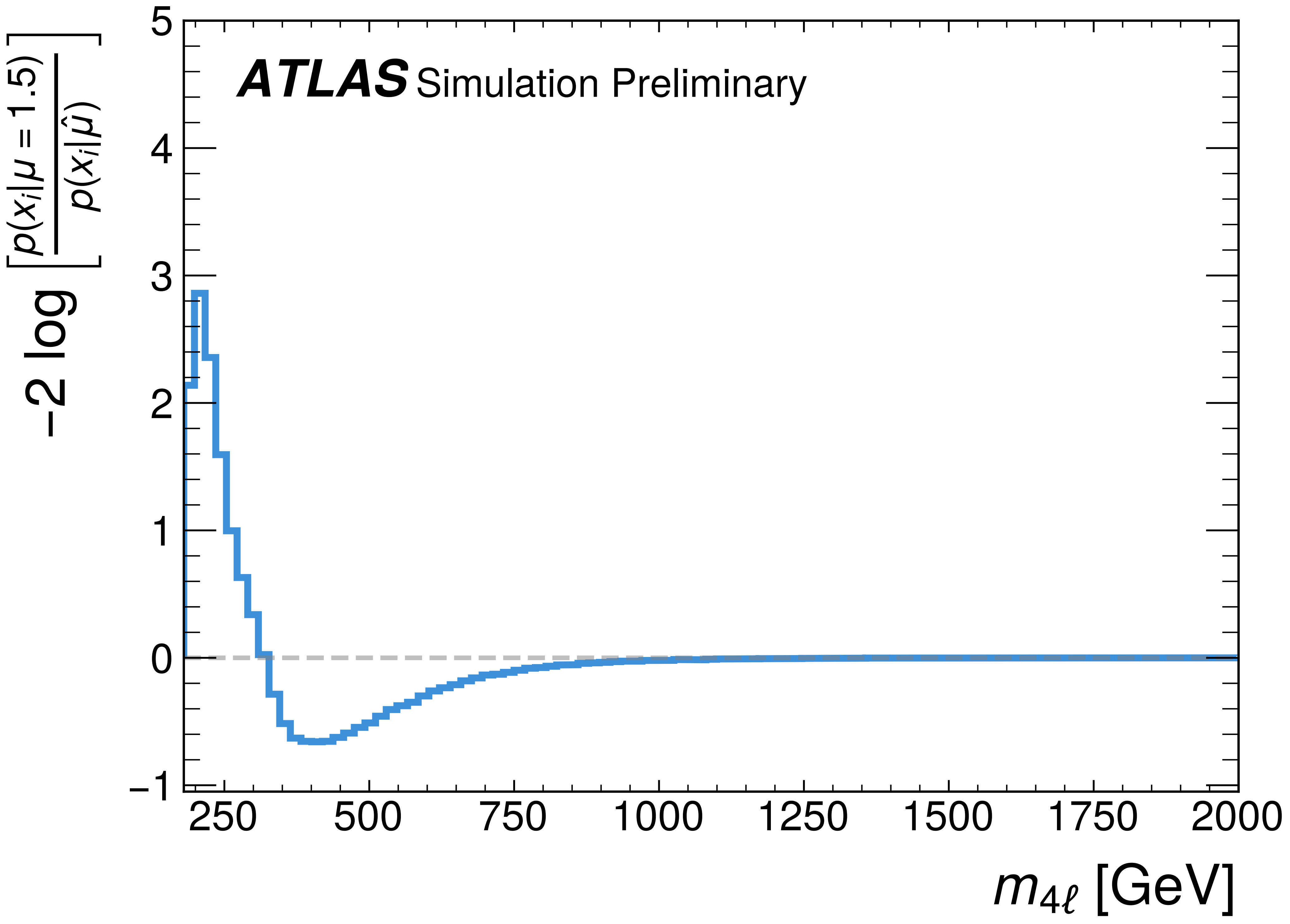
Ensemble prediction

Interpretability:
Which phase space favours one hypothesis over another?

