

HistFactory introduction

Alexander Held¹

¹ University of Wisconsin–Madison

IRIS-HEP Simulation Based Inference Blueprint

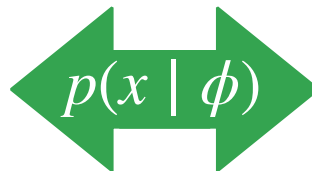
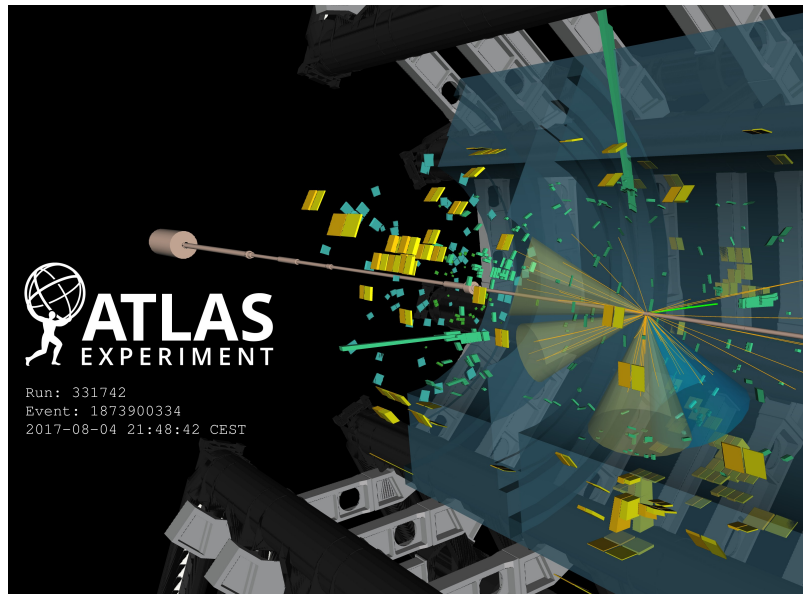
<https://indico.cern.ch/event/1600677/>

Feb 26, 2026

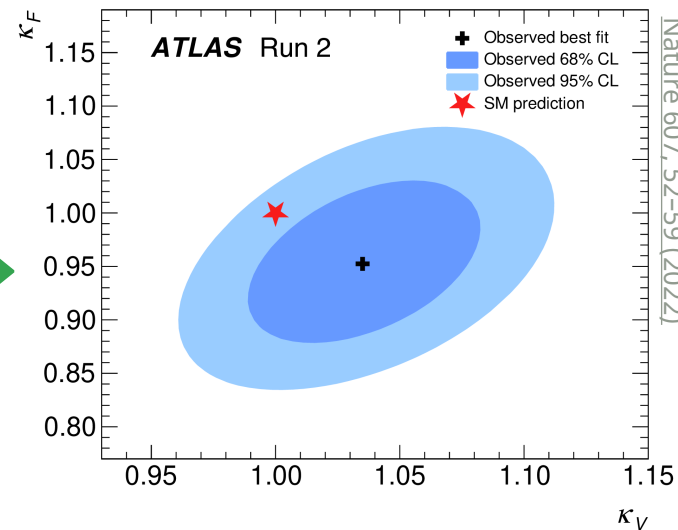
Big picture: turning collisions into publications

- **What we want:** statements about physical parameters ϕ , given data x_i collected by an experiment
 - connection: the **likelihood** $L_x(\phi) = p(x | \phi)$ — key ingredient for all subsequent statistical inference
 - $p(x | \phi)$ means: pick a ϕ and you get a probability density function over x

observations x_i



statements about parameters ϕ



An intractable likelihood function

- We **need** $p(x | \phi)$ — unfortunately this very high-dimensional **integral** is **intractable**, **cannot evaluate** this

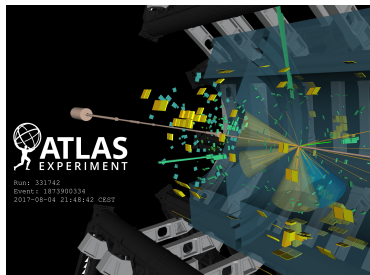
$$p(x | \phi) = \int dz_D dz_S dz_P p(x | z_D) p(z_D | z_S) p(z_S | z_P) p(z_P | \phi)$$

observables x

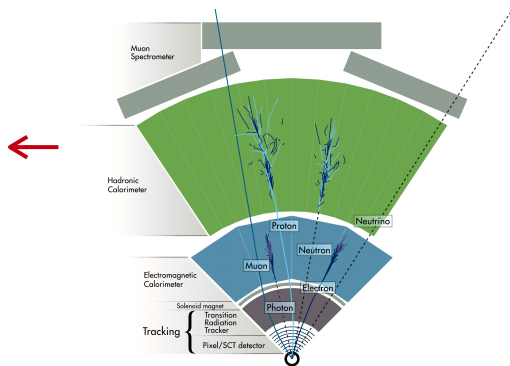
detector interaction z_D

parton shower z_S

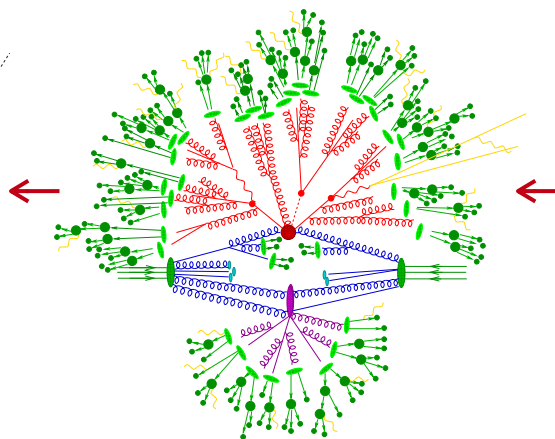
parton level z_P



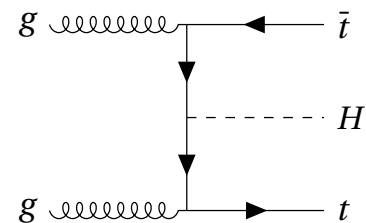
Phys.Lett. B 784 (2018) 173



CERN-EX-1301009



JHEP 0902 (2009) 007



The dependence on parameters ϕ is here.

