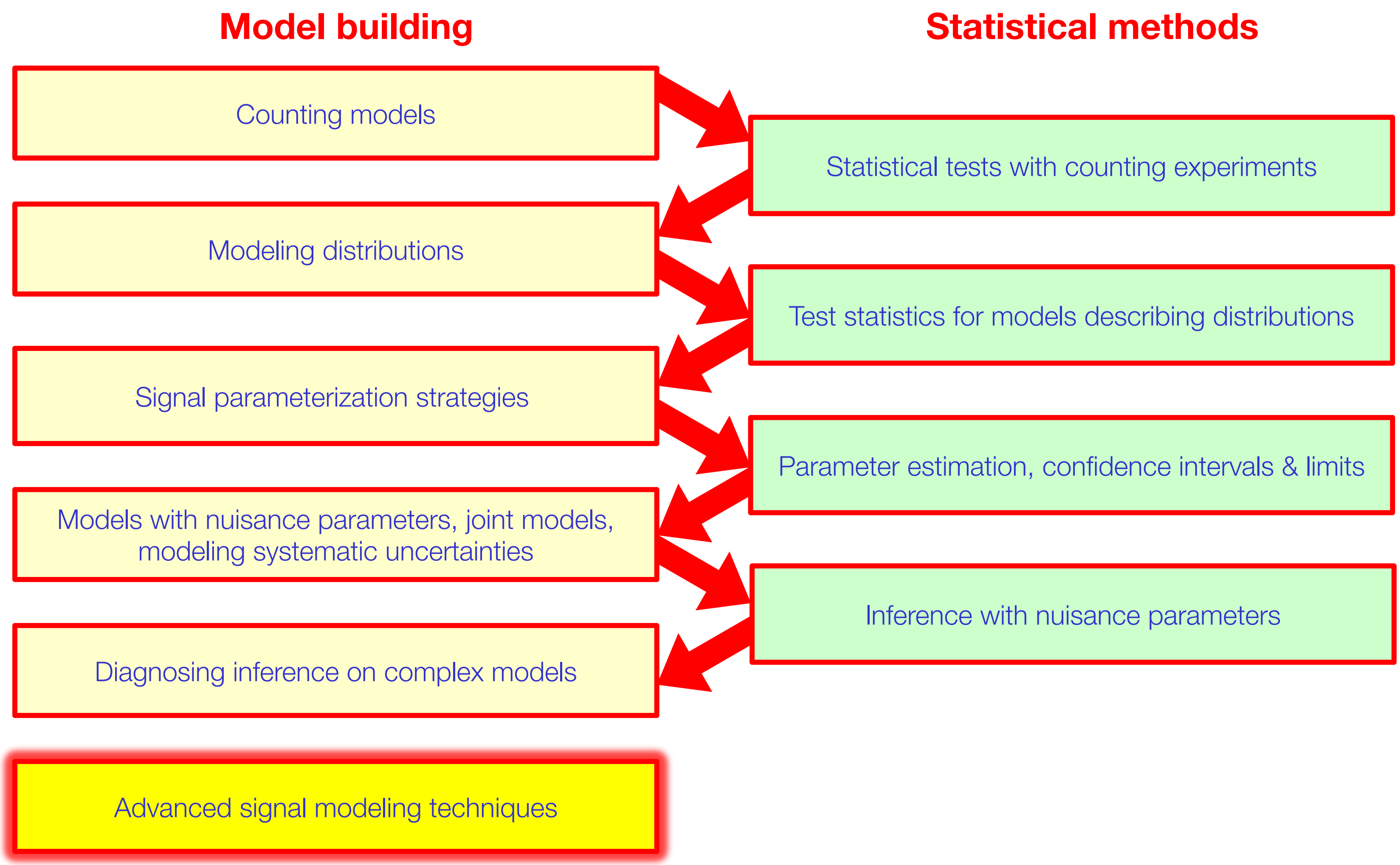
Model building 6

Advanced signal modeling:
convolutions, matrix element methods, amplitude-based morphing models

Wouter Verkerke, NIKHEF

Roadmap of this course

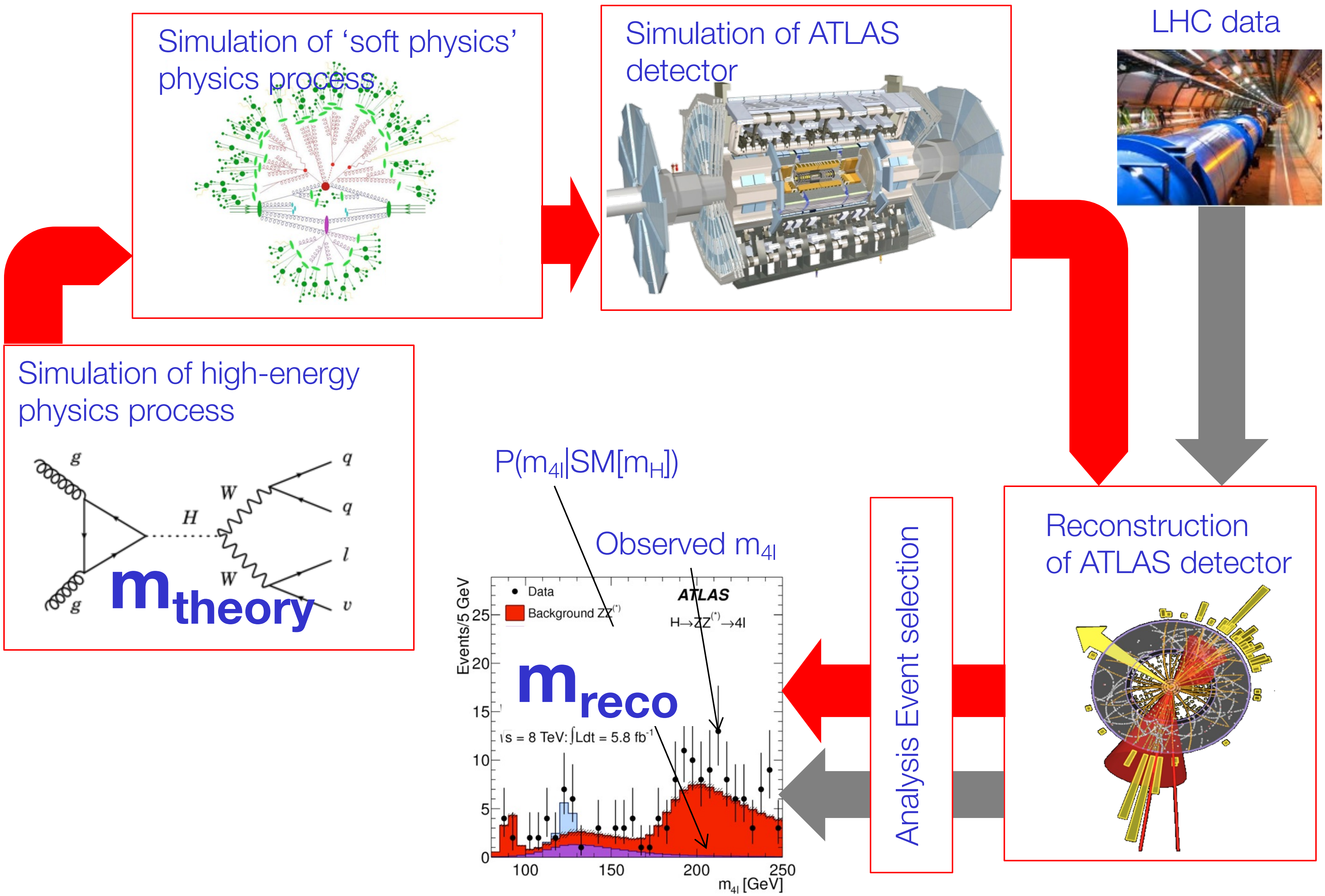
- Start with basics, gradually build up to complexity



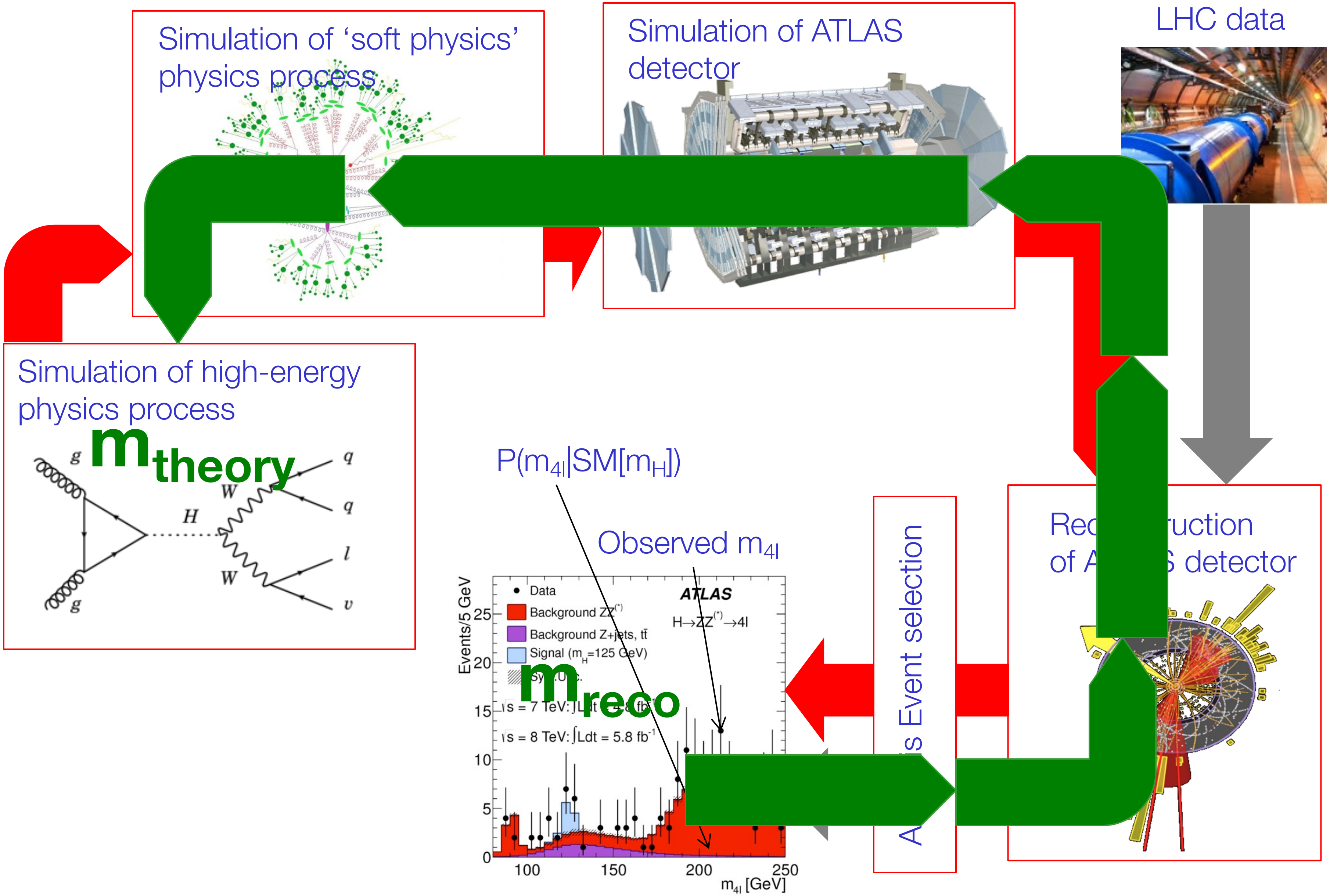
Goal of *measurement* – inference on a theory parameter

- So far – a lot of the material discussed modeling of background, general modeling uncertainties, and extract of a signal (discovery or limit setting)
- For established signals, the main goal is usually to measure as precisely as possible the value of parameter of the model, which connects to an underlying physics theory.
- In many cases, measured model parameters (e.g. mean of invariant mass) don't map *exactly* to underlying theory parameters (pole mass) because of smearing and bias effects in the detector.
- In this section we cover some examples of techniques to infer the underlying theory parameters

The great smearing machine



Can we invert this proces?



