Understanding discrepancies in tracking efficiencies

Martin Tat

Heidelberg University

16th July 2025

UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

Introduction

I have had a look at the work on tracking efficiencies by Rowina and Maurice

- Some discrepancy between different tag-and-probe methods:
  1. Combined: VeloMuon $\times$ Downstream
     - VeloMuon: Determine SciFi efficiency
     - Downstream: Determine Velo efficiency
  2. MuonUT: Determine long track efficiency
- Presentation today:
  1. I have looked at this with fresh eyes, and I have some ideas about what to study in more detail
     - Thanks to Rowina for a well written thesis, which simplifies this detective work
  2. Today I will show work produced by TrackCalib2
     - Thanks for Maurice for all the assistance
  3. Specifically, today I will show some studies on fit biases

Tag-and-probe method

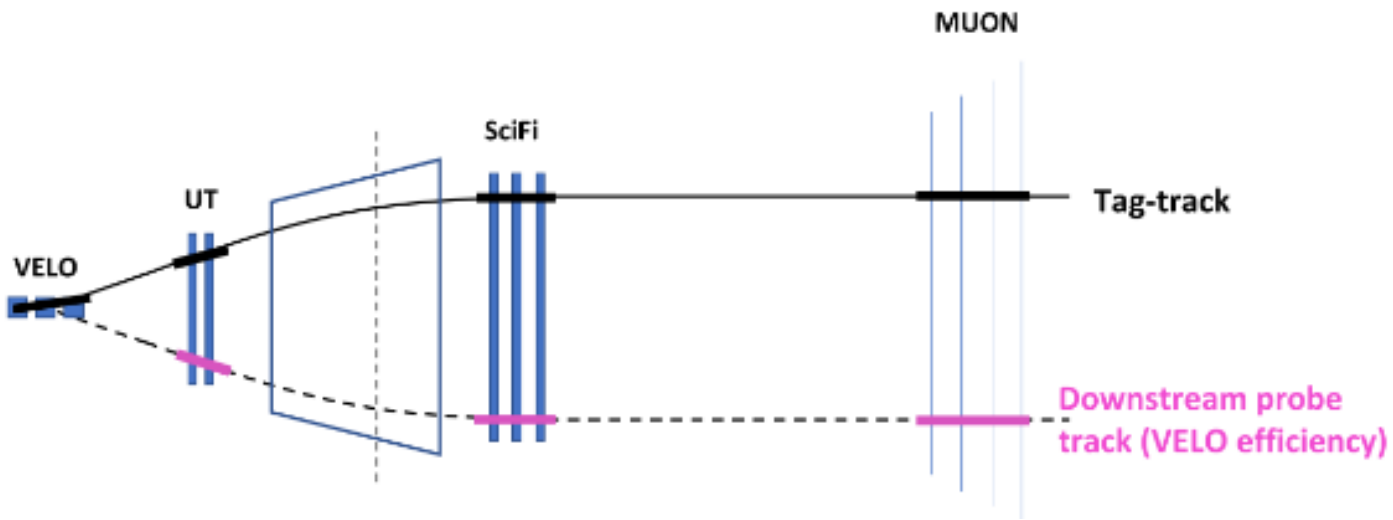
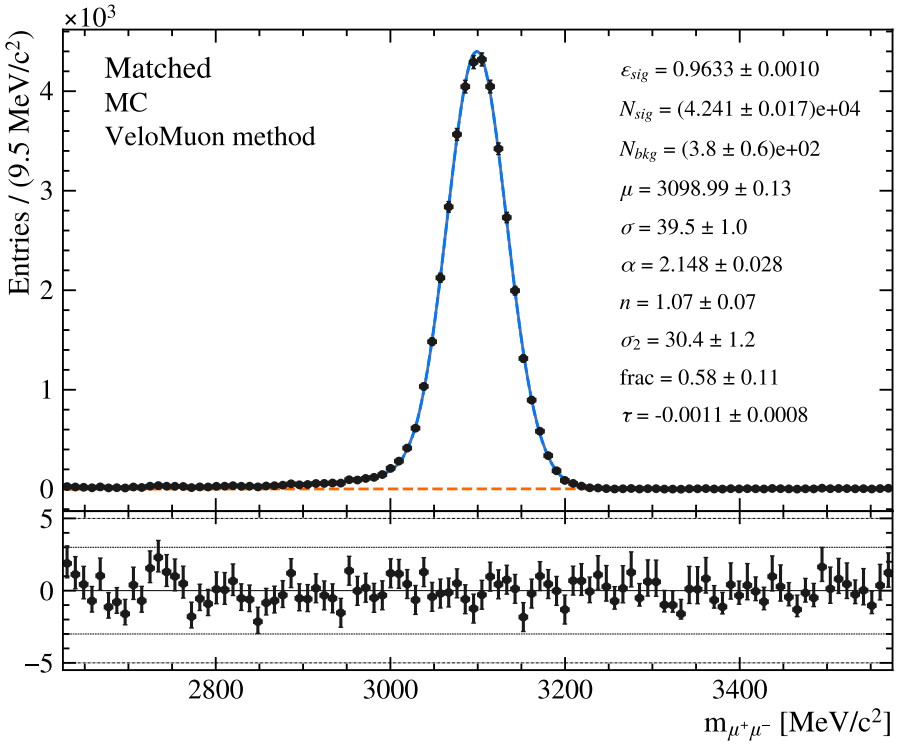