Histograms & summary statistics

- Use MC samples to **estimate the density** $p(x | \phi)$, e.g. by **filling histograms** with the samples x_i ✓

- histograms are a **convenient method** for density estimation

- Histograms are hit by the **curse of dimensionality** ✗

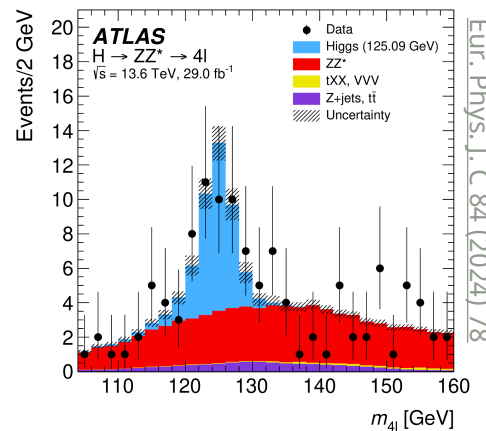
- number of samples x_i needed scales **exponentially** with **dimension of observation**

- We use **summary statistics** to reduce dimensionality of our measurements ✓

- operate on objects like **jets** instead of **detector channel responses**

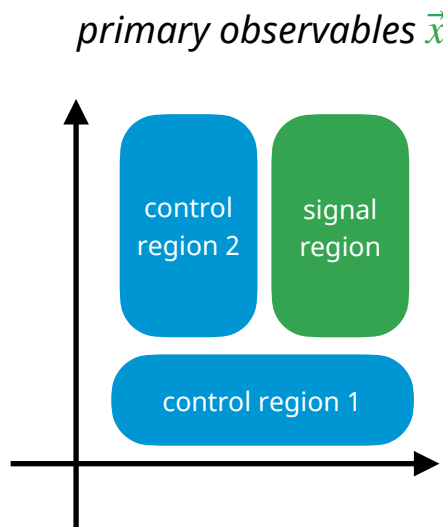
- use **physicists & machine learning** to efficiently compress information

- **Challenge:** finding the right low-dimensional summary statistic — crucial for sensitivity

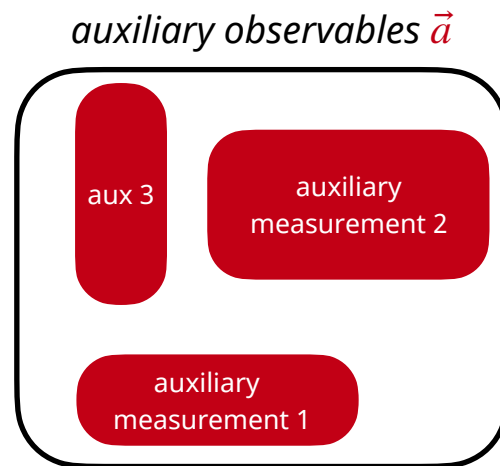


HistFactory for binned template models

A measurement: primary and auxiliary observables



data in our analysis



calibration measurements + theory
(assumed to be statistically independent)

- Our models are a **combination of primary and auxiliary measurements** $p_{\text{primary}}(\vec{x} | \vec{v}) \cdot p_{\text{aux}}(\vec{a})$
 - auxiliary: both experimental (e.g. detector calibration) and theory (e.g. changes in simulation)

The HistFactory model: overview



see also CMS Combine

[arXiv:2404.06614](https://arxiv.org/abs/2404.06614)

- **HistFactory** is a statistical model for **binned template fits** ([CERN-OPEN-2012-016](https://arxiv.org/abs/1207.1332))
 - prescription for constructing probability density functions (pdfs) from **small set of building blocks**
 - covers a **wide range of use cases** and extensible, used extensively in ATLAS with ~equivalent model in CMS
 - here: primary observables are \vec{n} , auxiliary observables are \vec{a}

$$p(\vec{n}, \vec{a} \mid \vec{k}, \vec{\theta}) = \prod_i \text{Pois}(n_i \mid \nu_i(\vec{k}, \vec{\theta})) \cdot \prod_j c_j(a_j \mid \theta_j)$$

The diagram illustrates the components of the HistFactory model. The probability density function $p(\vec{n}, \vec{a} \mid \vec{k}, \vec{\theta})$ is shown as a product of two terms: a **primary term** and an **auxiliary term**.

- primary term:** $\prod_i \text{Pois}(n_i \mid \nu_i(\vec{k}, \vec{\theta}))$. This term represents the prediction (summed over samples) for the primary observables n_i . It is a product over all bins i . The mean ν_i depends on the constrained nuisance parameters \vec{k} and the unconstrained parameters $\vec{\theta}$.
- auxiliary term:** $\prod_j c_j(a_j \mid \theta_j)$. This term represents a constraint term (e.g. Gaussian) for the auxiliary observables a_j . It is a product over all bins j .

Annotations in the diagram include:

- observed data:** points to \vec{n} (green arrow).
- auxiliary data, e.g. from calibration measurement:** points to \vec{a} (red arrow).
- unconstrained parameters, e.g. POI:** points to $\vec{\theta}$ (blue arrow).
- constrained nuisance parameters:** points to \vec{k} (purple arrow).
- prediction (summed over samples):** points to the primary term.
- constraint term (e.g. Gaussian):** points to the auxiliary term.
- product over all bins:** points to the product symbols in both terms.

