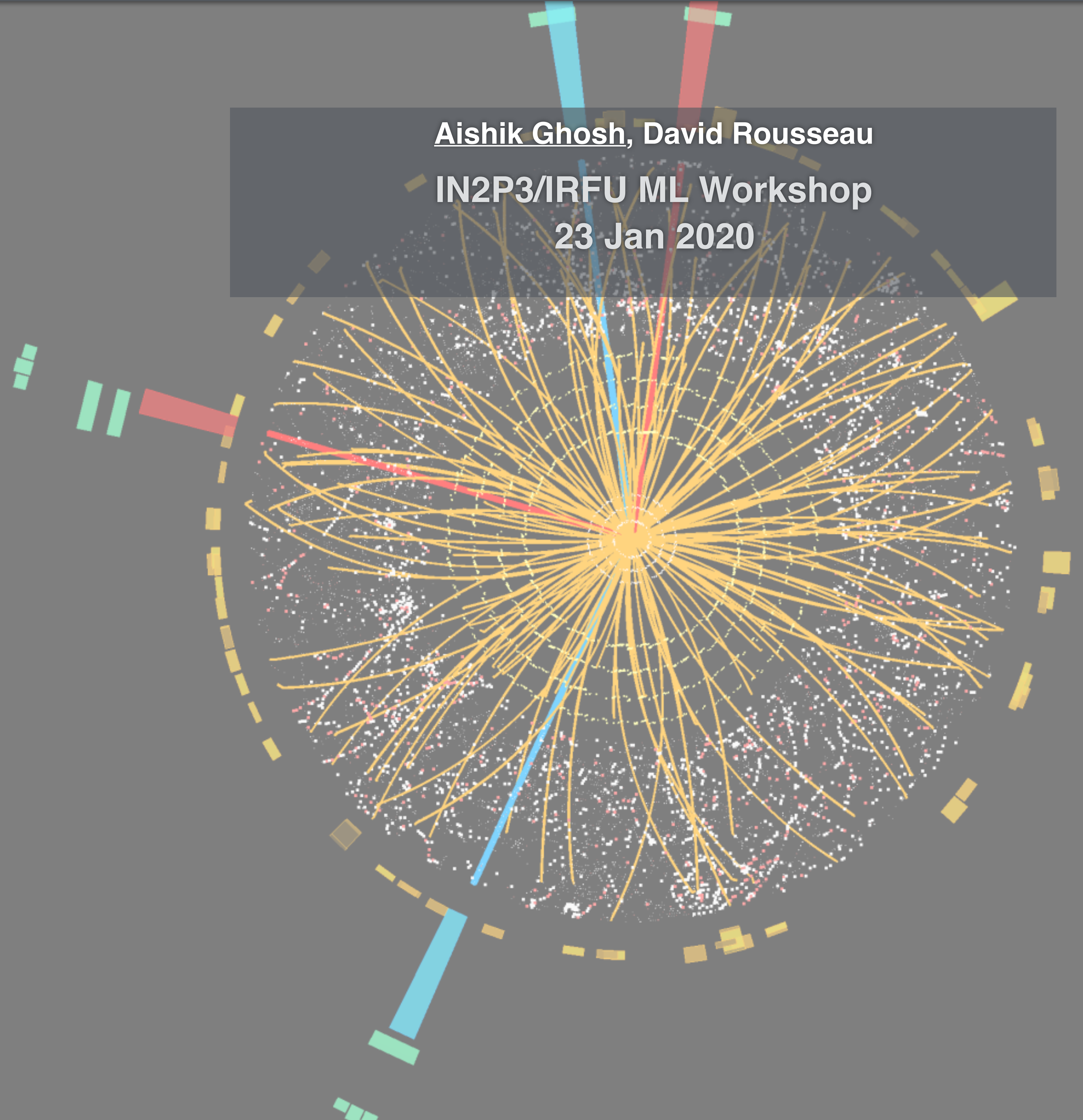
Measuring Quantum Interference in the Off-shell Higgs to 4 Leptons with ML: A First Look

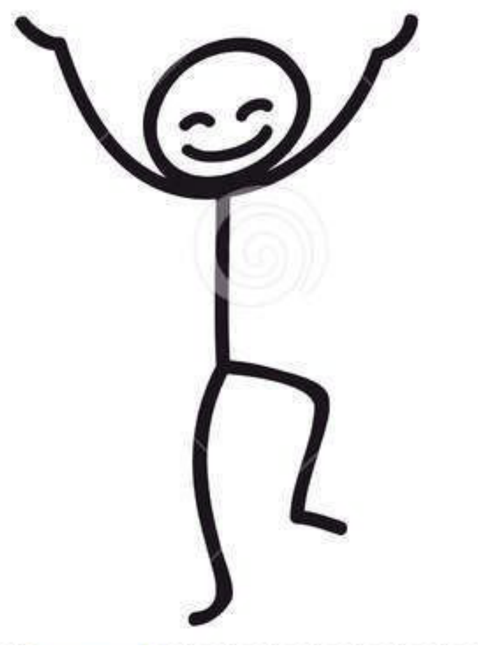
Aishik Ghosh, David Rousseau

IN2P3/IRFU ML Workshop
23 Jan 2020

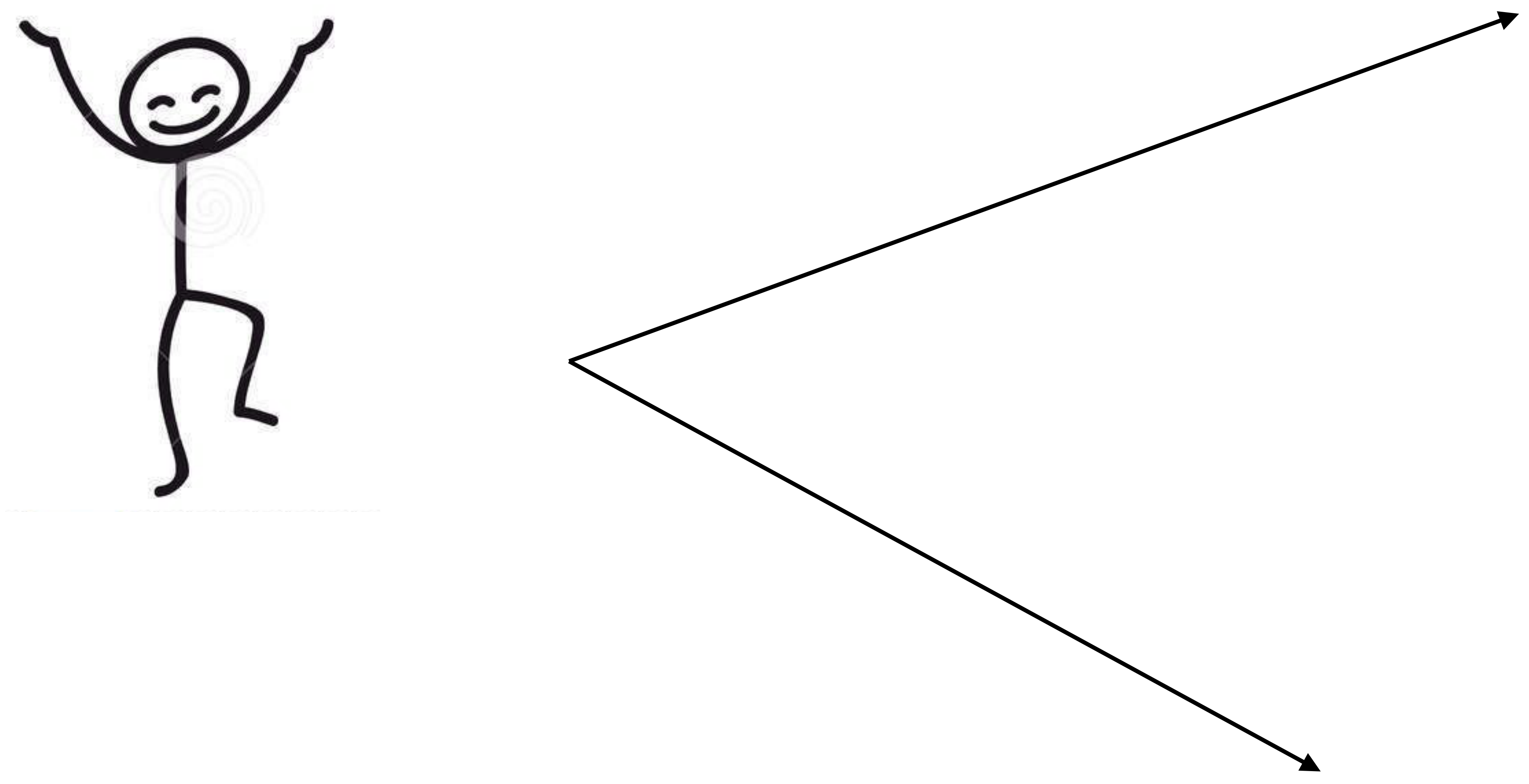


With constant, invaluable support from Johann Brehmer, NYU on likelihood-free inference with Madminer

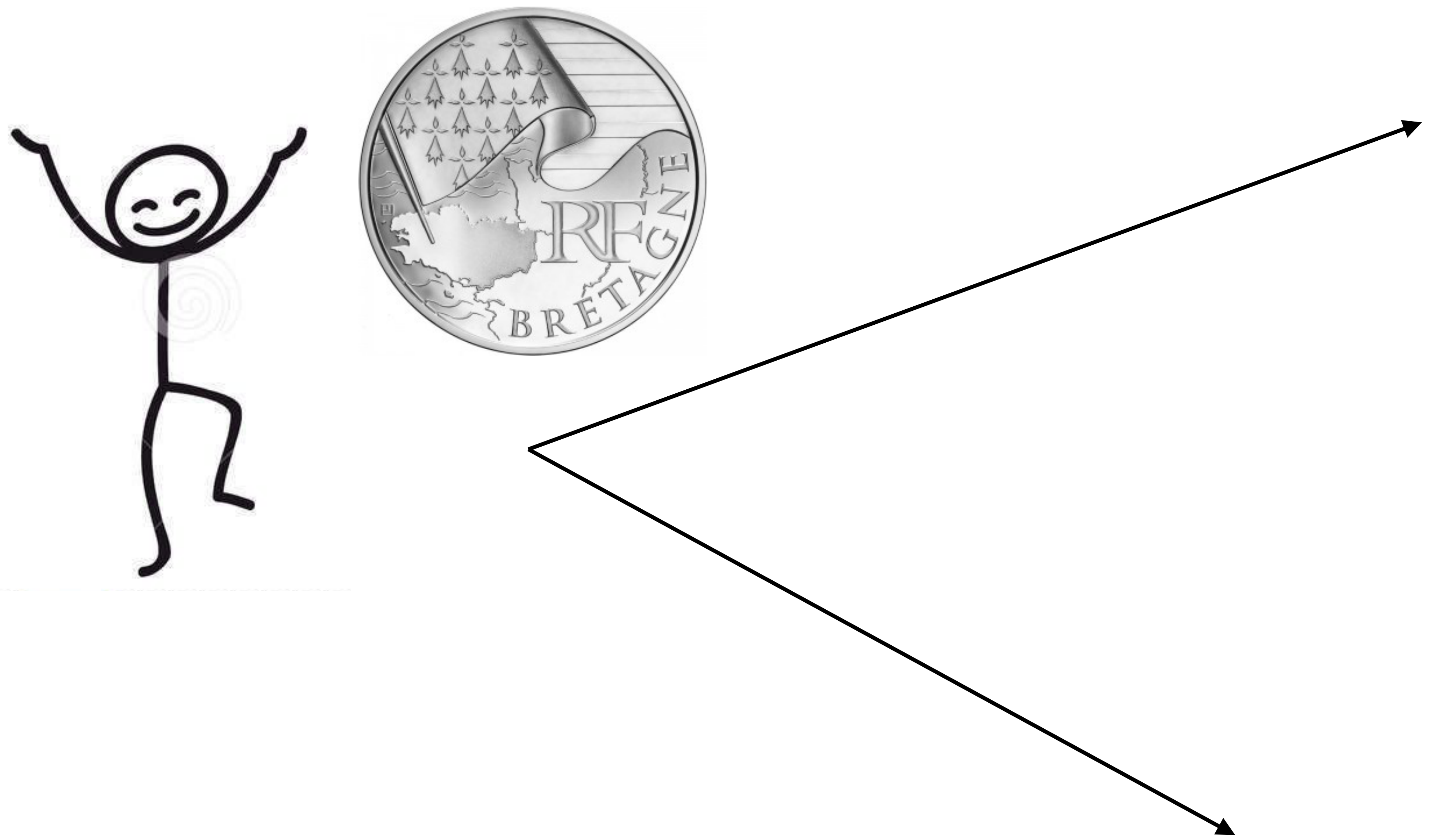
Simulation (Classical System)



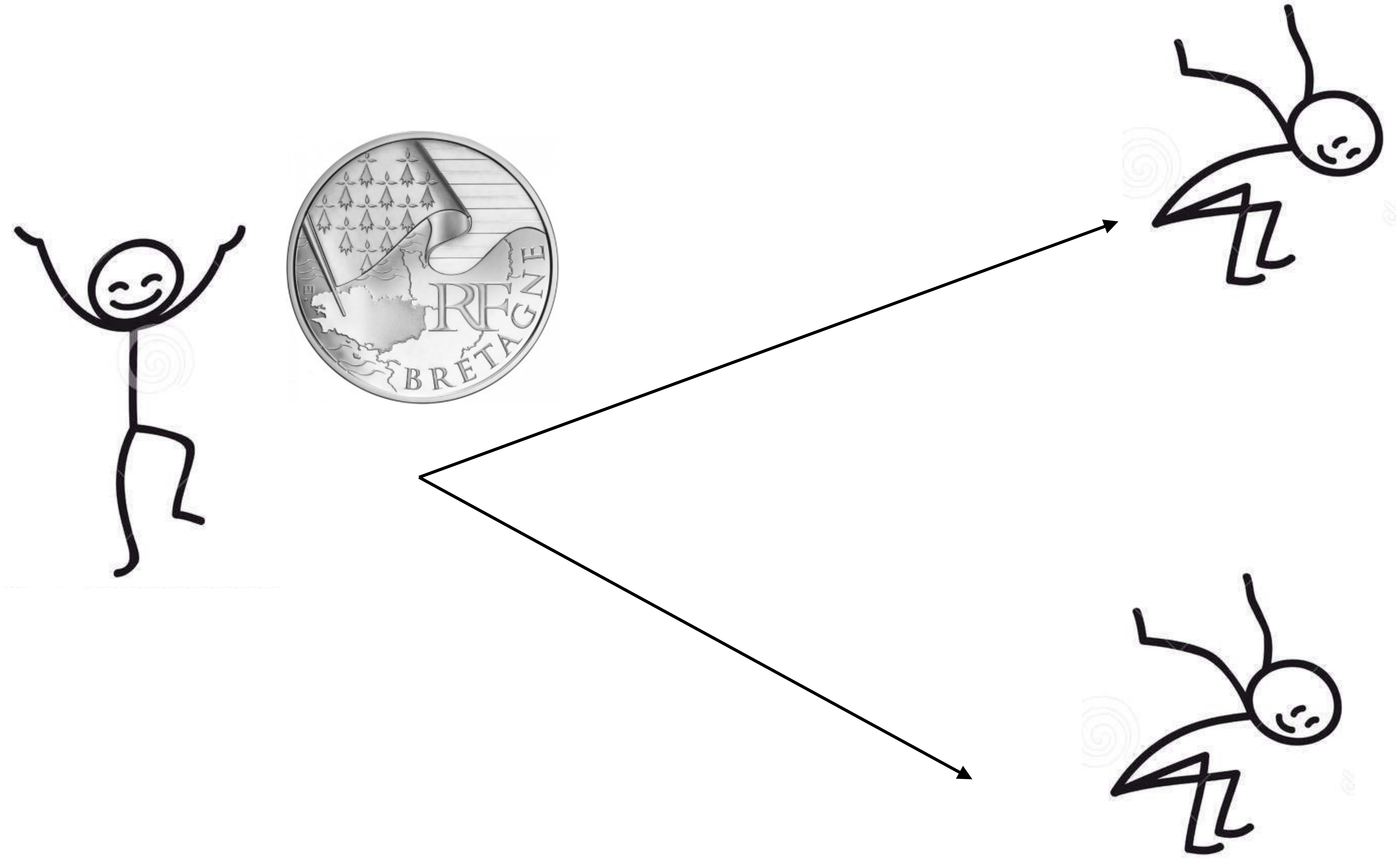
Simulation (Classical System)



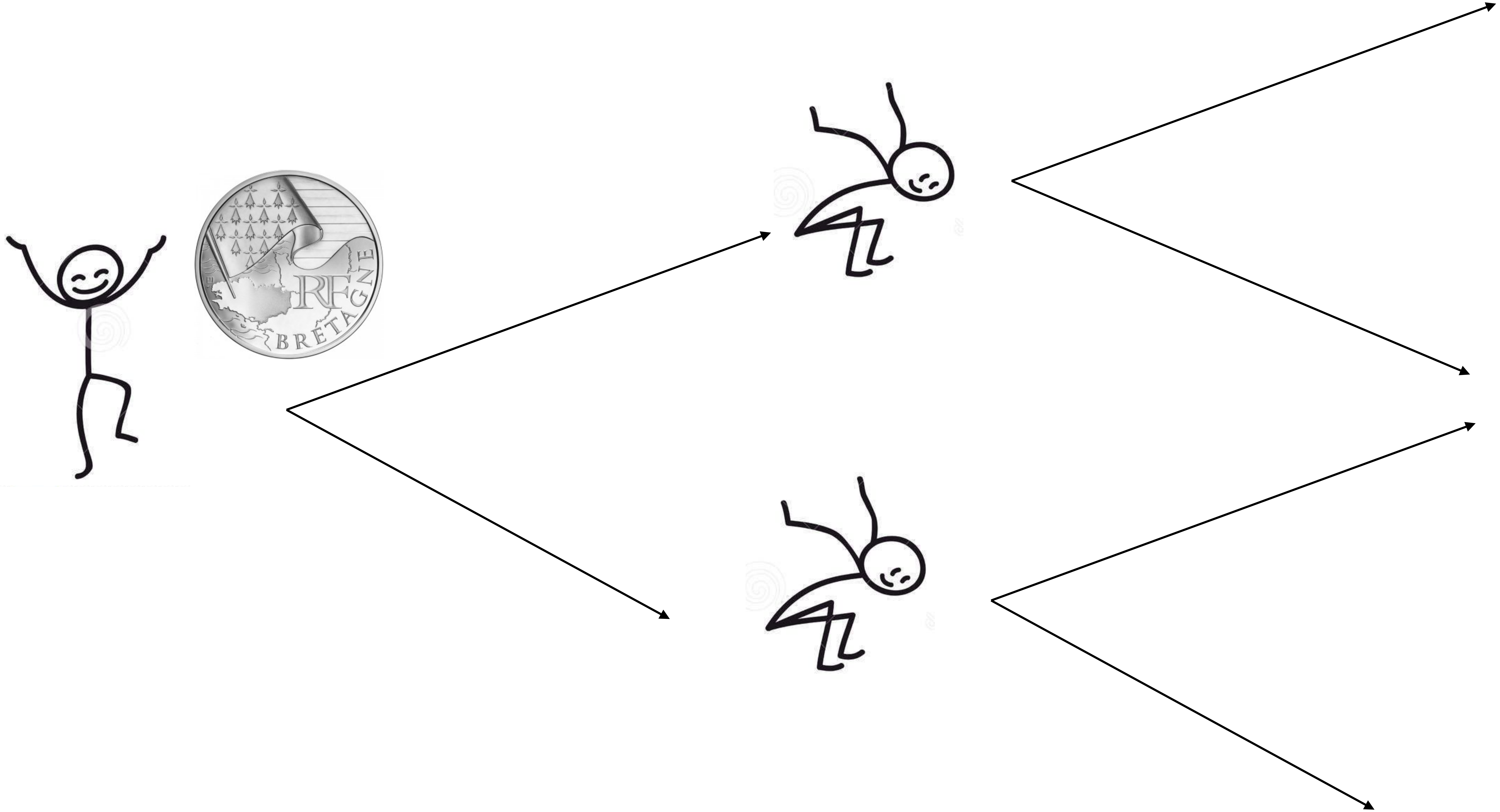
Simulation (Classical System)



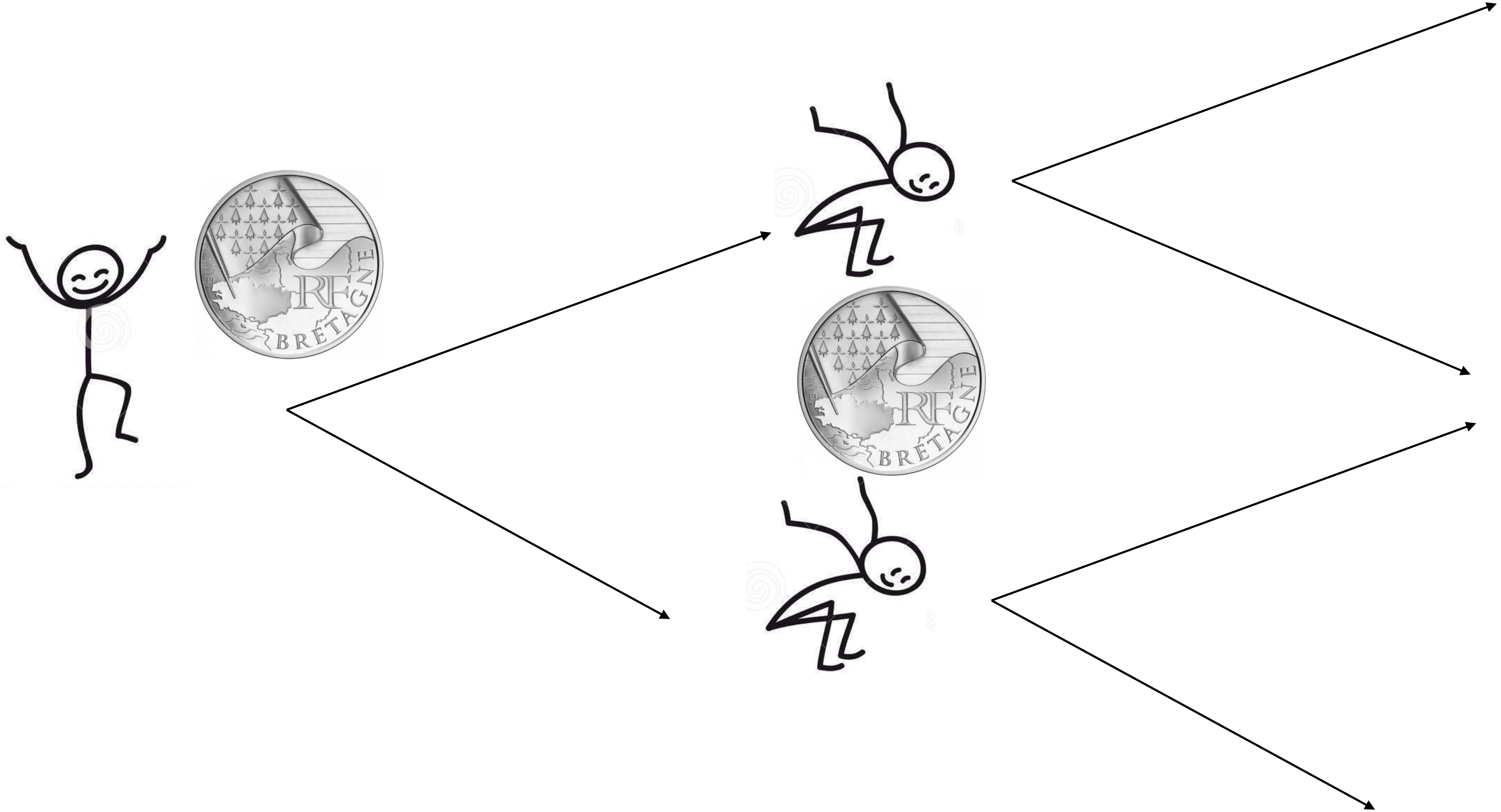
Simulation (Classical System)



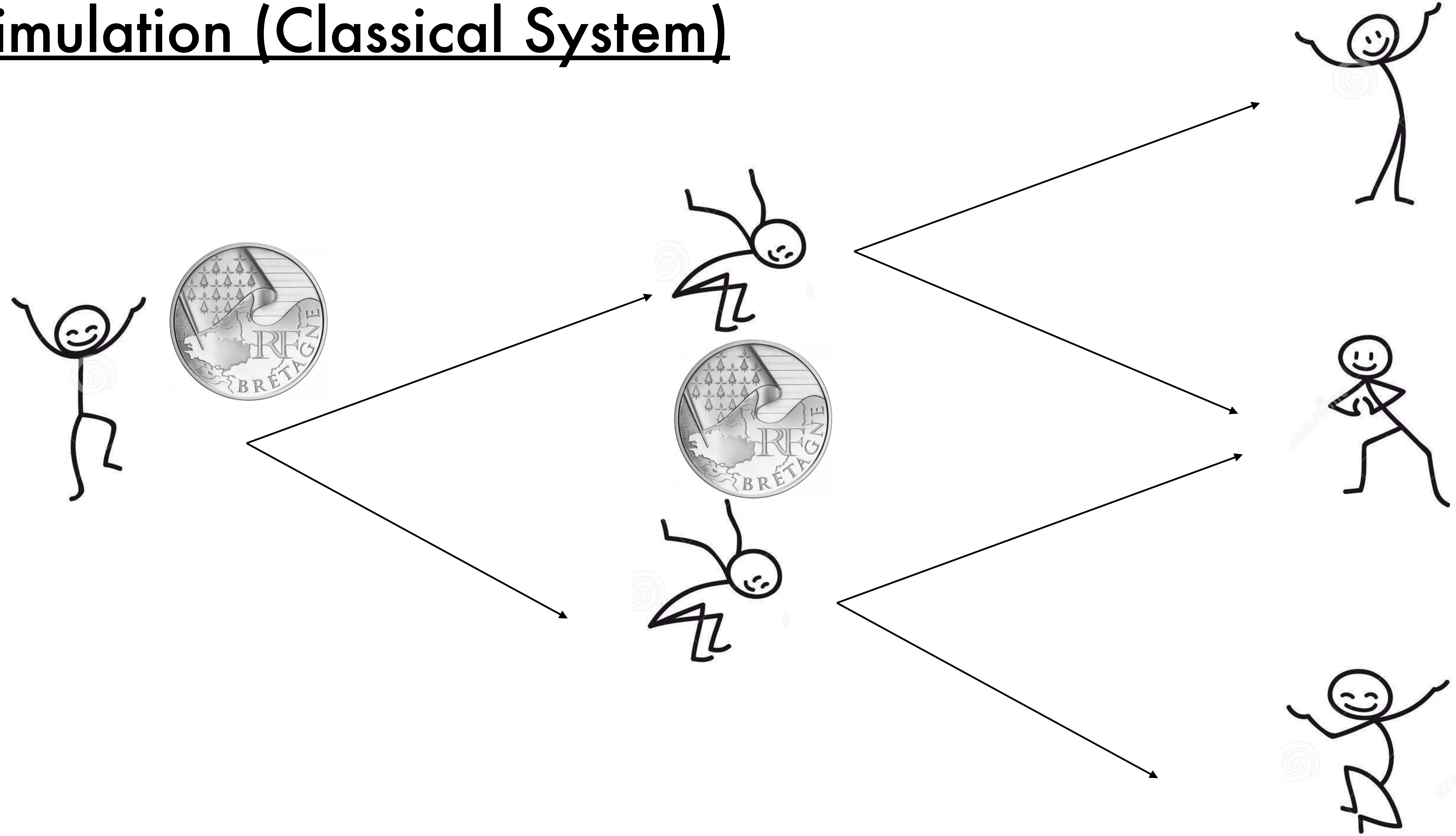
Simulation (Classical System)



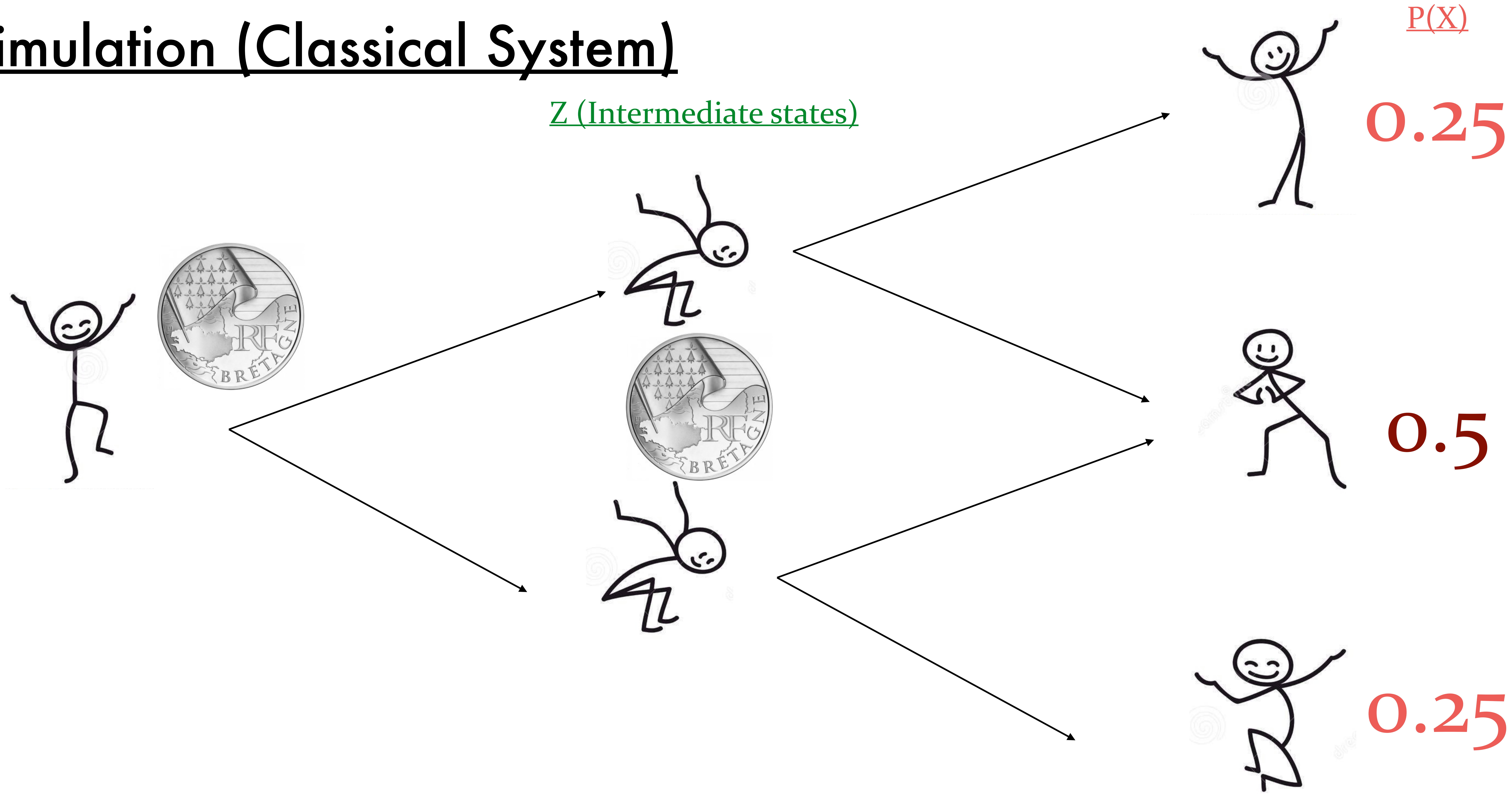
Simulation (Classical System)



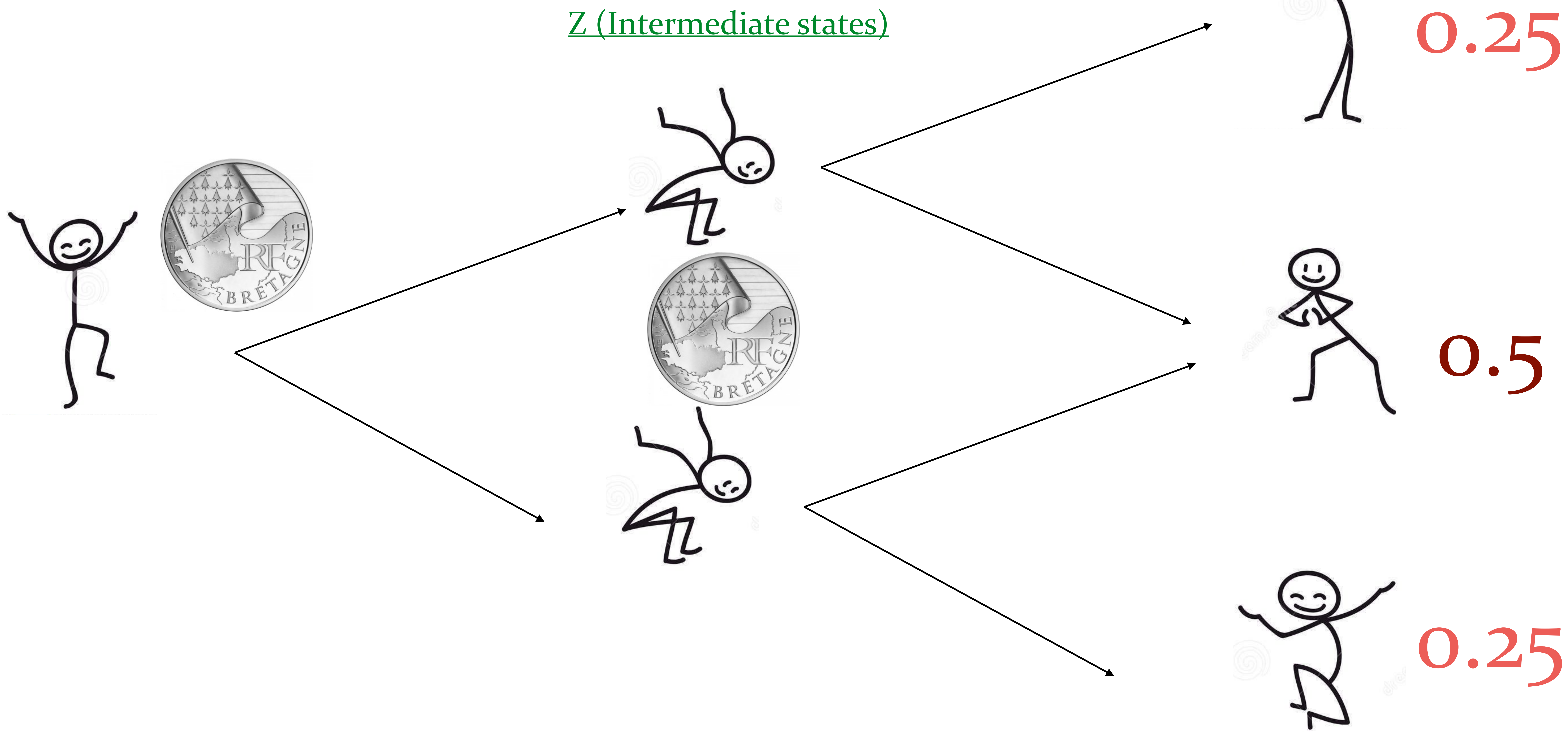
Simulation (Classical System)



Simulation (Classical System)



Simulation (Classical System)



Who needs a simulator? We can arrive at this analytically!

Modelling particle physics processes

Theory
parameters
θ

Evolution

Modelling particle physics processes

Theory
parameters
θ

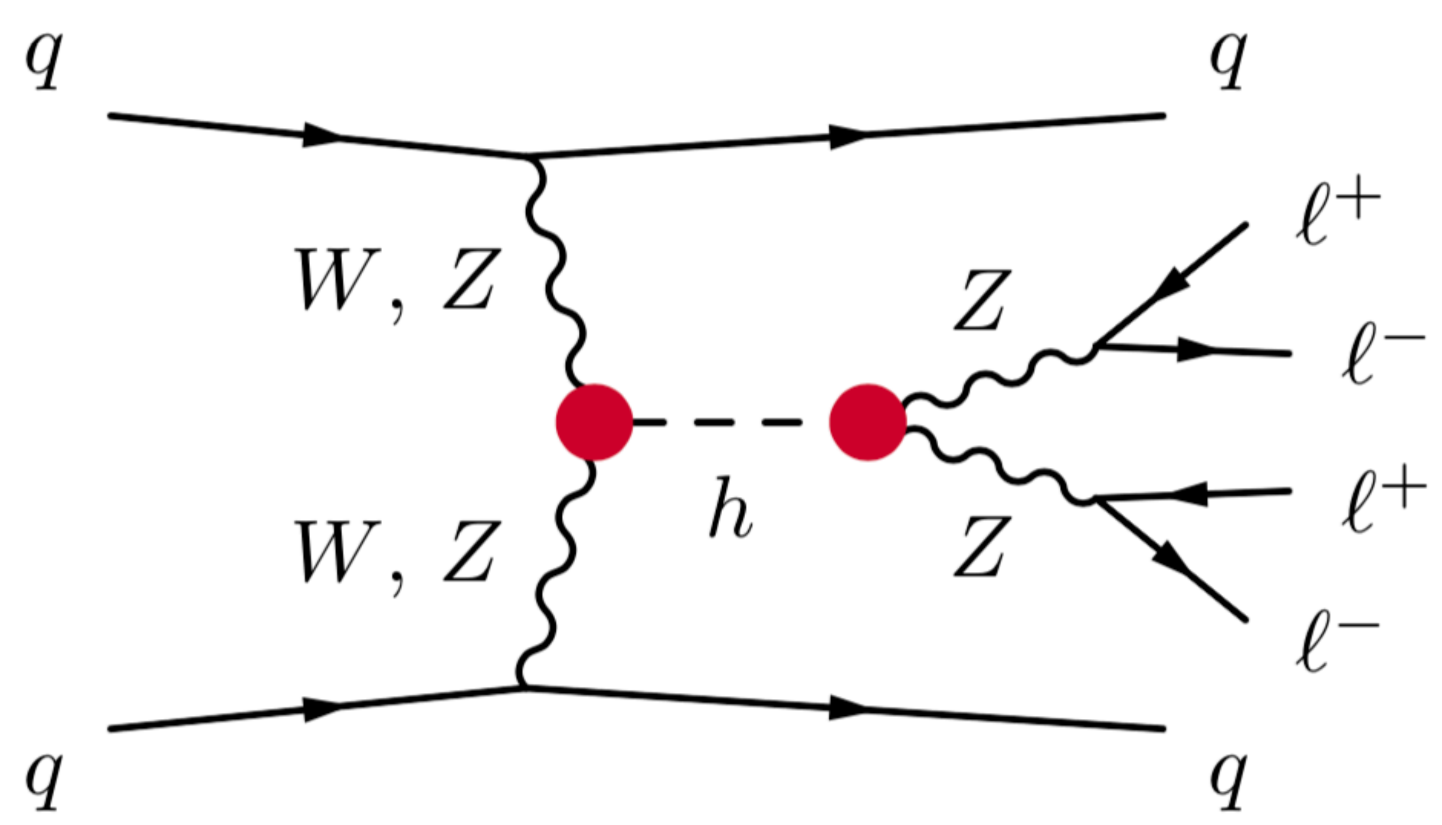
Evolution

Modelling particle physics processes

Latent variables

Parton-level momenta | Theory parameters

$z_p \leftarrow \theta$

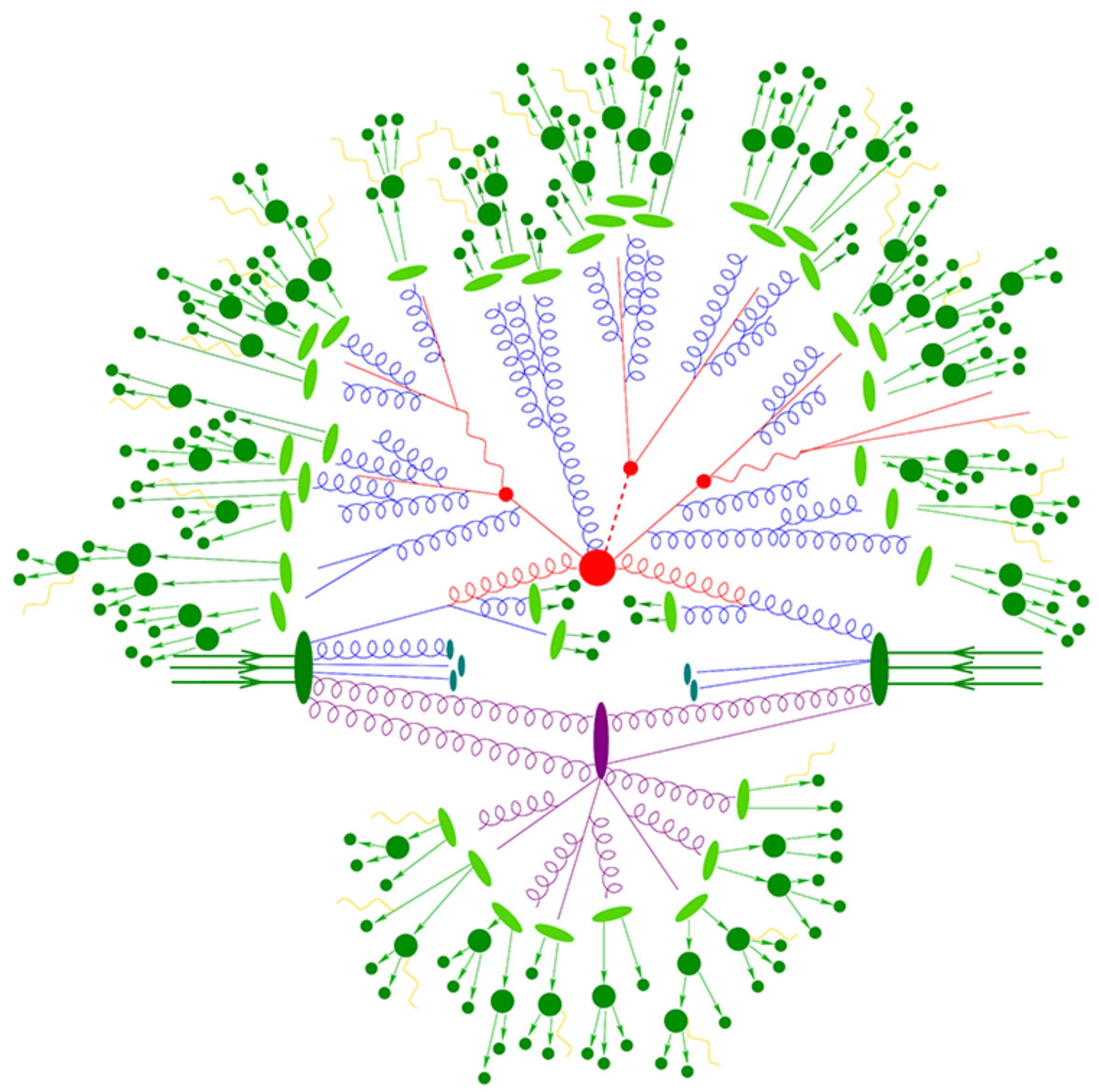


Evolution

Modelling particle physics processes

Latent variables

Shower splittings	Parton-level momenta	Theory parameters
z_s ←	z_p ←	θ



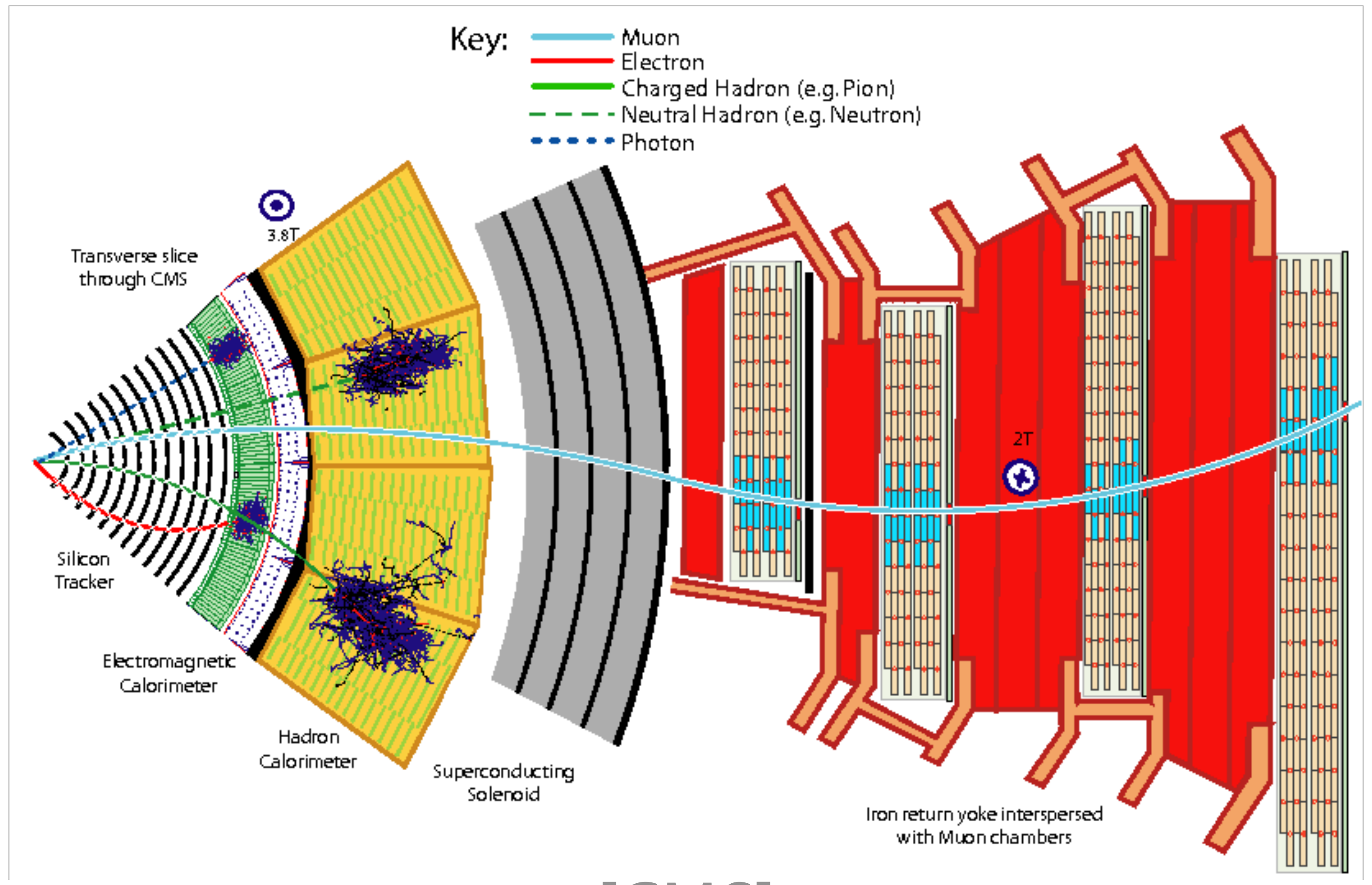
[F. Krauss]