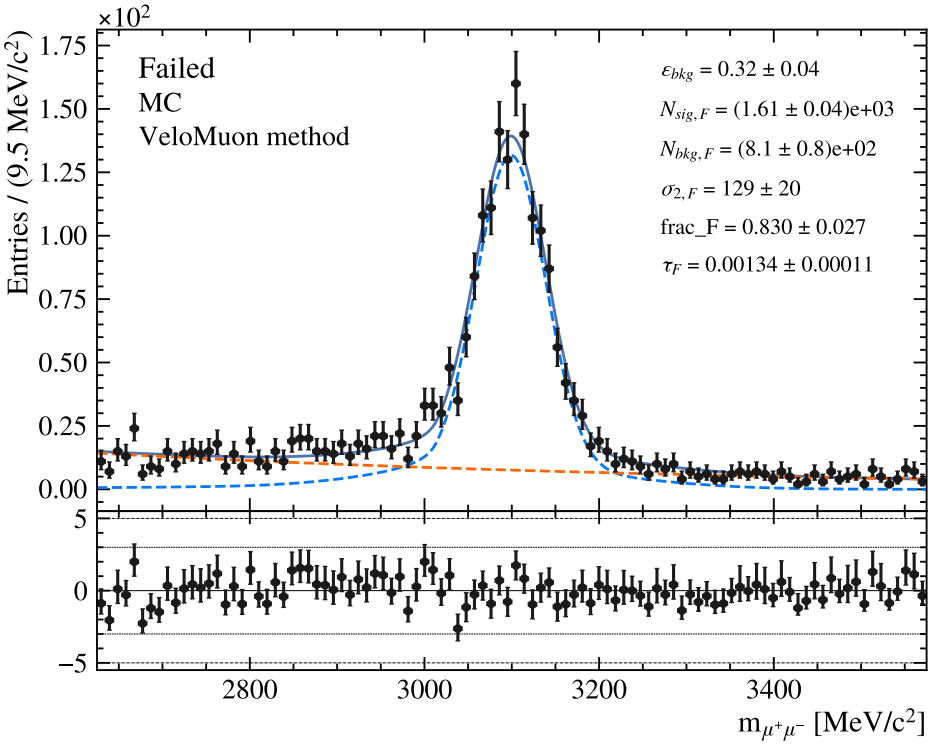
Figure from Rowina's thesis

- Fully reconstruct one muon from $J/\psi \to \mu^+\mu^-$
- Partially reconstruct the other muon
- Match hits in specific sub-detector with partially reconstructed track

$$\epsilon_{\text{track}} = \frac{N_{\text{matched}}}{N_{\text{matched}} + N_{\text{failed}}}$$

TrackCalib2

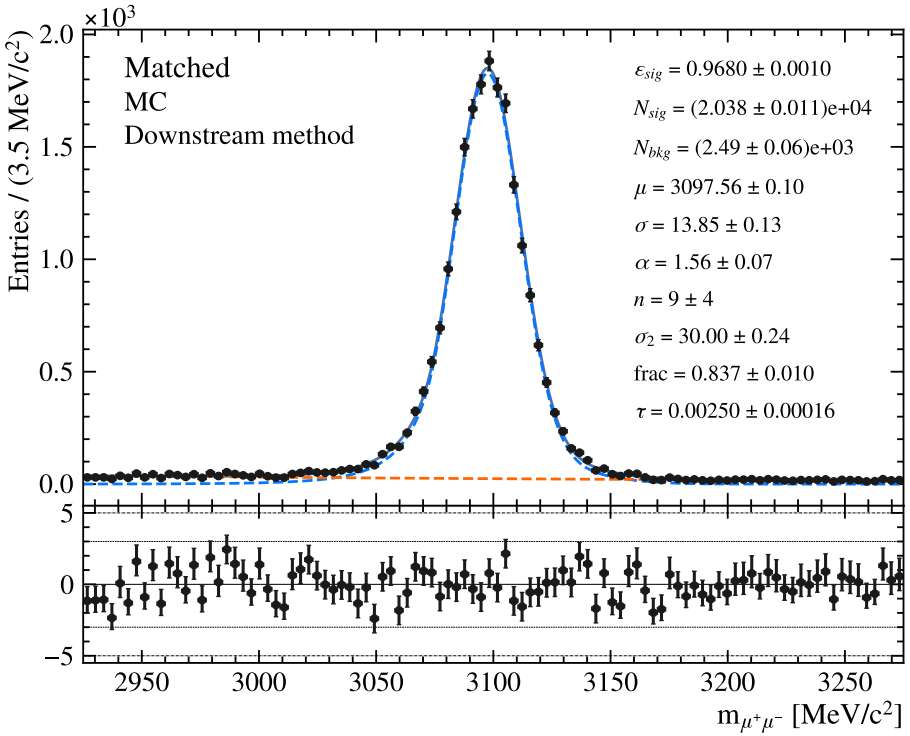


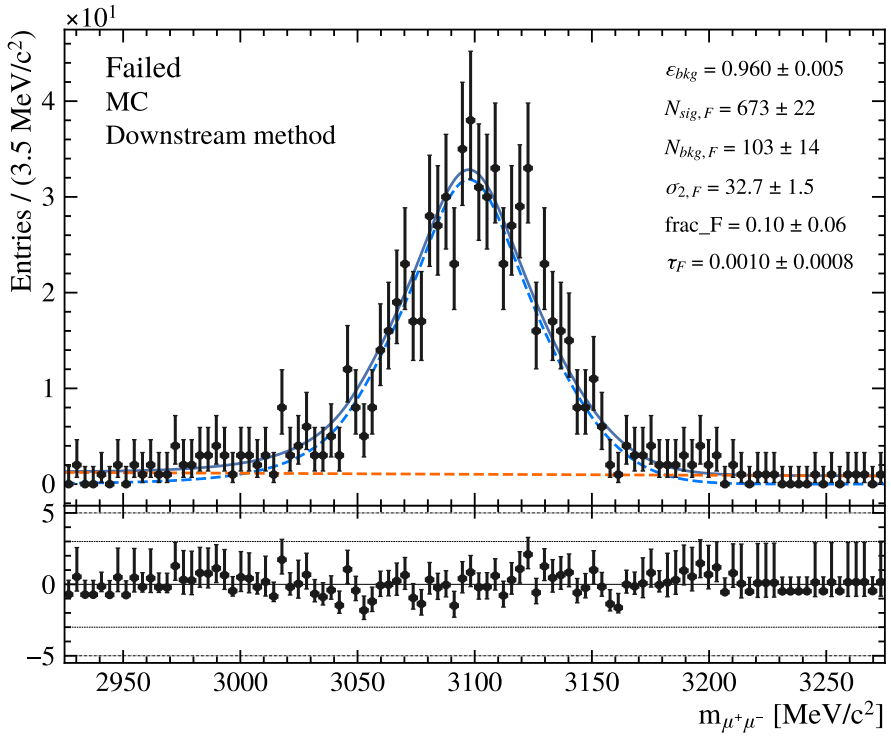


VeloMuon (p, η) bin $(0, 0)$

- I tried running the default fits with TrackCalib2 on MC...
- ... and things are working more or less out-of-the-box!
- The fit outputs ϵ_{track} directly as a fit parameter