The model prediction: $\nu_i(\vec{k}, \vec{\theta})$

- The **prediction** in each bin is a **sum of all contributing samples**, e.g. $\nu_i = \mu \cdot S_i(\vec{\theta}) + B_i(\vec{\theta})$
 - template histograms are obtained from our simulator chain
 - samples correspond to different kinds of collision processes
 - nuisance parameters $\vec{\theta}$ affect the model prediction

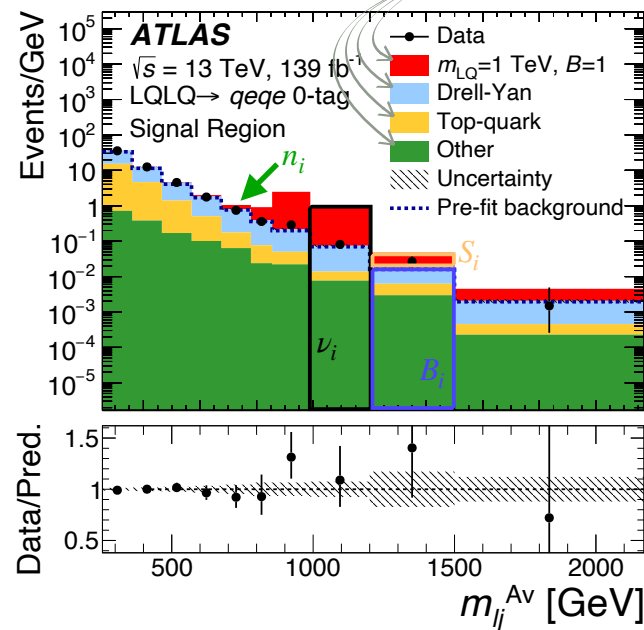
observed data

prediction (summed over samples)

$$p(\vec{n}, \vec{a} \mid \vec{k}, \vec{\theta}) = \prod_i \text{Pois}(n_i \mid \nu_i(\vec{k}, \vec{\theta})) \cdot \prod_j c_j(a_j \mid \theta_j)$$

unconstrained parameters, e.g. POI

a "channel" in HistFactory with different samples



Systematic variations

- Need to model $\nu(\vec{k}, \vec{\theta})$ for any value of nuisance parameters $\vec{\theta}$ encoding systematic uncertainties

- **Ideal case:** just run simulator for any value of $\vec{\theta}$

- not computationally feasible in practice

- **Instead:** pick some values & **interpolate**

- in practice we use on-axis variations

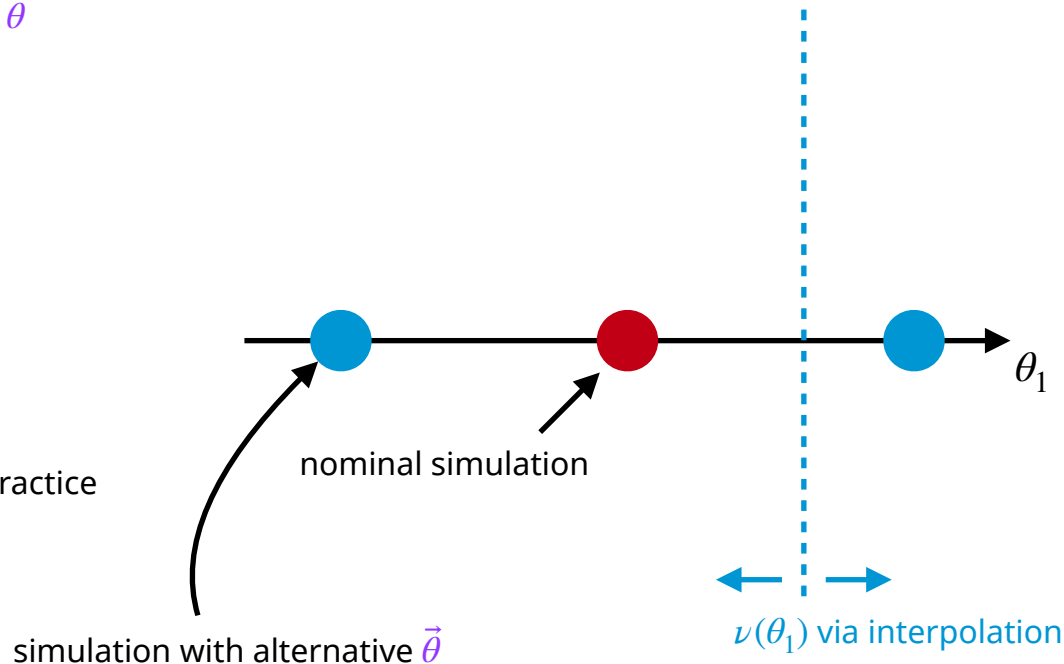
- variations typically are “one at a time”

- Lots of **assumptions** here that we rely on in practice

- where to simulate

- interpolation choice

- effects factorize



Systematic variations

- Need to model $\nu(\vec{k}, \vec{\theta})$ for any value of nuisance parameters $\vec{\theta}$ encoding systematic uncertainties

- **Ideal case:** just run simulator for any value of $\vec{\theta}$

- not computationally feasible in practice

- **Instead:** pick some values & **interpolate**

- in practice we use on-axis variations

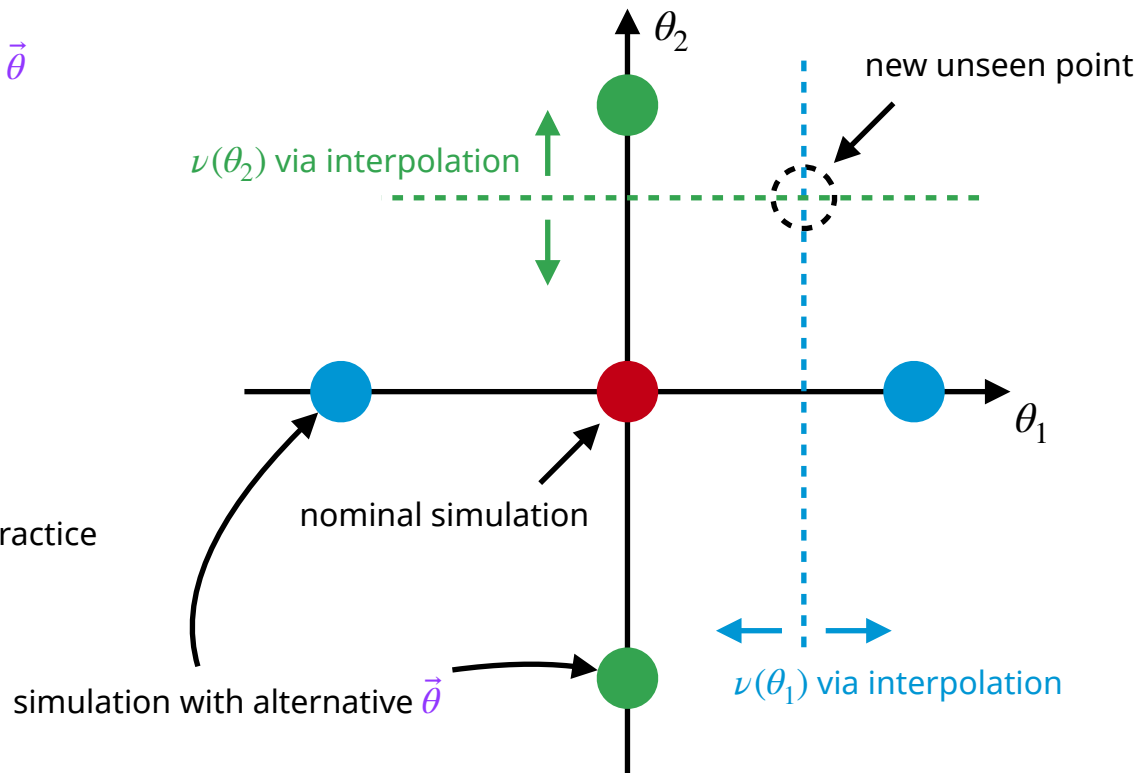
- variations typically are “one at a time”

