Flavour Anomalies:

Angular Analysis of $B^0 \to K^{*0}(892)\ \mu^+\mu^-$ Decay Channel

Bi-weekly Meeting

Student: Thiago de Andrade Rangel Monteiro

17/06/2025

Outline

BDT Classification:

- Code adjustments for the classification BDT, now featuring GPU implementation. This reduces the hyperparameter search time from approximately 6 hours to just 40 minutes.
- BDT classification results when altering the track pT, with tests at 0.8, 1.0, and 1.2 GeV.

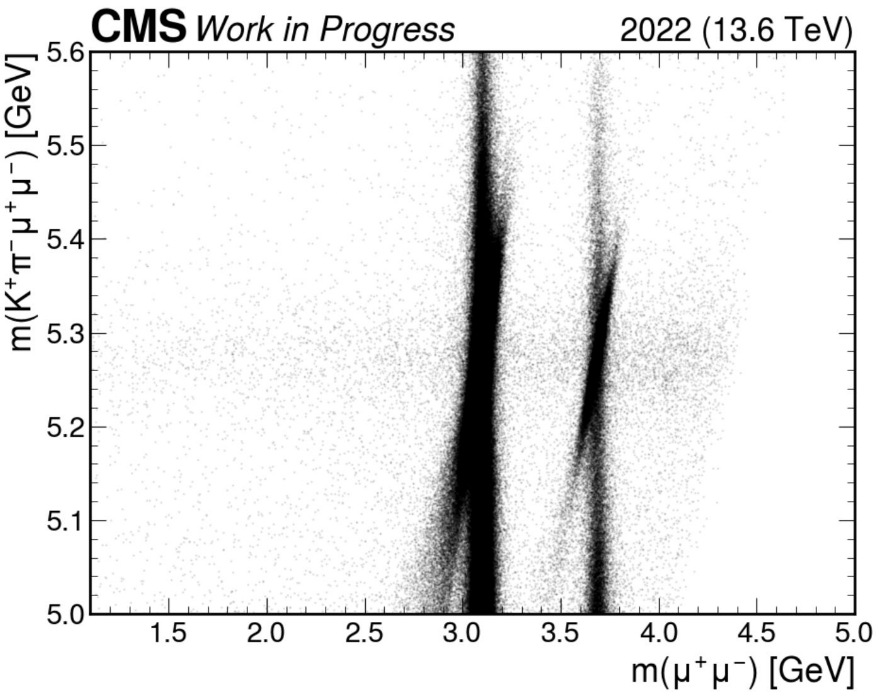
Background Studies:

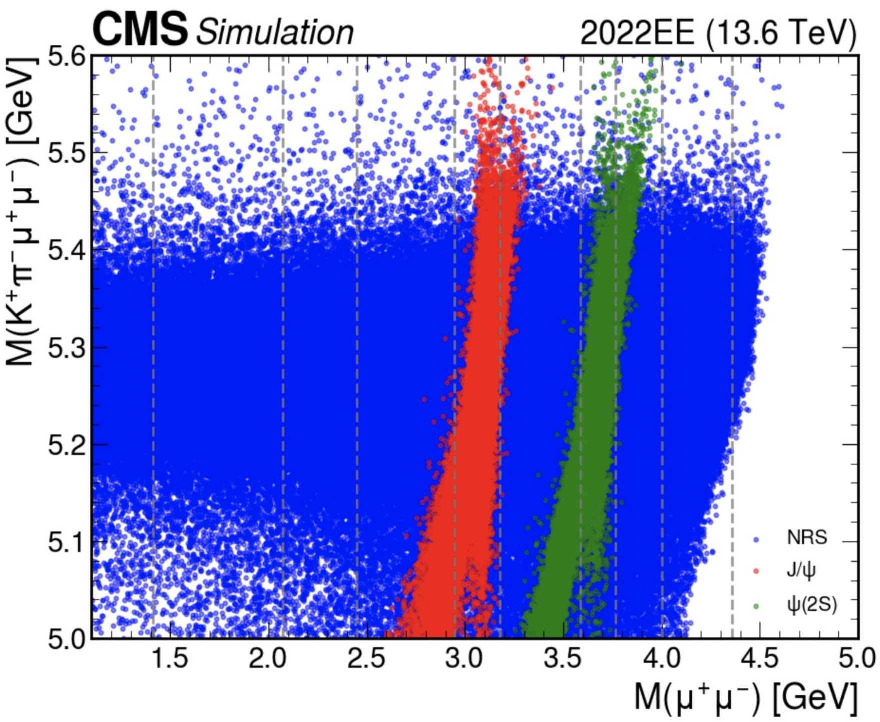
- Analysis of the background study for the $B^+ \to J/\psi K^+$ channel. (On going)
  a. Samples produced with and without filter.
- Analysis of event leakage into bins 3, 5, and 7 due to Final State Radiation (FSR).

Reweighting Studies:

- Use of the Class Flipping technique (First Approach).
- Use of the GB Reweighting technique (Second Approach).
- Use of the Run II technique (Third Approach).
- Use of the techniques suggested in the paper: *Machine Learning on data with sPlot background subtraction* (Possible Fourth Approach).

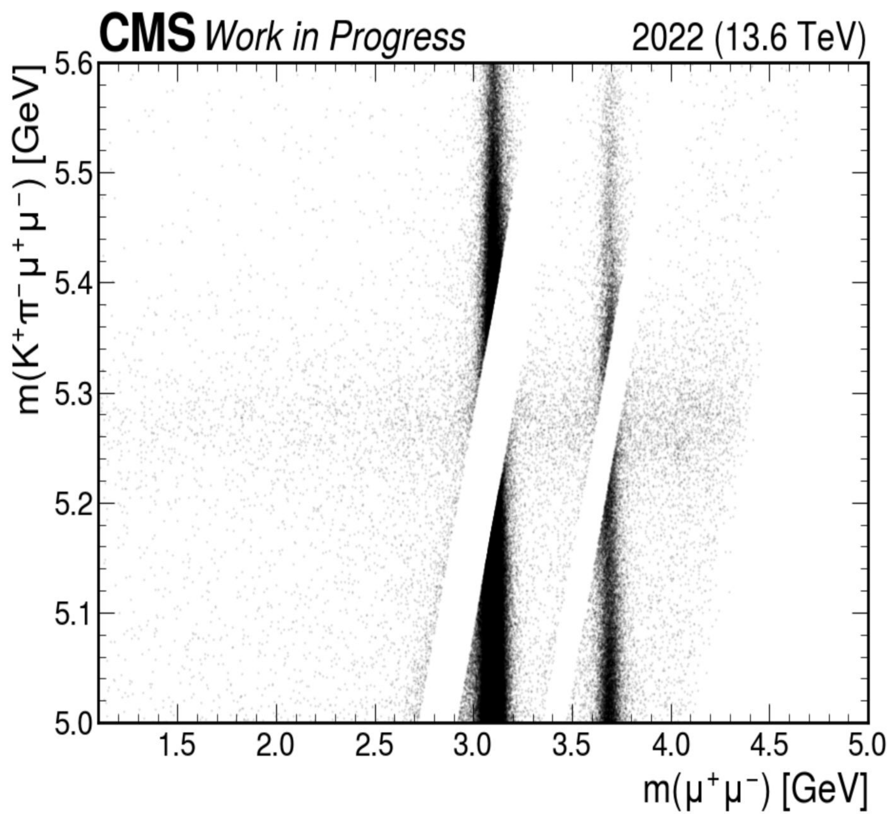
Background Studies





- ❑ Both Datasets have passed the preselection criteria and the BDT threshold
- ❑ We can see quite clear the contamination of the resonant bins in the nonresonant bins like bin 3, 5 and 7 (bin before the j/psi, between the j/psi and psi(2s) and after the psi(2s) region)

Background Studies



An totally arbitrary cut has been imposed here and the optimization procedure will be done by next meeting (Following the Run II)

$$|m(\mu^+\mu^-) - m_{\psi}^{PDG}| > 3\sigma_{\mu\mu}$$

$$|(m(K\pi\mu\mu) - m_{B0}^{PDG}) - (m(\mu\mu) - m_{\psi}^{PDG})| < \Delta m.$$

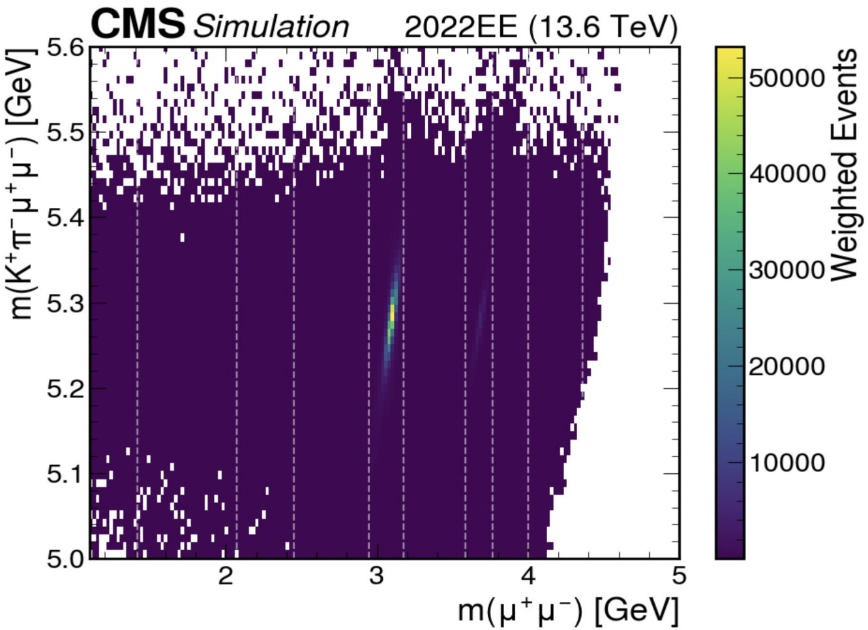
The optimisation criteria is the reduction of the feed-through from the control channels to a negligible level, below X% of the total background on data within the mass window defined by $|m(K^\pm\pi^\mp\mu^+\mu^-) - m^{PDG}_{B0}| < 0.28$ GeV, where X is defined as:

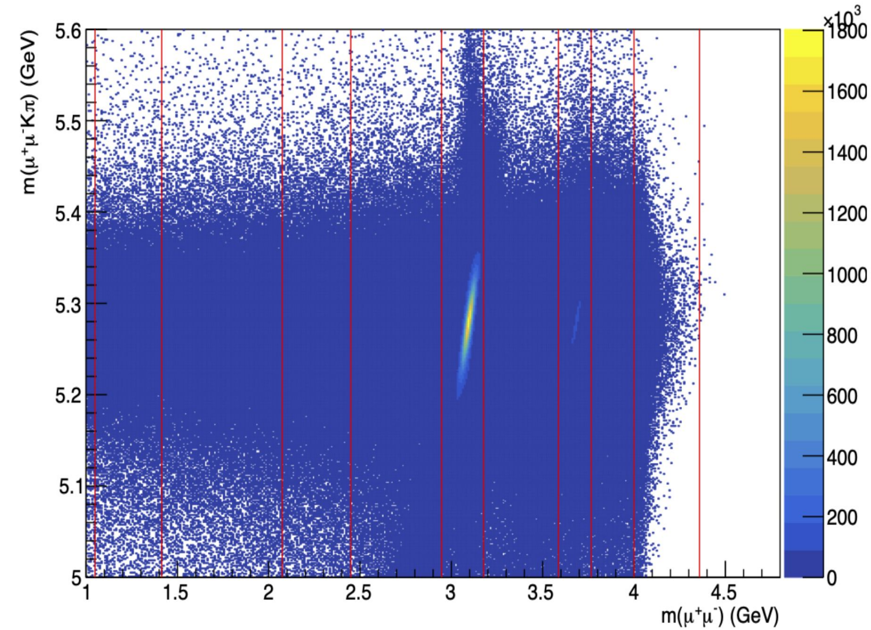
- 5% for each of the two resonant channels for bin 3 and 5,
- 1% for the sum of the two channels for bin 7.

B_{peak}: number of feed-through events from the control channel simulations;

B_{tot}: number of events in the data minus the number of events from signal (non-resonant) simulation.

Background Studies





Using the totally arbitrary cuts the difference was brutal in the fits (see slides from the last presentation) and I expect to improve those fits with the optimization criteria of the Run II.

Reweighting Studies - (First Approach)

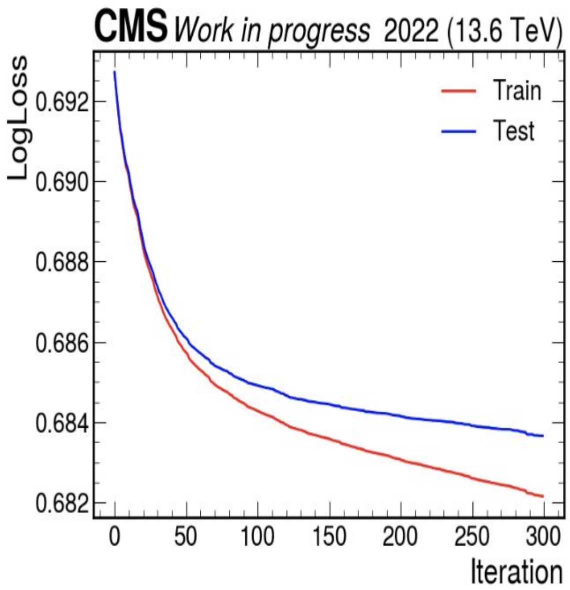
The method resolves a technical limitation in XGBoost by converting an invalid instruction into a processable rule. This allows the algorithm to learn and isolate the data of interest without crashing, even if the approach lacks physical meaning.

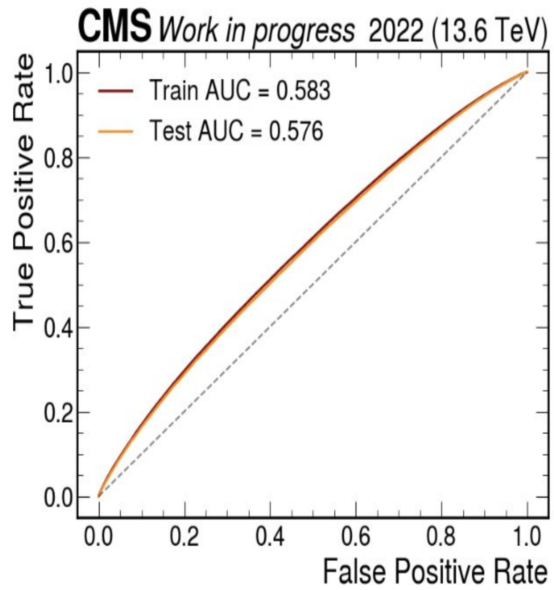
Key Points:

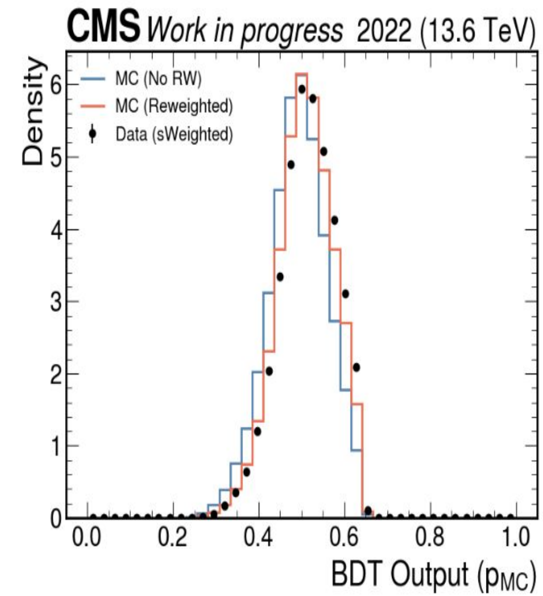
- **Rule Adaptation:** It transforms a negative weight in the Signal class into a positive weight in the Background class.
- **Objective Maintenance:** The algorithm remains focused on reducing the probability of an event being classified as Signal.
- **Computational Stability:** It enables the model to train correctly without causing the collapse of internal calculations (avoiding numerical instability).
- **Theoretical Limitation:** The solution is a purely mathematical/computational trick and holds no physical validity.

```
model = xgb.XGBClassifier(
    n_estimators=300, max_depth=4, learning_rate=0.05,
    subsample=0.8, colsample_bytree=0.8, scale_pos_weight=scale_pos_weight,
    eval_metric="logloss", random_state=random_state, n_jobs=-1
)
```

Reweighting Studies - (First Approach)