TrackCalib2

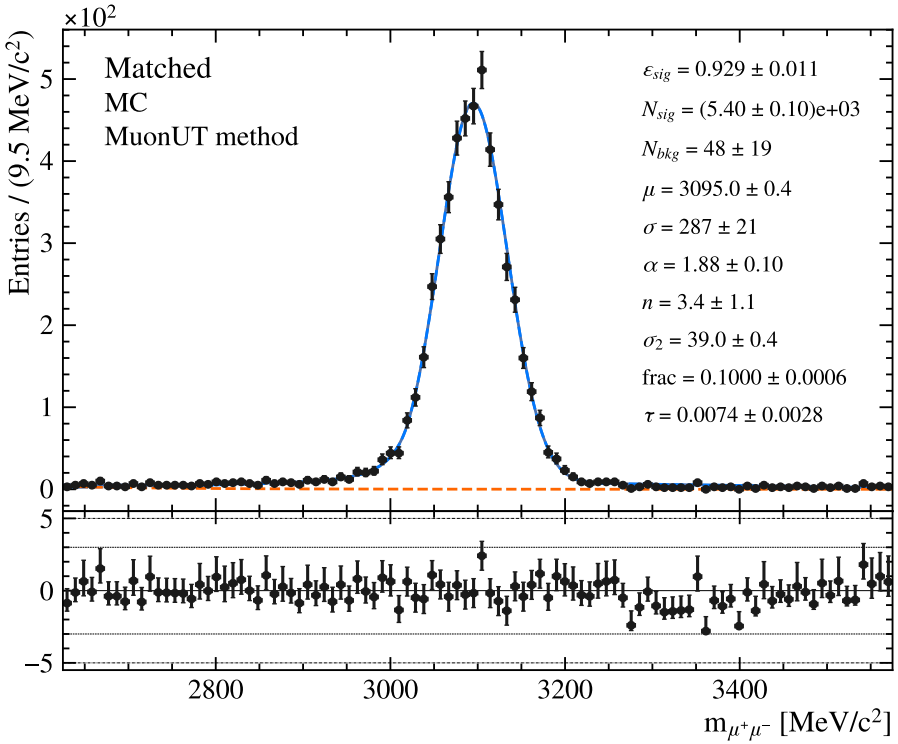


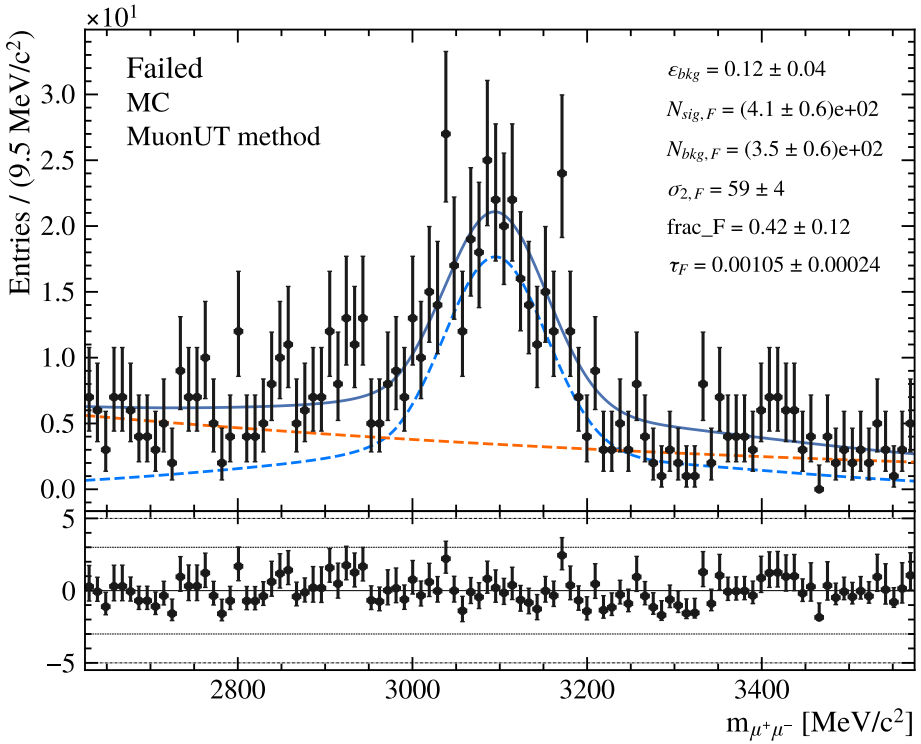


Downstream (p, η) bin $(0, 0)$

- I tried running the default fits with TrackCalib2 on MC...
- ... and things are working more or less out-of-the-box!
- The fit outputs ϵ_{track} directly as a fit parameter