- Lots of **assumptions** here that we rely on in practice

- where to simulate

- interpolation choice

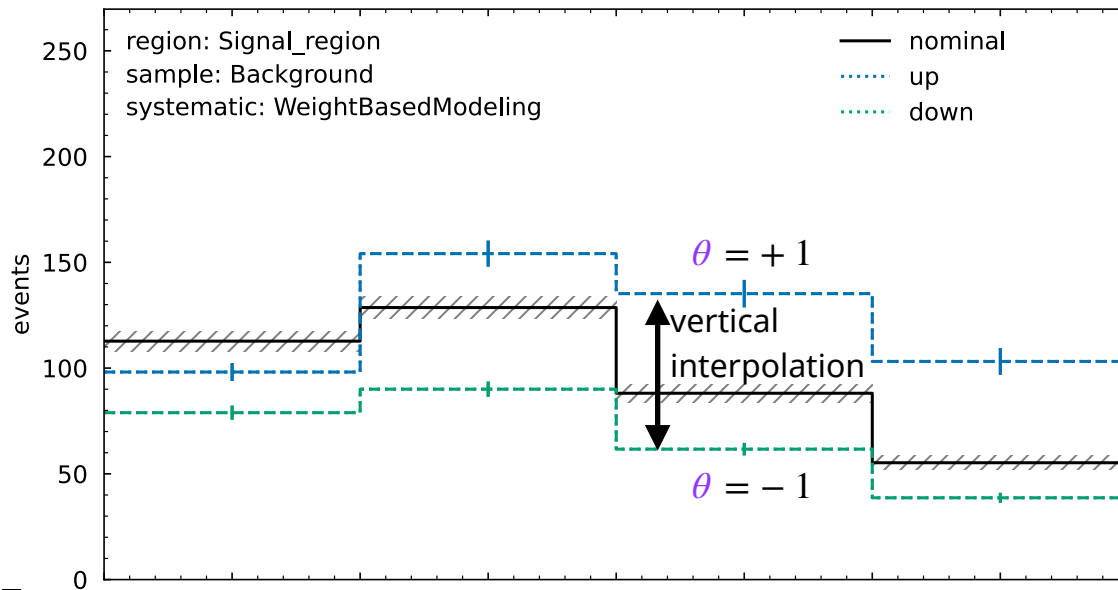
- effects factorize



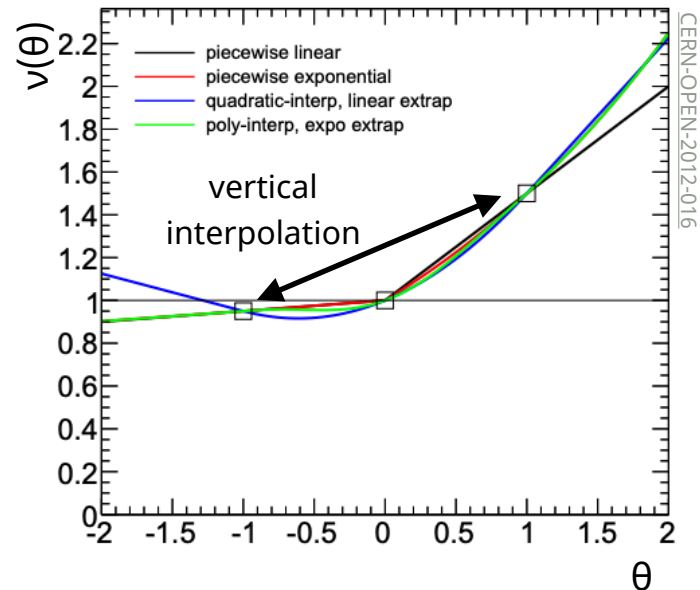
Interpolating between points

- Use model prediction $\nu_i(\vec{k}, \vec{\theta})$ for three points θ , **interpolate to generalize**
 - interpolation is typically “vertical”, other approaches exist (but more specialized)
 - note: information about **statistical uncertainties** in varied templates **is lost** here ([arXiv:1809.05778](https://arxiv.org/abs/1809.05778))

toy example: distributions for $\theta = -1, 0, +1$



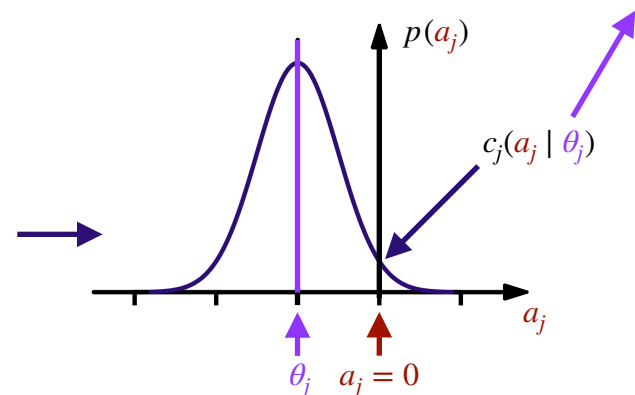
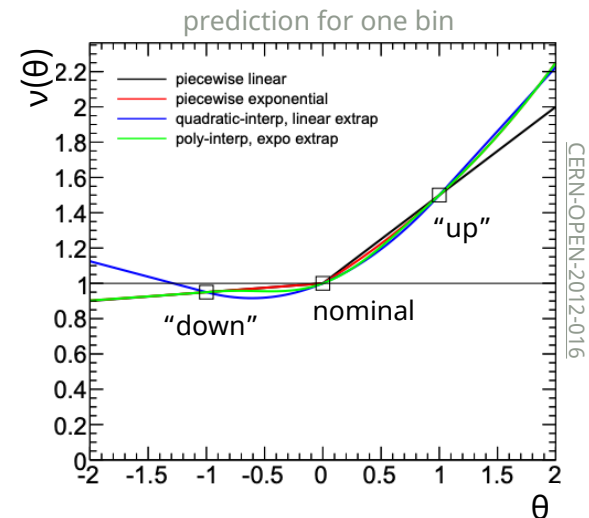
interpolation in one bin