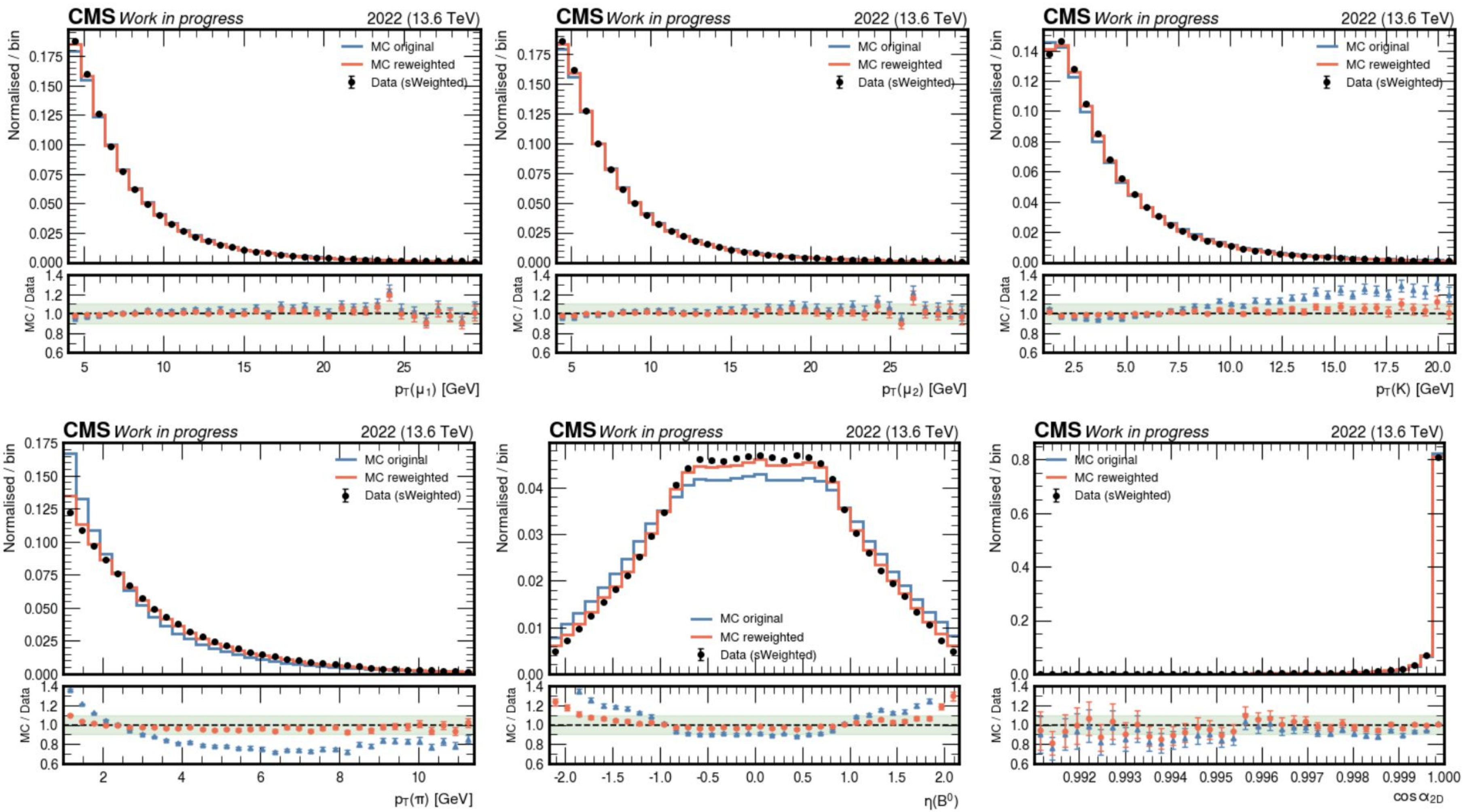


- **LogLoss Plot:** From the LogLoss plot, we observe that the curves take slightly longer to diverge compared to the AN (Analysis Note). However, at a higher number of iterations, the differences begin to emerge.

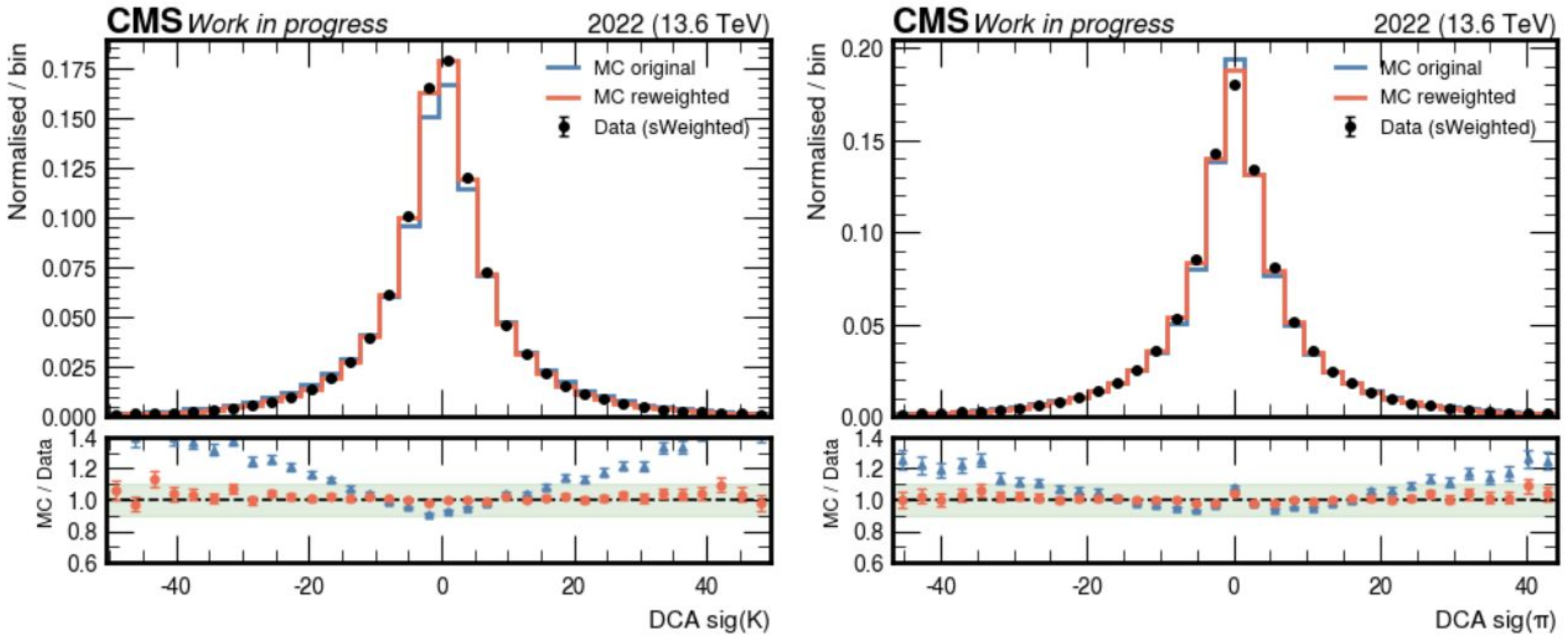
- **ROC Curve:** Both ROC curve results demonstrate good agreement between the training and testing samples. The same strategy from the AN was used, splitting the data into even and odd samples based on the event number. Consequently, the BDT was trained on one fold and applied to the other.

- **BDT Output:** The BDT Output exhibits a discrepancy on the right side of the plot, but it maintains the overall shape of the data distribution.

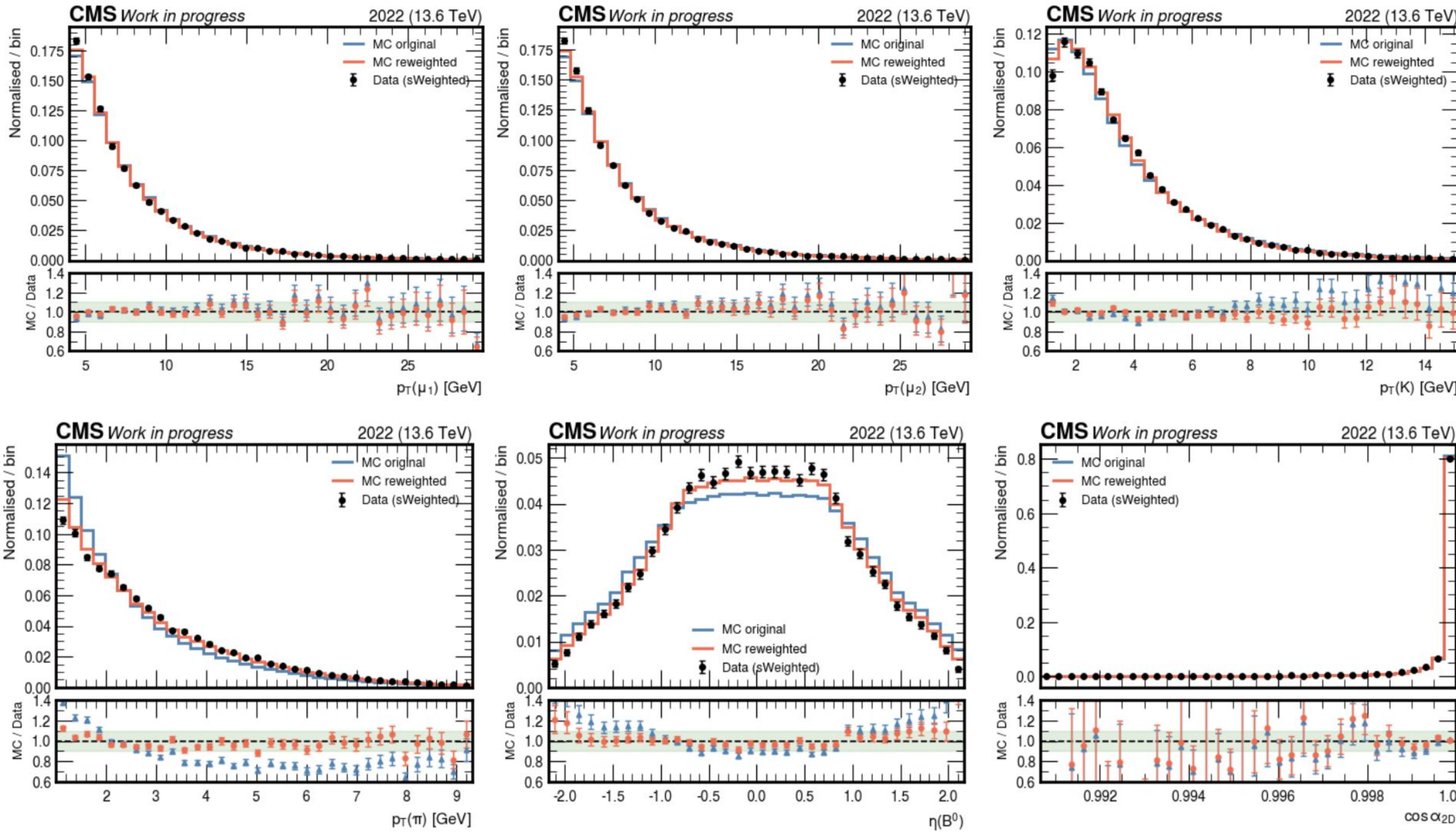
Closure Test J/Psi - (First Approach)