← Evolution

Modelling particle physics processes

Latent variables

Detector interactions z_d ← Shower splittings z_s ← Parton-level momenta z_p ← Theory parameters θ



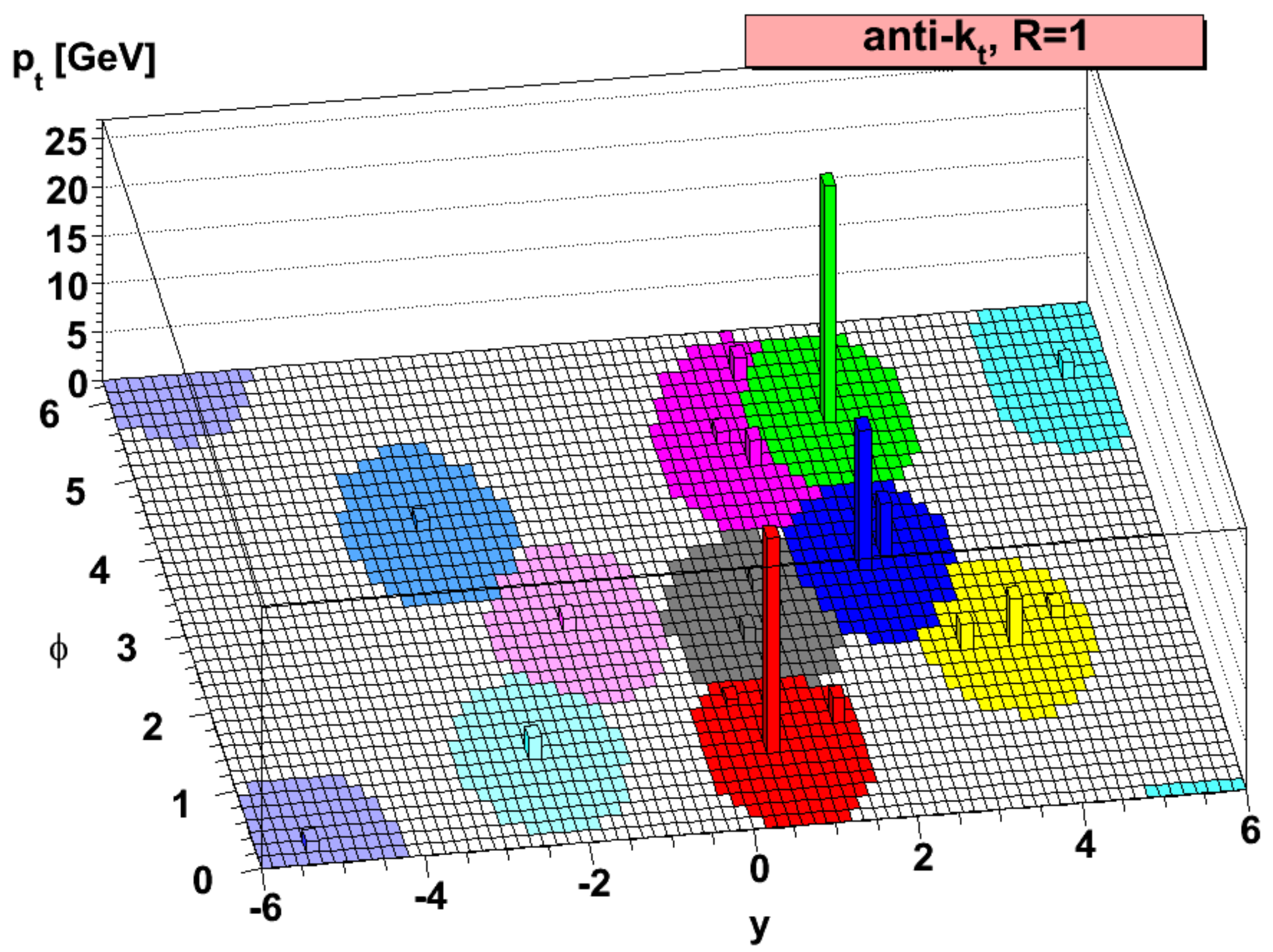
[CMS]

← Evolution

Modelling particle physics processes

Latent variables

Observables		Detector interactions		Shower splittings		Parton-level momenta		Theory parameters
x	←	z_d	←	z_s	←	z_p	←	θ



← Evolution

Modelling particle physics processes

Latent variables

Observables	Detector interactions	Shower splittings	Parton-level momenta	Theory parameters
x ←	z_d ←	z_s ←	z_p ←	θ

$$p(x|\theta) = \int \mathrm{d}z_d \int \mathrm{d}z_s \int \mathrm{d}z_p \quad p(x|z_d) \qquad p(z_d|z_s) \qquad p(z_s|z_p) \qquad p(z_p|\theta)$$

Inference →

Modelling particle physics processes

Latent variables

Observables	Detector interactions	Shower splittings	Parton-level momenta	Theory parameters
x ←	z_d ←	z_s ←	z_p ←	θ

$$p(x|\theta) = \boxed{\int \mathrm{d}z_d \int \mathrm{d}z_s \int \mathrm{d}z_p} \; p(x|z_d) \qquad p(z_d|z_s) \qquad p(z_s|z_p) \qquad p(z_p|\theta)$$

It's infeasible to calculate the integral over this enormous space!

Inference →

Mining gold from the simulator

Latent variables

Observables		Detector interactions		Shower splittings		Parton-level momenta		Theory parameters
x	$\longleftarrow$	z_d	$\longleftarrow$	z_s	$\longleftarrow$	z_p	$\longleftarrow$	θ

$$p(x|\theta) = \int \mathrm{d}z_d \int \mathrm{d}z_s \int \mathrm{d}z_p \quad p(x|z_d) \quad p(z_d|z_s) \quad p(z_s|z_p) \quad \boxed{p(z_p|\theta)}$$

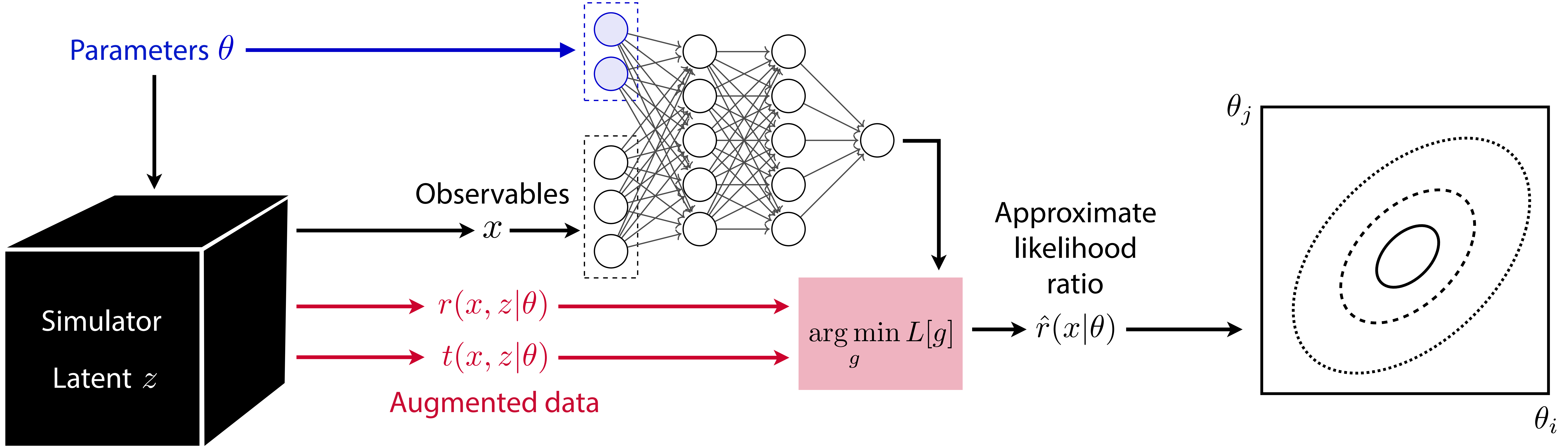
Parton-level likelihood is given by matrix element and can be evaluated!

⇒ For each simulated event, we can calculate the joint likelihood ratio which depends on the specific evolution of the simulation:

$$r(x, z|\theta_0, \theta_1) \equiv \frac{p(x, z_d, z_s, z_p|\theta_0)}{p(x, z_d, z_s, z_p|\theta_1)} = \frac{p(x|z_d)}{p(x|z_d)} \frac{p(z_d|z_s)}{p(z_d|z_s)} \frac{p(z_s|z_p)}{p(z_s|z_p)} \boxed{\frac{p(z_p|\theta_0)}{p(z_p|\theta_1)} \sim \frac{|\mathcal{M}(z_p|\theta_0)|^2}{|\mathcal{M}(z_p|\theta_1)|^2}}$$

if we knew the entire history of each event

Bird's-eye view



1. Simulation

"Mining gold": Extract additional information from simulator

2. Machine Learning

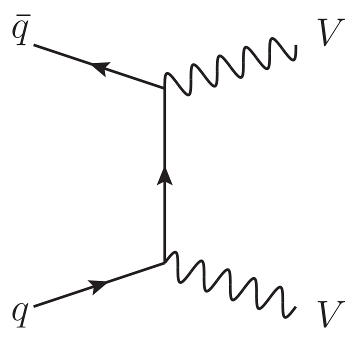
Use this information to train estimator for likelihood ratio

3. Inference

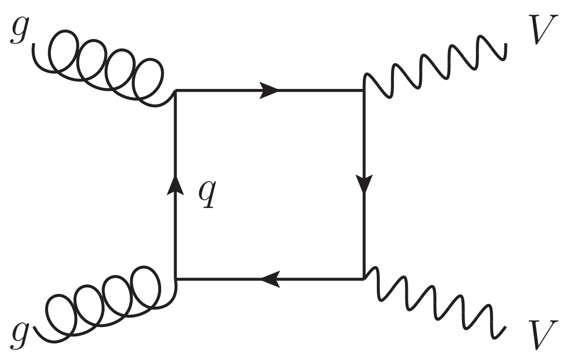
Limit setting with standard hypothesis tests

Outline

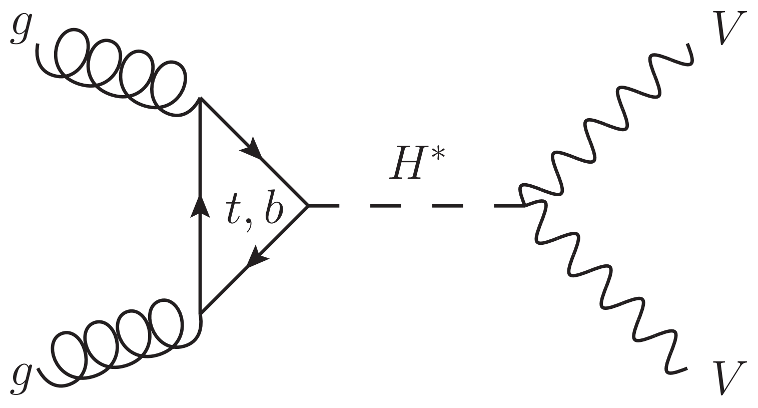
1. **The problem of interference in the offshell H4l analysis**
2. **Introduce Likelihood-free Inference with Madminer: “ML version of Matrix Element Method”**
3. **Preliminary Results**
4. **Future**



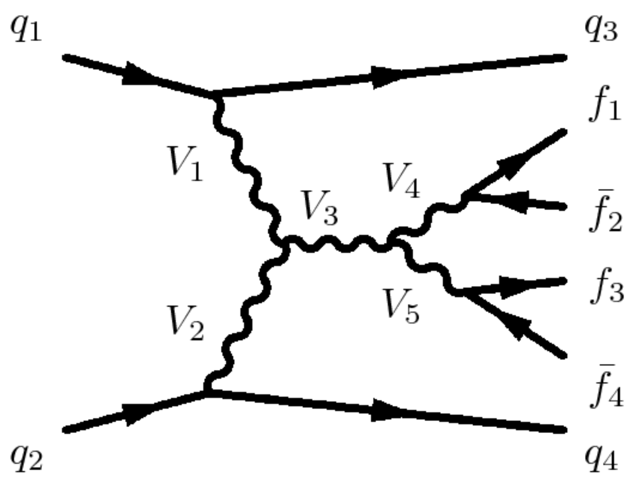
qqZZ



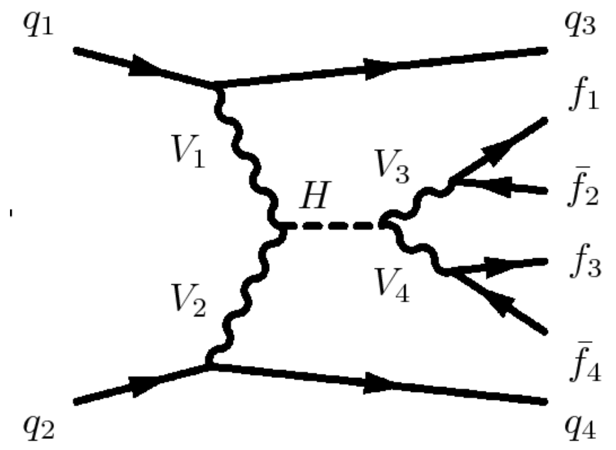
ggZZ



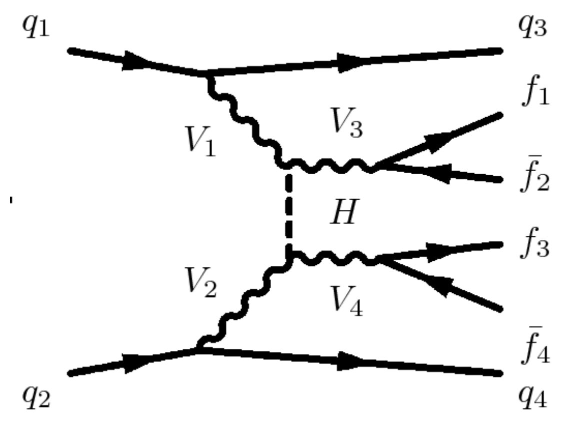
ggF



VBS

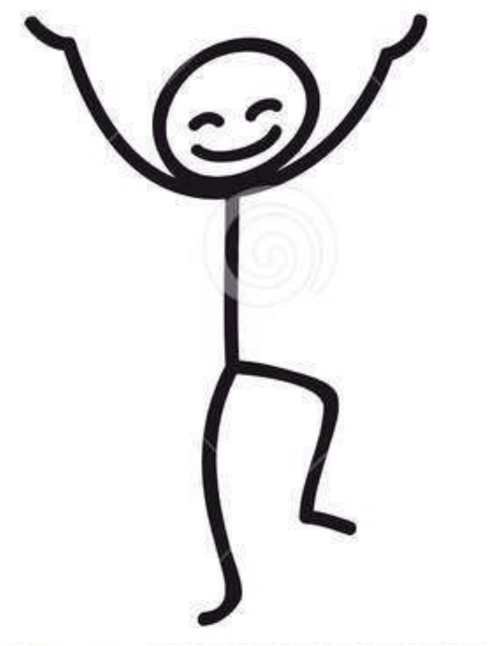


VBF (“s-channel”)



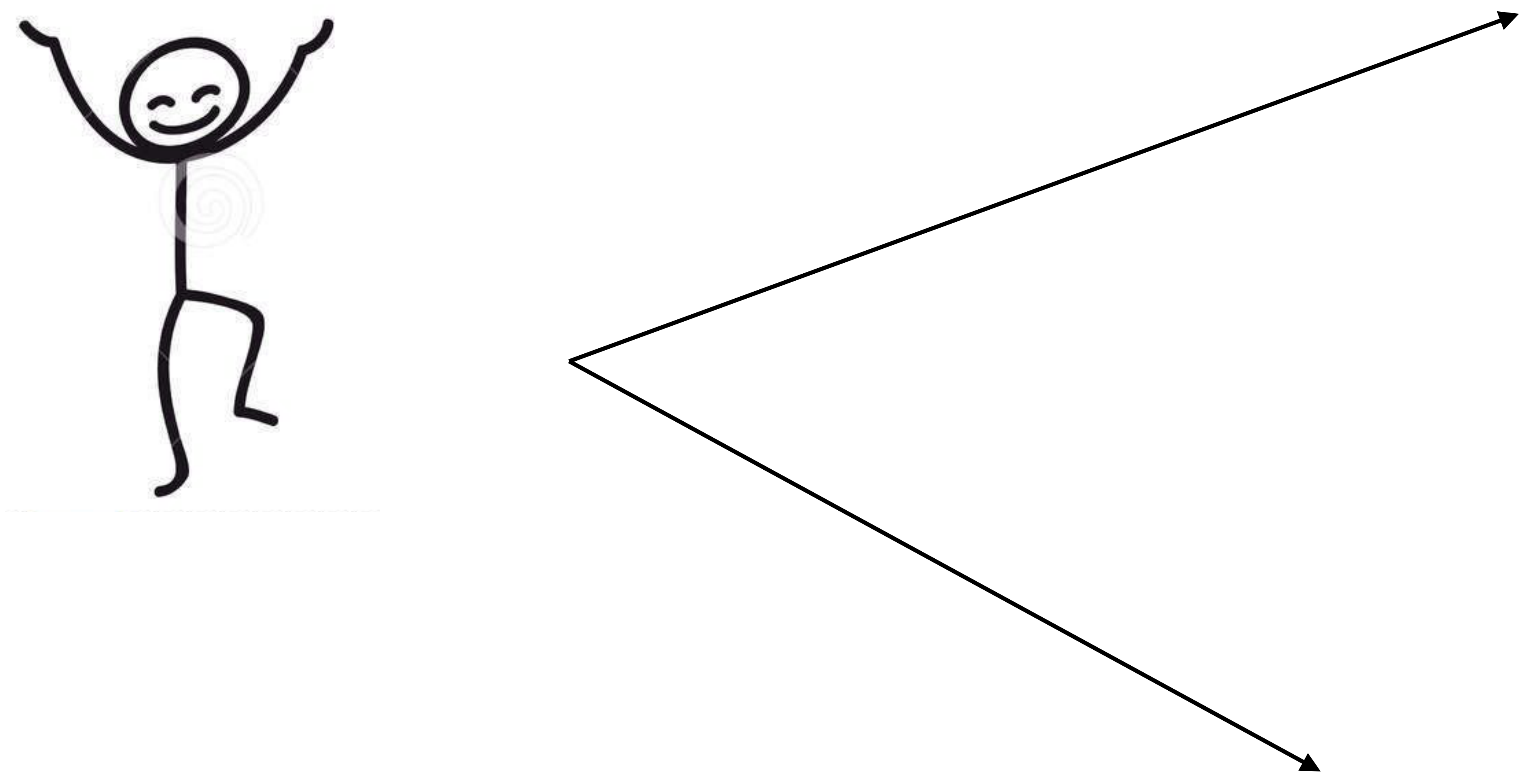
VBF (“t-channel”)

Quantum Interference (Destructive)



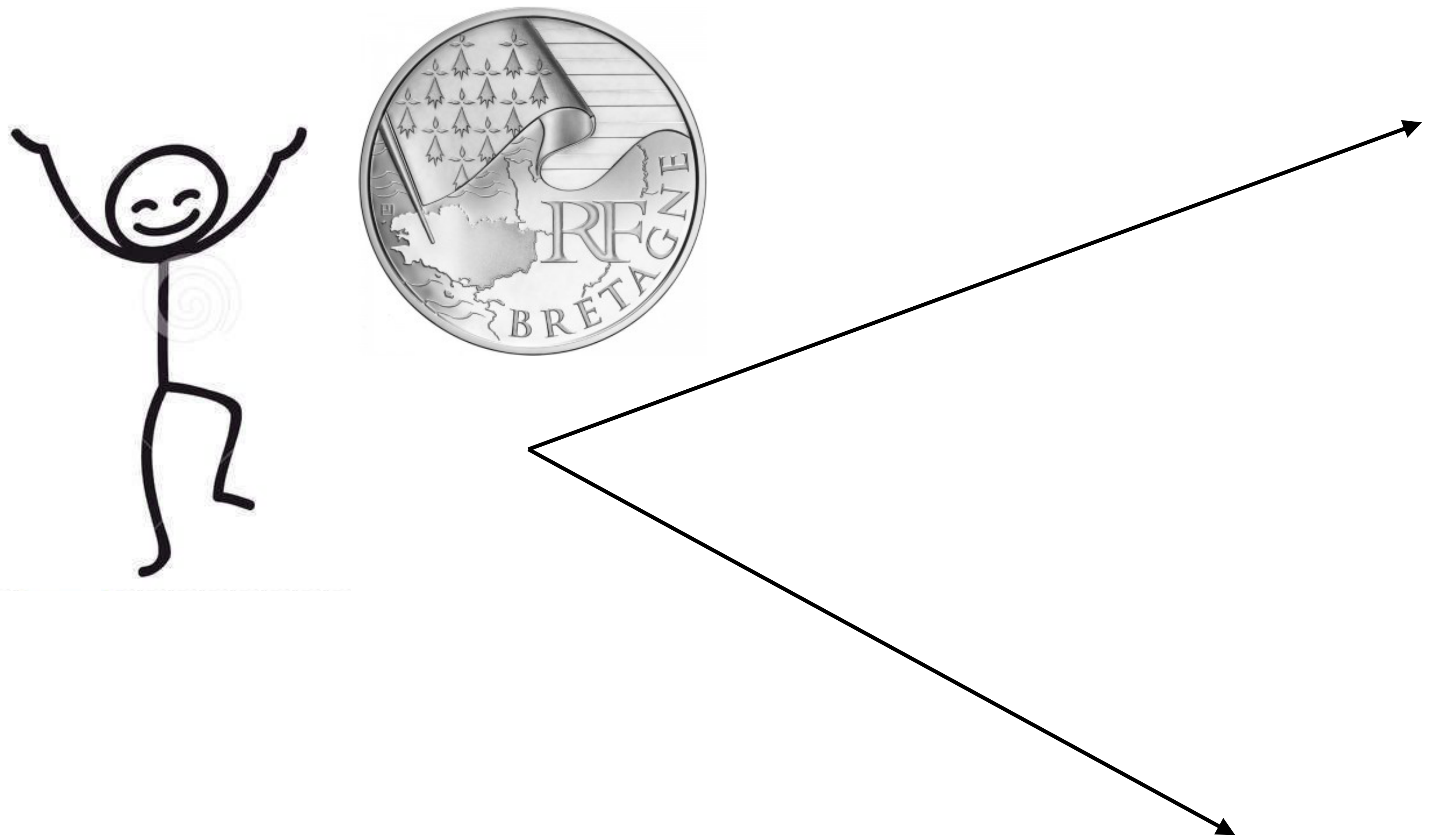
Not to be confused with a
classical, time ordered
simulation as before

Quantum Interference (Destructive)



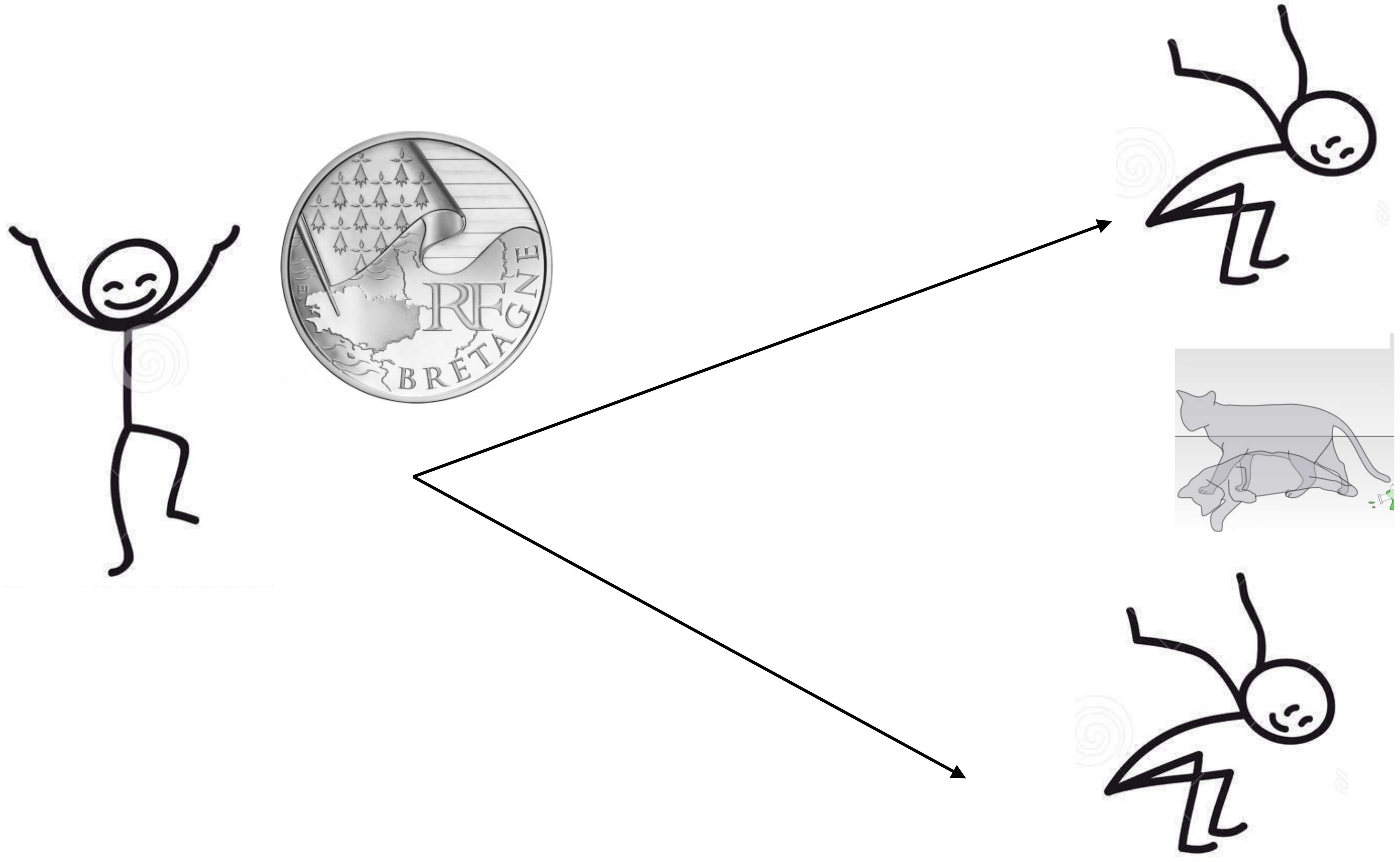
Not to be confused with a
classical, time ordered
simulation as before

Quantum Interference (Destructive)



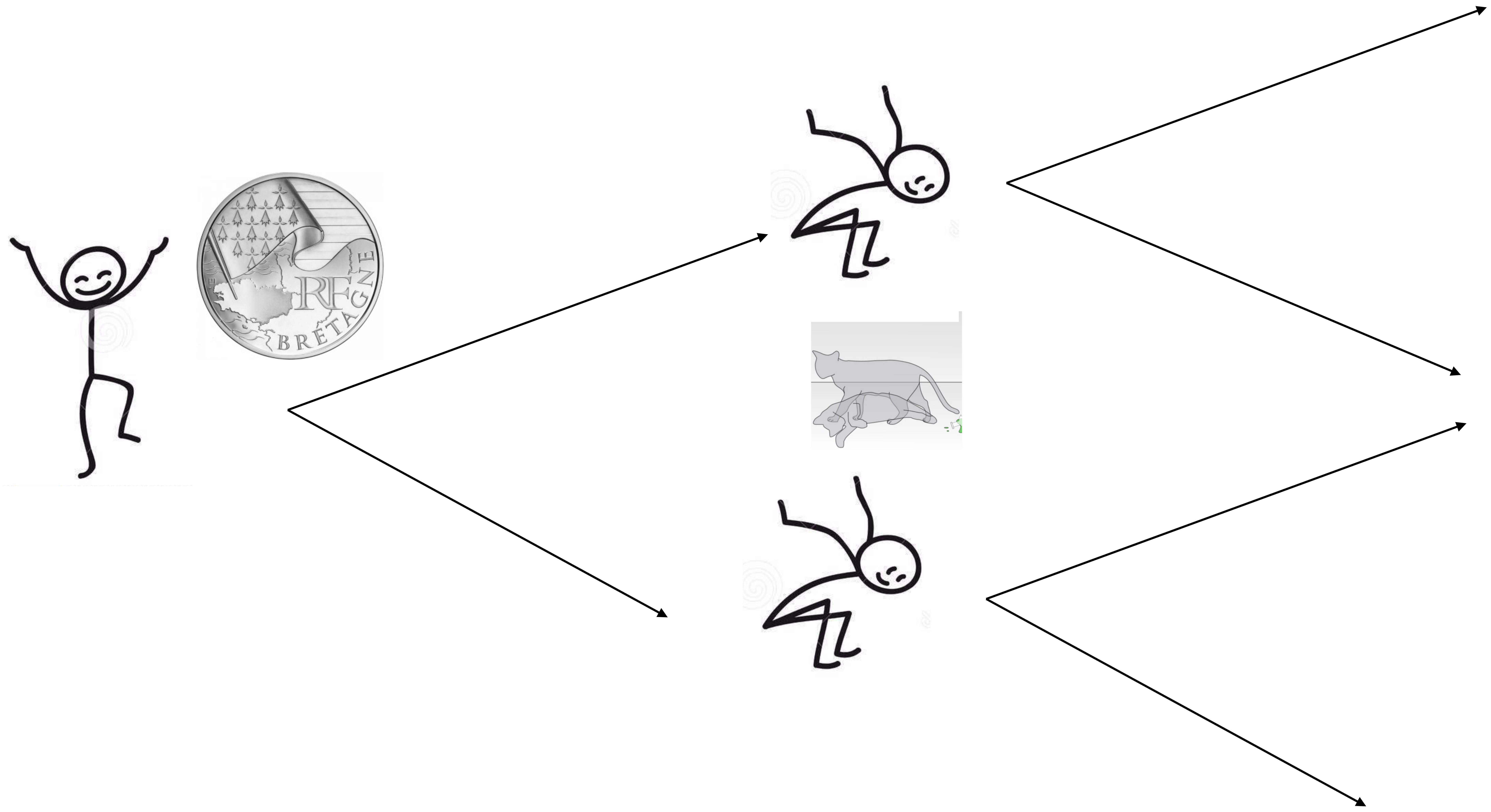
Not to be confused with a
classical, time ordered
simulation as before

Quantum Interference (Destructive)



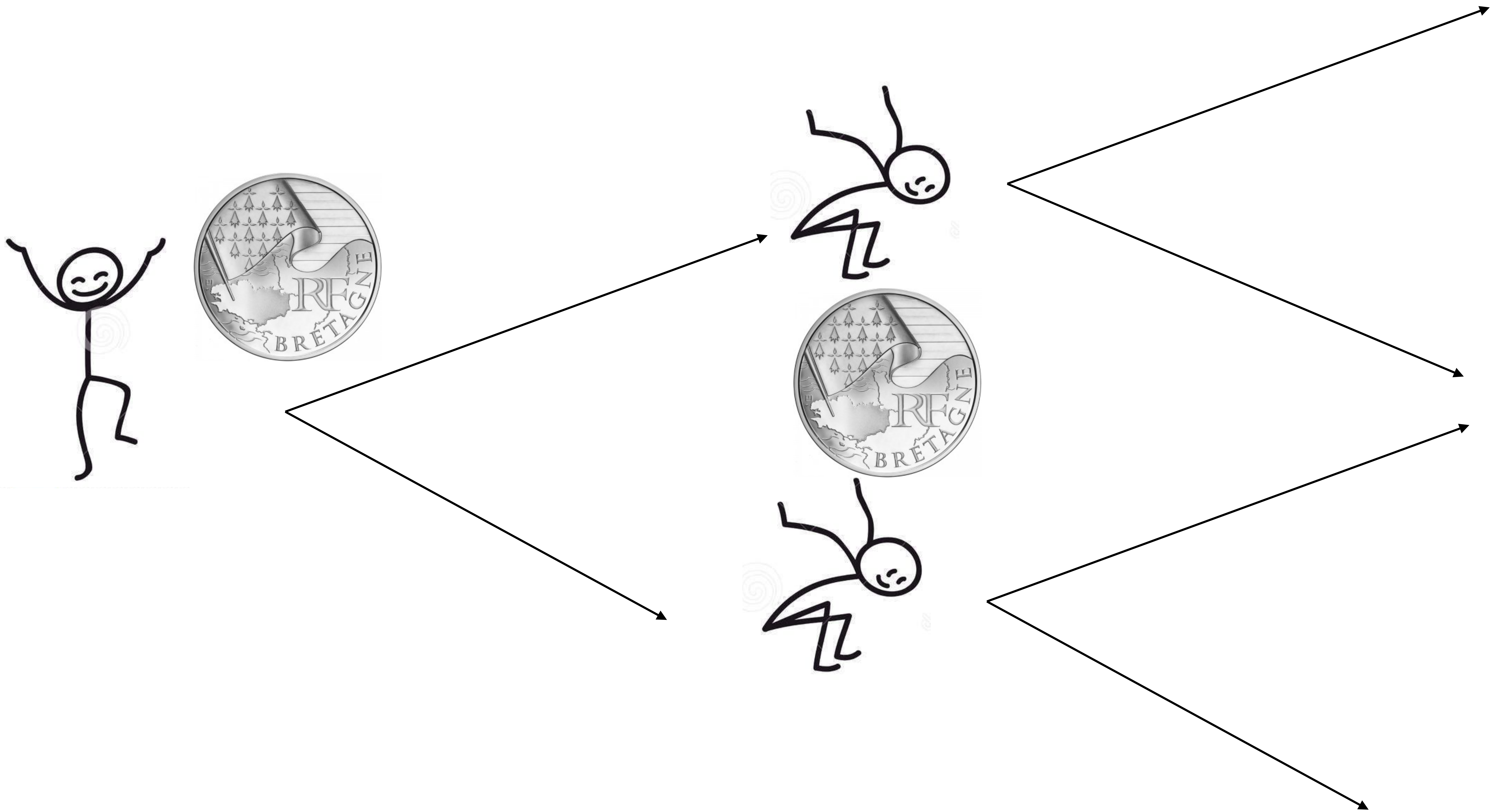
Not to be confused with a classical, time ordered simulation as before

Quantum Interference (Destructive)



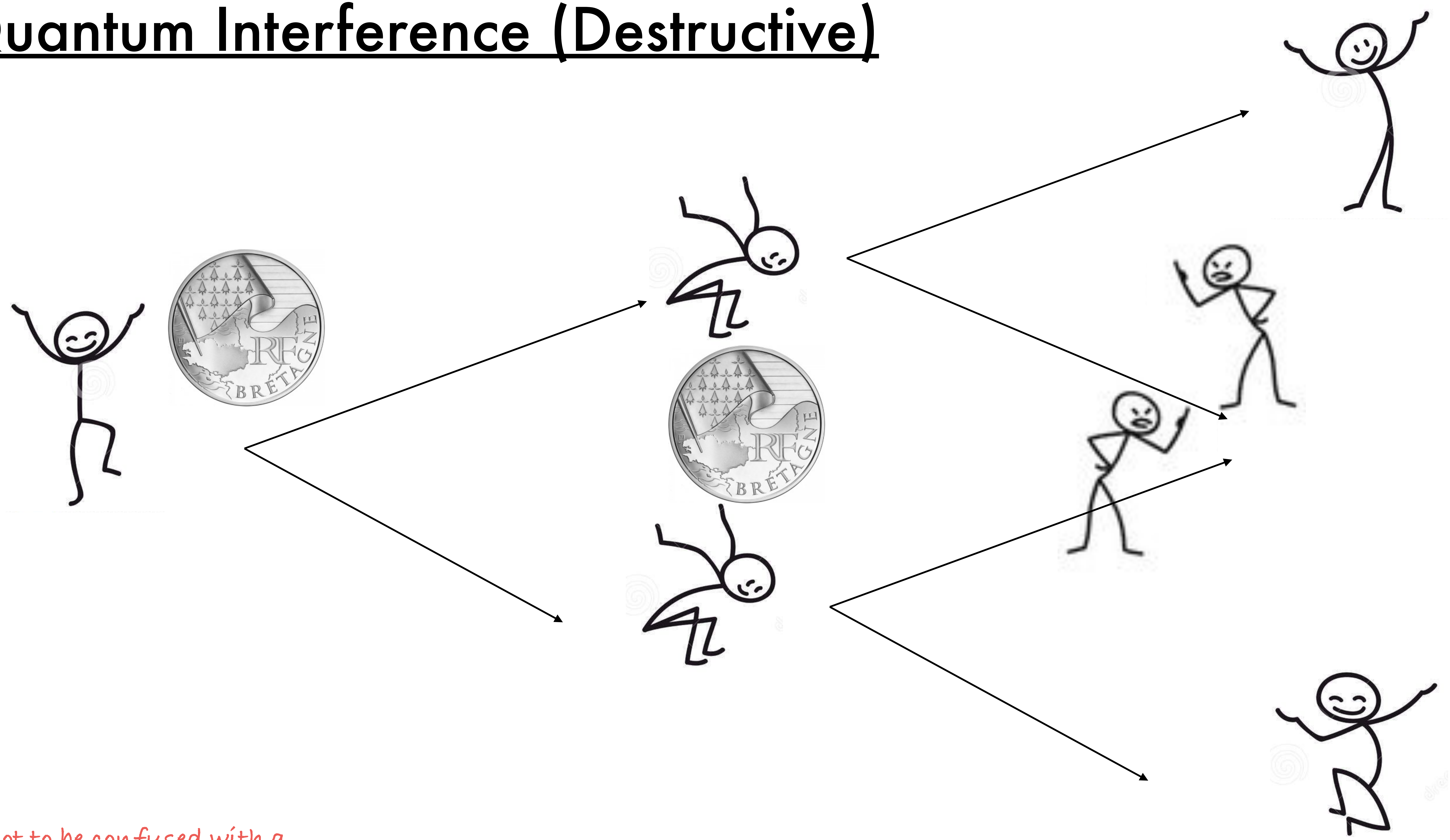
Not to be confused with a
classical, time ordered
simulation as before

Quantum Interference (Destructive)



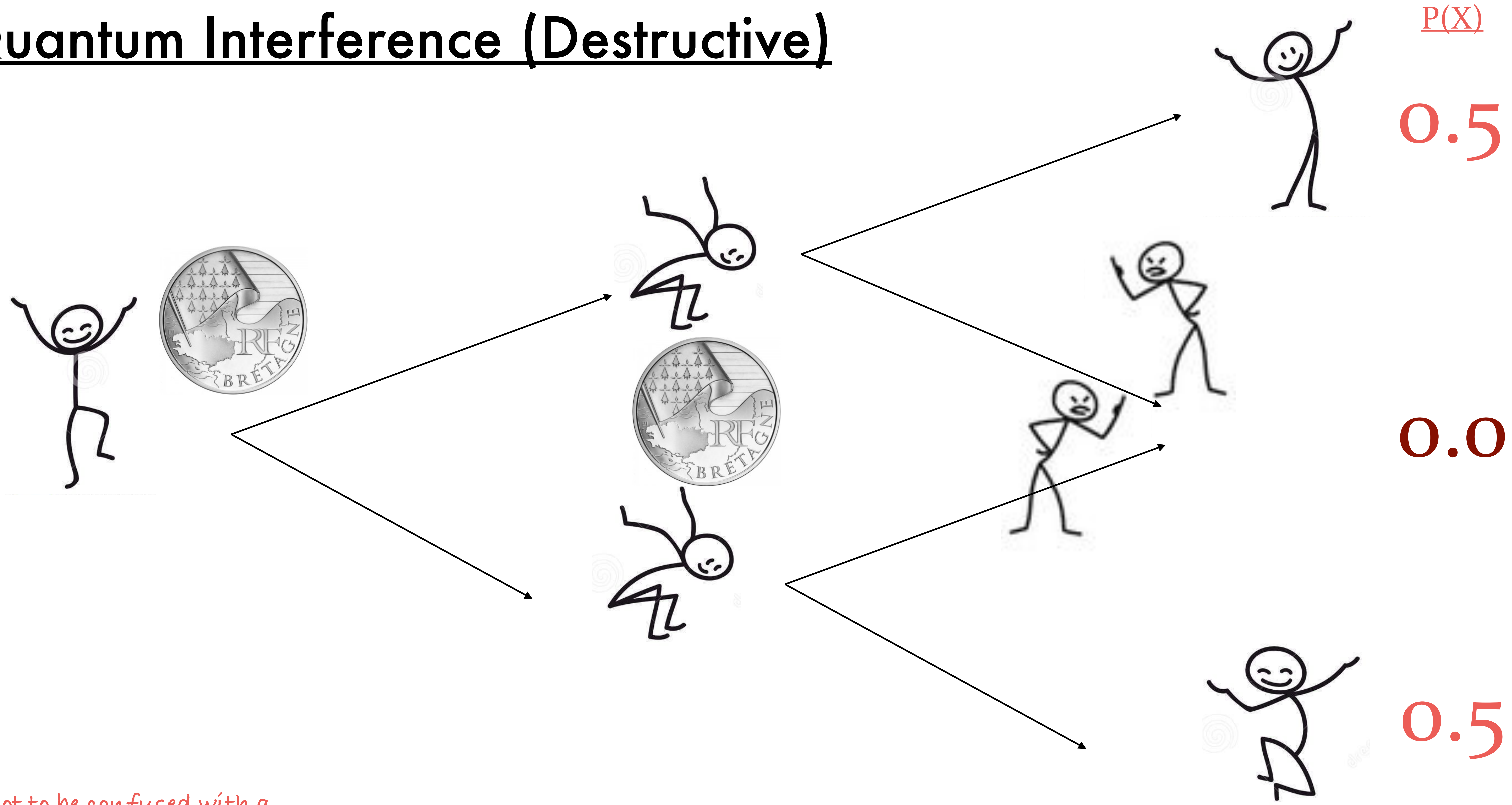
Not to be confused with a
classical, time ordered
simulation as before

Quantum Interference (Destructive)