Systematic uncertainties with HistFactory

- Common **systematic uncertainties** specified with **two template histograms**
 - “up variation”: model prediction for $\theta = +1$
 - “down variation”: model prediction for $\theta = -1$
 - interpolation & extrapolation provides **model predictions ν for any $\vec{\theta}$**
- Gaussian constraint terms** used to model auxiliary measurements (in most cases)
 - centered around nuisance parameter (NP) θ_j
 - normalized width ($\sigma = 1$) and mean (auxiliary data $a_j = 0$)
 - penalty for pulling NP away from best-fit auxiliary measurement value

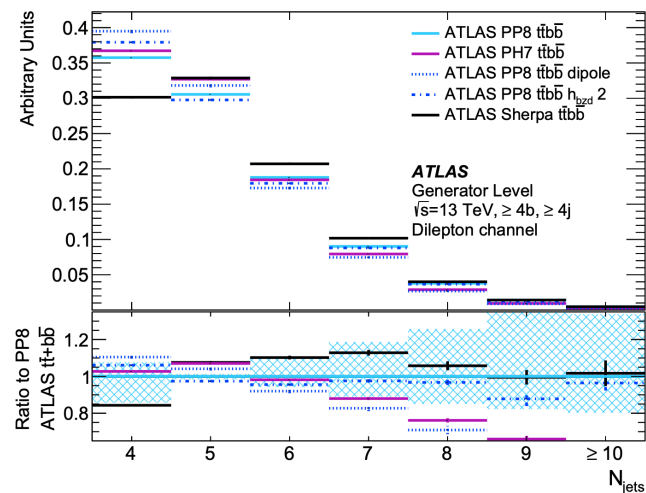
$$p(\vec{n}, \vec{a} \mid \vec{k}, \vec{\theta}) = \prod_i \text{Pois}(n_i \mid \nu_i(\vec{k}, \vec{\theta})) \cdot \prod_j c_j(a_j \mid \theta_j)$$



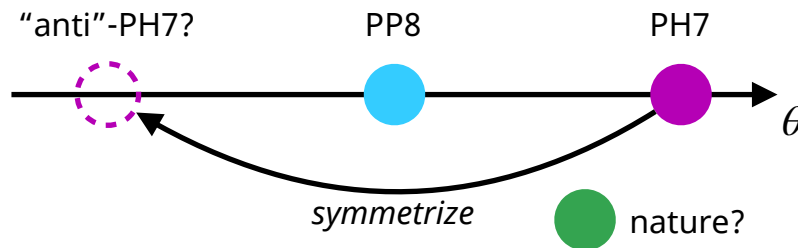
Complication: two-point systematics

- Sometimes have cases where **variations in simulator chain are discrete**
 - e.g. **choice of one simulator vs alternative**
- Typical treatment: **interpolate to treat as continuous, symmetrize**
 - **lots of assumptions** here, but need to make a choice to profile
- Especially **tricky to deal with** when these play a large role
 - concerns about **overly constraining** uncertainty of nuisance parameter
 - best-fit model prediction may lie away from both choices

modeling choices for main background of $t\bar{t}(b\bar{b})$



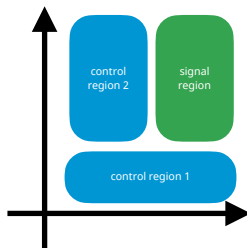
LHCWGW-2022-003



The HistFactory model: structure

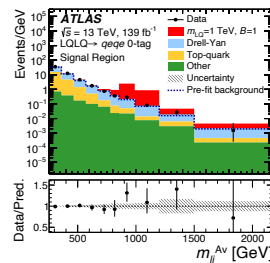
- **HistFactory** models are **highly structured**

channels
subsets of data



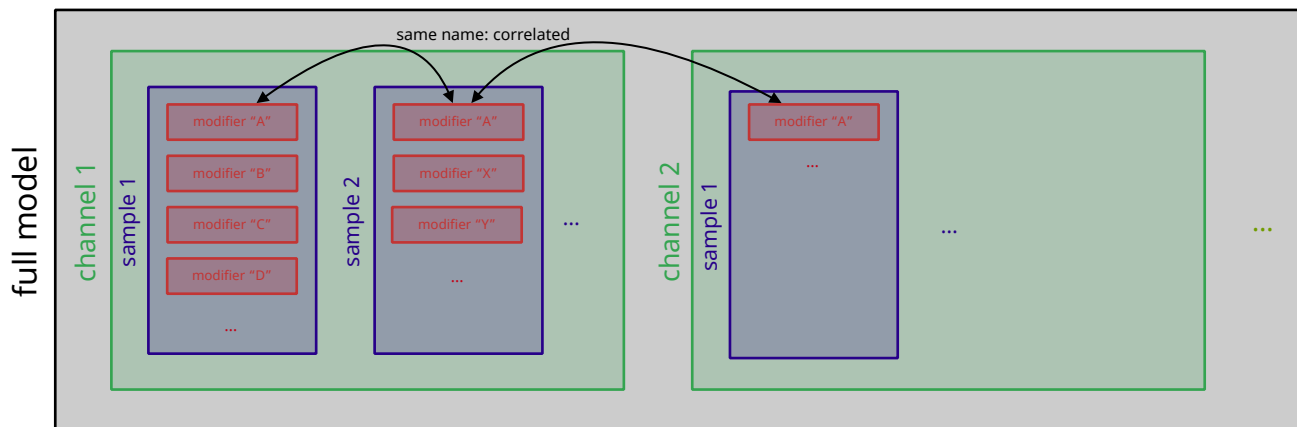
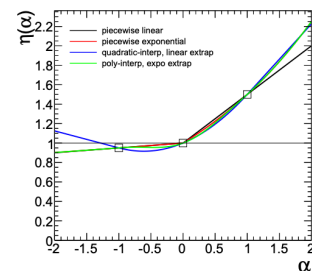
samples