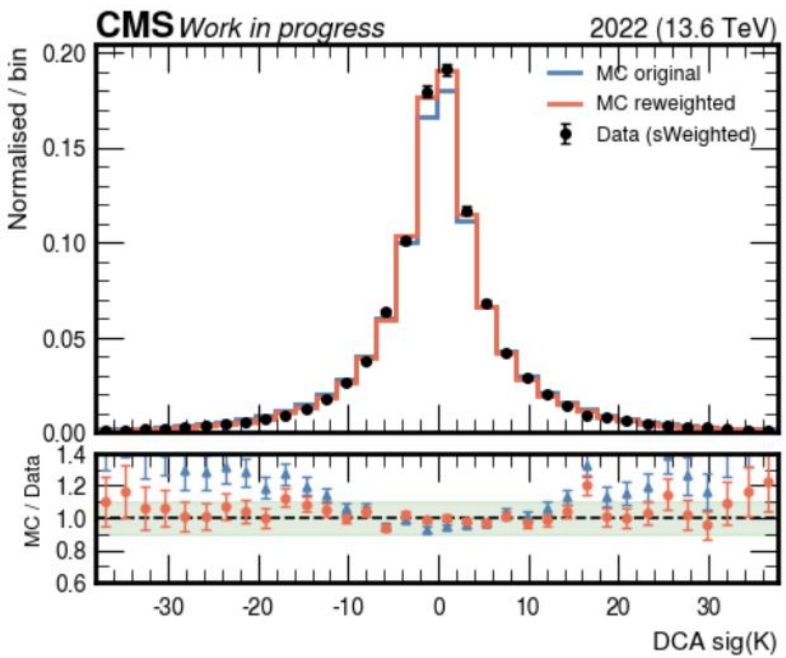
Closure Test J/Psi - (First Approach)

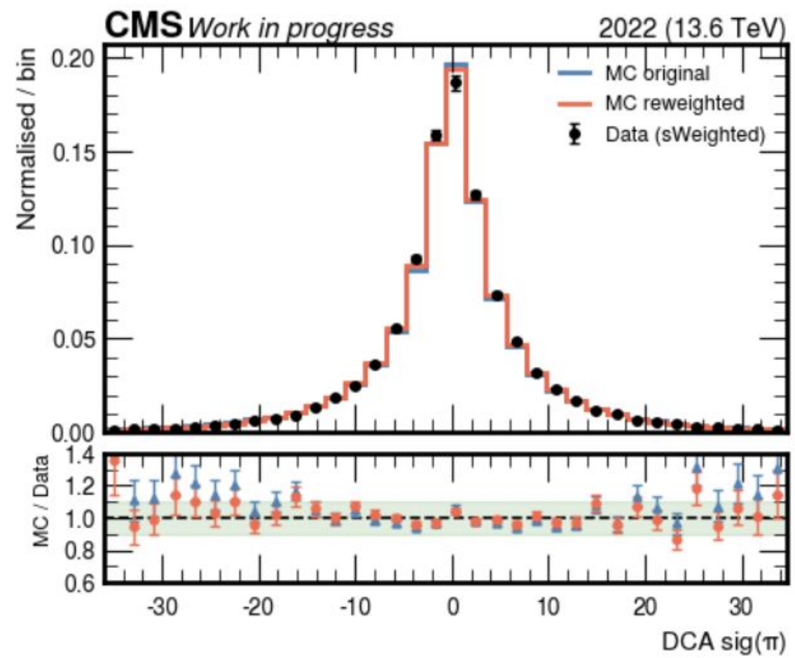


Closure Test Psi(2s) - (First Approach)



Closure Test Psi(2s) - (First Approach)





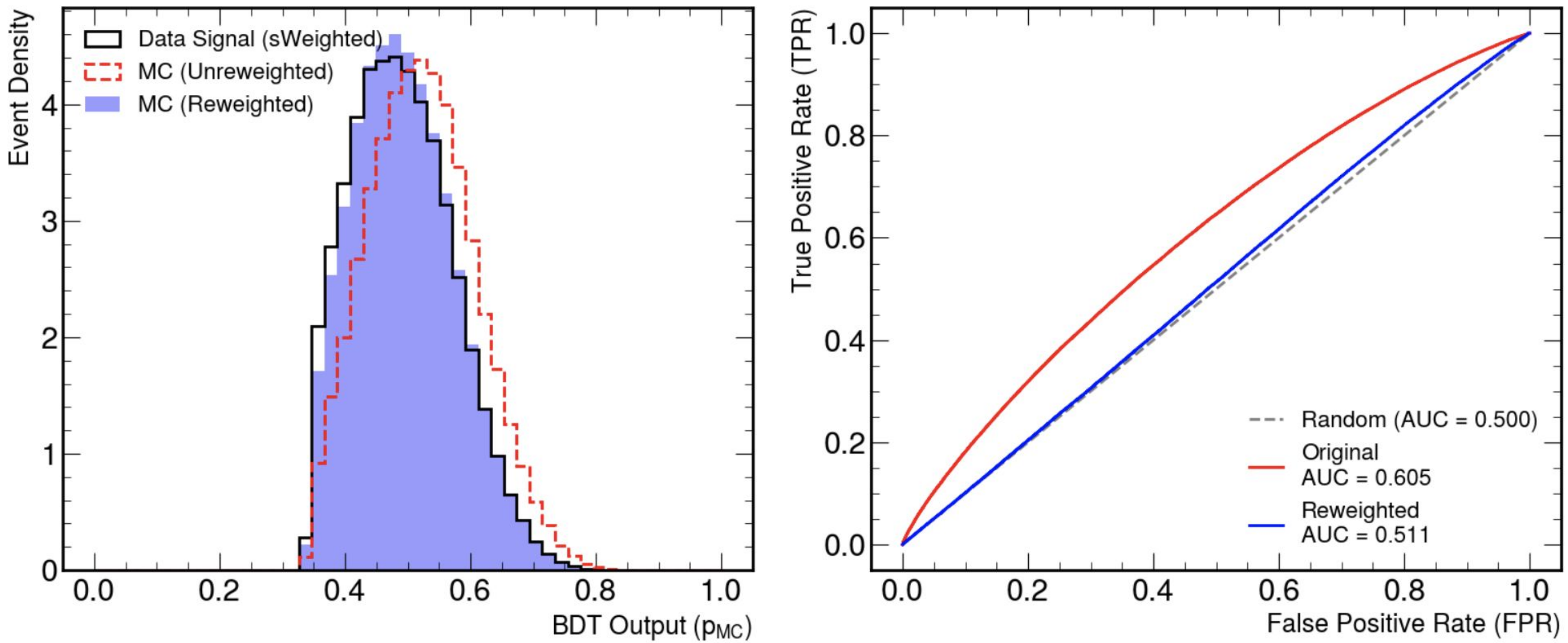
Reweighting Studies - (Second Approach)

The Gradient Boosted Reweighting technique (implemented in the `hep_ml` library), based on the paper *Reweighting with Boosted Decision Trees* (arXiv), is indeed highly promising. It solves classical problems associated with multi-dimensional data reweighting and successfully handles negative weights derived from sPlot. However, it does present some limitations:

- **Overfitting and the Absence of Global Metrics:** There is no single, definitive metric generated by the model to warn against multi-dimensional overfitting. Consequently, the primary analytical tool for validation ends up being the Kolmogorov-Smirnov (KS) test.
- **Dependency on a Secondary Classifier (Indirect Validation):** Due to the lack of a native validation metric, an extra and computationally expensive step is required in the pipeline. To evaluate whether the reweighting was successful and did not simply "memorize" the training data (overfit), it is necessary to train an independent classifier (a new decision tree or neural network). The objective of this secondary model is to attempt to distinguish the reweighted data from the target data in the test set.

```
reweighter = GBReweighter(
    n_estimators      = 300,
    learning_rate     = 0.05,
    max_depth         = 4,
    min_samples_leaf  = 200,
    gb_args           = {"subsample": 0.7},
)
```

