Searching for CP violation in ttH multilepton process with CMS at the LHC Run 3
-
Generalization of particle track fitting for the HL-LHC

PHD Day

Giorgio Mauceri[1][2]

Adriano Di Florio[2], Andrea Giammanco[3],
Jindrich Lidrych[3], Nicolas Chanon[1], Zak Lawrence[3]

22/04/26

1=IP2I Lyon 2=CC-IN2P3 Lyon 3=CP3 Louvain

Summary

0: LHC timeline

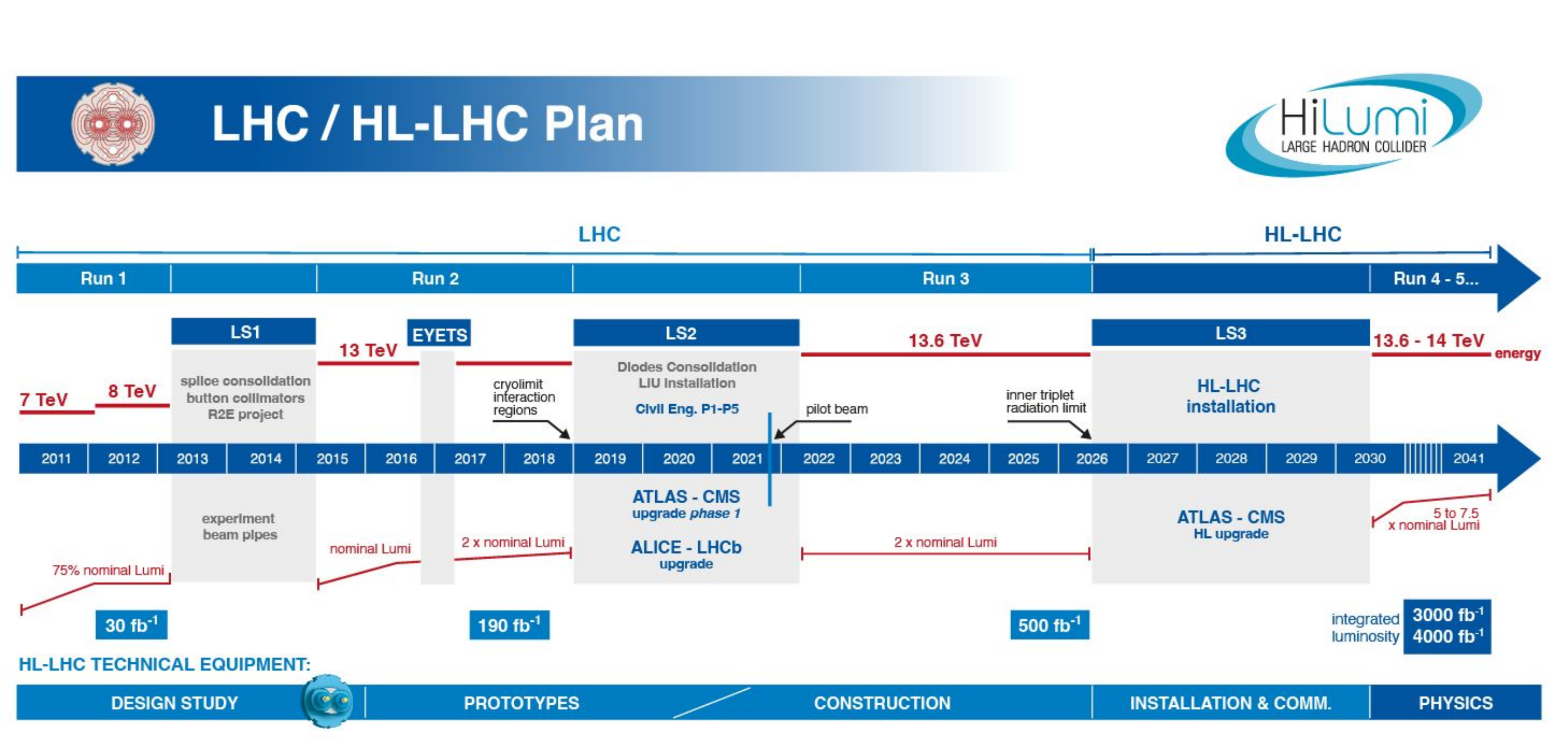
1: Search for CP Violation in ttH multilepton

- Introduction of the Physics and the Motivation
- ttH analysis and usage of the CP-BDT
- Training of the Models
- Performance of the Models

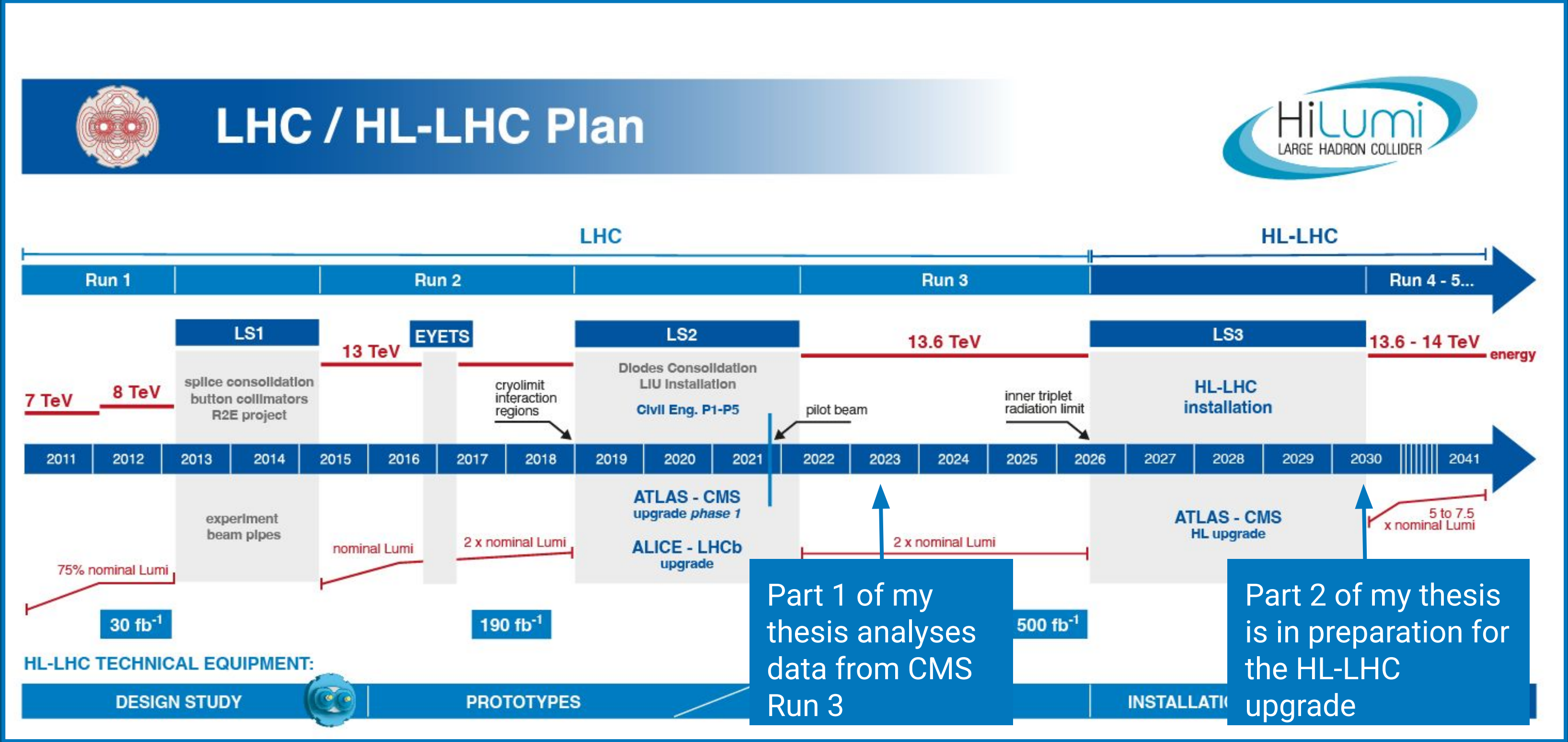
2: Generalization of the Broken Line Fit in CMSSW

- Introduction of the CMS Tracker and tracking
- The Broken Line fit
- Objectives

LHC Timeline

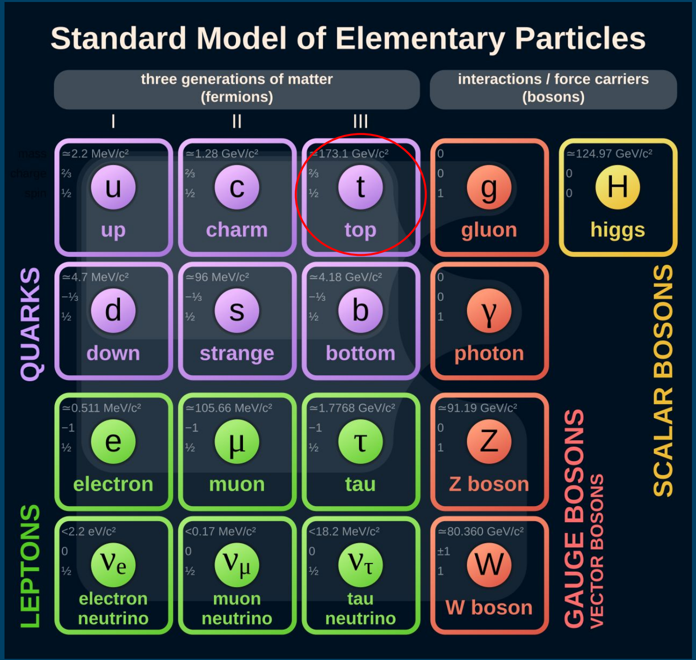


LHC Timeline



Part 1

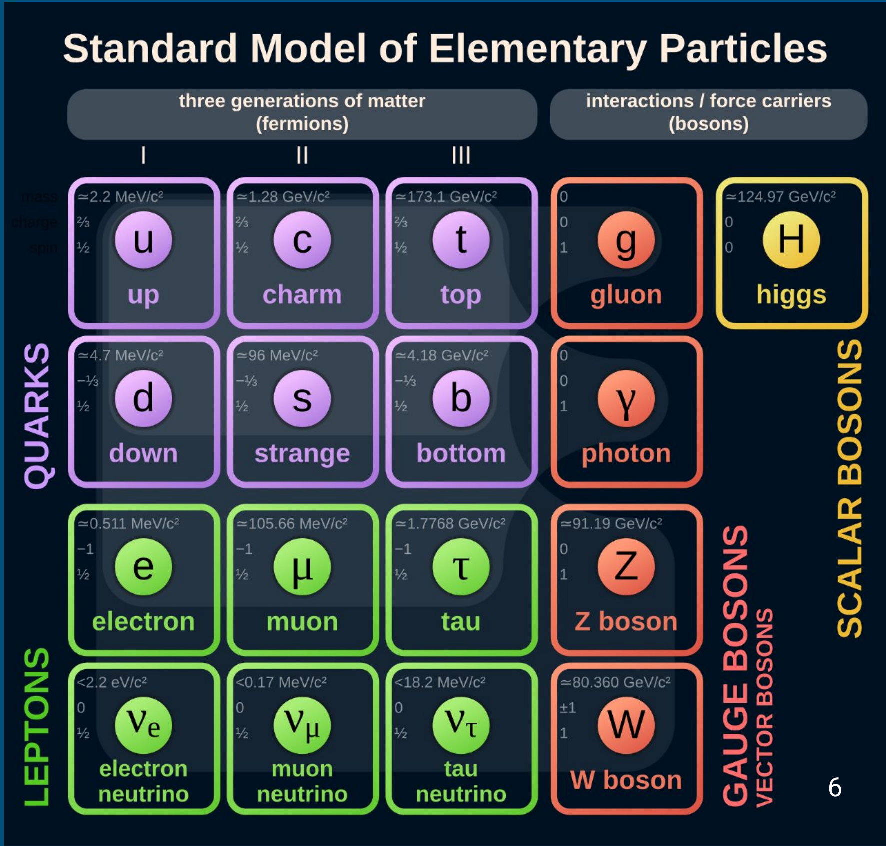
Search for CP-violation in ttH multilepton final state



In the Standard Model (SM), the heaviest elementary particle of the Standard Model is the top quark

Introduction: the Top Quark

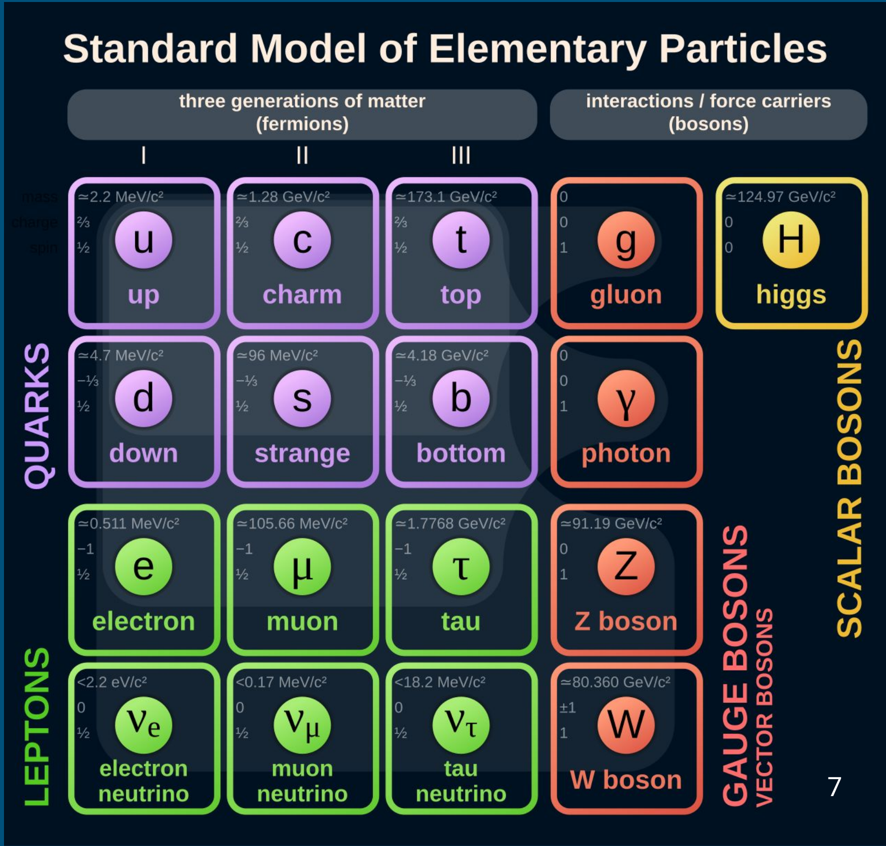
The top quark (t) has unique properties when compared to the other quarks:



Introduction: the Top Quark

The top quark (*t*) has unique properties when compared to the other quarks:

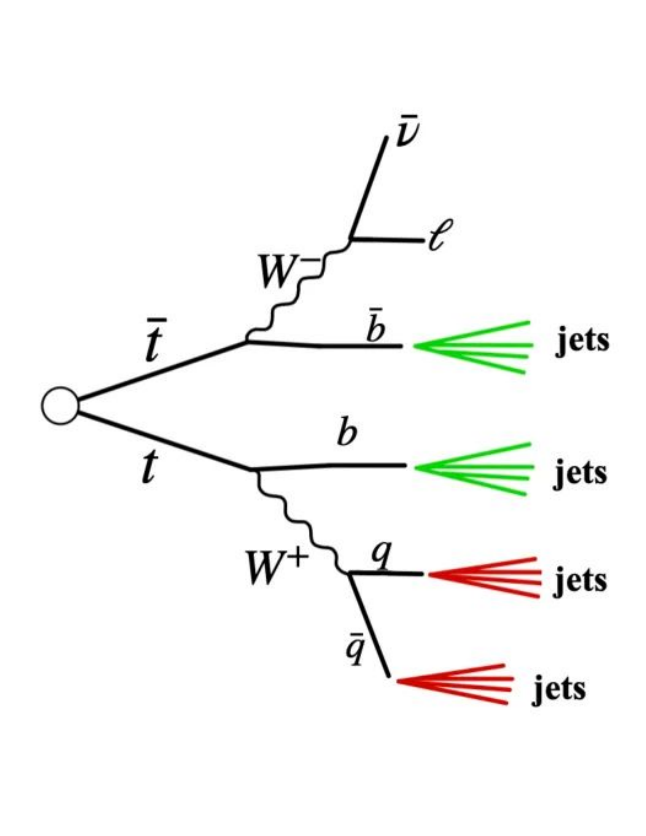
1. It has a mass of **171,77±0,38 GeV**, over 2 orders of magnitude larger than that of all other quarks;
2.