different contributions to a channel



modifiers

acting on the samples



Summary

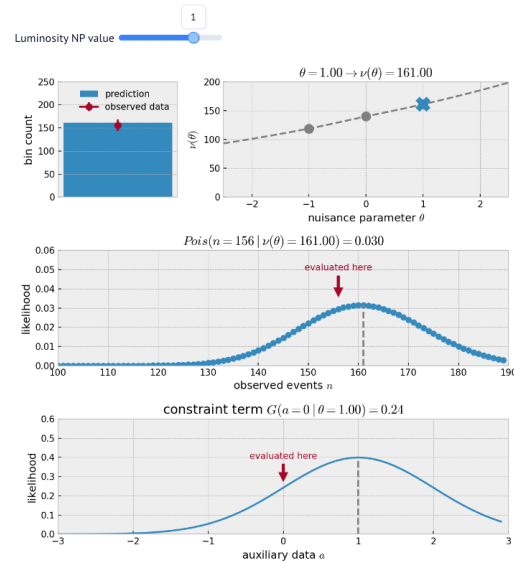
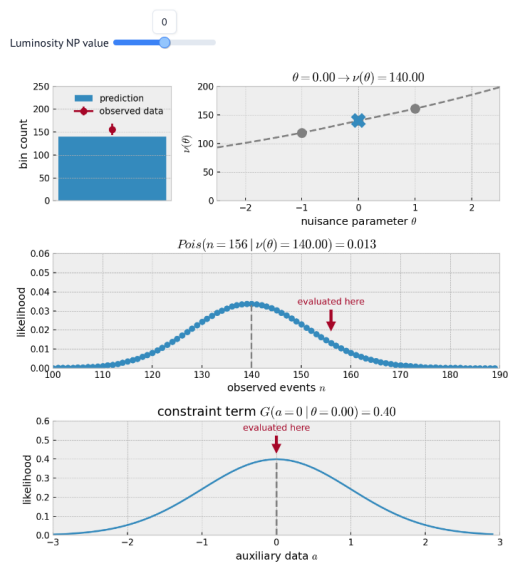
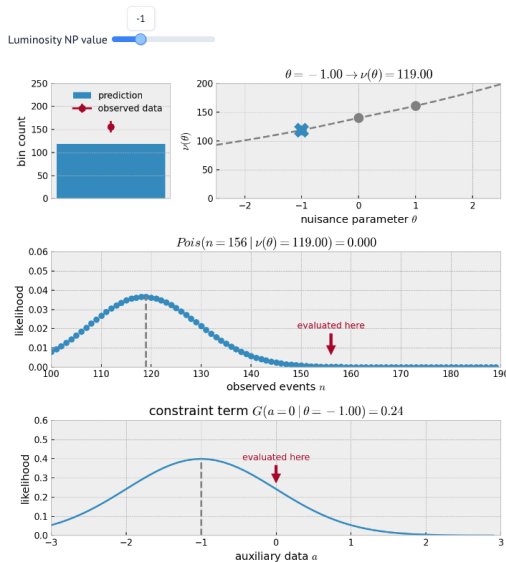
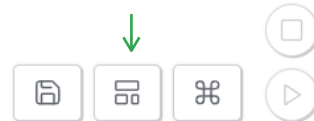
- **HistFactory** is a popular example for a family of **histogram-based models for statistical analysis**
 - covers broad range of use cases from **few building blocks**
 - maps well onto
 - **language** (channel/region, sample)
 - available **information / format** (e.g. one-at-a-time systematic variations)
 - human-readable **visualizations** (all 1d distributions)
 - **extensible**: e.g. profile likelihood unfolding, parameterized shapefactors for EFT applications
- There is a lot of **power in its simplicity**
 - **1.5 decades of experience** with how models behave
 - a lot of **tooling** exists (e.g. <https://gitlab.cern.ch/TRExStats/TRExFitter>)
 - **multiple implementations**: ROOT, pyhf, LiteHF.jl

Backup

Hands-on with a HistFactory model

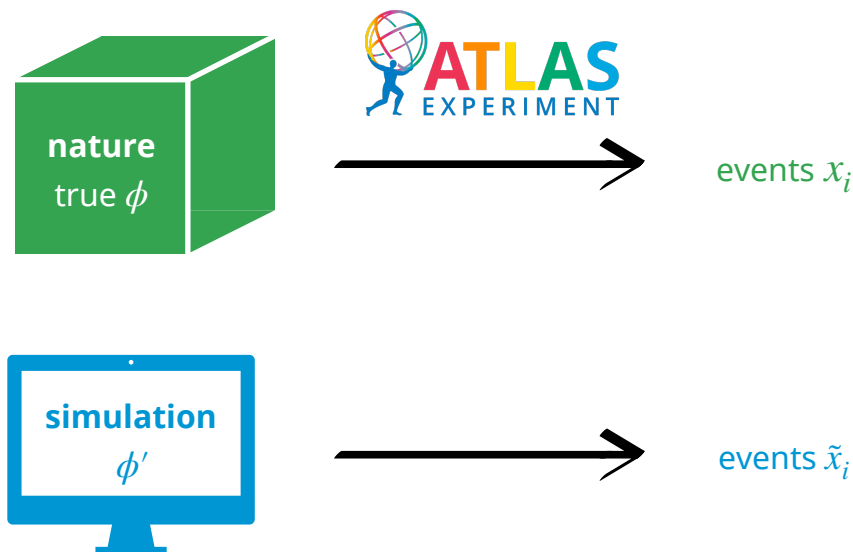
- We can have a look at a **HistFactory model** together: <https://marimo.app/l/mo6jgx>
 - might load for a few seconds, make sure to use “toggle app view” on the bottom right
 - control the **nuisance parameter value** and observed the effect
 - see [alheld/interactive-histfactory-example](#) in case of issues with the link above

use “toggle app view”