$$-2 \cdot log \frac{P(x_i | \mu = 0.5)}{P(x_i | \mu = 1)}$$

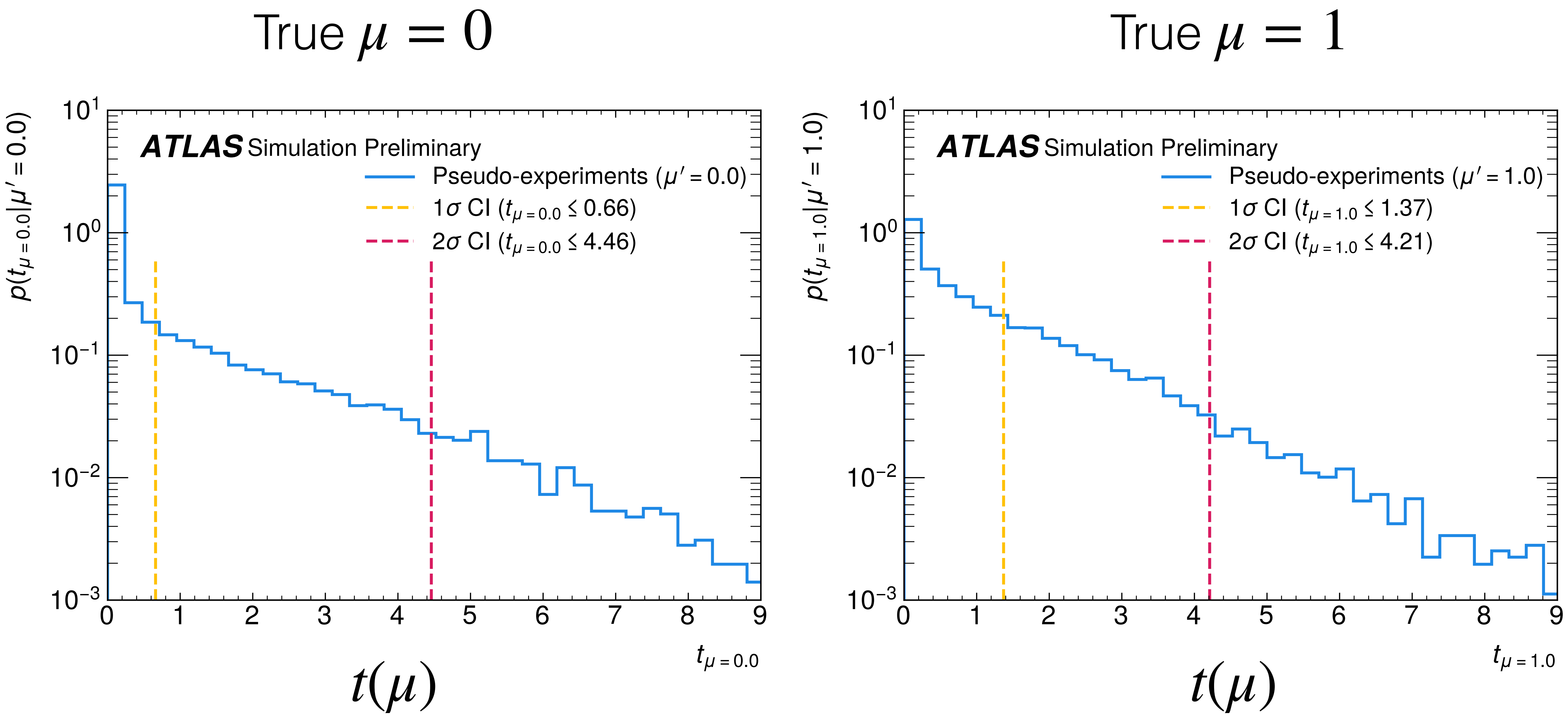
$$-2 \cdot log \frac{P(x_i | \mu = 1.5)}{P(x_i | \mu = 1)}$$





Neyman Construction

- To build confidence intervals, we need to 'invert the hypothesis test'
- Generate pseudo-experiments ('toys') and determine 1σ & 2σ CI as a function of parameter of interest



Negative Weighted Events

1. Start from a positive weighted reference sample instead
2. Re-weight to intended parameter point
3. Throw toys from this sample

$$w_i^{\text{rwt-ref}} \rightarrow w_i^{\text{Asimov}}(\mu) = \frac{\nu(\mu)}{\nu_{\text{rwt-ref}}} \cdot \frac{p(x_i\,|\,\mu)}{P_{\text{rwt-ref}}(x_i)} \cdot w_i^{\text{rwt-ref}}$$