Introduction: the Top Quark

The top quark (*t*) has unique properties when compared to the other quarks:

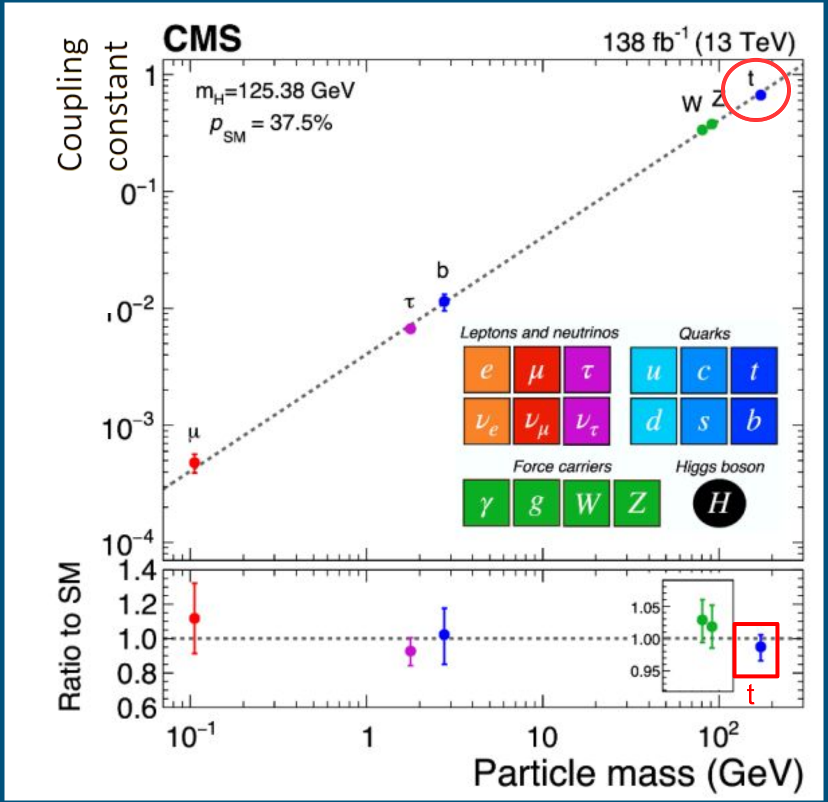
1. It has a mass of **171,77±0,38 GeV**, over 2 orders of magnitude larger than that of all other quarks;
2. Its lifetime is just **5×10^−25 s**, too short to decay hadronically like other quarks. It decays instead mostly into ***W+b***, and the W decays hadronically or leptonically;



Introduction: the Top Quark

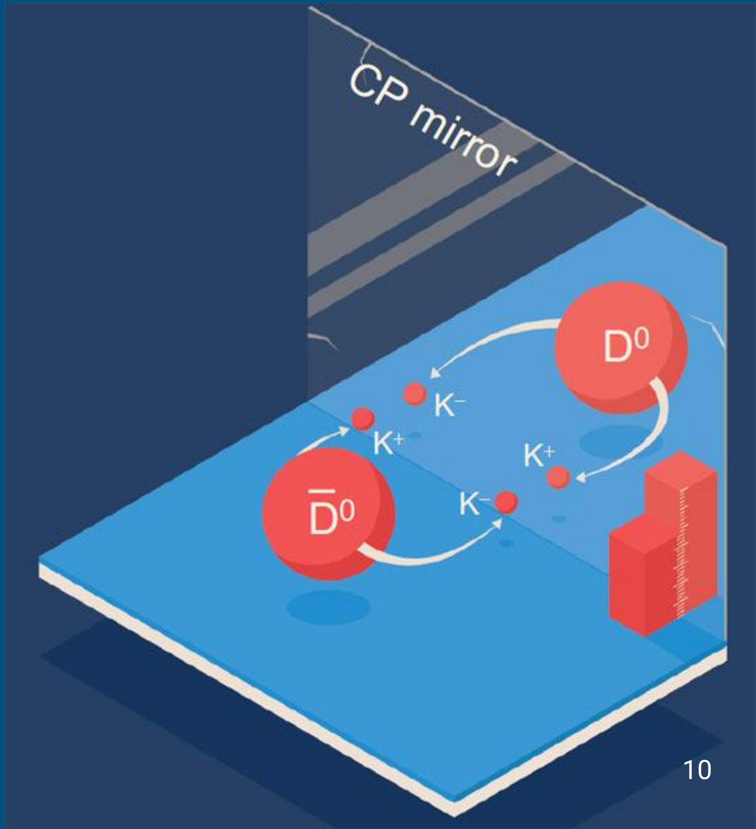
The top quark (t) has unique properties when compared to the other quarks:

1. It has a mass of **171,77±0,38 GeV**, over 2 orders of magnitude larger than that of all other quarks;
2. Its lifetime is just **5×10^−25 s**, too short to decay hadronically like other quarks. It decays instead mostly into ***W+b***, and the W decays hadronically or leptonically;
3. It has an unusually high coupling to the Higgs field, measured in CMS Run 2 to be **0.95 ±0.08**. Predictions for HL-LHC foresee that the precision on this measurement will increase to 4%



Introduction: CP symmetry and its violation

Charge Parity (CP) symmetry states that the laws of physics should be the same if a particle is interchanged with its antiparticle while its spatial coordinates are inverted. The breaking of this symmetry is called CP-violation.



Introduction: CP simmetry and its violation

Charge Parity (CP) symmetry states that the laws of physics should be the same if a particle is interchanged with its antiparticle while its spatial coordinates are inverted. The breaking of this symmetry is called CP-violation.

The SM allows for a few examples of CP-violation, which gives us η, the measure of matter-antimatter imbalance.

$$\eta = \frac{n_B - n_{\bar{B}}}{n_\gamma}$$

$$\eta_{\text{SM CP}} \sim 10^{-20}$$

Introduction: CP simmetry and its violation

Charge Parity (CP) symmetry states that the laws of physics should be the same if a particle is interchanged with its antiparticle while its spatial coordinates are inverted. The breaking of this symmetry is called CP-violation.

$$\eta = \frac{n_B - n_{\bar{B}}}{n_\gamma}$$

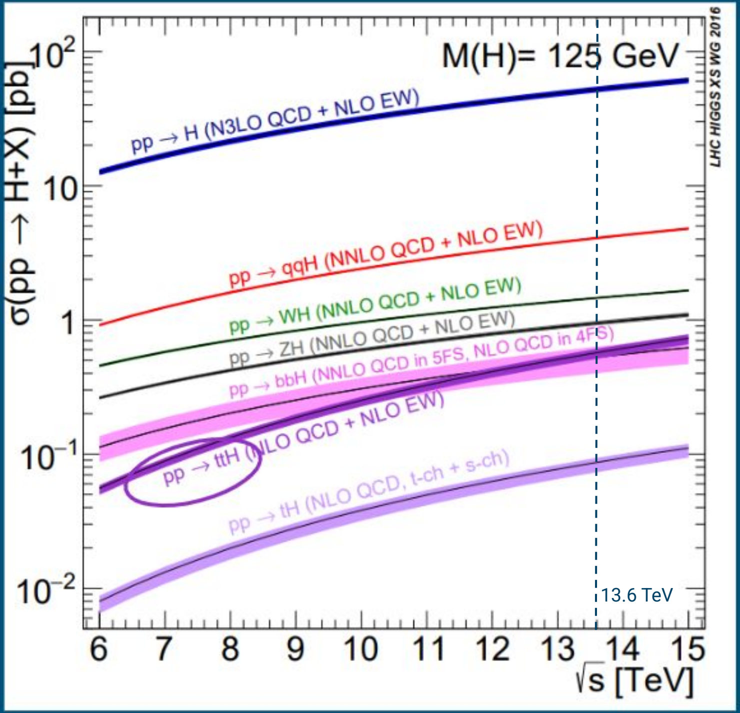
$$\eta_{\text{SM CP}} \sim 10^{-20}$$

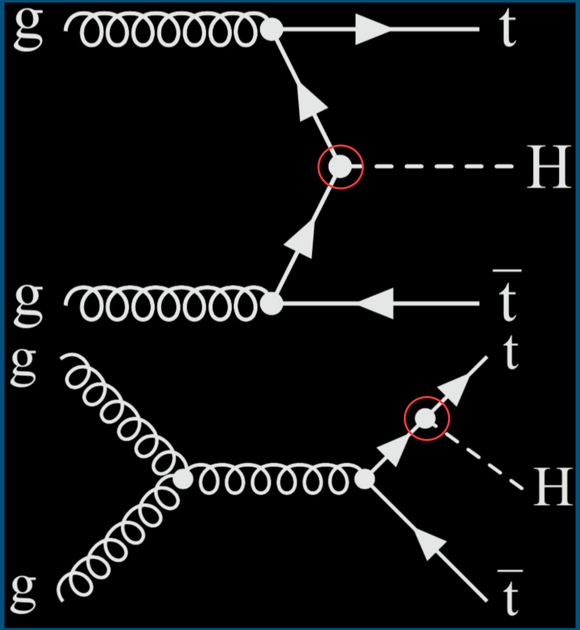
The SM allows for a few examples of CP-violation, which gives us η, the measure of matter-antimatter imbalance.

However, the value we observe in our universe is many orders of magnitude larger => There must be other sources of CP-violation the SM doesn't take into account.

$$\eta = (6.047 \pm 0.074) \times 10^{-10}$$

Introduction: the ttH process

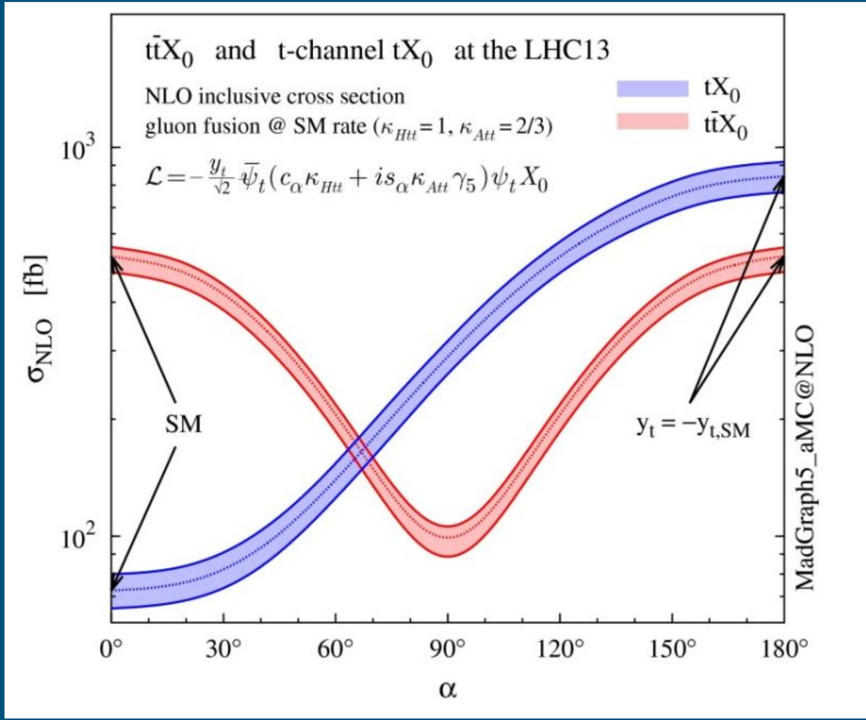




Introduction: CP symmetry in the ttH process

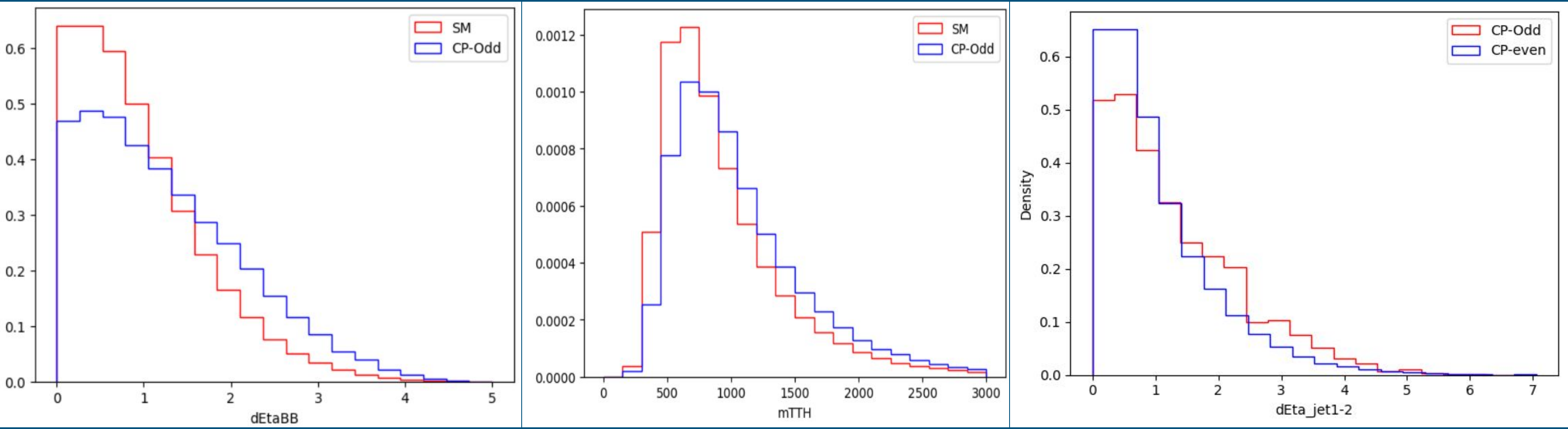
$$\mathcal{L} = -\frac{y_t}{\sqrt{2}}\,\bar{\psi}_t\big(\underbrace{c_\alpha \kappa_{Htt}}_{\text{CP-even}} + \underbrace{i s_\alpha \kappa_{Att}\gamma_5}_{\text{CP-odd}}\big)\psi_t X_0$$

- **α** is the CP mixing angle (0 or 180° in SM)
- $\kappa_{Htt,Att}$ are dimensionless rescaling parameters
- **c_α and s_α** are respectively cos(α) and sin(α), meaning the CP-even and CP-odd terms of the interaction
- y_t is the Yukawa coupling constant of the top quark to the Higgs field
- X0 labels a generic spin-0 particle with CP-violating coupling (in this case, the Higgs boson)



But the angle **α** doesn't affect only the cross-section of the process: it also influences the **kinematics** of it.

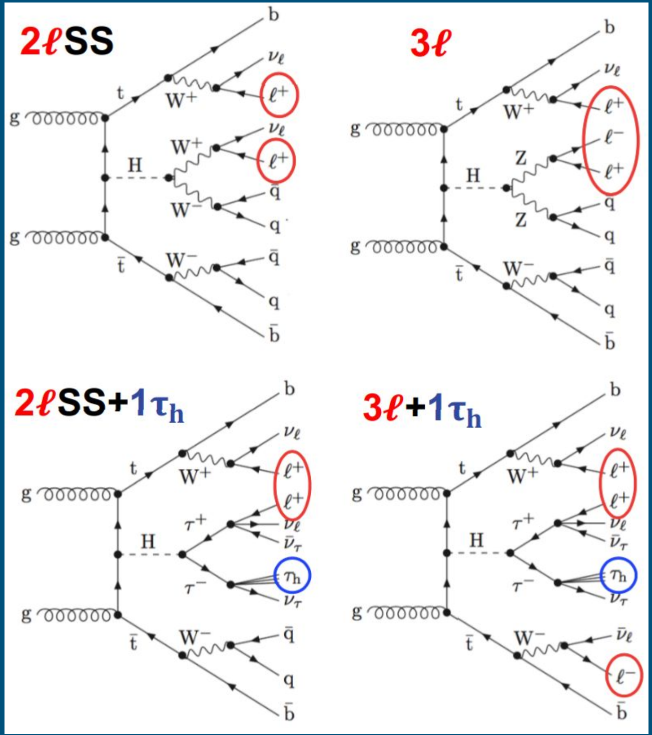
Introduction: CP symmetry in the ttH process

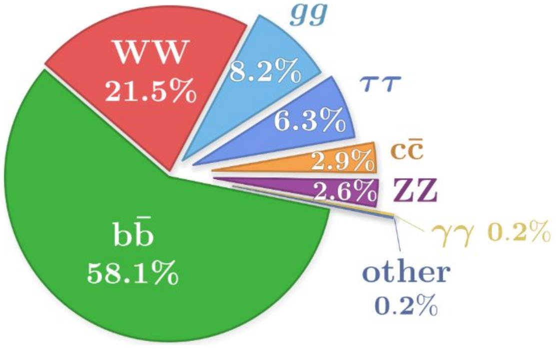


Comparison of distributions of different kinematic variables of the ttH process, for the CP-even and the CP-odd case, renormalized.

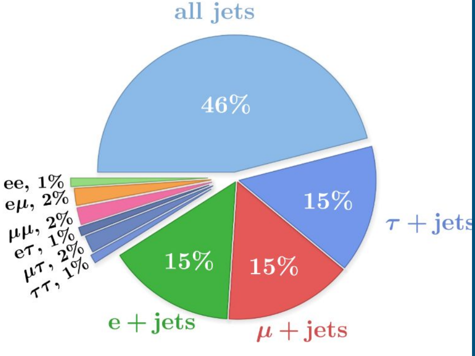
Introduction: the ttH Analysis

As explained, *t* usually decays into W and b, meanwhile *H* decays in either a WW couple or a ZZ couple in the multileptonic case. These can combine to form multiple combinations of final products. We classify them into categories based on the number of leptons and taus.





Higgs decay products

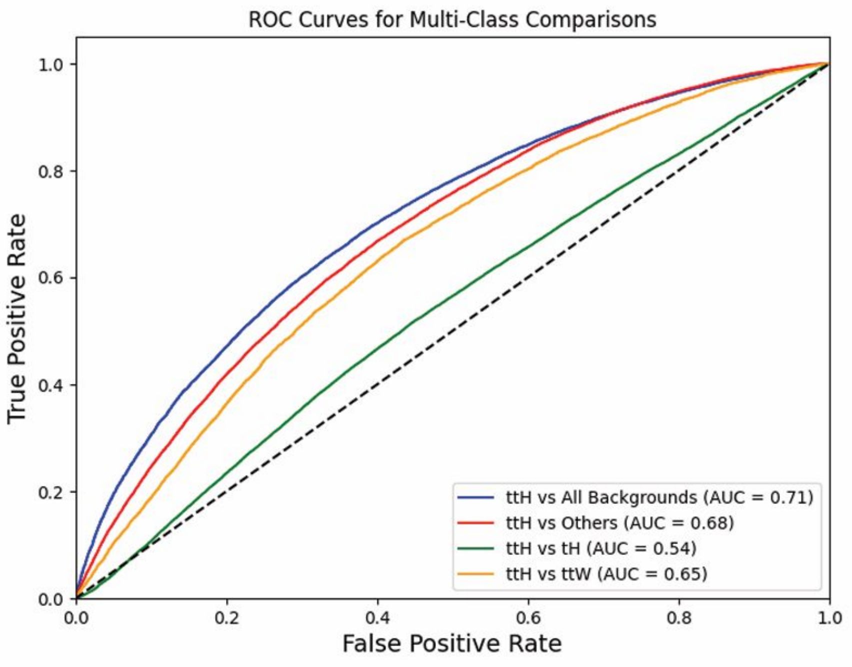


ttbar decay products

Introduction: the ttH Analysis

As explained, *t* usually decays into W and b, meanwhile *H* decays in either a WW couple or a ZZ couple in the multileptonic case. These can combine to form multiple combinations of final products. We classify them into categories based on the number of leptons and taus.