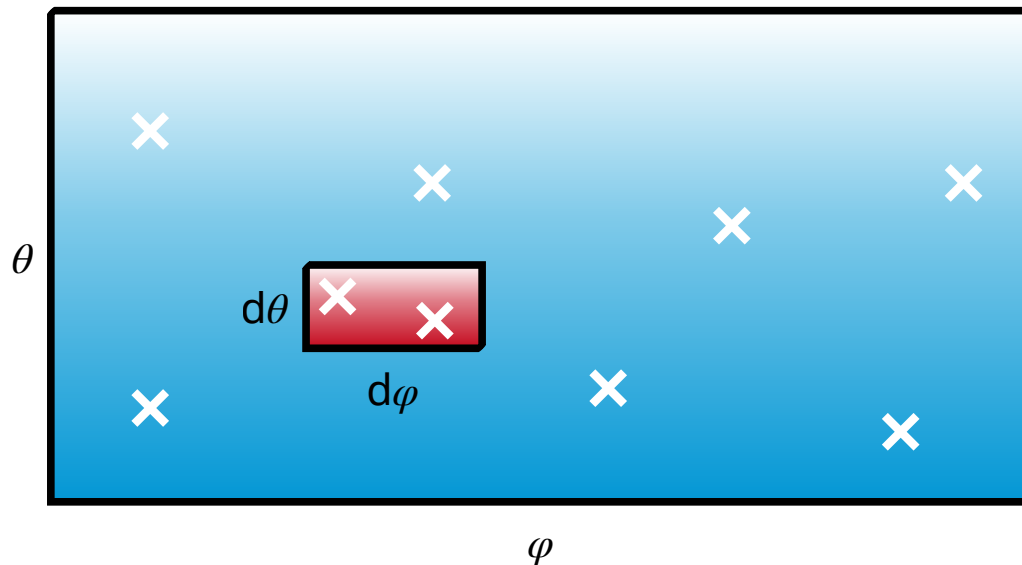
Simulation to approximate nature

- We wrote down $p(x | \phi)$, yet **cannot evaluate** it directly
- Have a **set of simulators** for all steps involved and **can draw samples** $\tilde{x}_i \sim p(x | \phi')$, which **approximate nature**
 - another way to say this: we *can* “run Monte Carlo”



Simulation-based density estimation

- Given **simulated events** $\tilde{x}_i \sim p(x | \phi')$ we can **construct the density** $p(\tilde{x} | \phi')$
 - this is an **approximation** of what we are after, the true $p(x | \phi)$
- Think of this as **MC integration**: with enough simulated events can construct approximate probability density

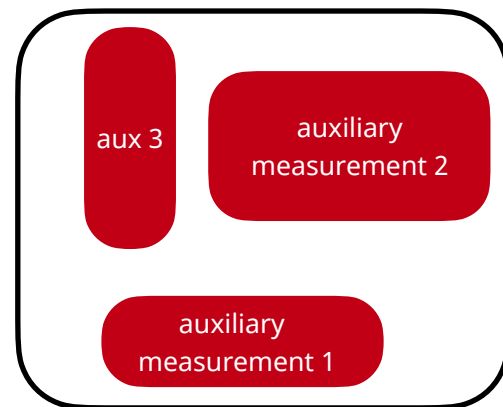


Auxiliary measurements

- We **know a lot about our detector** from existing **calibration measurements**
 - these typically take the form of a likelihood $L_a(\theta) = p(a | \theta)$
 - likelihood encodes information about calibration parameters θ
- Want to benefit from calibration to **control effect of systematic uncertainties**
 - achieved via multiplication with our likelihood: $L_x(\phi) \cdot L_{a_1}(\theta_1) \cdot L_{a_2}(\theta_2) \cdot \dots$

- It is **typically impractical** to combine all likelihoods in full detail, so we simplify the treatment
 - sometimes only have a **central value** for a parameter alongside some **notion of uncertainty**
 - e.g. provided by CP groups & accessed via CP algorithms
 - common to **approximate auxiliary measurements** with e.g. a single Gaussian
 - these then become the “**constraint terms**” in the likelihood $\prod_j c_j(a_j | \theta_j)$

auxiliary observables \vec{a}



Constraining nuisance parameters

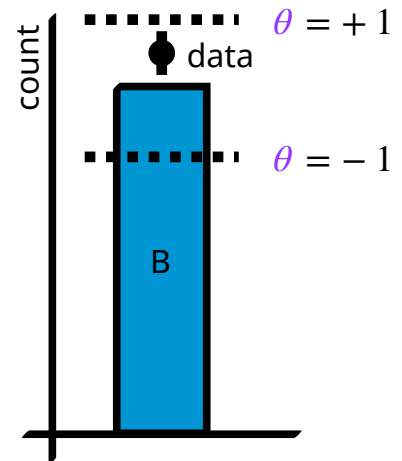
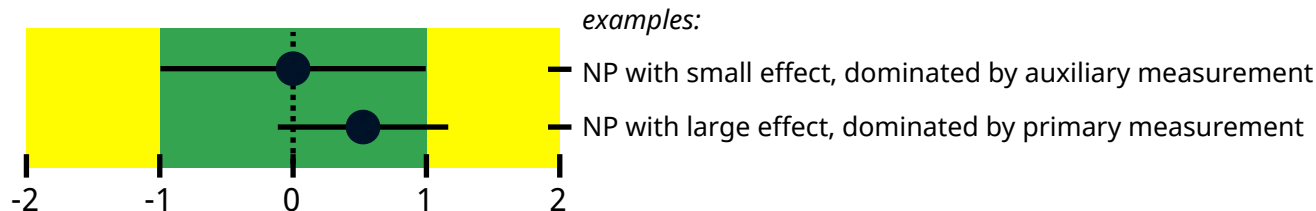
response function: $\theta = 1$
scales background up by 20%

$$p(n, a \mid \theta) = \text{Pois}(n \mid S + (1 + \theta/5)B) \cdot G(a \mid \theta)$$