Not to be confused with a
classical, time ordered
simulation as before

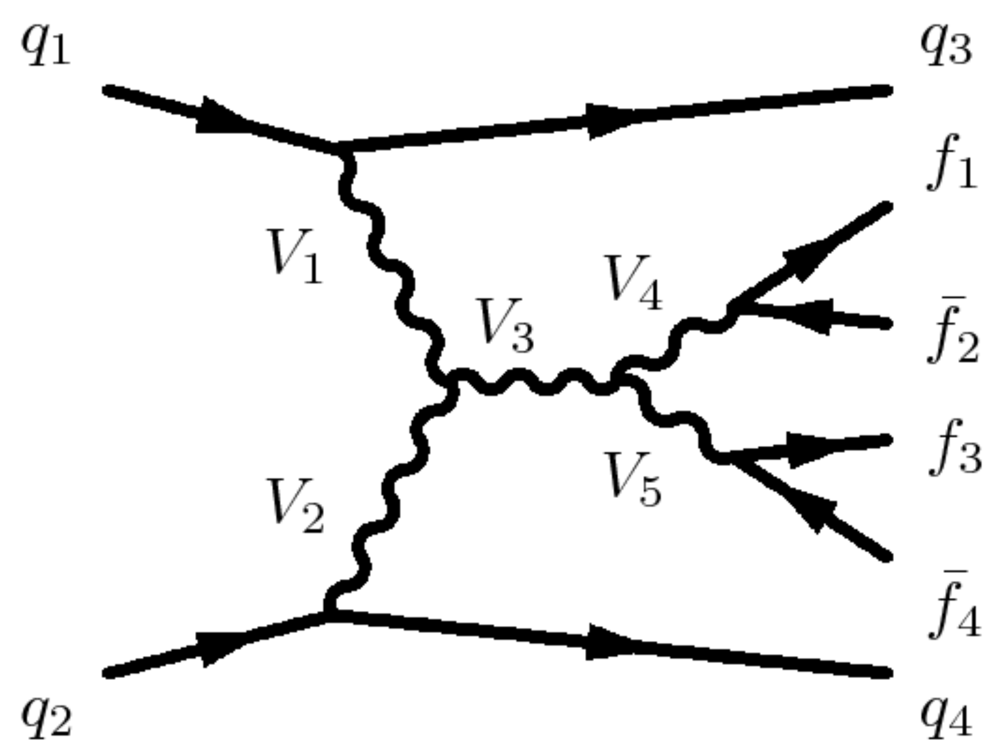
Quantum Interference (Destructive)



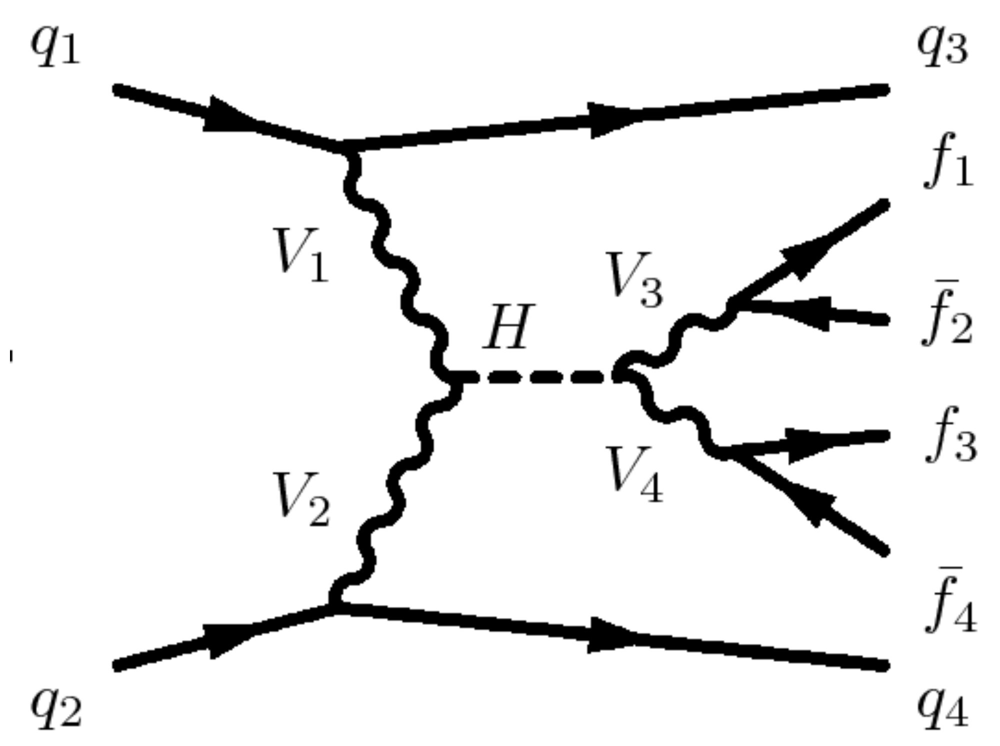
Not to be confused with a classical, time ordered simulation as before

Our problem: Interference between VBS, VBF in Higgs to 4 leptons analysis

Main objective: To **measure** Higgs off shell **signal strength (μ)** in the 4 leptons final state. I'm concentrating on the VBF production mode for now, will expand also to ggF



Vector Boson Propagator
(Background)



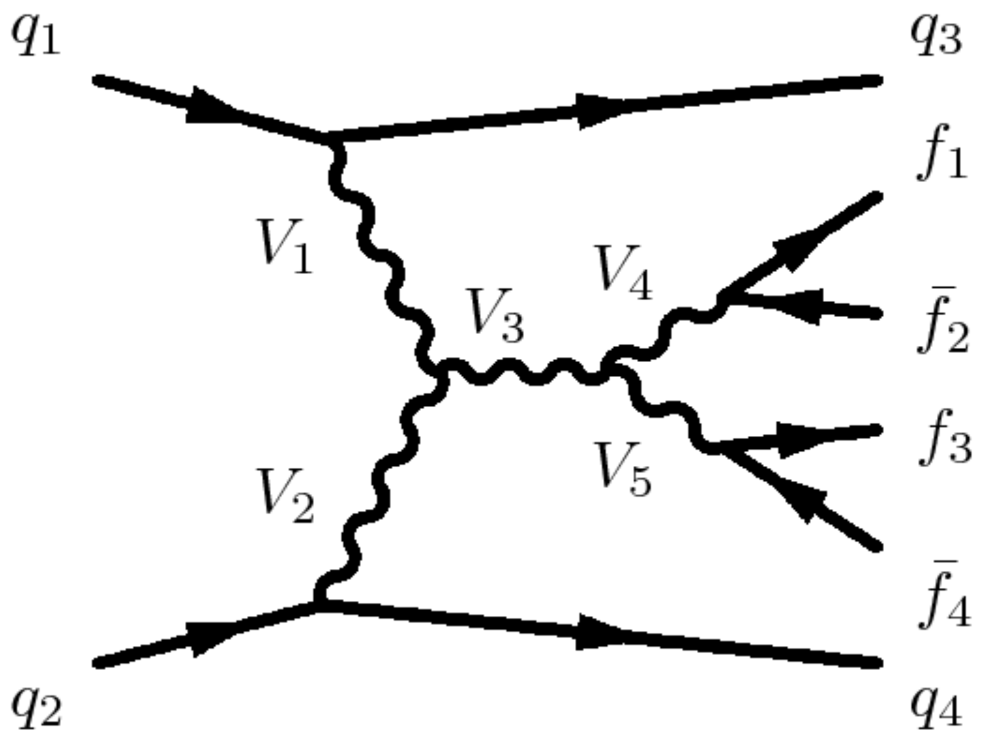
Higgs Propagator
(Signal)

S = VBF-Higgs, B= VBS, SBI = Combined Simulation

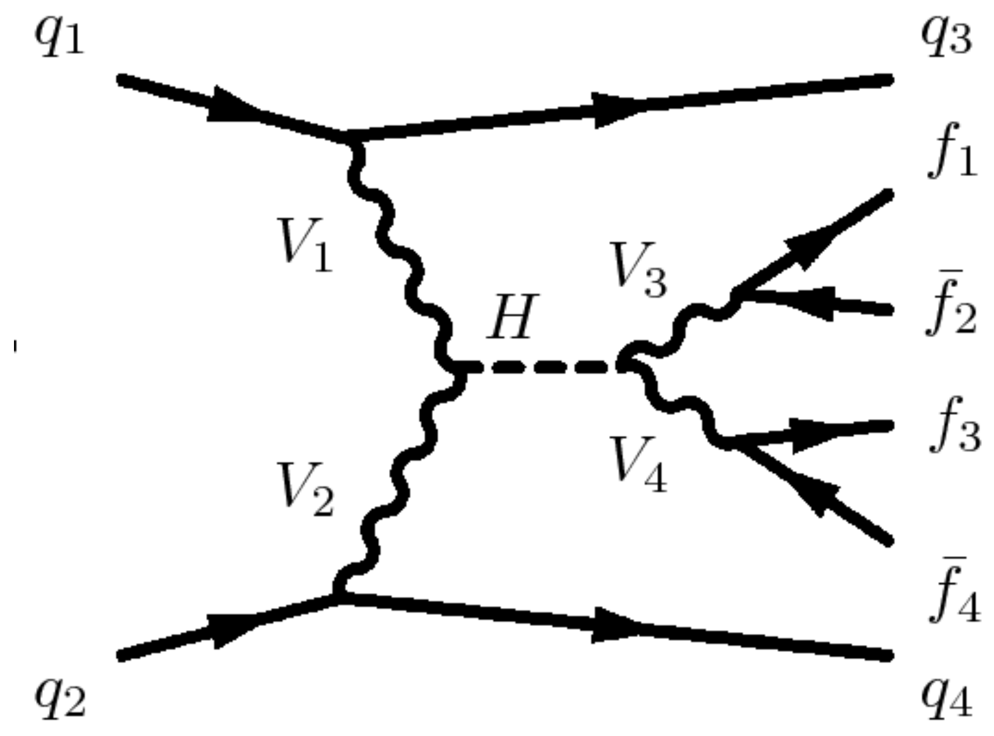
I = SBI - S - B

Our problem: Interference between VBS, VBF in Higgs to 4 leptons analysis

Main objective: To **measure** Higgs off shell **signal strength (μ)** in the 4 leptons final state. I'm concentrating on the VBF production mode for now, will expand also to ggF

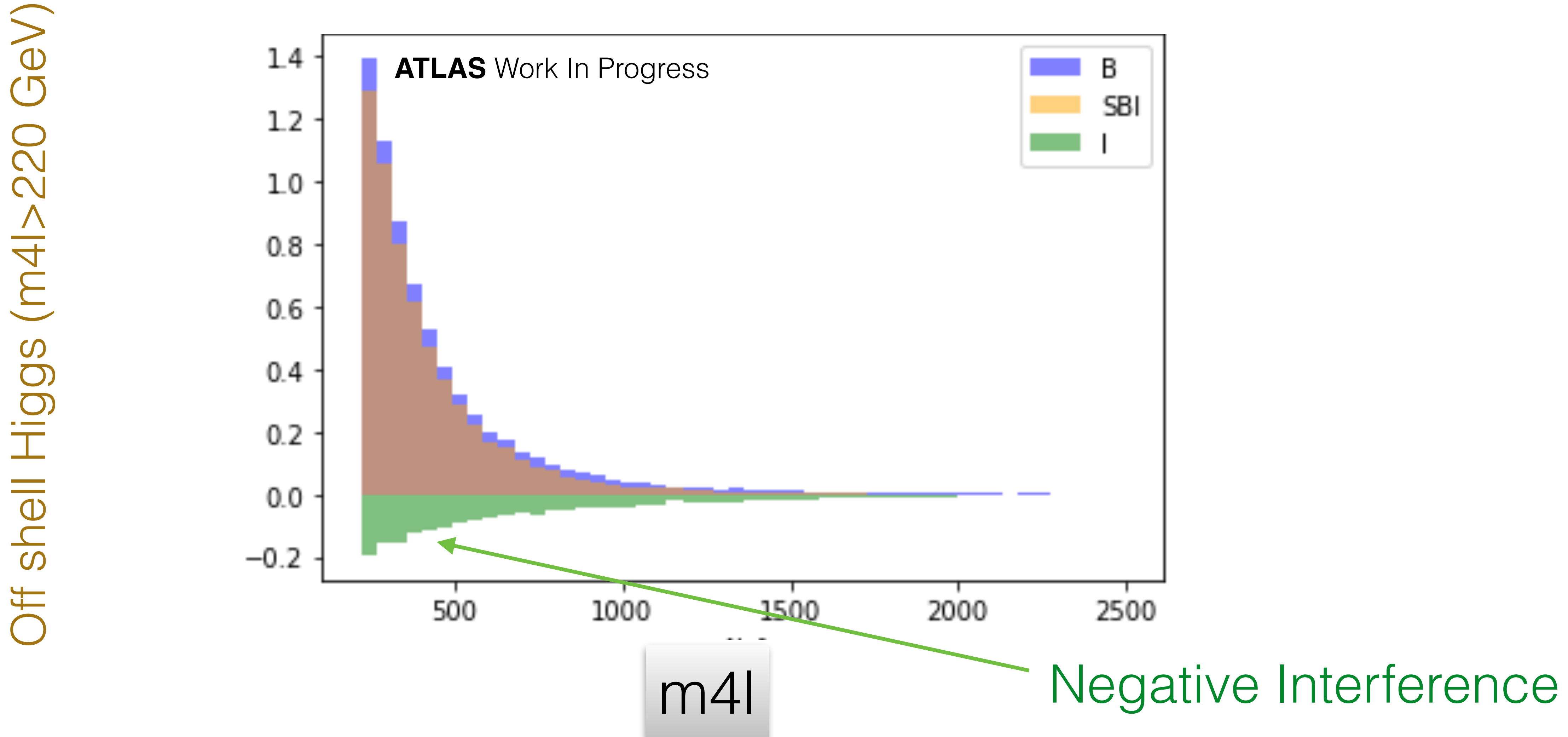


Vector Boson Propagator (Background)



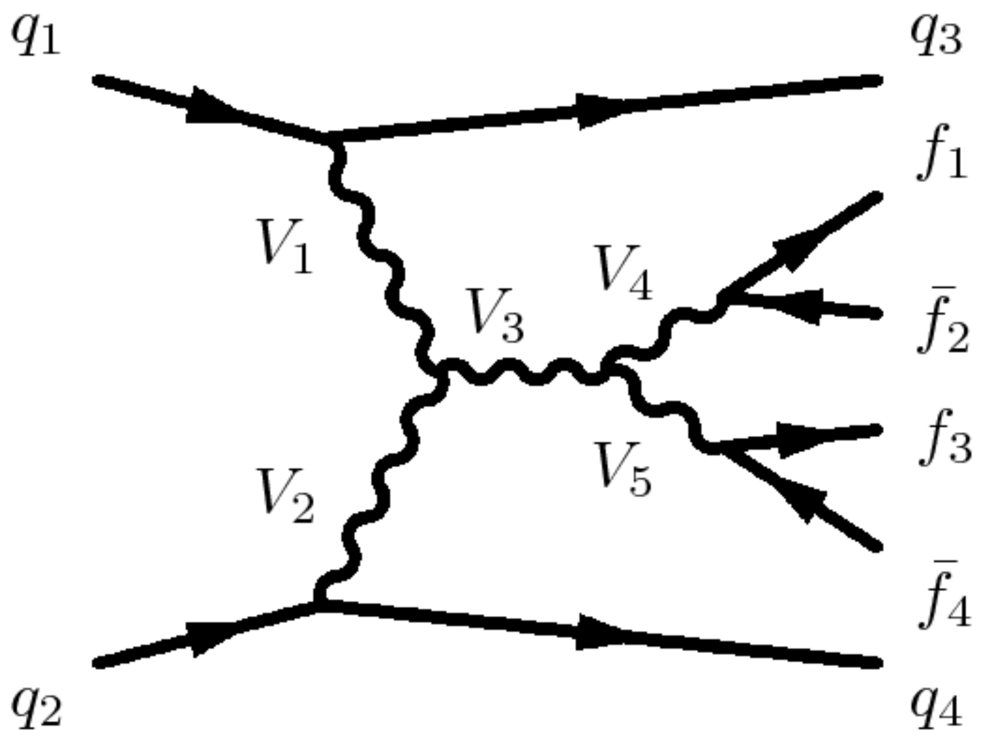
Higgs Propagator (Signal)

S = VBF-Higgs, B= VBS, SBI = Combined Simulation
I = SBI - S - B

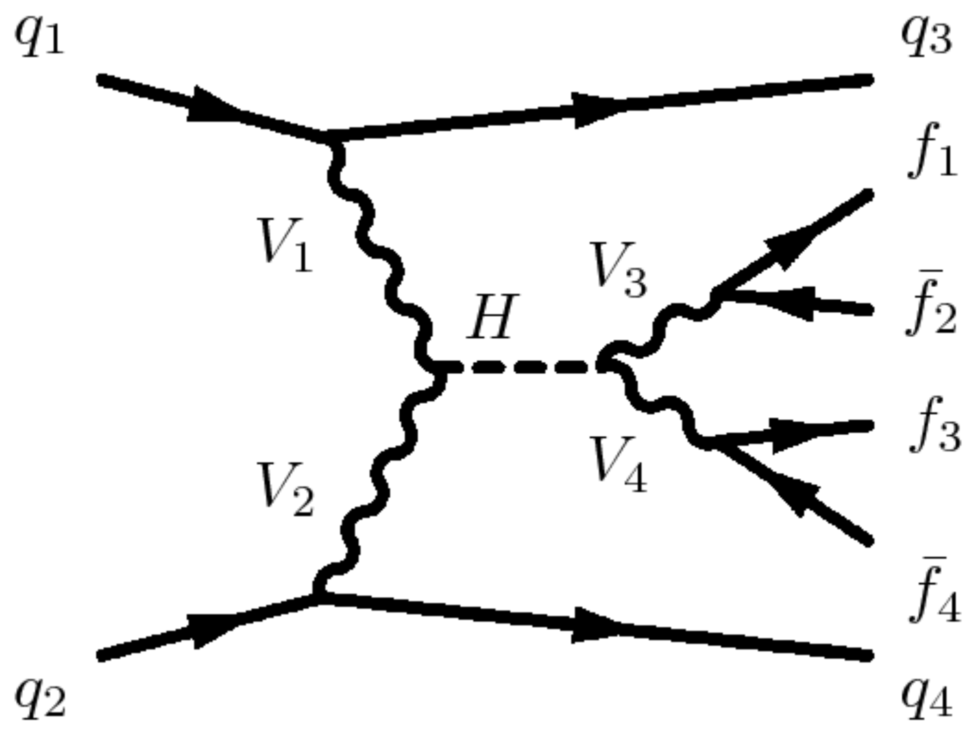


Our problem: Interference between VBS, VBF in Higgs to 4 leptons analysis

Main objective: To **measure** Higgs off shell **signal strength (μ)** in the 4 leptons final state. I'm concentrating on the VBF production mode for now, will expand also to ggF



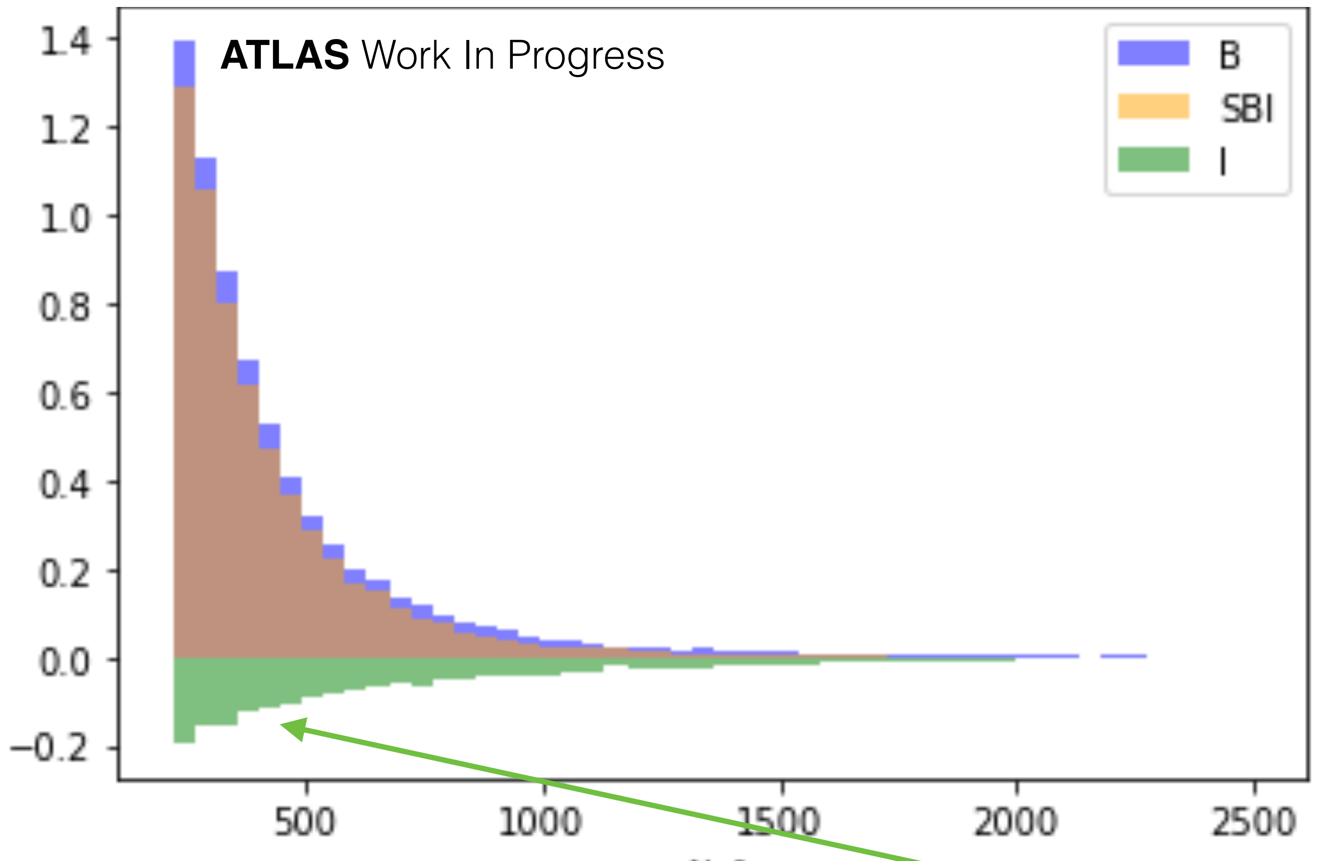
Vector Boson Propagator
(Background)



Higgs Propagator
(Signal)

S = VBF-Higgs, B= VBS, SBI = Combined Simulation
I = SBI - S - B

Off shell Higgs (m4l>220 GeV)



m4l

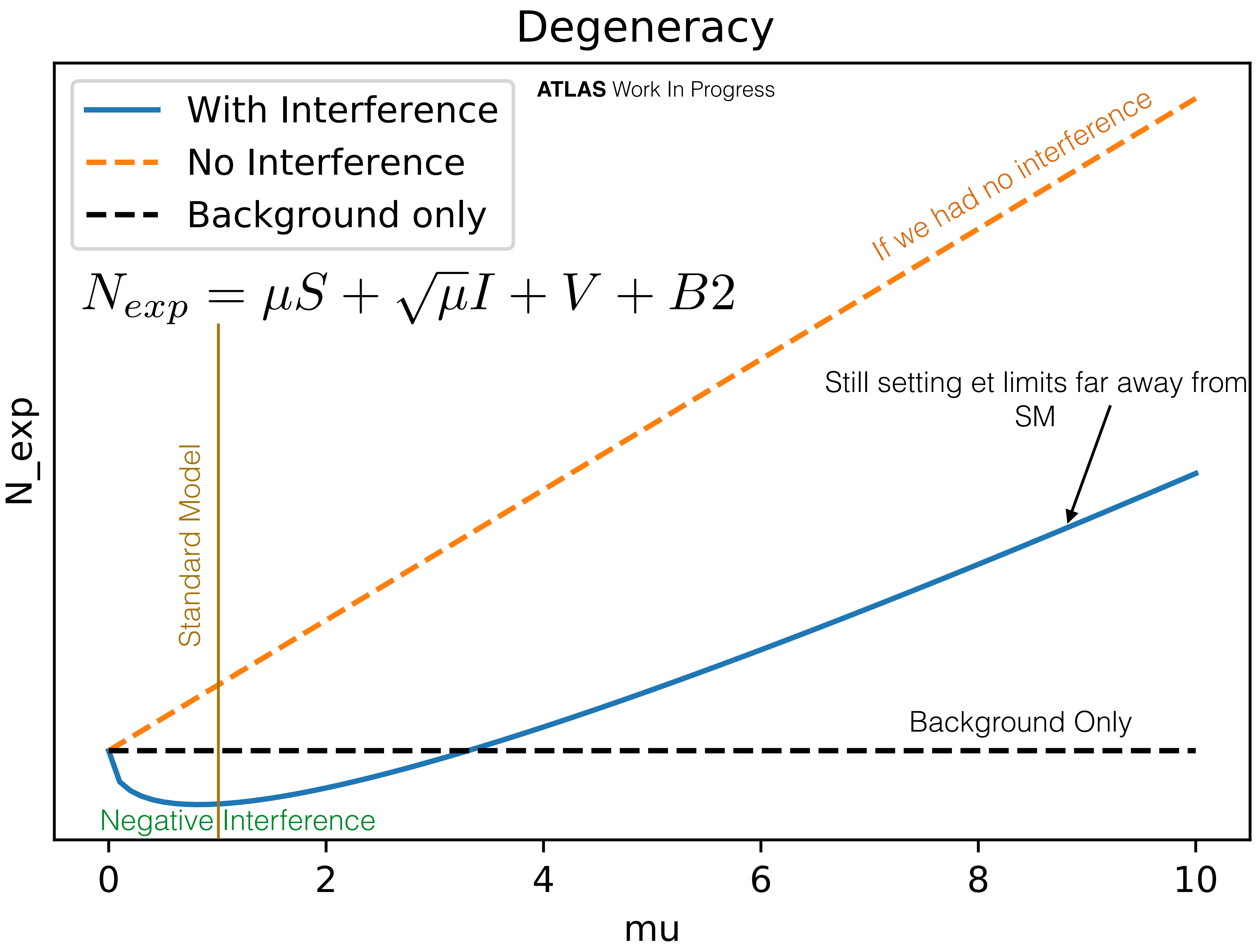
Negative Interference

Only physical dataset is a combined SBI simulation ⇒ **No Class Labels!**

Cannot train an ML classifier

Non-linear N_exp(μ), Degenerate μ

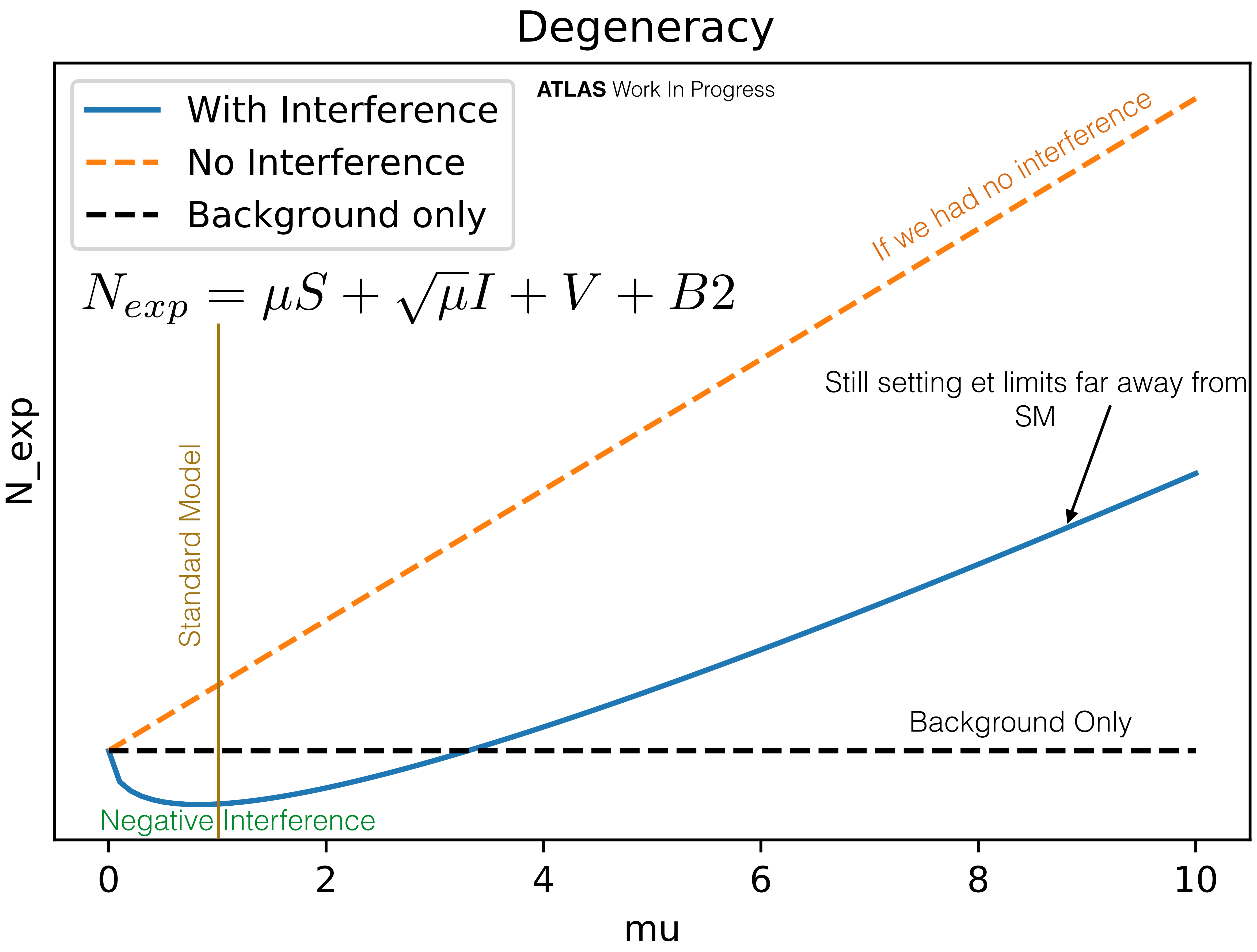
Demonstrative plot: Just a rough sketch



⇒ Negative Log-Likelihood will have 2 minima on Asimov dataset

Non-linear N_exp(μ), Degenerate μ

Demonstrative plot: Just a rough sketch

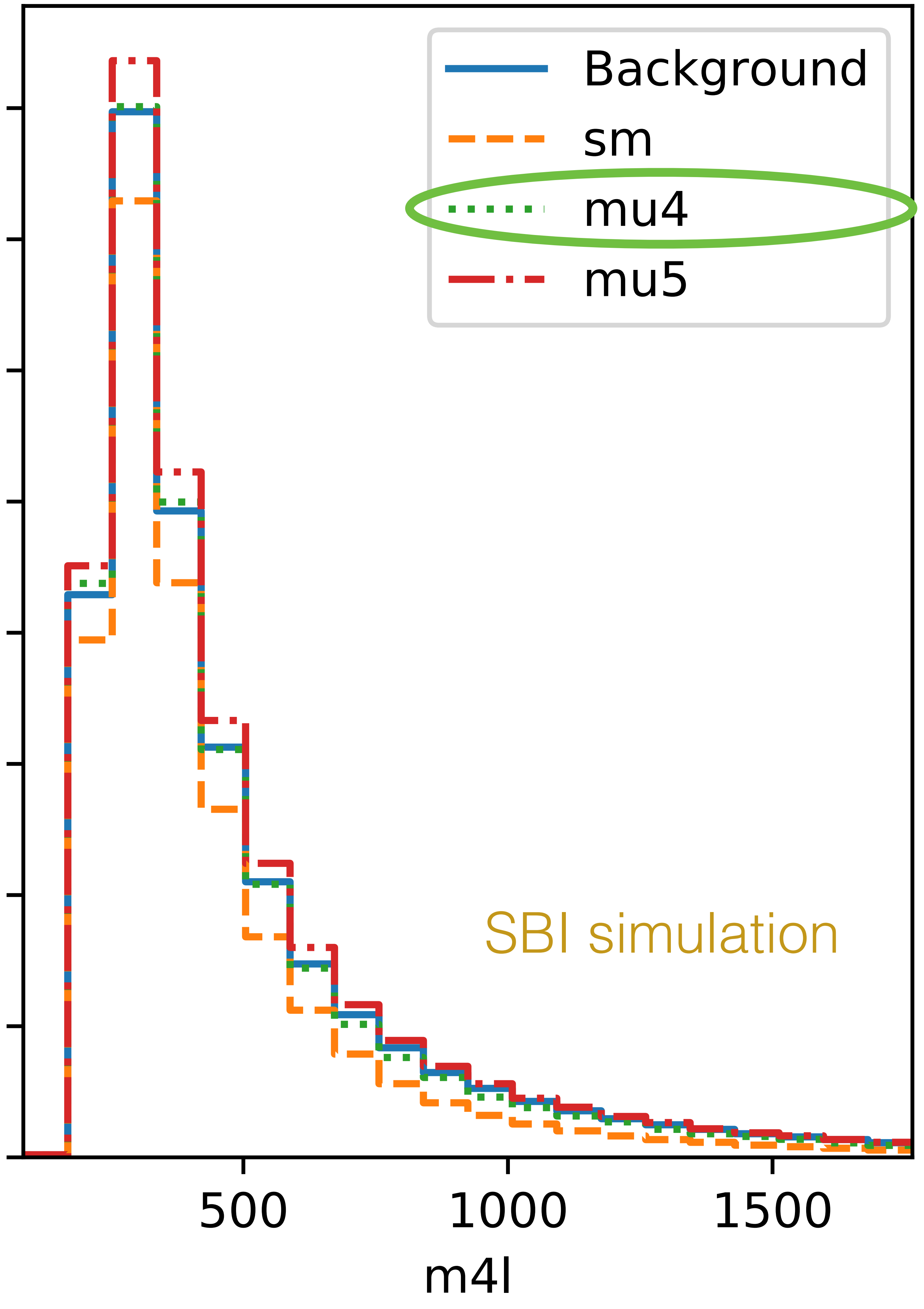


μ is no longer just a scaling, it changes the distributions. **Cannot just look at the SM distributions to optimise a strategy**

⇒ Negative Log-Likelihood will have 2 minima on Asimov dataset

Disclaimer: Private simulations with Madgraph+Pythia+**Delphes**, Not real ATLAS

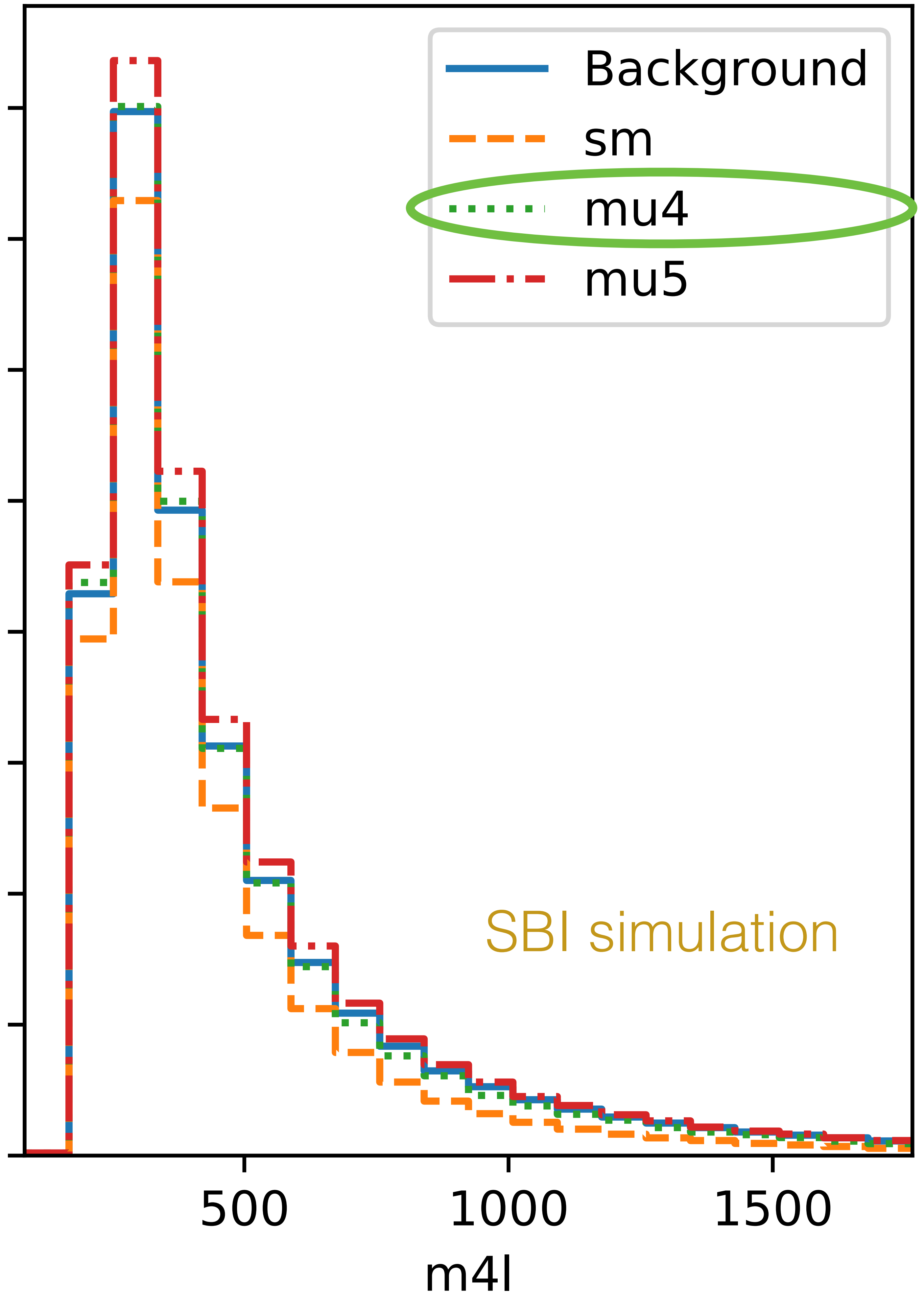
Offshell Higgs signal strength, μ = 0 or μ = 4 ?



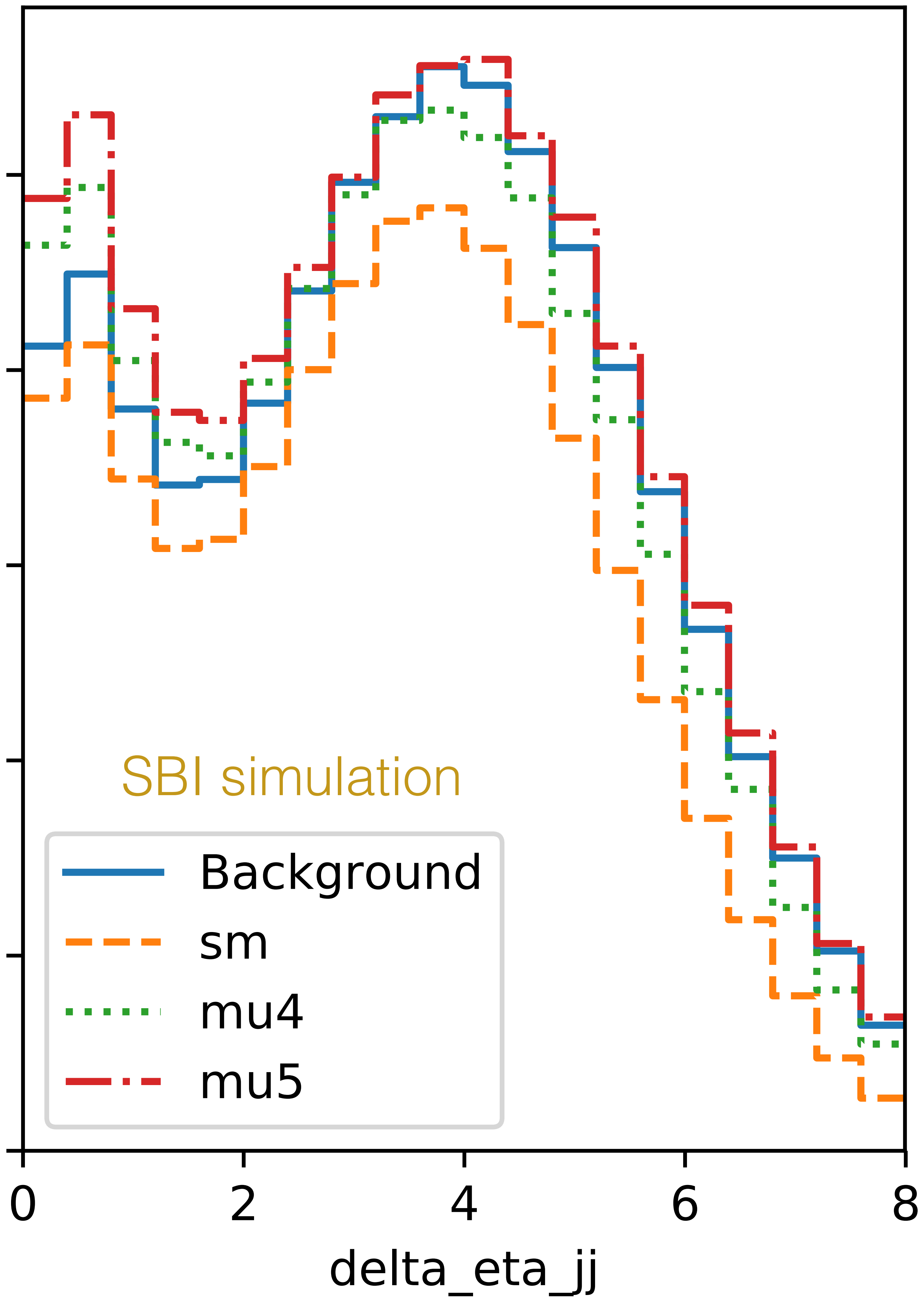
Can you spot the green plot?

Disclaimer: Private simulations with Madgraph+Pythia+**Delphes**, Not real ATLAS

Offshell Higgs signal strength, μ = 0 or μ = 4 ?



Can you spot the green plot?



μ=4 indistinguishable from μ=0 but other observables can break the degeneracy

Disclaimer: Private simulations with Madgraph+Pythia+**Delphes**, Not real ATLAS

p-value Scan on Test Dataset at μ=0.5

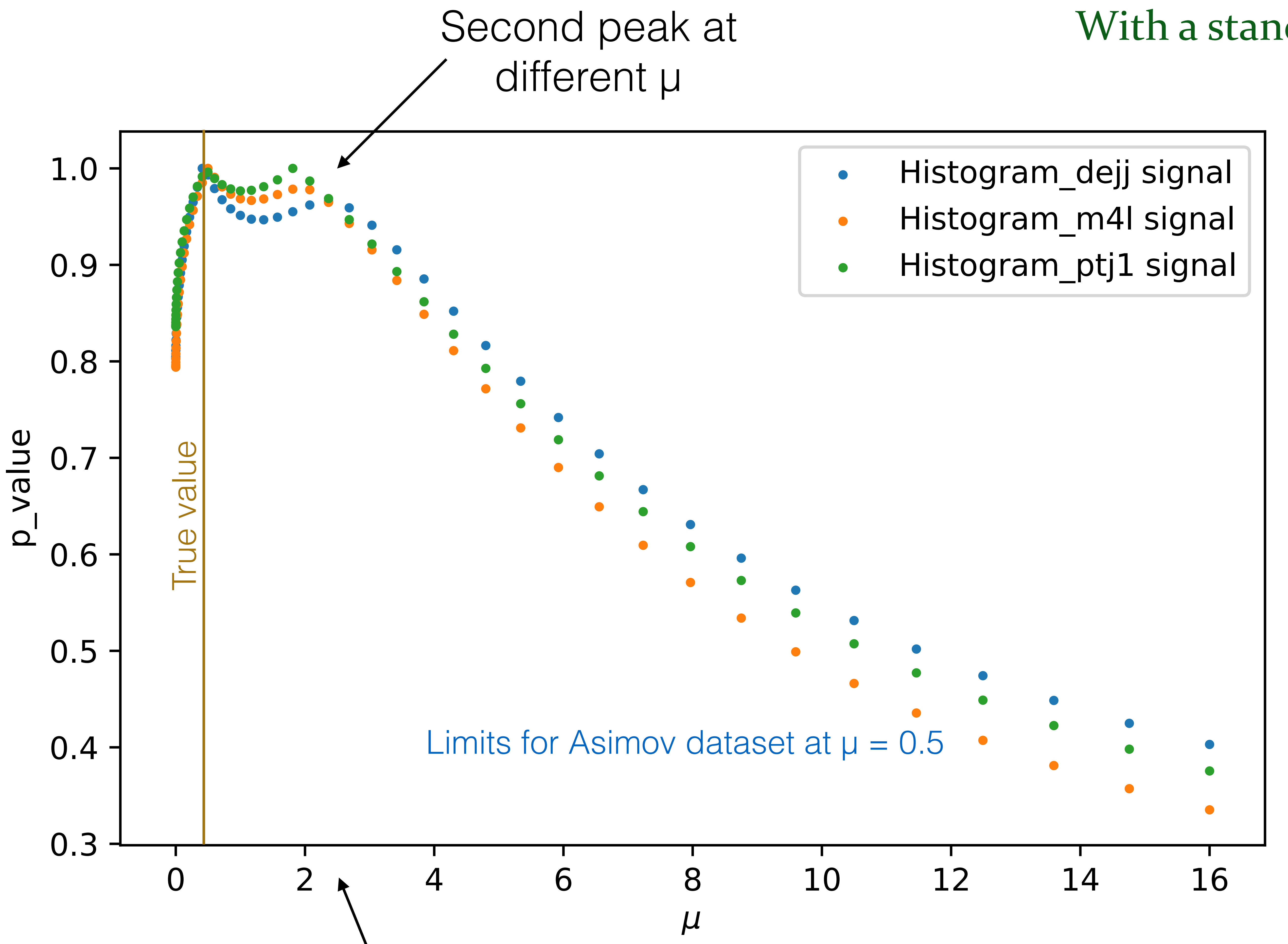
With a standard strategy

We are asking the question:
What is the p-value for the μ=2.5 (for example) on an Asimov dataset generated with true μ=0.5 ?