Since other processes (such as ttW and tH) besides the ttH process can give us the same final products, we utilize a Deep Neural Network (DNN). This is a machine learning algorithm, where each output node trained to distinguish a specific process category.



ROC curve for the multiclass DNN utilized on 2022 data for the 2lss0tau category

Introduction: the ttH Analysis

Once the DNN assigns a score to each event, to determine what process it might be, we can apply to ttH-like events another algorithm, called a Boosted Decision Tree (BDT), to determine whether it is CP-even or CP-odd.

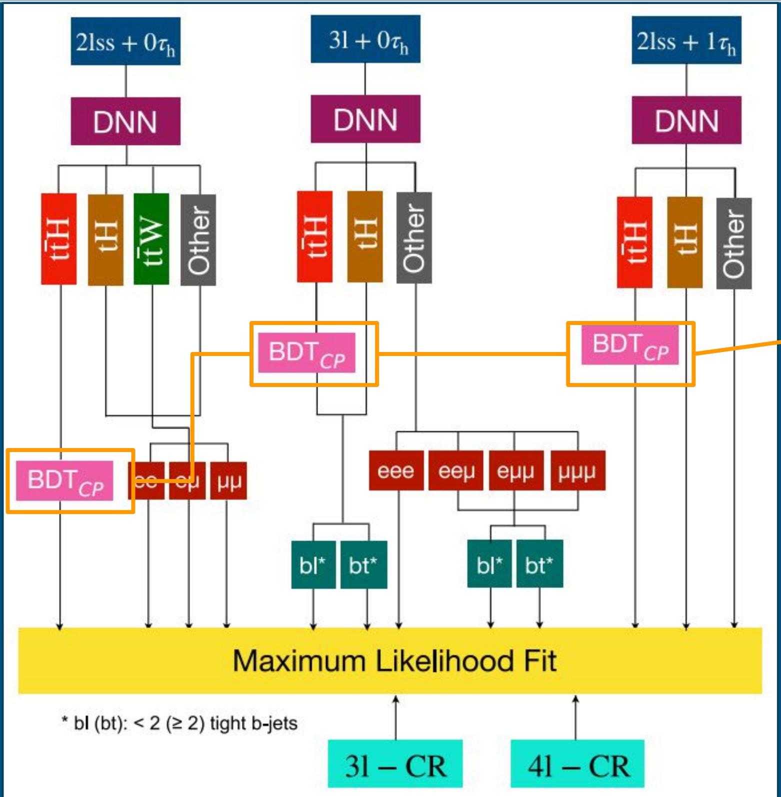
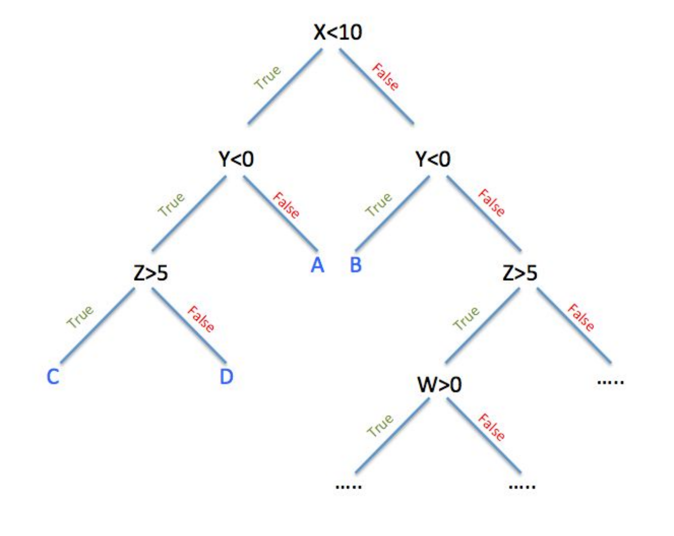
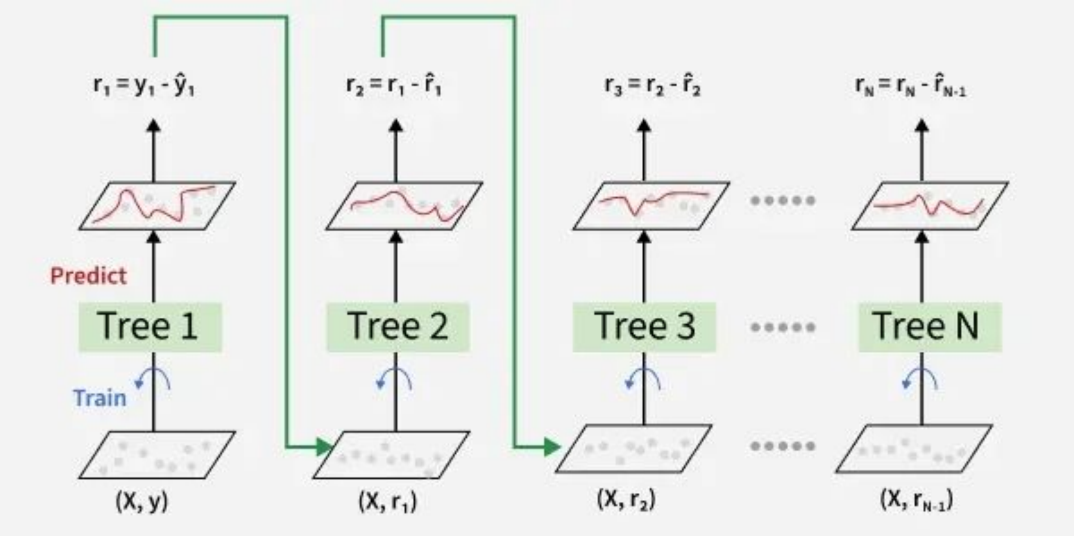


Diagram of the analysis process for ttH

Boosted Decision Trees (BDTs) which discriminate CP-odd from CP-even events in the DNN ttH node based on the score of the CP classifiers

ttH Analysis and usage of the CP-BDT





Example of a Binary Decision Tree

Example of a Boosted Decision Tree (BDT). When creating the training data for the second tree, we apply to misidentified events a weight α = (1 - r_1)/r_1

Training: Dataset

- The BDT was trained on ttH Monte Carlo samples, with both CP-even and CP-odd weight.

- eras = 2022 and 2023 MC samples, used inclusively for the training

- The signal regions analyzed are 2lss0tau, 2lss1tau and 3l0tau. For now, all events of the signal regions were used, without selecting the ttH node of the multi-target DNN

- The signal was taken as the events with the CP-odd weight, meanwhile the background was taken as the events with the SM weight

- Split into Training and Validation in a ratio 4:1

Training: Input Variables

2lss0tau

Tabella 5: Variables definitions 2lss0τ

Variable Name	Definition
SelJet_pt	pT of leading jet
SelJet_Eta	η of leading jet
SelJet_Phi	ϕ of leading jet
SelJet_Mass	Mass of leading jet
SelJet_isBtag	Btag class of the leading jet
SelJet_isFromHadTop	Whether the leading jet comes from the hadronic top
SelJet_BTagDeepFlavB	Deep flavour Btag of the leading jet
mindRlep1jet	dR of lep 1 to its closest jet
mindRlep2jet	dR of lep 2 to its closest jet
mTTH	invariant mass of jets+met+leptons
dEtaBB	dEta of two jets with highest b tagging score
dEtaLL_BBframe	dη of the two leptons in the B-B system frame
avg_dr_jet	average dR distance among all jets
dEtaBB_LLframe	dEta BB in the l-l system frame
Hj_tagger_hadTop	Higgs-jet tagger
HTT_score	highest BDT score of jet triplet from t
met_phi	ϕ of met
nSelJets	number of jets passing the cuts
met	missing transverse energy

What changed in run3:

- HiggsJet_tagger (that tells whether a jet comes from H->WW decay) was not included because unavailable
- SelJet_BTagDeepFlavB was not included
- theta_higgs_ttbar_ttHsystem and theta_toptop_ttbarframe were not included because of previously faulty definition

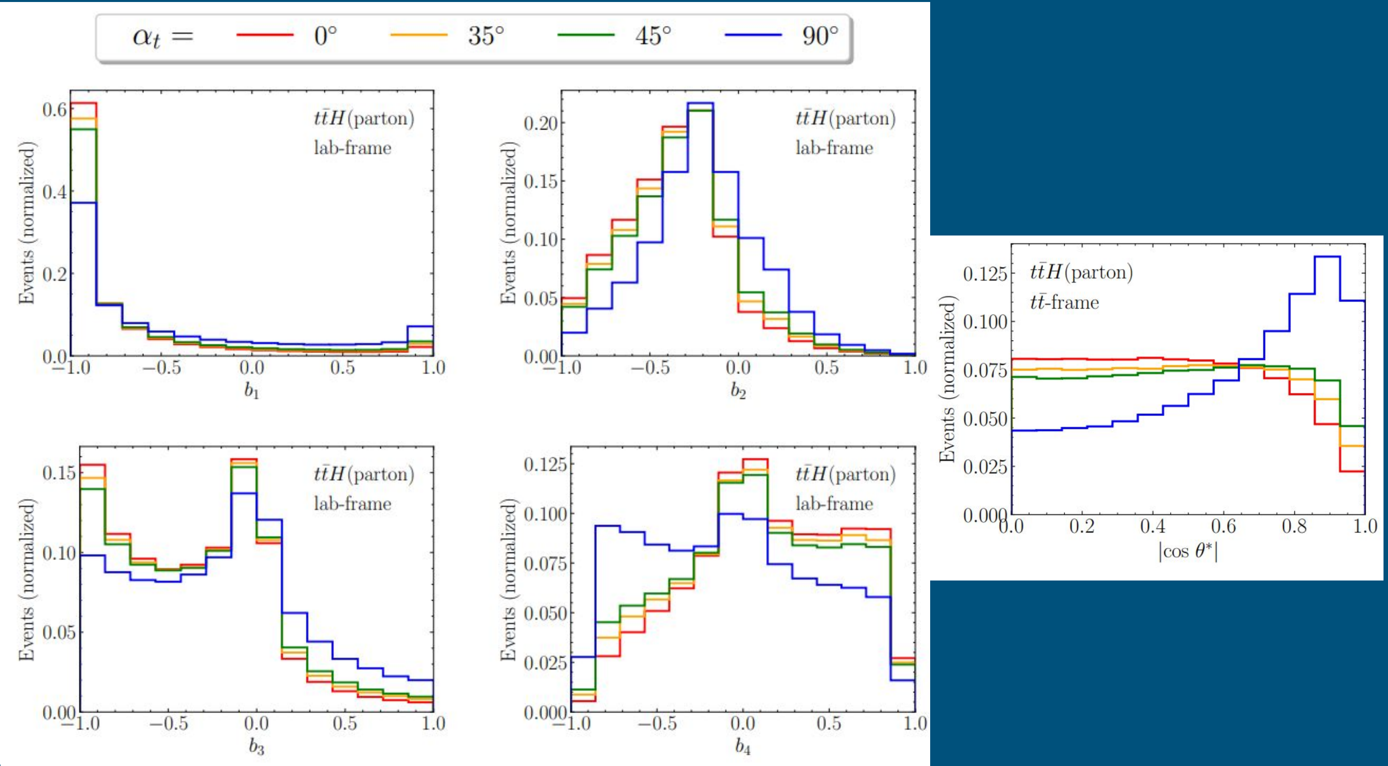
Training: Added Variables

To the input variables already listed, I have added 17 of those discussed in the paper [CP-sensitive simplified template cross-sections for ttH (H. Bahl, 2024, arXiv:2406.03950)] to all three regions.

|cosθ*| is calculated in the ttbar frame, meanwhile b1, b2, b3 and b4 are calculated in the ttH frame

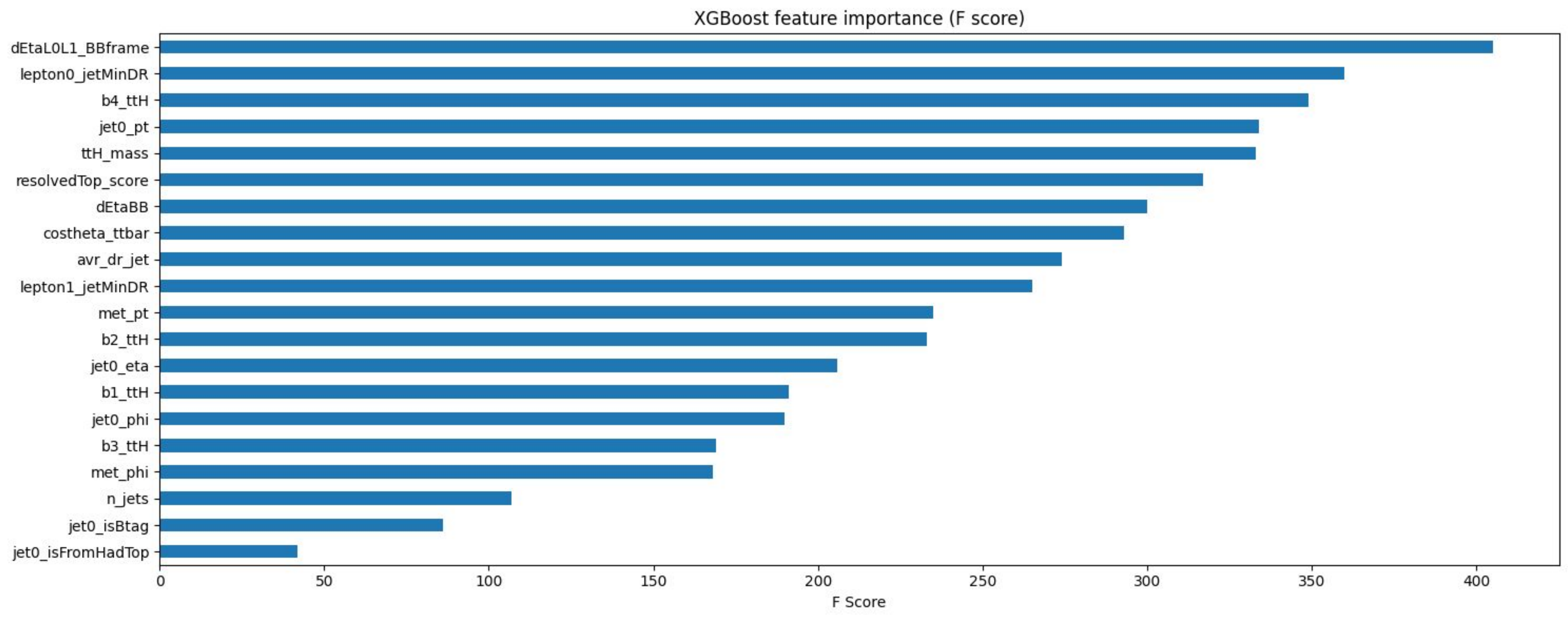
$\lvert \cos\theta^* \rvert$	$\frac{\lvert \boldsymbol{p}_t \cdot \boldsymbol{n} \rvert}{\lvert \boldsymbol{p}_t \rvert \cdot \lvert \boldsymbol{n} \rvert}$
b_1	$\frac{(\boldsymbol{p}_t \times \boldsymbol{n}) \cdot (\boldsymbol{p}_{\bar{t}} \times \boldsymbol{n})}{p_{T,t} p_{T,\bar{t}}}$
b_2	$\frac{(\boldsymbol{p}_t \times \boldsymbol{n}) \cdot (\boldsymbol{p}_{\bar{t}} \times \boldsymbol{n})}{\lvert \boldsymbol{p}_t \rvert \ \lvert \boldsymbol{p}_{\bar{t}} \rvert}$
b_3	$\frac{p_t^x \ p_{\bar{t}}^x}{p_{T,t} p_{T,\bar{t}}}$
b_4	$\frac{p_t^z \ p_{\bar{t}}^z}{\lvert \boldsymbol{p}_t \rvert \ \lvert \boldsymbol{p}_{\bar{t}} \rvert}$

Training: Added Variables



Distribution of the variables for multiple values of α, from the paper arXiv:2406.03950

Training: Feature importance



Training: Hyperparameter Optimizaiton

When training a Machine Learning algorithm (in our case a BDT), the variables that define or constrain the properties of that algorithm are called **Hyperparameters**.

For BDTs:
- learning_rate is the weight applied to the correction from one era to the next;
- n_estimators is the number of trees;
- max_depth is the maximum number of consecutive cuts that can be present in the tree;
- subsample and colsample_bytree are the sizes of random subsection of samples or variables used;
- early_stopping interrupts the training when overtraining is detected, at least after a number of ages equal to early_stopping_rounds.

Tabella 1: HyperParameters

Hyperparameter	Updated Range
learning_rate	[0.1, 4]
n_estimators	[1500]
max_depth	[3, 6]
subsample	[0.8, 1]
colsample_bytree	[0.8, 1]
gamma	[0, 3]
early_stopping	[True]
early_stopping_rounds	[2000]

Training: Hyperparameter Optimizaiton

Optuna is a define-by-run tool (meaning that it allows the user to dynamically construct the search space) that allows us to automatically search for the best hyperparameter combination to optimize a certain objective.

It works by using a **Bayesian Optimization Algorithm** to either minimize or maximize an “objective” function given to it by the user.

The process (called “study”) is repeated for a given number of times (called “trials”), and each time the hyperparameter set is based on the results of the previous trials.

It is framework agnostic, and compatible with XGBoost.