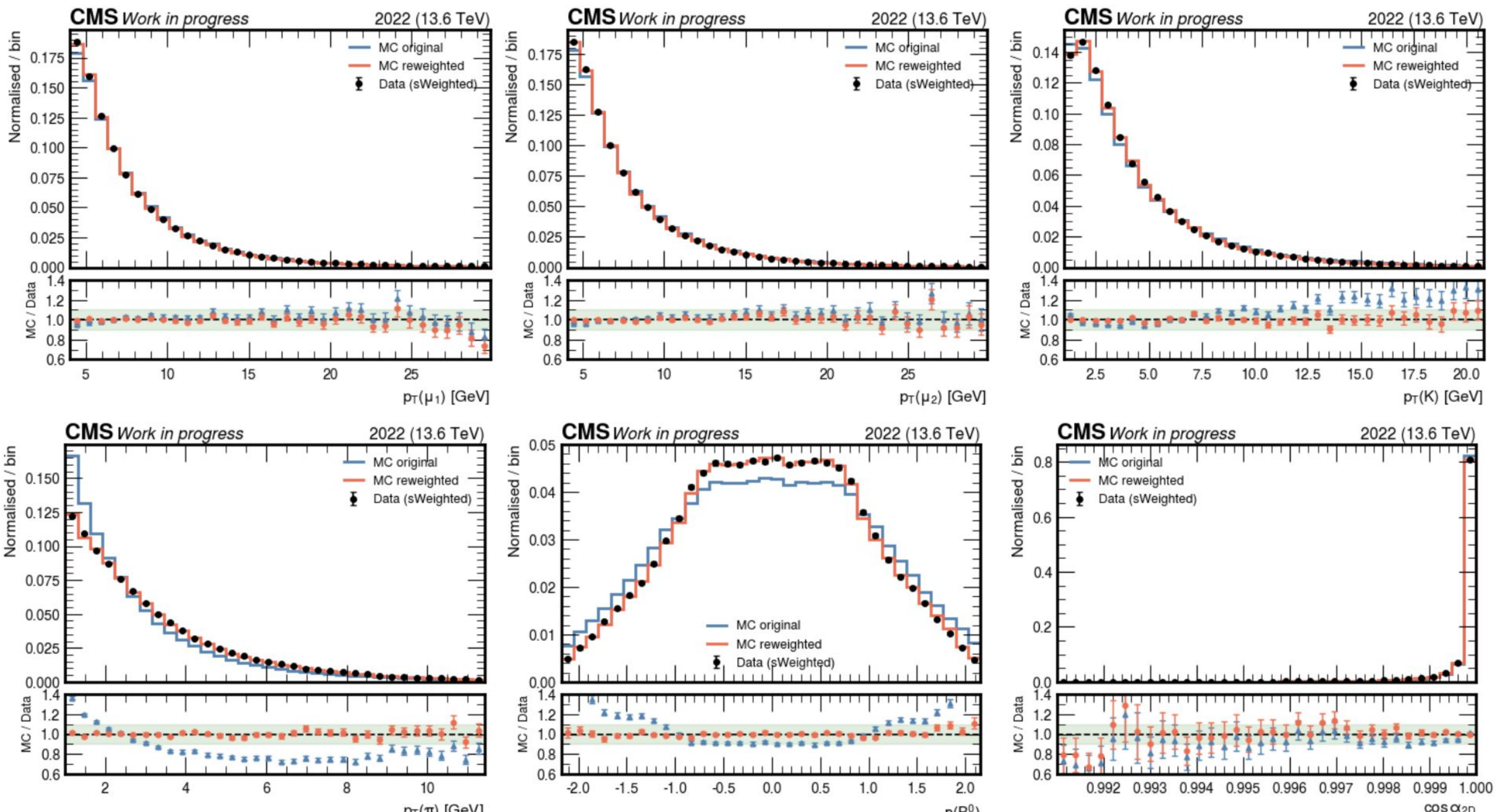
Reweighting Studies - (Second Approach)



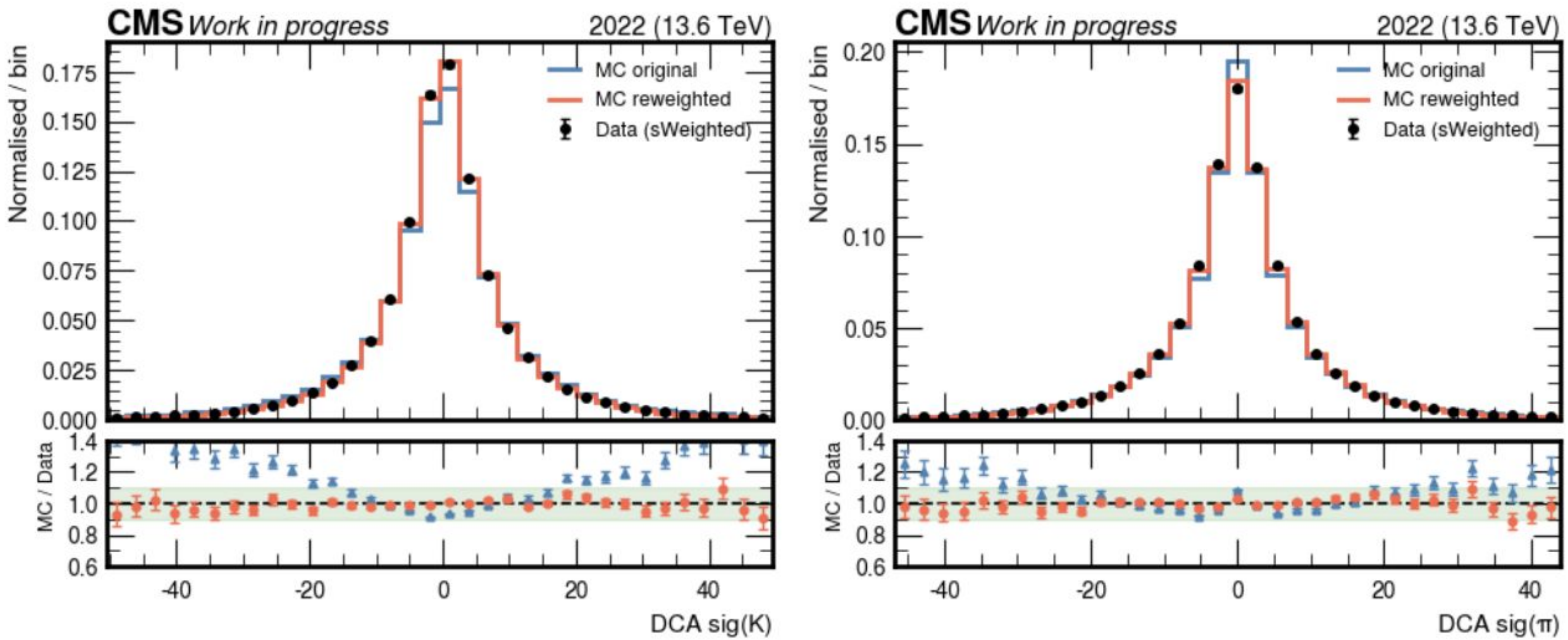
- Generating the BDT output plot is straightforward, as it simply requires going back one step and redefining the BDT output from the reweighting weight.

- However, regarding the ROC curve, there might be an issue because I did not use `scikit-learn`'s `roc_curve` function. Since some events have negative statistical weights (sWeights), this violates the monotonicity requirement of `scikit-learn` when calculating the Area Under the Curve (AUC). Consequently, it became necessary to compute the area using direct integration via the trapezoidal rule.

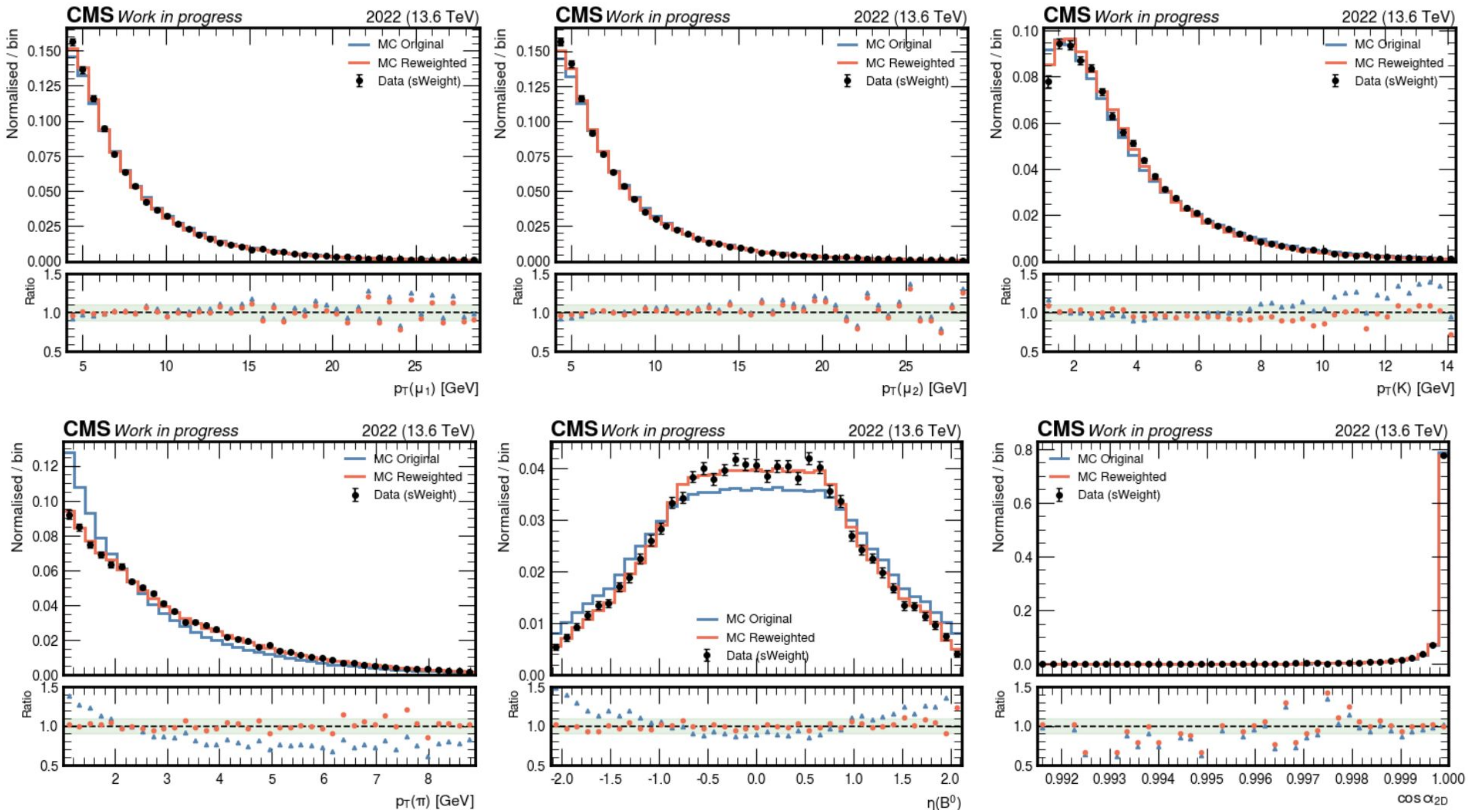
Closure Test J/Psi - (Second Approach)