Disclaimer: Private simulations with Madgraph+Pythia+**Delphes**, Not real ATLAS

p-value Scan on Test Dataset at μ=0.5

With a standard strategy



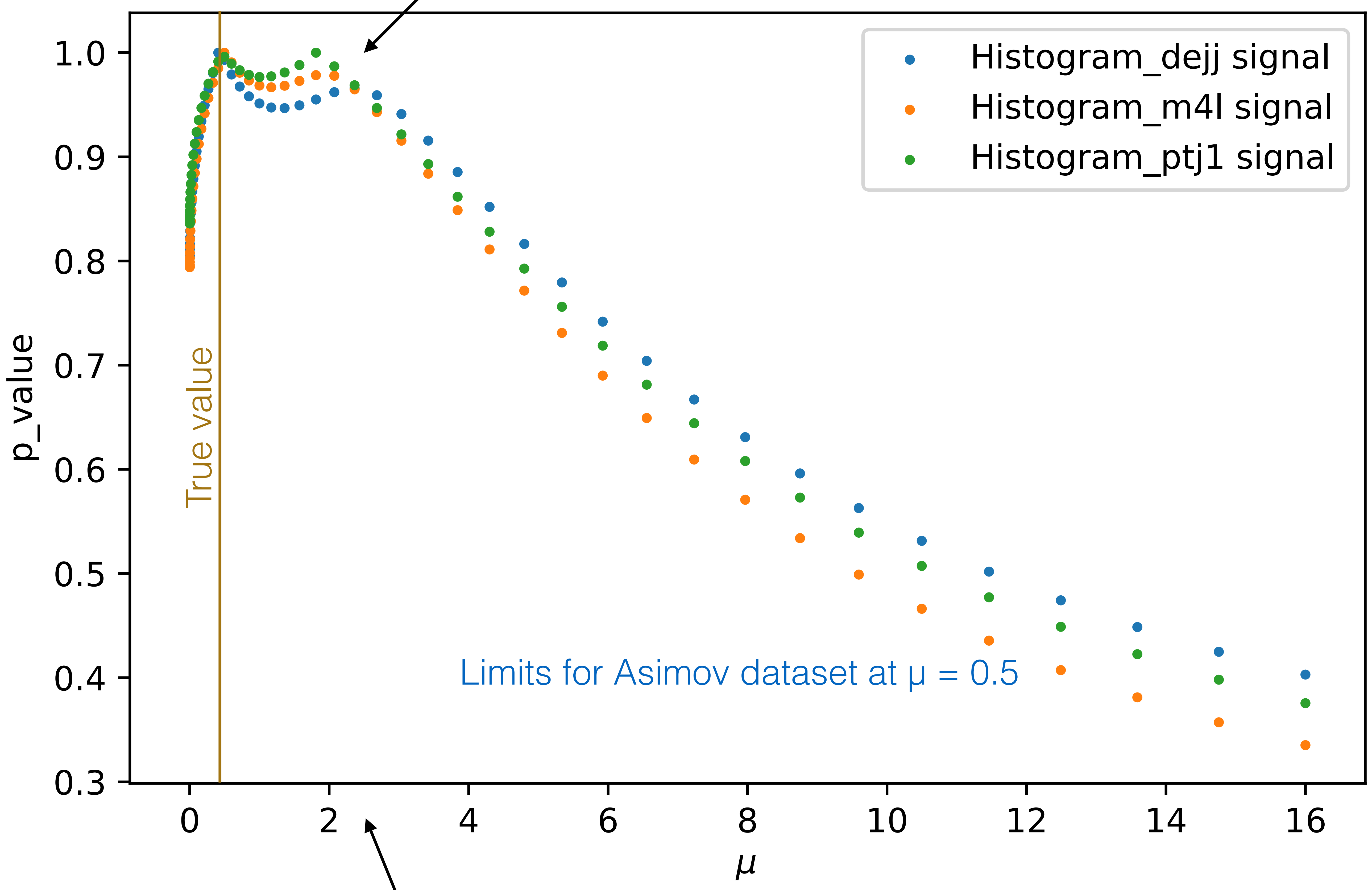
We are asking the question:
What is the p-value for the μ=2.5 (for example) on an Asimov dataset generated with true μ=0.5 ?

Disclaimer: Private simulations with Madgraph+Pythia+**Delphes**, Not real ATLAS

p-value Scan on Test Dataset at μ=0.5

Second peak at different μ

With a standard strategy



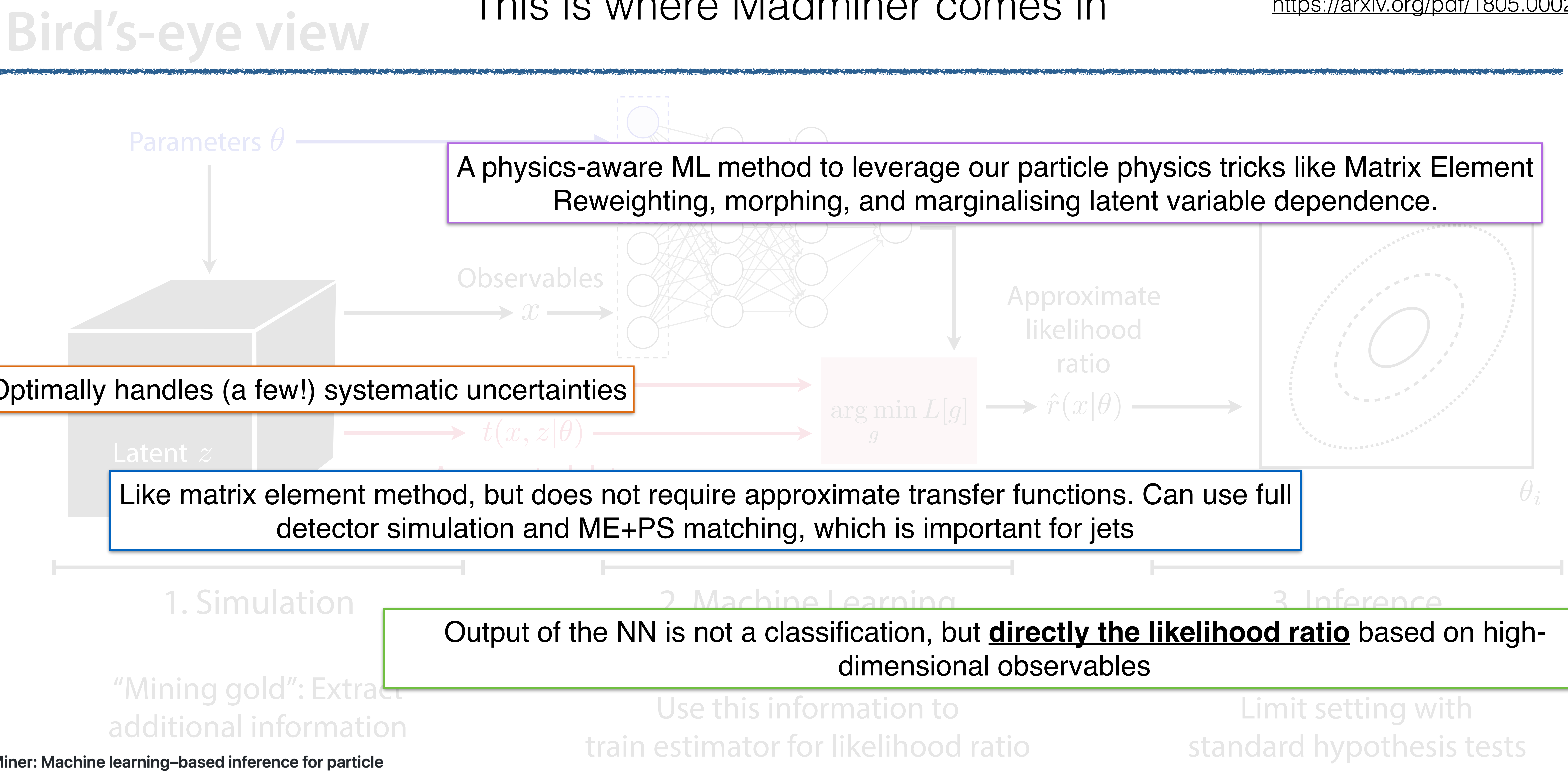
We are asking the question:
What is the p-value for the μ=2.5 (for example) on an Asimov dataset generated with true μ=0.5 ?

Need an inference machinery that is **aware of** how the **physics** changes **away from the Standard Model point**

A Machine Learning method could **re-optimise the analysis for each hypothesis (μ) being tested automatically.**
If we could train such a model without class labels …

This is where Madminer comes in

https://arxiv.org/pdf/1805.00013.pdf
https://arxiv.org/pdf/1805.00020.pdf



A physics-aware ML method to leverage our particle physics tricks like Matrix Element Reweighting, morphing, and marginalising latent variable dependence.

Optimally handles (a few!) systematic uncertainties

Like matrix element method, but does not require approximate transfer functions. Can use full detector simulation and ME+PS matching, which is important for jets

Output of the NN is not a classification, but **directly the likelihood ratio** based on high-dimensional observables

MadMiner: Machine learning–based inference for particle physics

By Johann Brehmer, Felix Kling, Irina Espejo, and Kyle Cranmer

pypi package 0.6.0 | build passing | docs passing | chat on gitter | docker pulls 127 | code style black | License MIT
DOI 10.5281/zenodo.1489147

https://arxiv.org/pdf/1805.00013.pdf

More informative targets to regress for a **neural network**

Dog Pictures Classification:

Should I train my neural network to learn to guess True (1) or False (0) ?

More informative targets to regress for a **neural network**

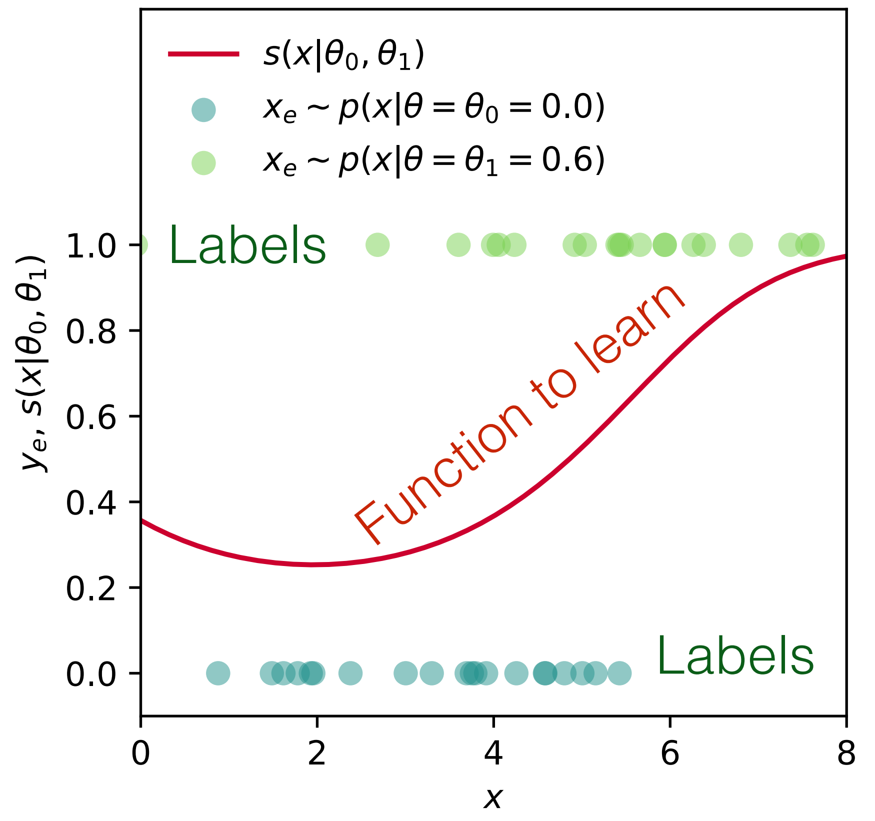
Dog Pictures Classification:

Should I train my neural network to learn to guess True (1) or False (0) ?

I would give this a true class label = 0.7, not 1

More informative targets to regress for a **neural network**

Dog Pictures Classification:

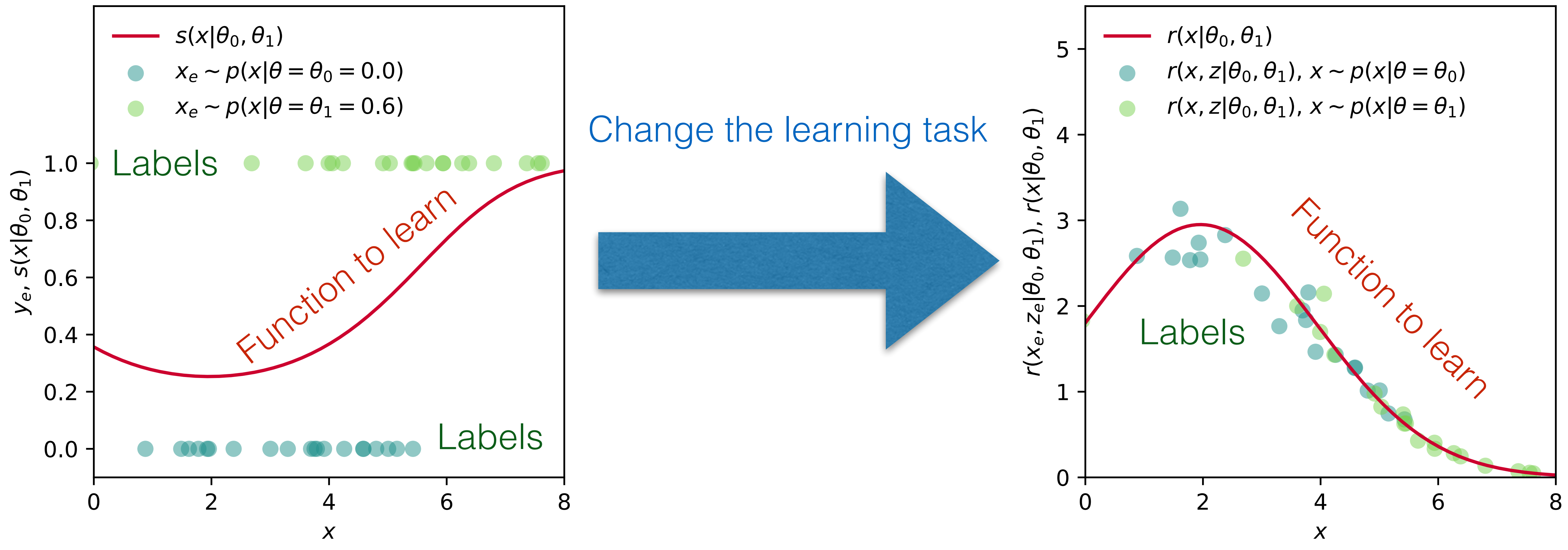


Should I train my neural network to learn to guess True (1) or False (0) ?

I would give this a true class label = 0.7, not 1

More informative targets to regress for a **neural network**

Dog Pictures Classification:



Should I train my neural network to learn to guess True (1) or False (0) ?

I would give this a true class label = 0.7, not 1

https://arxiv.org/pdf/1907.10621.pdf

MadMiner Inference Models

Method	Run simulation at	Loss fn. uses $r(x,z)$	Loss fn. uses $t(x,z)$	Asympt. exact	Generative	Ref.
Likelihood estimators						
NDE	$\theta \sim \pi(\theta)$			✓	✓	[54]
SCANDAL	$\theta \sim \pi(\theta)$		✓	✓	✓	[65]
Likelihood ratio estimators						
CARL	$\theta \sim \pi(\theta)$, θ_{ref}			✓		[39]
ROLR	$\theta \sim \pi(\theta)$, θ_{ref}	✓		✓		[67]
ALICE	$\theta \sim \pi(\theta)$, θ_{ref}	✓		✓		[68]
CASCAL	$\theta \sim \pi(\theta)$, θ_{ref}		✓	✓		[67]
RASCAL	$\theta \sim \pi(\theta)$, θ_{ref}	✓	✓	✓		[67]
ALICES	$\theta \sim \pi(\theta)$, θ_{ref}	✓	✓	✓		[68]
Doubly parameterized likelihood ratio estimators						
CARL	$\theta_0 \sim \pi(\theta)$, $\theta_1 \sim \pi(\theta)$			✓		[39]
ROLR	$\theta_0 \sim \pi(\theta)$, $\theta_1 \sim \pi(\theta)$	✓		✓		[67]
ALICE	$\theta_0 \sim \pi(\theta)$, $\theta_1 \sim \pi(\theta)$	✓		✓		[68]
CASCAL	$\theta_0 \sim \pi(\theta)$, $\theta_1 \sim \pi(\theta)$		✓	✓		[67]
RASCAL	$\theta_0 \sim \pi(\theta)$, $\theta_1 \sim \pi(\theta)$	✓	✓	✓		[67]
ALICES	$\theta_0 \sim \pi(\theta)$, $\theta_1 \sim \pi(\theta)$	✓	✓	✓		[68]
Score estimators						
SALLY	θ_{ref}		✓	in local approx.		[67]
SALLINO	θ_{ref}		✓	in local approx.		[67]

Approximate **L**ikelihood with **I**mproved **C**ross-entropy **E**stimator and **S**core

(parameterised on all values of μ)

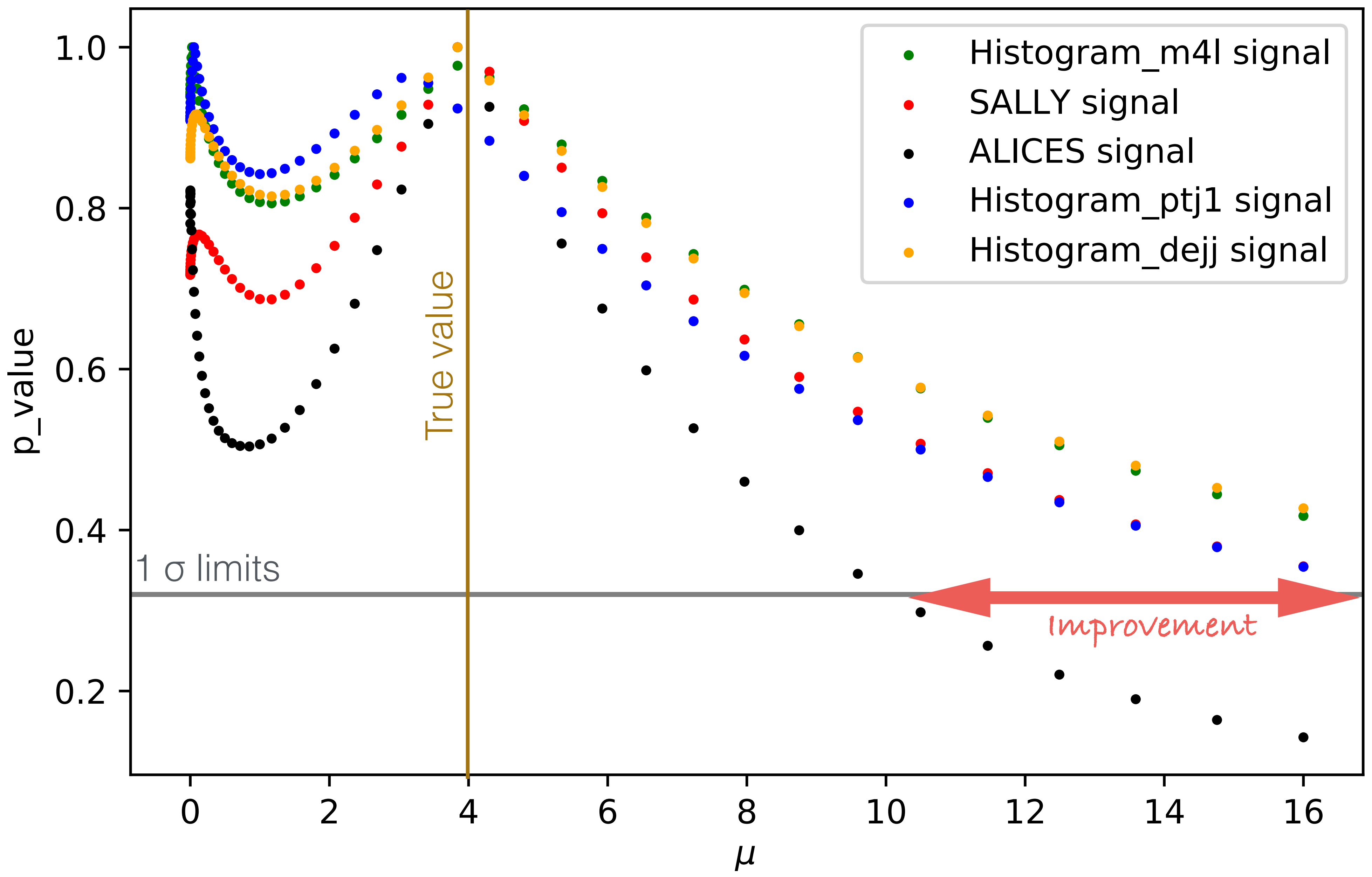
Score **A**pproximates **L**ikelihood **L**ocall**y**

(locally optimal near the SM)

Table I. Inference techniques implemented in MadMiner. We separate them into four groups, depending on which quantity is estimated by the neural network; see the text for more details. We list for parameter values the Monte-Carlo samples have to be generated and whether the augmented data (joint likelihood ratio $r(x,z)$ and joint likelihood ratio $t(x,z)$) is used. "Asymptotically exact" quantifies whether a method should give theoretically optimal results in the limit of sufficient network capacity, perfect optimization, and enough training data. Some network architectures also allow for fast generation of event data directly from the neural network, they are marked as "generative". Finally, for each method we provide the reference that provides the clearest explanation (and spells out the acronym).

Preliminary: Expected Limits with “Alices” and “Sally”

Very preliminary work, qqbarZZ background, gg(H)zz signal not taken into account, only the VBF + VBS interference is studied



Limits for Asimov dataset at μ = 4

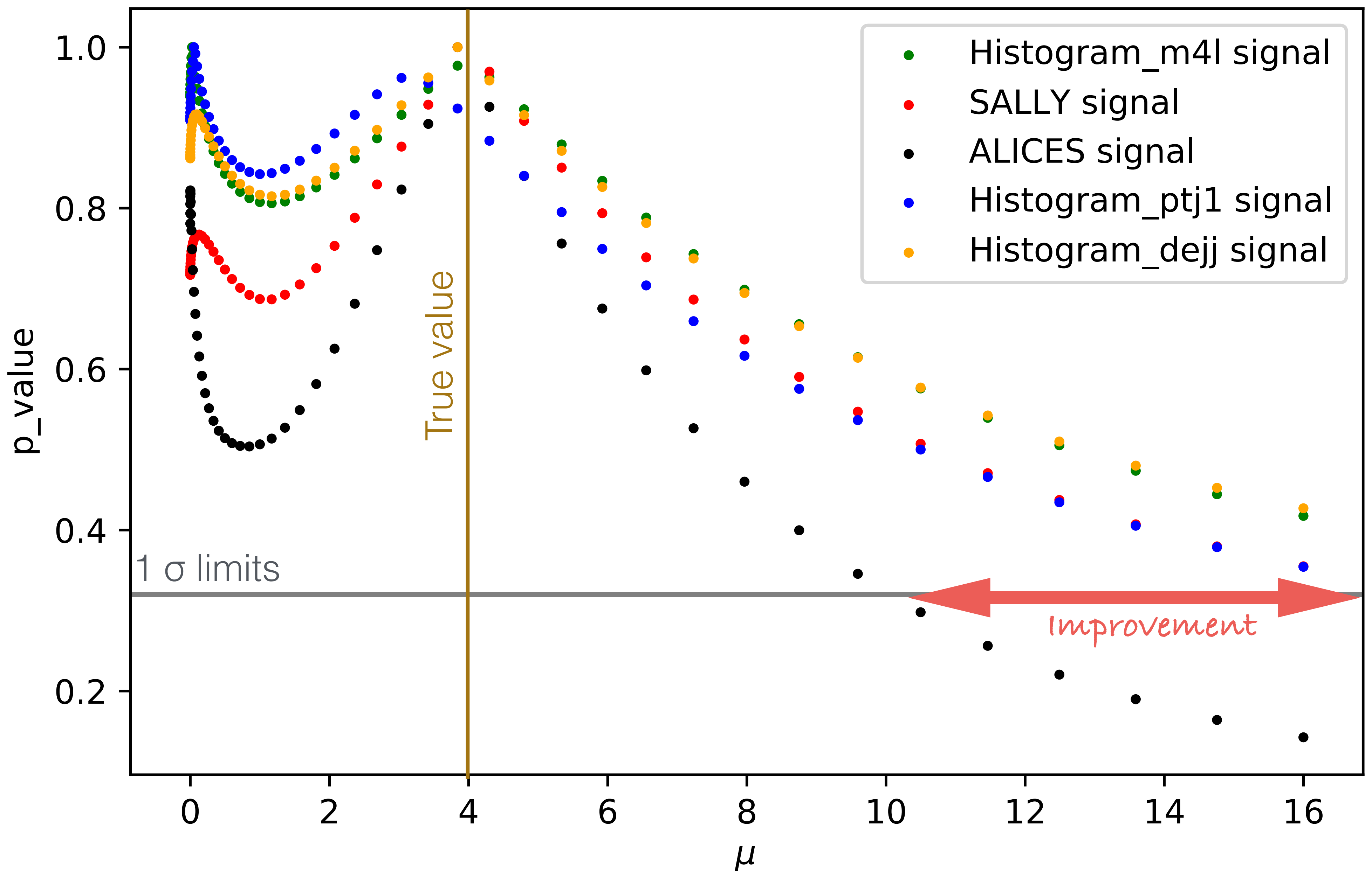
lumi= 36 ifb,
Preselection nJets>=2
m4l> 220 GeV
Only kinematics used,
not total cross section

SALLY (Score Approximates Likelihood Locally): Locally **optimal near the SM**. Requires fewer training samples.

ALICES (Approximate Likelihood with Improved Cross-entropy Estimator and Score): More powerful over a **large range of μ**

Preliminary: Expected Limits with “Alices” and “Sally”

Very preliminary work, qqbarZZ background, gg(H)zz signal not taken into account, only the VBF + VBS interference is studied



Limits for Asimov dataset at μ = 4

Alices is better at breaking the degeneracy because it’s a parameterised model

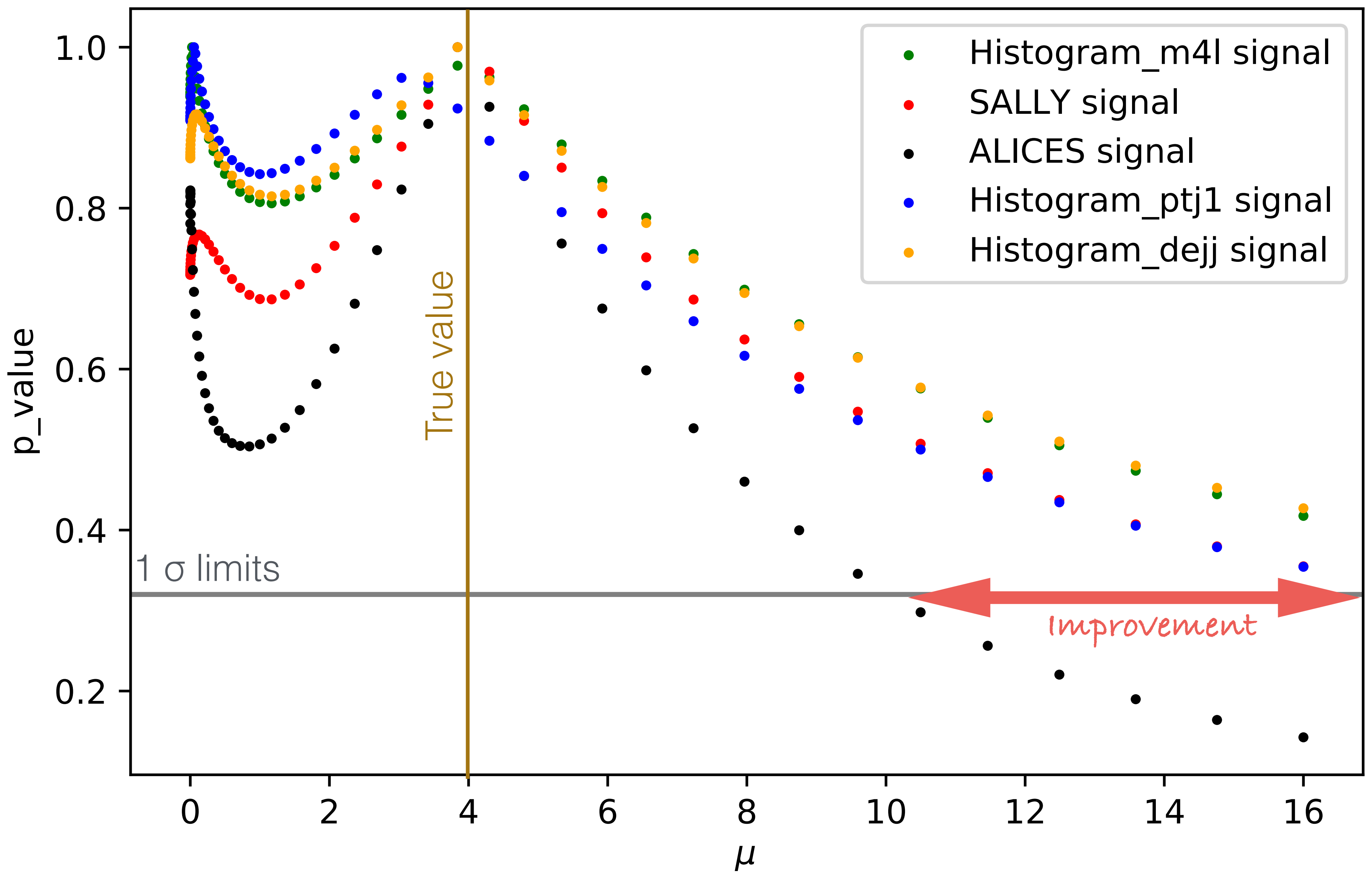
lumi= 36 ifb,
Preselection nJets>=2
m4l> 220 GeV
Only kinematics used, not total cross section

SALLY (Score Approximates Likelihood Locally): Locally **optimal near the SM**. Requires fewer training samples.

ALICES (Approximate Likelihood with Improved Cross-entropy Estimator and Score): More powerful over a **large range of μ**

Preliminary: Expected Limits with “Alices” and “Sally”

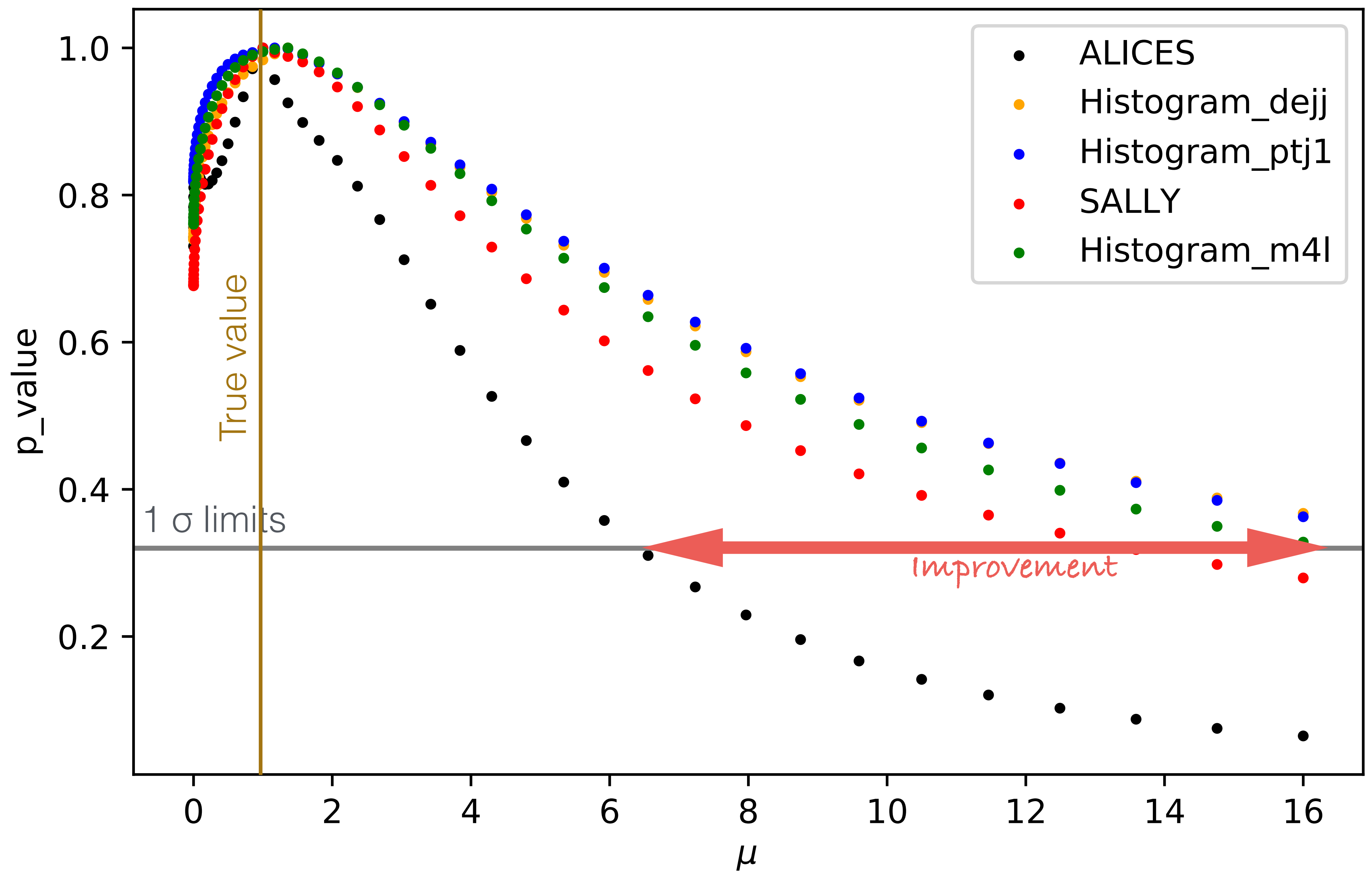
Very preliminary work, qqbarZZ background, gg(H)zz signal not taken into account, only the VBF + VBS interference is studied



Limits for Asimov dataset at μ = 4

Alices is better at breaking the degeneracy because it’s a parameterised model

lumi= 36 ifb,
Preselection nJets>=2
m4l> 220 GeV
Only kinematics used, not total cross section



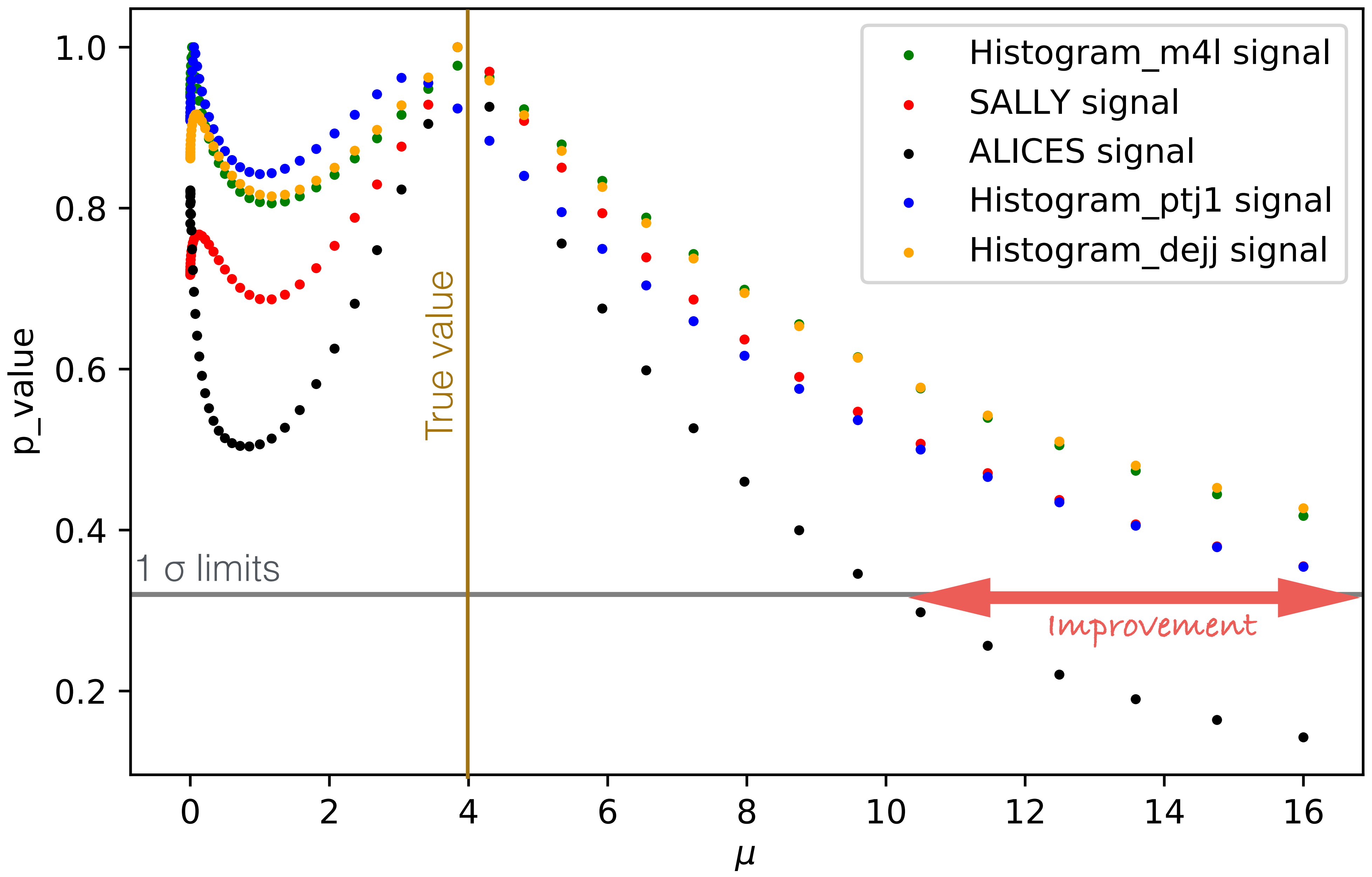
Limits for Asimov dataset at μ = 1 (SM)

SALLY (Score Approximates Likelihood Locally): Locally **optimal near the SM**. Requires fewer training samples.

ALICES (Approximate Likelihood with Improved Cross-entropy Estimator and Score): More powerful over a **large range of μ**

Preliminary: Expected Limits with “Alices” and “Sally”

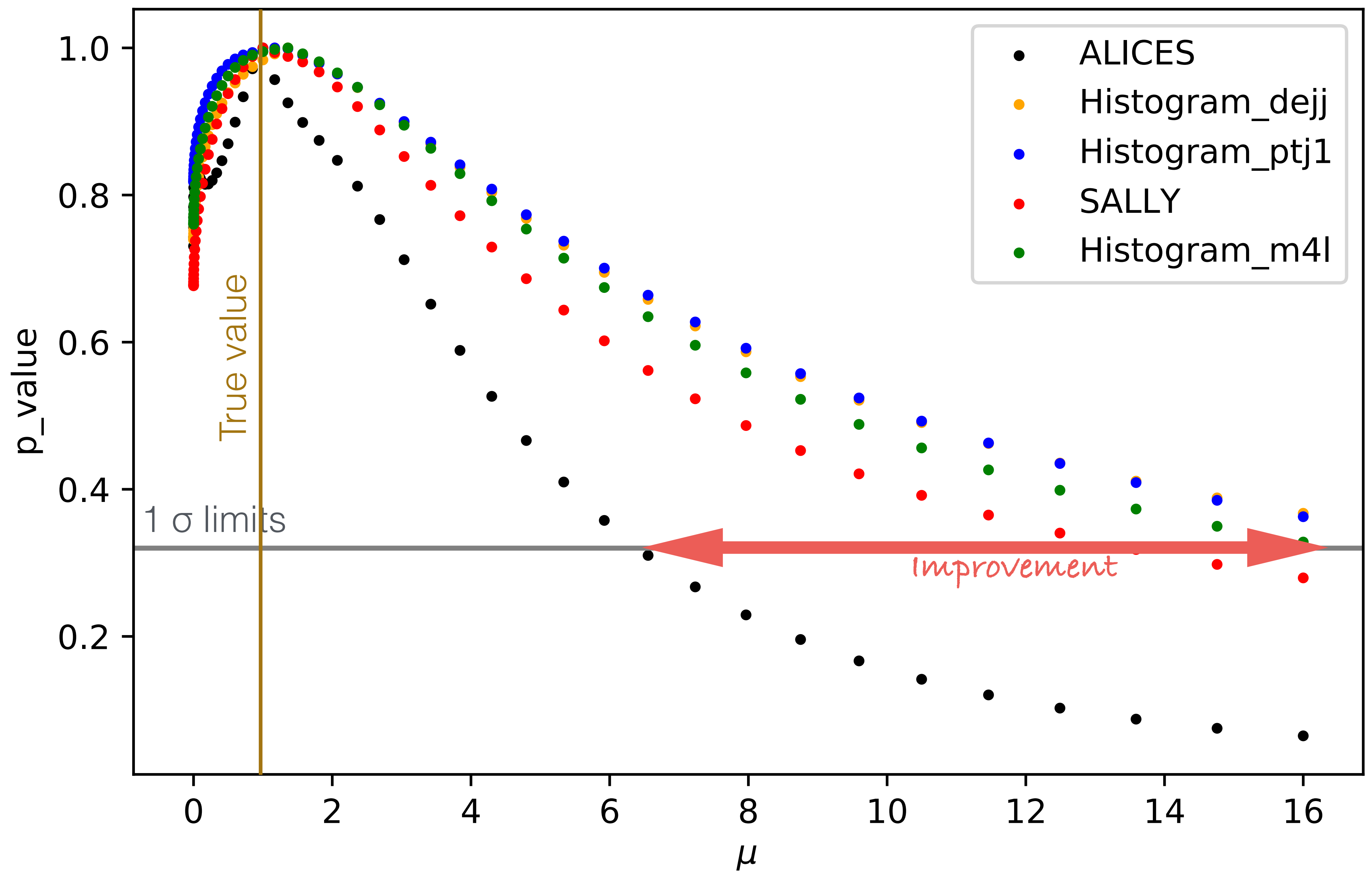
Very preliminary work, qqbarZZ background, gg(H)zz signal not taken into account, only the VBF + VBS interference is studied



Limits for Asimov dataset at μ = 4

Alices is better at breaking the degeneracy because it’s a parameterised model

lumi= 36 ifb,
Preselection nJets>=2
m4l> 220 GeV
Only kinematics used, not total cross section



Limits for Asimov dataset at μ = 1 (SM)

Alices >> Sally > Histograms

Actual ATLAS VBF offshell h4l baseline analysis would be better than a simple histogram of m4l

SALLY (Score Approximates Likelihood Locally): Locally **optimal near the SM**. Requires fewer training samples.