TrackCalib2

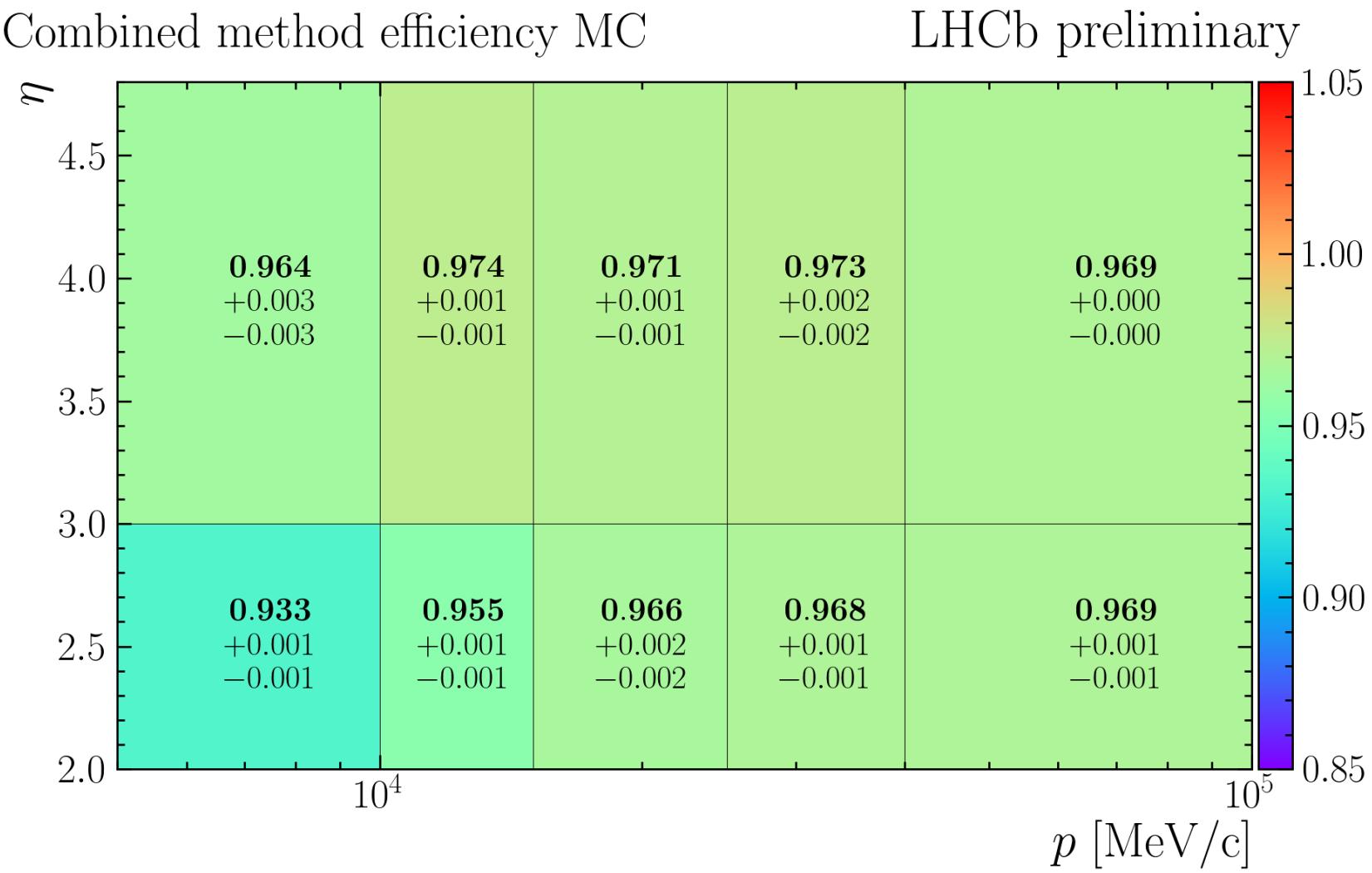




MuonUT (p, η) bin $(0, 0)$

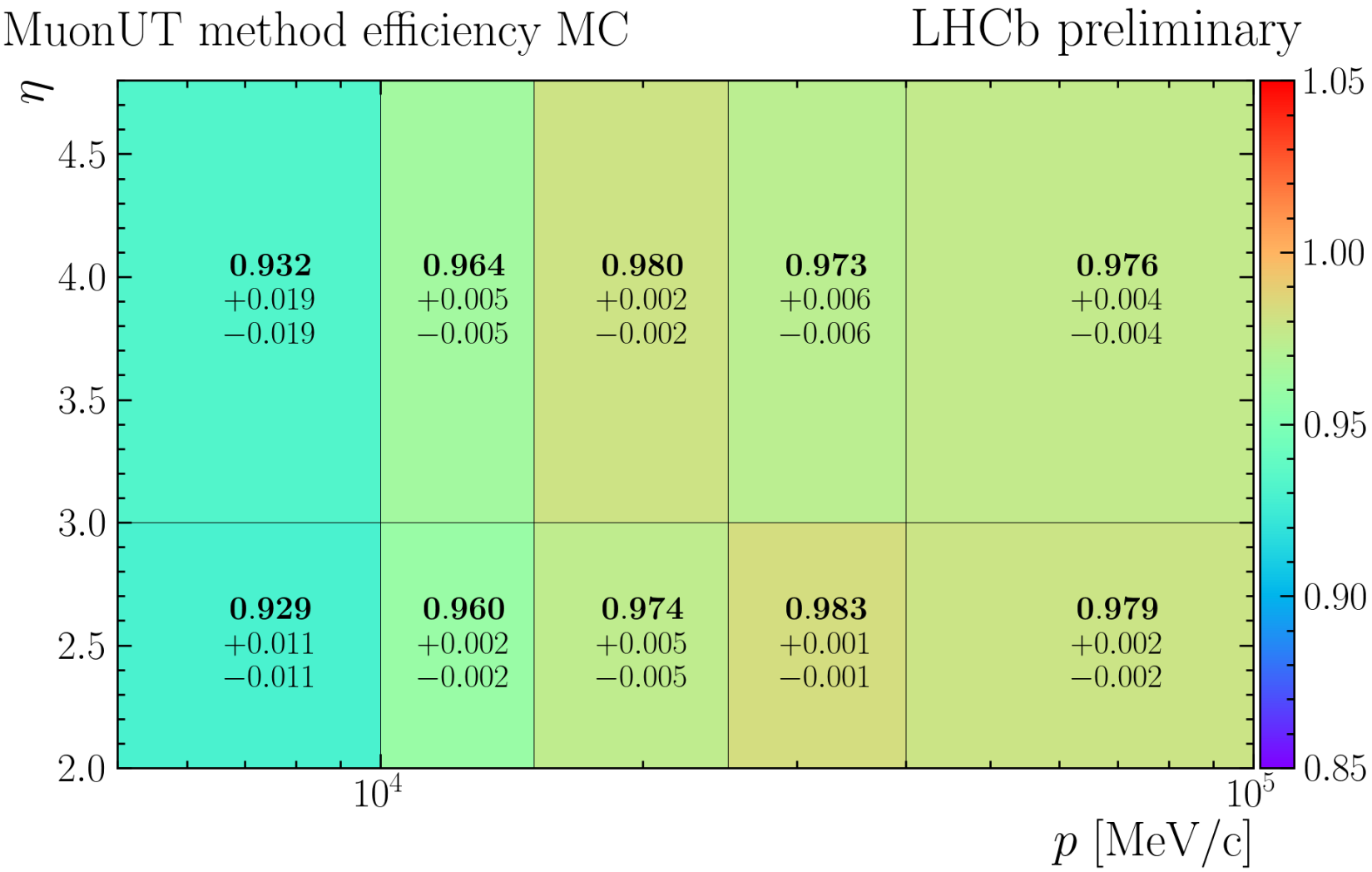
- I tried running the default fits with TrackCalib2 on MC...
- ... and things are working more or less out-of-the-box!
- The fit outputs ϵ_{track} directly as a fit parameter

TrackCalib2



Track efficiencies with Combined method produced by TrackCalib2

TrackCalib2



Track efficiencies with MuonUT method produced by TrackCalib2

Track efficiencies

Some discrepancies found...:

p bin	η bin	Difference (10^{-2})
0	0	0.4 ± 1.1
1	0	-0.5 ± 0.2
2	0	-0.8 ± 0.5
3	0	-1.5 ± 0.1
4	0	-1.0 ± 0.2
0	1	3.2 ± 1.9
1	1	1.0 ± 0.5
2	1	-0.9 ± 0.2
3	1	0.0 ± 0.6
4	1	-0.7 ± 0.4

Discrepancies in track efficiencies

Do we care about these discrepancies?