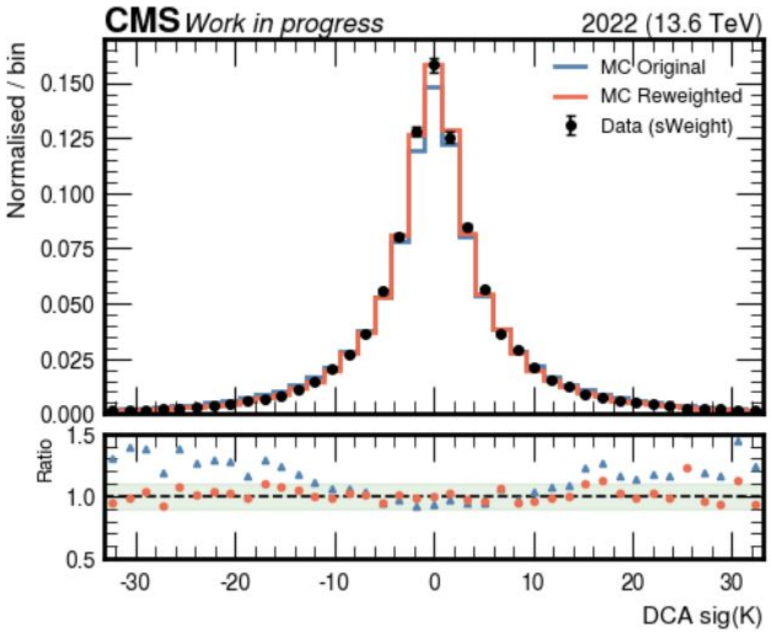
Closure Test J/Psi - (Second Approach)

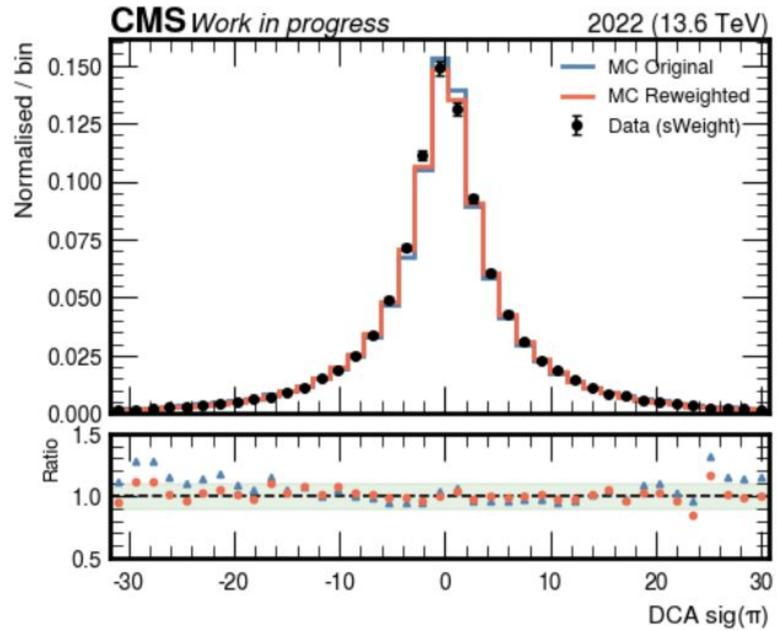


Closure Test Psi(2s) - (Second Approach)

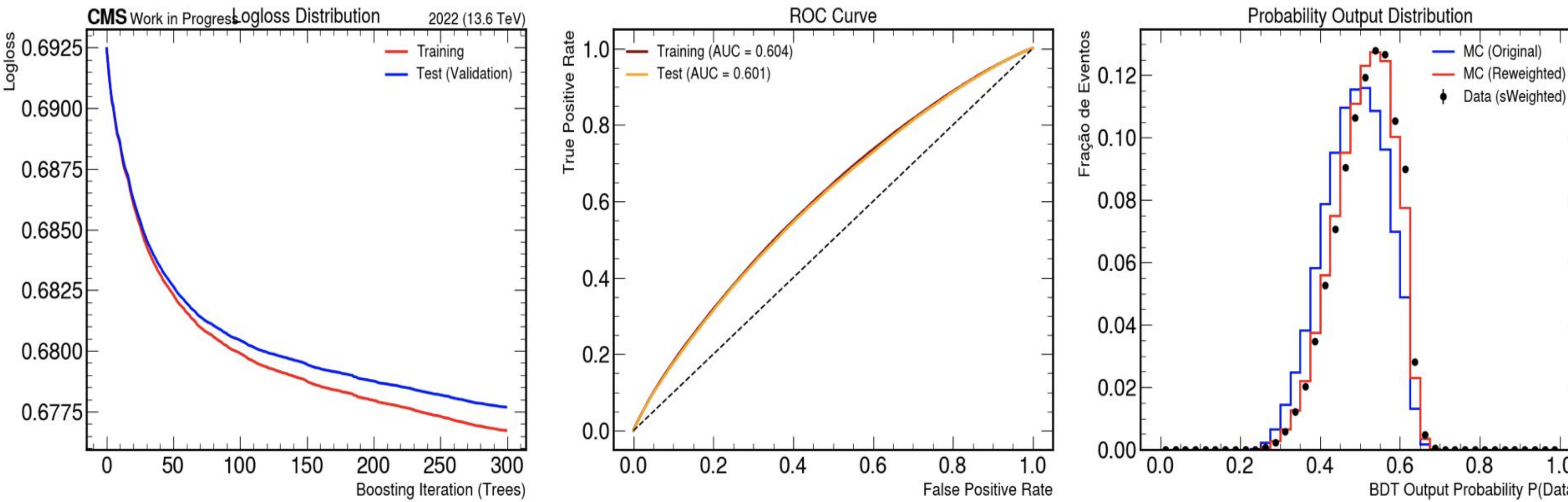


Closure Test Psi(2s) - (Second Approach)