ALICES (Approximate Likelihood with Improved Cross-entropy Estimator and Score): More powerful over a **large range of μ**

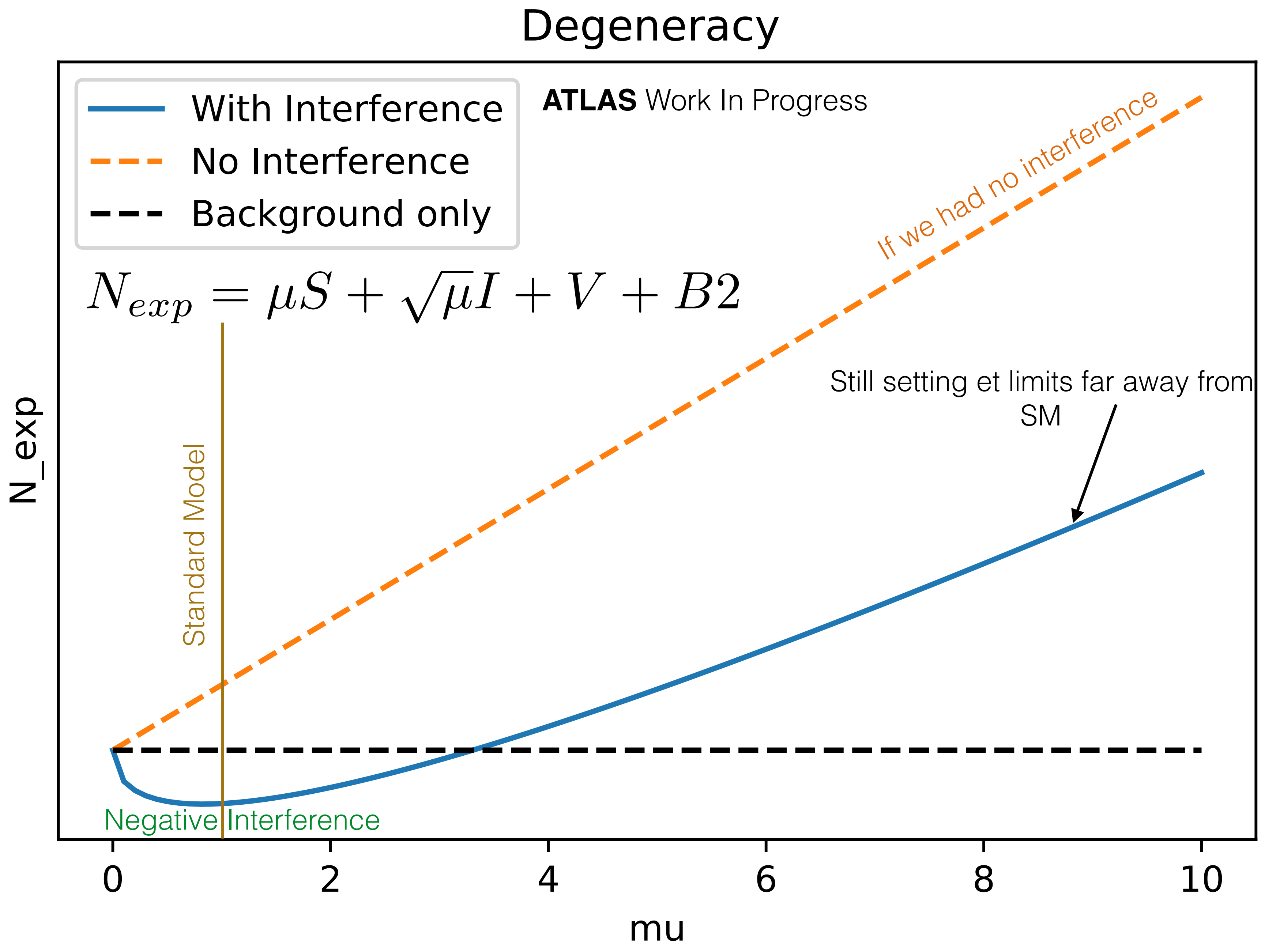
Conclusion

- Aim: Measure Higgs signal strength in the presence of **Interference** ⇒
  - ML without class labels: Directly **learn the likelihoods**
  - ML parameterised on μ: extra information of **physics away from SM**
- Madminer bonus:

Machine Learning that handles systematics better ??

  - Handle a few(!) systematic uncertainties
  - Avoid expensive Matrix Element based discriminant calculations

- Preliminary results promising, **need to add qqbar background and gg(H)zz signal**
- **Requires some work for ATLAS integration**



Many many thanks to Johann Brehmer, Antoine Laudrain, Samyukta Krishnamurthy, Martina Javurkova for the help

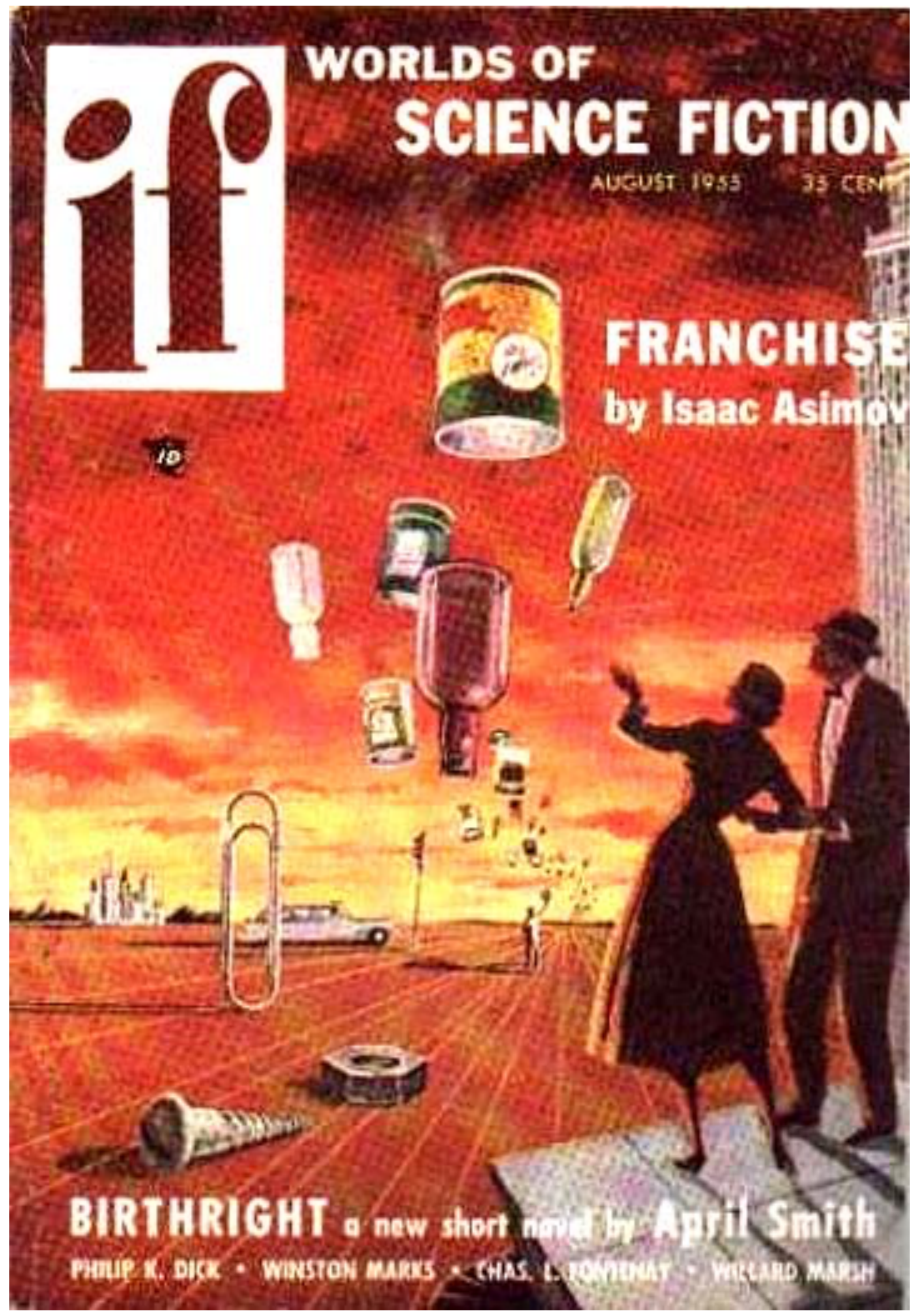
Backup

https://arxiv.org/abs/1007.1727

“Asimov Dataset”

- To get a median expected measurement result of an observation where we expect few events, need to generate many “toy” observation datasets
- Replacing the ensemble of simulated experiments (toys) by a single representative one, the “Asimov” dataset
- A dataset upon which unbiased measurements yield exactly the correct theory parameters
- In practice we cannot have perfectly Asimov datasets, but a very large simulation can approximate an Asimov dataset
- Statistical uncertainties are not quadratic sum of weights, they are $\sqrt{(N)}$ where $N = \Sigma(W)$, to give a feeling of realistic expected uncertainties on the actual observed data

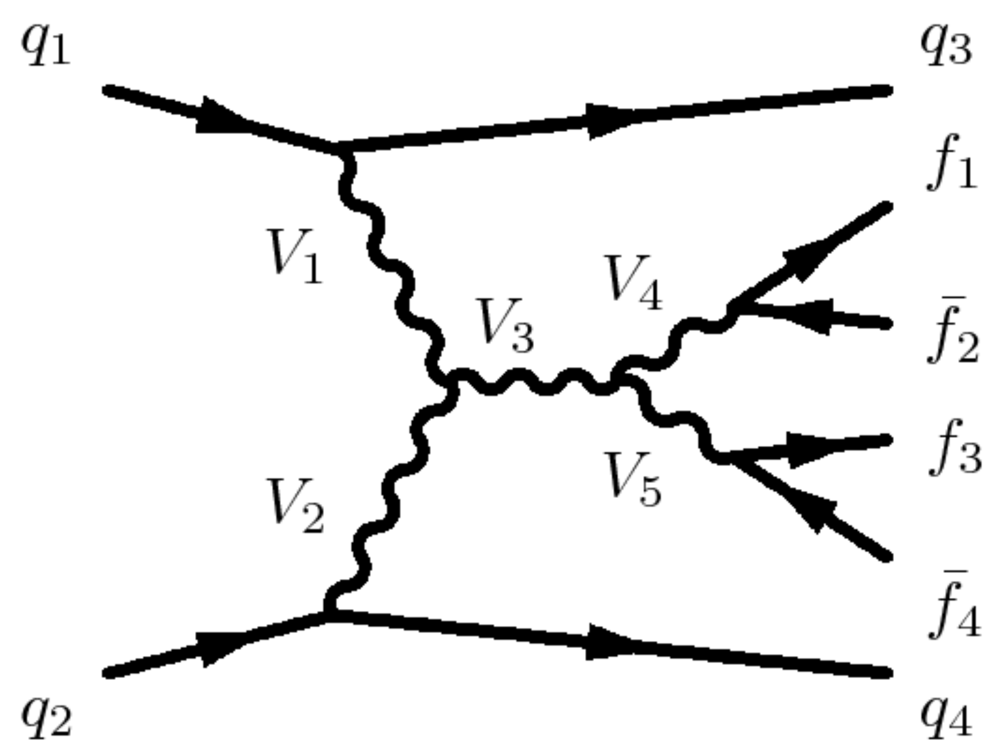
In the future, the United States has converted to an "electronic democracy" where the **AI** selects a **single person to answer a number of questions**. The **AI** will then use the answers and other data to determine what the results of an election would be, **avoiding the need for an actual election** to be held.



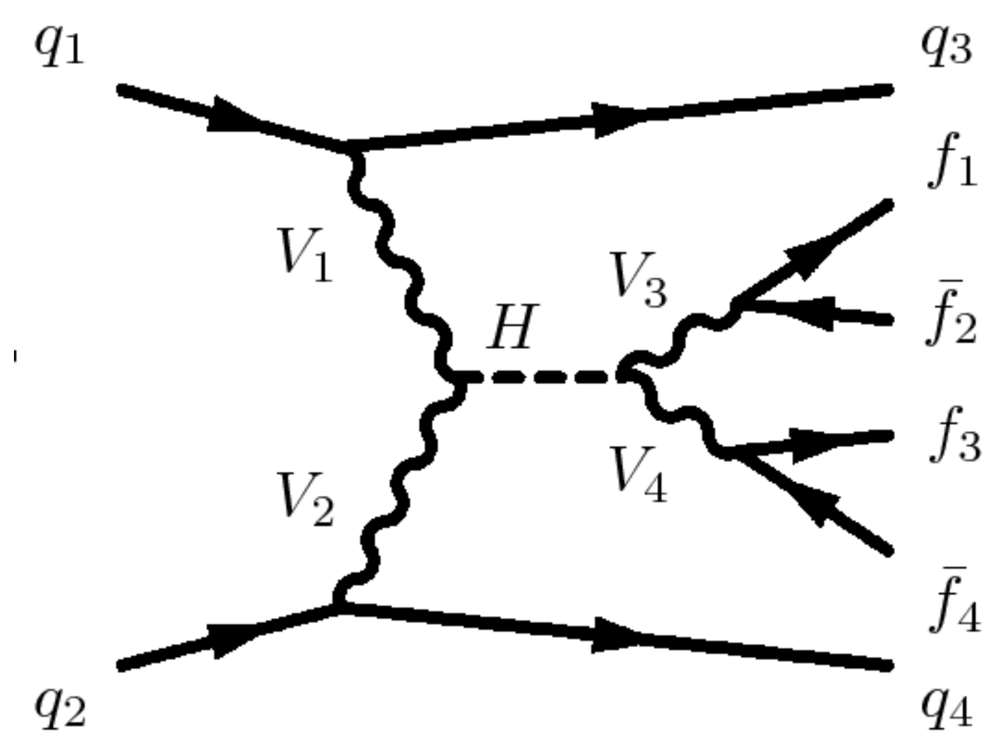
https://en.wikipedia.org/wiki/Franchise_(short_story)

Interference Kills "Signal like" background events

S = VBF-Higgs, B = VBS, SBI = Combined Simulation

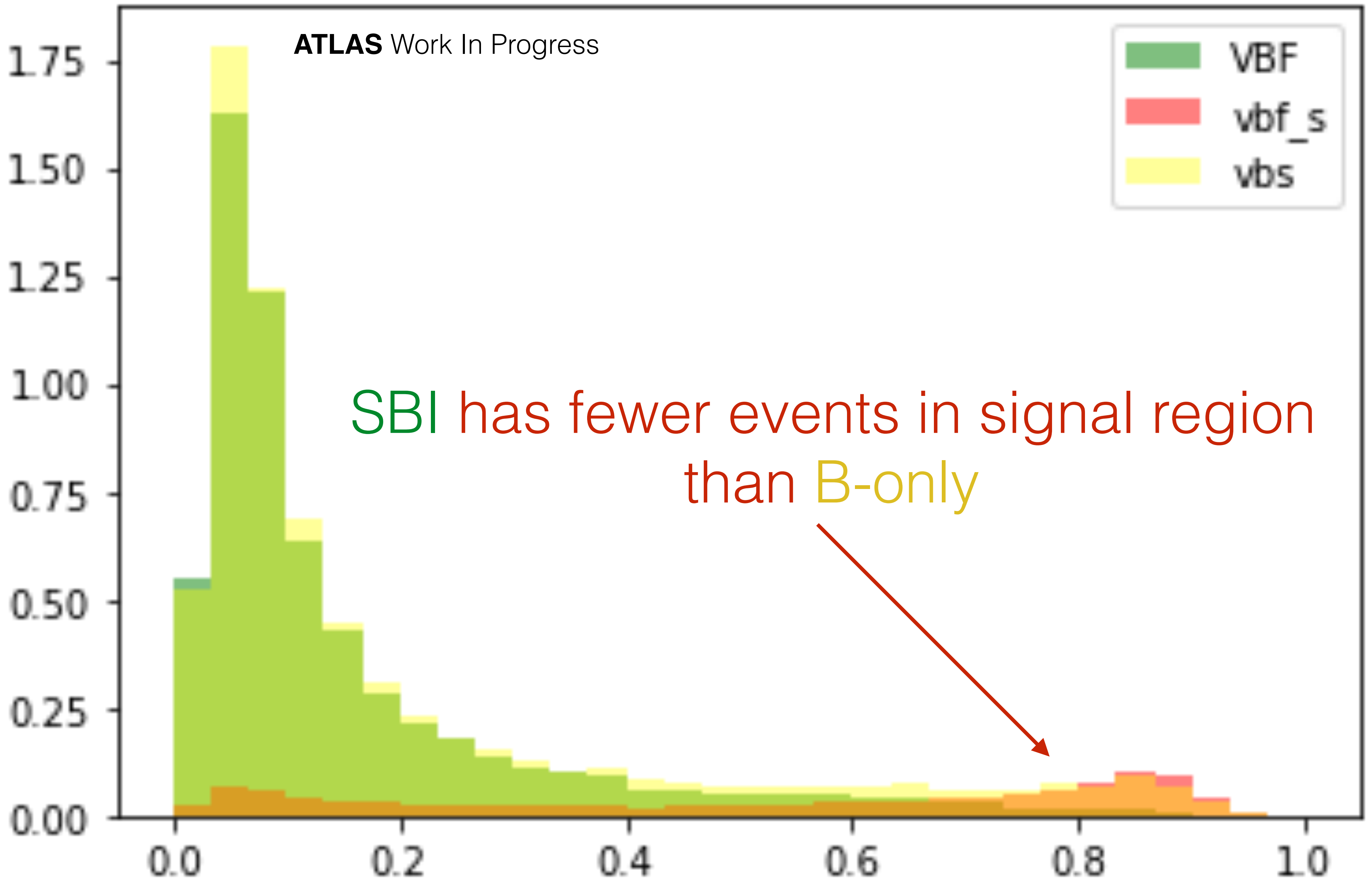


Vector Boson Propagator (Background)



Higgs Propagator (Signal)

Classifier trained on S vs B



- The background distribution also peaks at the "signal region", BDT_score > 0.8
  - Events so similar to signal that we would have interference
- The <u>SBI combined simulation does not peak at the "signal region"</u> BDT_score> 0.8
  - **Interference is almost perfectly destructive**

Derive New Metric

$$Z = \frac{s}{\sqrt{b}}$$

Analogous version with interference →

$$¡Z = \frac{S + SVI - V}{2\sqrt{SVI + B2}}$$

S = VBF-Higgs, V= VBS, SVI = Combined Simulation, B2 = gg(H)ZZ + qqZZ

See derivation and asymptotic version here

What is the metric to optimise? No longer (s/√b)

μ = signal strength

$$N_{exp} = \mu S + \sqrt{\mu} I + V + B2$$

S = VBF-Higgs, V= VBS, SVI = Combined Simulation, B2 = gg(H)ZZ + qqZZ

$$I = SVI - S - V$$

$$B = V + B2$$

$$L = \text{Poisson}(\mu S + \sqrt{\mu} I + B, N) = \frac{(\mu S + \sqrt{\mu} I + B)^N}{N!} e^{-(\mu S + \sqrt{\mu} I + B)}$$

$$\sigma_\mu = \frac{\sqrt{S + I + B}}{S + \frac{I}{2}}$$

$$¡Z = \frac{S + SVI - V}{2\sqrt{SVI + B2}}$$

Maximise this [analogue to (s/√b) in the usual scenario]

Problem: Not trustworthy at low statistics, what about asymptotic formula?

Assuming we are trying to reject μ=0, (which is not the case anymore)

What is the metric to optimise? No longer (s/√b)

μ = signal strength

$$N_{exp} = \mu S + \sqrt{\mu} I + V + B2$$

S = VBF-Higgs, V= VBS, SVI = Combined Simulation, B2 = gg(H)ZZ + qqZZ

$$I = SVI - S - V$$

$$B = V + B2$$

$$\text{L} = \text{Poisson}(\mu S + \sqrt{\mu} I + B, N) = \frac{(\mu S + \sqrt{\mu} I + B)^N}{N!} e^{-(\mu S + \sqrt{\mu} I + B)}$$

$$\sigma_\mu = \frac{\sqrt{S + I + B}}{S + \frac{I}{2}}$$

$$¡Z = \frac{S + SVI - V}{2\sqrt{SVI + B2}}$$

Maximise this [analogue to (s/√b) in the usual scenario]

Problem: Not trustworthy at low statistics, what about asymptotic formula?

$$Z_0 = \sqrt{2\left[(SVI + B2)\ln\left(1 + \frac{SVI - V}{V + B2}\right) - (SVI - V)\right]}$$

Assuming we are trying to reject μ=0, (which is not the case anymore)

Analogue to the AMS Asimov formula by Cowan, Cranmer, Gross, Vitells

What is the metric to optimise? No longer (s/√b)

μ = signal strength

$$N_{exp} = \mu S + \sqrt{\mu} I + V + B2$$

S = VBF-Higgs, V= VBS, SVI = Combined Simulation, B2 = gg(H)ZZ + qqZZ

$$I = SVI - S - V$$

$$B = V + B2$$

$$L = \text{Poisson}(\mu S + \sqrt{\mu} I + B, N) = \frac{(\mu S + \sqrt{\mu} I + B)^N}{N!} e^{-(\mu S + \sqrt{\mu} I + B)}$$

$$\sigma_\mu = \frac{\sqrt{S + I + B}}{S + \frac{I}{2}}$$

¡$$Z = \frac{S + SVI - V}{2\sqrt{SVI + B2}}$$

← Maximise this [analogue to (s/√b) in the usual scenario]

Problem: Not trustworthy at low statistics, what about asymptotic formula?

$$Z_0 = \sqrt{2\left[(SVI + B2)\ln\left(1 + \frac{SVI - V}{V + B2}\right) - (SVI - V)\right]}$$

Assuming we are trying to reject μ=0, (which is not the case anymore)

← Analogue to the AMS Asimov formula by Cowan, Cranmer, Gross, Vitells

Interestingly, "S" dropped out of the formula completely

What is the metric to optimise? No longer (s/√b)

μ = signal strength

$$N_{exp} = \mu S + \sqrt{\mu} I + V + B2$$

S = VBF-Higgs, V= VBS, SVI = Combined Simulation, B2 = gg(H)ZZ + qqZZ

$$I = SVI - S - V$$

$$B = V + B2$$

$$\text{L} = \text{Poisson}(\mu S + \sqrt{\mu} I + B, N) = \frac{(\mu S + \sqrt{\mu} I + B)^N}{N!} e^{-(\mu S + \sqrt{\mu} I + B)}$$

$$\sigma_\mu = \frac{\sqrt{S + I + B}}{S + \frac{I}{2}}$$

$$¡Z = \frac{S + SVI - V}{2\sqrt{SVI + B2}}$$

Maximise this [analogue to (s/√b) in the usual scenario]

Problem: Not trustworthy at low statistics, what about asymptotic formula?

$$Z_0 = \sqrt{2\left[(SVI + B2)\ln\left(1 + \frac{SVI - V}{V + B2}\right) - (SVI - V)\right]}$$

Assuming we are trying to reject μ=0, (which is not the case anymore)

Analogue to the AMS Asimov formula by Cowan, Cranmer, Gross, Vitells

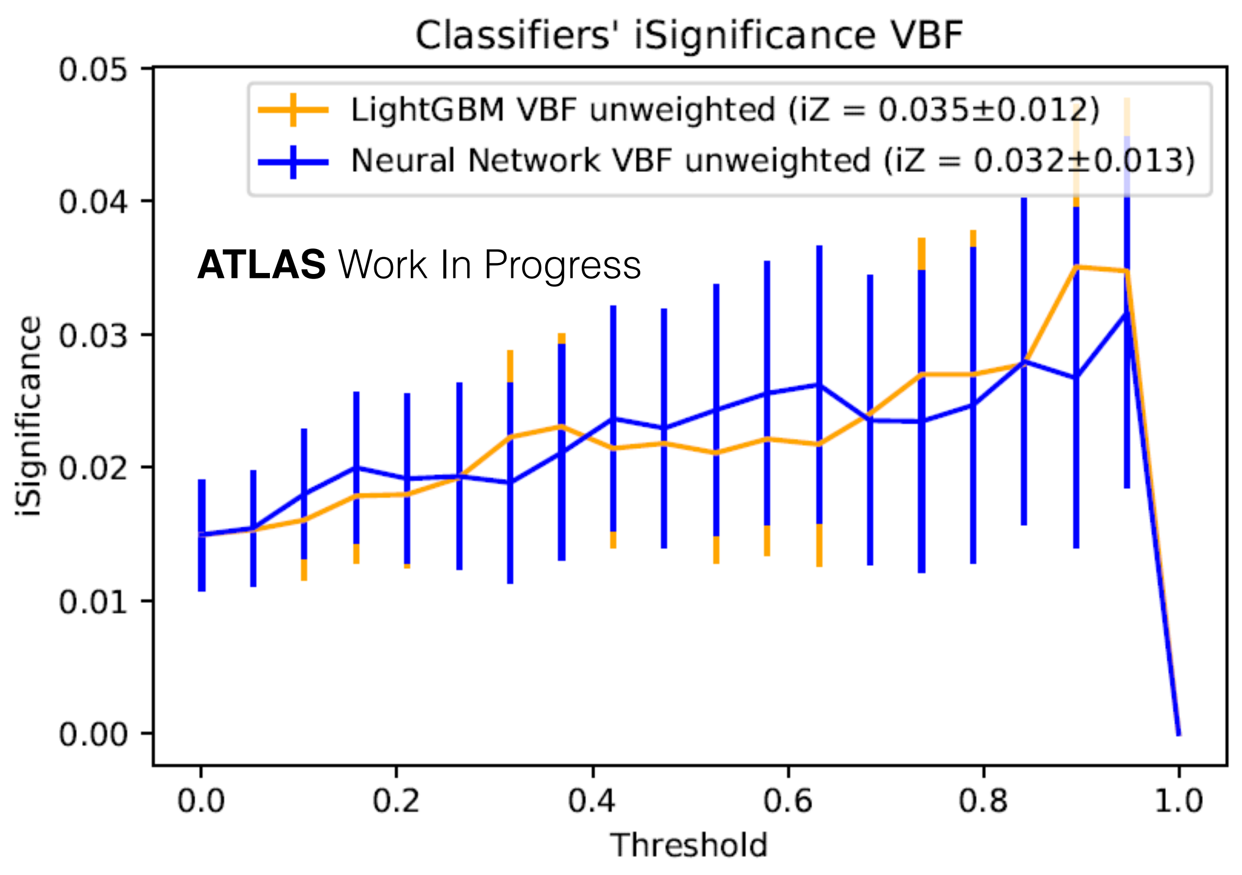
for I=0 →

$$\sqrt{2\left((s + b)\ln(1 + s/b) - s\right)}$$

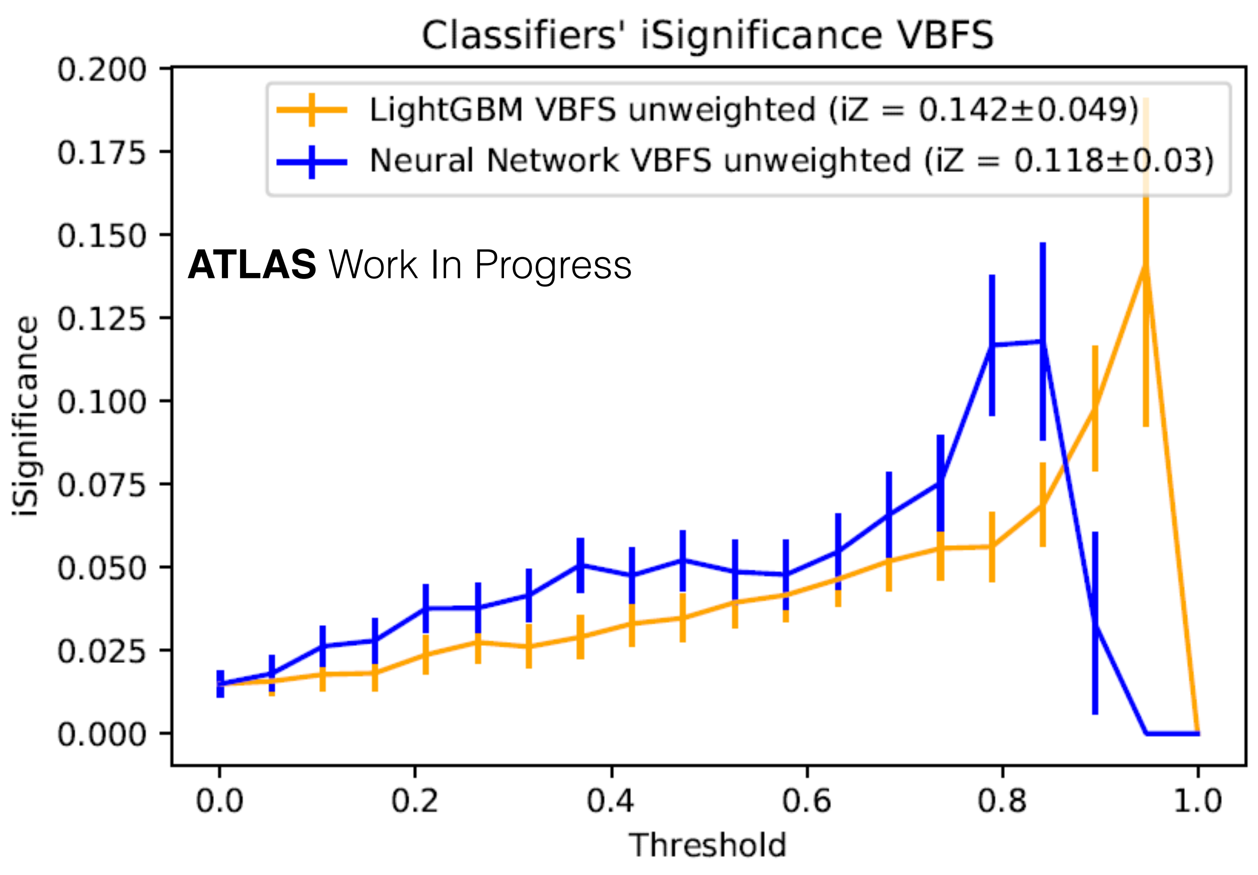
Usual AMS formula

Interestingly, “S” dropped out of the formula completely

Find the Best Classification Task to Improve μ Sensitivity



<u>iZ for classifier trained to separate:</u>
<u>VBF_SVI vs qqbar + gg(H)zz</u>



<u>iZ for classifier trained to separate:</u>
<u>VBF_Higgs vs VBS + qqbar + gg(H)zz</u>

Significance with interference:

$$iZ = \frac{S + SBI - B}{2 * \sqrt{SBI + B\_gg\_qq}}$$

S: VBF_s, SBI:VBF
B: VBS, B_gg_qq alias B2: gg+qq

$$\frac{\Delta iZ}{iZ} = \left|\frac{\Delta S + \Delta B}{S + SBI - B}\right| + \left|\frac{\Delta B2}{2 \times (SBI + B2)}\right| + \Delta SBI \left|\frac{1}{S + SBI - B} - \frac{1}{2 \times (SBI + B2)}\right|$$

Second approach is better for iZ consistently,
but is this the best we can do?

Likelihood Ratio Trick no longer applicable to guarantee optimality
Can we go beyond classification?

More informative targets to regress

Dog Pictures Classification:

More informative targets to regress

Dog Pictures Classification:

I would give this a true class label = 0.7, not 1

More informative targets to regress

Dog Pictures Classification:

I would give this a true class label = 0.7, not 1

"joint likelihood ratio" is very closely connected to matrix elements

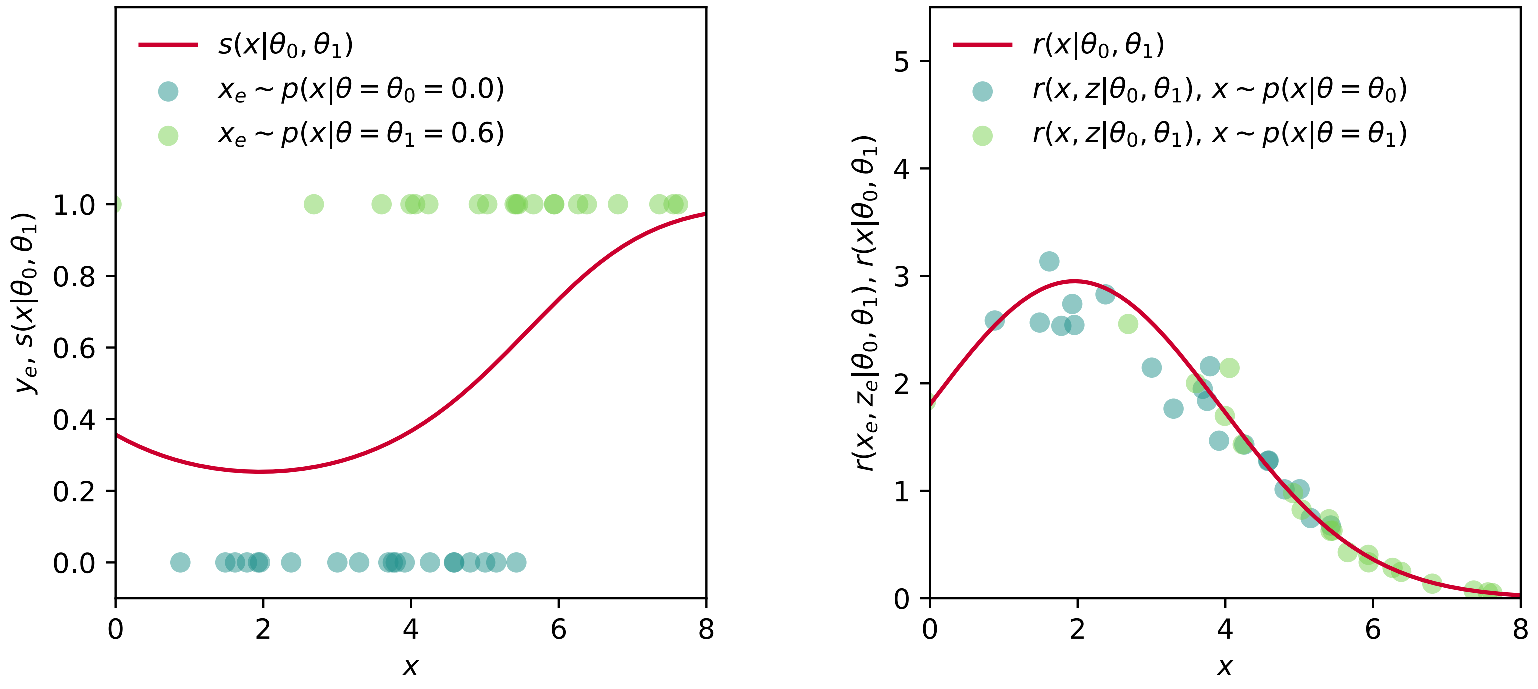


Figure 5: Illustration of some key concepts with a one-dimensional Gaussian toy example. Left: classifiers trained to distinguish two sets of events generated from different hypotheses (green dots) converge to an optimal decision function $s(x|\theta_0, \theta_1)$ (in red) given in Eq. (17). This lets us extract the likelihood ratio. Right: regression on the joint likelihood ratios $r(x_e, z_e|\theta_0, \theta_1)$ of the simulated events (green dots) converges to the likelihood ratio $r(x|\theta_0, \theta_1)$ (red line).

More informative targets to regress

Dog Pictures Classification:

I would give this a true class label = 0.7, not 1

"joint likelihood ratio" is very closely connected to matrix elements

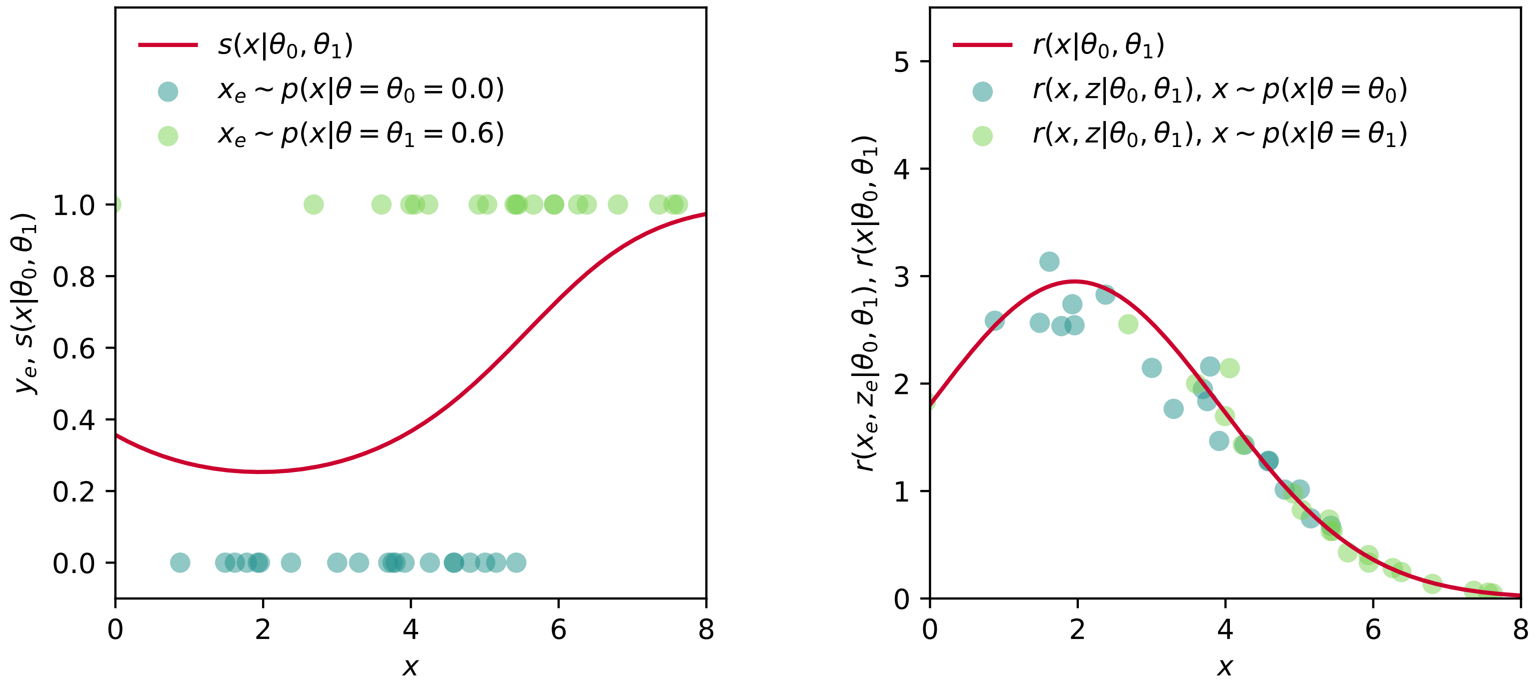


Figure 5: Illustration of some key concepts with a one-dimensional Gaussian toy example. Left: classifiers trained to distinguish two sets of events generated from different hypotheses (green dots) converge to an optimal decision function $s(x|\theta_0, \theta_1)$ (in red) given in Eq. (17). This lets us extract the likelihood ratio. Right: regression on the joint likelihood ratios $r(x_e, z_e|\theta_0, \theta_1)$ of the simulated events (green dots) converges to the likelihood ratio $r(x|\theta_0, \theta_1)$ (red line).

An almost unique privilege of Particle Physics: we can do better than just give the true **class label as a target**. We can set the more informative joint likelihood ratio as the target helping the model converge to the **likelihood ratio** itself.
Possibility to add the score (t) as an auxiliary task.

More informative targets to regress

Dog Pictures Classification:

I would give this a true class label = 0.7, not 1

"joint likelihood ratio" is very closely connected to matrix elements

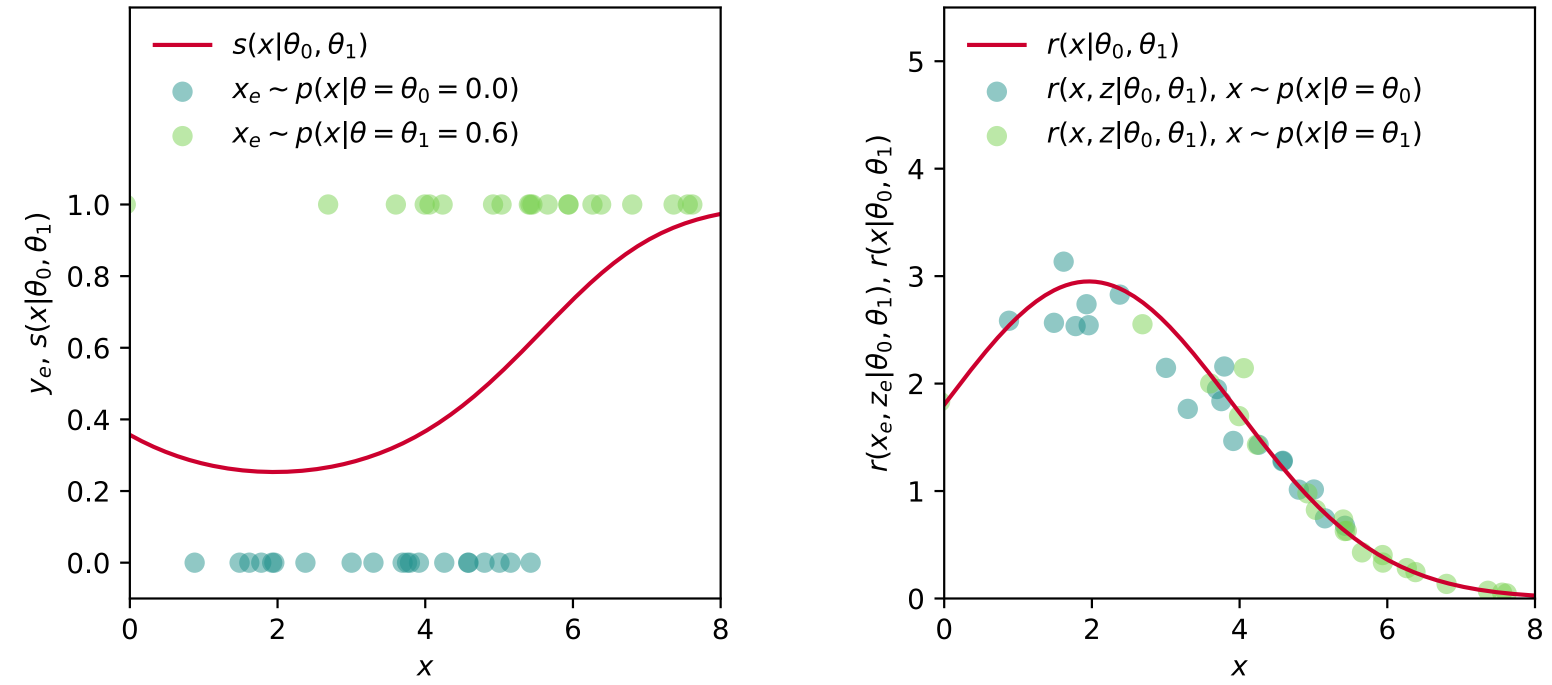


Figure 5: Illustration of some key concepts with a one-dimensional Gaussian toy example. Left: classifiers trained to distinguish two sets of events generated from different hypotheses (green dots) converge to an optimal decision function $s(x|\theta_0, \theta_1)$ (in red) given in Eq. (17). This lets us extract the likelihood ratio. Right: regression on the joint likelihood ratios $r(x_e, z_e|\theta_0, \theta_1)$ of the simulated events (green dots) converges to the likelihood ratio $r(x|\theta_0, \theta_1)$ (red line).

