Boyang Zhang
(University of Hawaii)
June 2025 @ Virginia Tech

Statistics and Systematics

A brief introduction to the basic concepts and technical details

Contents

When we read a Belle II paper, we often find a table like this.

They are the uncertainties of the measurement.

Q: What do they mean and how can I get them?

Table V. Fractional contributions to the total uncertainty on the extracted value of $|V_{cb}|$. The sizes of the contributions are given relative to the central value.

Source		Uncertainty [%]
Statistical		0.9
Systematic		1.5
	$B^{0/+}$ lifetime	0.1
	Signal form factor	0.1
	$B \to D^* \ell\nu$ form factor	0.1
	$\mathcal{B}(B \to X_c \ell\nu)$	0.3
	$\mathcal{B}(D \to K\pi(\pi))$	0.5
	Tracking efficiency	0.5
	$N_{\Upsilon(4S)}$	0.7
	f_{00}/f_{+-}	0.1
	$f_{\mathcal{B}}$	0.4
	Background w modelling	0.3
	(E_Y^*, m_Y) reweighing	0.3
	Lepton identification	0.3
	Kaon identification	0.6
	Vertex fit χ^2 correction	0.3
	Simulation sample size	0.5
Theoretical		1.3
	Lattice QCD inputs	1.2
	Long-distance QED	0.5
Total		2.1

P.Horak, C. Schwanda
|V_cb| using B → D*lν at Belle II
https://authors.belle2.org/dir.pl?p=81

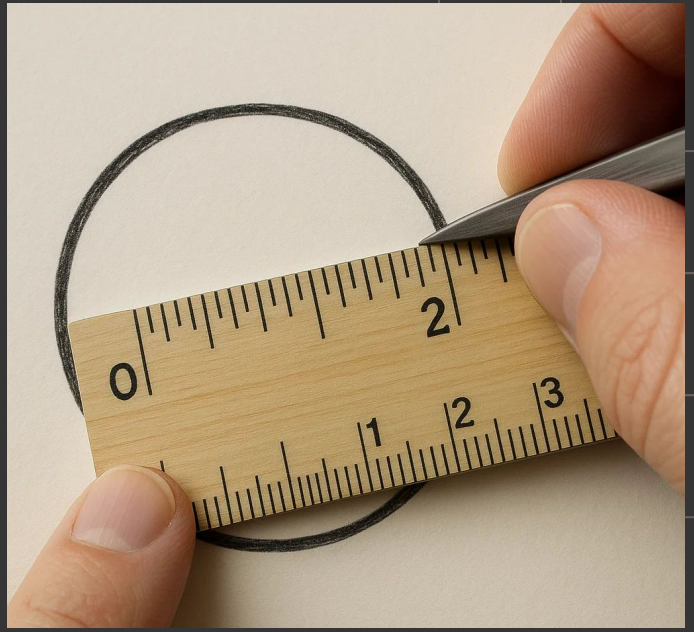
What are the uncertainties of a measurement?

↓

What is a measurement first?

A comparison between:

- A feature of the object of interest
- A model that represents our expectation of that feature



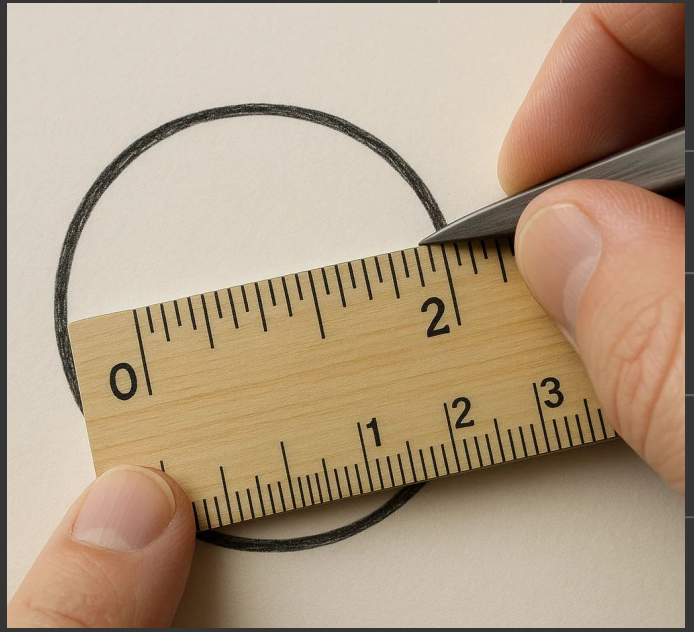
For example, measuring the diameter is also a comparison:

- Align the circle edge and ticks on the ruler
- Find the closest tick and read the number

These steps introduce uncertainties!

- Where exactly the circle edges are (from the object)
- Ticks are drawn at wrong place (from the model)

Q: What about in data analysis?



In data analysis, we compare

Object of interest: Data

Model: Statistical model

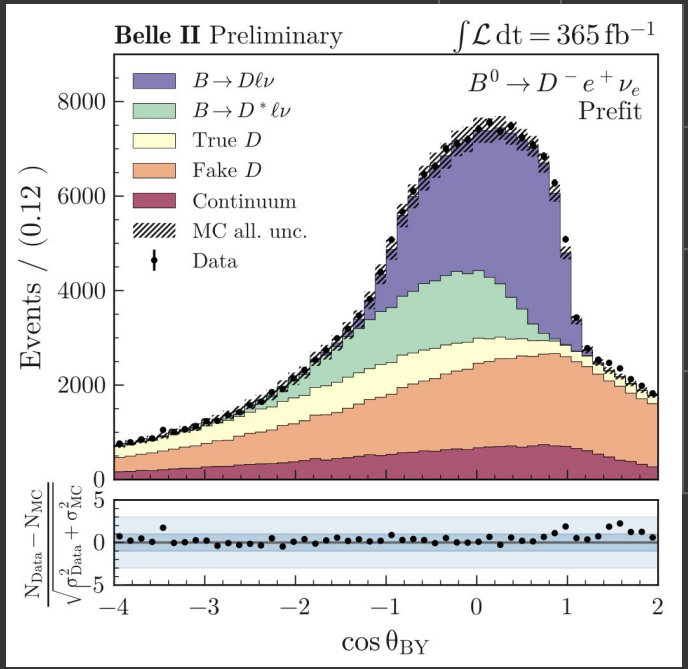
Math/Statistics quantifies the comparison

Naturally, both of them introduce uncertainties!