The detector as convolution

- The effect of the detector (and analysis machinery) on the theory parameter can be expressed as a *convolution*

Theory distribution

$$f(x_{reco}) = \int f(x_{theo}) K(x_{reco}, x_{theo}) dx_{theo}$$

Reconstructed distribution

Kernel function
(resolution / migration)

The detector as convolution

- For a perfect detector the Kernel K is delta function

$$f(x_{reco}) = \int f(x_{theo})\delta(x_{reco} - x_{theo})dx_{theo}$$

- But if detector response function to x_{theo} is (to good approximation) independent of the value x_{theo} over a wide enough range, can then also represent problem as

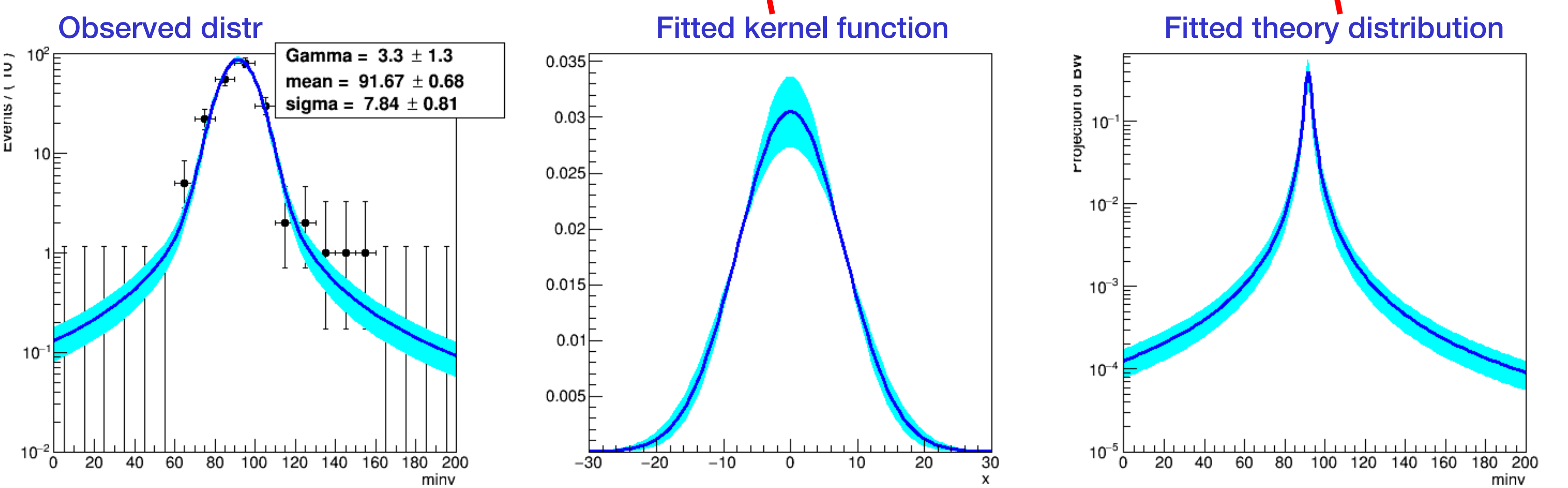
$$f(x_{reco}) = \int f(x_{theo})R(x_{reco} - x_{theo})dx_{theo}$$

- Here R is a resolution model, typically a Gaussian (or sum of Gaussians)

Explicit modeling of the detector convolution

- Example: reconstruction of a particle theory mass m_{th}, from a reconstructed invariant mass m_{inv}

$$f(m_{inv} \mid M, \Gamma, b, \sigma) = \int Gaussian(m_{inv}^{th} - m_{inv}^{reco}, b, \sigma) \cdot BreitWigner(m_{inv}^{th} \mid \Gamma, M) dm_{inv}^{th}$$



- Note: probability model for observed m_{inv} now directly describes parameter M of underlying theory
  - Can also introduce parameters of resolution model as fittable model parameters
  - But not always sensitivity (e.g. bias *b* is inseparable from *M* in above model)

'ex18.C'

FFT Convolutions in RooFit

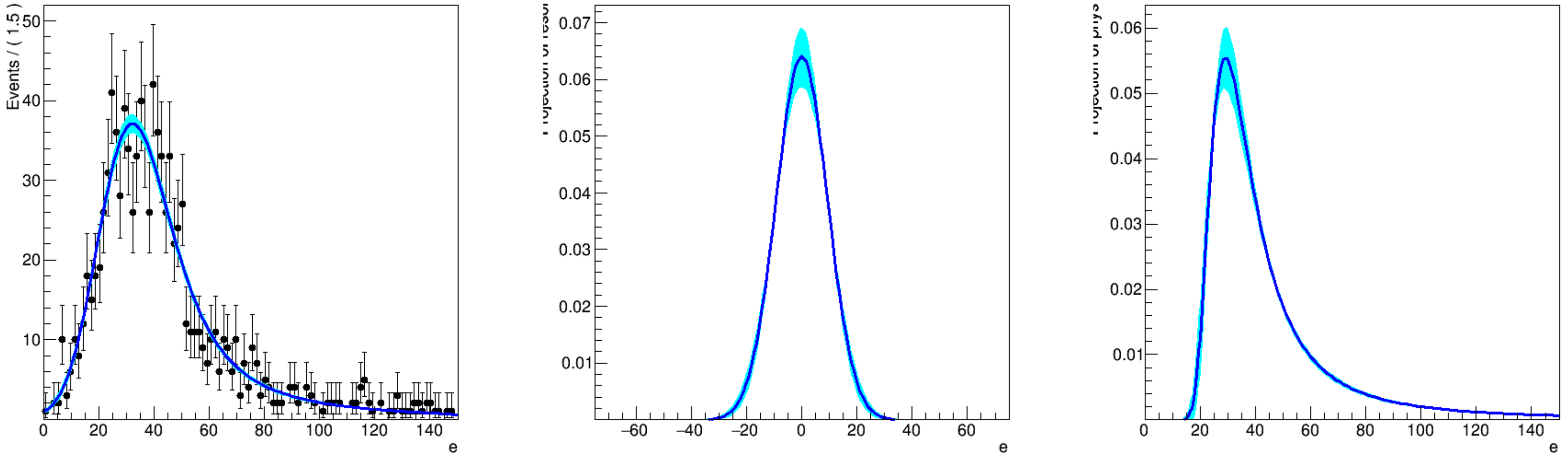
- Fourier convolution implement in FCONV operator

```
RooWorkspace w("w") ;
w.factory("Landau::phys(e[0,150],mean[30,0,60],sigma[5,1,10])") ;
w.factory("Gaussian::resol(e,0,sigma_gauss[10,0.1,20])") ;
w.factory("FCONV::conv(e,phys,resol)") ;

RooDataSet* d = w.pdf("conv")->generate(*w.var("e"),1000) ;

RooFitResult* r = w.pdf("conv")->fitTo(*d,Save()) ;
```

Observed distribution | Fitted kernel function | Fitted theory distribution



CPU time of fit = 400msec (1000 events, 53 likelihood evaluations)

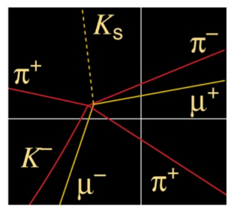
A more ambitious example – per-event errors

- In some cases kernel function (resolution model) depends on other observables y, but not on x → Can also model try to model that.
- Example per-event errors

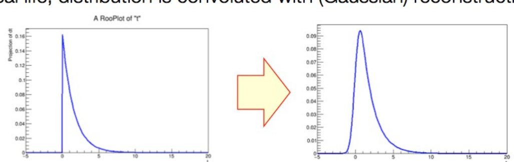
Resolution kernel depends on 2nd observable δt

$$f(t\,|\,\delta t) = \int f(t_{theo}) \cdot Gaussian(t - t_{theo}, b, \sigma \cdot \delta t)\, dt_{theo}$$

Case study – per-event errors



- Another common variant of this type of modeling problem is the so-called 'per-event' error
- Example: observable = decay time distribution, measured from reconstructed vertex.
  - In absence of a detector resolution, exponential decay distribution
  - In real life, distribution is convoluted with (Gaussian) reconstruction resolution



- But vertex reconstruction gives also estimate of uncertainty for every reconstructed vertex → the 'per-event error'
  - Can take this into account: well-reconstructed events carry more information
- How? Scale assumed resolution with per-event error

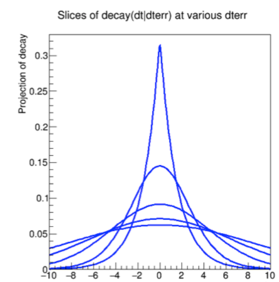
$$f(t\,|\,\delta t) = Decay(t) \otimes Gaussian(t, 0, \sigma \cdot \delta t)$$

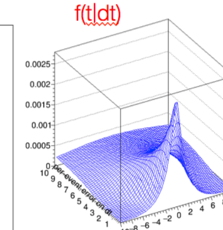
Case study – per-event errors

- Visualization of decay function with variable resolution

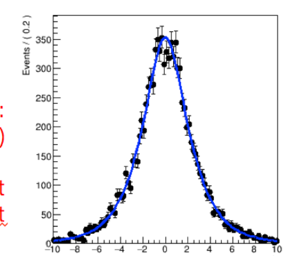
Decay function (symmetrized) convoluted with Gaussian resolution at 4 different values of per-event error

$$f(t\,|\,\delta t) = Decay(t) \otimes Gaussian(t, 0, \sigma \cdot \delta t)$$





Gain: high-resolution events carrymore weight in likelihood → better estimate of model parameters



Full 2D-model: $F(t,dt) = F_1(t|dt) * F_2(dt)$

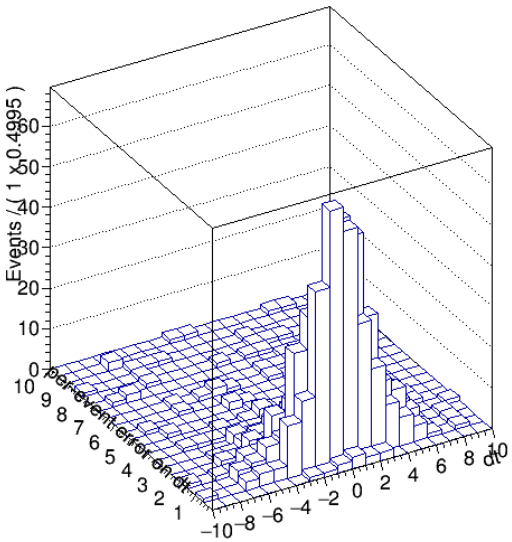
Shown here: *projection* on t
$F(t) = \text{Int}\,[\,F_1(t|dt) * F_2(dt)\,]\,dt$

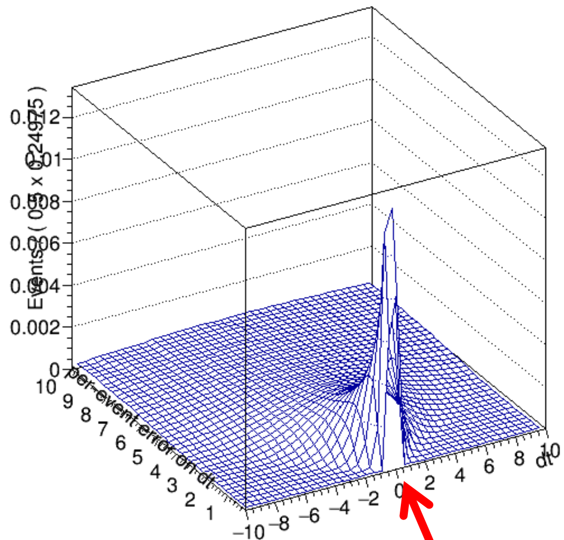
A more ambitious example – per-event errors

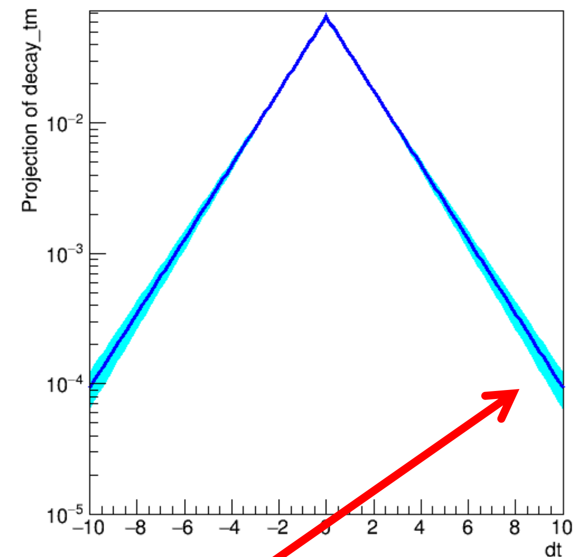
- Example fit with resolution kernel using per-event errors

$$f(t\,|\,\delta t) = \int Gaussian(t - t_{theo}, b, \sigma \cdot \delta t) \cdot f(t_{theo}) dt_{theo}$$

Observed distribution **Fitted kernel function** **Fitted theory distribution**







bias = 0.073 ± 0.055
sigma = 0.994 ± 0.083
tau = 1.525 ± 0.085

- For this physics example can fit all kernel parameters (b,σ) in addition to theory parameter τ

But it quickly gets very complicated

- Example 1: Convolution kernel constant in x_{reco}

$$f(x_{reco}) = \int Gaussian(x_{reco} - x_{theo}) \cdot f(x_{theo})\, dx_{theo}$$

- Example 2: Convolution kernel constant in x_{reco}, but depends on y_{reco}

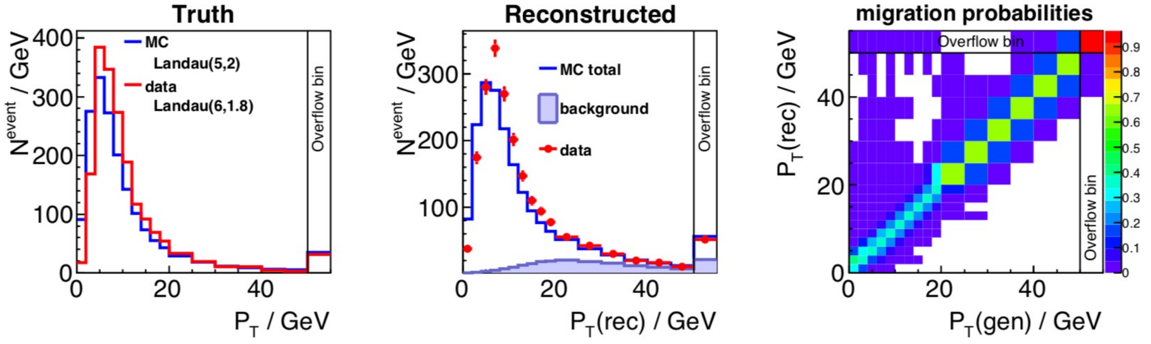
$$f(t \mid \delta t) = \int Gaussian(t - t_{theo}, b, \sigma \cdot \delta t) \cdot f(t_{theo}) dt_{theo}$$

- What happens if resolution also varies in x_{reco}?
  - E.g. mass resolution depends on mass..
  - Then in very quickly becomes numerically very complex (hard-to-solve)
  - Numeric precision issues due to possible degeneracies.