Gradient information

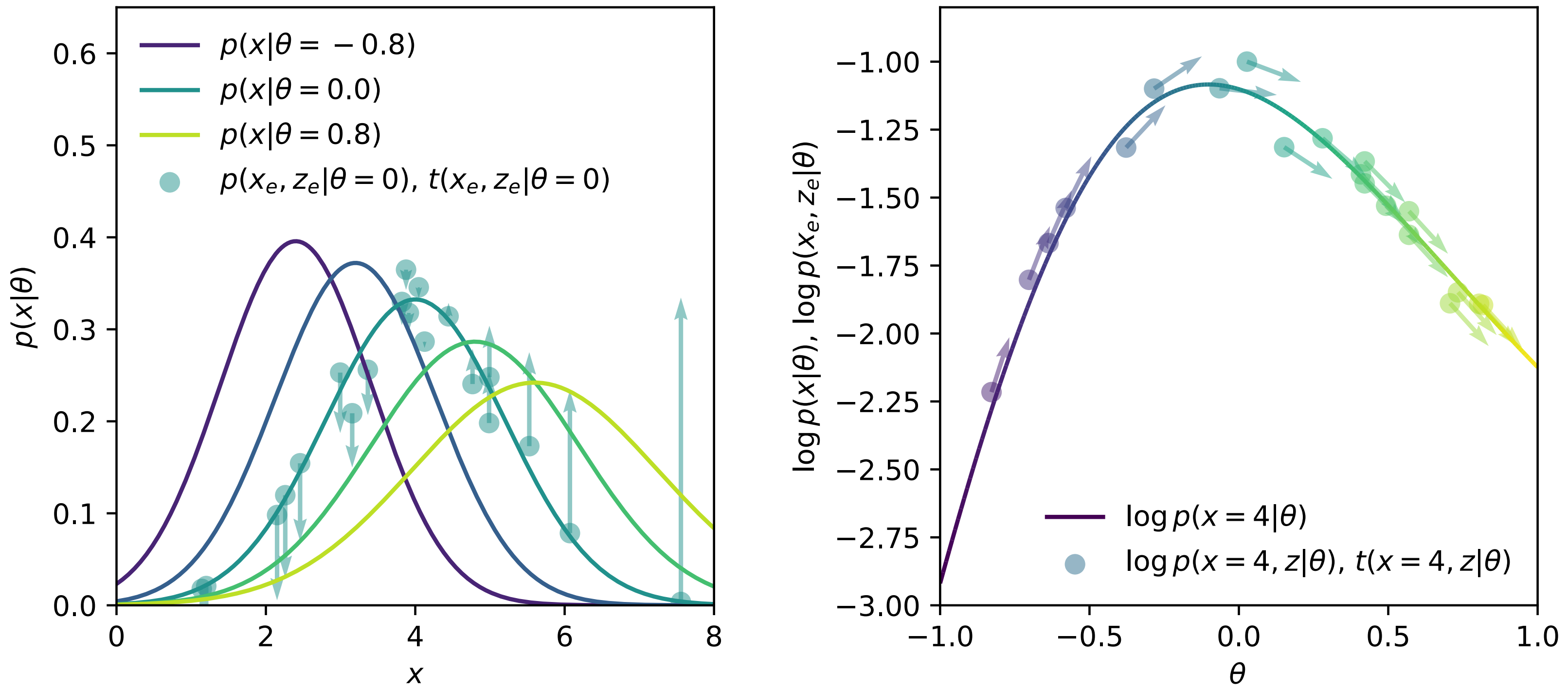


Figure 6: Illustration of some key concepts with a one-dimensional Gaussian toy example. Left: probability density functions for different values of θ and the scores $t(x_e, z_e|\theta)$ at generated events (x_e, z_e). These tangent vectors measure the relative change of the density under infinitesimal changes of θ. Right: dependence of $\log p(x|\theta)$ on θ for fixed $x = 4$. The arrows again show the (tractable) scores $t(x_e, z_e|\theta)$.

An almost unique privilege of Particle Physics: we can do better than just give the true **class label as a target**. We can set the more informative joint likelihood ratio as the target helping the model converge to the **likelihood ratio** itself.
Possibility to add the score (t) as an auxiliary task.

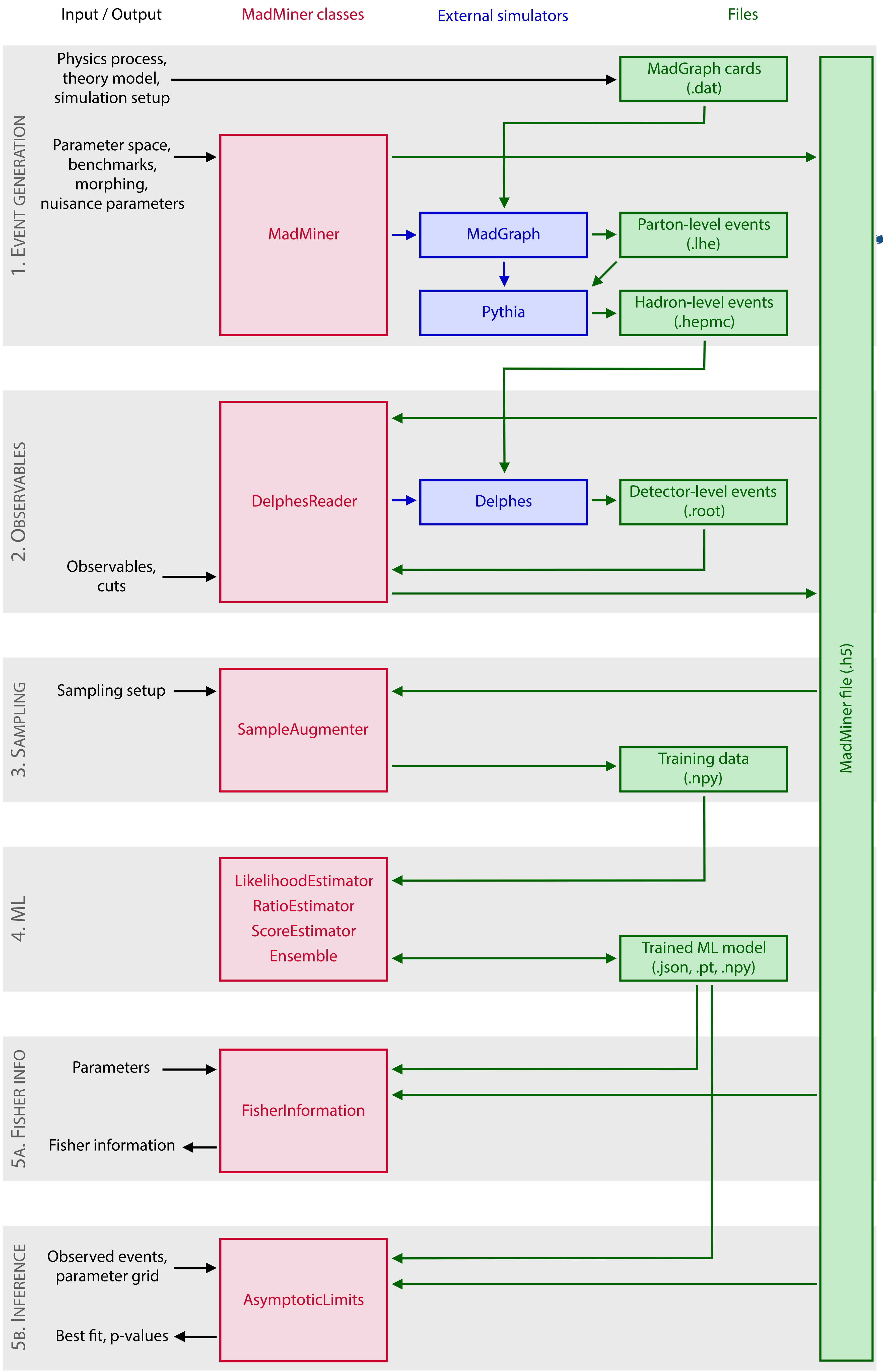


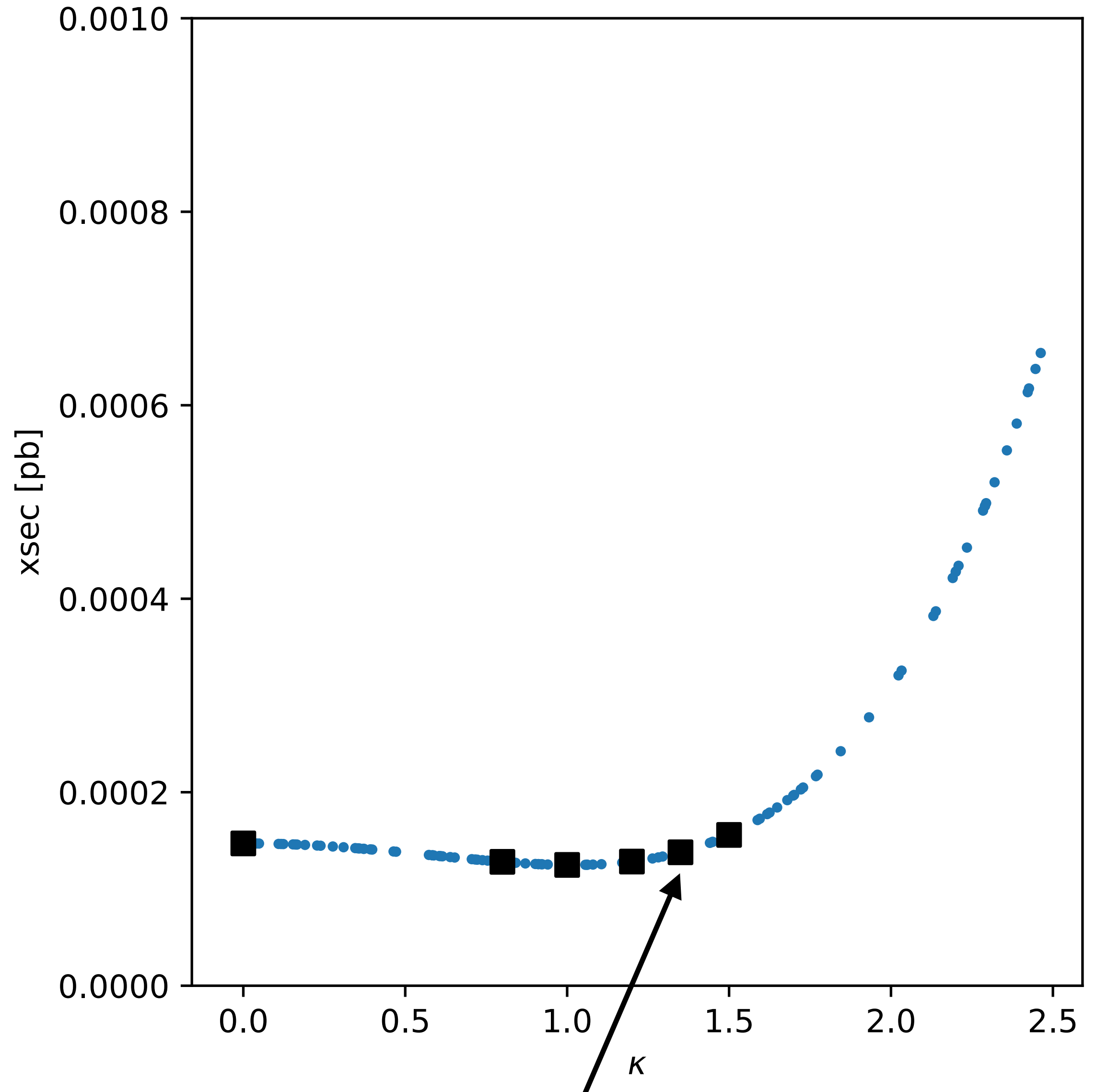
Figure 1. Schematical workflow, with classes in red, external simulations in blue, and files in green.

Madminer: How it works

Madminer package wraps around Madgraph, Pythia, Delphes, to simplify ‘mining gold’, all the way to inference:

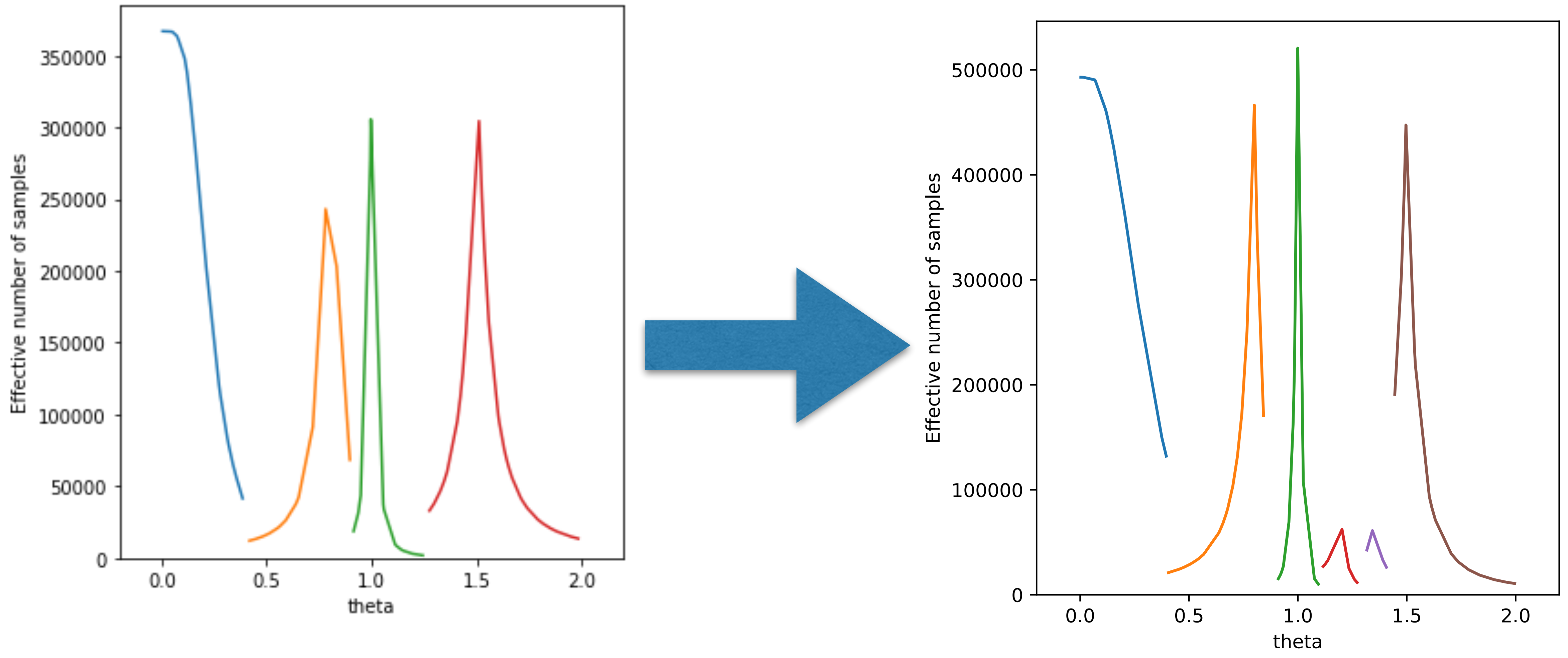
- Simulates events in Madgraph at fixed values of a theory parameter
- Re-weights each event to other benchmark theory parameter points
- Does event-wise ‘morphing’ to allow having training/test events at **any value of the parameter point**
- **Calculates** the targets to regress, **joint score, joint likelihood ratio**, which will allow for **data augmented training**
- Defines various PyTorch models with losses based on the likelihood ratio and augmented data
- Allows simple asymptotic limits on Asimov datasets (for more involved statistics, need to integrate with pyhf)

Trouble with morphing near SM



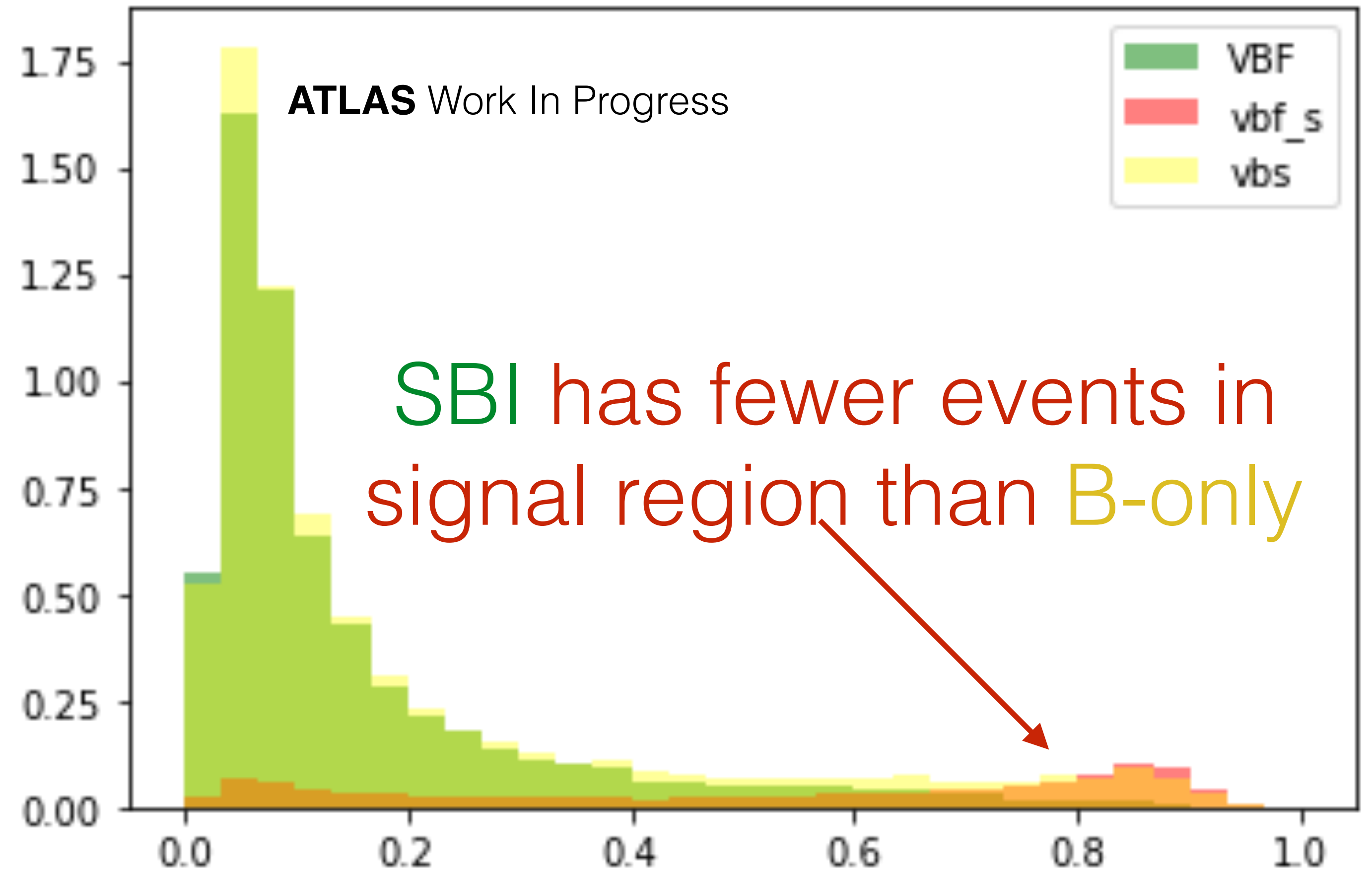
Validation at independent point 1.35

κ^4 = μ (To scale H signal strength by μ scale HVV couplings by κ^4)



Effective number of events = 1/max(event_weights)

Classifier trained on S vs B



N_effective rapidly falls as we try to morph events from one point to another near the SM (mu=1)
The physics changes too fast, probably because there is almost maximal interference near SM, so very few ‘signal-only like’ events to morph

https://arxiv.org/pdf/1805.00013.pdf
https://arxiv.org/pdf/1805.12244.pdf

Marginalise z by minimising L

$$t(x,z|\theta) \equiv \nabla_\theta \log p(x,z|\theta) = \frac{p(x|z_d)\,p(z_d|z_s)\,p(z_s|z_p)\,\nabla_\theta p(z_p|\theta)}{p(x|z_d)\,p(z_d|z_s)\,p(z_s|z_p)\,p(z_p|\theta)} = \frac{\nabla_\theta d\sigma(z_p|\theta)}{d\sigma(z_p|\theta)} - \frac{\nabla_\theta \sigma(\theta)}{\sigma(\theta)}\,.$$

z are latent variables / intermediate information like Parton-level four-momenta, Parton shower trajectories, Detector interactions

$$L_t = \mathbb{E}_{p(x,z|\theta_0)}\left[\left(t(x,z|\theta_0) - \hat{t}(x|\theta_0)\right)^2\right]$$

θ_0, θ_1 are 2 alternative values of the theory parameter being measured (like μ = 5 vs μ = 1)

which is minimized by $t^*(x) = \mathbb{E}_{p(z|x,\theta_0)}\left[t(x,z|\theta_0)\right] = t(x|\theta_0)$.

it tells you how the weights of an events will be affected for a small change in θ near θ_0

$$r(x|\theta_0,\theta_1) = \frac{p(x|\theta_0)}{p(x|\theta_1)}$$

$$L_r = \mathbb{E}_{p(x,z|\theta_1)}\left[\left(r(x,z|\theta_0,\theta_1) - \hat{r}(x)\right)^2\right]$$

We have that from distribution of events from the simulator

We calculate this additional information (based on ME) after the event generation & re-weighting to other theory points with Madminer

Output of the network

Under the hood: Data Augmentation and Loss Functions

Data Augmentation:

```
augmented_data = []
for definition in augmented_data_definitions:
    if definition[0] == "ratio":
        _, i_num, i_den = definition
        ratio = (weights[i_num] / xsecs[i_num]) / (weights[i_den] / xsecs[i_den])
        ratio = ratio.reshape((-1, 1))  # (n_samples, 1)
        augmented_data.append(ratio)
    elif definition[0] == "score":
        _, i = definition
        score = weight_gradients[i, :, :] / weights[i, np.newaxis, :]  # (n_gradients, n_samples)
        score = score - xsec_gradients[i, :, np.newaxis] / xsecs[i, np.newaxis, np.newaxis]
        score = score.T  # (n_samples, n_gradients)
        augmented_data.append(score)
    else:
        raise ValueError("Unknown augmented data type {}".format(definition[0]))
```

Sally Loss:

```
def local_score_mse(t_hat, t_true):
    return MSELoss()(t_hat, t_true)
```

Alices Losses:

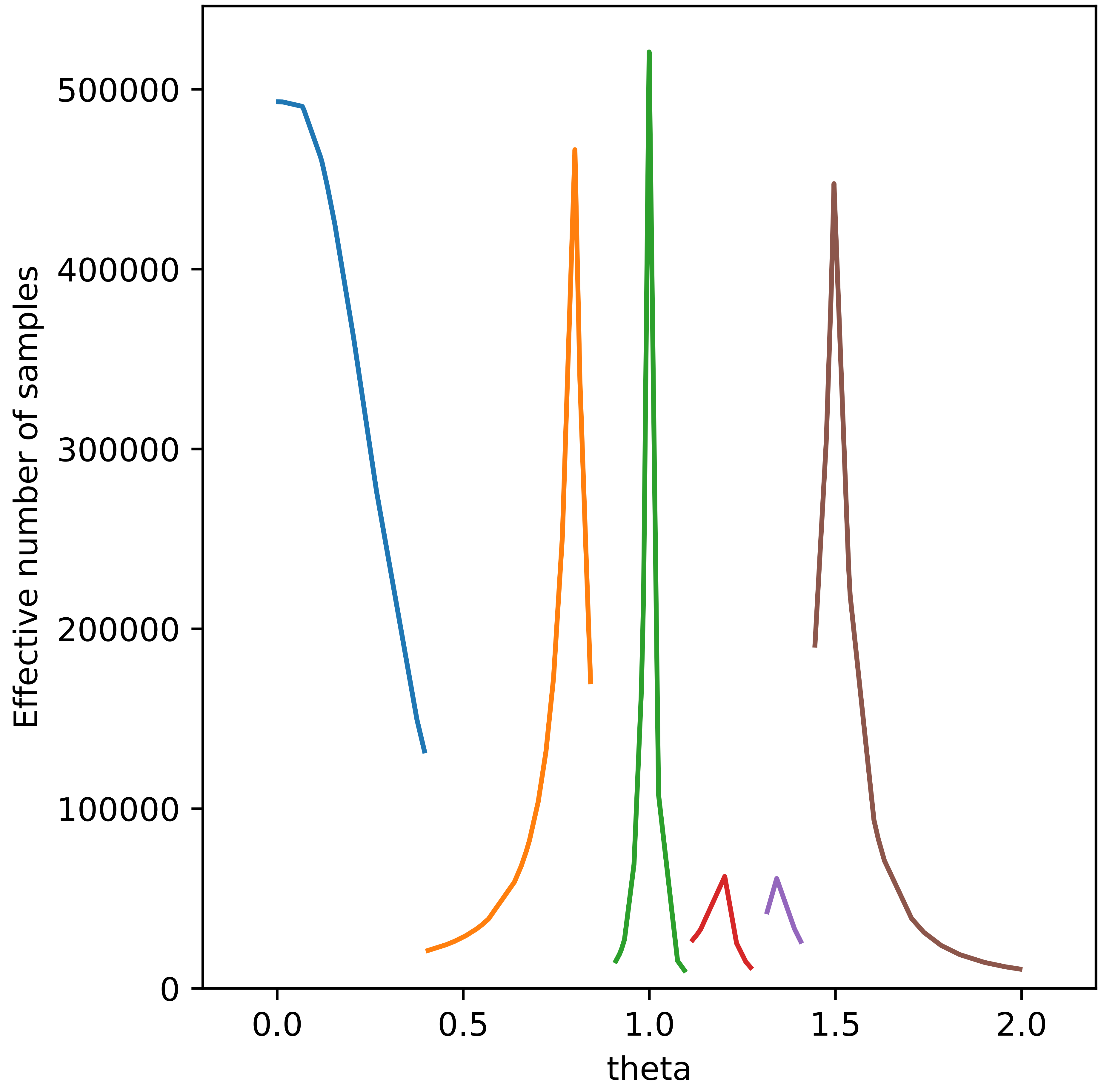
```
def ratio_augmented_xe(s_hat, log_r_hat, t0_hat, t1_hat, y_true, r_true, t0_true, t1_true):
    s_hat = 1.0 / (1.0 + torch.exp(log_r_hat))
    s_true = 1.0 / (1.0 + r_true)

    return BCELoss()(s_hat, s_true)

def ratio_score_mse_num(s_hat, log_r_hat, t0_hat, t1_hat, y_true, r_true, t0_true, t1_true):
    return MSELoss()((1.0 - y_true) * t0_hat, (1.0 - y_true) * t0_true)
```

y_true means theta0 vs theta1, not signal vs background

Morphing for Gold

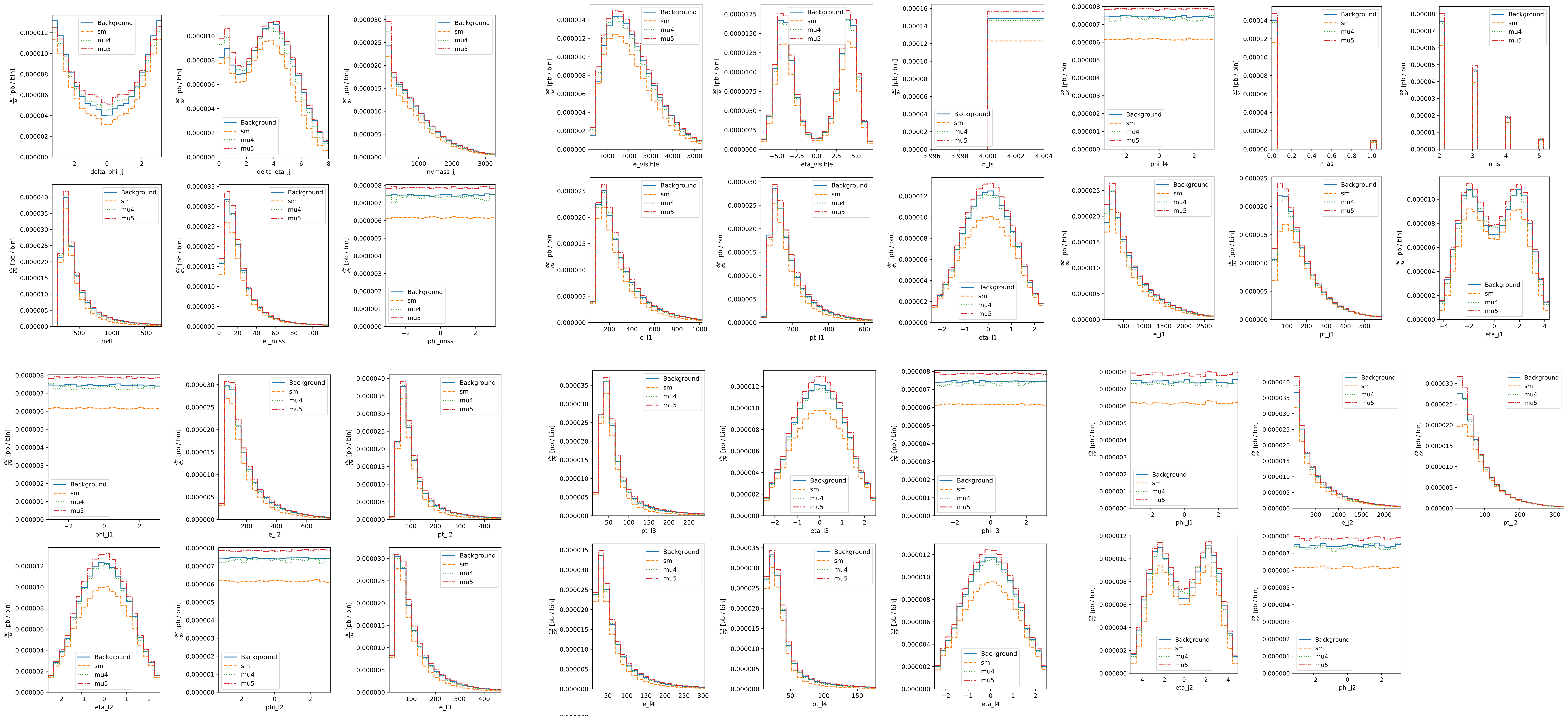


Effective number of events = 1/max(event_weights)

- How much more likely would this event be if the true value of mu was 2 instead of 1?: Matrix Element Re-Weighting
- 5 benchmarks is enough to fit a 4-polynomial
- From the polynomial you now have gradients for score t

To make training easier, an unweighting is done by allowing to sample with replacement

Distribution of Features Used



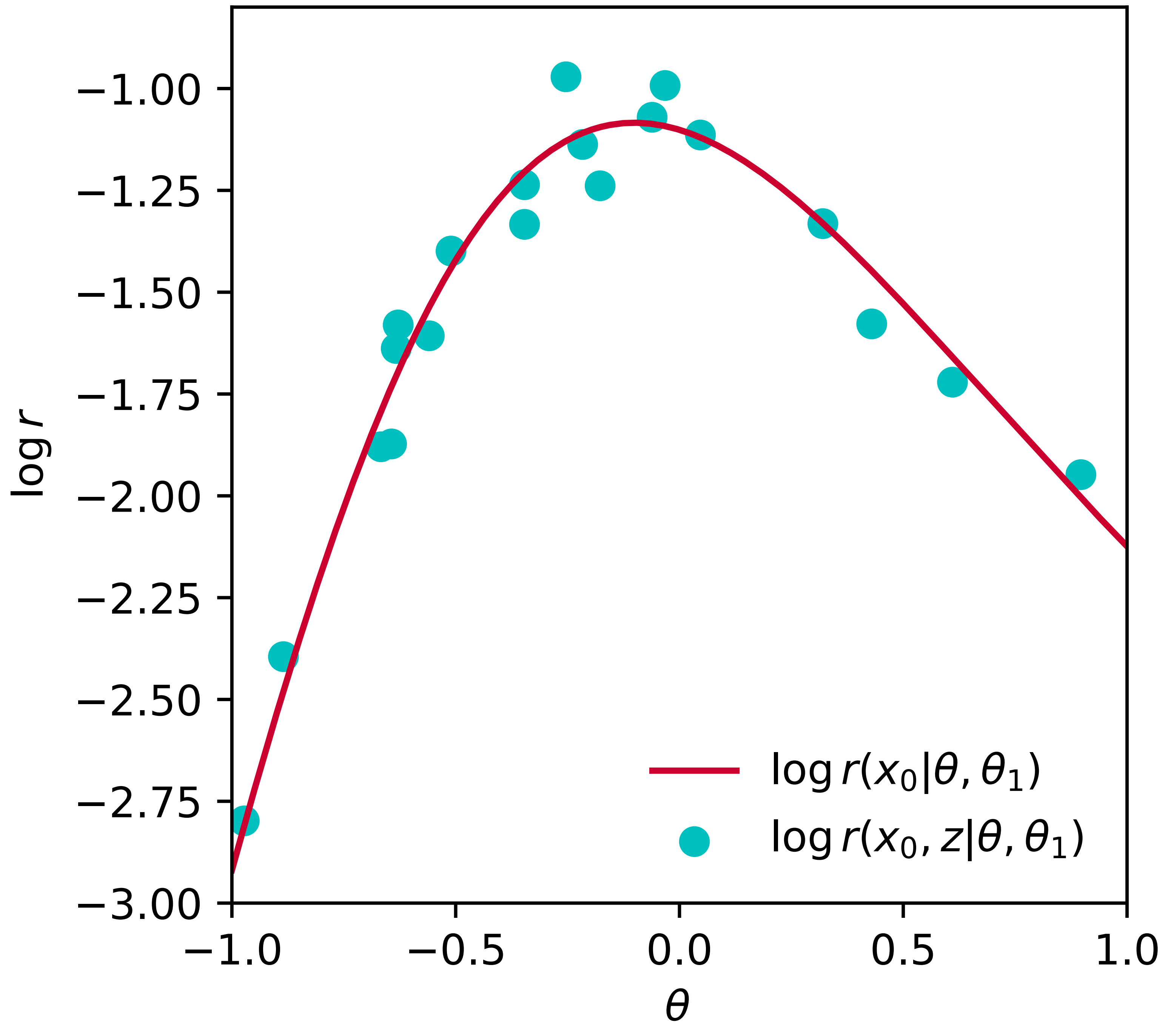
4 vector of 4l, 2j, dijet variables, n_jets, n_leptons, variables for visible & missing

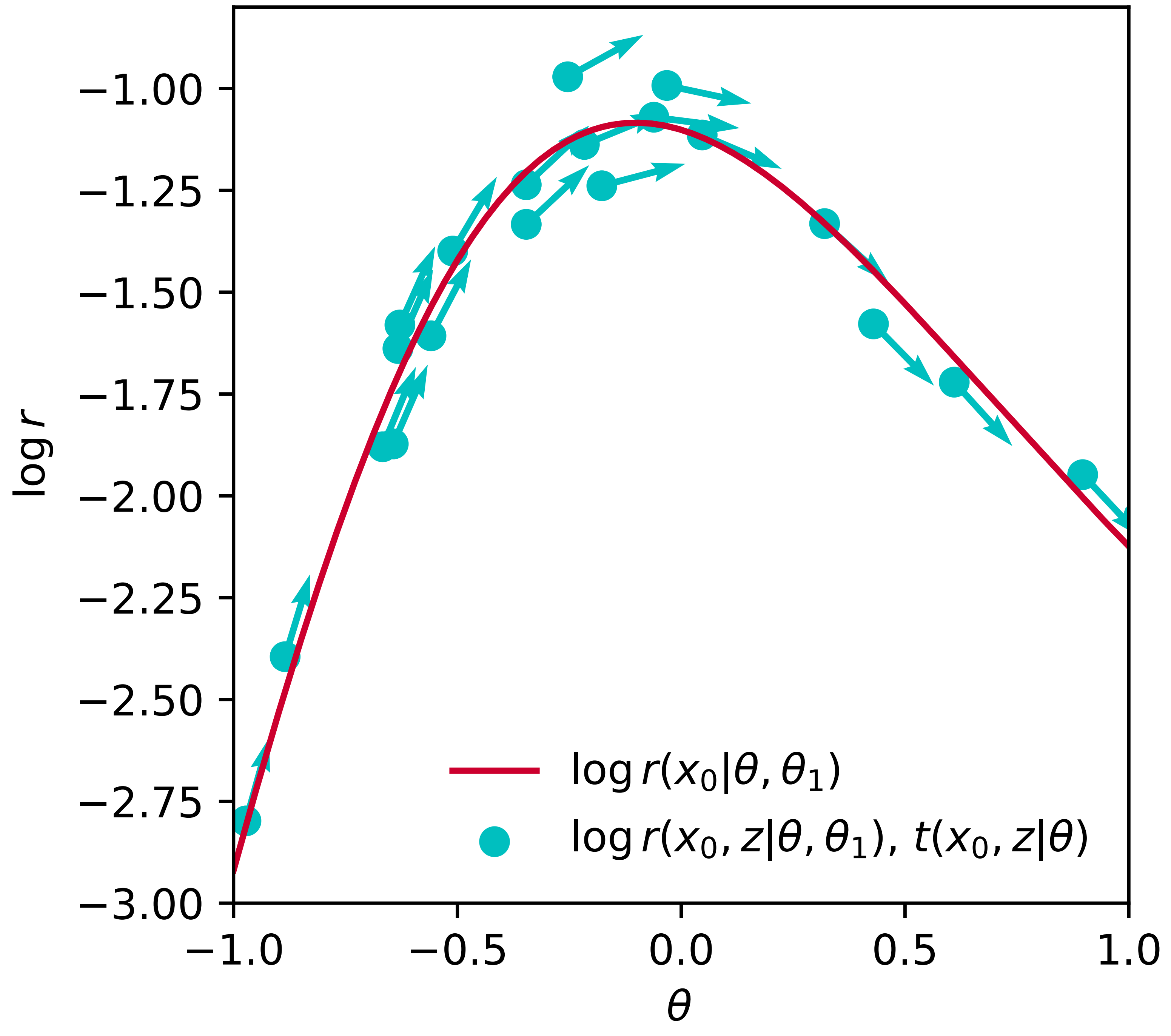
Permutation Importance developed for Physicists

- “Permutation Importance” I mentioned in my last talk at h4l got noticed but using it correctly can be tricky; ELI5 package silently ignores weights, can use wrong default metric irrelevant to particle physics
- Now available: a pip installable PI package for particle physics (https://github.com/aghoshpub/permutationImportancePhysics) with predefined metrics (AUC that handles negative weights, Significance of discovery) with a simple tutorial to get started
    - I will soon get parallelisation support for speed up (but already faster than ELI5 implementation)

One more piece: the score

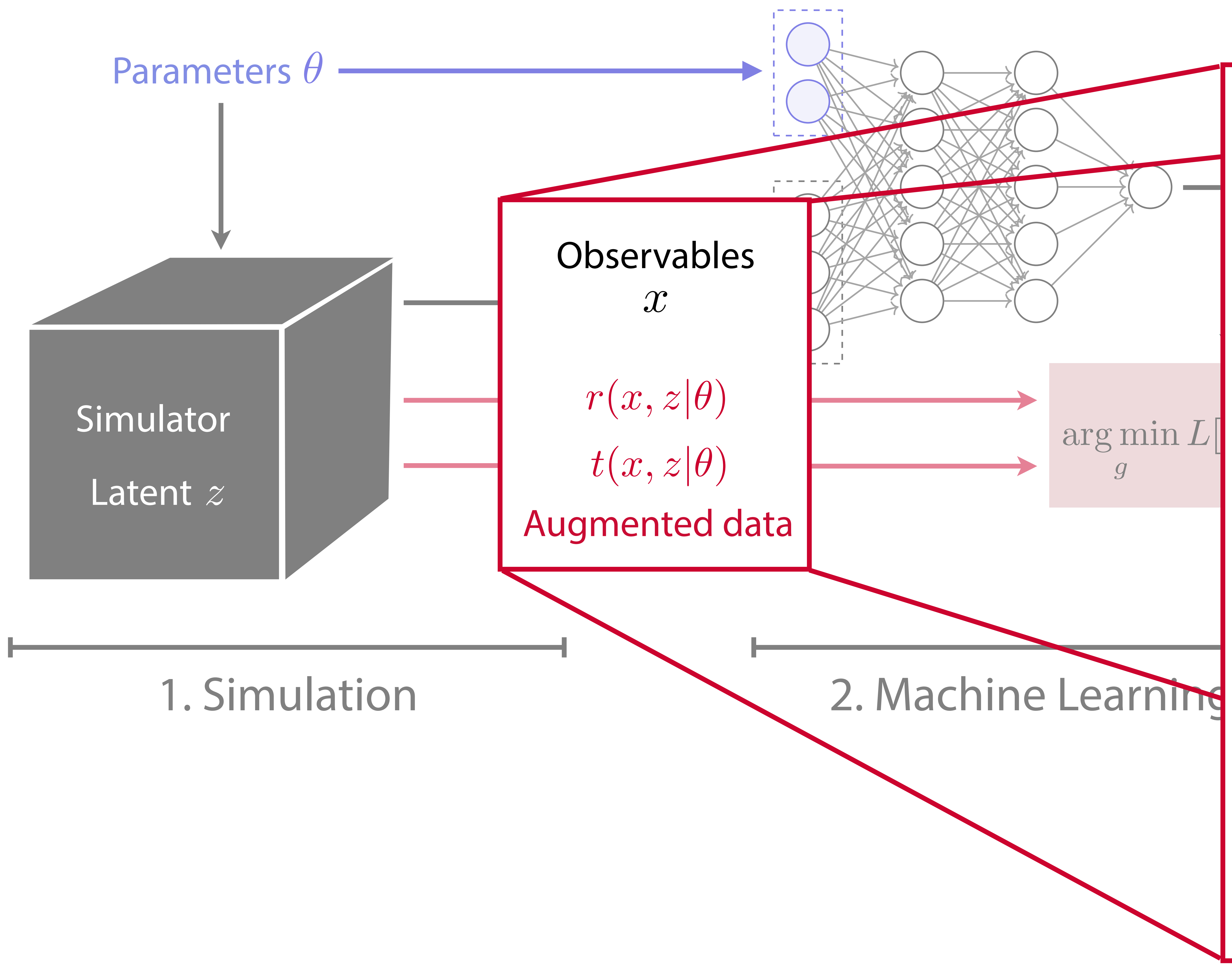
- Knowing derivative often helps fitting:



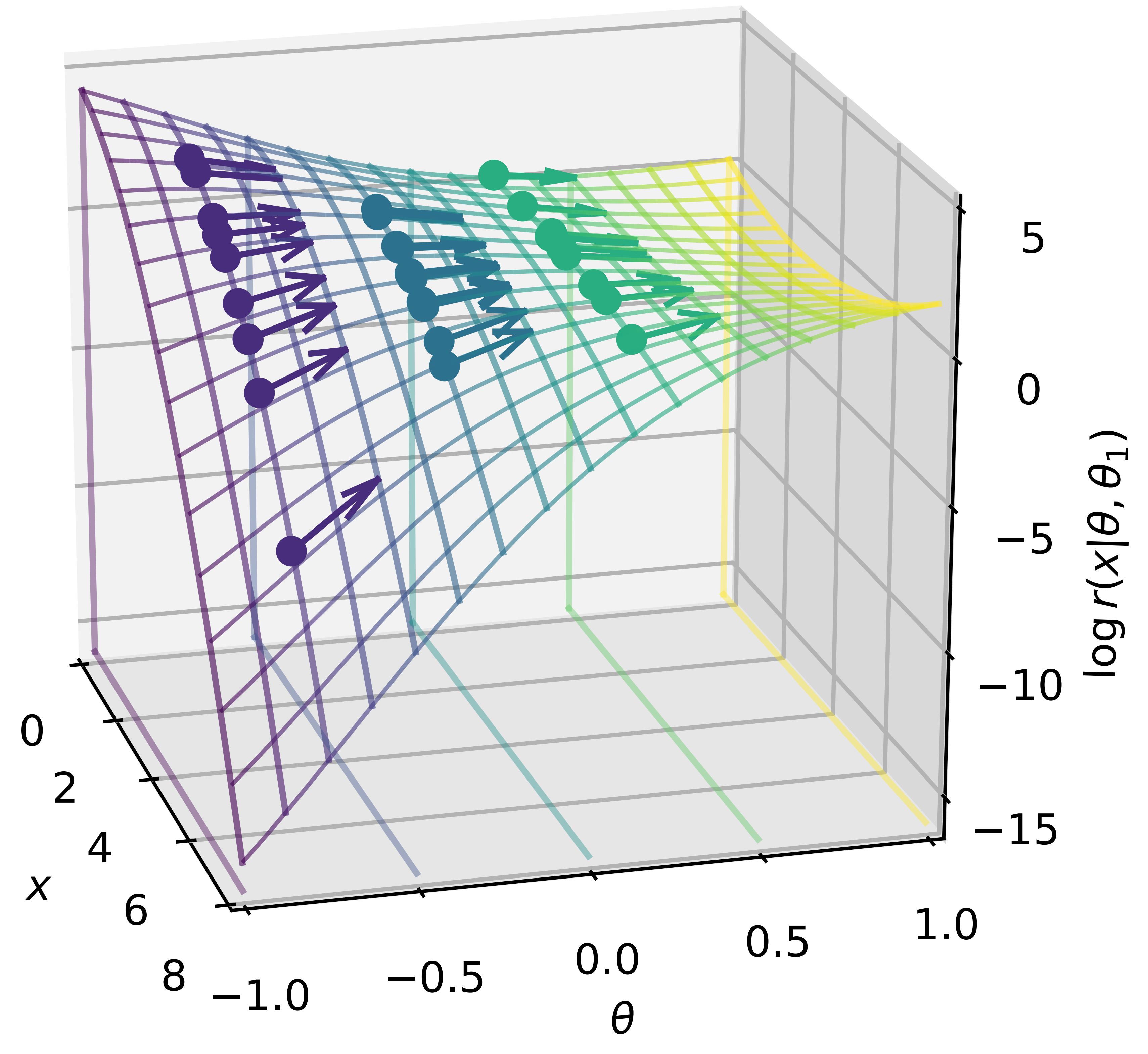


- In our case, the relevant quantity is the score $t(x|\theta_0) \equiv \nabla_\theta \log p(x|\theta) \Big|_{\theta_0}$.
  - The score itself is intractable. But…