Uncertainty from data: statistical uncertainty

Uncertainty from model: systematic uncertainty

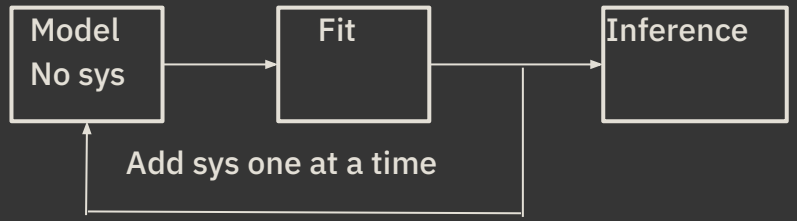
Q: How are they handled in the statistical inference?



P.Horak, C. Schwanda
|V_cb| using B → D*lν* at Belle II
https://authors.belle2.org/dir.pl?p=81

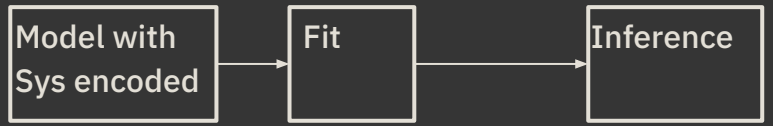
Statistical inference (e.g. confidence intervals, upper limits)

Traditionally:



$$f(\boldsymbol{n}, \boldsymbol{a} \,|\, \boldsymbol{\eta}, \boldsymbol{\chi}) = \underbrace{\prod_{c \in \text{ channels}} \prod_{b \in \text{ bins}_c} \text{Pois}\left(n_{cb} \,|\, \nu_{cb}\left(\boldsymbol{\eta}, \boldsymbol{\chi}\right)\right)}_{\substack{\text{Simultaneous measurement} \\ \text{of multiple channels}}} \underbrace{\prod_{\chi \in \boldsymbol{\chi}} c_\chi(a_\chi \,|\, \chi)}_{\substack{\text{constraint terms} \\ \text{for “auxiliary measurements”}}} ,$$

With pyhf:



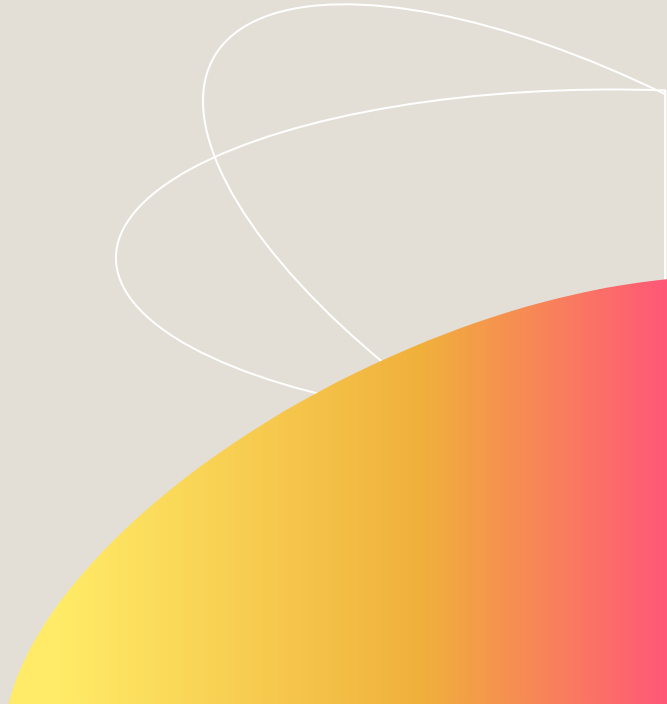
Q: How to encode the systematics into the model?

What is a pyhf model

- the expected distribution of (a) feature in data
  - both shape and normalization for pyhf
- Histograms
  - i.e. the model inputs are the bin counts (binned by user)

Q: How do we typically build one?

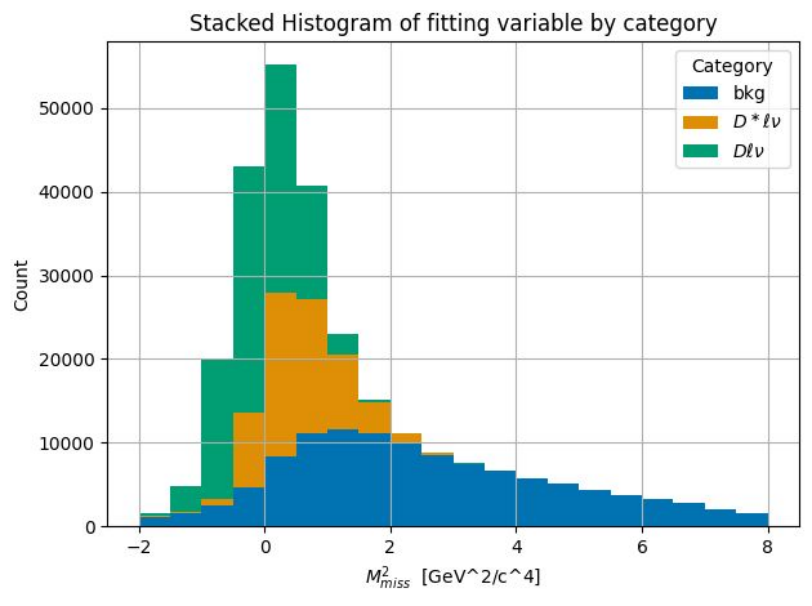
A: MC or data driven (sidebands, or controls)