Training: Hyperparameter Optimizaiton

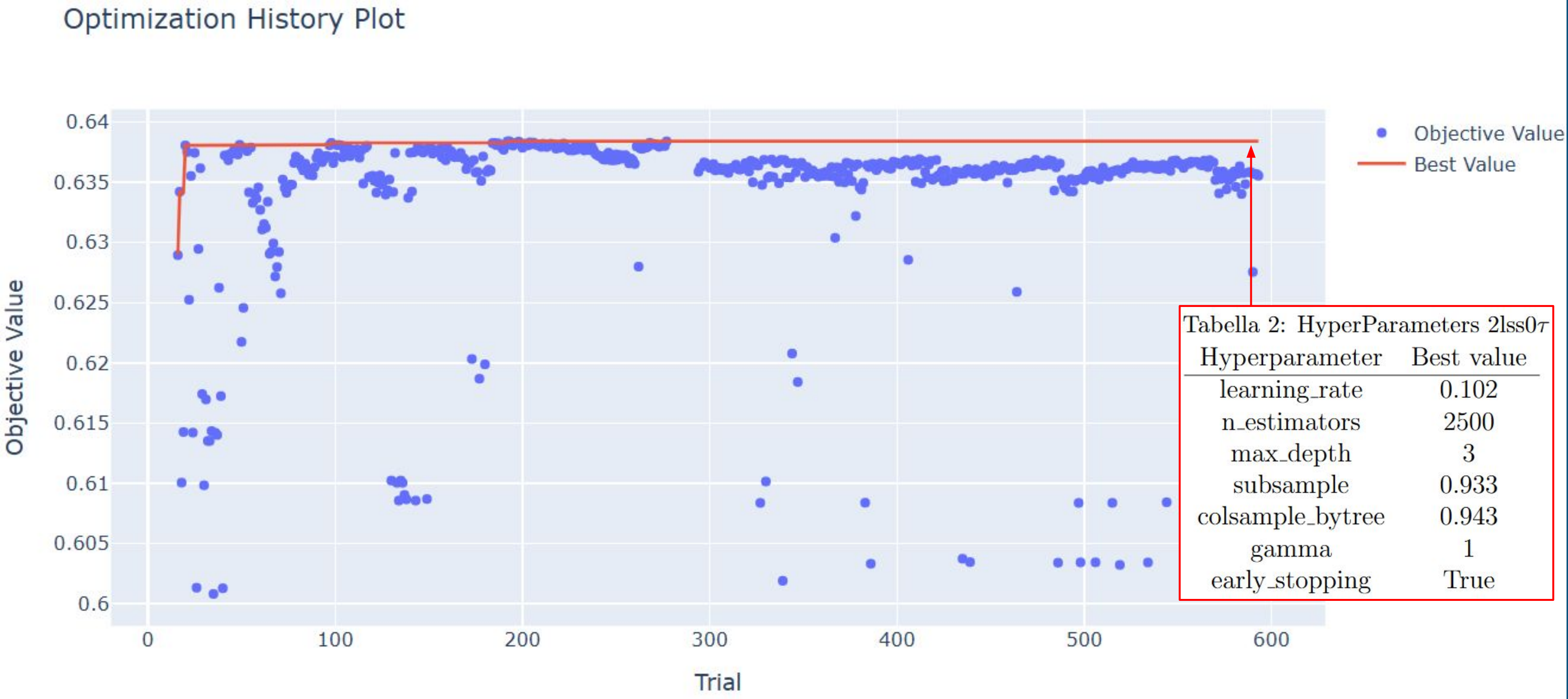
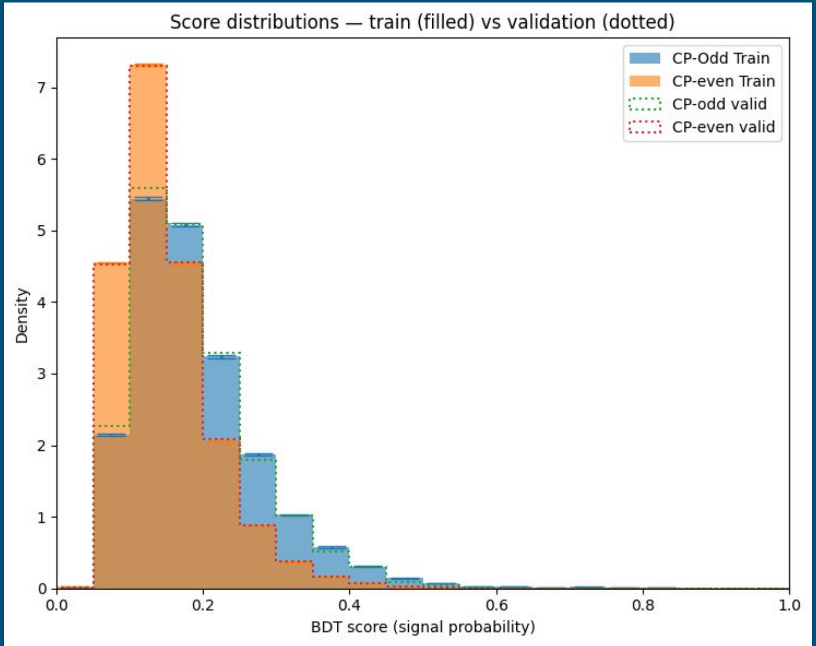


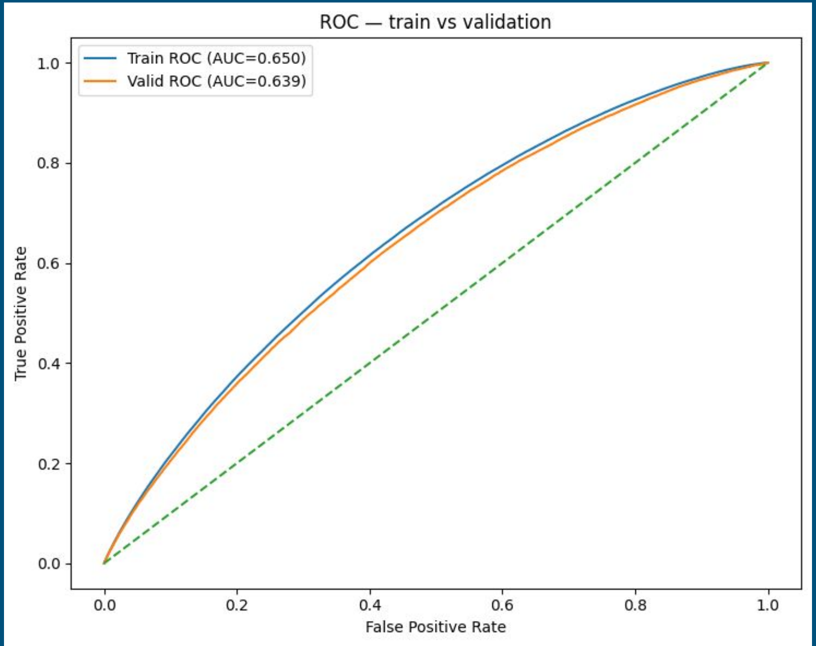
Tabella 2: HyperParameters 2lss0τ

Hyperparameter	Best value
learning_rate	0.102
n_estimators	2500
max_depth	3
subsample	0.933
colsample_bytree	0.943
gamma	1
early_stopping	True

AUC of the validation set for successive trials, for 2lss0tau

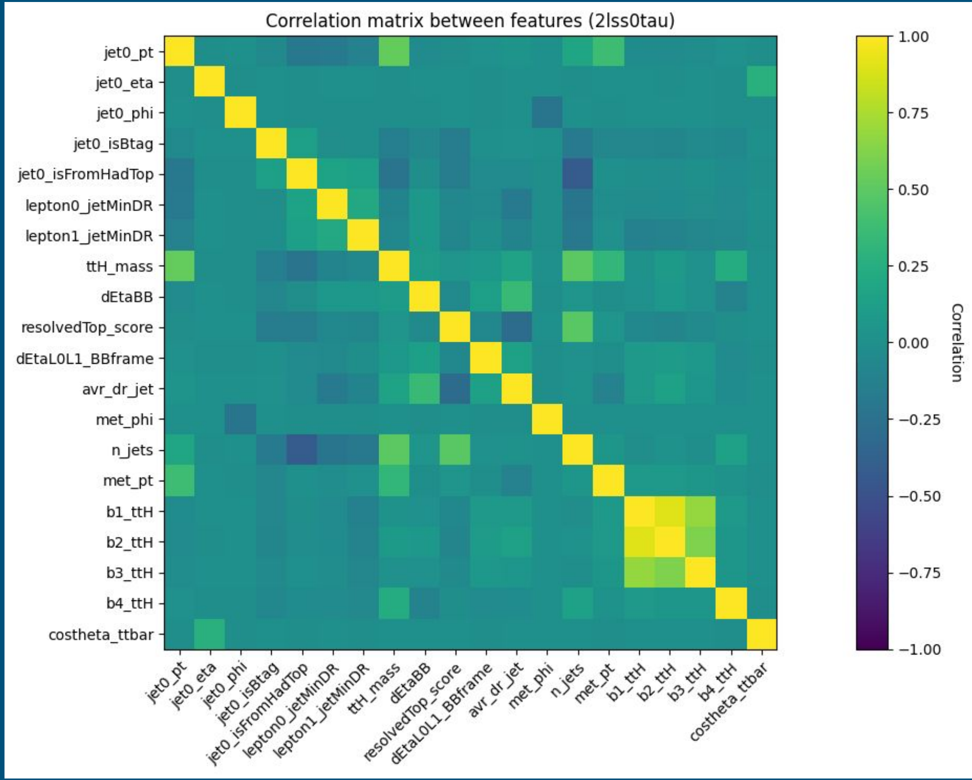
Performance: score and ROC curve (2lss0tau)





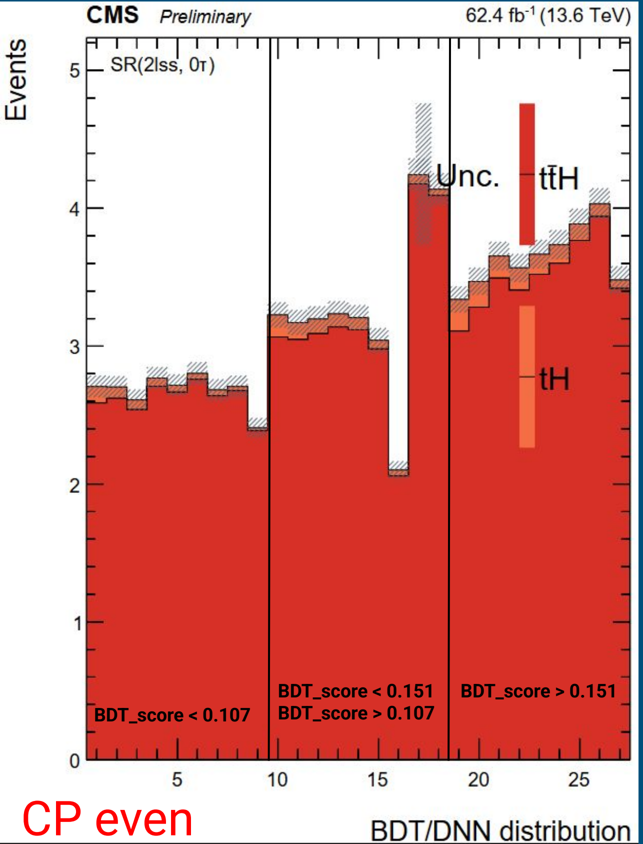
Predicted distributions for 2lss0tau for CP-even and CP-odd(left) and corresponding ROC curve with AUC=0.639 (right). In Run 2, the AUC of the ROC curve was 0.637

Performance: correlation matrix



Correlation Matrix of the features of the BDT, for 2lss0tau

Performance: usage in the analysis



CP even

Once the BDT is trained, we can apply it to the ttH node of the multi-class DNN, and observe its effective score distribution.

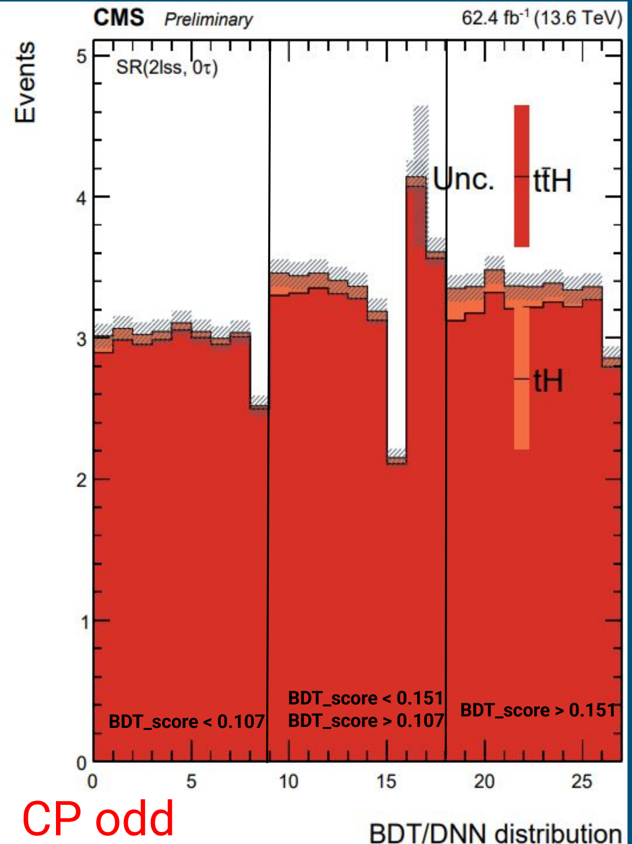
We then split the score in 3 regions:

- BDT_score < 0.107
- 0.107 < BDT_score < 0.151
- BDT_score > 0.151

We then further split the events in each region based on the score assigned by the DNN, to maximize the significance:

$$Z = \sum_{i=0}^{n_{bin}} \sqrt{2\left(N_i \ln \frac{N_i}{bkg_i} - sig_i\right)}$$

Performance: usage in the analysis



Once the BDT is trained, we can apply it to the ttH node of the multi-class DNN, and observe its effective score distribution.

We then split the score in 3 regions:

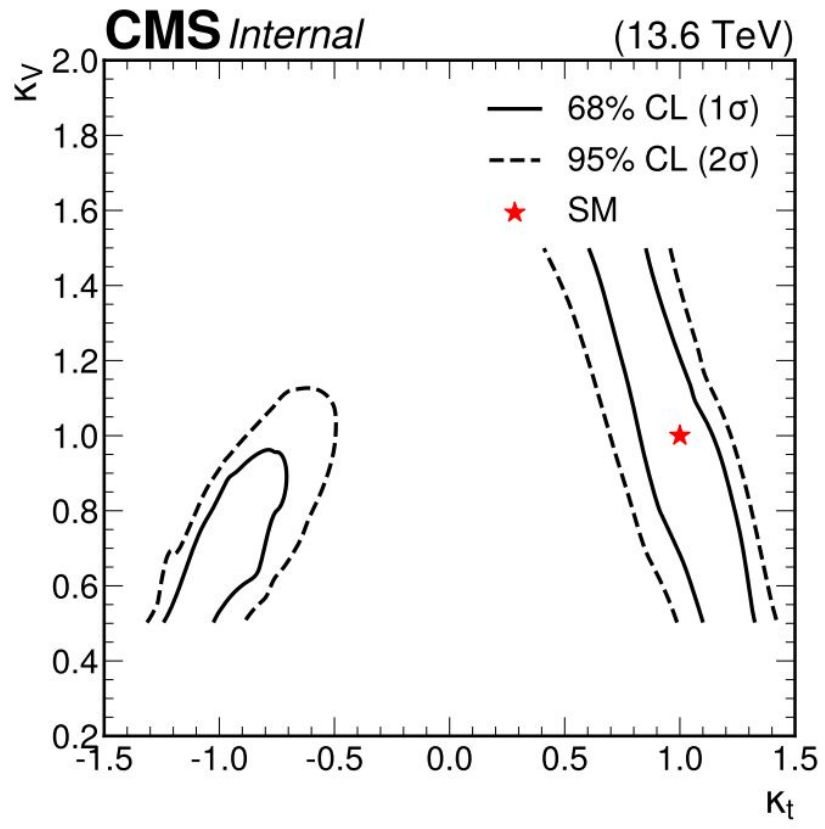
- BDT_score < 0.107
- 0.107 < BDT_score < 0.151
- BDT_score > 0.151

We then further split the events in each region based on the score assigned by the DNN, to maximize the significance:

$$Z = \sum_{i=0}^{n_{bin}} \sqrt{2\left(N_i \ln \frac{N_i}{bkg_i} - sig_i\right)}$$

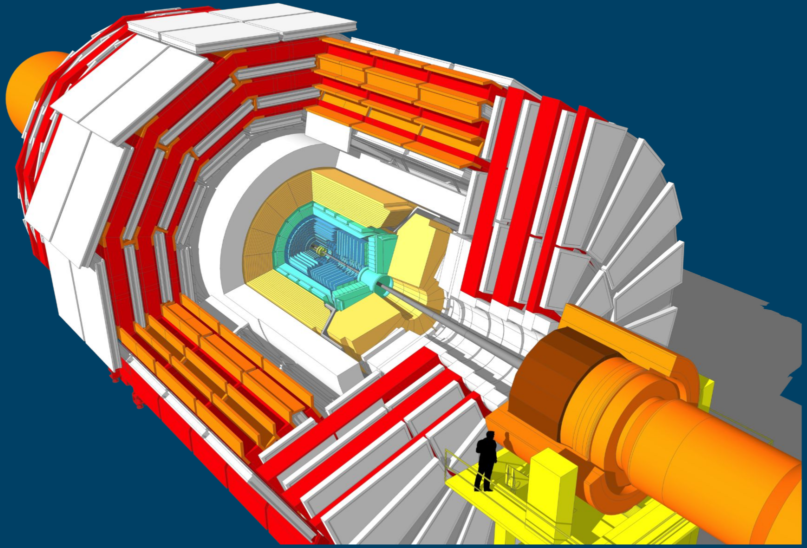
Performance

Confidence intervals are set using Maximum Likelihood scans, first look at the 1D likelihood scan of kappa_top looks reasonable to proceed with full interpretation → work in progress, this iteration does not include 3l and 4l CRs