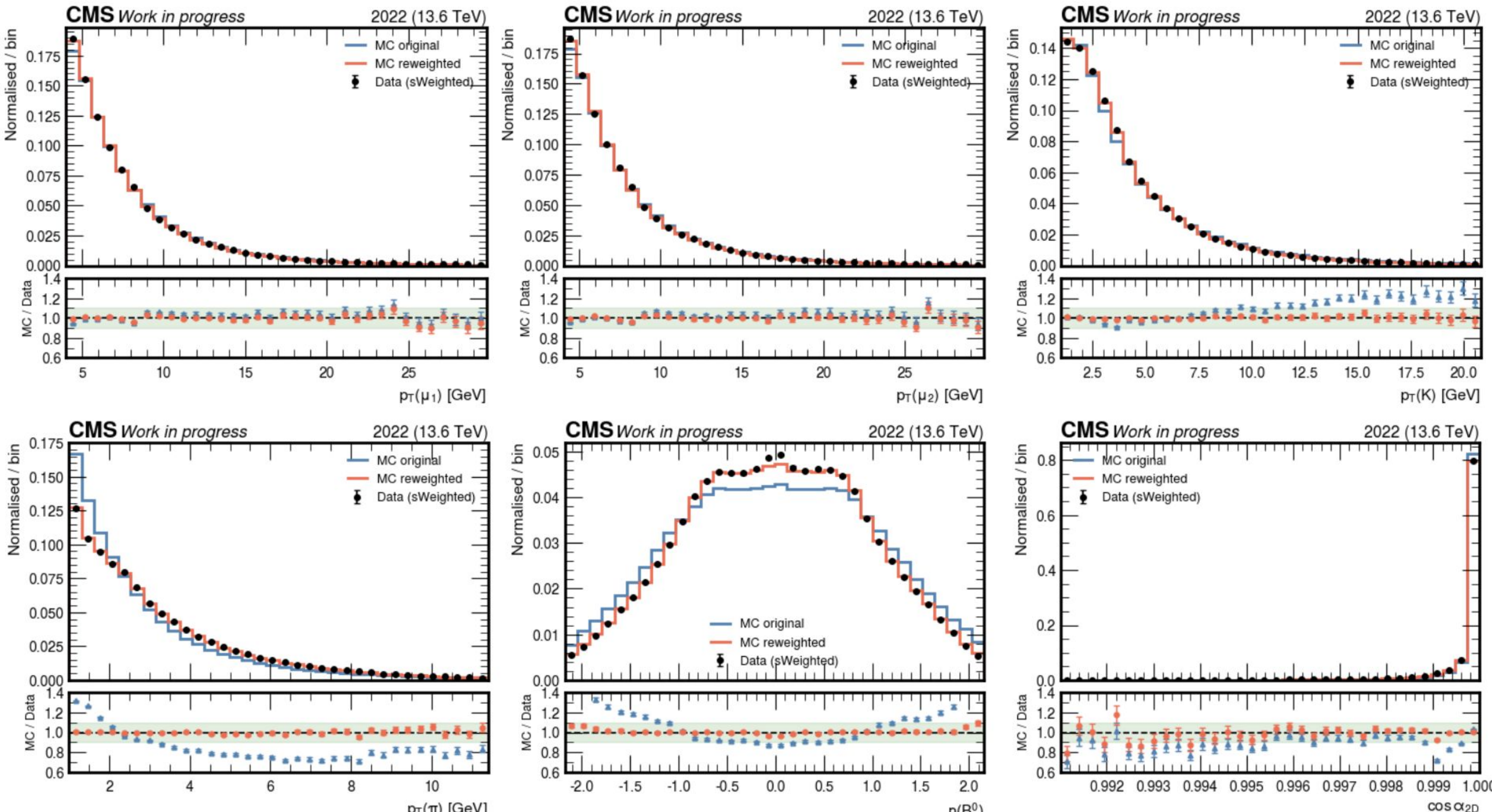




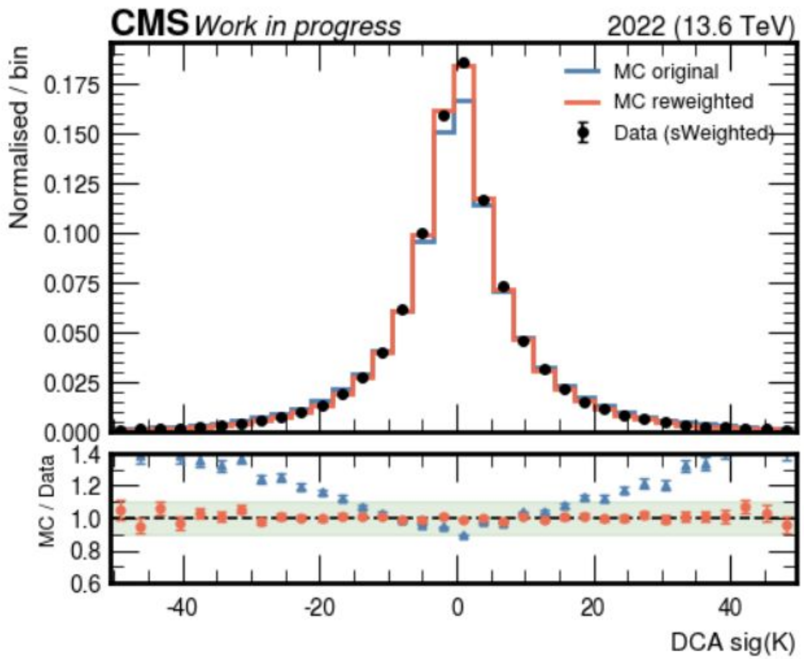
Reweighting Studies - (Third Approach)

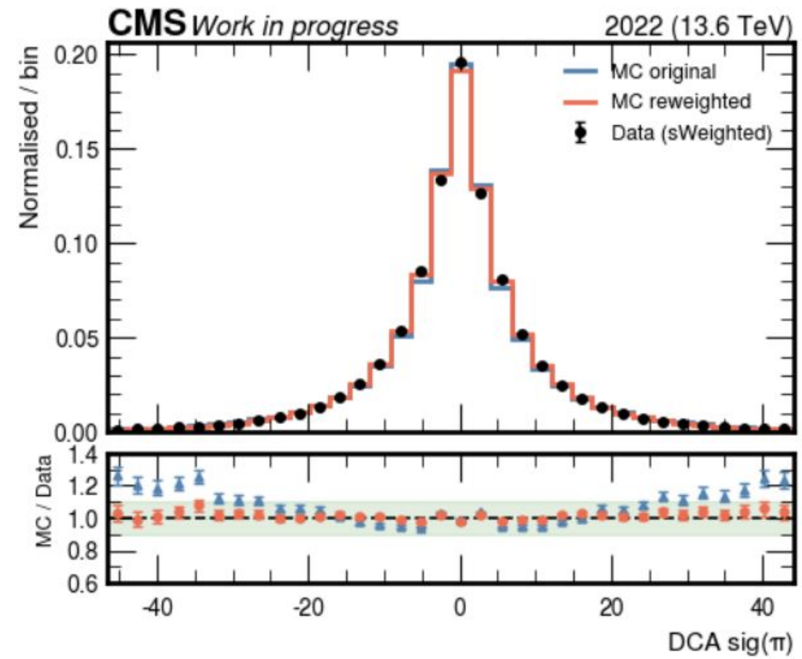


Closure Test J/Psi - (Third Approach)

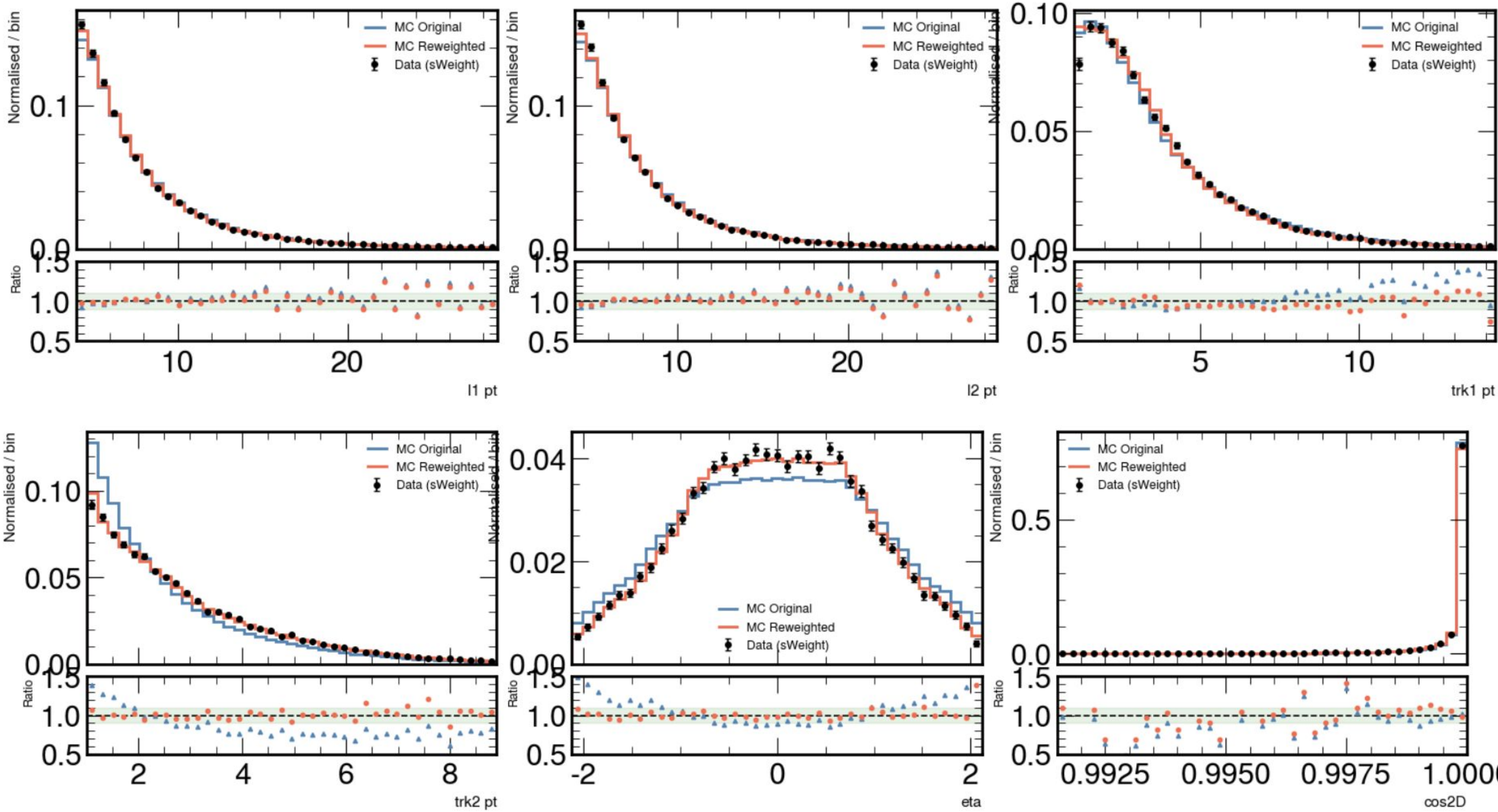


Closure Test J/Psi - (Third Approach)