Hands-on exercise: build a model for a B → D$l\nu$ analysis

- **Signal events have 1 missing neutrino**
- Background events have more missing particles
- Choose M^2_{miss} to be the fitting variable
- 20 uniform bins

$ cp /home/belle/zhangboy/B2SW/2025_VirginiaTech/Stat_Sys_hands_on.ipynb <your directory>



Systematic uncertainty:

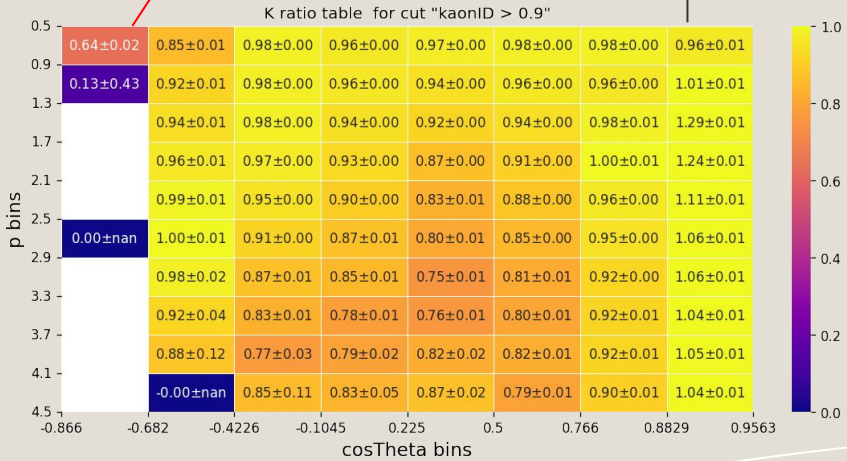
- Bin count uncertainty due to MC statistical fluctuation or MC mismodelling.
  - E.g. if the MC sample is small e.g. PID cuts

The 2 systematics covered in this hands-on exercise

1. Lepton ID (electron), difficult:
    a. Cut on PID variables in the reconstruction → Data/MC disagreement
    b. Calibrate the electron (based on p and θ) by using control samples
        i. E.g. ee → ee for efficiency; K_s → ππ for fake rate
        ii. Weight table → correction weights for each electron
        iii. The uncertainty of the correction weights are the source for these systematics
    c. Propagate the uncertainty from the correction weights to the fitting variables (bin counts)
    d. Determine if this uncertainty is correlated across bins or uncorrelated
    e. Feed the correlated portion to the `histosys` and `normsys` in pyhf
    f. Discard the uncorrelated part (small) or feed it to the `staterror` in pyhf

2. MC stat uncertainty, simpler:
    a. Calculate the poisson error with weights
    b. Feed them to the `staterror` in pyhf

1. Lepton ID (electron), difficult:
   a. Cut on PID variables in the reconstruction → Data/MC disagreement
   b. Calibrate the electron (based on p and θ) by using control samples
      i. E.g. ee → ee for efficiency; K_s → ππ for fake rate
      ii. Weight table → correction weights for each electron
      iii. The uncertainty of the correction weights are the source for this systematics



	ell_p	ell_theta	B0_CMS3_weMissM2	template	ell_eID_eff_weight
0	1.205724	0.972190	4.148000	bkg	1.000000
1	1.299124	0.815572	1.837669	bkg	0.991163
2	1.415534	1.193000	1.463991	bkg	1.003654
3	0.362656	1.274201	5.461838	bkg	0.987793
4	1.451889	1.221380	1.775905	bkg	1.006147
...	...	...	...	...	...
272603	1.984586	0.495626	0.248509	$D^*\ell\nu$	0.984110
272604	1.054710	1.565140	1.685410	$D^*\ell\nu$	1.006147
272605	2.768823	0.861458	-0.224378	$D^*\ell\nu$	0.997852
272606	1.313779	2.005175	0.339968	$D^*\ell\nu$	0.984630
272607	0.684441	1.459065	1.337592	$D^*\ell\nu$	1.002800

C. Propagate the uncertainty from the correction weights to the fitting variables (bin counts)

Propagation of PID-Correction Statistical Uncertainty

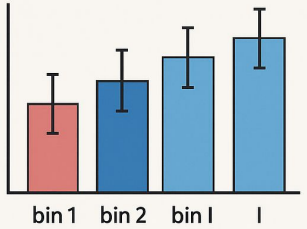
DataFrame of events

event index	fitting variable	toy weight
1	0.23	0.79
2	0.47	0.80
3	0.85	0.82
…	0.82	0.77
N	…	*M*
1	0.79	0.80
2	0.76	0.82
…	0.82	0.77
N	…	