- Solution in that case is discretize model for x_{reco}
  in which case convolution *kernel $K(x_{reco}, x_{theo})$ becomes a matrix*
  - *Can no longer fit, but perform an unfolding procedure to obtain $f(x_{theo})$*

Unfolding – basic idea

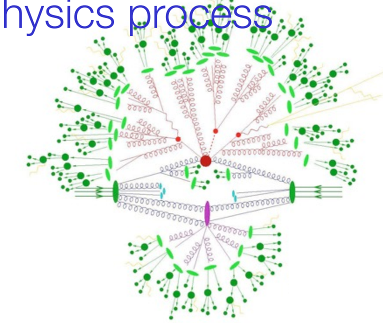
- Unfolding basic idea – from simulation you know for each event both x_{reco} and x_{theo} → Use this to populate a **response matrix K**



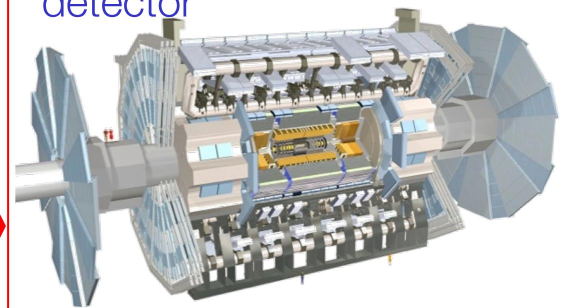
- Essence of unfolding: $f(x_{reco}) = \int f(x_{theo}) K(x_{reco}, x_{theo}) dx_{theo}$ → $\mathbf{f(X_{theo}) = K^{-1} f(X_{reco})}$
- Easy in concept, but difficult in practice
  → response inversion is numerically unstable
- Solutions to stabilize exist ('regularization') but invariably trade improved stability ('reduced variance') for bias ('systematic effects')
- Traditional solutions largely restricted to 1D histograms for this reason, but many new developments in recent years, driven by ML/AI
- Unfolding is a large and complex topic → See dedicated lecture by Lydia

The experiment as convolution – one step back…

Simulation of 'soft physics' physics process

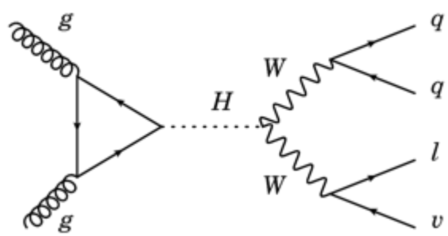


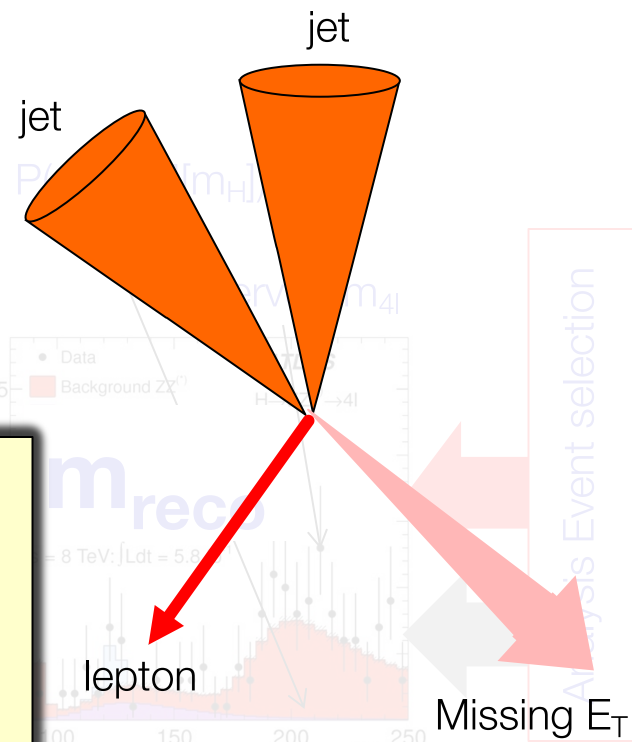
Simulation of ATLAS detector



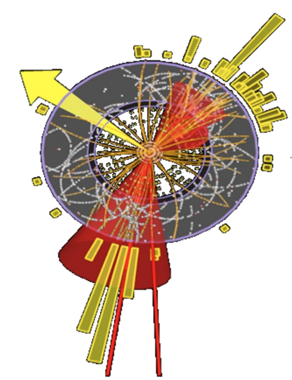
LHC data

Simulation of high-energy physics process





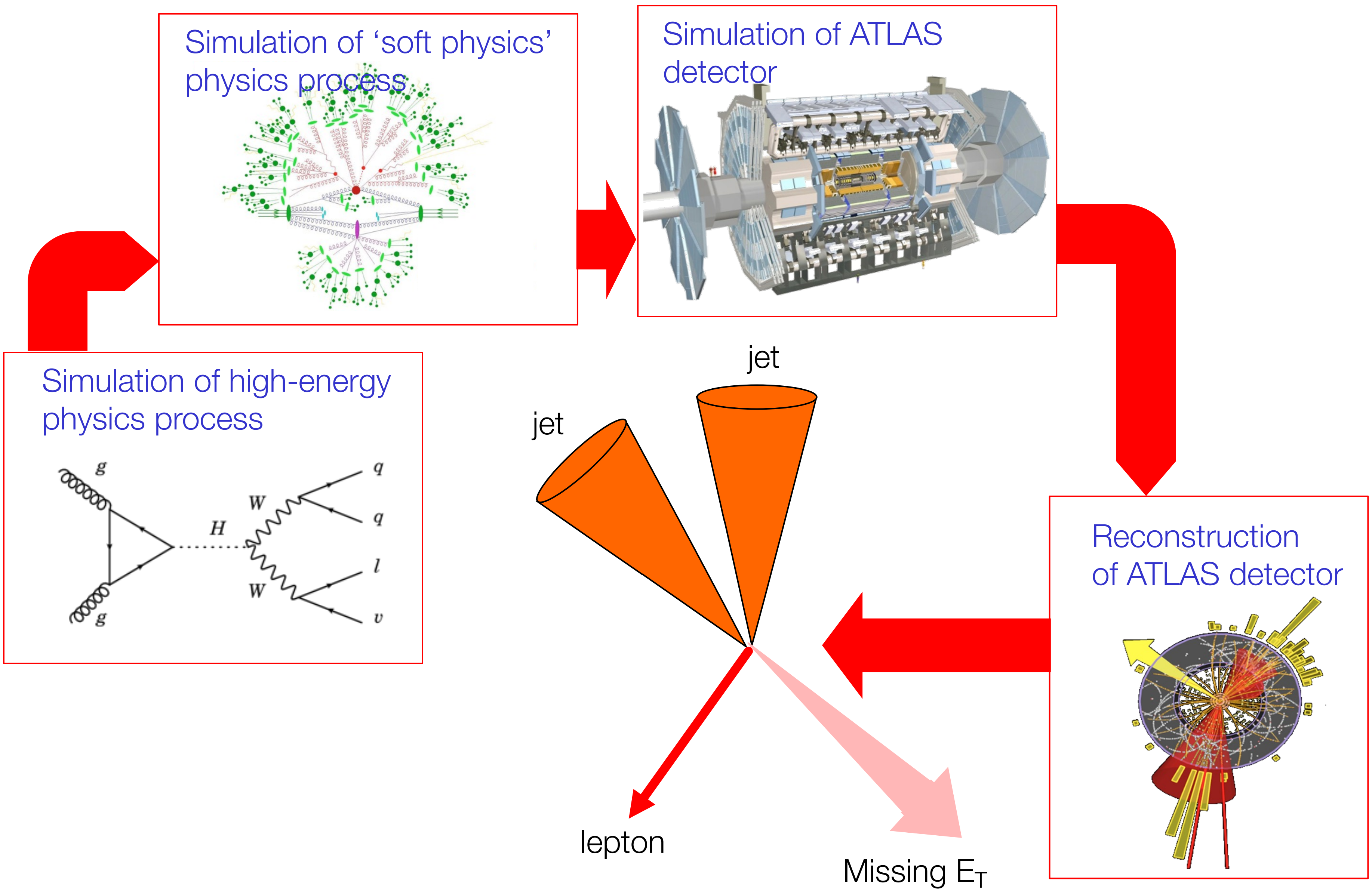
Reconstruction of ATLAS detector



What if we scale back our ambition one notch formulate convolution and take **4-vectors of reco objects as observables** in convolution?

$$f(x_{reco}) = \int f(x_{theo}) K(x_{reco}, x_{theo}) dx_{theo}$$

The experiment as convolution – one step back…



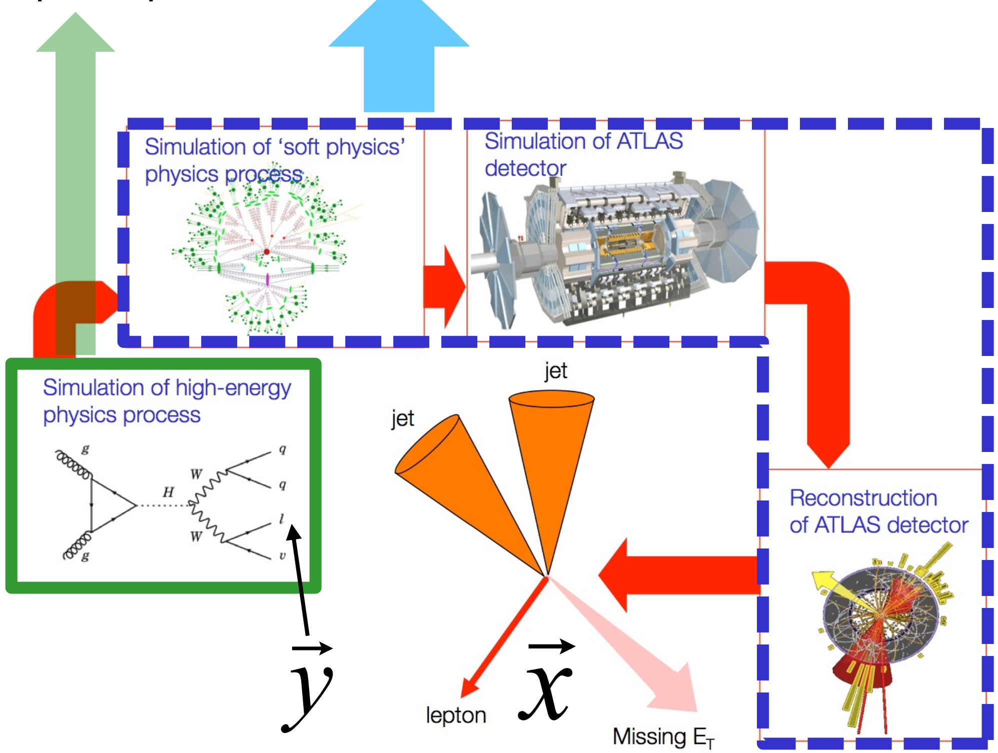
The detector as convolution

- Then convolution can be written in terms of parton kinematics

$$f(x_{reco}) = \int f(x_{theo}) K(x_{reco}, x_{theo}) dx_{theo}$$

phase-space factor

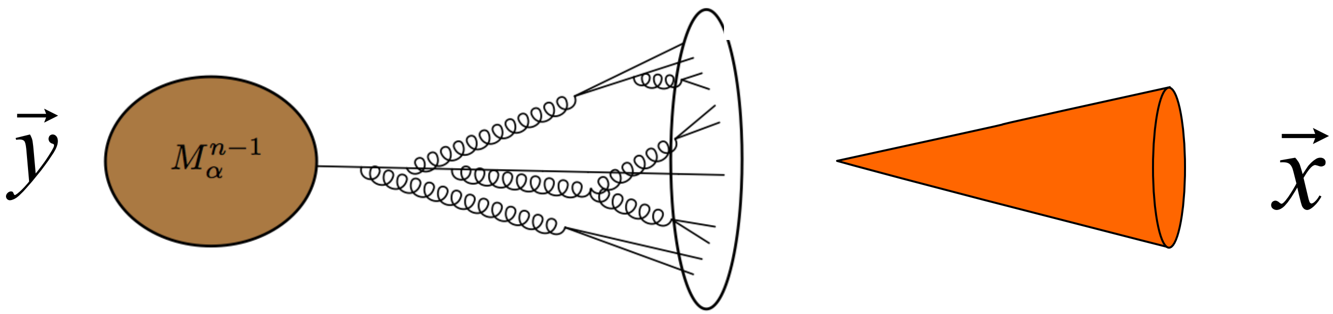
$$f_\alpha(\vec{x}) = \frac{1}{\sigma} \int \left| M_\alpha \right|^2 (\vec{y}) \cdot W(\vec{x}, \vec{y}) \cdot d\varphi(\vec{y})$$



The detector as convolution – matrix element methods

- The transfer function W(x,y) maps *parton kinematics* onto *reco kinematics*

$$f_\alpha(\vec{x}) = \frac{1}{\sigma}\int \left|M_\alpha\right|^2 (\vec{y})\cdot W(\vec{x},\vec{y})\cdot d\varphi(\vec{y})$$