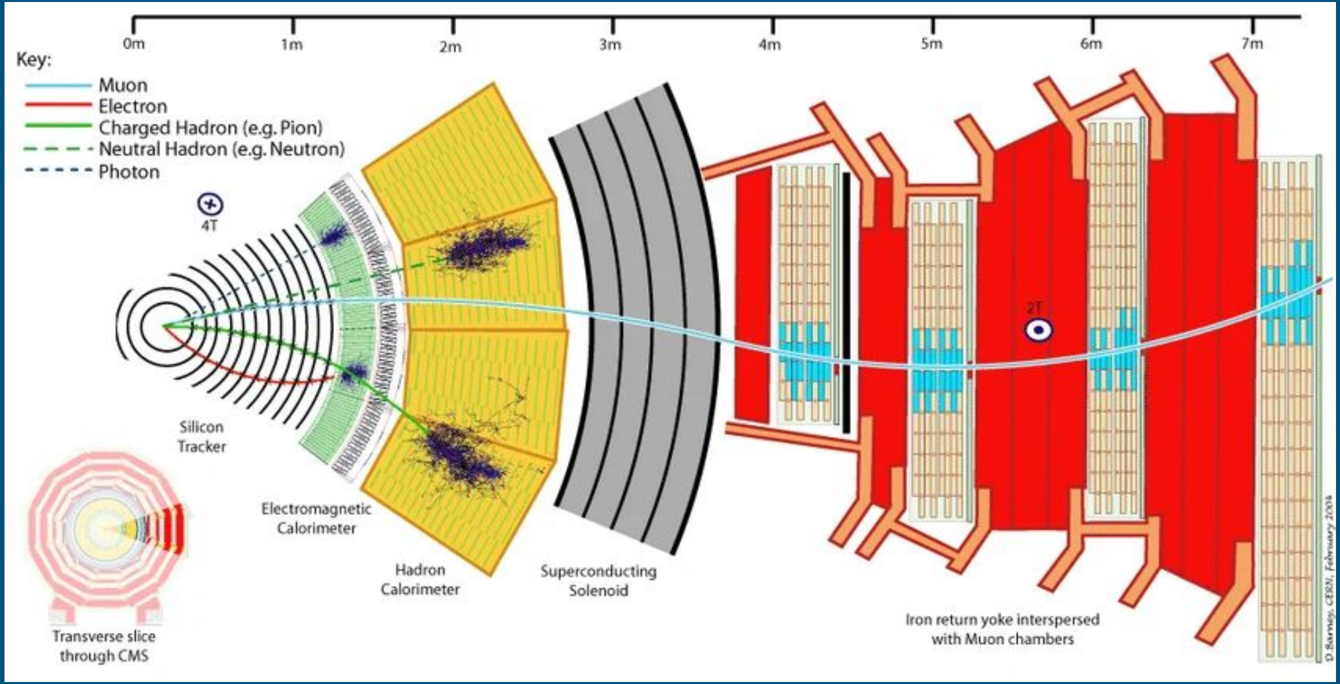


Part 2

Generalization of particle track fitting for the HL-LHC



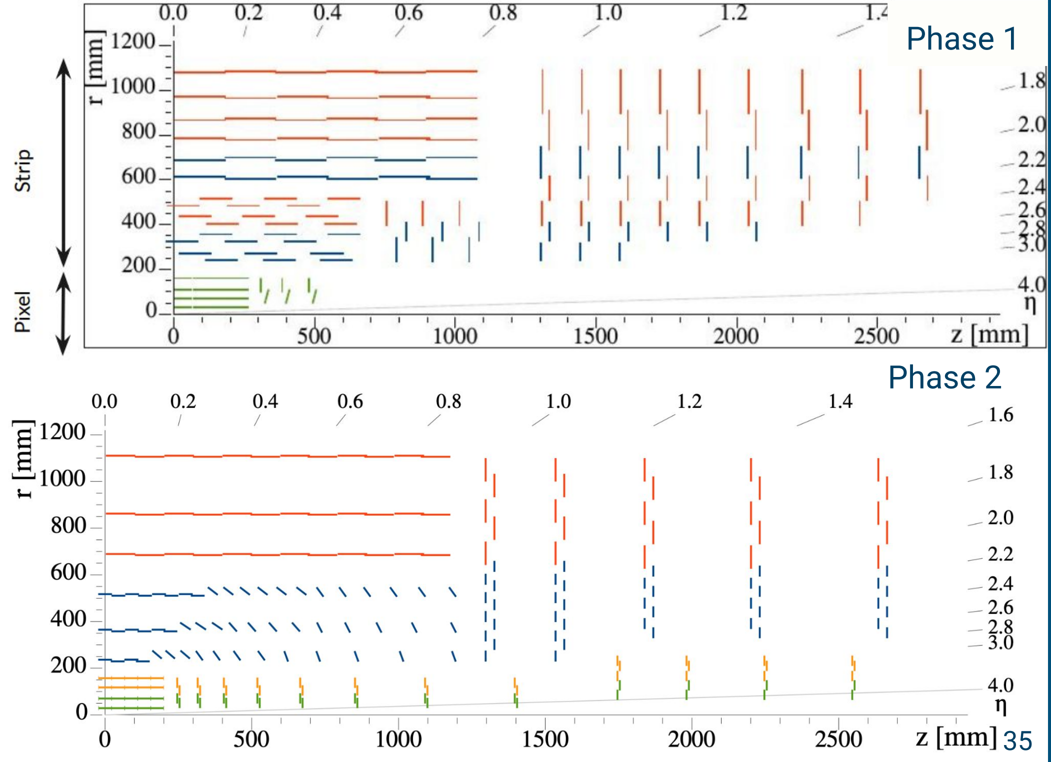
Introduction: CMS



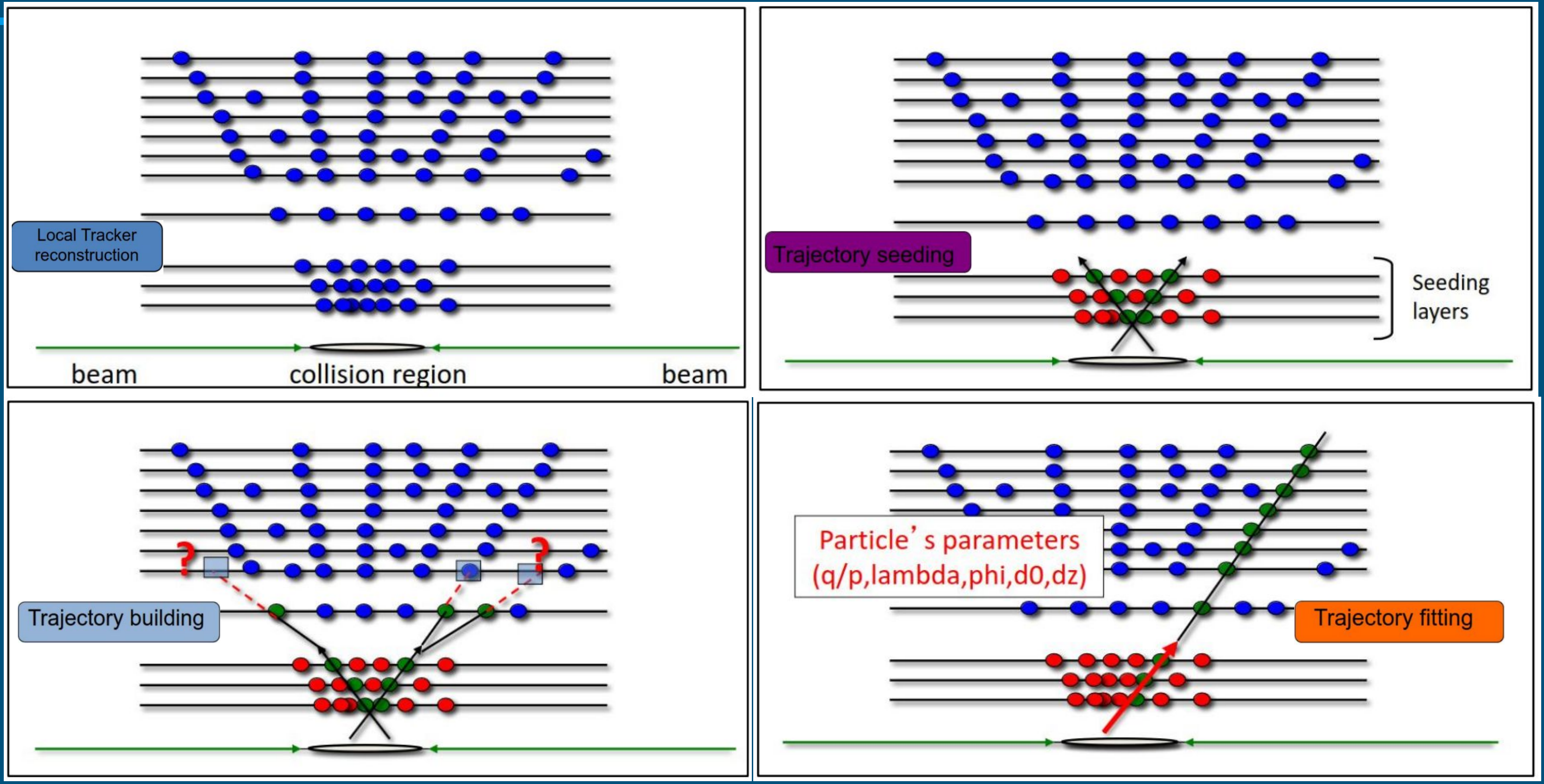
To reconstruct the collision at its centre, signals from multiple detectors are collected, to reconstruct the particles and their paths through the detector.

Introduction: the Tracker upgrade

In preparation for the HiLumi upgrade, CMS is improving its tracker, to better handle the high luminosity and the radiation that it will be exposed to.
Furthermore, the pixels that constitute it will be shrunk from 100 × 150 μm^2 to 25 × 100 μm^2, to allow for better granularity.
Finally, the coverage in η has been expanded from 2.5 to 4.0



Introduction: track reconstruction

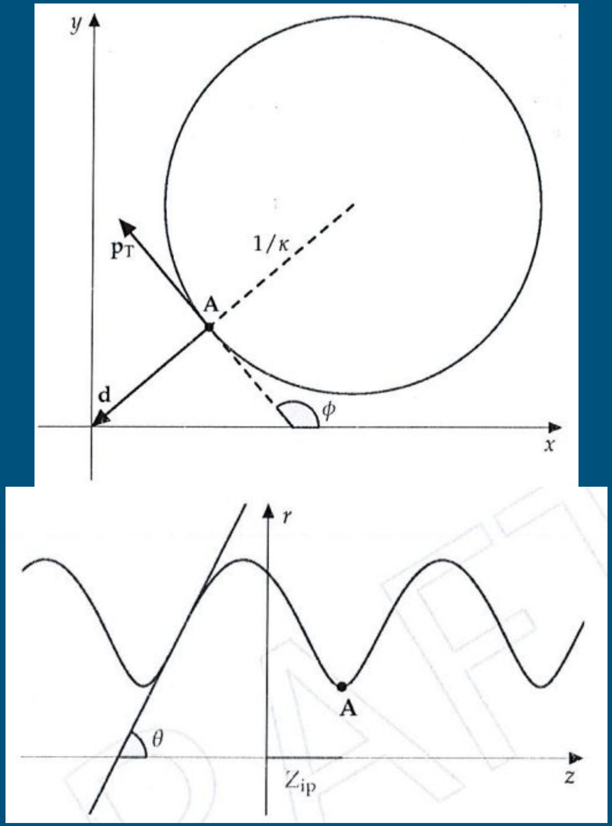


The Broken Line Fit

Since charged particles will move through the magnetic field of CMS, the tracks will be helicoidal.

To fit the trajectory of the particles in the pixel detector, the Broken Line fit method is commonly utilized, which uses the following parameters:

- the angle ϕ between pT in A and the positive x axis;
- the signed distance **d** between A and the z axis, positive if the following is a right handed system
- the signed curvature **k**, positive if the particle travels clockwise.



The Broken Line Fit

Starting from a quadruplet of hits, the fit follows three steps:

1. A fast pre-fit that assumes an ideal helicoidal path
2. The line fit, where a least square fit reconstructs the trajectory by determining a correction to the position of every measured hit, also providing the uncertainty of such correction
3. The circle fit, which works similarly to the line fit, but with a circular shape.

This fit, in steps 2 and 3, uses the rough estimates obtained in step 1 to take into account possible scatterings that the particle might go through during its path.

The Broken Line Fit

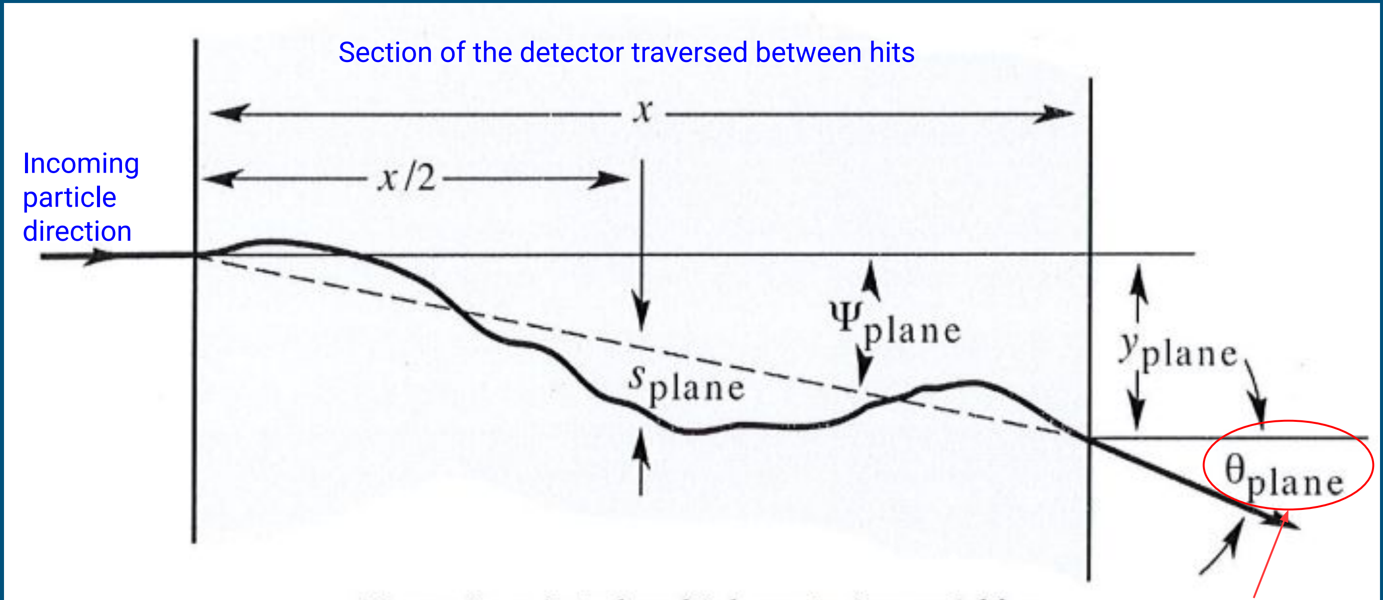
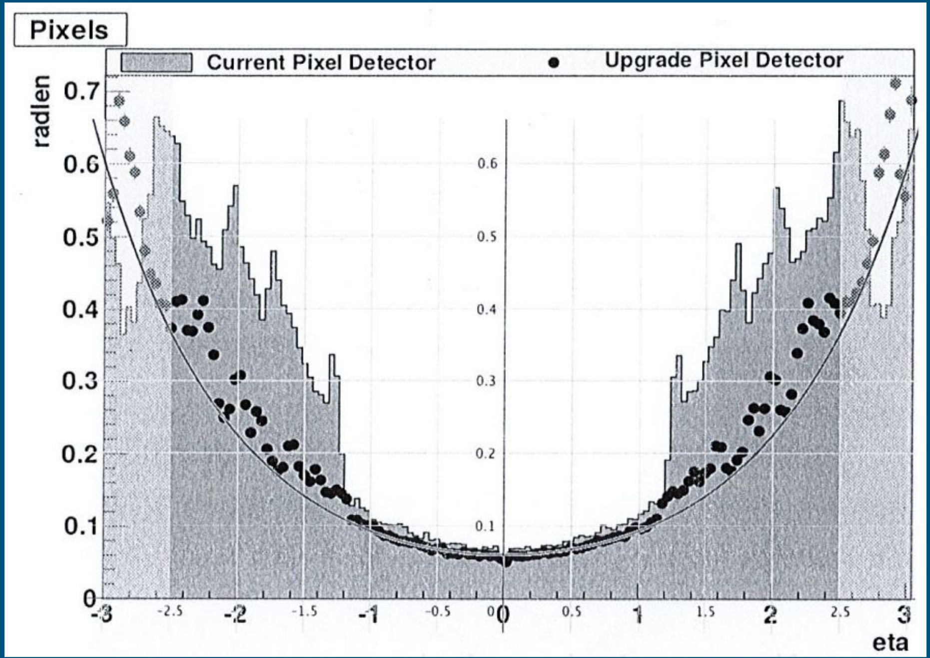


Figure 6: projected multiple scattering variables.