Estimating the variance on mean: Bootstrapping

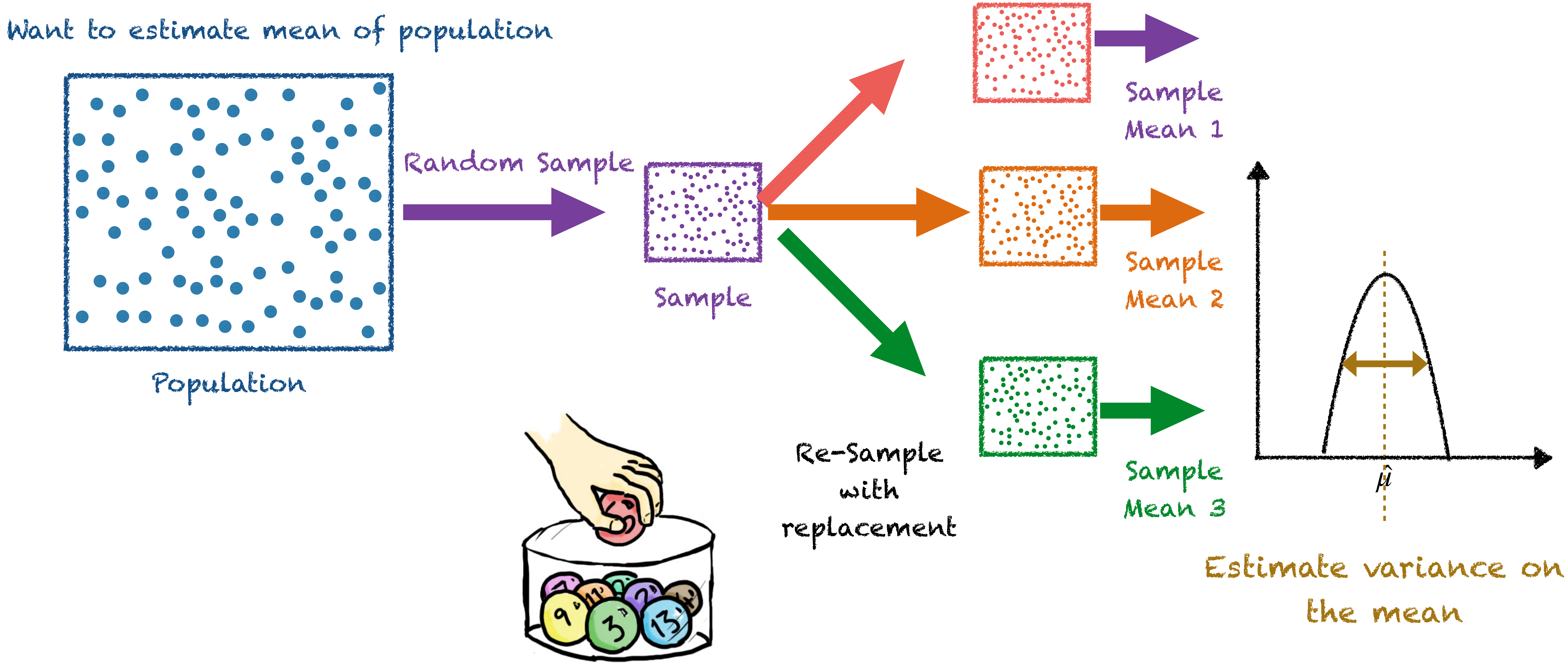


Image: Source

Quantifying uncertainty on estimated density ratio

$$w_i \rightarrow w_i \cdot Pois(1)$$

- Train an ensemble of networks, each on a Poisson fluctuated version of the training dataset

- Ensemble average used as final prediction, estimate the variance on mean from bootstrapped ensembles