- The transfer function factorizes by parton. Usually it is further approximated to also kinematically factorize in terms for E,ϕ,y

$$W(\vec{x},\vec{y}) = \prod_{partons} W_i(\vec{x}_i,\vec{y}_i) = \prod_{partons} W_i^E(E_i^{reco},E_i^{part})\cdot W_i^E(\varphi_i^{reco},\varphi_i^{part})\cdot W_i^E(y_i^{reco},y_i^{part})$$

typically Gaussian models

The detector as convolution – matrix element methods

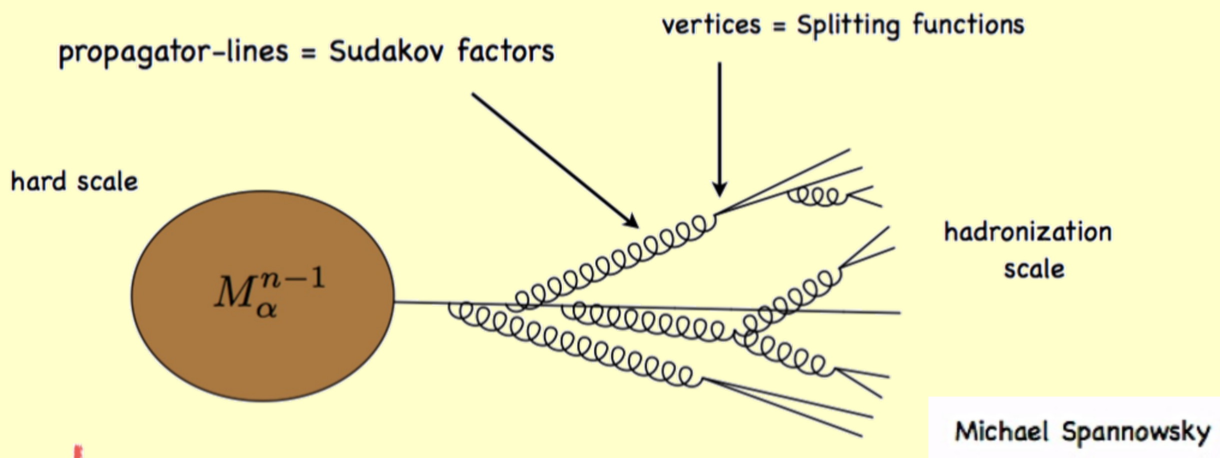
- The transfer function W(x,y) maps *parton kinematics* onto *reco kinematics*

$$f_\alpha(\vec{x}) = \frac{1}{\sigma}\int \left|M_\alpha\right|^2(\vec{y})\cdot W(\vec{x},\vec{y})\cdot d\varphi(\vec{y})$$

The hard part of W(x,y) is dealing with gluon splittings
(makes mapping of partons to truth jets fuzzy)

Factorization of emissions in soft/collinear limit

and Sudakov factors allow semiclassical approximation of quantum process:



→ Can calculate weight for shower history iteratively
→ Can use smaller objects and more objects (more information)

- y

$, y_i^{part})$

typically Gaussian models

What can you do MEM models?

- The output of the Matrix Element Method is a probability model for events of a fixed reco-level topology (e.g. 2 jets, 1 lepton, MET) under a physics process hypothesis (e.g. pp→H→WW→qqlv)
- With MEM models for multiple hypothesis, can do hypothesis testing (event selection) in an Neyman-Pearson optimal way

The Neyman-Pearson lemma

- In 1932-1938 Neyman and Pearson developed a theory in which one must consider competing hypotheses
  - Null hypothesis (H_0) = Background only
  - Alternate hypotheses (H_1) = e.g. Signal + Background

and proved that

- The region W that minimizes the rate of the type-II error (not reporting true discovery) is a contour of the Likelihood Ratio

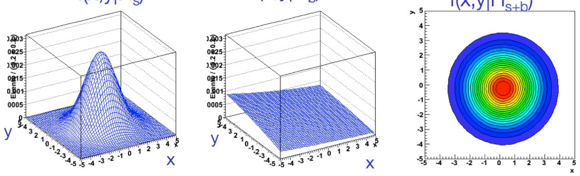
$$\frac{P(x|H_1)}{P(x|H_0)} > k_\alpha$$

- Any other region of the same size will have less power

Wouter Verkerke,

The Neyman-Pearson lemma

- Example of application of NP-lemma with two observables



- Cut-off value c controls type-I error rate ('size' = bkg rate)
  Neyman-Pearson: LR cut gives best possible 'power' = signal eff.
- So why don't we *always* do this? (instead of training neural networks, boosted decision trees etc)

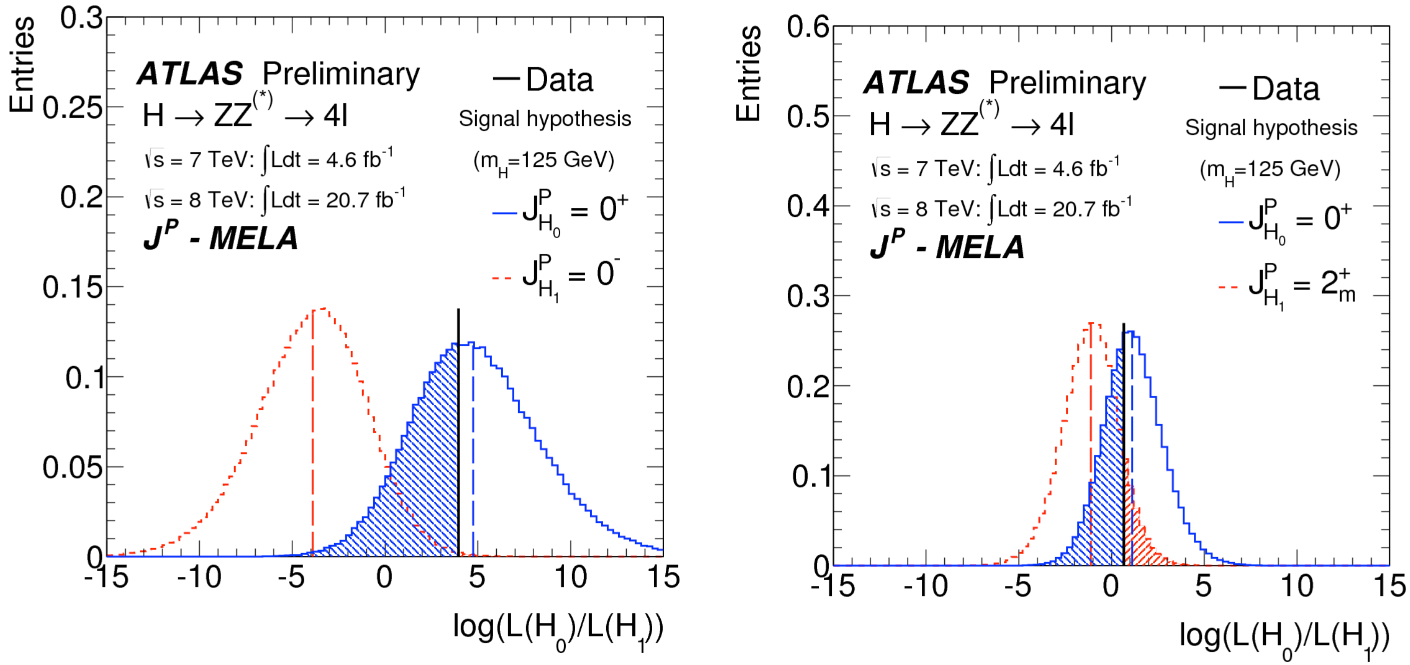
The problem is that we usually don't have explicit formulae for the pdfs $f(\vec{x}|\mathrm{s})$, $f(\vec{x}|\mathrm{b})$.

But with MEM we do!

Wouter Verkerke, NIKHEF

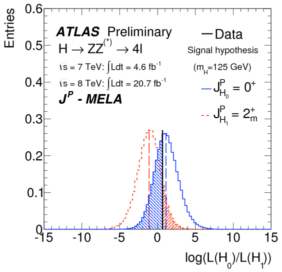
Separate signal from background, or spin-0 from spin-2

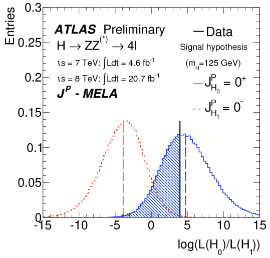
- Ratio of MEM probability models is Neyman-Pearson optimal discriminant between two hypothesis
  - Can be signal vs background
  - Can also be two different types of signal
- Example MEM applications in HEP: ATLAS H→ZZ decays:
  - Comparison between pairs in Higgs spin/parity states 0^+, 0^-, 2^+



What can you do MEM models?

- Given the computational complexity of MEM models, application largely restricted to discriminating between two simple hypothesis
- For example Higgs Spin-2 vs Higgs Spin-0, or CP states of Higgs





- What about testing theories with parameters?
- Option 1 – **brute-force** modeling in MEM process

$$f(\vec{x}|\vec{\alpha}) = \frac{1}{\sigma}\int |M|^2(\vec{y}|\vec{\alpha}) \cdot W(\vec{x},\vec{y}) \cdot d\varphi(\vec{y})$$

  very significant increase in numerical computations, validation difficult (does it work well for all α?)

- Option 2 – **factorize** parametric dependence (if possible)

$$f(\vec{x}|\vec{\alpha}) = \frac{1}{\sigma} m(\vec{\alpha},\vec{\alpha}_0)\int |M|^2(\vec{y}|\vec{\alpha}_0) \cdot W(\vec{x},\vec{y}) \cdot d\varphi(\vec{y})$$

  for example when only magnitude of M depends on a, and not the differential distribution in y

What can you do MEM models?

- Option 2 – **factorize** parametric dependence (if possible)

$$f(\vec{x}|\vec{\alpha}) = \frac{1}{\sigma} m(\vec{\alpha}, \vec{\alpha}_{ref}) \int |M|^2(\vec{y}|\vec{\alpha}_{ref}) \cdot W(\vec{x}, \vec{y}) \cdot d\varphi(\vec{y})$$

→ Can also model certain parameters *that effect distributions in y, in case M consists of multiple amplitudes and parameters only affect rates of amplitudes*

$$M = \alpha_0 M_0 + \alpha_1 M_1$$

$$|M|^2 = \alpha_0^2 |M_0|^2 + \alpha_0 \alpha_1 |M_0 M_1| + \alpha_1^2 |M_1|^2$$

$$f(\vec{x}|\vec{\alpha}) = \alpha_0^2 \int |M_0|^2(\vec{y}) \cdot W(\vec{x}, \vec{y}) \cdot d\varphi(\vec{y})$$

$$+\alpha_0 \alpha_1 \int |M_0 M_1|(\vec{y}) \cdot W(\vec{x}, \vec{y}) \cdot d\varphi(\vec{y})$$

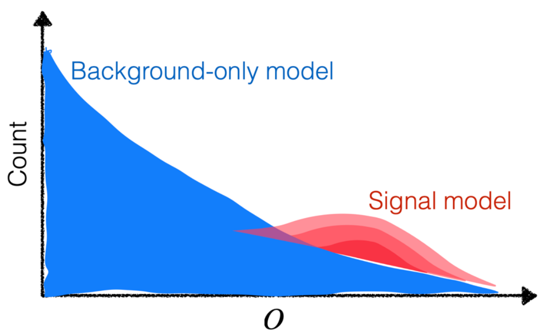
$$+\alpha_1^2 \int |M_1|^2(\vec{y}) \cdot W(\vec{x}, \vec{y}) \cdot d\varphi(\vec{y})$$

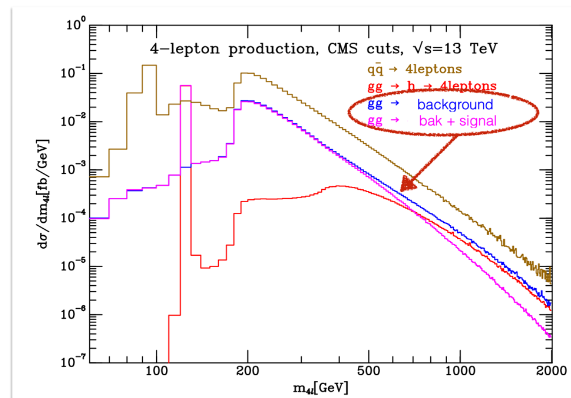
Original MEM formulation for simple hypothesis

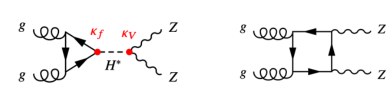
Amplitude extraction – exploiting quantum mechanics

- Theory parameters *entangled with the detector response* are difficult as some form deconvolution is needed, e.g. particle masses, particle lifetimes etc…
- But also plenty of theory parameters are effectively scaling amplitudes → factorization is widely possible!