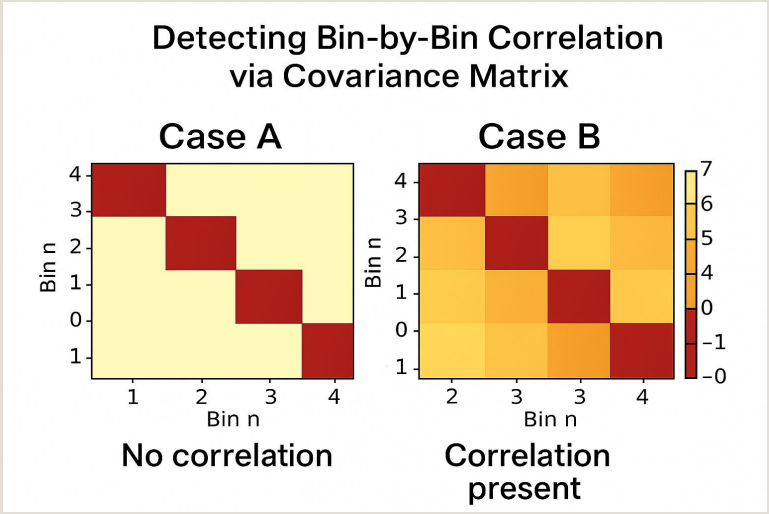
Apply weights, histogram

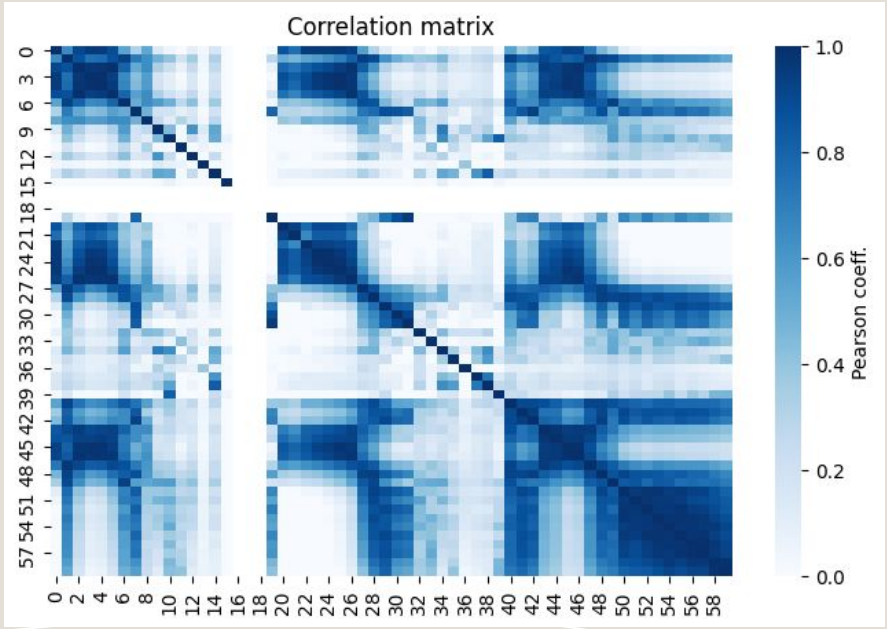
Histogram bins

Histogram bins	toy event counts	
bin 1	25	24
bin 2	26	23
bin I	…	23



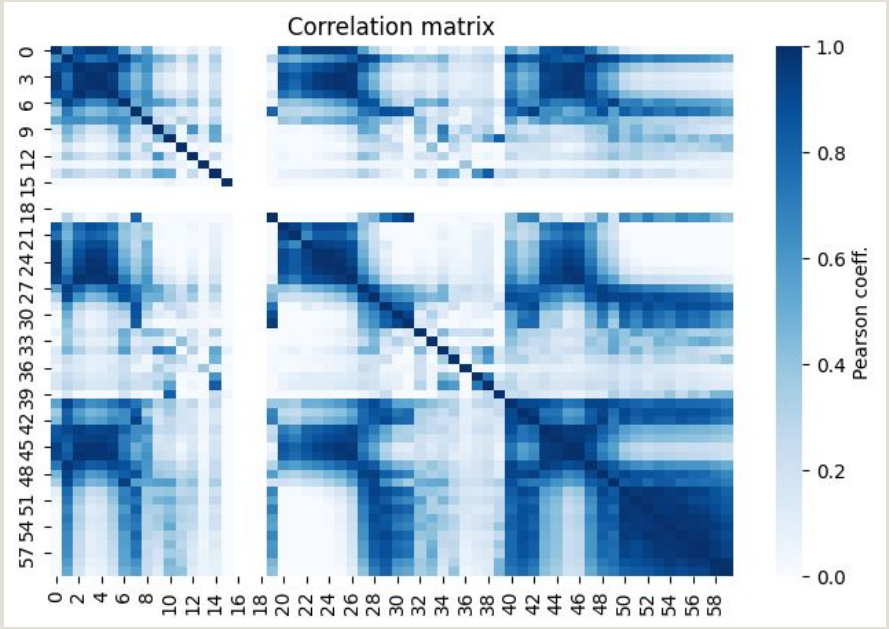
D1. Determine if this uncertainty is correlated or uncorrelated across bins and fitting components