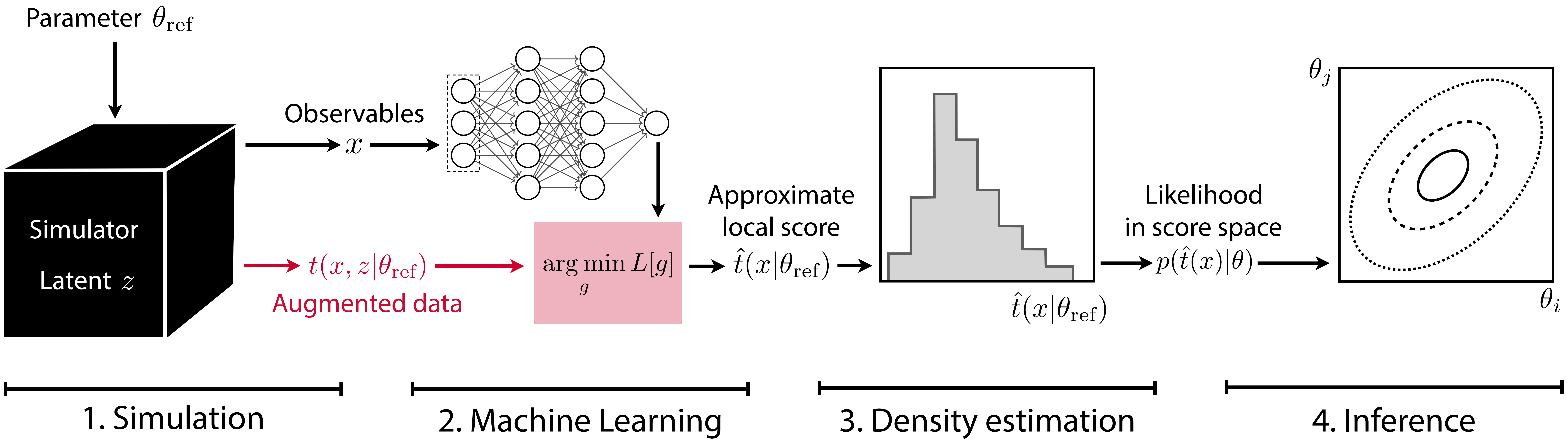
Putting the pieces together: RASCAL (Ratio and score approximate likelihood ratio)



RASCAL combines three orthogonal pieces of information



SALLY (Score approximates likelihood locally)

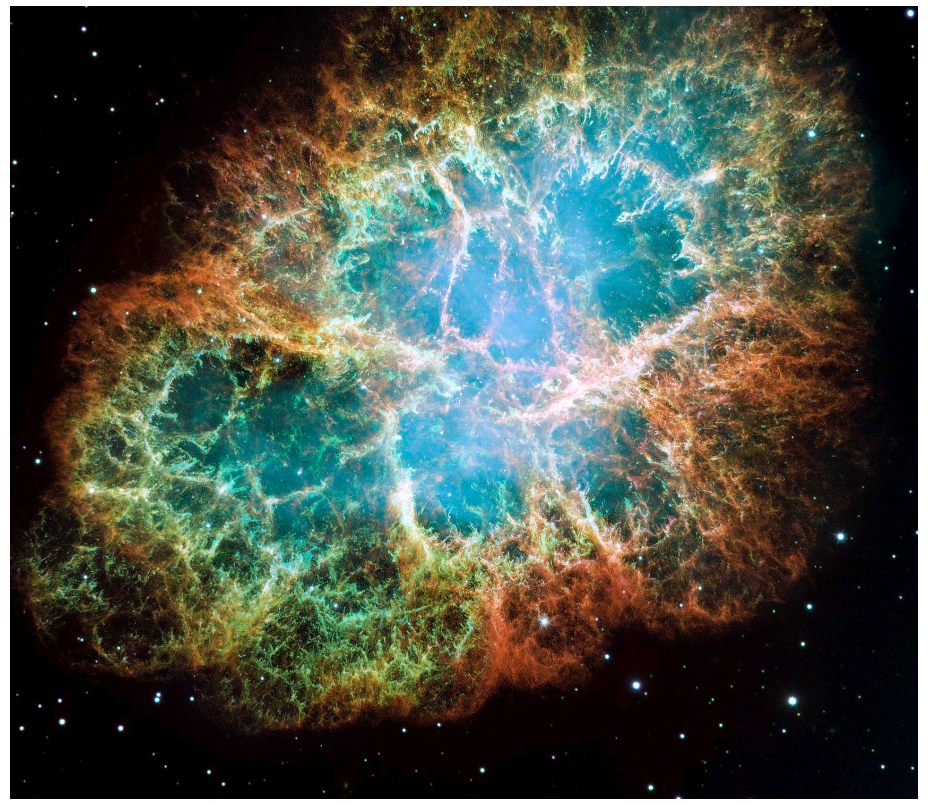


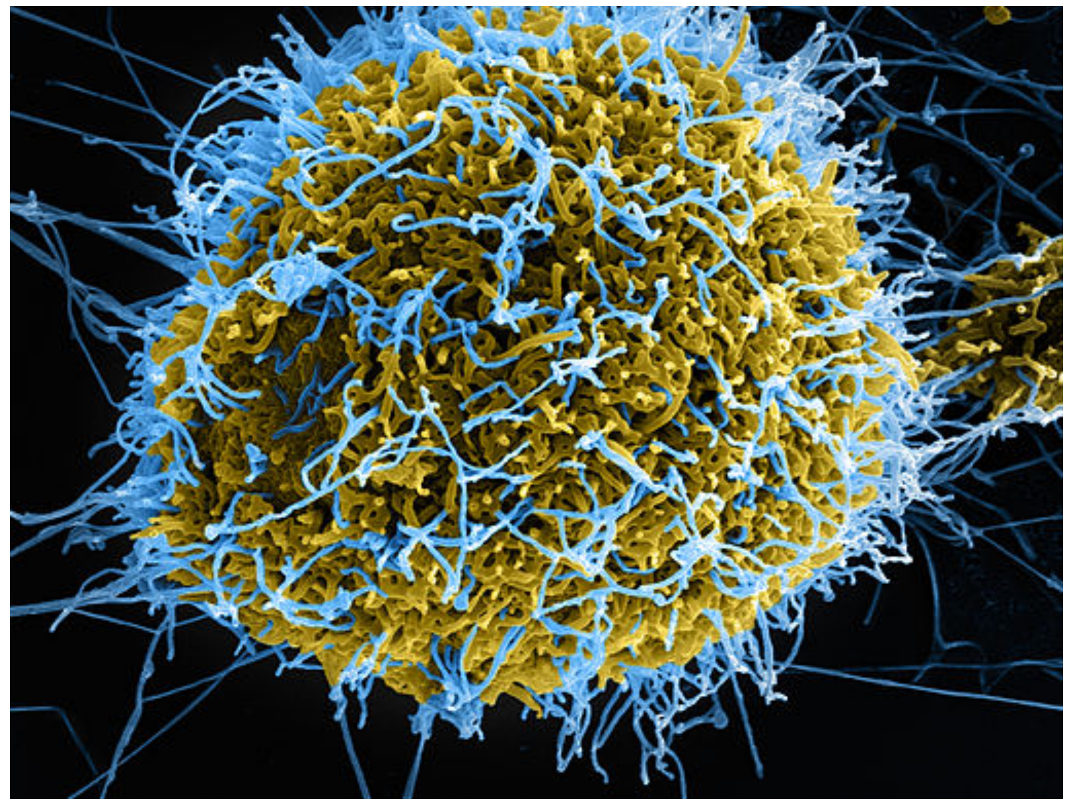
"The machine learning version of Optimal Observables":

- Simpler & more robust than RASCAL
- Just as powerful close to θ_{ref}, but can lead to suboptimal limits further away

- Don't blindly trust the neural network?
  - Many diagnostic cross checks
  - Calibration / Neyman construction: badly trained network can lead to suboptimal limits, but not to wrong limits
- Applications beyond particle physics

[NASA, NIAID]





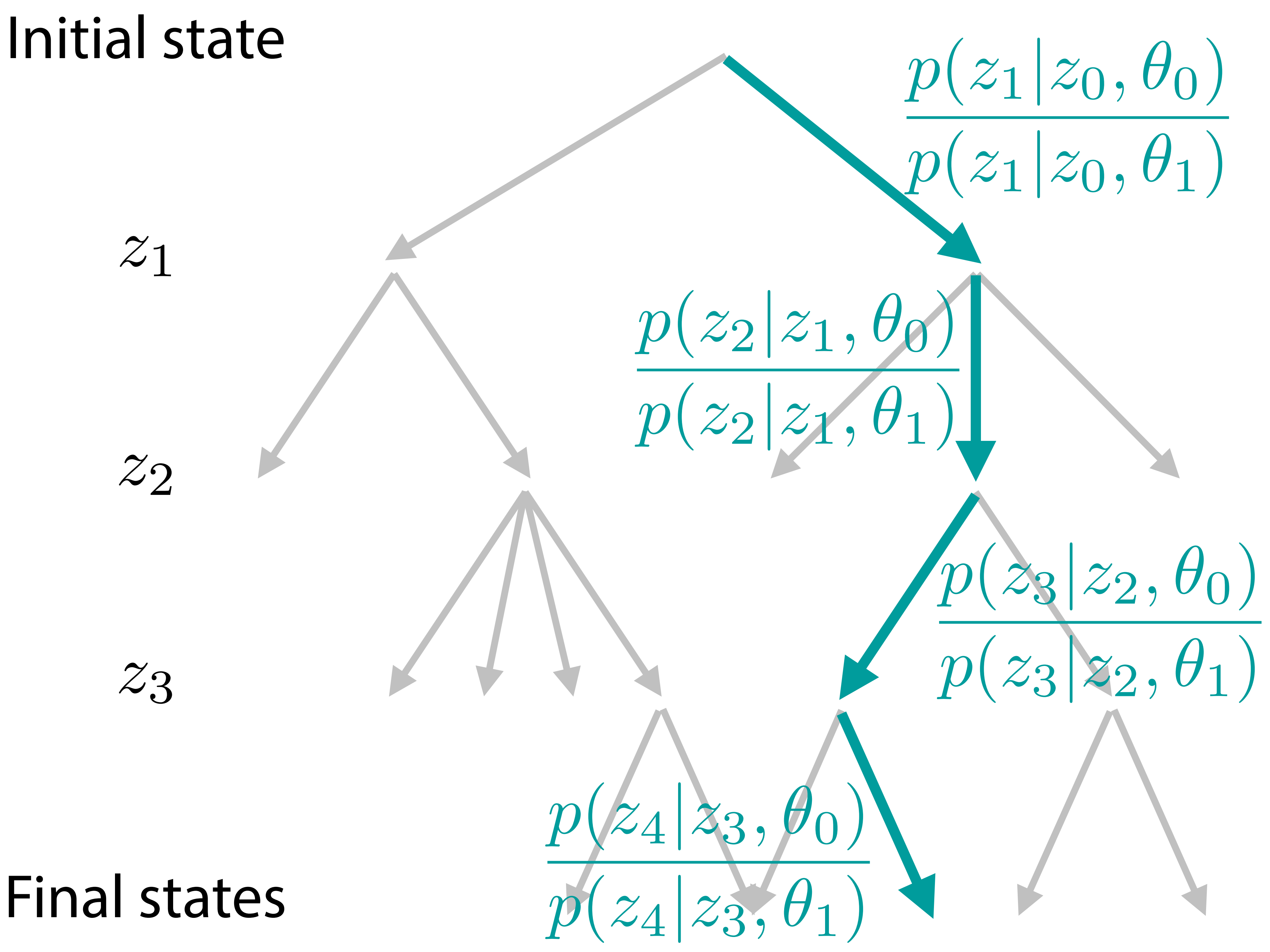
Extracting the joint likelihood ratio from any simulation

- Computer simulation typically evolve along a tree-like structure of successive random branchings
- The probabilities of each branching $p_i(z_i|z_{i-1},\theta)$ are often clearly defined in the code:

```
if random() > 0.1 + 2.5 * model_parameter:
    do_one_thing()
else:
    do_another_thing()
```

- For each run of the simulator, we can calculate the probability of the chosen path for different values of the parameters, and the "joint likelihood ratio":

$$r(x,z|\theta_0,\theta_1) = \frac{p(x,z|\theta_0)}{p(x,z|\theta_1)} = \prod_i \frac{p(z_i|z_{i-1},\theta_0)}{p(z_i|z_{i-1},\theta_1)}$$



Variational calculus

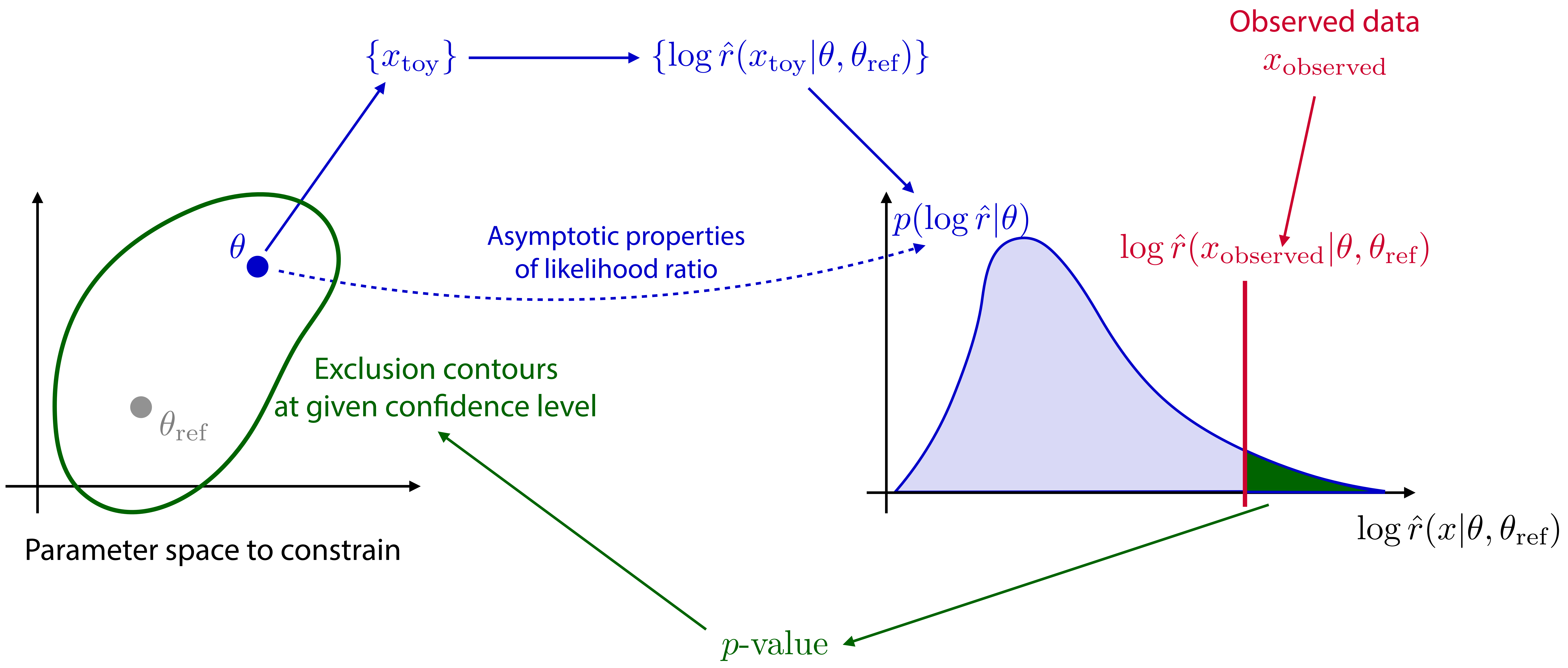
$$L[\hat{g}(x)] = \int \mathrm{d}x\, \mathrm{d}z\; p(x,z|\theta)\; |g(x,z) - \hat{g}(x)|^2$$

$$= \int \mathrm{d}x\; \underbrace{\left[\hat{g}^2(x) \int \mathrm{d}z\; p(x,z|\theta) - 2\hat{g}(x) \int \mathrm{d}z\; p(x,z|\theta)\, g(x,z) + \int \mathrm{d}z\; p(x,z|\theta)\, g^2(x,z)\right]}_{F(x)}$$

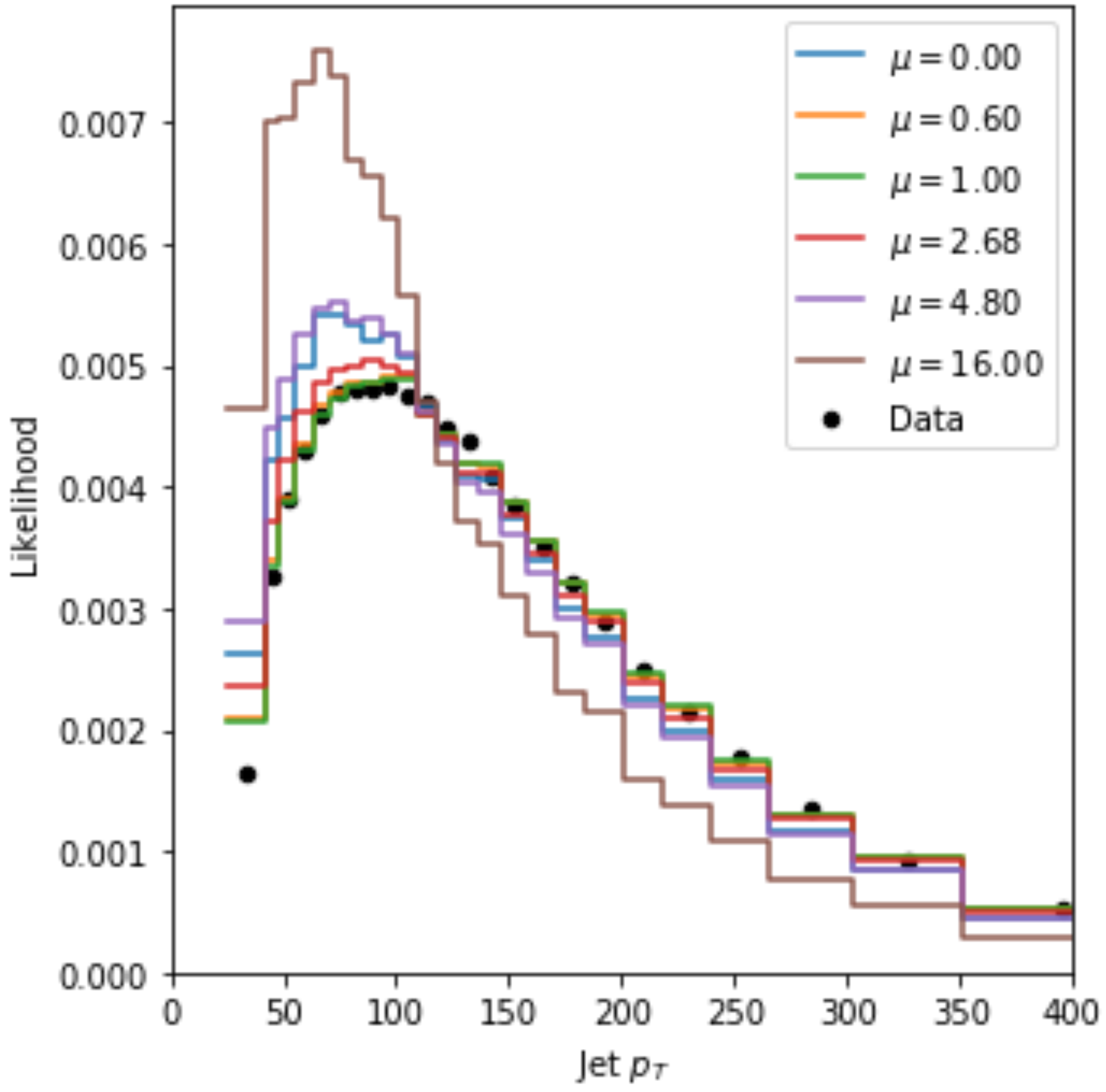
$$0 = \left.\frac{\delta F}{\delta \hat{g}}\right|_{g^*} = 2\hat{g} \underbrace{\int \mathrm{d}z\; p(x,z|\theta)}_{=p(x|\theta)} - 2 \int \mathrm{d}z\; p(x,z|\theta)\, g(x,z)$$

$$g^*(x) = \frac{1}{p(x|\theta)} \int \mathrm{d}z\; p(x,z|\theta)\, g(x,z)$$

Limit setting (frequentist)



Limits from Histograms

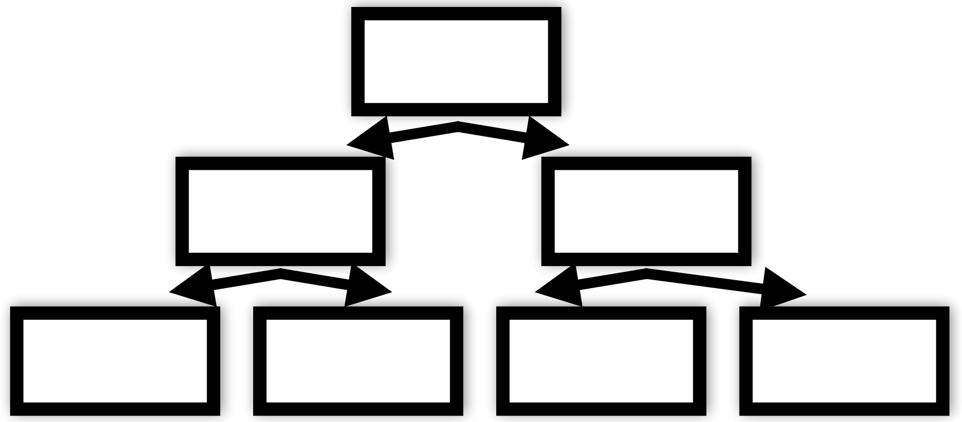


Could do better with pyhf

Which is the best ML solution?

Which is the best ML solution?

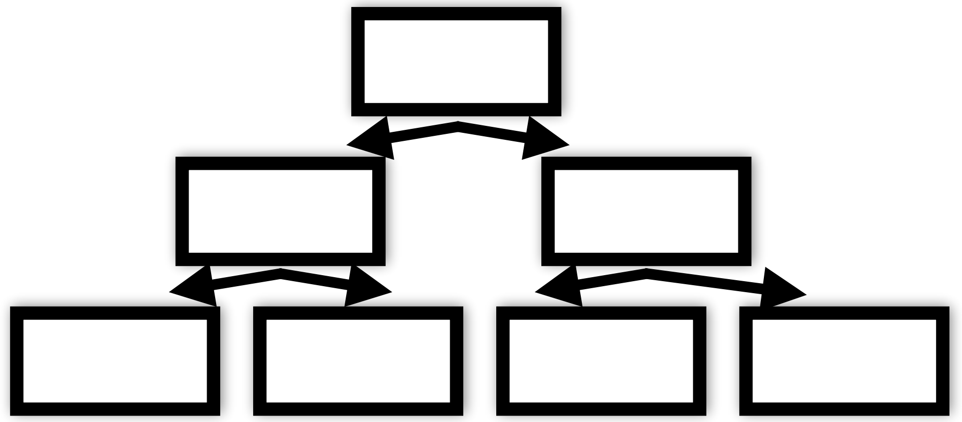
$$Z_0 = \sqrt{2\left[(SVI + B2)\ln\left(1 + \frac{SVI - V}{V + B2}\right) - (SVI - V)\right]}$$



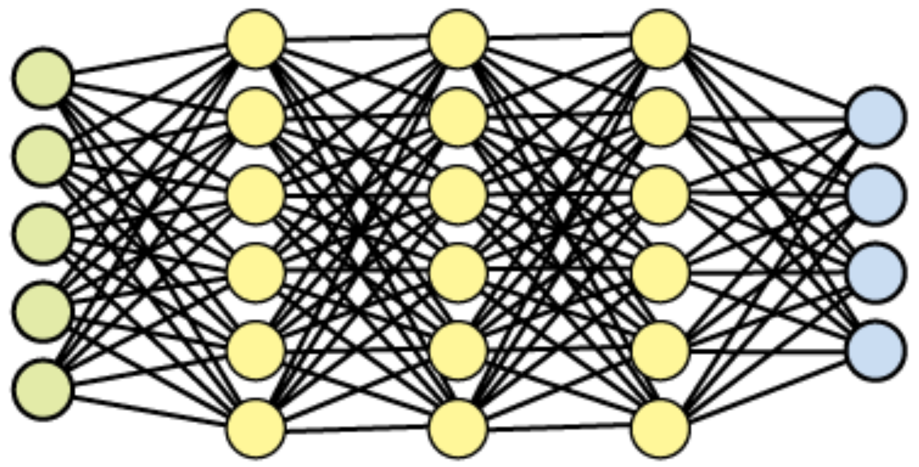
Decision Tree like categoriser to optimise $\sum_{categories} {Z_0}^2$ instead of gini

Which is the best ML solution?

$$Z_0 = \sqrt{2\left[(SVI + B2)\ln\left(1 + \frac{SVI - V}{V + B2}\right) - (SVI - V)\right]}$$



Decision Tree like categoriser to optimise $\sum_{categories} {Z_0}^2$ instead of gini



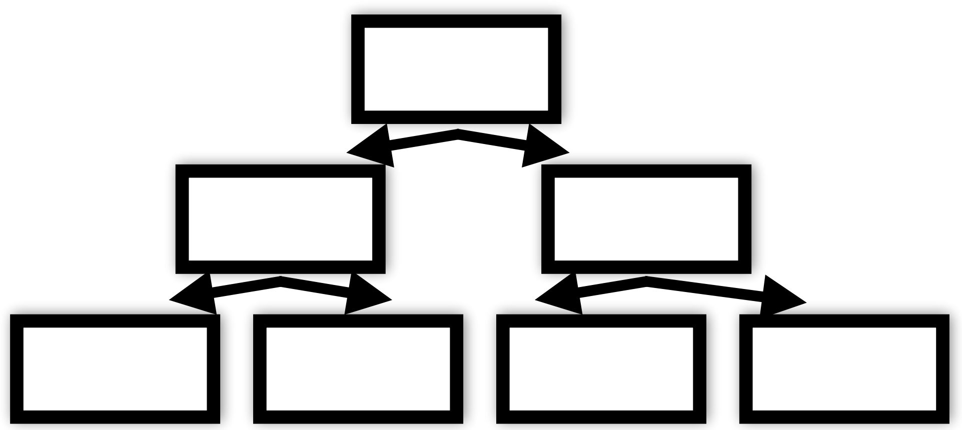
Not trying to predict class labels Y

NN to categorise with batch level loss: $\sum_{categories} {Z_0}^2$

Simultaneous fit on all categories for Likelihood

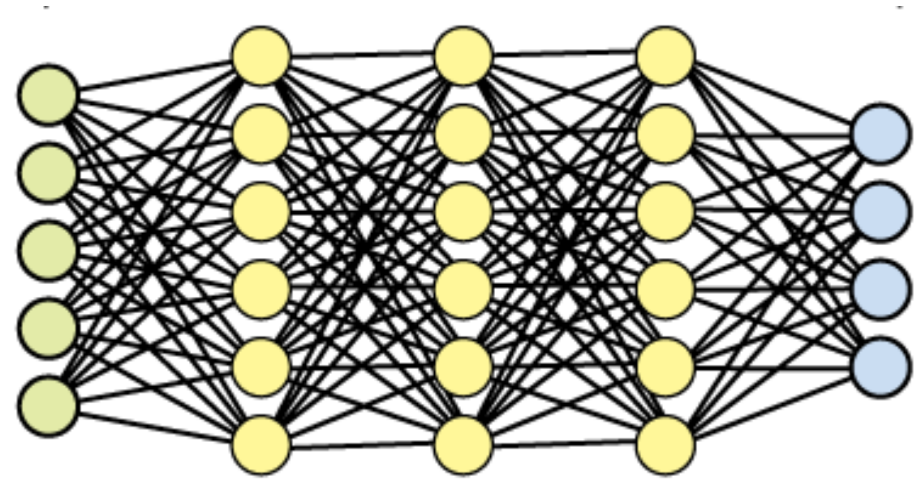
Which is the best ML solution?

$$Z_0 = \sqrt{2\left[(SVI + B2)\ln\left(1 + \frac{SVI - V}{V + B2}\right) - (SVI - V)\right]}$$



Decision Tree like categoriser to optimise $\sum_{categories} {Z_0}^2$ instead of gini

Simple classifier(s) on :
S vs SVI + B2
or
SVI + B2 vs V + B2 …



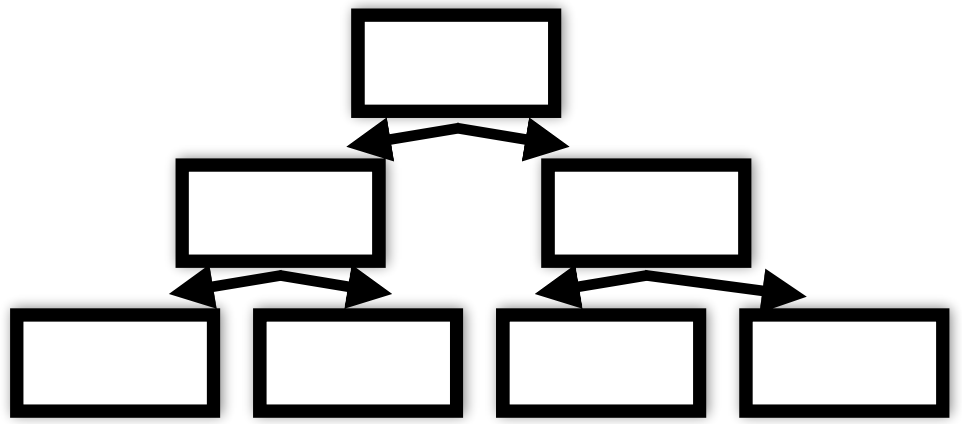
Not trying to predict class labels Y

NN to categorise with batch level loss: $\sum_{categories} {Z_0}^2$

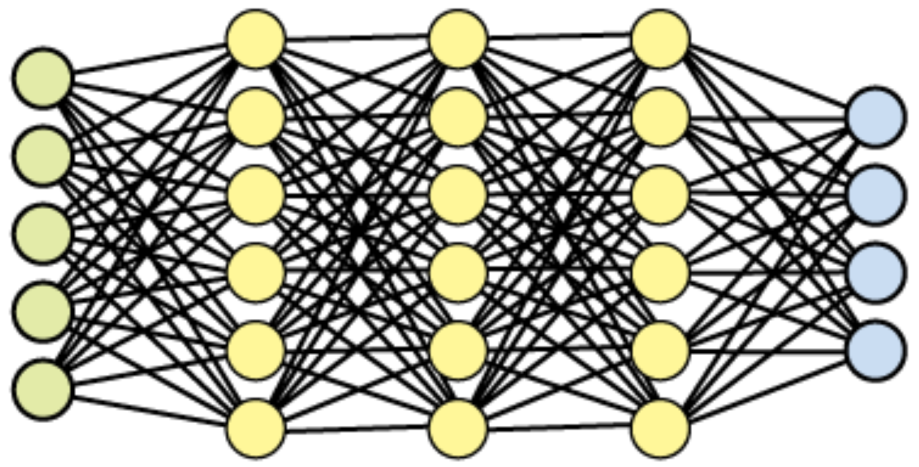
Simultaneous fit on all categories for Likelihood

Which is the best ML solution?

$$Z_0 = \sqrt{2\left[(SVI + B2)\ln\left(1 + \frac{SVI - V}{V + B2}\right) - (SVI - V)\right]}$$



Decision Tree like categoriser to optimise $\sum_{categories} {Z_0}^2$ instead of gini



NN to categorise with batch level loss: $\sum_{categories} {Z_0}^2$

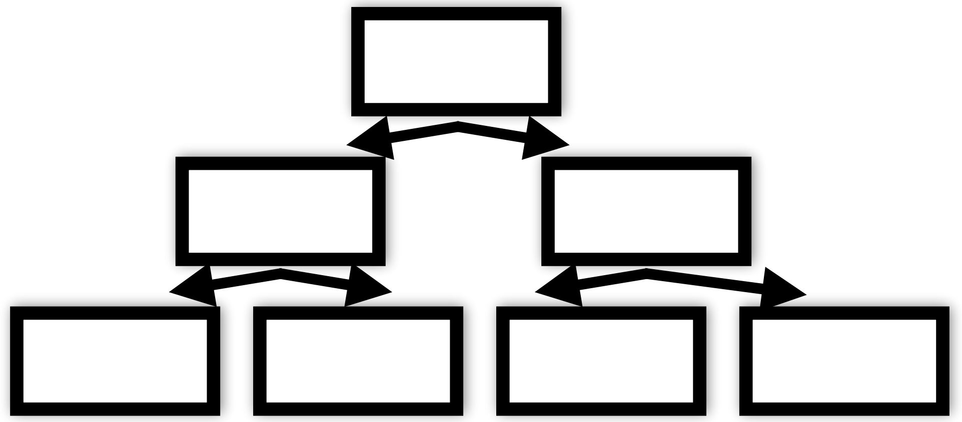
Simultaneous fit on all categories for Likelihood

Simple classifier(s) on :
S vs SVI + B2
or
SVI + B2 vs V + B2 …

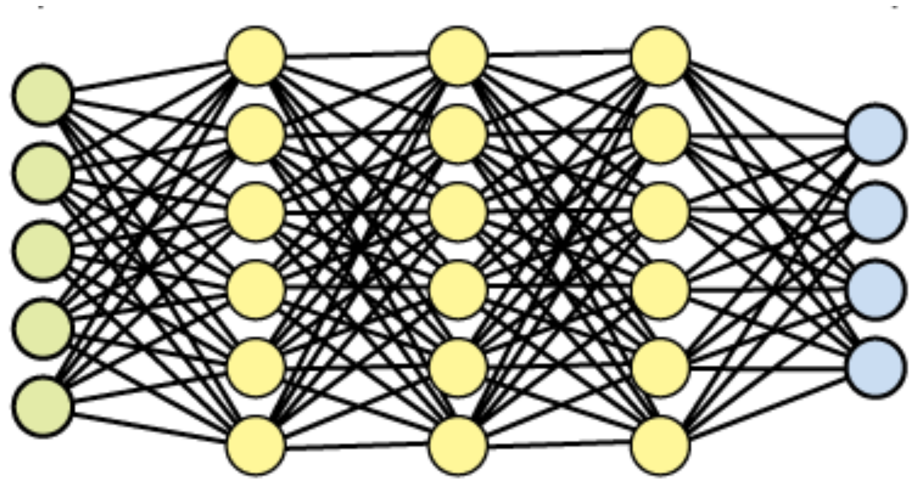
Combine with Reparameterisation + Template method as in ATLAS A->tt analysis, and Noam Tal Hod ?

Which is the best ML solution?

$$Z_0 = \sqrt{2\left[(SVI + B2)\ln\left(1 + \frac{SVI - V}{V + B2}\right) - (SVI - V)\right]}$$



Decision Tree like categoriser to optimise $\sum_{categories} {Z_0}^2$ instead of gini



Not trying to predict class labels Y

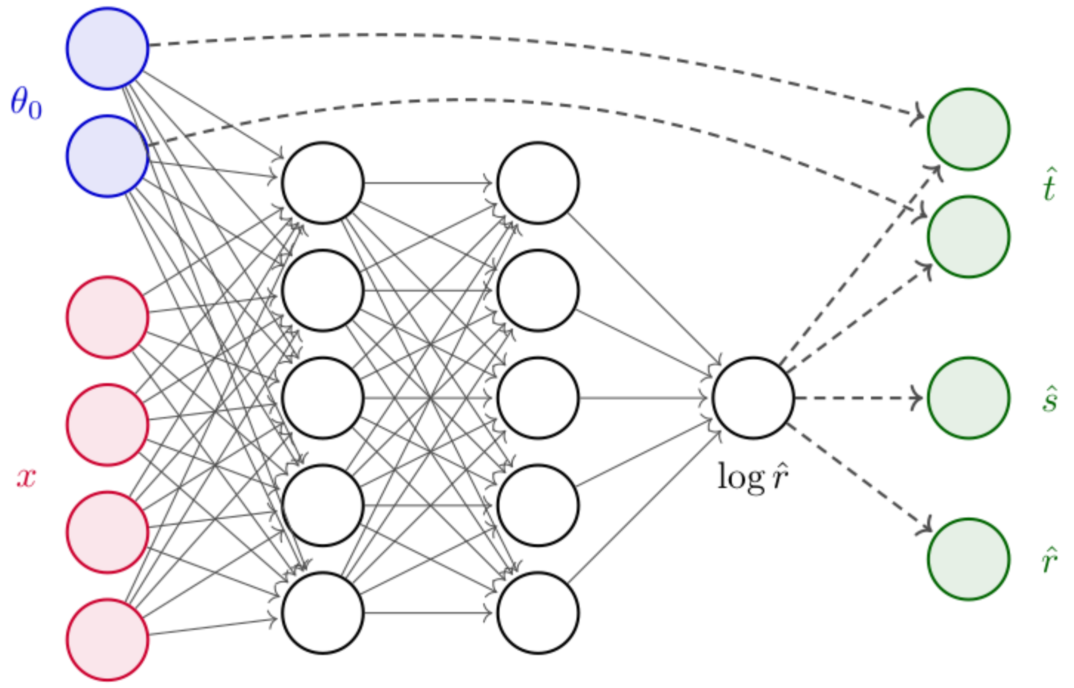
NN to categorise with batch level loss: $\sum_{categories} {Z_0}^2$

Simultaneous fit on all categories for Likelihood

Simple classifier(s) on :
S vs SVI + B2
or
SVI + B2 vs V + B2 …

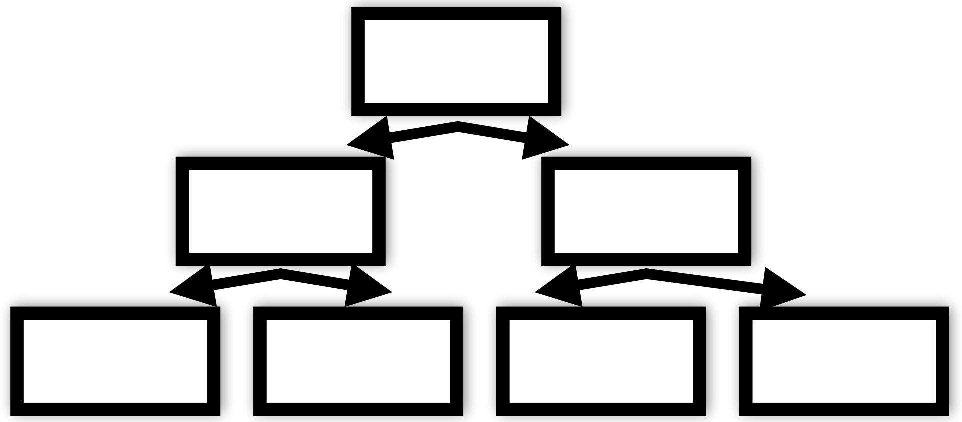
Combine with Reparameterisation + Template method as in ATLAS A->tt analysis, and Noam Tal Hod ?

Likelihood-free inference: Brehmer, Cranmer, Louppe, Pavez

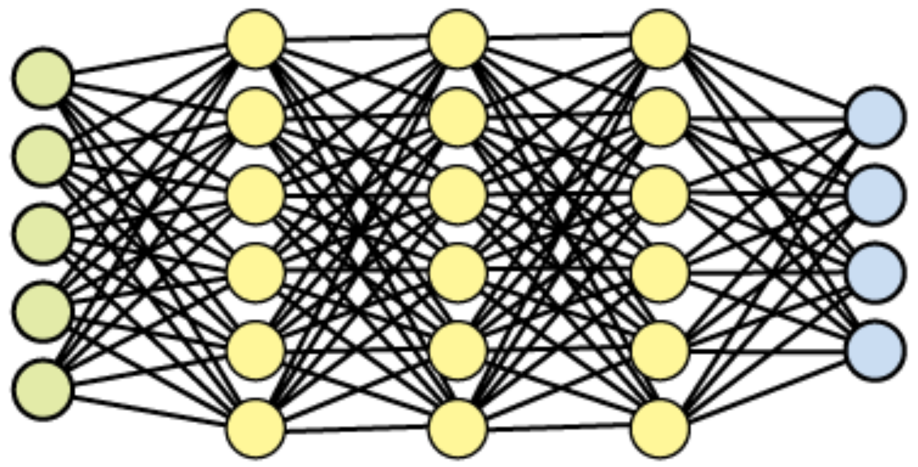


Which is the best ML solution?

$$Z_0 = \sqrt{2\left[(SVI + B2)\ln\left(1 + \frac{SVI - V}{V + B2}\right) - (SVI - V)\right]}$$



Decision Tree like categoriser to optimise $\sum_{categories} {Z_0}^2$ instead of gini



Not trying to predict class labels Y

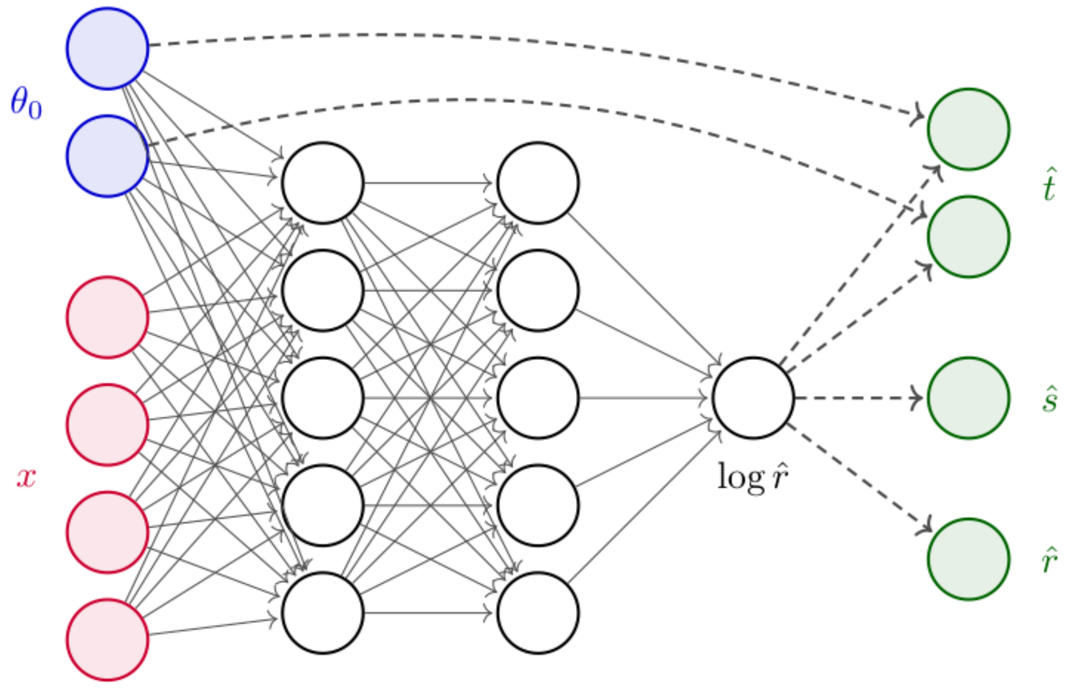
NN to categorise with batch level loss: $\sum_{categories} {Z_0}^2$

Simultaneous fit on all categories for Likelihood

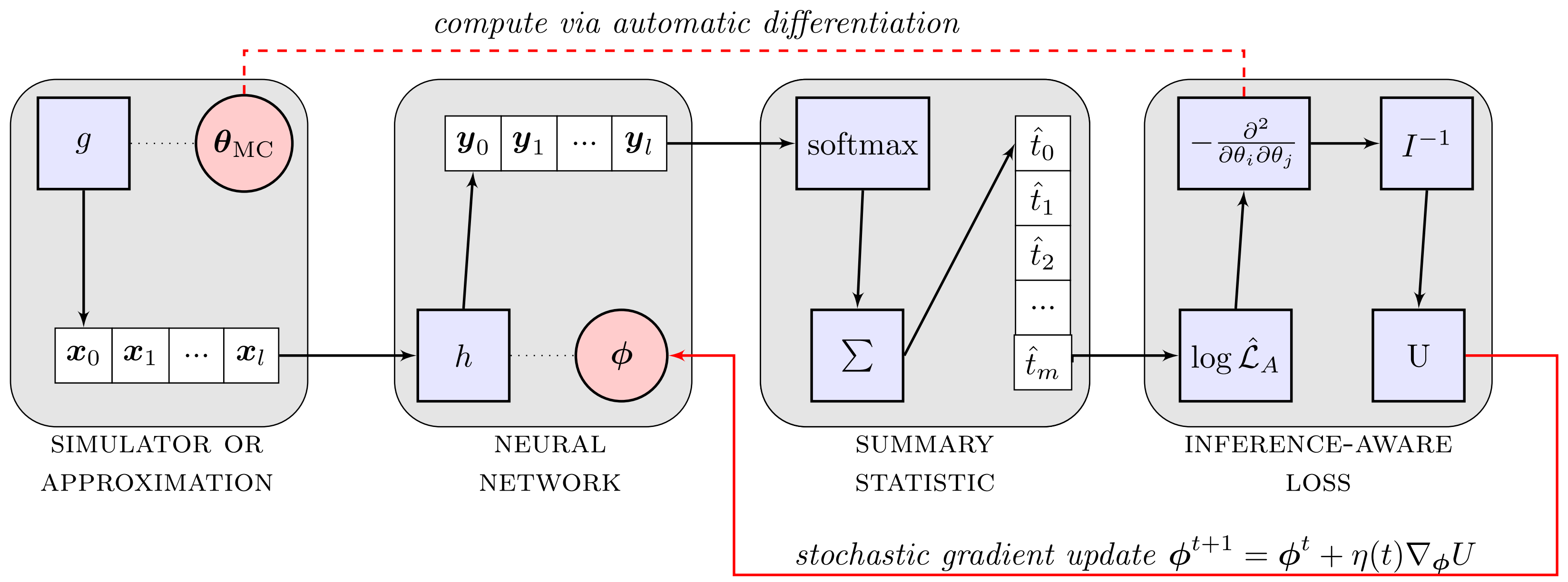
Simple classifier(s) on :
S vs SVI + B2
or
SVI + B2 vs V + B2 …

Combine with Reparameterisation + Template method as in ATLAS A->tt analysis, and Noam Tal Hod ?