- Simplest (trivial) case: signal + background model → factorization
- General case: signal(s) + background(s) + interference term(s)







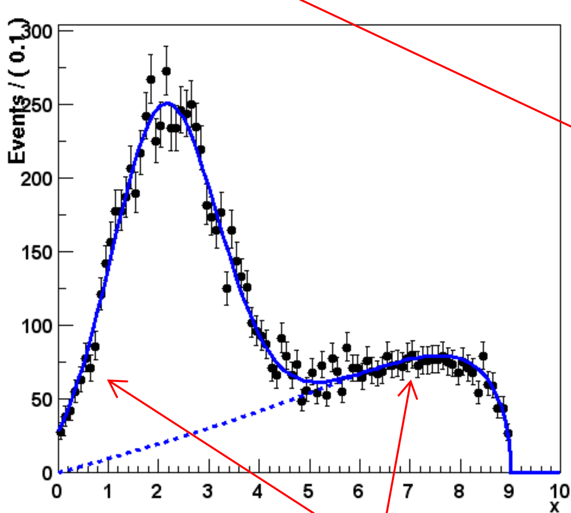
Amplitude modeling – the basics

- When it's possible to formulate p.d.f.s for observable distributions analytically, constructing probability models that sum physics amplitudes with interference effects is straightforward

Sum of p.d.f.s

Components are pdfs

$$f(x\,|\,s,b) = \frac{s}{s+b} f_s(x) + \frac{b}{s+b} f_b(x)$$



Each component is positive definite

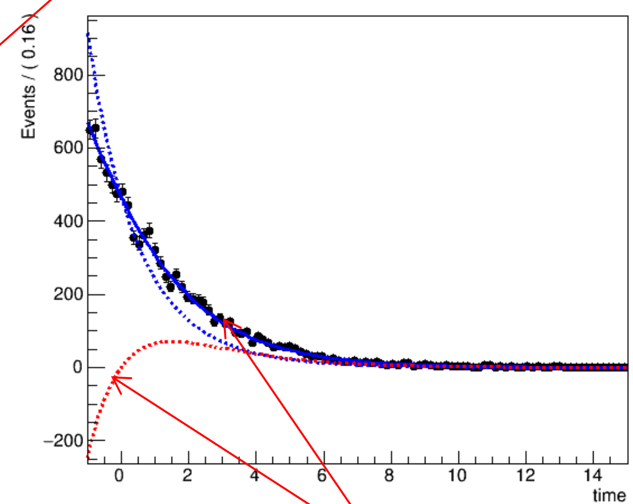
Both models are p.d.f.s

Sum of amplitudes

Components are functions

$$f(x\,|\,\vec{c}) = \frac{\sum_i c_i \cdot H_i(x)}{\sum_i c_i \cdot \int H_i(x)\,dx}$$

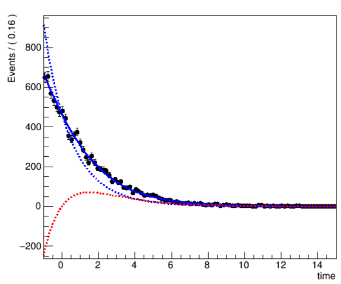
Explicit normalization required



Components can be negative, sum must be positive definite

Amplitude modeling – amplitudes from MC generators

- In many LHC analyses, observable distributions can only be obtained from MC simulation chain. *If so, this is also true for amplitudes*
- How can we simulate observable distributions corresponding to individual amplitudes?
  - Requires some support in MC generators → ability to selectively enable/disable individual amplitudes
  - Amplitudes can be negative → How does this translate to an MC event sample? Potential difficulty (e.g. allow for negative event weights)
- But otherwise straightforward – no complex deconvolution or template morphing needed to deform template histograms
  - Every physics model with (only) amplitude parameters can always be described as a weighted sum of amplitude templates (histograms!)



$$f(x|\vec{c}) = \frac{\sum_i c_i \cdot H_i(x)}{\sum_i c_i \cdot \int H_i(x)\,dx}$$

Generating a probability model from a Lagrangian

- Given a Lagrangian describing a Field Theory → can now model any transition amplitude (pp → X → Y) as a sum of real-valued amplitudes

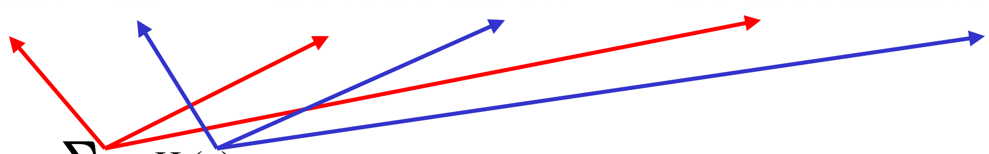
- Consider example with two operators labeled SM and BSM, with strengths g_{SM} and g_{BSM} respectively. Matrix Element is

$$\mathcal{M}(g_{SM}, g_{BSM}) = g_{SM}\mathcal{O}_{SM} + g_{BSM}\mathcal{O}_{BSM}$$

- Transition amplitude is $|M|^2$:

$$T(g_{SM}, g_{BSM}) \propto |\mathcal{M}(g_{SM}, g_{BSM})|^2$$

$$|\mathcal{M}(g_{SM}, g_{BSM})|^2 = g_{SM}^2|\mathcal{O}_{SM}|^2 + g_{BSM}^2|\mathcal{O}_{BSM}|^2 + 2g_{SM}g_{BSM}\mathcal{R}(\mathcal{O}_{SM}^*\mathcal{O}_{BSM})$$



$$f(x|\vec{c}) = \frac{\sum_i c_i \cdot H_i(x)}{\sum_i c_i \cdot \int H_i(x)\,dx}$$

Mapping of Wilson coefficients to template scale factors

The mapping of templates to operators

- Note that templates do not need to correspond one-to-one to single operators or pure interference terms

$$|\mathcal{M}(g_{\text{SM}}, g_{\text{BSM}})|^2 = g_{\text{SM}}^2 |O_{\text{SM}}|^2 + g_{\text{BSM}}^2 |O_{\text{BSM}}|^2 + 2 g_{\text{SM}} g_{\text{BSM}} \mathcal{R}(O_{\text{SM}}^* O_{\text{BSM}})$$

- For 2 operator, any three independent pairs of g_{SM},g_{BSM} values can generate templates that will span the whole parameter space. E.g.

$$T_{in}(1,0) \propto |O_{\text{SM}}|^2$$

$$T_{in}(0,1) \propto |O_{\text{BSM}}|^2$$

$$T_{in}(1,1) \propto |O_{\text{SM}}|^2 + |O_{\text{BSM}}|^2 + 2\mathcal{R}(O_{\text{SM}}^* O_{\text{BSM}})$$

$$|\mathcal{M}(g_{\text{SM}}, g_{\text{BSM}})|^2 = \underbrace{(g_{\text{SM}}{}^2 - g_{\text{SM}} g_{\text{BSM}})}\, T_{in}(1,0) + \underbrace{(g_{\text{BSM}}{}^2 - g_{\text{SM}} g_{\text{BSM}})}\, T_{in}(0,1) + \underbrace{g_{\text{SM}} g_{\text{BSM}}}\, T_{in}(1,1)$$

7 / 24

Note that in this choice for T_{in} there are *no templates corresponding to pure interference terms*

→ All templates are positive definite! even though the underlying templates are not (necessarily)

→ This result is general: for any amplitude sum there is always a configuration of templates that are all positive

$$T_{in}(1,0) \propto |O_{SM}|^2$$

$$T_{in}(0,1) \propto |O_{BSM}|^2$$

$$T_{in}(1,1) \propto |O_{SM}|^2 + |O_{BSM}|^2 + 2\mathcal{R}(O^*_{SM} O_{BSM})$$

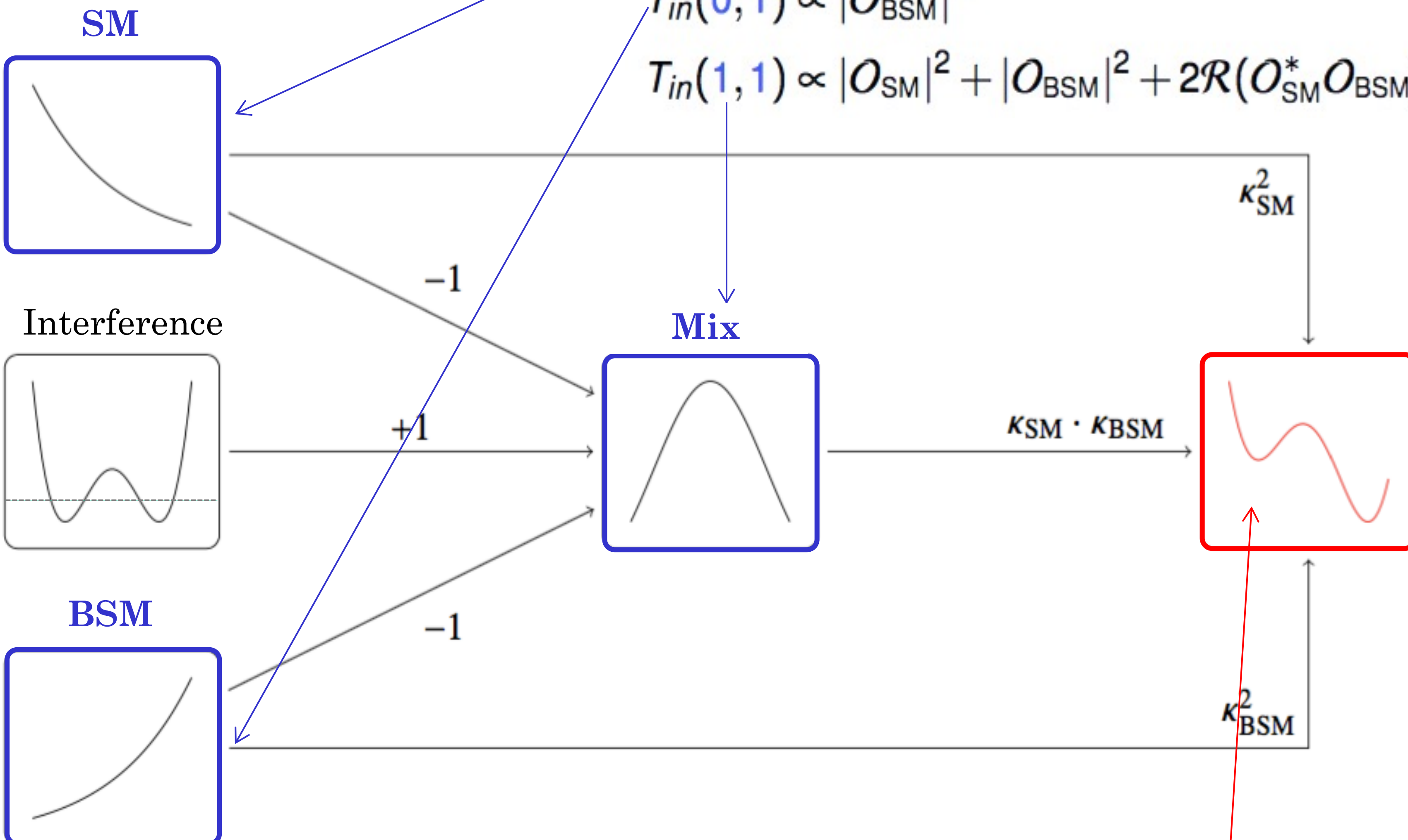
$$|\mathcal{M}(g_{SM}, g_{BSM})|^2 = \underbrace{(g_{SM}{}^2 - g_{SM} g_{BSM})}\, T_{in}(1,0) + \underbrace{(g_{BSM}{}^2 - g_{SM} g_{BSM})}\, T_{in}(0,1) + \underbrace{g_{SM} g_{BSM}}\, T_{in}(1,1)$$

Rearranged amplitude sums

$$T_{in}(1,0) \propto |O_{SM}|^2$$

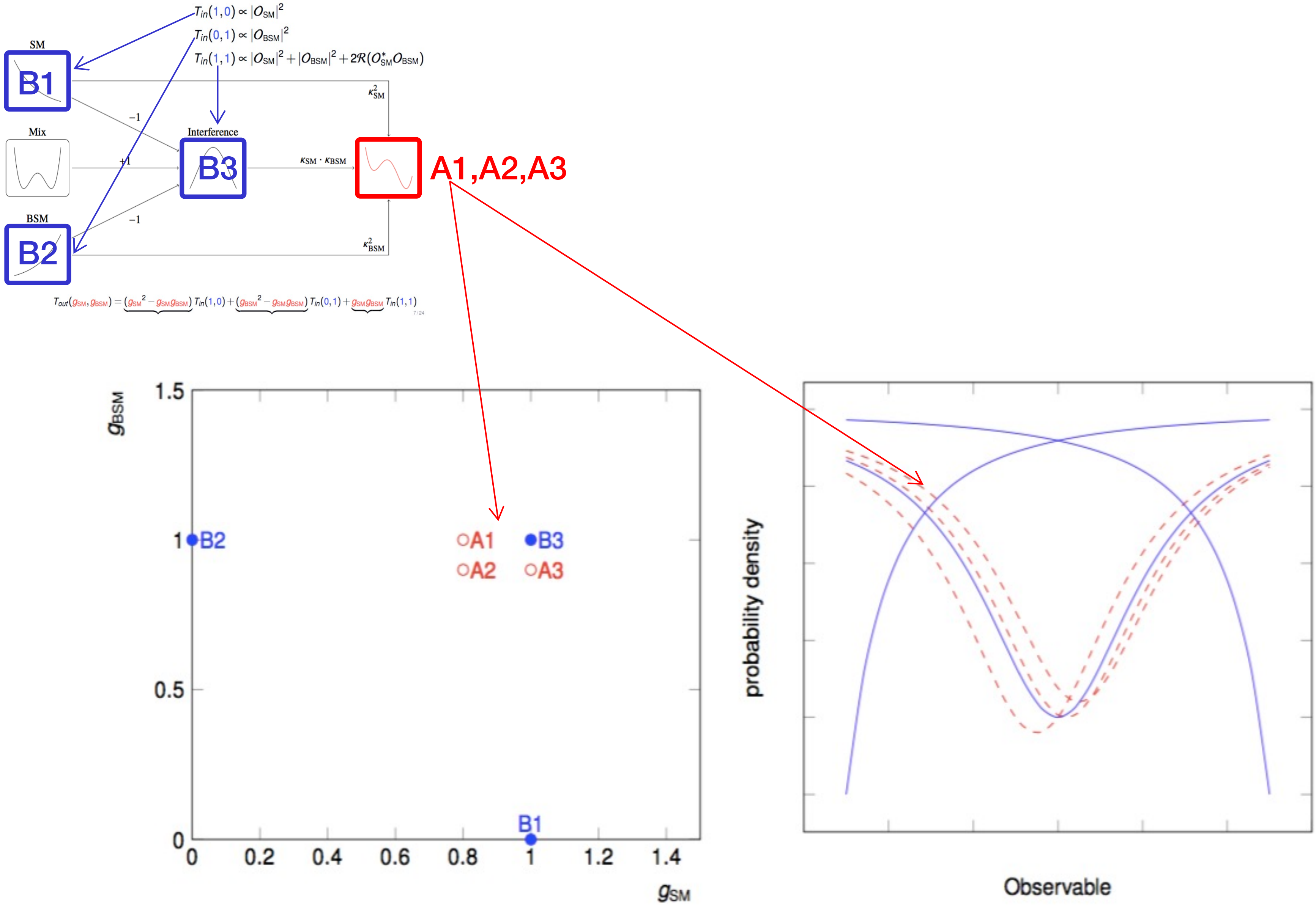
$$T_{in}(0,1) \propto |O_{BSM}|^2$$

$$T_{in}(1,1) \propto |O_{SM}|^2 + |O_{BSM}|^2 + 2\mathcal{R}(O^*_{SM} O_{BSM})$$