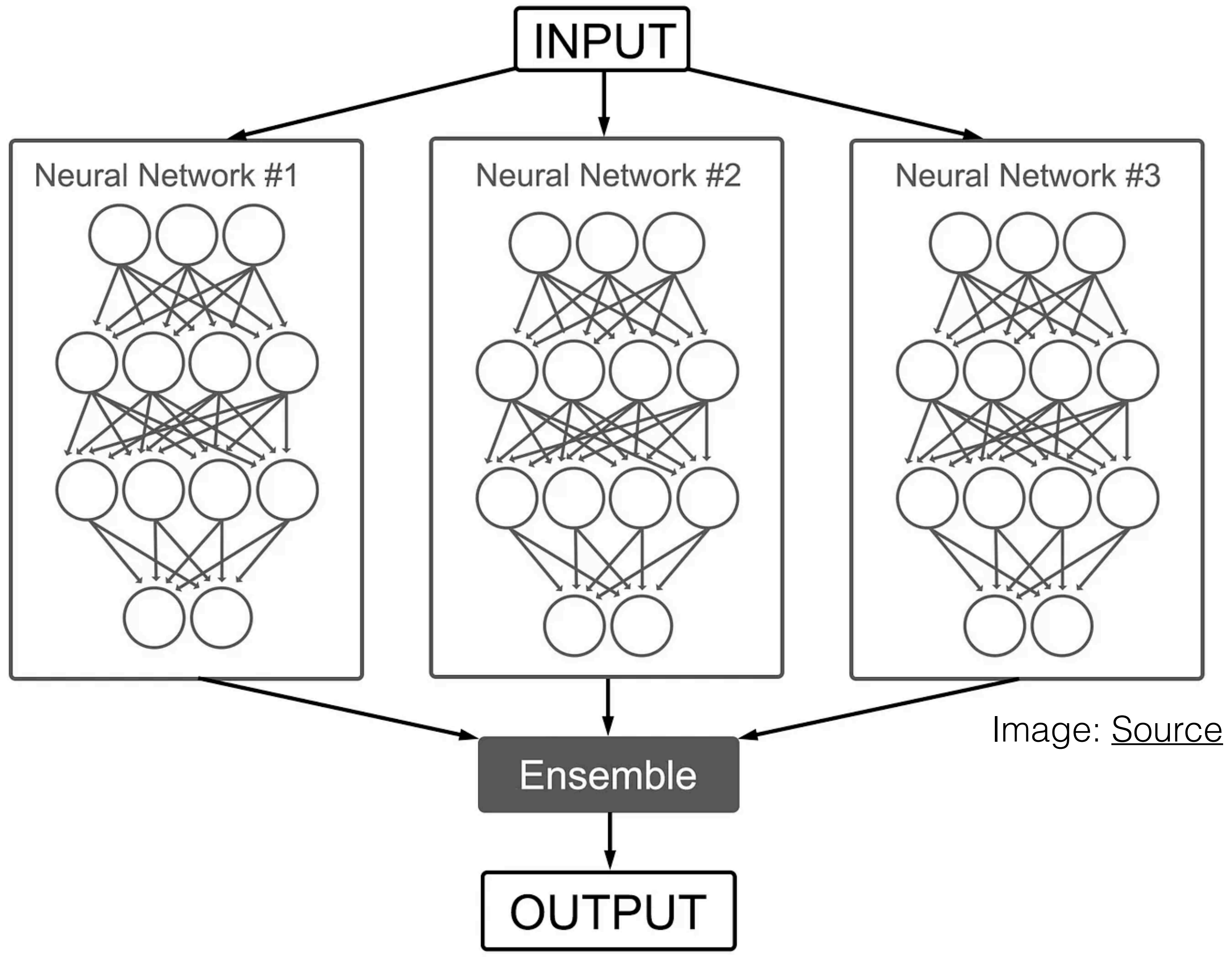


Image: Source

Quantifying uncertainty on estimated density ratio

$$w_i \rightarrow w_i \cdot Pois(1)$$

- Train an ensemble of networks, **each on a Poisson fluctuated version of the training dataset**
- Ensemble average used as final prediction, **estimate the variance on mean from bootstrapped ensembles**