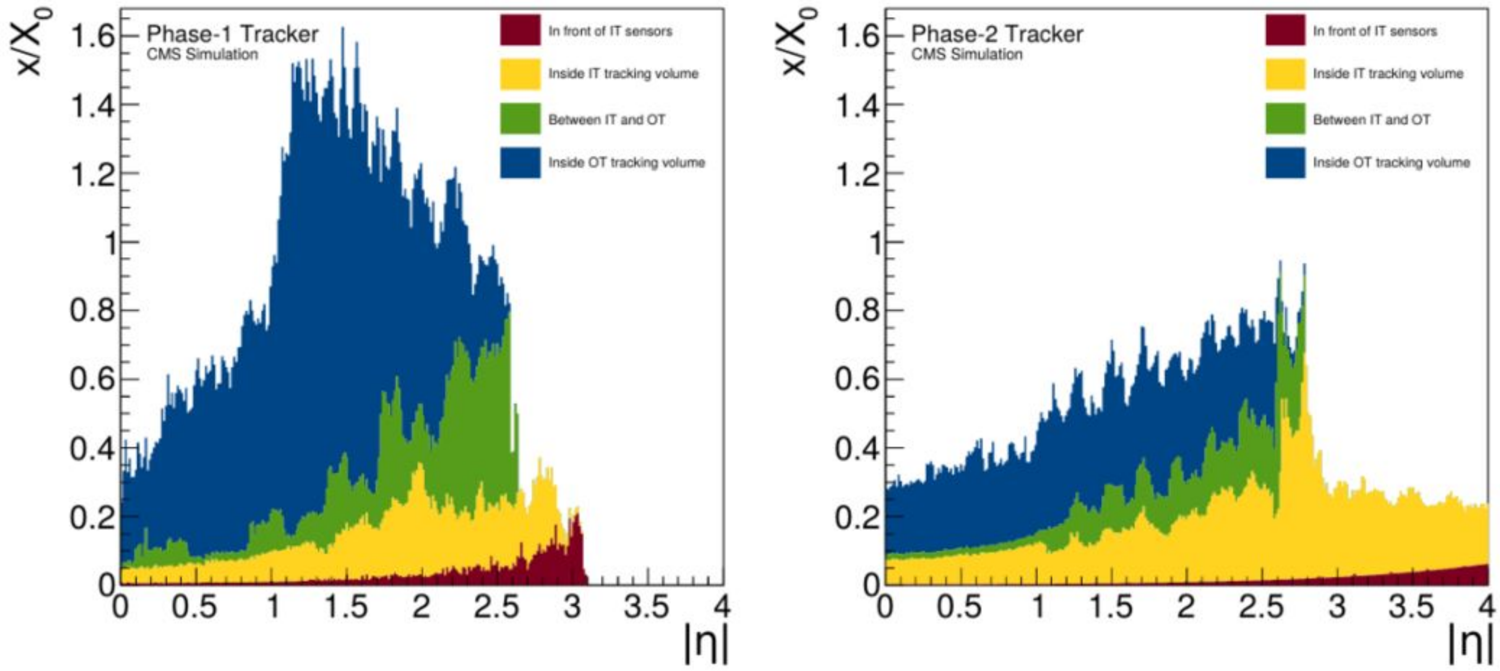
The Broken Line Fit

$$\theta_0 = \frac{13.6\ \text{MeV}}{\beta c p} z \sqrt{\frac{x}{X_0}} \left(1 + 0.038 \log\left(\frac{x}{X_0}\right)\right)$$

The deflection, as it can be seen here, depends on the ratio **x/X0**, meaning of the length of the pixel detector in units of radiation length, which was considered **uniform** throughout the detector



The Broken Line Fit: Phase 2



Comparison of the material budget for the tracker between Phase 1 and Phase 2. La Rosa, A. (2019). The CMS Outer Tracker for the High Luminosity LHC upgrade arXiv.1912.02061

Objectives

The objective on this 2nd part of my work is:

- recalculate the density and its effect on θ_0 for the HiLumi upgrade

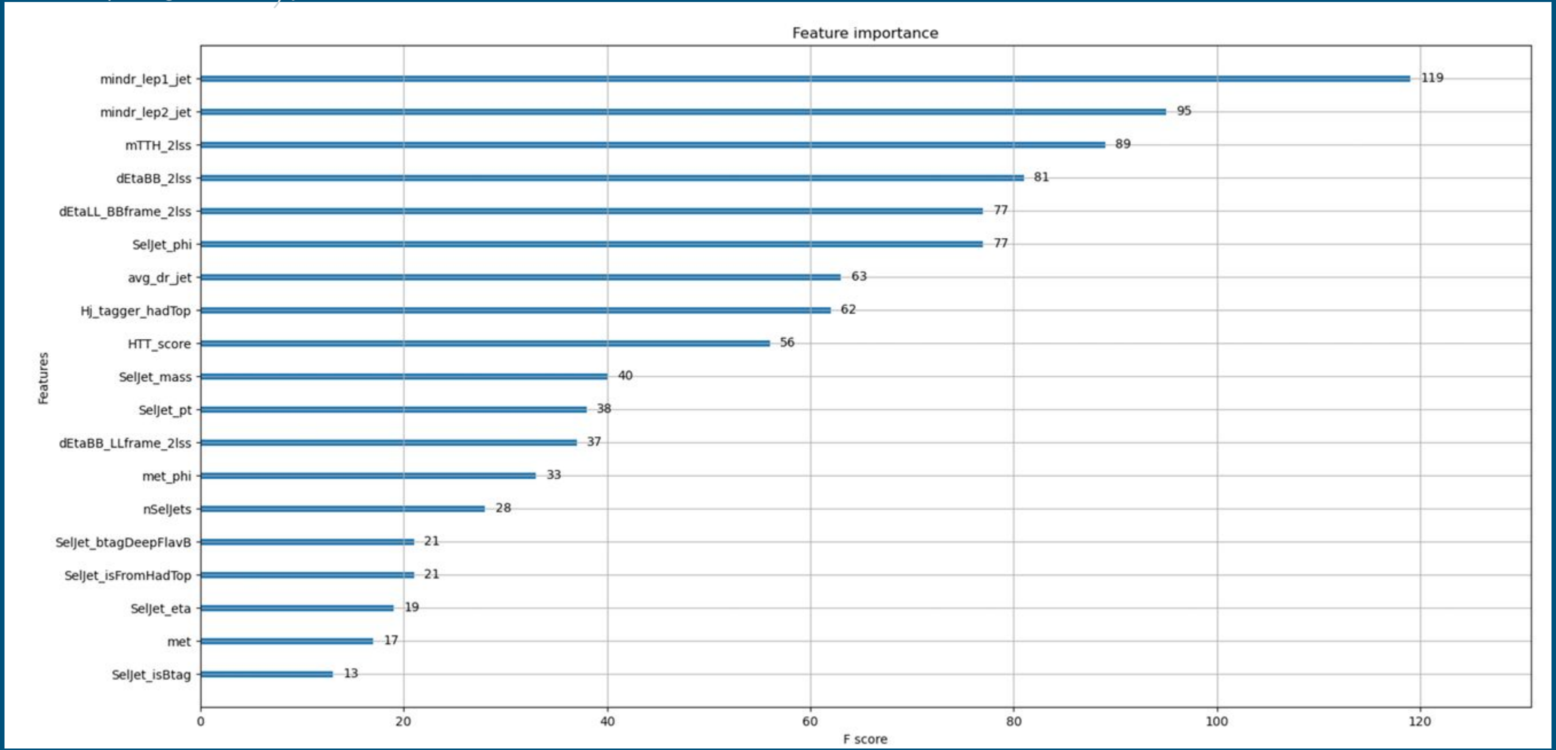
- expand this fit to the entire detector, not just the pixel detector
  - create a map of the density, instead of using only the global average
  - define an interpolation method to be run on GPU
  - calculate the term θ_0 based on the local density

- test this new version against its current form

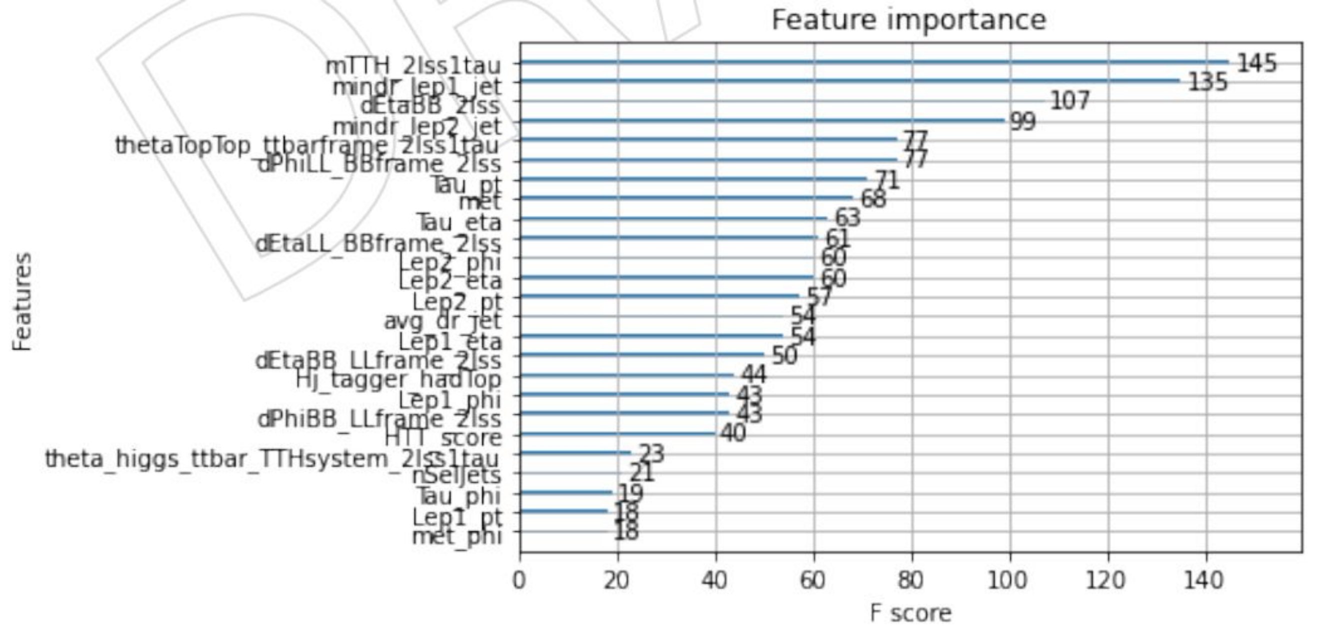
Thank you for the Attention!

The Input Variables: Variable ranking in Run 2 (2lss0tau)

- All features used for the 2lss0tau CP-BDT, with relative importance (from the CMS AN-20-241):



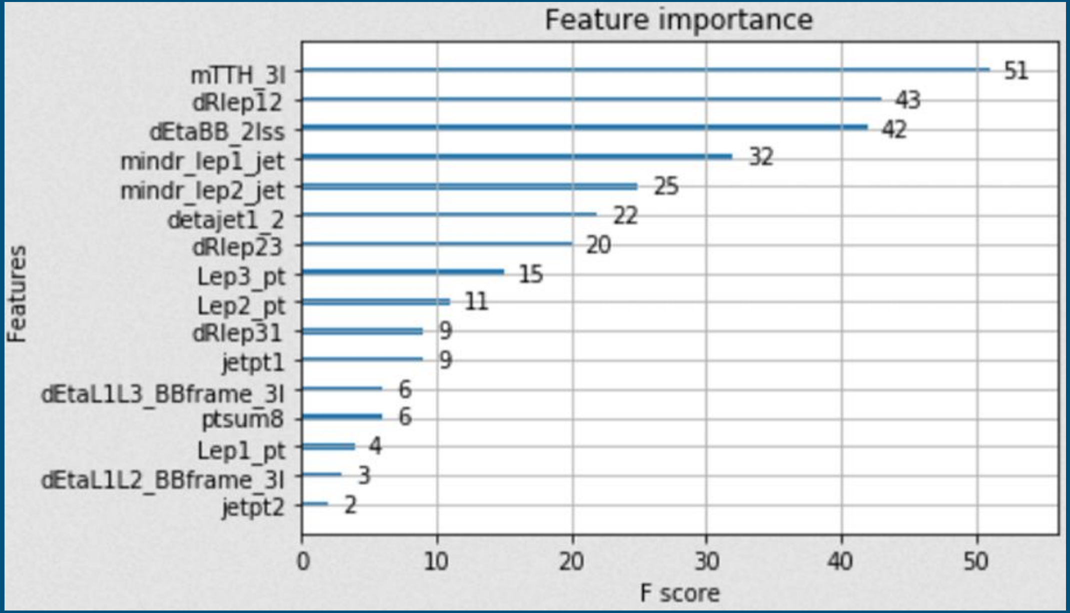
Features in Run 2



These are the input features of the CP-BDT in the Run2 AN. Of these, I have only avoided the HJet-tagger, and I have added the variables from here

The Input Variables: Variable ranking in Run 2 (3l0tau)

- All features used for the 3l0tau CP-BDT in Run2, with relative importance (from the CMS AN-20-241):



Training Method

The BDT is trained using XGBoost, with the following functions:

The evaluation metric is the AUC of the validation set

The number of rounds has been increased to 2000, but the best iteration is still picked

```
param = {
    "verbosity": 0,
    "objective": "binary:logistic",
    "eval_metric" : "auc",
    "tree_method": "hist",
    #"callbacks" : pruning_callback,
    #fixed parameters
    "n_estimators" : 2500,
    "reg_lambda" : 1.0,
    "reg_alpha" : 0.1,
    "min_child_weight" : 2.0,
    "scale_pos_weight" : 1.055,
    "random_state" : 723575,
    "early_stopping_rounds" : 2000,
    # Hyperparameters to exploit
    "learning_rate" : trial.suggest_float("learning_rate", 0.1, 4.0),
    "subsample": trial.suggest_float("subsample", 0.8, 1.0),
    "colsample_bytree": trial.suggest_float("colsample_bytree", 0.8, 1.0),
    "gamma" : trial.suggest_int("gamma", 0., 3.0),
    "max_depth" : trial.suggest_int("max_depth", 3, 6)
}
clf = xgb.XGBClassifier(**param)
clf.fit(Xtr, ytr, sample_weight=wtr,
    eval_set=[(Xva, yva)],
    sample_weight_eval_set=[(wva)], verbose=100)
```

Updated Output: Error bars

Since last time, error bars have been added to both the score distribution and the ROC curve.

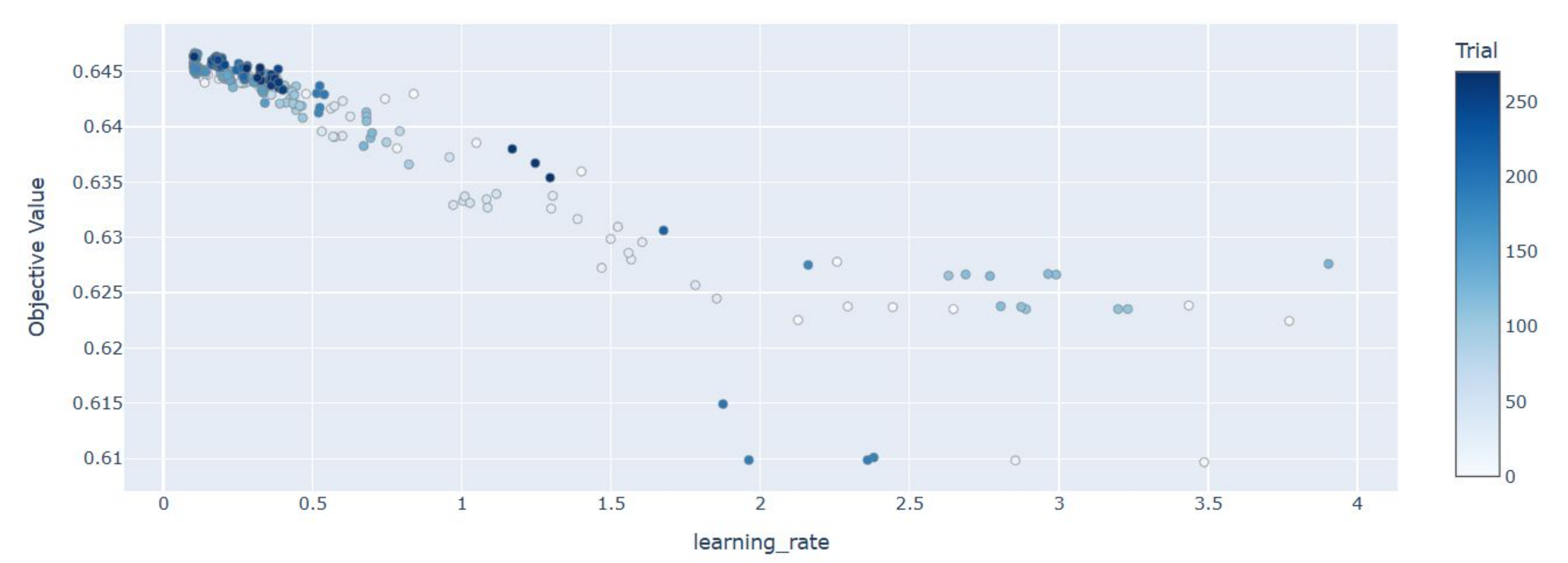
For the score distribution, each channel was assigned the following error:

$$\Delta n = \sqrt{n}$$

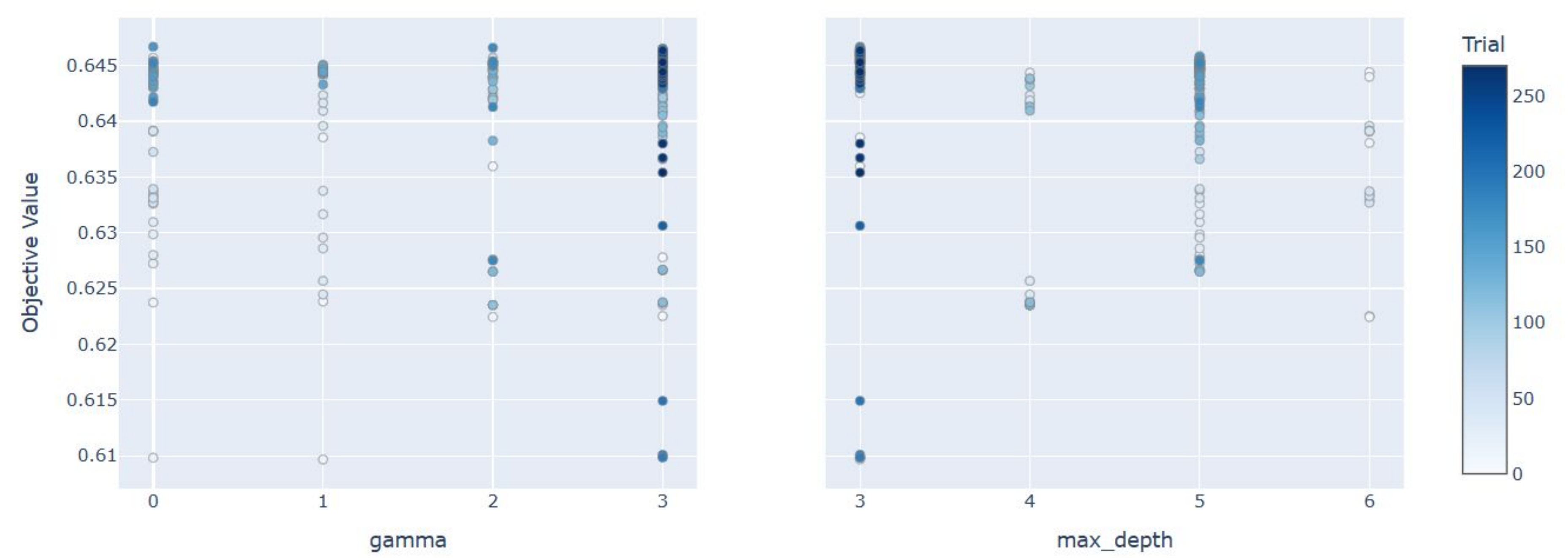
For the ROC curve, we assigned to the TPR (True Positive Rate) the following error:

$$\Delta tpr = \sqrt{\frac{tpr \cdot (1 - tpr)}{N}}$$

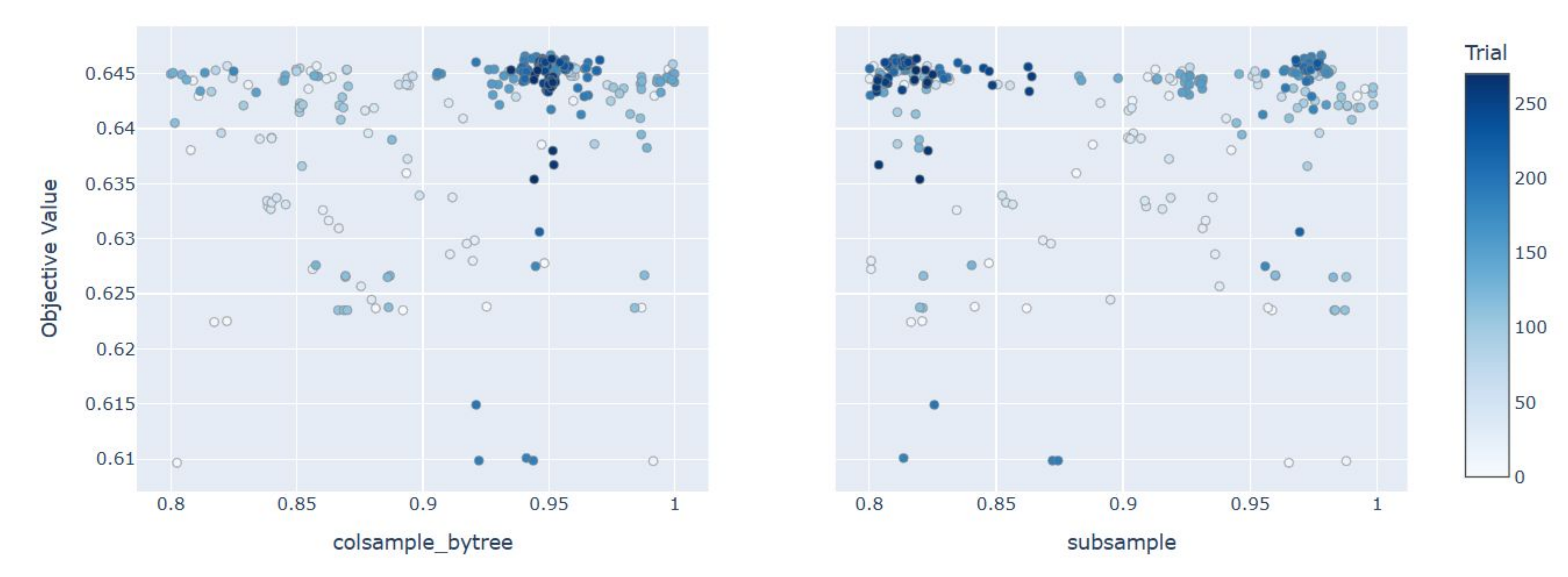
2lss0tau optimization plots



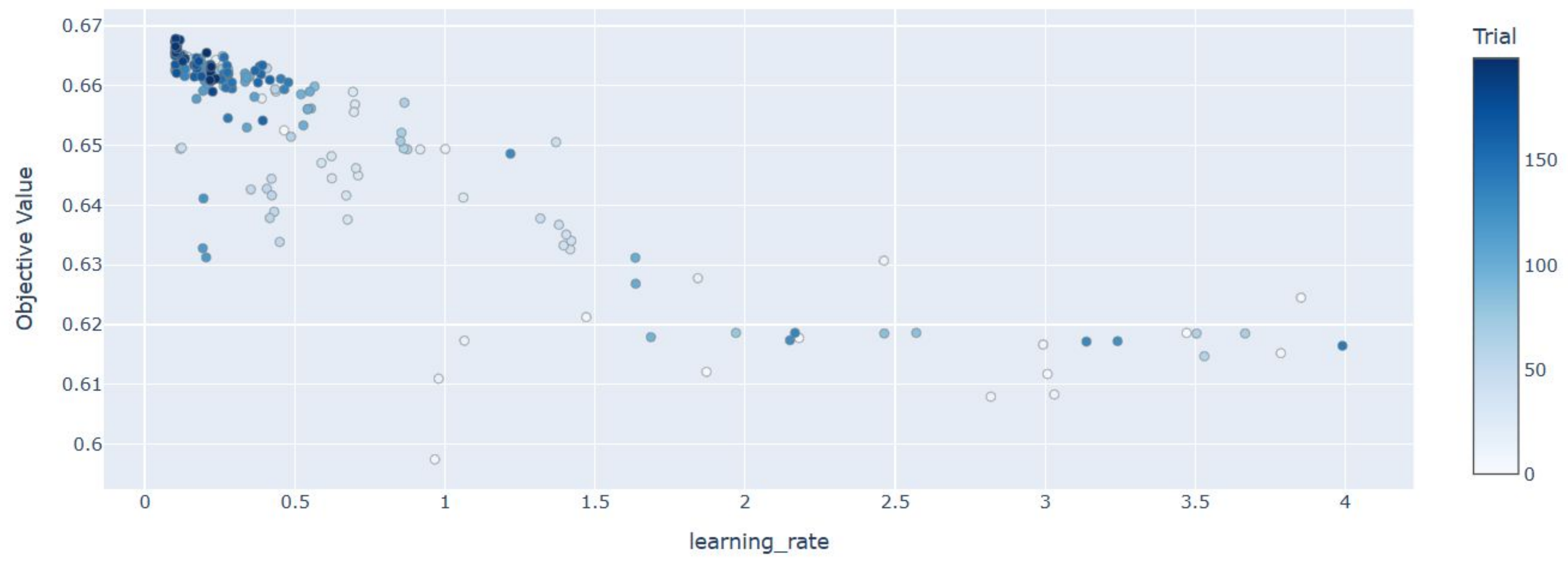
2lss0tau optimization plots



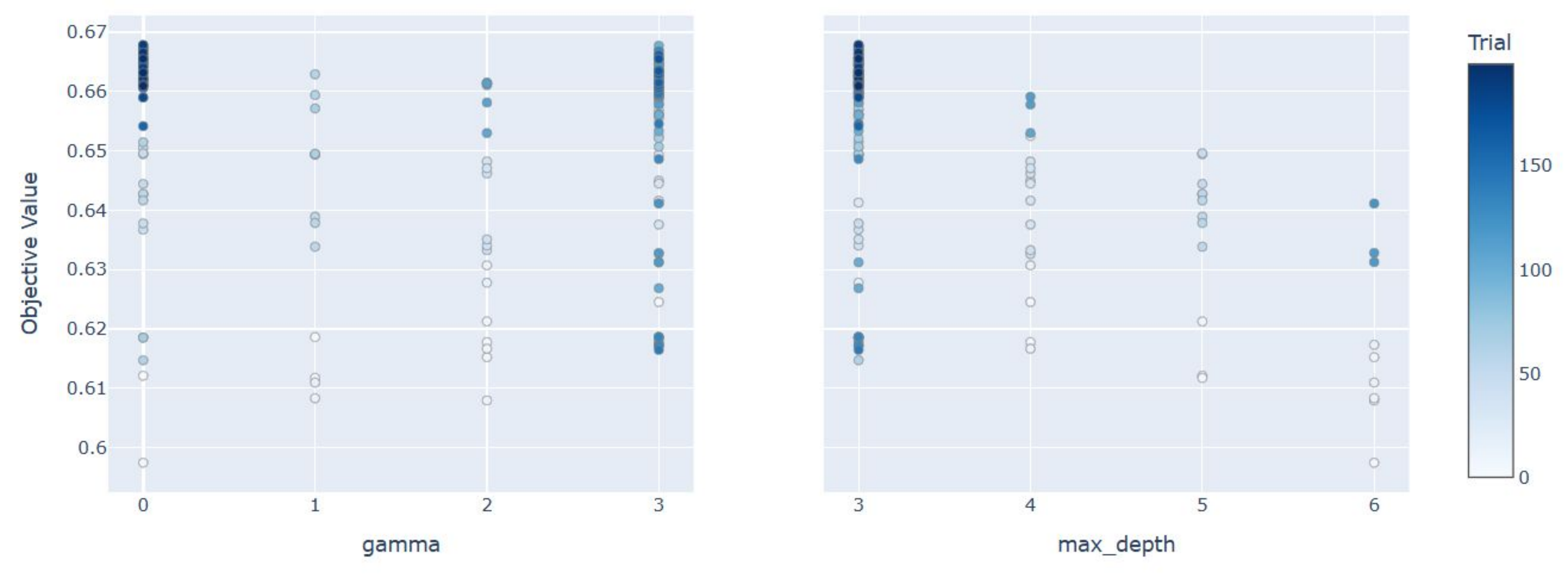
2lss0tau optimization plots



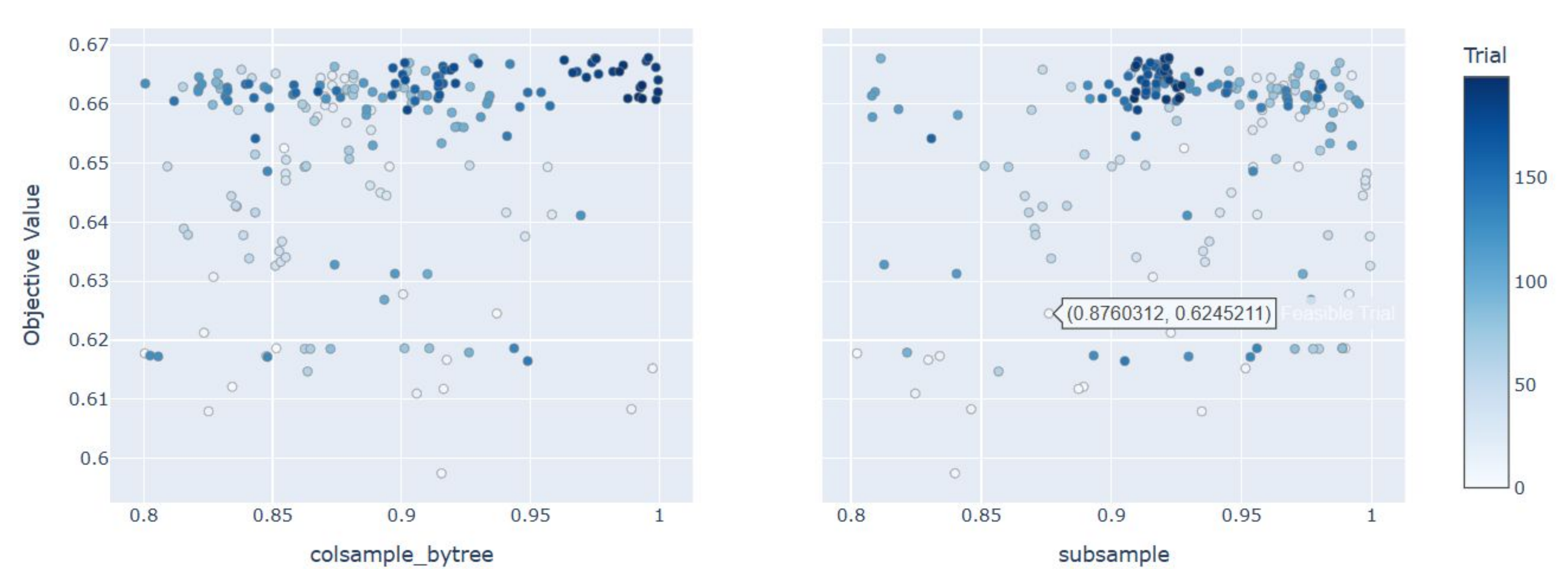
3l0tau optimization plots



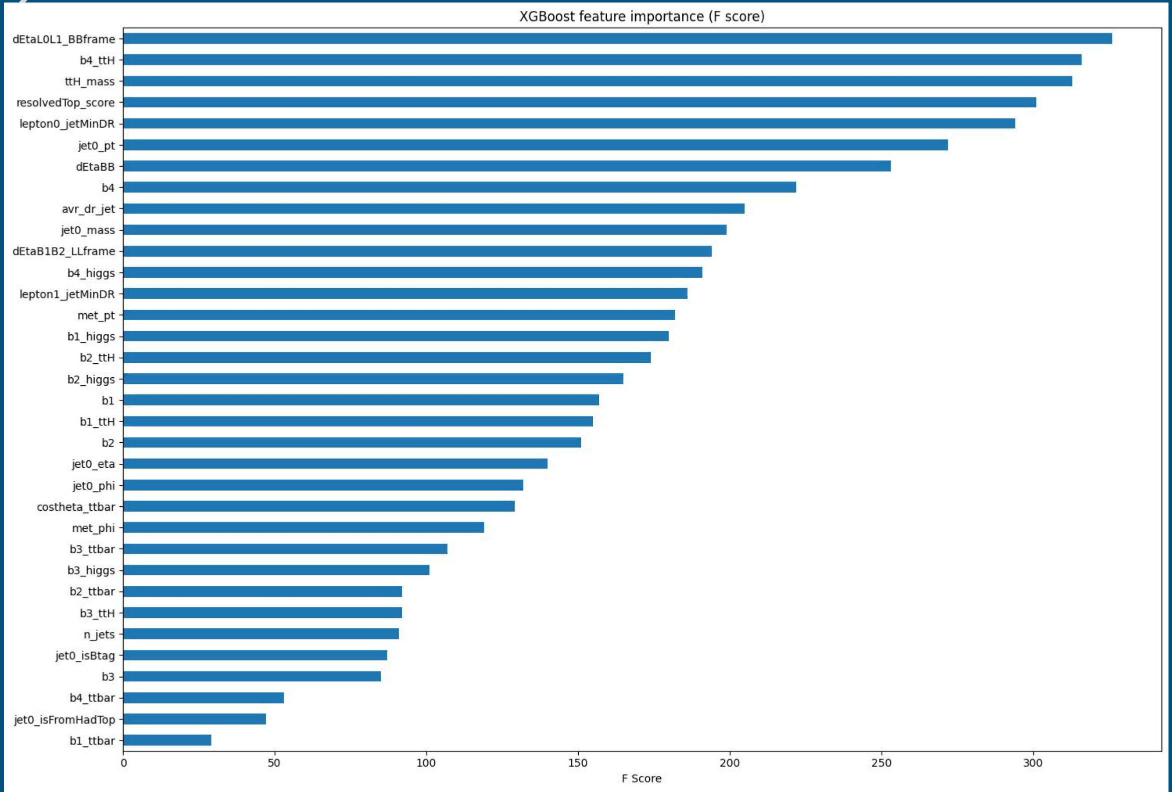
3l0tau optimization plots



3l0tau optimization plots



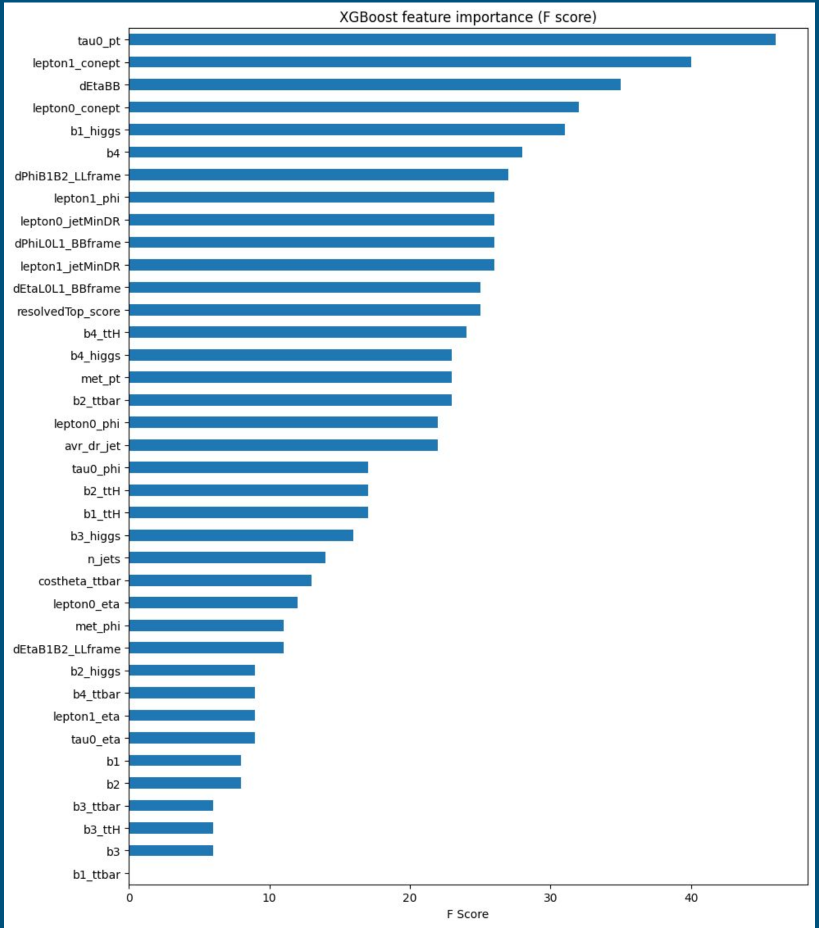
The Input Variables: Variable ranking in Run 3 (2lss0tau)