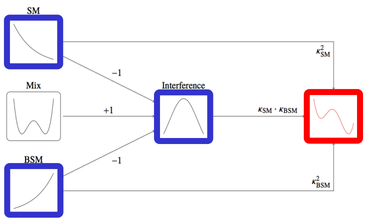
$$T_{out}(g_{SM}, g_{BSM}) = \underbrace{\left(g_{SM}{}^2 - g_{SM} g_{BSM}\right)} T_{in}(1,0) + \underbrace{\left(g_{BSM}{}^2 - g_{SM} g_{BSM}\right)} T_{in}(0,1) + \underbrace{g_{SM} g_{BSM}} T_{in}(1,1)$$

Rearranged amplitude sums



The mapping of templates to operators

- Generalizing further, we will now work with 3 templates T_1,T_2,T_3 that are sampled at arbitrary points in the (g_{SM},g_{BSM}) parameter space



$$T_{\rm in}(1,0) \qquad T_{\rm in}(g_{\rm SM,1}, g_{\rm BSM,1})$$
$$T_{\rm in}(0,1) \qquad T_{\rm in}(g_{\rm SM,2}, g_{\rm BSM,2})$$
$$T_{\rm in}(1,1) \qquad T_{\rm in}(g_{\rm SM,3}, g_{\rm BSM,3}).$$

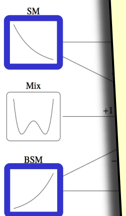
- The output probability model then takes the general form

$$T_{\rm out}(g_{\rm SM}, g_{\rm BSM}) = \underbrace{(g_{\rm SM}^2 - g_{\rm SM}g_{\rm BSM})}_{=w_1} T_{\rm in}(1,0) + \underbrace{(g_{\rm BSM}^2 - g_{\rm SM}g_{\rm BSM})}_{=w_2} T_{\rm in}(0,1) + \underbrace{g_{\rm SM}g_{\rm BSM}}_{=w_3} T_{\rm in}(1,1).$$

$$\begin{aligned} T_{\rm out}(g_{\rm SM}, g_{\rm BSM}) = {} & \underbrace{(a_{11}g_{\rm SM}^2 + a_{12}g_{\rm BSM}^2 + a_{13}g_{\rm SM}g_{\rm BSM})}_{w_1} T_{\rm in}(g_{\rm SM,1}, g_{\rm BSM,1}) \\ & + \underbrace{(a_{21}g_{\rm SM}^2 + a_{22}g_{\rm BSM}^2 + a_{23}g_{\rm SM}g_{\rm BSM})}_{w_2} T_{\rm in}(g_{\rm SM,2}, g_{\rm BSM,2}) \\ & + \underbrace{(a_{31}g_{\rm SM}^2 + a_{32}g_{\rm BSM}^2 + a_{33}g_{\rm SM}g_{\rm BSM})}_{w_3} T_{\rm in}(g_{\rm SM,3}, g_{\rm BSM,3}). \end{aligned}$$

The mapping of templates to operators

- Generalizing further ... at



This then only leaves the calculation of the appropriate coefficients a_{ij} that occur in the weights functions w_1,w_2,w_3 for templates T_1,T_2,T_3

Their solution is found by solving the matrix equation

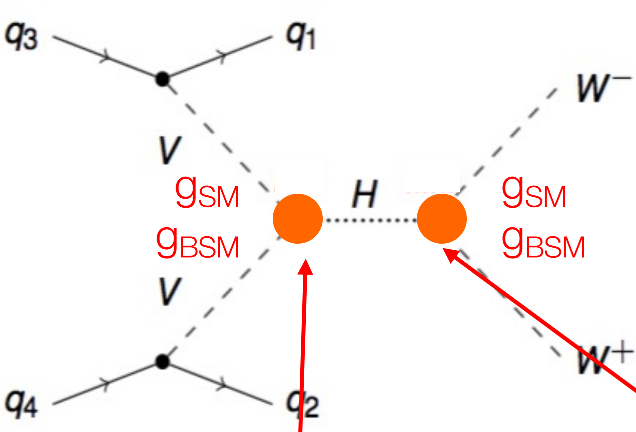
$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \cdot \begin{pmatrix} g_{SM,1}^2 & g_{SM,2}^2 & g_{SM,3}^2 \\ g_{BSM,1}^2 & g_{BSM,2}^2 & g_{BSM,3}^2 \\ g_{SM,1}g_{BSM,1} & g_{SM,2}g_{BSM,2} & g_{SM,3}g_{BSM,3} \end{pmatrix} = \mathbb{1}$$

$=w_2$ $=w_3$

$$\begin{aligned} T_{out}(g_{SM}, g_{BSM}) = & \underbrace{(a_{11}g_{SM}^2 + a_{12}g_{BSM}^2 + a_{13}g_{SM}g_{BSM})}_{w_1} T_{in}(g_{SM,1}, g_{BSM,1}) \\ & + \underbrace{(a_{21}g_{SM}^2 + a_{22}g_{BSM}^2 + a_{23}g_{SM}g_{BSM})}_{w_2} T_{in}(g_{SM,2}, g_{BSM,2}) \\ & + \underbrace{(a_{31}g_{SM}^2 + a_{32}g_{BSM}^2 + a_{33}g_{SM}g_{BSM})}_{w_3} T_{in}(g_{SM,3}, g_{BSM,3}). \end{aligned}$$

A more realistic physics example

- In many scenarios new physics can enter amplitudes in both the production and decay vertex of a t-channel process



- Corresponding matrix element of this process

$$\mathcal{M}(g_{\text{SM}}, g_{\text{BSM}}) = \left(g_{\text{SM}} \cdot O_{\text{SM},p} + g_{\text{BSM}} \cdot O_{\text{BSM},p}\right) \cdot \left(g_{\text{SM}} \cdot O_{\text{SM},d} + g_{\text{BSM}} \cdot O_{\text{BSM},d}\right).$$

A more realistic physics example

- A little math shows we now need 5 independent templates

$$\mathcal{M}(g_{\mathrm{SM}}, g_{\mathrm{BSM}}) = \left(g_{\mathrm{SM}} \cdot O_{\mathrm{SM},p} + g_{\mathrm{BSM}} \cdot O_{\mathrm{BSM},p}\right) \cdot \left(g_{\mathrm{SM}} \cdot O_{\mathrm{SM},d} + g_{\mathrm{BSM}} \cdot O_{\mathrm{BSM},d}\right).$$

$$\begin{aligned}
|\mathcal{M}(g_{\mathrm{SM}}, g_{\mathrm{BSM}})|^2 &= \left(g_{\mathrm{SM}} O_{\mathrm{SM},p} + g_{\mathrm{BSM}} O_{\mathrm{BSM},p}\right)^2 \cdot \left(g_{\mathrm{SM}} O_{\mathrm{SM},d} + g_{\mathrm{BSM}} O_{\mathrm{BSM},d}\right)^2 && \leftarrow 1 \\
&= g_{\mathrm{SM}}^4 \cdot O_{\mathrm{SM},p}^2 O_{\mathrm{SM},d}^2 + g_{\mathrm{BSM}}^4 \cdot O_{\mathrm{BSM},p}^2 O_{\mathrm{BSM},d}^2 && \leftarrow 2 \\
&+ g_{\mathrm{SM}}^3 g_{\mathrm{BSM}} \cdot \left(O_{\mathrm{SM},p}^2 \Re(O_{\mathrm{SM},d}^* O_{\mathrm{BSM},d}) + \Re(O_{\mathrm{SM},p}^* O_{\mathrm{BSM},p}) O_{\mathrm{SM},d}^2\right) && \leftarrow 3 \\
&+ g_{\mathrm{SM}}^2 g_{\mathrm{BSM}}^2 \cdot \left(O_{\mathrm{SM},p}^2 O_{\mathrm{BSM},d}^2 + O_{\mathrm{BSM},p}^2 O_{\mathrm{SM},d}^2\right) && \leftarrow 4 \\
&+ g_{\mathrm{SM}} g_{\mathrm{BSM}}^3 \cdot \left(O_{\mathrm{BSM},p}^2 \Re(O_{\mathrm{SM},d}^* O_{\mathrm{BSM},d}) + \Re(O_{\mathrm{SM},p}^* O_{\mathrm{BSM},p}) O_{\mathrm{BSM},d}^2\right). && \leftarrow 5
\end{aligned}$$

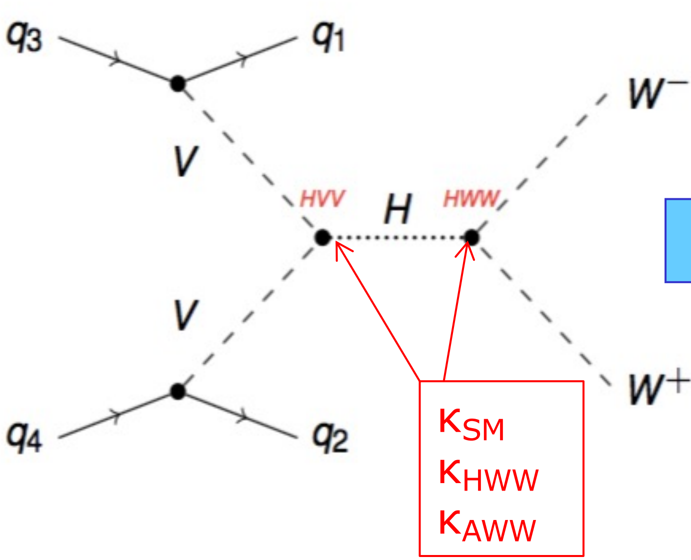
- And the template model can be written as

$$\begin{aligned}
T_{\mathrm{out}}(g_{\mathrm{SM}}, g_{\mathrm{BSM}}) &= \underbrace{\left(a_{11} g_{\mathrm{SM}}^4 + a_{12} g_{\mathrm{SM}}^3 g_{\mathrm{BSM}} + a_{13} g_{\mathrm{SM}}^2 g_{\mathrm{BSM}}^2 + a_{14} g_{\mathrm{SM}} g_{\mathrm{BSM}}^3 + a_{15} g_{\mathrm{BSM}}^4\right)}_{w_1} T_{\mathrm{in}}(g_{\mathrm{SM},1}, g_{\mathrm{BSM},1}) \\
&= \underbrace{\left(a_{21} g_{\mathrm{SM}}^4 + a_{22} g_{\mathrm{SM}}^3 g_{\mathrm{BSM}} + a_{23} g_{\mathrm{SM}}^2 g_{\mathrm{BSM}}^2 + a_{24} g_{\mathrm{SM}} g_{\mathrm{BSM}}^3 + a_{25} g_{\mathrm{BSM}}^4\right)}_{w_2} T_{\mathrm{in}}(g_{\mathrm{SM},2}, g_{\mathrm{BSM},2}) \\
&= \underbrace{\left(a_{31} g_{\mathrm{SM}}^4 + a_{32} g_{\mathrm{SM}}^3 g_{\mathrm{BSM}} + a_{33} g_{\mathrm{SM}}^2 g_{\mathrm{BSM}}^2 + a_{34} g_{\mathrm{SM}} g_{\mathrm{BSM}}^3 + a_{35} g_{\mathrm{BSM}}^4\right)}_{w_3} T_{\mathrm{in}}(g_{\mathrm{SM},3}, g_{\mathrm{BSM},3}) \\
&= \underbrace{\left(a_{41} g_{\mathrm{SM}}^4 + a_{42} g_{\mathrm{SM}}^3 g_{\mathrm{BSM}} + a_{43} g_{\mathrm{SM}}^2 g_{\mathrm{BSM}}^2 + a_{44} g_{\mathrm{SM}} g_{\mathrm{BSM}}^3 + a_{45} g_{\mathrm{BSM}}^4\right)}_{w_4} T_{\mathrm{in}}(g_{\mathrm{SM},4}, g_{\mathrm{BSM},4}) \\
&= \underbrace{\left(a_{51} g_{\mathrm{SM}}^4 + a_{52} g_{\mathrm{SM}}^3 g_{\mathrm{BSM}} + a_{53} g_{\mathrm{SM}}^2 g_{\mathrm{BSM}}^2 + a_{54} g_{\mathrm{SM}} g_{\mathrm{BSM}}^3 + a_{55} g_{\mathrm{BSM}}^4\right)}_{w_5} T_{\mathrm{in}}(g_{\mathrm{SM},5}, g_{\mathrm{BSM},5}).
\end{aligned}$$