- Yes: Some are statistically significantly different from zero
- No: Method is not perfect, expect $\mathcal{O}(1\%)$ differences
- Yes: We want to keep systematics under control (roughly 2%)
- No: Most of the differences are around 1%, except for one bin with a large uncertainty

Fitting the track efficiency

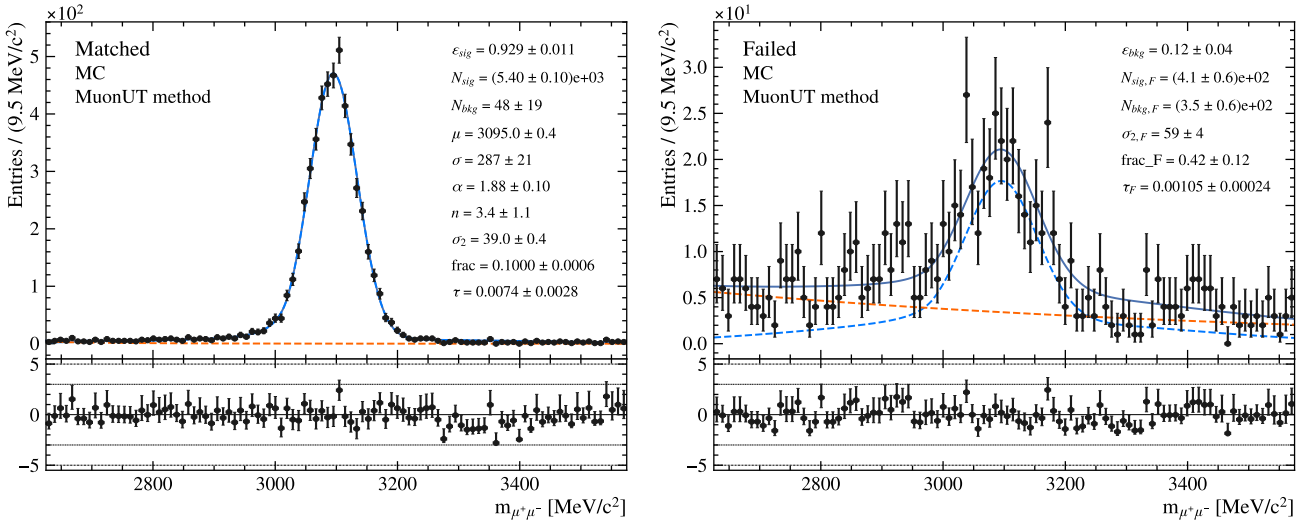
$$\epsilon_{\text{track}} = \frac{N_{\text{matched}}}{N_{\text{matched}} + N_{\text{failed}}}$$

- Fit total yield $N_{\text{tot}} = N_{\text{matched}} + N_{\text{failed}}$ and ϵ_{track} directly
- N_{tot} and ϵ_{track} can be related to the yields:

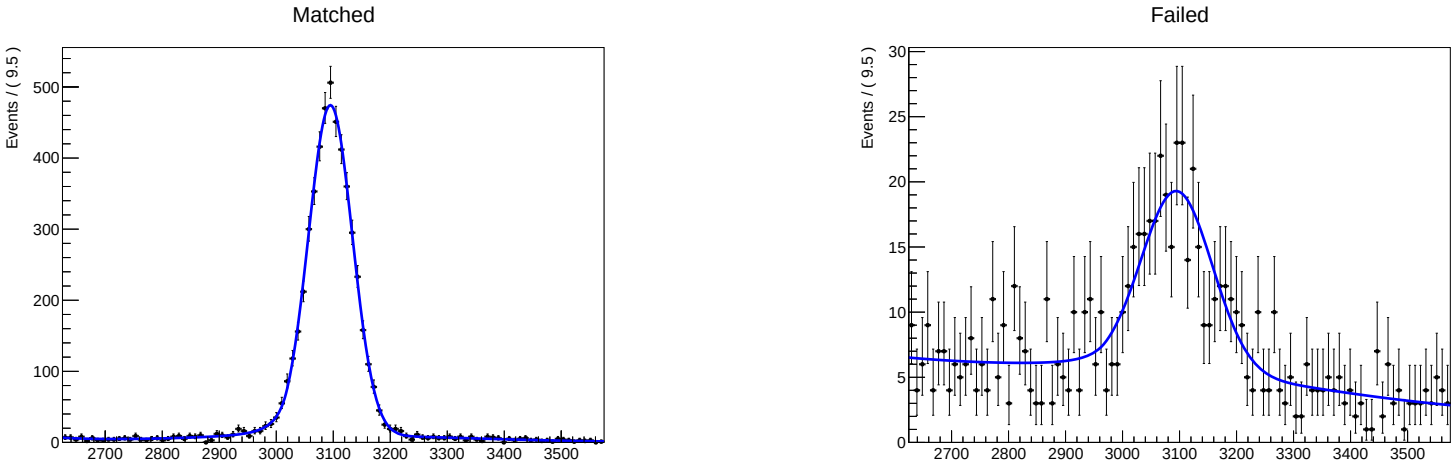
$$\begin{aligned} N_{\text{matched}} &= N_{\text{tot}} \times \epsilon_{\text{track}} \\ N_{\text{failed}} &= N_{\text{tot}} \times (1 - \epsilon_{\text{track}}) \end{aligned}$$

- An analogous set of parameters exist for the background yields
- My suspicion: $\epsilon_{\text{track}} \in [0, 1]$, but ϵ_{track} is close to 1
  - Maximum-likelihood fit might be biased
  - Particularly important then $N_{\text{failed}} \approx 0$
- Generate toys to assess this effect
- For now, only check MuonUT method as the yields are lower