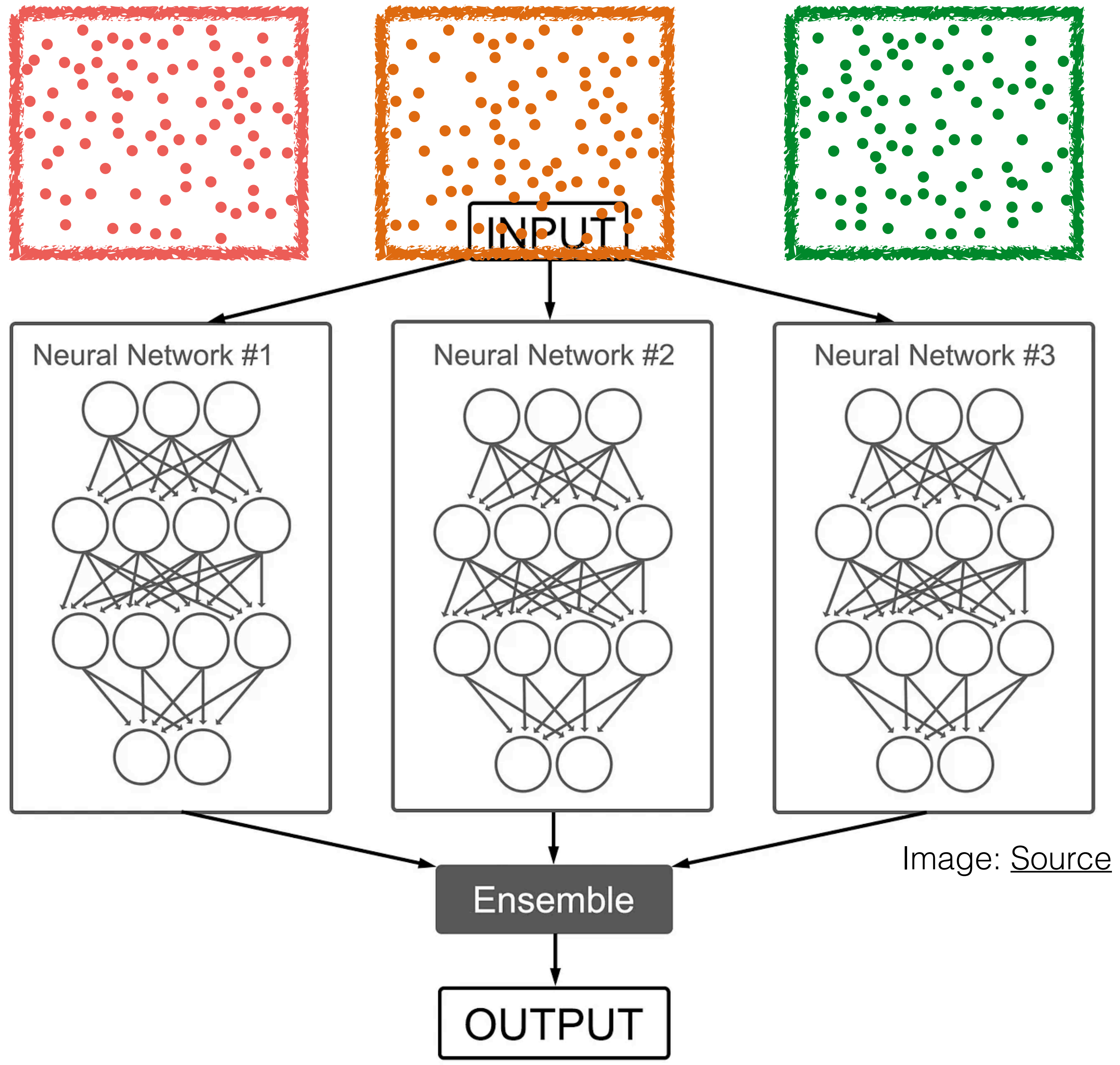


Image: Source

Quantifying uncertainty on estimated density ratio

$$w_i \rightarrow w_i \cdot Pois(1)$$

- Train an ensemble of networks, **each on a Poisson fluctuated version of the training dataset**
- Ensemble average used as final prediction, **estimate the variance on mean from bootstrapped ensembles**