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \end{pmatrix} \cdot \begin{pmatrix} g_{\mathrm{SM},1}^4 & g_{\mathrm{SM},2}^4 & g_{\mathrm{SM},3}^4 & g_{\mathrm{SM},4}^4 & g_{\mathrm{SM},5}^4 \\ g_{\mathrm{SM},1}^3 g_{\mathrm{BSM},1} & g_{\mathrm{SM},2}^3 g_{\mathrm{BSM},2} & g_{\mathrm{SM},3}^3 g_{\mathrm{BSM},3} & g_{\mathrm{SM},4}^3 g_{\mathrm{BSM},4} & g_{\mathrm{SM},5}^3 g_{\mathrm{BSM},5} \\ g_{\mathrm{SM},1}^2 g_{\mathrm{BSM},1}^2 & g_{\mathrm{SM},2}^2 g_{\mathrm{BSM},2}^2 & g_{\mathrm{SM},3}^2 g_{\mathrm{BSM},3}^2 & g_{\mathrm{SM},4}^2 g_{\mathrm{BSM},4}^2 & g_{\mathrm{SM},5}^2 g_{\mathrm{BSM},5}^2 \\ g_{\mathrm{SM},1} g_{\mathrm{BSM},1}^3 & g_{\mathrm{SM},2} g_{\mathrm{BSM},2}^3 & g_{\mathrm{SM},3} g_{\mathrm{BSM},3}^3 & g_{\mathrm{SM},4} g_{\mathrm{BSM},4}^3 & g_{\mathrm{SM},5} g_{\mathrm{BSM},5}^3 \\ g_{\mathrm{BSM},1}^4 & g_{\mathrm{BSM},2}^4 & g_{\mathrm{BSM},3}^4 & g_{\mathrm{BSM},4}^4 & g_{\mathrm{BSM},5}^4 \end{pmatrix} = \mathbb{1}$$

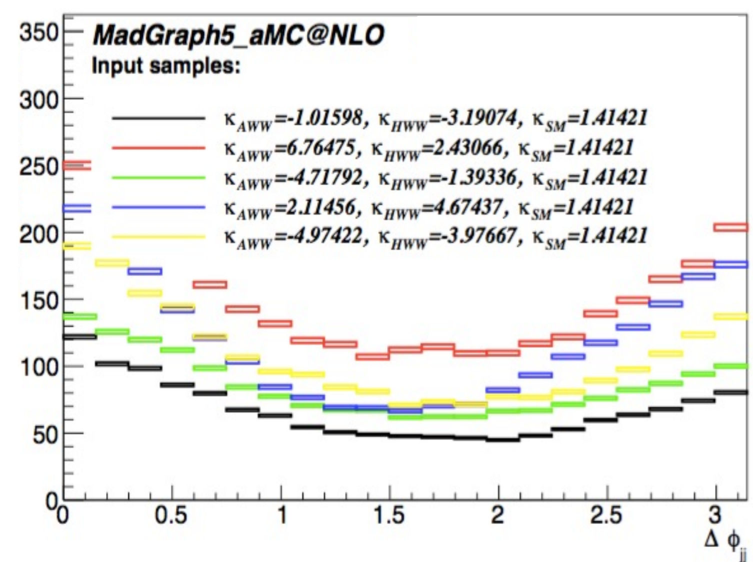
A concrete example VBH → H → WW

3 shared parameters → 15 terms in $|M|^2$ expression → 15 input distributions needed



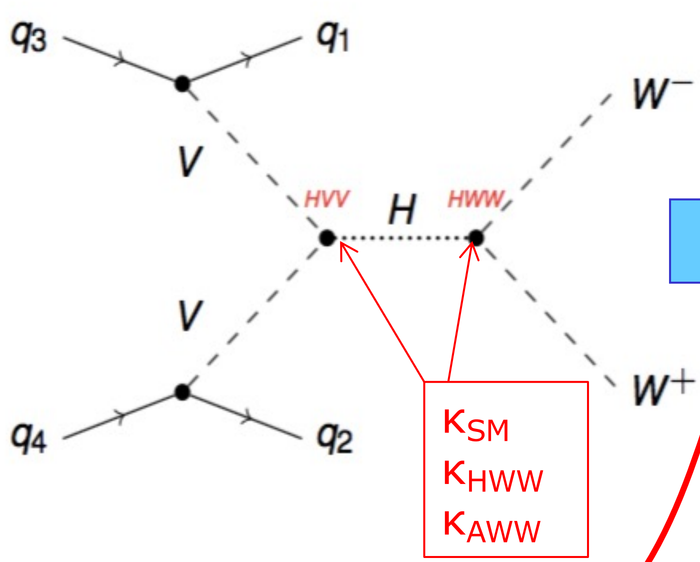
$$T_{out}(\Delta\phi_{jj} | \kappa_{SM}, \kappa_{HWW}, \kappa_{AWW}) = \sum w_i(\kappa_{SM}, \kappa_{HWW}, \kappa_{AWW}) \cdot T_{in,i}(\Delta\phi_{jj})$$

T_{out}

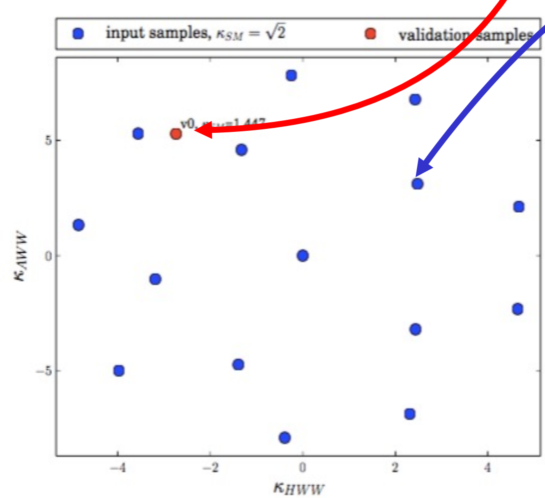


A concrete example VBH → H → WW

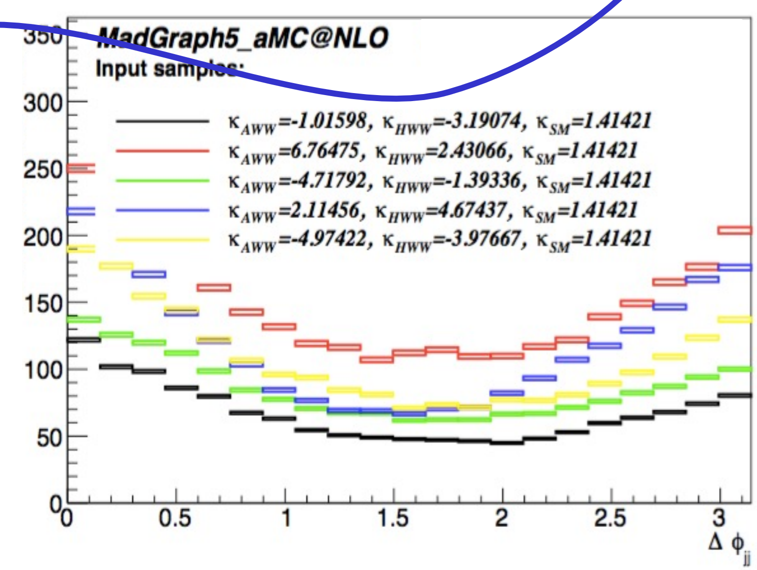
3 shared parameters → 15 terms in $|M|^2$ expression → 15 input distributions needed



$$T_{out}(\Delta\phi_{jj}|\kappa_{SM},\kappa_{HWW},\kappa_{AWW}) = \sum w_i(\kappa_{SM},\kappa_{HWW},\kappa_{AWW}) \cdot T_{in,i}(\Delta\phi_{jj})$$



T_{out}

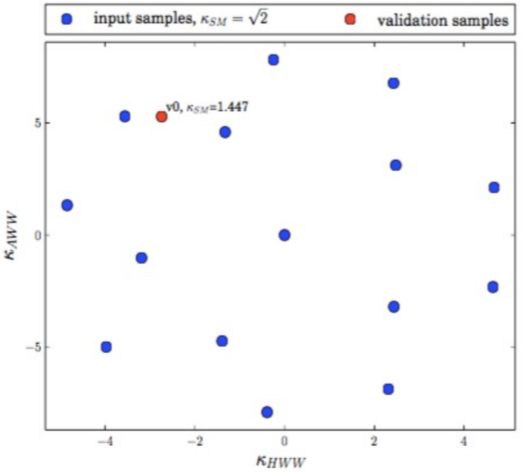


Summary on amplitude models

- Amplitude sum models work (in terms of mathematics) exactly the same as template interpolation models.
  - But have difference choice of coefficient (polynomials instead of linear terms)
  - Appropriate choice results in interpolation mechanism that is physically meaningful → no approximation in interpolation (beyond assumption of LO physics)
  - Freedom of choice in sampling points ensures that all sampled distributions are positive definite (no interference-only terms)
  - Computationally fast & efficient process
- But need to watch configuration of sampled points

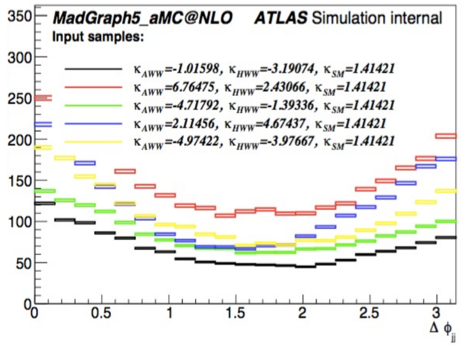
  → if interpolated states (e.g. measured minimum) is far from important samples then large scale factors might be applied

  → If so, blow-up of MC statistical fluctuations occurs…



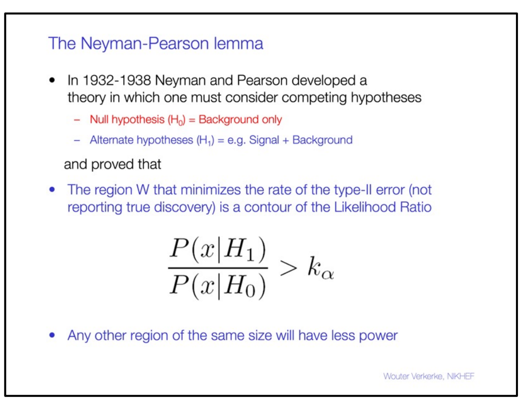
Choosing optimal observables in amplitude models

- All templates in morphing models describe *the same observable (here* $\Delta\phi_{jj}$)



$$T_{out}(\Delta\phi_{jj}|\kappa_{SM},\kappa_{HWW},\kappa_{AWW}) = \sum w_i(\kappa_{SM},\kappa_{HWW},\kappa_{AWW}) \cdot T_{in,i}(\Delta\phi_{jj})$$

- *What observable has the most statistical power?* In this example, a strategically chosen detector-level observable was used ($\Delta\phi_{jj}$)
- General answer: use the Neyman-Pearson Lemma to construct an optimal observable (as was done for the earlier Higgs Spin MEM example)



The Neyman-Pearson lemma

- In 1932-1938 Neyman and Pearson developed a theory in which one must consider competing hypotheses
  - Null hypothesis (H_0) = Background only
  - Alternate hypotheses (H_1) = e.g. Signal + Background

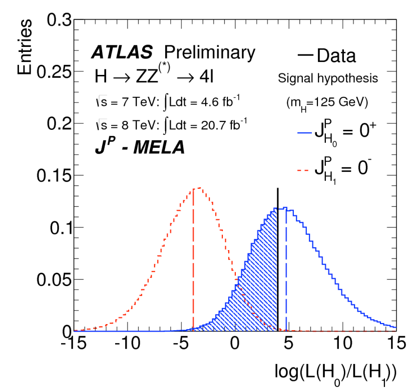
and proved that

- The region W that minimizes the rate of the type-II error (not reporting true discovery) is a contour of the Likelihood Ratio

$$\frac{P(x|H_1)}{P(x|H_0)} > k_\alpha$$

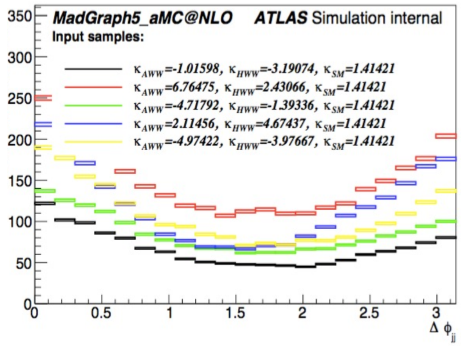
- Any other region of the same size will have less power

Wouter Verkerke, NIKHEF



Amplitude models & optimal observables → NBSI

- **Q**: What is the optimal observable in amplitude model with many terms $T_{in,i}$



$$T_{out}(\Delta\phi_{jj}|\kappa_{SM},\kappa_{HWW},\kappa_{AWW}) = \sum w_i(\kappa_{SM},\kappa_{HWW},\kappa_{AWW}) \cdot T_{in,i}(\Delta\phi_{jj})$$

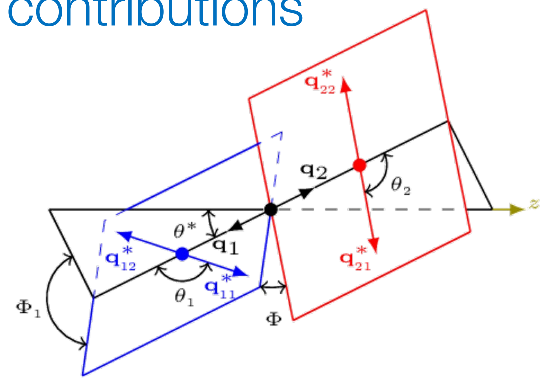
E.g. $\Delta\phi_{jj}$ might be optimal to distinguish $|O_{SM}|^2$ from $|O_{BSM}|^2$
but it is automatically optimal to distinguish between $|O_{SM}|^2$ *and* $|O_{SM}O_{BSM}|$?