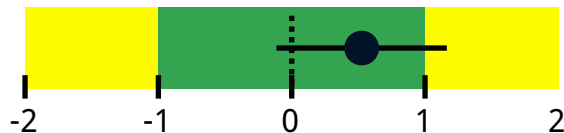
primary
measurement

auxiliary
measurement

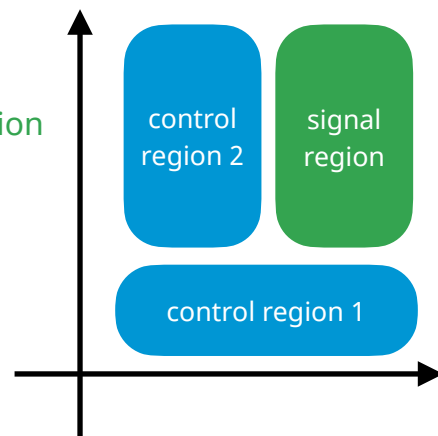
- Common mental picture: **auxiliary measurement** controls nuisance parameters θ
 - in a **profile likelihood** context, both terms are on equal footing!
- If effect of θ on model prediction is large, **constraints will arise** from primary term



Constraints and concerns



- **Constrained nuisance parameters:** primary observables provide control over $\vec{\theta}$
 - general concern: may underestimate uncertainties due to (local?) **model misspecification**
 - e.g. a bug which blows up effect of nuisance parameter only in control region
 - try to **locate & understand source of effect**
 - typical operation: replace single nuisance parameter by multiple parameters



Special consideration is given to the correlation of modelling uncertainties across different p_T^H bins, in order to provide the fit with enough flexibility to cover background mismodelling without biasing the signal extraction. The $t\bar{t} + \geq 1b$ NLO matching uncertainty is shown to depend on p_T^H and is therefore decorrelated across p_T bins in the SRs.

A HistFactory JSON workspace with pyhf

- **JSON** structure maps directly to workspace structure
 - highly human-readable!

```
{
  "channels": [
    {
      "name": "SR",
      "samples": [
        {
          "data": [10.0, 15.0],
          "modifiers": [
            {
              "data": null,
              "name": "mu",
              "type": "normfactor"
            }
          ],
          "name": "Signal"
        },
        {
          "data": [50.0, 45.0],
          "modifiers": [
            {
              "data": {"hi": 1.1, "lo": 0.9},
              "name": "Modeling_unc",
              "type": "normsys"
            }
          ],
          "name": "Background"
        }
      ]
    }
  ],
  "measurements": [
    {
      "config": {"parameters": [], "poi": "mu"},
      "name": "minimal_example"
    }
  ],
  "observations": [{"data": [60.0, 60.0], "name": "SR"}],
  "version": "1.0.0"
}
```

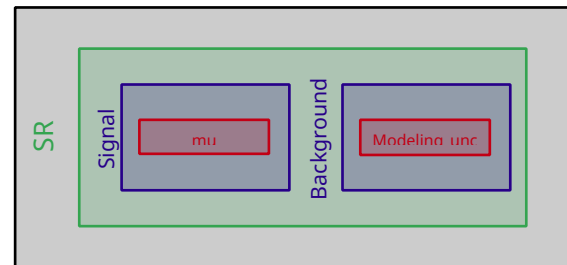
single channel → {

two samples

modifiers

← measurement configuration

← observed data



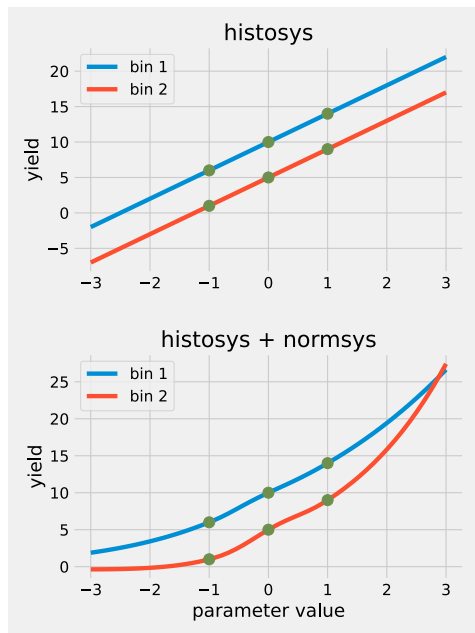
The HistFactory model: modifiers

- The **following modifiers are available** (pyhf [ROOT] names):
 - **normsys** [OverallSys]: changing event yield coherently across all bins of a sample in a channel
 - **histosys** [HistoSys]: correlated event yield variation across bins in a channel, but different effect per bin
 - **shapesys** [ShapeSys]: like **histosys**, but one independent parameter per bin (less frequently used)
 - **normfactor** [NormFactor]: free-floating normalization, linearly scaling event yields
 - **shapefactor** [ShapeFactor]: like **normfactor**, but one independent parameter per bin (e.g. data-driven fakes)
 - **statererror** [StatError]: MC statistical uncertainties (Barlow-Beeston “light”)
 - **lumi** [Lumi]: works like **normsys** plus extra features (can be replaced by **normsys**)
- Each **modifier is controlled by one or multiple parameters** ($\vec{k}, \vec{\theta}$)
- Modifiers other than **normfactor** and **shapefactor** have an **associated constraint term** ($c_j(a_j | \theta_j)$)
 - constraint term is Gaussian / Poisson, in some cases configurable
- **Template inter-/extrapolation method** can matter

Normalizing histosys modifiers

- Due to the use of **linear extrapolation**, **histosys** modifiers can cause **negative yield predictions**
 - example: [Gist](#)
 - (partial) solution: split overall channel normalization effect into correlated **normsys** [[OverallSys](#)]

exact match
where templates
are defined
(green points) by
design



pure histosys

```
"data": [10, 5],  
"modifiers": [  
  {  
    "data": {"hi_data": [14, 9], "lo_data": [6, 1]},  
    "name": "histosys_example",  
    "type": "histosys",  
  },  
],
```

correlated histosys + normsys

```
"data": [10, 5],  
"modifiers": [  
  {  
    "data": {  
      "hi_data": [9.1304347826, 5.8695652174],  
      "lo_data": [12.8571428571, 2.1428571429],  
    },  
    "name": "histosys_and_normsys",  
    "type": "histosys",  
  },  
  {  
    "data": {"hi": 1.5333333333, "lo": 0.4666666667},  
    "name": "histosys_and_normsys",  
    "type": "normsys",  
  },  
],
```