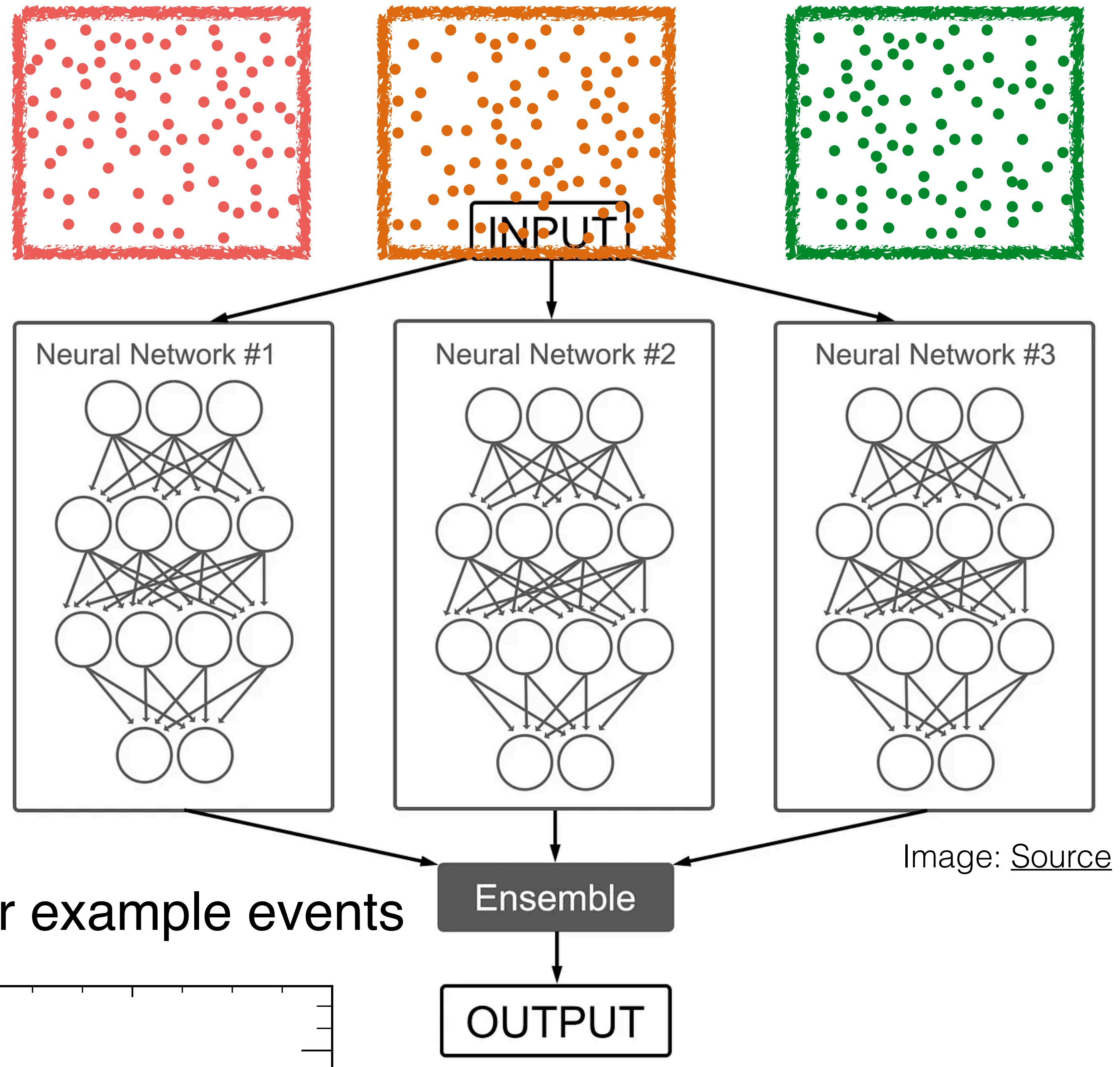
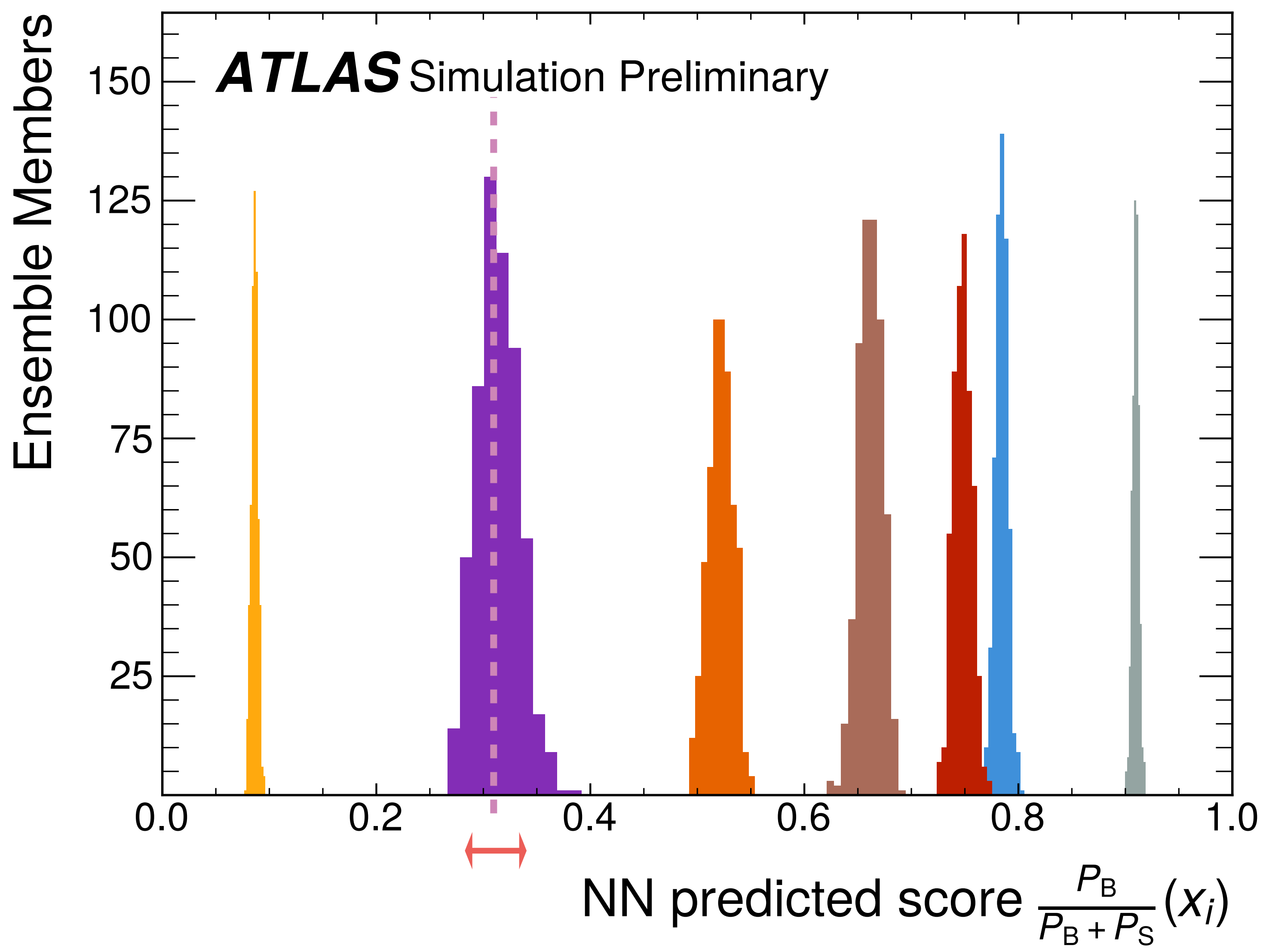