- **A**: **For i>2, there is generally <u>not</u> a single optimal observable to descriminate between all operator terms $T_{in,i}$**

- Effect: **Sensitivity** to constrain parameters κ ***reduced*** *for values of κ≠0*.
  - Close to SM, κ =0, where $T_{in,SM}$ will dominate all other terms it can still be close to optimal
  - Notably weak measurements (with interval boundaries on κ far from 0), are further weakened

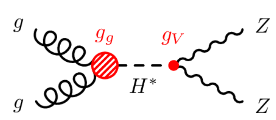
Optimal sensitivity for amplitude models → NSBI

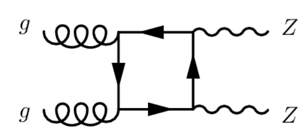
- You can regain optimal sensitivity with a (small) paradigm change: instead of aiming to model the likelihood, model the likelihood ratio
- Example case: $H \to ZZ \to l^+l^-l^+l^-$ production in the offshell mass range, which has significant S-B interference contributions
  - **14 observables** that describe Higgs production and decay kinematics

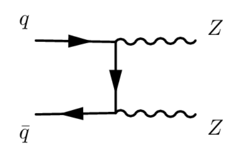


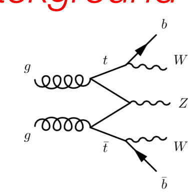
  - **Likelihood** in any observable x for signal strength μ

$$p(x|\mu) = \frac{1}{\sigma(\mu)}\left[\underbrace{(\mu - \sqrt{\mu})\,\sigma_S\,P_S(x)}_{\textit{signal}} + \underbrace{\sqrt{\mu}\,\sigma_{SBI}\,P_{SBI}(x)}_{\textit{mix}} + \underbrace{(1-\sqrt{\mu})\,\sigma_B\,P_B(x)}_{\textit{background}} + \underbrace{\sigma_{qqZZ}\,P_{qqZZ}(x) + (\text{ttV \& EW terms})}_{\textit{non-interf. background}}\right]$$







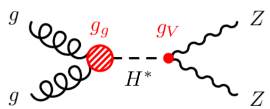


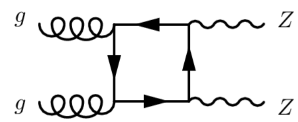
Optimal sensitivity for amplitude models → Likelihood ratios

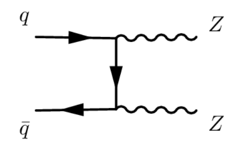
- Example case: $H \to ZZ \to l^+l^-l^+l^-$ production in the offshell mass range, which has significant S-B interference contributions
  - **Likelihood** in any observable x for signal strength μ

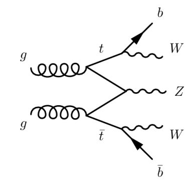
$$p(x|\mu) = \frac{1}{\sigma(\mu)}\left[(\mu - \sqrt{\mu})\,\sigma_S\,P_S(x) + \sqrt{\mu}\,\sigma_{SBI}\,P_{SBI}(x) + (1-\sqrt{\mu})\,\sigma_B\,P_B(x) + \sigma_{qqZZ}\,P_{qqZZ}(x) + (\text{ttV \& EW terms})\right]$$

signal — mix — background — non-interf. background









  - **Likelihood ratio** formulation of the same problem

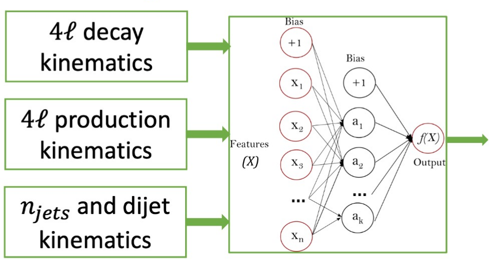
$$\frac{p(x|\mu)}{p'(x)} = \frac{1}{\sigma(\mu)}\left[(\mu - \sqrt{\mu})\,\sigma_S \boxed{\frac{P_S(x)}{p'(x)}} + \sqrt{\mu}\,\sigma_{SBI} \boxed{\frac{P_{SBI}(x)}{p'(x)}} + (1-\sqrt{\mu})\,\sigma_B \boxed{\frac{P_B(x)}{p'(x)}} + \sigma_{qqZZ} \boxed{\frac{P_{qqZZ}(x)}{p'(x)}}\right]$$

Now free to model each ratio with a **separate optimal observable**

Estimation Likelihood / Density ratios → ML/AI methods

- Constructing the optimal observables:
  - Use modern ML techniques to 'learn' the best way to model the density ratios in the 14-dimensional observable space
  - Train separate ML networks *for each relevant pair of amplitudes*

Training



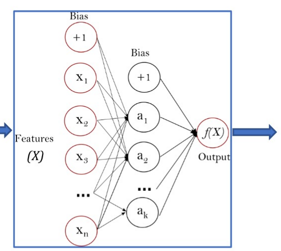
$$\frac{p(x|\mu)}{p'(x)}$$

Density ratio estimate

Illustration: A. Ghosh

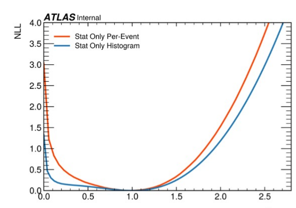
Application

Data



$$\frac{p(x|\mu)}{p'(x)}$$

$$t_\mu = \sum_i^{N_{events}} -2\log\frac{p(x_i|\mu)/p'(x_i)}{p(x_i|\hat{\mu})/p'(x_i)} = \sum_i^{N_{events}} -2\log\frac{p(x_i|\mu)}{p(x_i|\hat{\mu})}$$



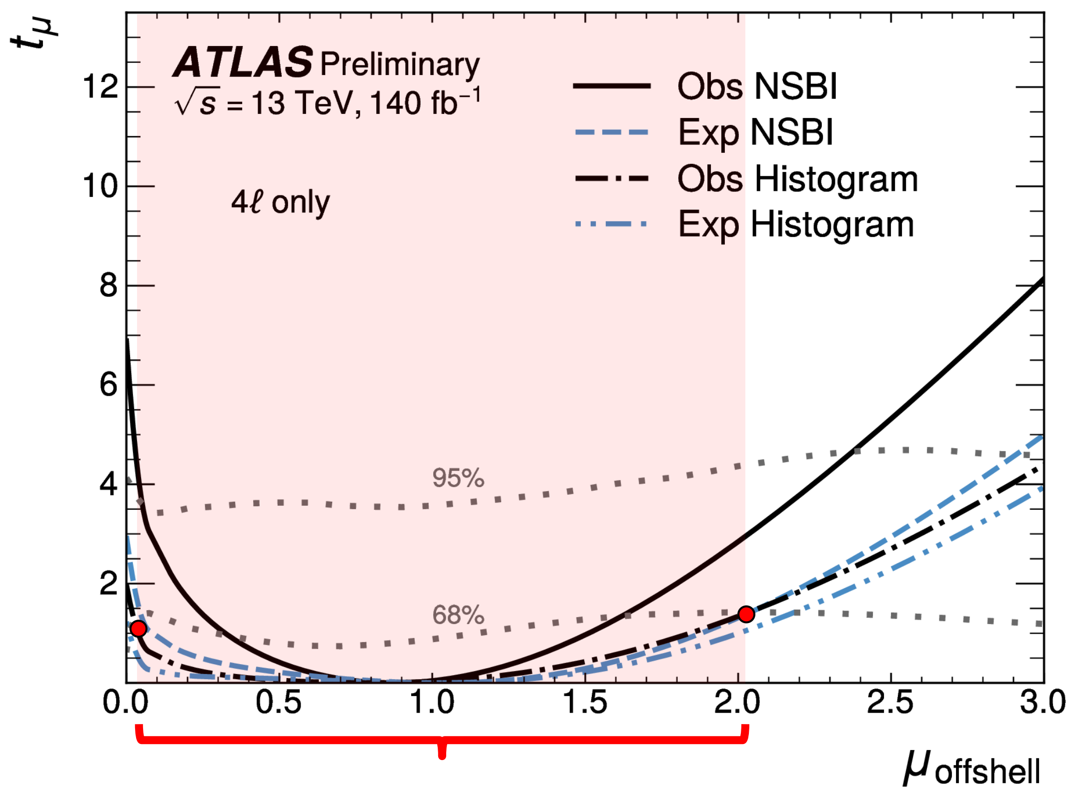
  - Ratio of likelihood ratios recovers usual likelihood ratio test statistic

$$t_\mu(\vec{x}|\mu) = \frac{L(\vec{x}|\mu)}{L(\vec{x}|\mu)}$$

  - Bonus: no more histograms → unbinned likelihood

Effect of optimal observables – Higgs offshell example

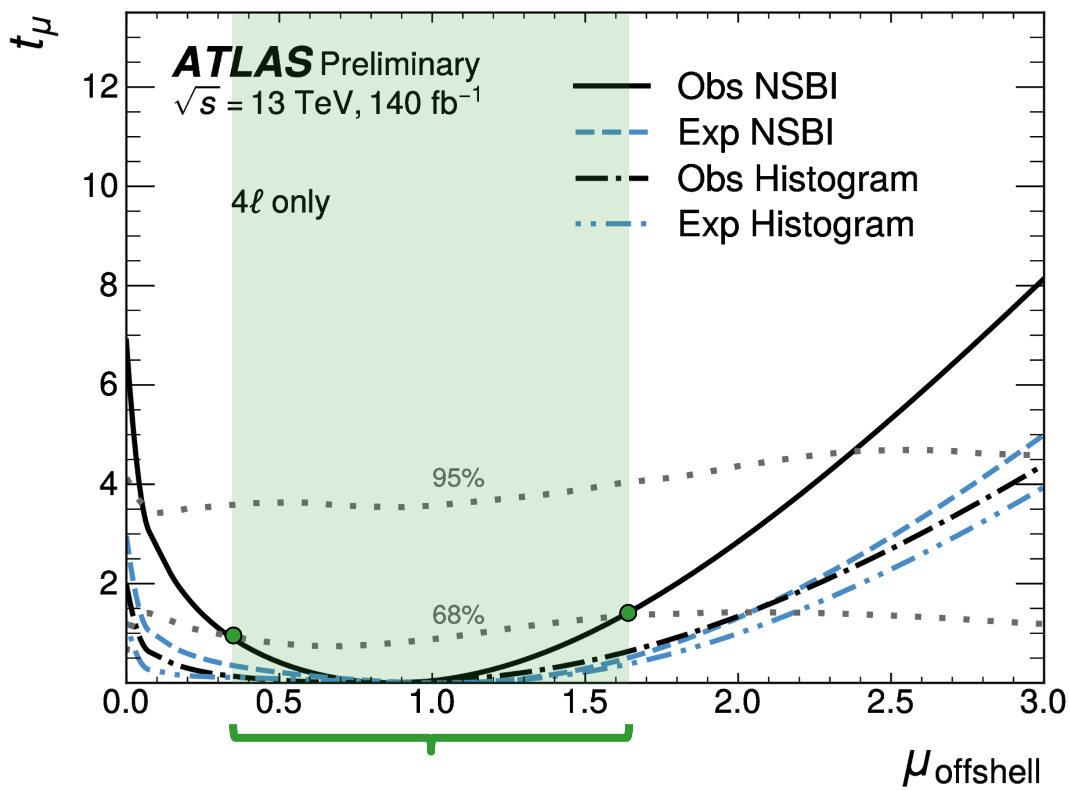
- Comparison: template morphing histogram analysis vs
  neural inference model with optimal observables
  on the same data



68% confidence interval – template histogram morphing

Effect of optimal observables – Higgs offshell example

- Comparison: template morphing histogram analysis vs
  neural inference model with optimal observables
  on the same data



68% confidence interval – neural SBI method

Implementation challenges of Neural SBI method

- Skipped over a very large amount of important details
  - **Calibration of NN is crucial** → output is directly used in calculation of confidence intervals → no correction mechanism to account for data/simulation mismatches in formalism
  - **Choice of denominator likelihood is crucial** – it must be positive-definite in the entire analyzed region (otherwise ratio is ill-defined)
  - **Systematic uncertainties** - implemented with a morphing-like approach using nuisance parameters and profiling over these
  - **Computation times huge** – training took 4000 hours on a NVidia A100 GPU

- But bottom line is clear – optimal observable models based on NSBI estimates on MEM-like density ratio have great potential
  - Excellent use case for Effective Field Theory measurements, now very ubiquitous at the LHC
  - *At present only a single LHC analysis has implement the method in full*

Roadmap of this course

- Start with basics, gradually build up to complexity