Fit biases

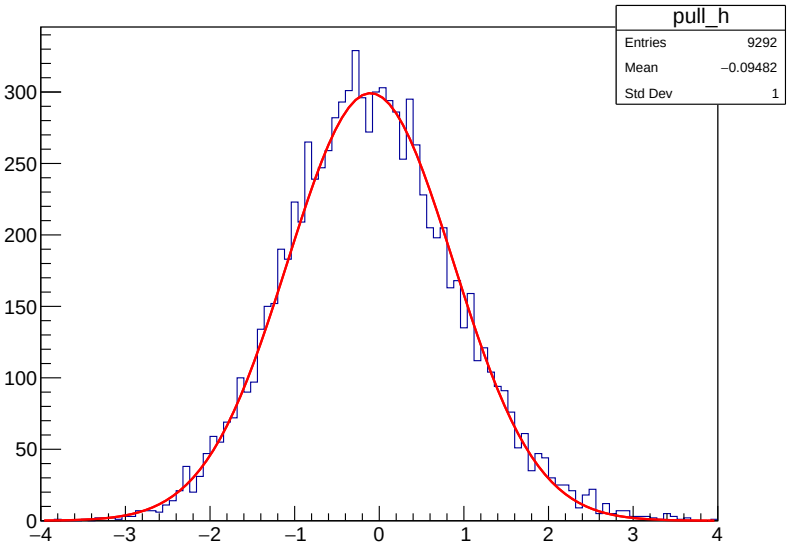


MuonUT (p, η) bin $(0, 0)$ MC fit

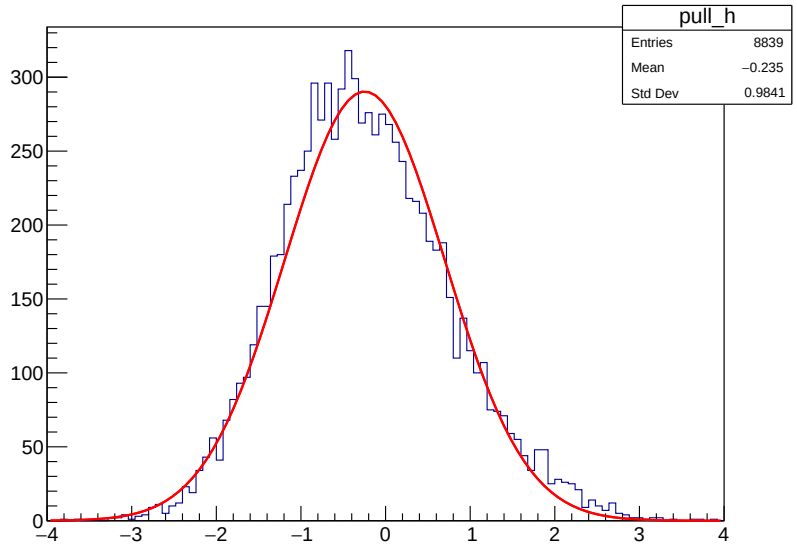


MuonUT (p, η) bin $(0, 0)$ toy fit

Well-behaved fits



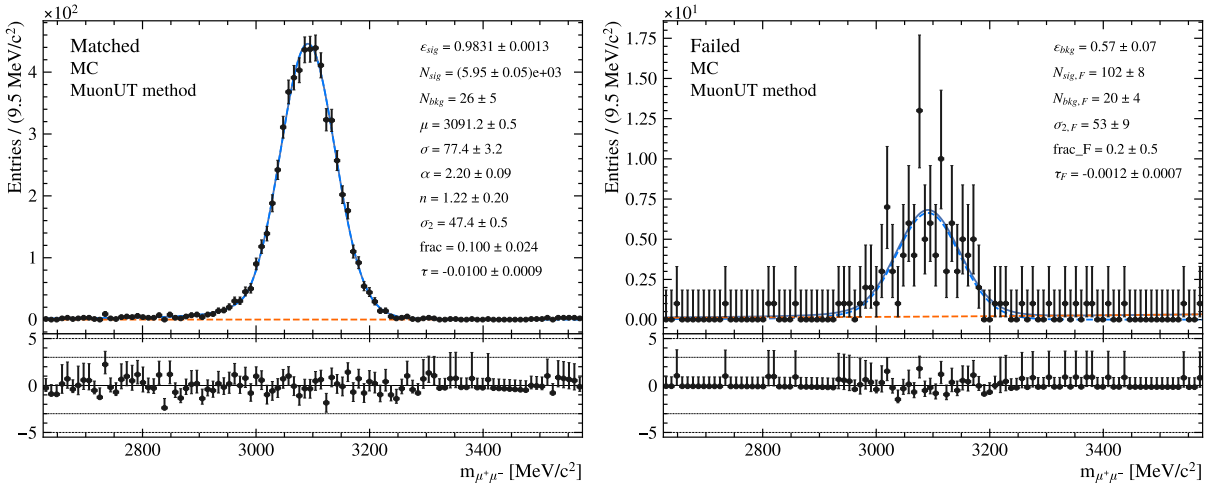
(p, η) bin $(3, 0)$



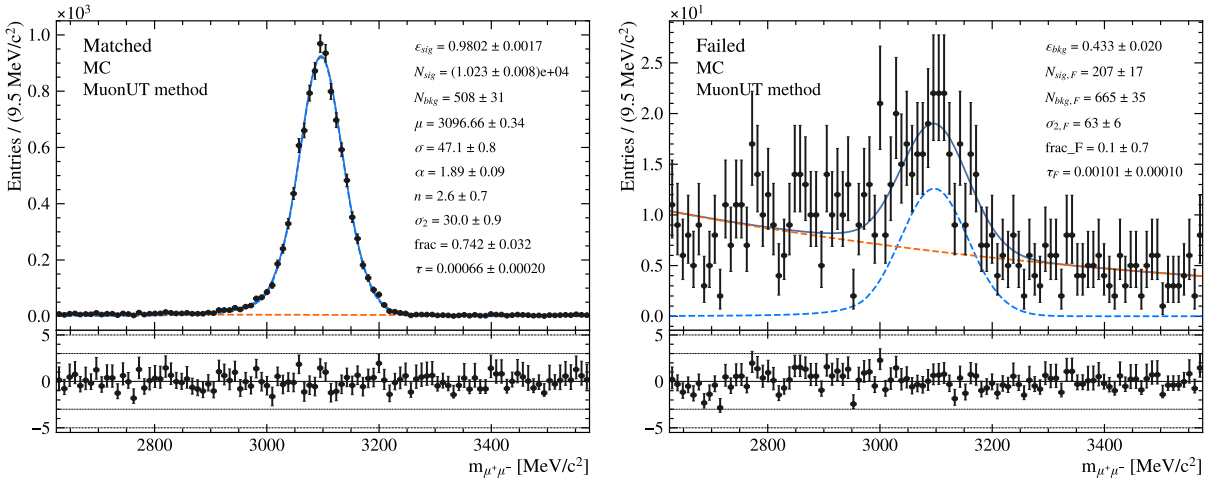
(p, η) bin $(2, 1)$

- In these bins we see reasonable behaviour (pulls are Gaussian shaped)
- Small negative bias, about 20-25% of the statistical uncertainty