Image: Source

Distribution of NN predictions for example events



Quantifying uncertainty on estimated density ratio

$$w_i \rightarrow w_i \cdot Pois(1)$$

- Train an ensemble of networks, **each on a Poisson fluctuated version of the training dataset**

- Ensemble average used as final prediction, **estimate the variance on mean from bootstrapped ensembles**

- Propagate with spurious signal method

$$f_j(\mu) \rightarrow f_j(\mu + \alpha \cdot \Delta\hat{\mu}(\mu))$$

Constraint term: $Gauss(0,1)$

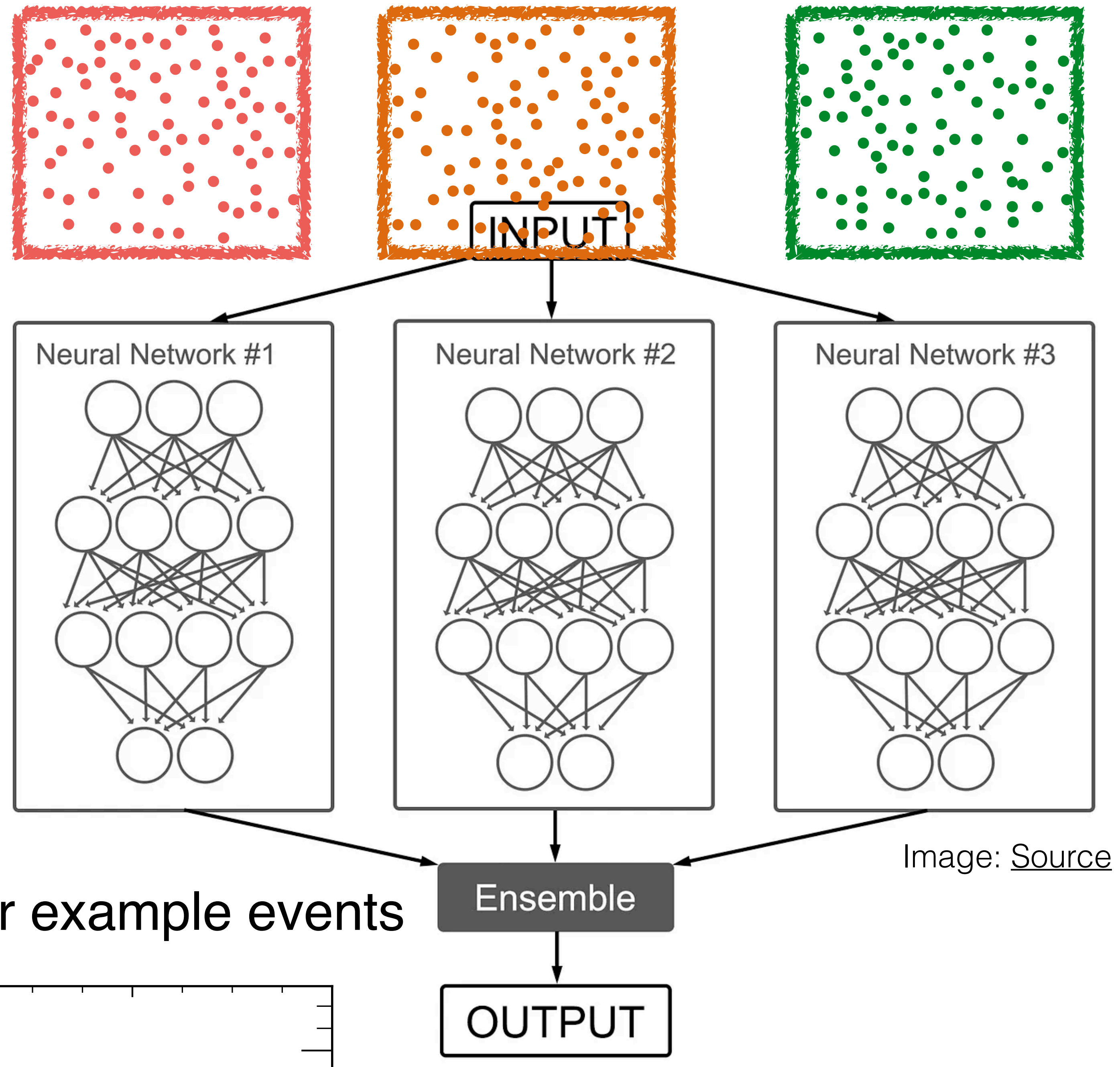
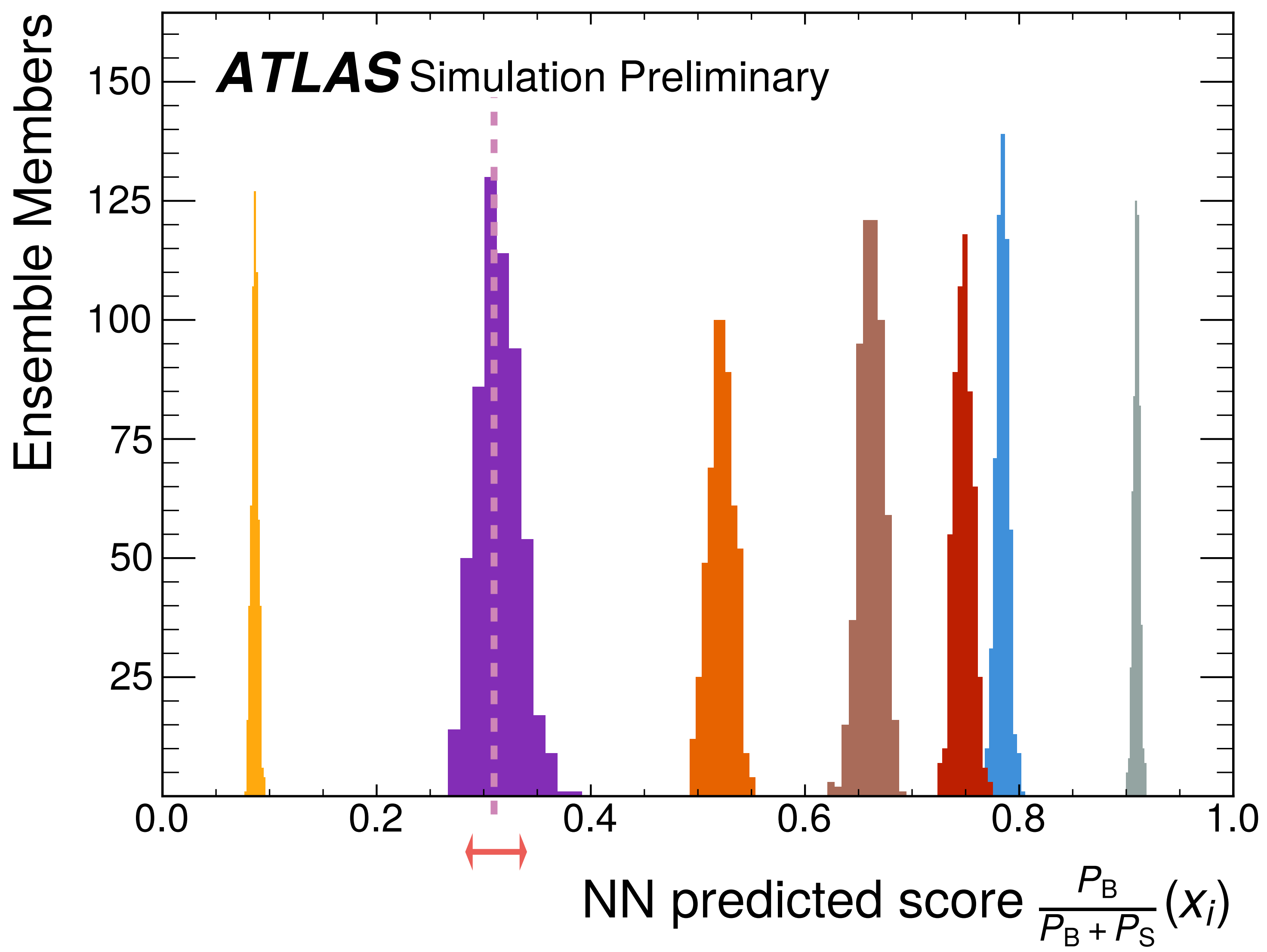