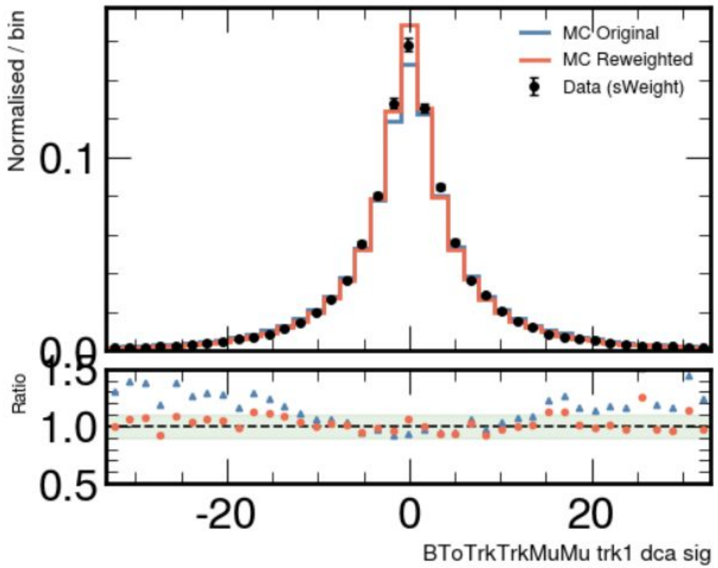


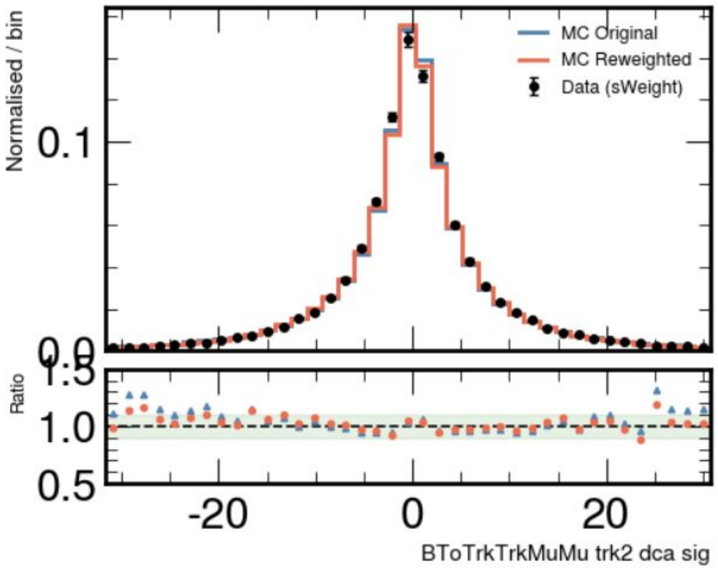


Closure Test Psi(2s) - (Third Approach)



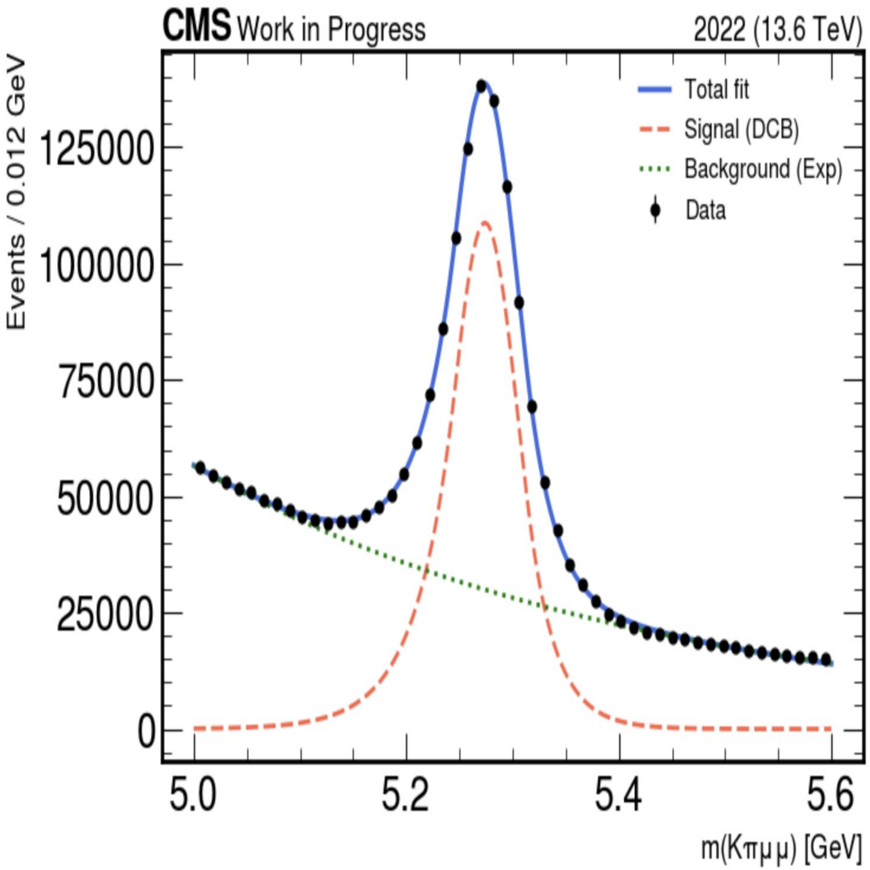
Closure Test Psi(2s) - (Third Approach)





Backup

Fitting Result J/Psi

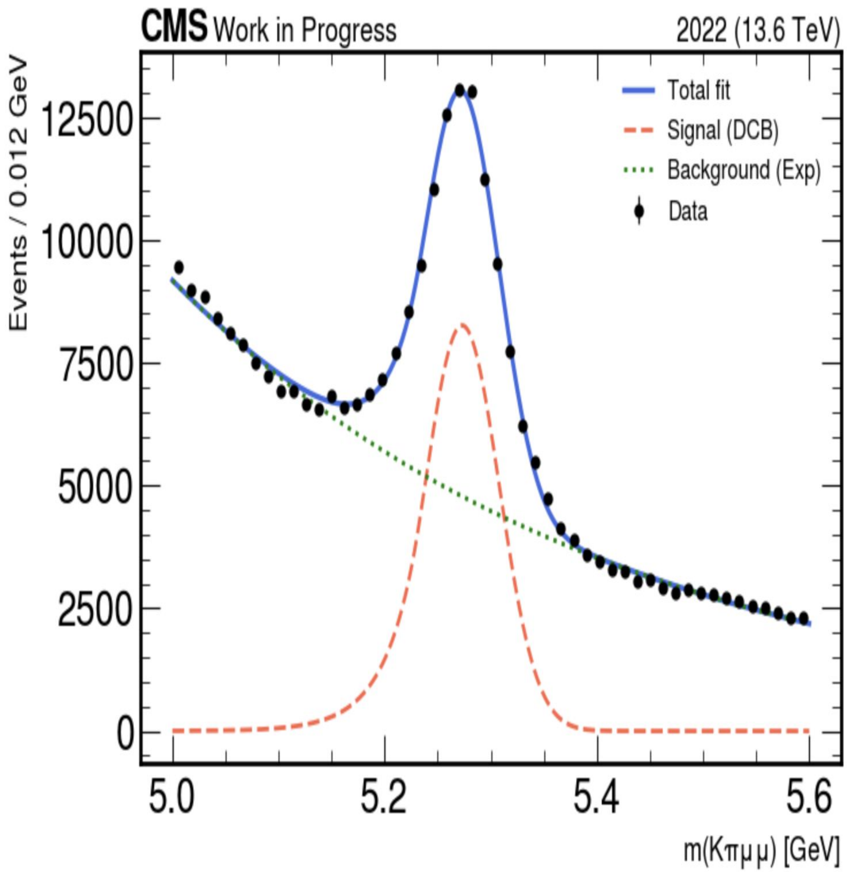


valid	converged	param at limit	edm	approx. fmin (full | opt.)
True	True	False	0.00022	-1687194.46 | -135358.2

```
Parameters
name        value  (rounded)    at limit
------  ------------------  ----------
n_sig               832852       False
n_bkg          1.52633e+06       False
mu                 5.27396       False
sigma            0.0316334       False
alphal            0.891848       False
nl                 99.9927       False
alphar             1.24429       False
nr                 147.389       False
tau               -2.31925       False
```

```
(((df['MuMu_mass'] > 2.9) & (df['MuMu_mass'] < 3.3)) &
(mask_kpi_window | mask_pik_window) &
(is_kpi_closer) &
(df['HLT_DoubleMu4_3_LowMass'] == 1) &
(df['BPH_1Muon_pt'] > 4.0) &
(df['BPH_2Muon_pt'] > 4.0) &
(abs(df['BPH_1Muon_eta']) < 2.4) &
(abs(df['BPH_2Muon_eta']) < 2.4) &
(df['BToTrkTrkMuMu_fit_trk1_pt'] > 1.0) &
(df['BToTrkTrkMuMu_fit_trk2_pt'] > 1.0) &
(df['BToTrkTrkMuMu_fit_cos2D'] > 0.99) &
((df["BToTrkTrkMuMu_fit_mass_Kpi"] > 5.0) & (df["BToTrkTrkMuMu_fit_mass_Kpi"] < 5.6))
```

Fitting Result Psi(2s)



valid	converged	param at limit	edm	approx. fmin (full | opt.)
True	True	False	0.0009	-195672.09 | 5738.595

Parameters

name	value (rounded)	at limit
n_sig	63707.5	False
n_bkg	243684	False
mu	5.27293	False
sigma	0.0349737	False
alphal	1.1508	False
nl	99.9952	False
alphar	4.93893	False
nr	22.2431	False
tau	-2.38904	False

```
mask = (
    (((df['MuMu_mass'] > 3.5) & (df['MuMu_mass'] < 3.8)) &
    (mask_kpi_window | mask_pik_window) &
    (is_kpi_closer) &
    (df['HLT_DoubleMu4_3_LowMass'] == 1) &
    (df['BPH_1Muon_pt'] > 4.0) &
    (df['BPH_2Muon_pt'] > 4.0) &
    (abs(df['BPH_1Muon_eta']) < 2.4) &
    (abs(df['BPH_2Muon_eta']) < 2.4) &
    (df['BToTrkTrkMuMu_fit_trk1_pt'] > 1.0) &
    (df['BToTrkTrkMuMu_fit_trk2_pt'] > 1.0) &
    (df['BToTrkTrkMuMu_fit_cos2D'] > 0.99) &
    ((df["BToTrkTrkMuMu_fit_mass_Kpi"] > 5.0) & (df["BToTrkTrkMuMu_fit_mass_Kpi"] < 5.6))
))
```