Well-behaved fits

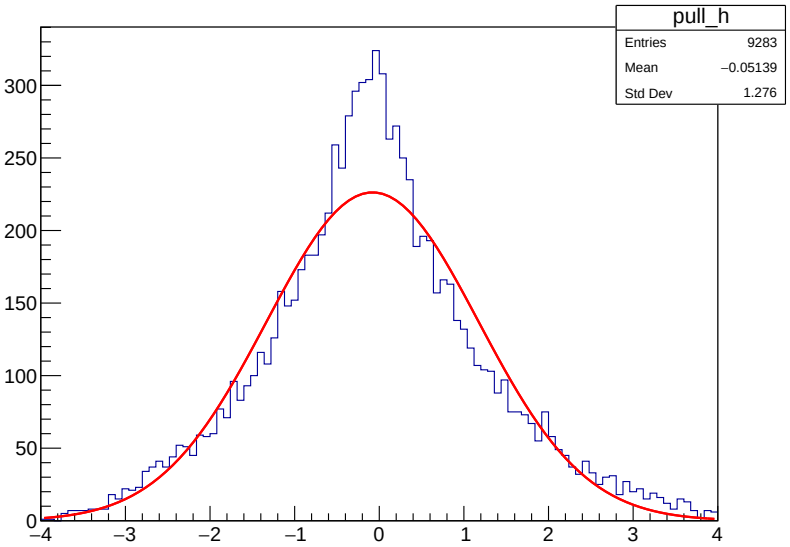


MuonUT (p, η) bin $(3, 0)$

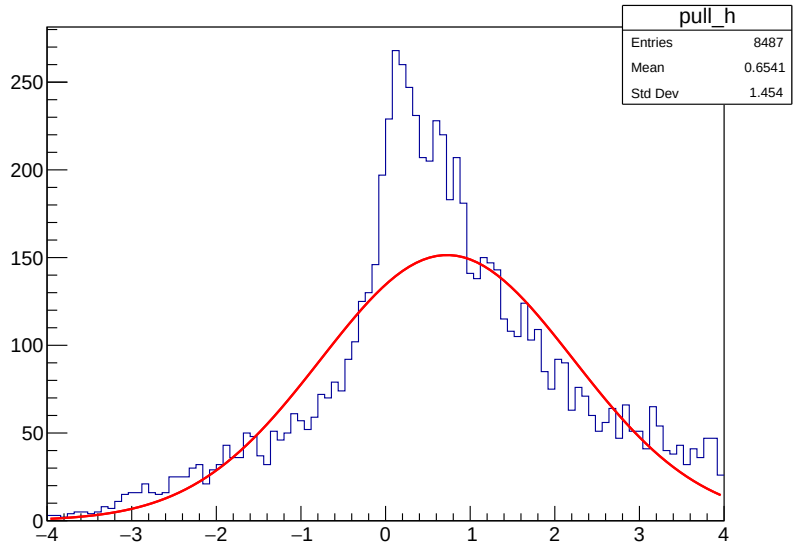


MuonUT (p, η) bin $(2, 1)$

Biased fits



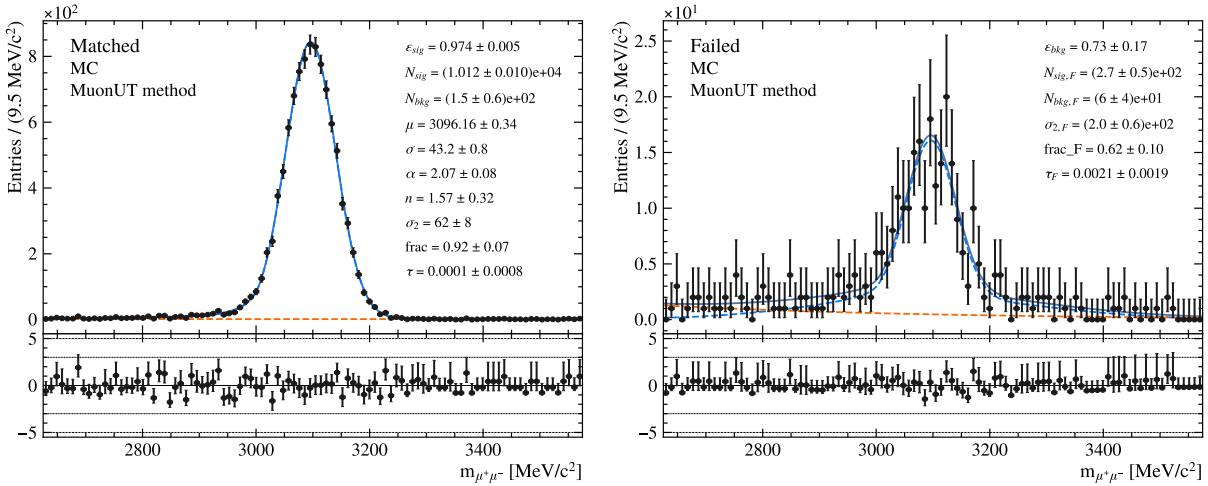
(p, η) bin $(2, 0)$



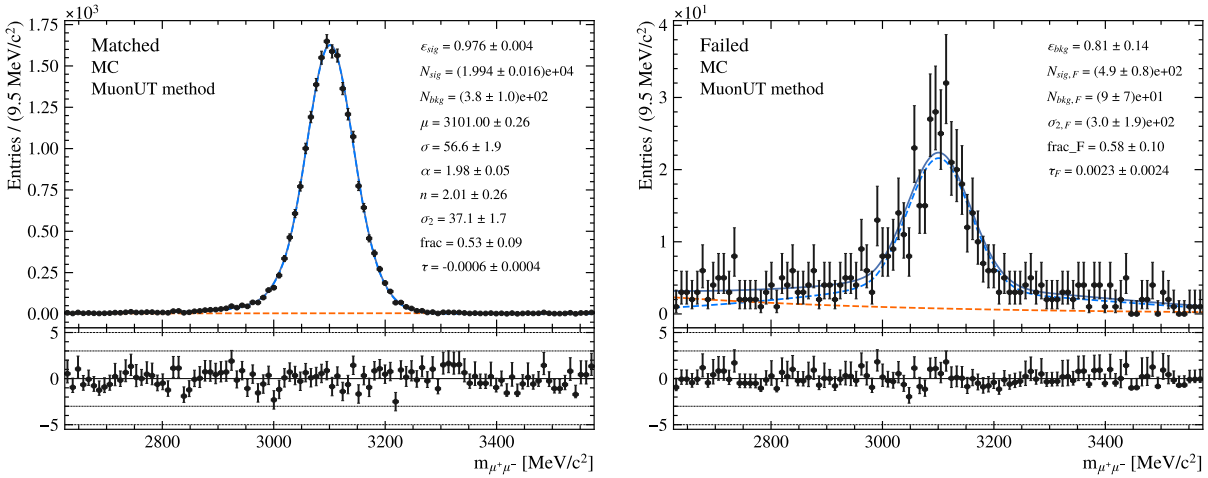
(p, η) bin $(4, 1)$

- Non-ideal statistical behaviour (pulls not Gaussian)
- Large tails and biases, actual uncertainties may be a lot larger

Biased fits



MuonUT (p, η) bin $(2, 0)$



MuonUT (p, η) bin $(4, 1)$

Fit biases

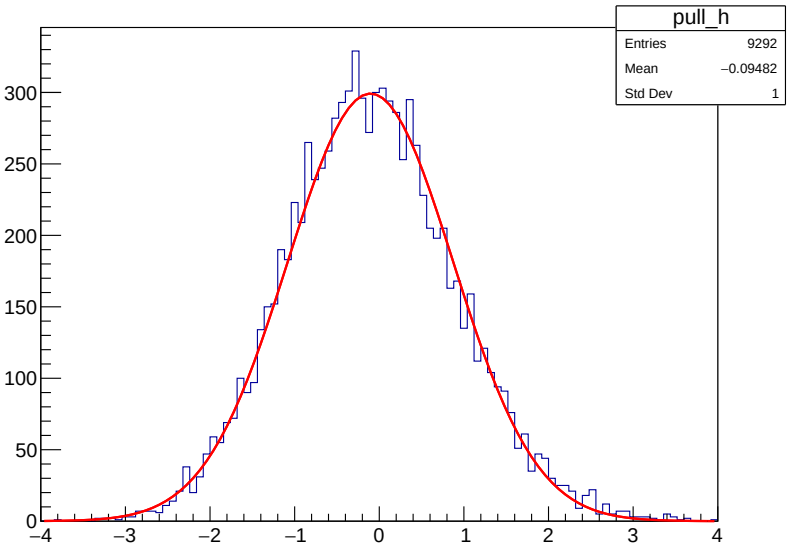
- My thoughts: Biases can occur when ϵ_{track} is close to 1...
- ... but this can happen for both signal and background
- In fact I see a 90% correlation between ϵ_{track} in signal and background in some toys!