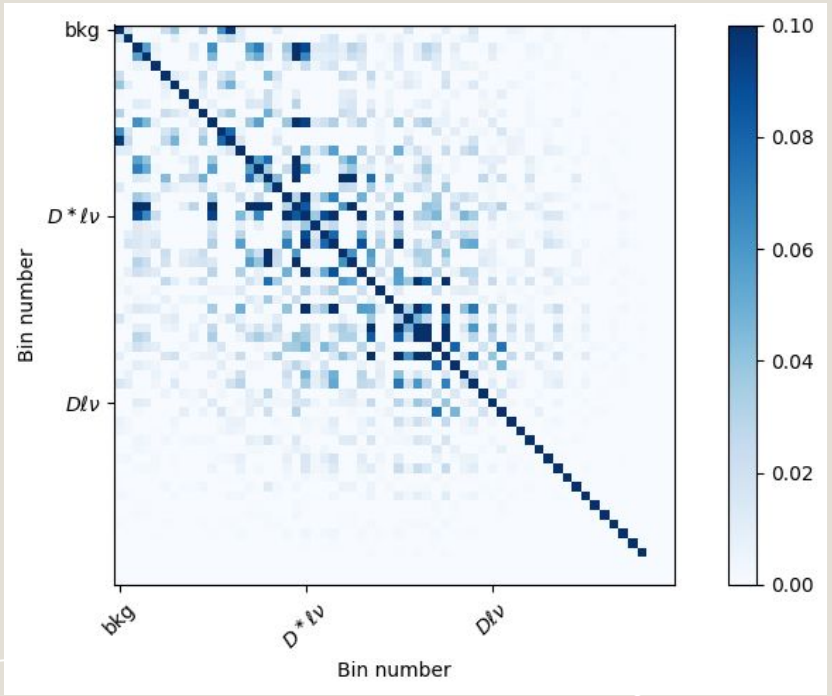


D2. Solve for eigenvectors (bin-by-bin correlated uncertainty)

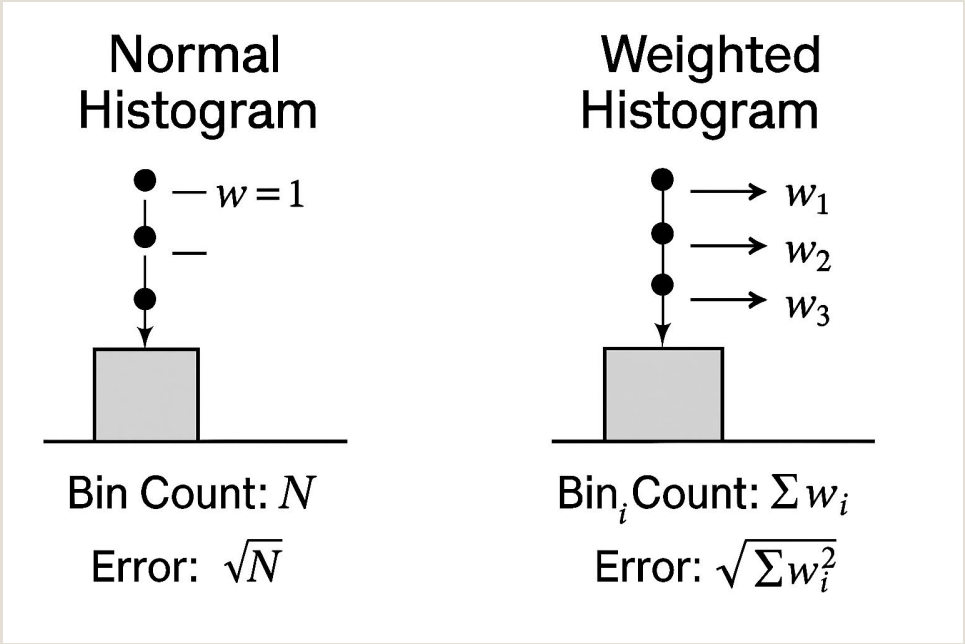
The full correlation matrix



The truncated matrix (with first n eigenvectors)



2. MC stat uncertainty: Calculate the Poisson error with weights



Example:
If $w_1 = w_2 = \ldots = w_i = W \neq 1$

$N' = W \times N$

$\delta N' = W \times \delta N$

$\delta N' = W \times \sqrt{N}$ assume poisson

$\delta N' = \sqrt{(W^2 \times N)}$

Create a pyhf model

```
# create the pyhf workspace to store the model
channels = []
observations = []
measurements = [{"name": "D_ell_nu", "config": {"poi": "$D\\ell\\nu$_norm", "parameters": []}}]
version = "1.0.0"

####### Store 360/fb MC as real data to be fitted ######
observations.append({
    'name': f'channel_1',
    'data': data.round(0).tolist()  # Extract nominal values from uncertainties
})

###### Store 1/ab MC as the model #######
# Initialize channel structure
channels.append({
    'name': f'channel_1',
    'samples': []
})

# Loop over each fitting component in the channel
for sample_index, sample_name in enumerate(category_order):
    # Add the nominal template data for the sample
    channels[0]['samples'].append({
        'name': sample_name,
        'data': hists[sample_name].tolist(),
```

$$f(\boldsymbol{n}, \boldsymbol{a} \,|\, \boldsymbol{\eta}, \boldsymbol{\chi}) = \underbrace{\prod_{c \in \text{ channels}} \prod_{b \in \text{ bins}_c} \text{Pois}\left(n_{cb} \,|\, \nu_{cb}\left(\boldsymbol{\eta}, \boldsymbol{\chi}\right)\right)}_{\substack{\text{Simultaneous measurement} \\ \text{of multiple channels}}} \underbrace{\prod_{\chi \in \boldsymbol{\chi}} c_\chi(a_\chi \,|\, \chi)}_{\substack{\text{constraint terms} \\ \text{for “auxiliary measurements”}}} ,$$

Add the modifiers

```python
# Loop over each fitting component in the channel
for sample_index, sample_name in enumerate(category_order):
    # Add the nominal template data for the sample
    channels[0]['samples'].append({
        'name': sample_name,
        'data': hists[sample_name].tolist(),
        'modifiers': [
            {
                'name': sample_name+'_norm',
                'type': 'normfactor',
                'data': None  # Normalization factor modifier
            }
        ]
    })

    # Add uncertainty modifiers for MC statistical errors
    channels[0]['samples'][sample_index]['modifiers'].append({
        'name': 'MCstat_ch1',
        'type': 'staterror',
        'data': MC_stat_error[sample_name].tolist()
    })
```