Likelihood-free inference: Brehmer, Cranmer, Louppe, Pavez

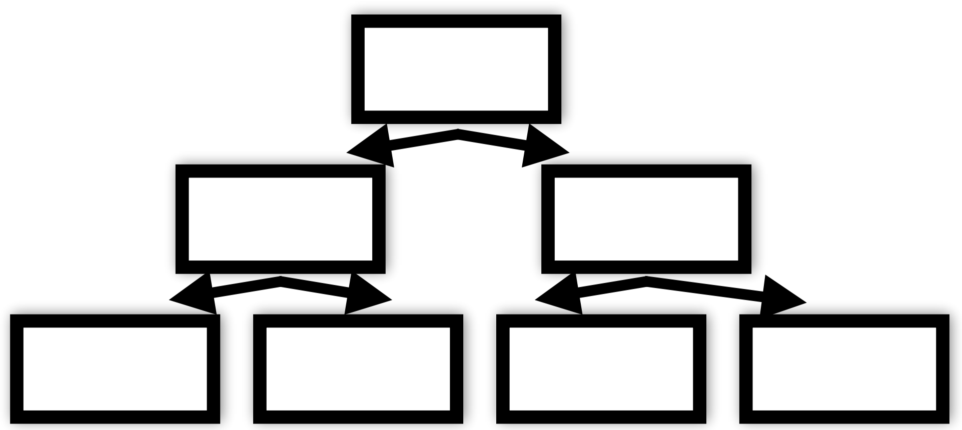


INFERNO: Castro, Dorigo

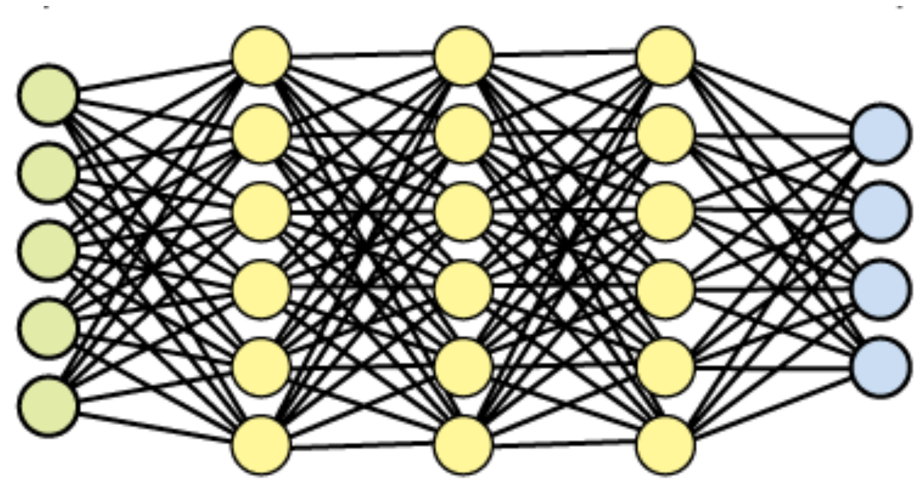


Which is the best ML solution?

$$Z_0 = \sqrt{2\left[(SVI + B2)\ln\left(1 + \frac{SVI - V}{V + B2}\right) - (SVI - V)\right]}$$



Decision Tree like categoriser to optimise $\sum_{categories} {Z_0}^2$ instead of gini



Not trying to predict class labels Y

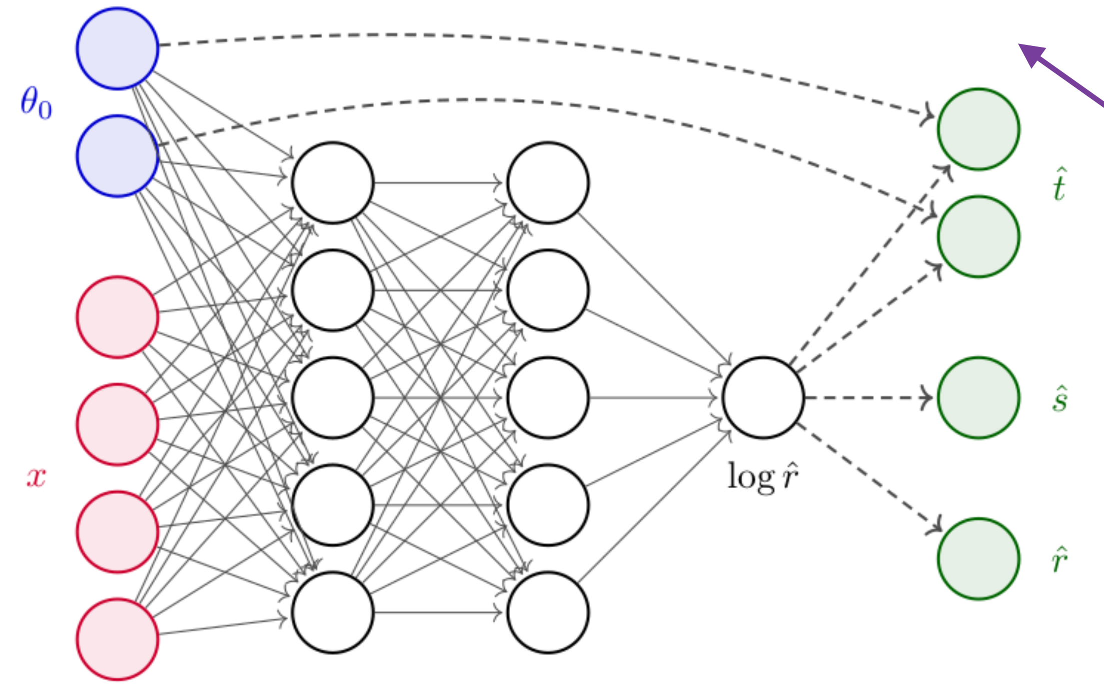
NN to categorise with batch level loss: $\sum_{categories} {Z_0}^2$

Simultaneous fit on all categories for Likelihood

Simple classifier(s) on :
S vs SVI + B2
or
SVI + B2 vs V + B2 …

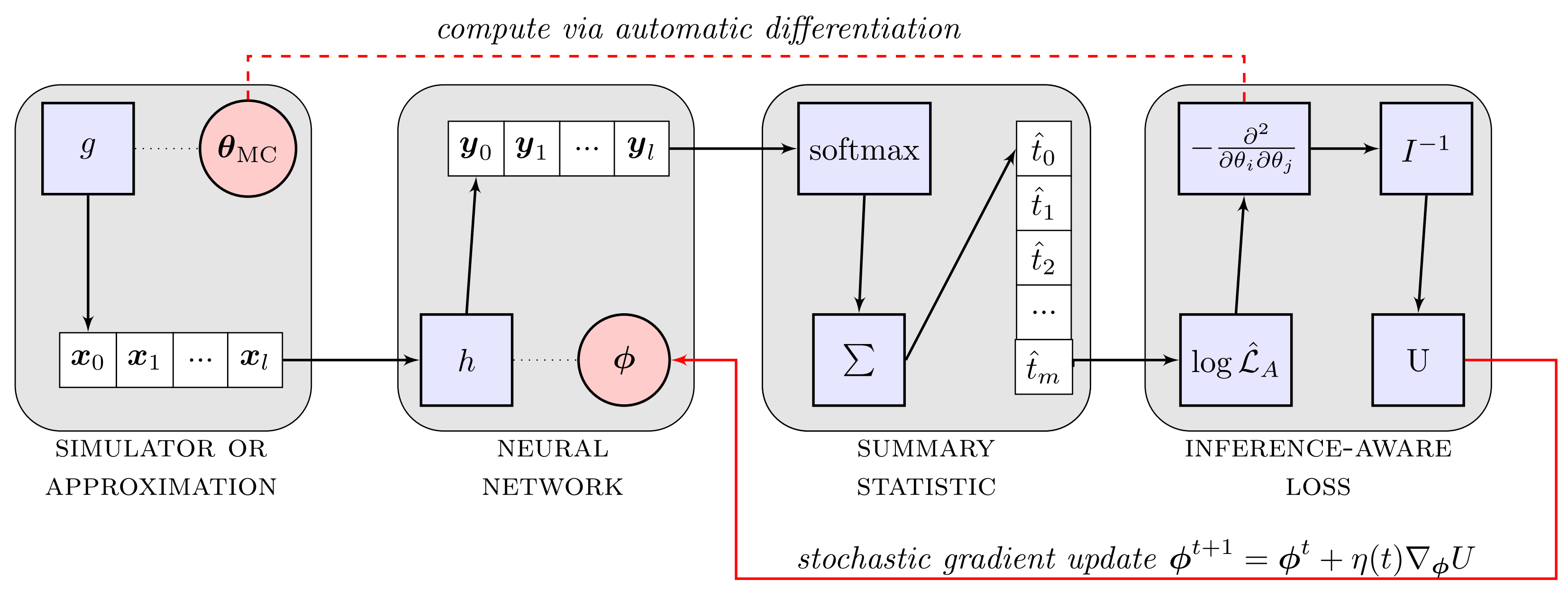
Combine with Reparameterisation + Template method as in ATLAS A->tt analysis, and Noam Tal Hod ?

Likelihood-free inference: Brehmer, Cranmer, Louppe, Pavez

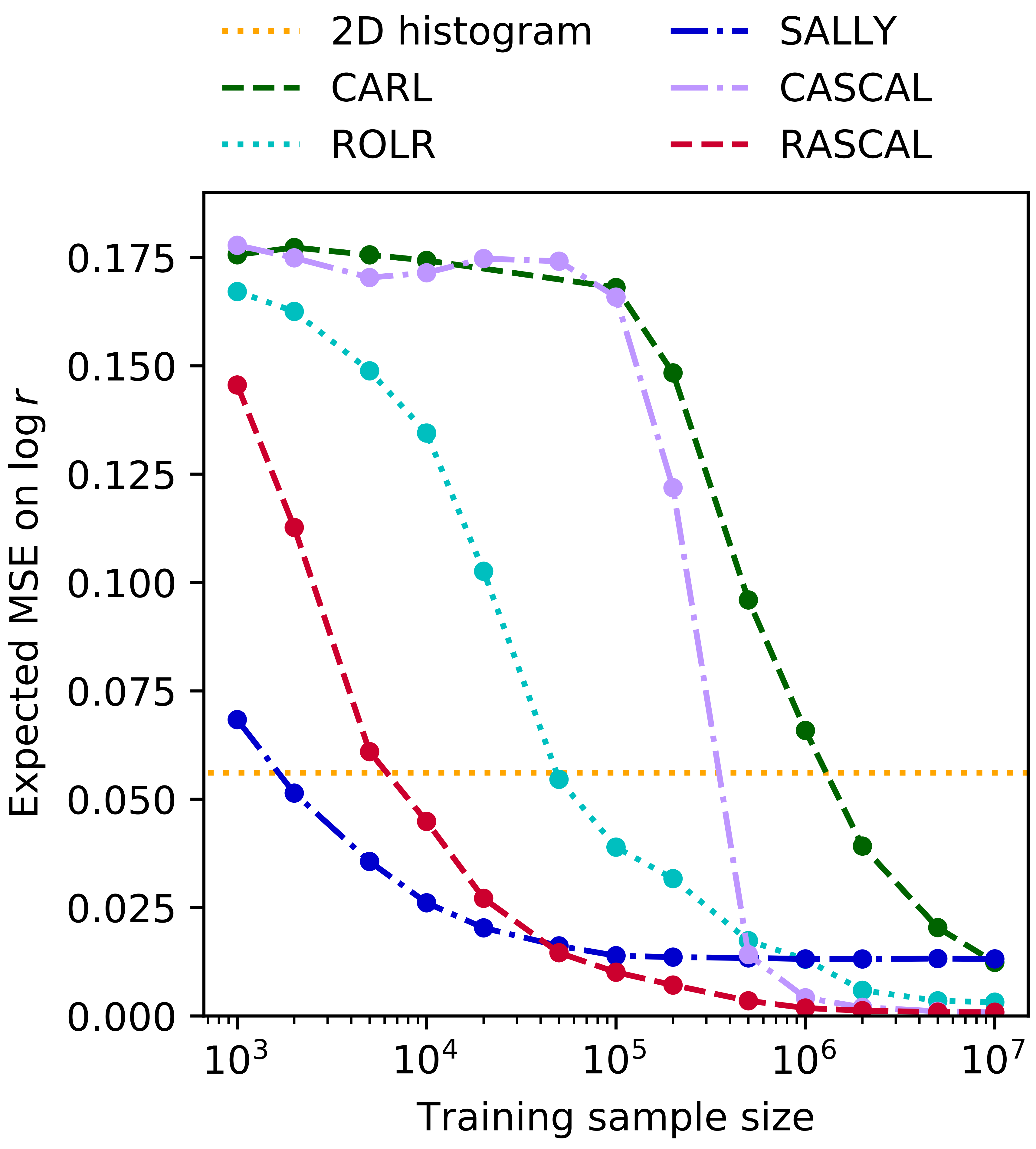


Can be trained on different values of μ, but not easy to implement in ATLAS ('mining gold' from simulator)

INFERNO: Castro, Dorigo



Madminer Techniques

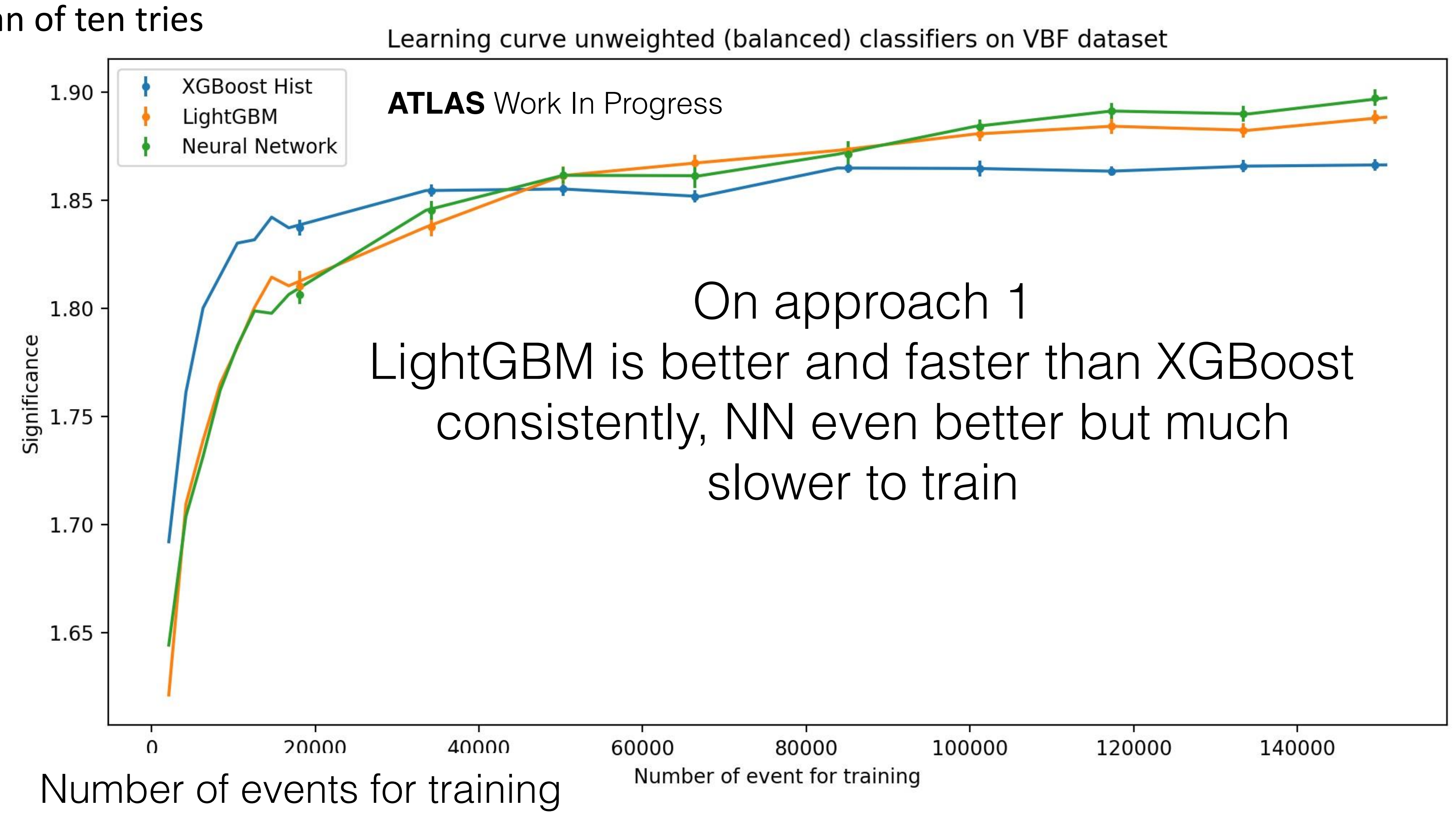


Thanks to Jeremy Couthures

Results (1/2)

Approach 1: VBF_SVI vs rest, Approach 2: VBF_Higgs_S_s_channel vs Rest

Each point: mean of ten tries

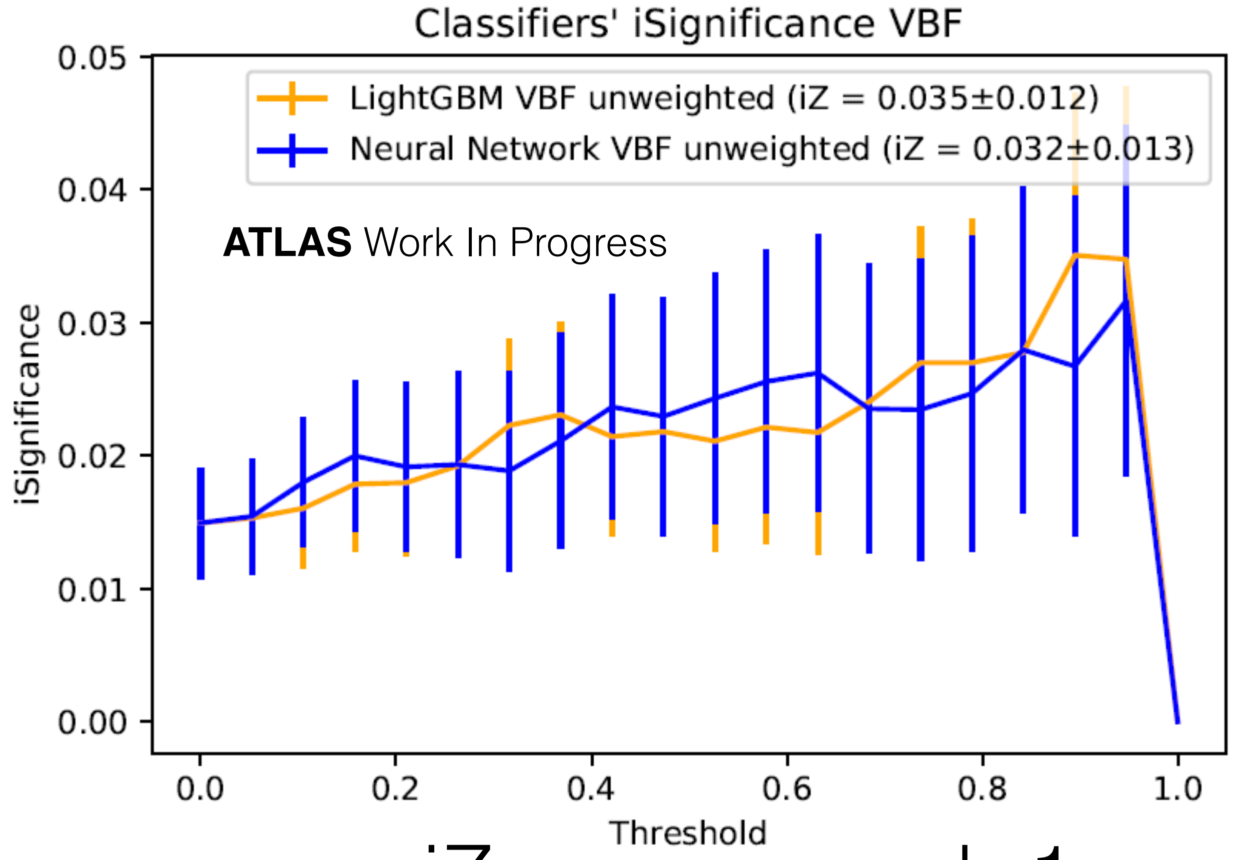


Traditional Significance

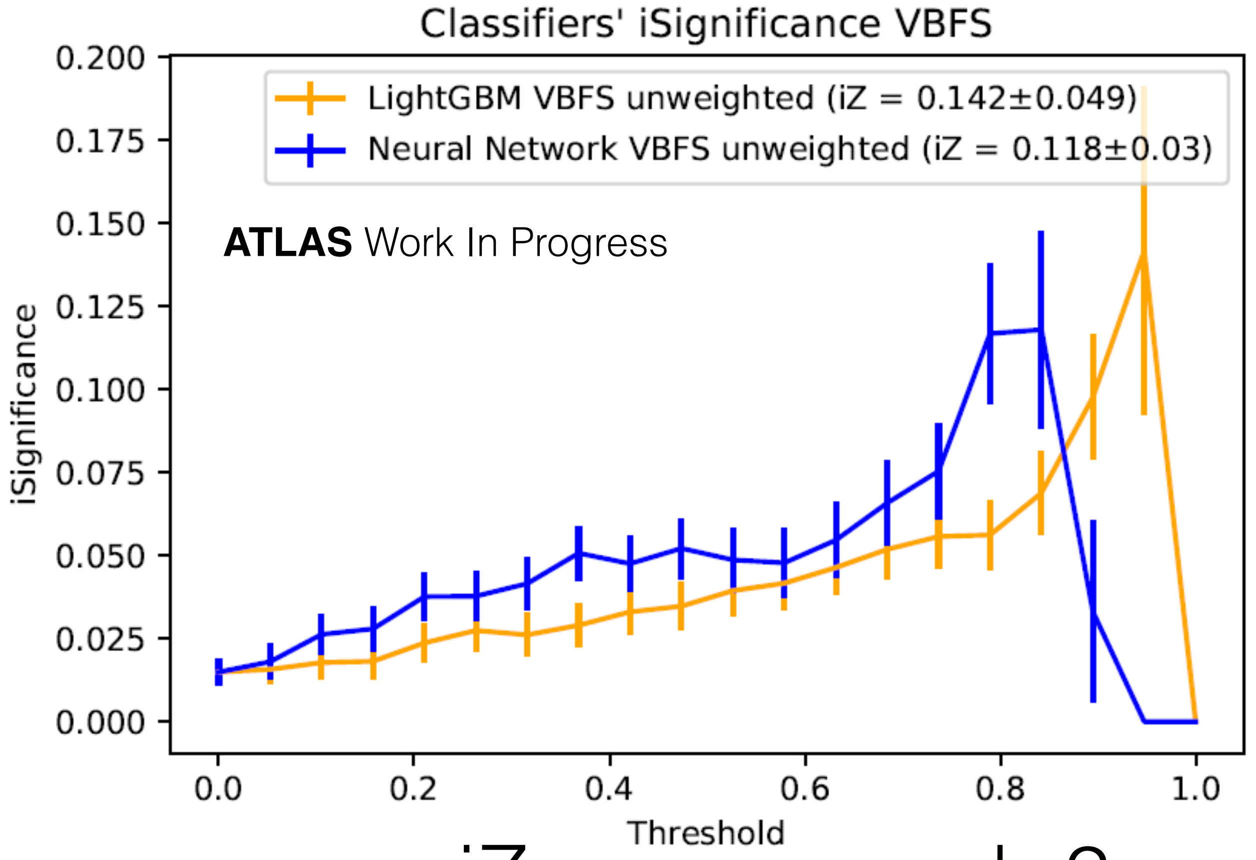
On approach 1
LightGBM is better and faster than XGBoost consistently, NN even better but much slower to train

Number of events for training

LightGBM outperform XGBoost for big datasets



iZ on approach 1



iZ on approach 2

iZ very unstable and not very reproducible compared to significance curves, highly sensitive to qqZZ negative weighted events

But approach 2 is better for iZ than approach 1 consistently

Significance with interference:

$$iZ = \frac{S + SBI - B}{2*\sqrt{SBI + B\_gg\_qq}}$$

S: VBF_s, SBI:VBF
B: VBS, B_gg_qq alias B2: gg+qq

$$\frac{\Delta iZ}{iZ} = \left|\frac{\Delta S + \Delta B}{S + SBI - B}\right| + \left|\frac{\Delta B2}{2 \times (SBI + B2)}\right| + \Delta SBI \left|\frac{1}{S + SBI - B} - \frac{1}{2 \times (SBI + B2)}\right|$$

Results (2/2)

Approach 1: VBF_SVI vs rest, Approach 2: VBF_Higgs_S_s_channel vs Rest

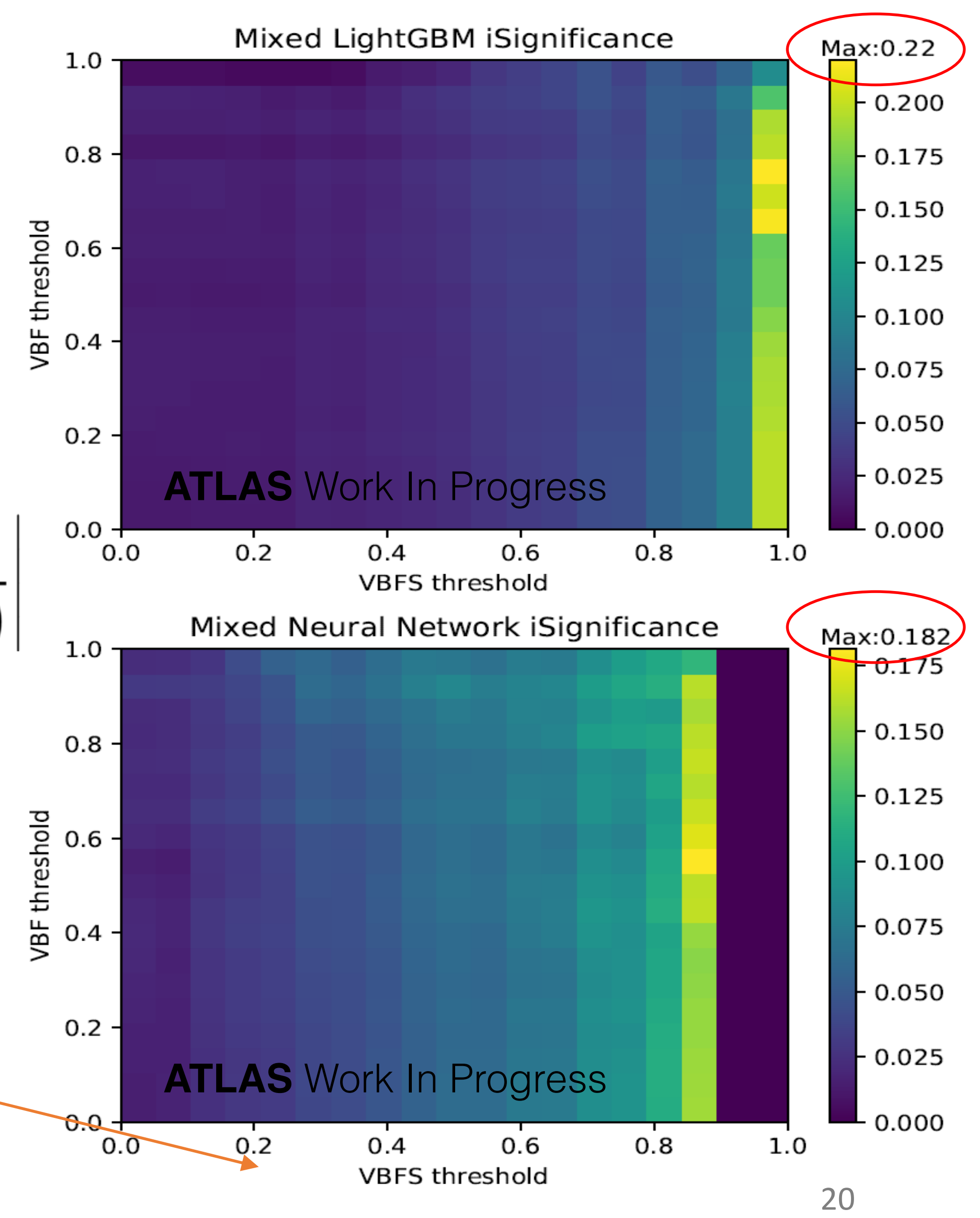
Significance with interference:

$$iZ = \frac{S + SBI - B}{2 * \sqrt{SBI + B\_gg\_qq}}$$

S: VBF_s, SBI:VBF
B: VBS, B_gg_qq alias B2: gg+qq

$$\frac{\Delta iZ}{iZ} = \left|\frac{\Delta S + \Delta B}{S + SBI - B}\right| + \left|\frac{\Delta B2}{2 \times (SBI + B2)}\right| + \Delta SBI \left|\frac{1}{S + SBI - B} - \frac{1}{2 \times (SBI + B2)}\right|$$

- LightGBM trained on VBF_s dataset: iZ: 0.197, itreshold: 0.947
- LightGBM trained on VBF dataset: iZ: 0.021, itreshold: 0.895
  2 Dimensions: iZ 0.220
- NN trained on VBF_s dataset: iZ: 0.155, itreshold: 0.842
- NN trained on VBF dataset: iZ: 0.030, itreshold: 0.895
  2 Dimensions: iZ 0.182



We also tried:

- Train 4 classifiers, multi-class classifiers, multi-label classifiers with 4 thresholds: Some improvement but much more complex
- "SM vs SM_without_Higgs": Slightly better than approach 2
- Remove 2 jet cut: much more statistics, similar final results
- Tried directly optimising for iZ with a custom designed DT, NN with some success, but optimal only for neighbourhood of mu=1

Talked to others in ATLAS who have dealt with interference but not perfect solution for our case:

- ATLAS A->tt analysis, and Noam Tal Hod thesis for reparameterisation
- ATLAS Interference in top

Combine approach 1 and 2: Slight gain in iZ but tendency to cut too hard if we have fine binning => low statistics
Finds region with negative weighted qqZZ unless we set iZ = 0 for such regions

Results from 1. VBF-SBI vs rest, 2. VBF-Higgs-S-Channel vs Rest (2/2)

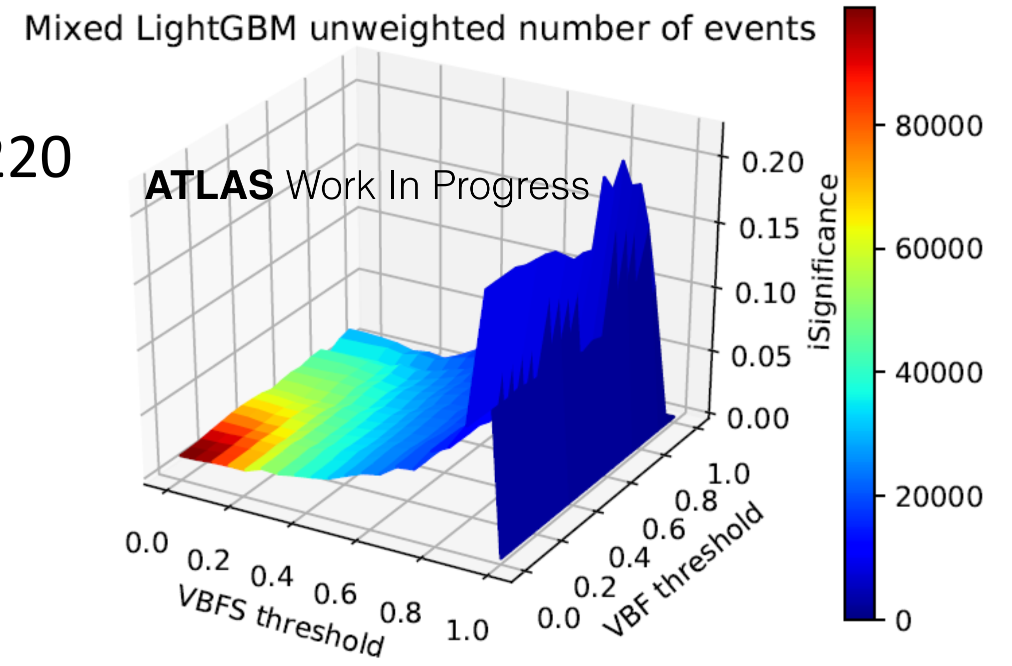
Color: number of events

Higher value might be due to a lack of statisctics

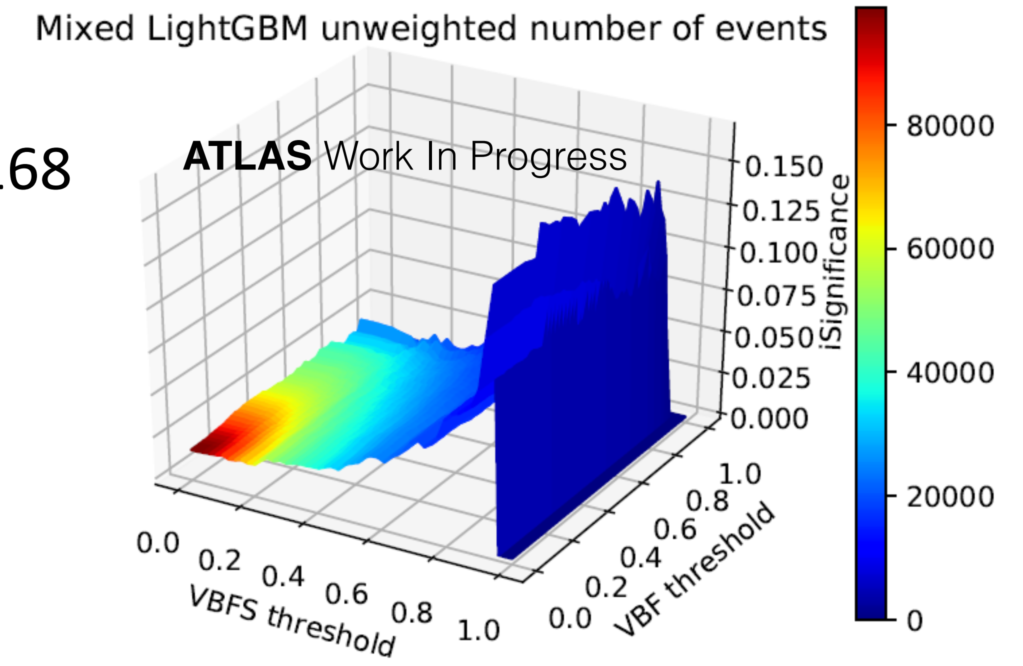
For 20 bins:

For 50 bins:

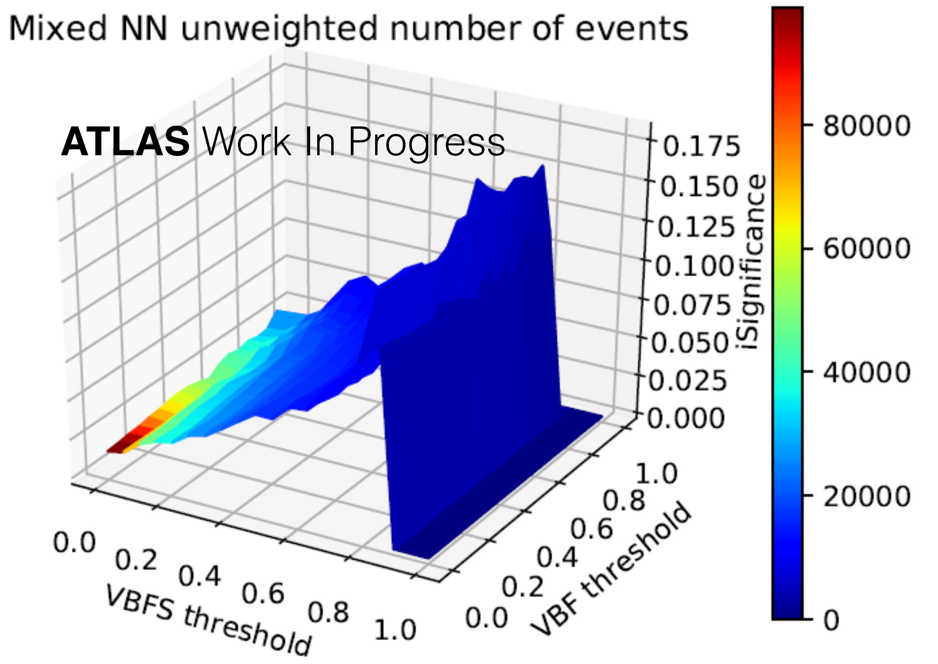
Max iZ:0.220



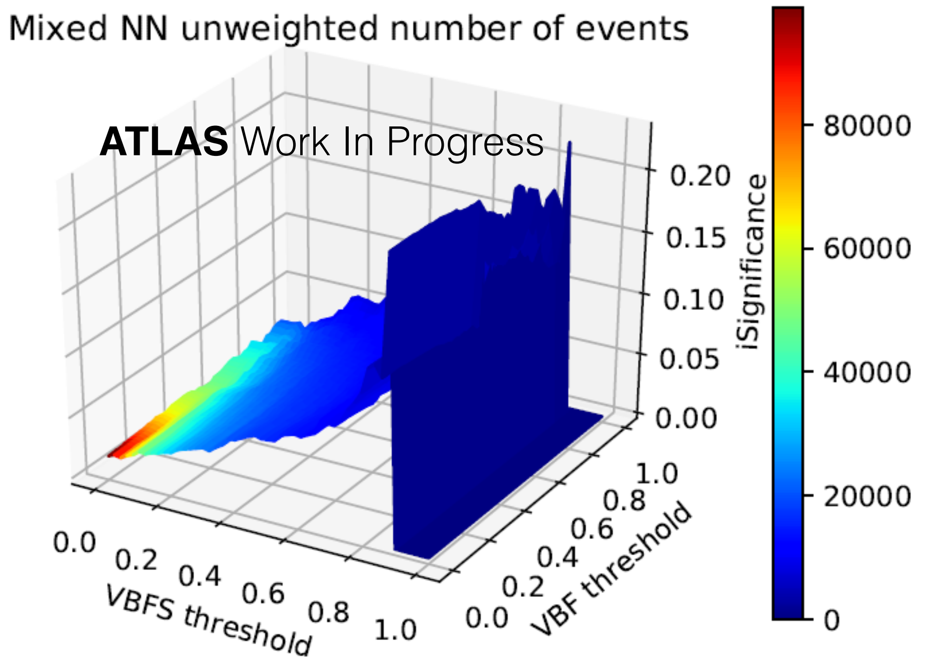
Max iZ:0.168



Max iZ:0.182

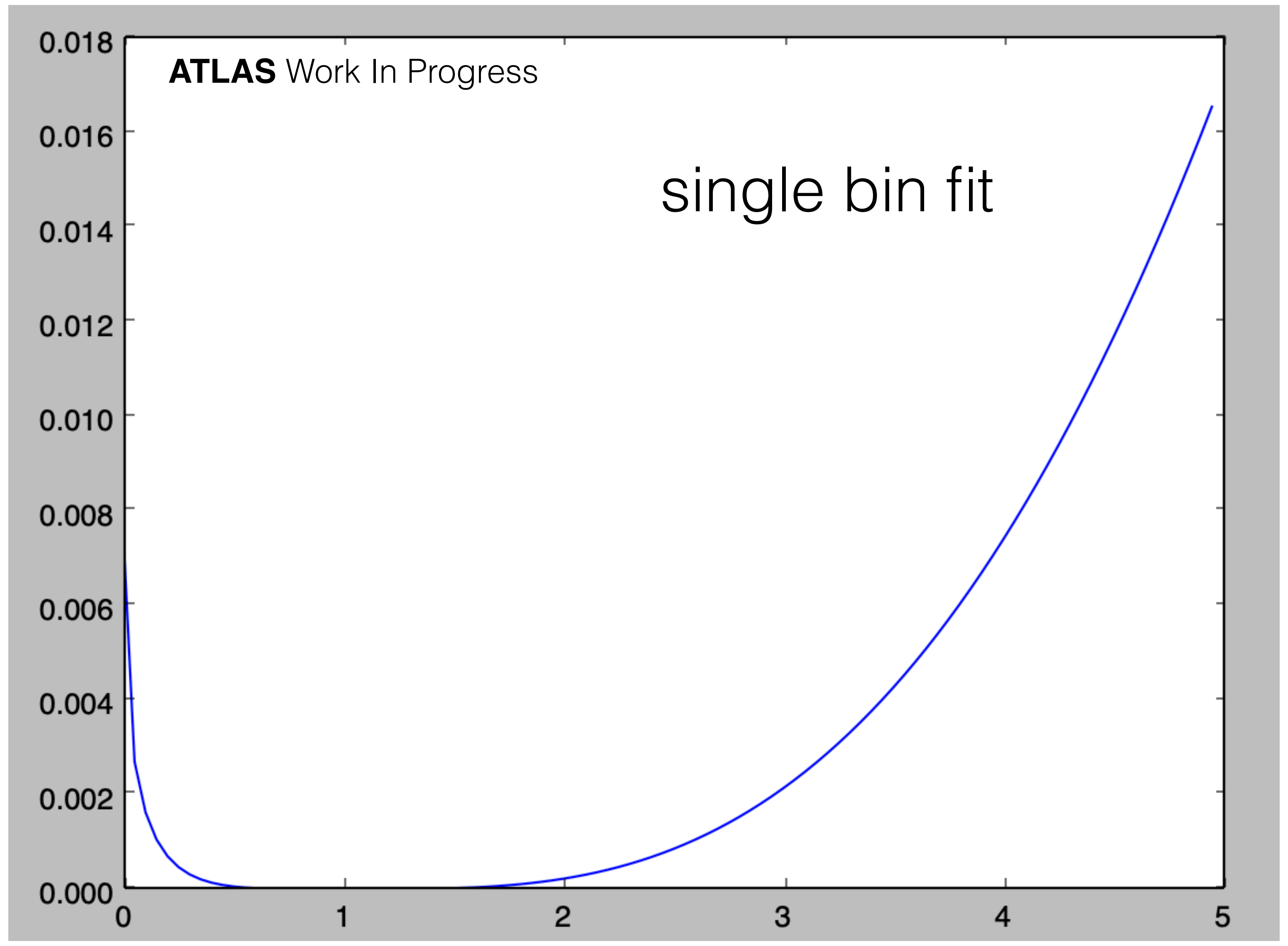


Max iZ:0.232



Tendency to cut too hard if we have fine binning => low statistics

VBF nll Curve



Previous Efforts in ATLAS for Interference

https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PAPERS/TOPQ-2017-05/

https://arxiv.org/pdf/1707.06025.pdf

Not directly useful for our case: To try to improve sensitivity to Higgs Signal Strength with selections / ML