Image: Source

Distribution of NN predictions for example events



Simulated Samples

- PoI: Signal strength μ
- Simplified, unphysical dataset:
  - Processes: S: $gg \rightarrow H^* \rightarrow 4l$ & B: $gg \rightarrow ZZ \rightarrow 4l$, SBI: full process
  - No VBF processes or qqZZ background
  - Two systematics: ggF NLO K-factor uncertainty (shape + norm) & luminosity uncertainty (norm only)

Input variables

Variable	**Definition**
Production Kinematics	
$m_{4\ell}$	Four-lepton invariant mass
$p_T^{4\ell}$	Four-lepton transverse momentum
$\eta^{4\ell}$	Four-lepton pseudo-rapidity
Decay Kinematics	
m_{Z1}	Z_1 mass
m_{Z2}	Z_2 mass
$\cos\theta^*$	Higgs decay angle
$\cos\theta_1$	Z_1 decay angle
$\cos\theta_2$	Z_2 decay angle
ϕ	Angle between Z_1, Z_2 decay planes
ϕ_1	Z_1 decay plane angle

Combination with histogram analyses

$$\frac{L_{\text{comb}}(\mu, \alpha)}{L_{\text{ref}}} = \frac{L_{\text{full}}(\mu, \alpha)}{L_{\text{ref}}} \; L_{\text{hist}}(\mu, \alpha)$$

Calculating pulls and impacts in JAX

$$\lambda(\mu, \alpha) = -2\ln(L_{full}(\mu, \alpha)/L_{ref})$$

Hessian:

$$C_{nm} = \left[\frac{1}{2}\frac{\partial^2 \lambda}{\partial \alpha_n \partial \alpha_m}(\widehat{\mu}, \widehat{\alpha})\right]^{-1}$$

Pulls:

$$\frac{\widehat{\alpha}_k - \alpha_k^0}{\sqrt{C_{kk}}}.$$

Post-fit Impact:

$$\Gamma_k = \frac{\partial \widehat{\mu}}{\partial \alpha_k} \times \sqrt{C_{kk}}$$

$$= -\left[\frac{\partial^2 \lambda}{\partial^2 \mu}(\widehat{\mu}, \widehat{\alpha})\right]^{-1} \frac{\partial^2 \lambda}{\partial \mu \partial \alpha_k}(\widehat{\mu}, \widehat{\alpha}) \times \sqrt{C_{kk}},$$

Vertical interpolation

$$G_j(\alpha_k) = \begin{cases} \left(\frac{\nu_j(\alpha_k^+)}{\nu_j(\alpha_k^0)}\right)^{\alpha_k} & \alpha_k > 1 \\ 1 + \sum_{n=1}^{6} c_n \alpha_k^n & -1 \le \alpha_k \le 1 \\ \left(\frac{\nu_j(\alpha_k^-)}{\nu_j(\alpha_k^0)}\right)^{-\alpha_k} & \alpha_k < -1 \end{cases}$$

$$g_j(x_i, \alpha_k) = \begin{cases} \left(g_j(x_i, \alpha_k^+)\right)^{\alpha_k} & \alpha_k > 1 \\ 1 + \sum_{n=1}^{6} c_n \alpha_k^n & -1 \le \alpha_k \le 1 \\ \left(g_j(x_i, \alpha_k^-)\right)^{-\alpha_k} & \alpha_k < -1 \end{cases}$$

With some continuity requirements