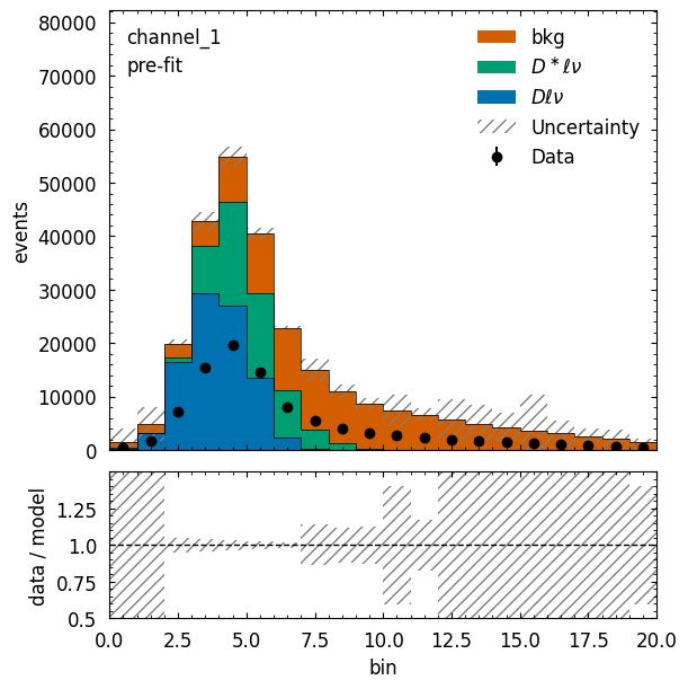
$$f(\boldsymbol{n}, \boldsymbol{a} \,|\, \boldsymbol{\eta}, \boldsymbol{\chi}) = \underbrace{\prod_{c \in \text{ channels}} \prod_{b \in \text{ bins}_c} \text{Pois}\left(n_{cb} \,|\, \nu_{cb}\left(\boldsymbol{\eta}, \boldsymbol{\chi}\right)\right)}_{\substack{\text{Simultaneous measurement} \\ \text{of multiple channels}}} \underbrace{\prod_{\chi \in \boldsymbol{\chi}} c_\chi(a_\chi | \, \chi)}_{\substack{\text{constraint terms} \\ \text{for “auxiliary measurements”}}},$$

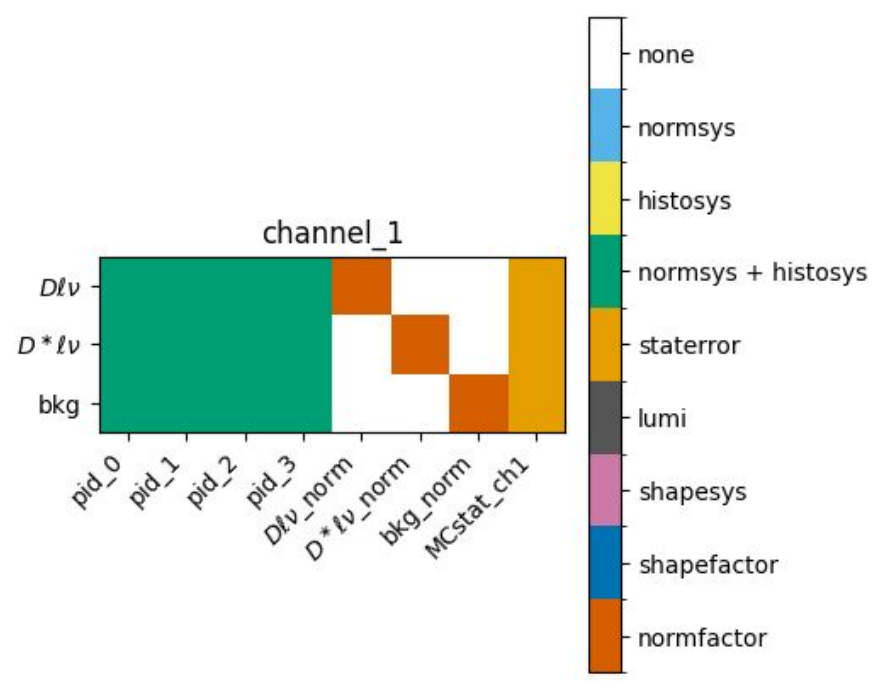
Modifiers and Constraints

Description	Modification	Constraint Term c_χ	Input
Uncorrelated Shape	$\kappa_{scb}(\gamma_b) = \gamma_b$	$\prod_b \text{Pois}\left(r_b = \sigma_b^{-2} \,\middle|\, \rho_b = \sigma_b^{-2}\gamma_b\right)$	σ_b
Correlated Shape	$\Delta_{scb}(\alpha) = f_p\left(\alpha \middle| \Delta_{scb,\alpha=-1}, \Delta_{scb,\alpha=1}\right)$	$\text{Gaus}\left(a = 0 \middle| \alpha, \sigma = 1\right)$	$\Delta_{scb,\alpha=\pm1}$
Normalisation Unc.	$\kappa_{scb}(\alpha) = g_p\left(\alpha \middle| \kappa_{scb,\alpha=-1}, \kappa_{scb,\alpha=1}\right)$	$\text{Gaus}\left(a = 0 \middle| \alpha, \sigma = 1\right)$	$\kappa_{scb,\alpha=\pm1}$
MC Stat. Uncertainty	$\kappa_{scb}(\gamma_b) = \gamma_b$	$\prod_b \text{Gaus}\left(a_{\gamma_b} = 1 \middle| \gamma_b, \delta_b\right)$	$\delta_b^2 = \sum_s \delta_{sb}^2$
Luminosity	$\kappa_{scb}(\lambda) = \lambda$	$\text{Gaus}\left(l = \lambda_0 \middle| \lambda, \sigma_\lambda\right)$	$\lambda_0, \sigma_\lambda$
Normalisation	$\kappa_{scb}(\mu_b) = \mu_b$		
Data-driven Shape	$\kappa_{scb}(\gamma_b) = \gamma_b$		