p bin	η bin	Difference (10^{-2})	Gaussian?
0	0	0.4 ± 1.1	Small tail
1	0	-0.5 ± 0.2	Yes
2	0	-0.8 ± 0.5	No, large tails
3	0	-1.5 ± 0.1	Yes
4	0	-1.0 ± 0.2	Yes
0	1	3.2 ± 1.9	No, small tail
1	1	1.0 ± 0.5	No, small tail
2	1	-0.9 ± 0.2	Yes
3	1	0.0 ± 0.6	No, large tails
4	1	-0.7 ± 0.4	No, large tails

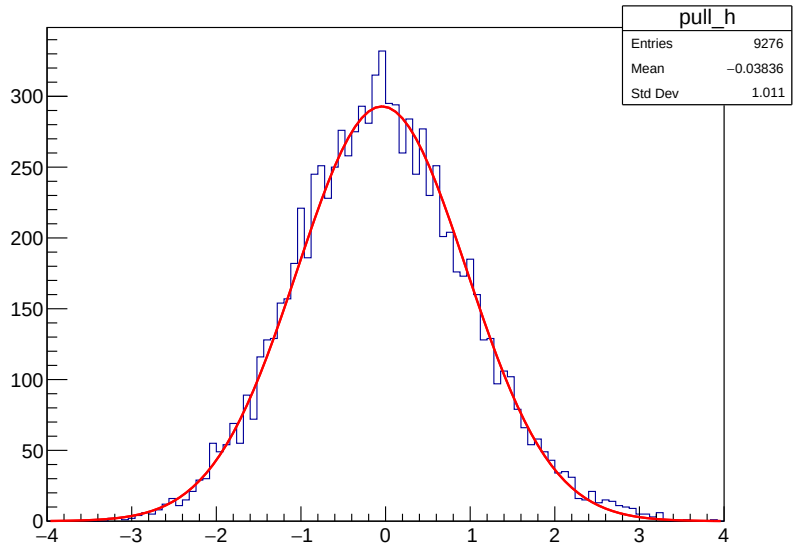
Alternative background parameterisation

- In fits with highly correlated variables, alternative parameterisations usually help
- First attempt:
  - For signal, float N_{tot} and $\epsilon_{\mathrm{track}}$
  - For background, float N_{matched} and N_{failed} separately
- Run same toy studies and check pull distributions

Alternative background parameterisation



Old parameterisation

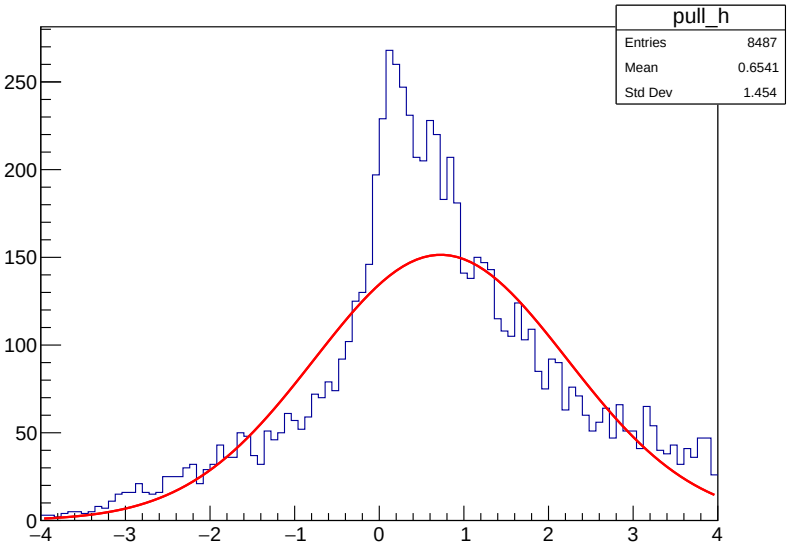


New parameterisation

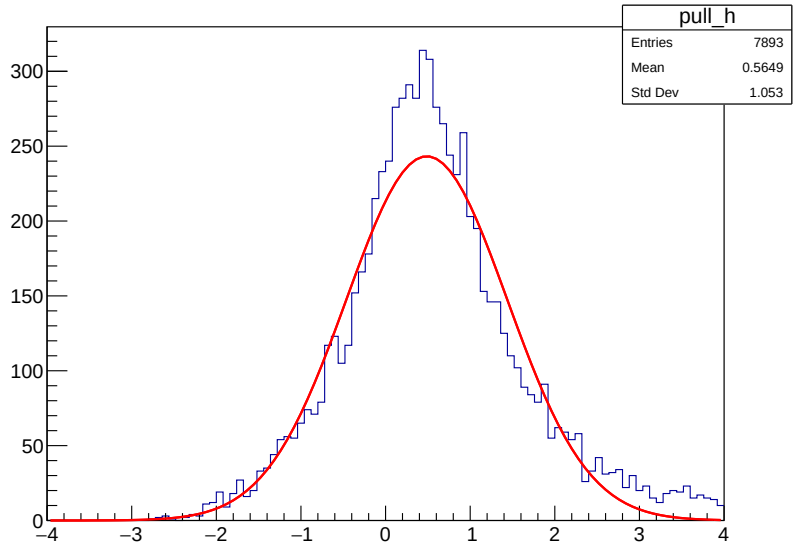
(p, η) bin $(3, 0)$

- Try floating the matched and failed yields directly
- Well behaved fits remain well behaved

Alternative background parameterisation



Old parameterisation



New parameterisation

(p, η) bin $(4, 1)$

- Try floating the matched and failed yields directly
- Statistical behaviour seems to improve in some fits!

Summary and next steps

- I checked the impact of fit biases in the MuonUT method
  1. In some bins, fit biases could be significant
  2. For a meaningful comparison, the MuonUT methods needs to:
     - Have small fit biases
     - Converge with correct uncertainties without need to rerun fit
  3. Only MC studied so far, but I suspect similar effects are present in data
- Next steps: Do the same checks in data

Thanks for listening!