Visualize the model before fitting to data (pre-fit)

The bkg_norm should actually be constrained, i.e. it should be a nuisance parameter (systematics)





Specify the fitting parameters

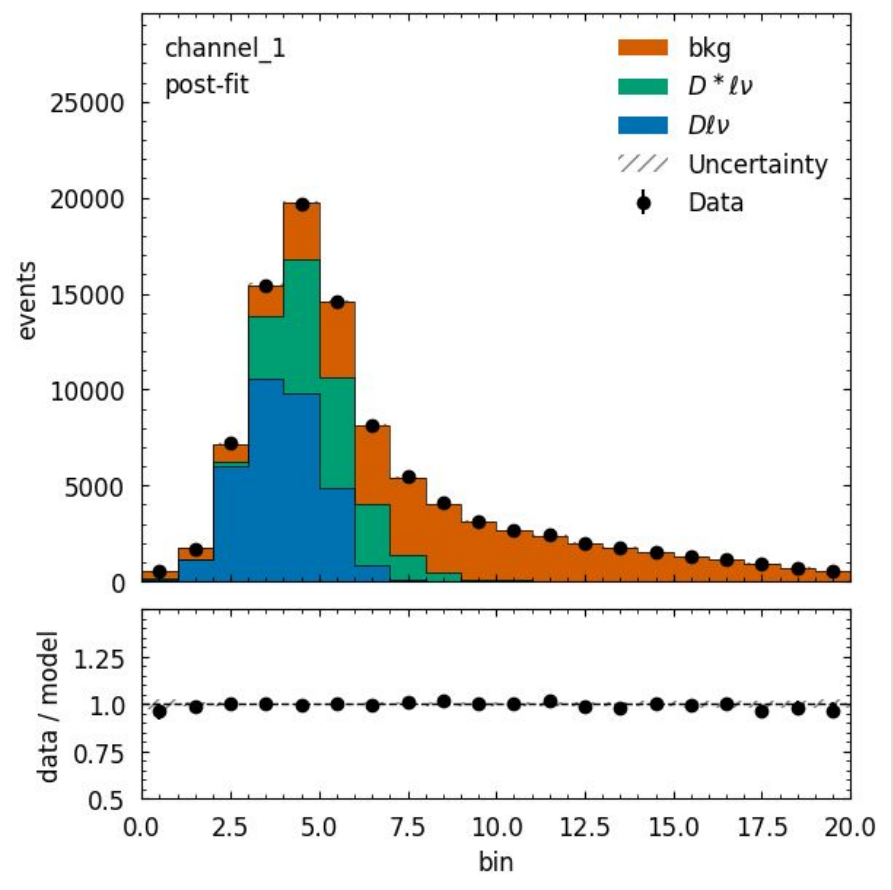
```
    # Define parameter bounds based on whether it's a background or signal sample
    if sample_name.startswith('bkg'):
        par_config = {"name": sample_name+'_norm', "bounds": [[0, 2]], "inits": [1.0], "fixed":False}
    else:
        par_config = {"name": sample_name+'_norm', "bounds": [[-5, 5]], "inits": [1.0]}

    # Add parameter configuration if it doesn't already exist
    if par_config not in measurements[0]['config']['parameters']:
        measurements[0]['config']['parameters'].append(par_config)

# Construct the final workspace dictionary
spec = {
    'channels': channels,
    'measurements': measurements,
    'observations': observations,
    'version': version
}
# cabinetry.workspace.save(spec, 'workspace_1.json')
```

Fit data with our model

```
INFO : fit: 478 :  performing maximum likelihood fit
INFO : _fit_model_pyhf: 108 :  Migrad status:
```

Migrad	
FCN = 70.6 EDM = 6.86e-05 (Goal: 0.0002)	Nfcn = 953
Valid Minimum	Below EDM threshold (goal x 10)
No parameters at limit	Below call limit
Hesse ok	Covariance accurate

```
INFO : print_results: 35 :  fit results (with symmetric uncertainties):
INFO : print_results: 38 :  pid_0                =  0.5715 +/- 0.8053
INFO : print_results: 38 :  pid_1                =  0.4480 +/- 0.8799
INFO : print_results: 38 :  pid_2                =  0.1071 +/- 0.9854
INFO : print_results: 38 :  pid_3                = -0.1065 +/- 0.9903
INFO : print_results: 38 :  $D\ell\nu$_norm       =  0.3615 +/- 0.0043
INFO : print_results: 38 :  $D^\ast\ell\nu$_norm =  0.3580 +/- 0.0077
INFO : print_results: 38 :  bkg_norm             =  0.3591 +/- 0.0028
INFO : print_results: 38 :  MCstat_ch1[0]        =  0.9866 +/- 0.0226
INFO : print_results: 38 :  MCstat_ch1[1]        =  0.9968 +/- 0.0126
INFO : print_results: 38 :  MCstat_ch1[17]       =  0.9877 +/- 0.0167
INFO : print_results: 38 :  MCstat_ch1[18]       =  0.9940 +/- 0.0191
INFO : print_results: 38 :  MCstat_ch1[19]       =  0.9884 +/- 0.0219
INFO : _goodness_of_fit: 408 :  calculating goodness-of-fit
INFO : _goodness_of_fit: 427 :  p-value for goodness-of-fit test: 87.38%
```



Statistical Inference (not covered this time)

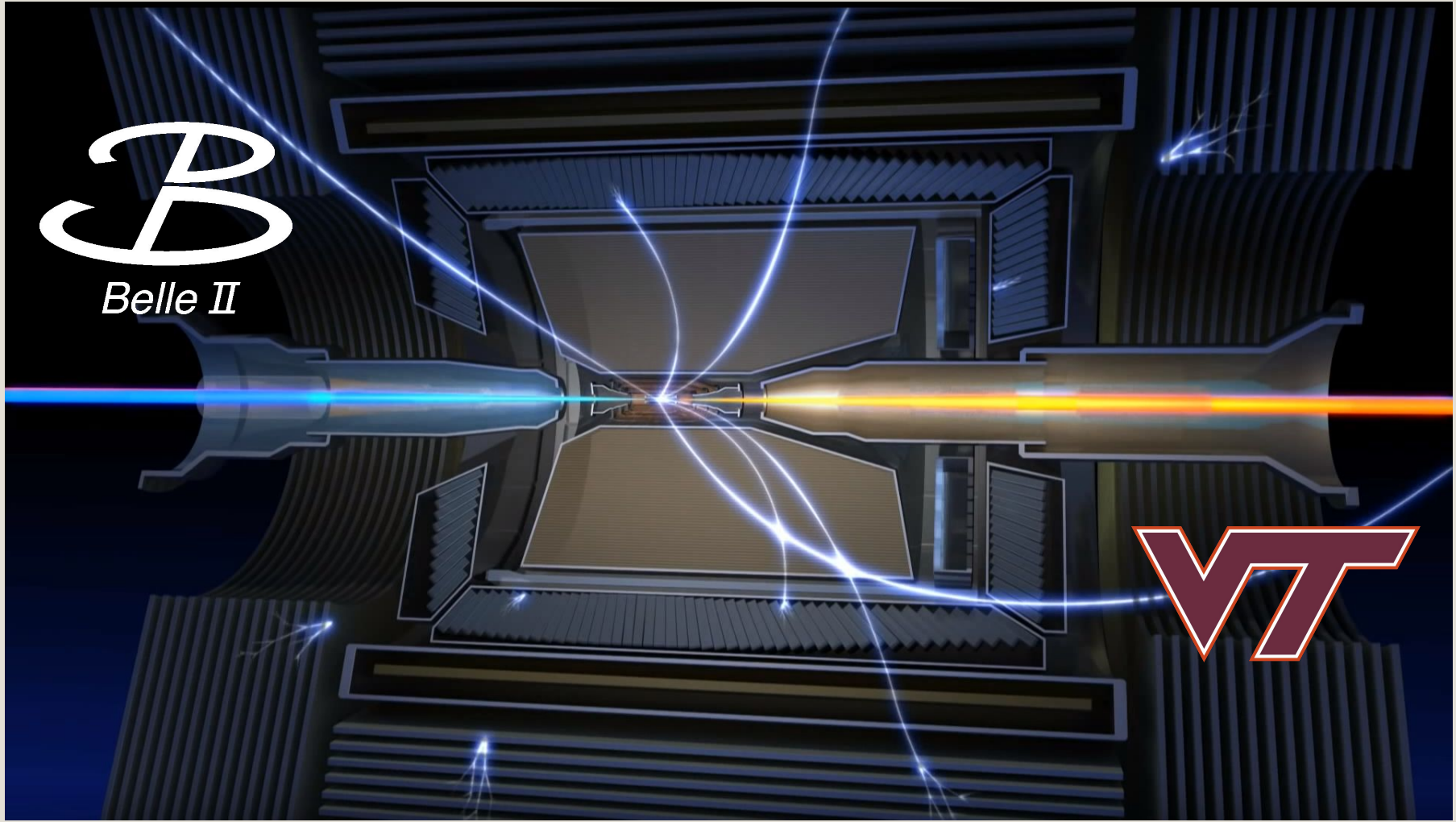
Goal	Typical questions you must answer	Core numbers (statistics) you quote
Precision measurement (POI + many nuisances)	*"What is the best value and uncertainty of the parameter?"* *"How well does my model reproduce the data?"*	• **Point estimate** (best fit, usually MLE) • **68 % (and often 95 %) confidence / credible interval** → Δ (–2 ln L)=1, 3.84 or MINOS errors • **Covariance / correlation matrix** of all fitted parameters • **Pulls & impacts** of the nuisance parameters (ranking plot) • **Goodness-of-fit** test: χ^2/ndf, saturated-likelihood ratio, Kolmogorov–Smirnov, etc.

Hypothesis testing(search, exclusion, discovery)	*"Is the null hypothesis disfavoured?"*
	
	"What's the smallest signal I can rule out?"
		• **Test statistic** (profile-likelihood ratio)
		  ○ q_0 for discovery ($\mu = 0$)
		  ○ q_μ for exclusion/limits
		• **p-value** of the null (or CL_s for limits)
		• **Significance $Z = \Phi^{-1}(1-p)$** (local & maybe global)
		• **Upper limit** on the POI at 95 % CL (or "5 σ discovery reach")

Summary

1. Data uncertainty → statistical uncertainty
2. Model uncertainty → systematic uncertainty
3. Typically each systematic requires (at least one) control/sideband sample to study
4. Pyhf provides a nice way to encode systematics into the fitting model
5. Statistical inference with pyhf fits all parameters at once
   a. Both statistical and systematic
6. Links to previous relevant hands-on sessions or documentations
   a. Systematics correction framework doc and hands-on
   b. Pyhf hands-on
   c. Iminuit and Zfit hands-on

Thank you


Belle II
VT