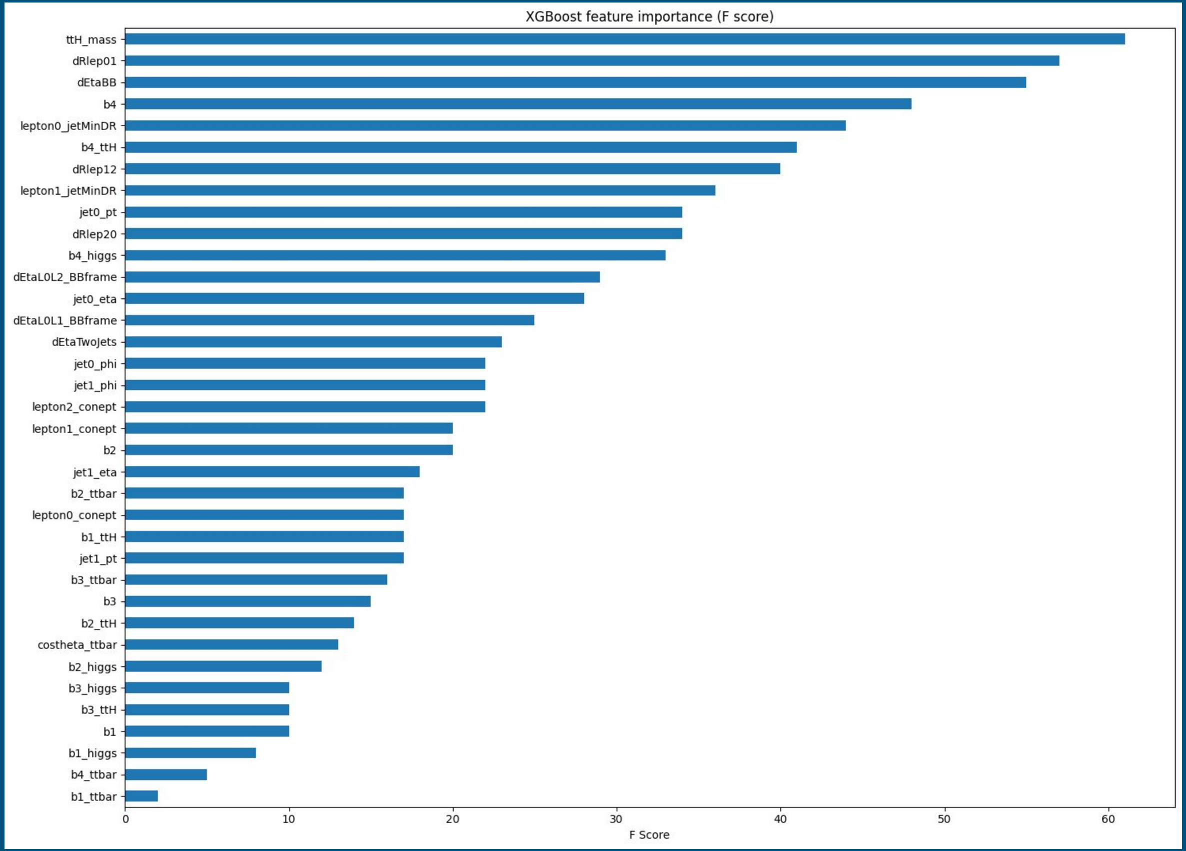
All features used for the 2lss0tau CP-BDT, with relative importance.

The Input Variables: Variable ranking in Run 3 (2lss1tau)

All features used for the 2lss0tau CP-BDT, with relative importance.



The Input Variables: Variable ranking in Run 3 (3l0tau)



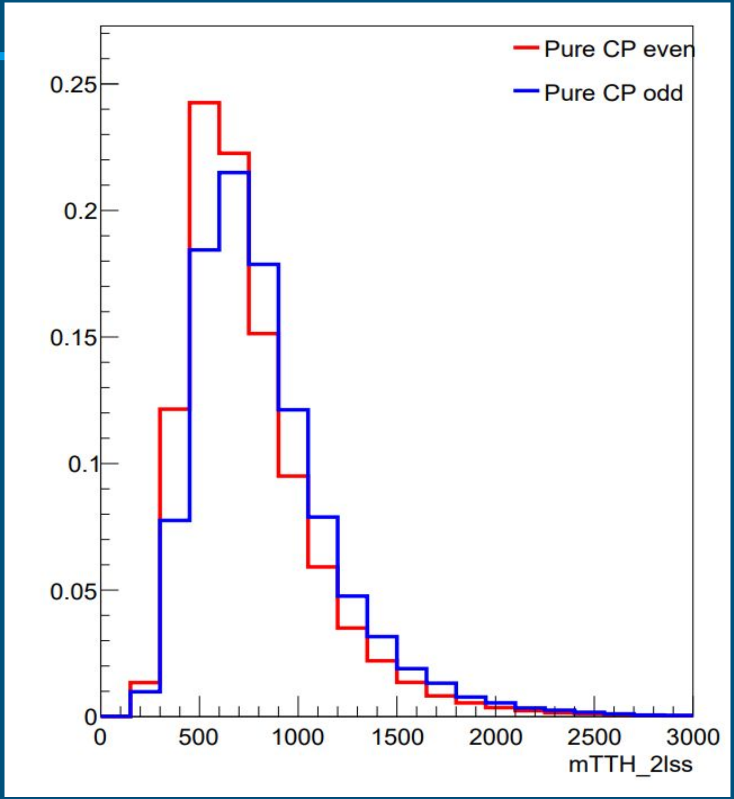
Features used for the 3l0tau BDT, with relative importance.

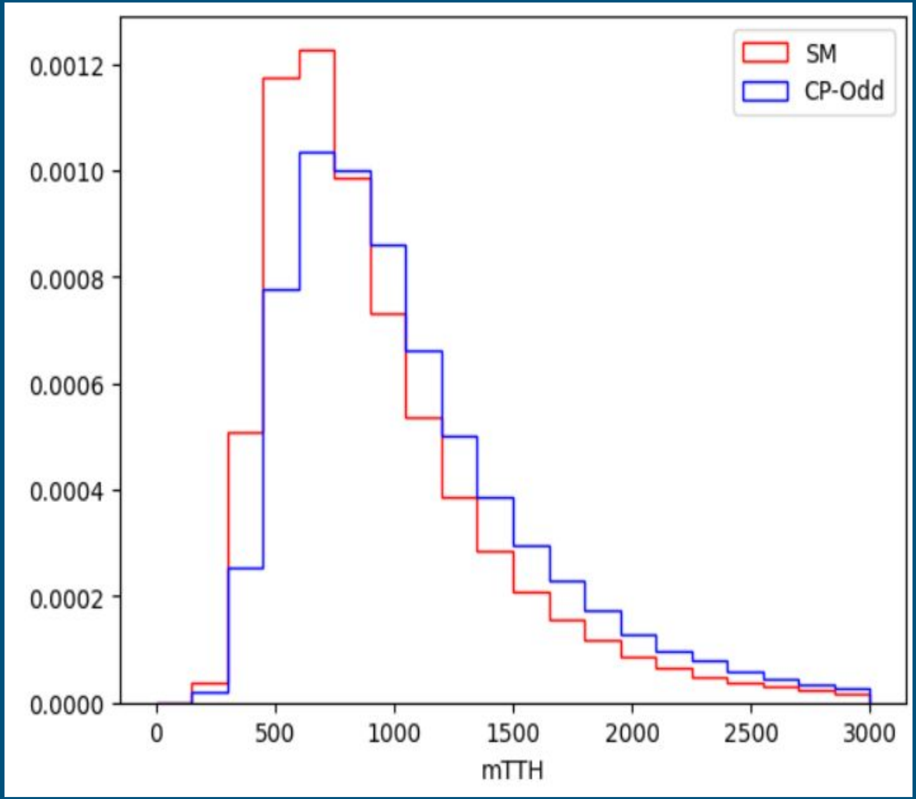
Variables from arXiv:2406.03950v2

observable	definition	frame
$p_{T,H}$	-	lab, $t\bar{t}$, $t\bar{t}H$
$\Delta\eta_{t\bar{t}}$	$\lvert\eta_t - \eta_{\bar{t}}\rvert$	lab, H, $t\bar{t}H$
$\Delta\phi_{t\bar{t}}$	$\lvert\phi_t - \phi_{\bar{t}}\rvert$	lab, H, $t\bar{t}H$
$m_{t\bar{t}}$	$(p_t + p_{\bar{t}})^2$	frame-invariant
$m_{t\bar{t}H}$	$(p_t + p_{\bar{t}} + p_H)^2$	frame-invariant
$\lvert\cos\theta^*\rvert$	$\frac{\lvert\boldsymbol{p}_t \cdot \boldsymbol{n}\rvert}{\lvert\boldsymbol{p}_t\rvert \cdot \lvert\boldsymbol{n}\rvert}$	$t\bar{t}$
b_1	$\frac{(\boldsymbol{p}_t \times \boldsymbol{n}) \cdot (\boldsymbol{p}_{\bar{t}} \times \boldsymbol{n})}{p_{T,t}\, p_{T,\bar{t}}}$	all
b_2	$\frac{(\boldsymbol{p}_t \times \boldsymbol{n}) \cdot (\boldsymbol{p}_{\bar{t}} \times \boldsymbol{n})}{\lvert\boldsymbol{p}_t\rvert\ \lvert\boldsymbol{p}_{\bar{t}}\rvert}$	all
b_3	$\frac{p_t^x\ p_{\bar{t}}^x}{p_{T,t}\, p_{T,\bar{t}}}$	all
b_4	$\frac{p_t^z\ p_{\bar{t}}^z}{\lvert\boldsymbol{p}_t\rvert\ \lvert\boldsymbol{p}_{\bar{t}}\rvert}$	all
ϕ_C	$\arccos\left(\frac{\lvert(\boldsymbol{p}_{p_1} \times \boldsymbol{p}_{p_2}) \cdot (\boldsymbol{p}_t \times \boldsymbol{p}_{\bar{t}})\rvert}{\lvert\boldsymbol{p}_{p_1} \times \boldsymbol{p}_{p_2}\rvert\ \lvert\boldsymbol{p}_t \times \boldsymbol{p}_{\bar{t}}\rvert}\right)$	H

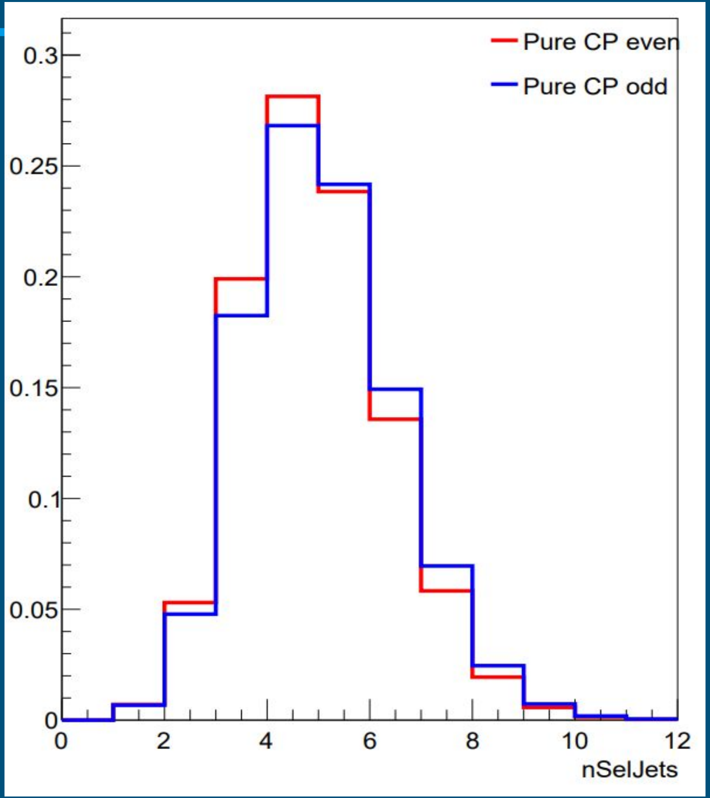
Except this one

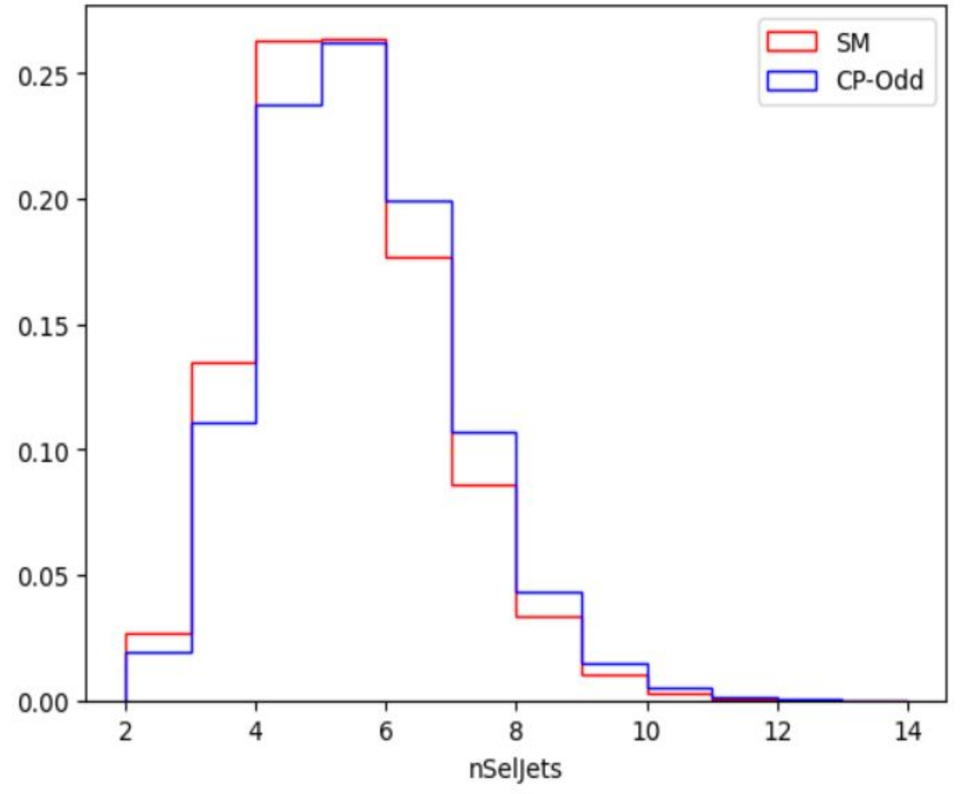
2lss0tau Input variables comparison: Run2 vs Run3



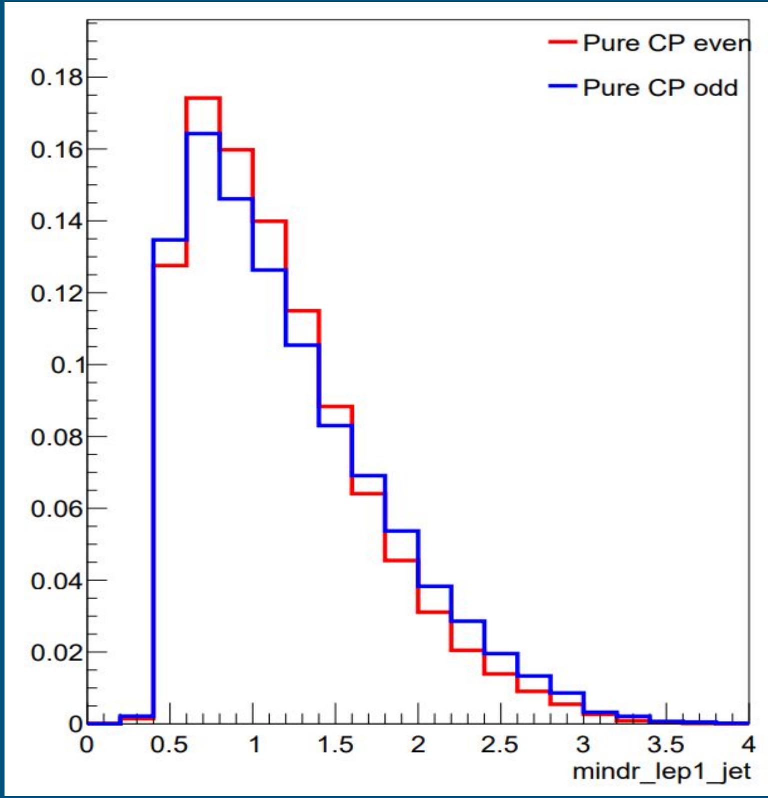


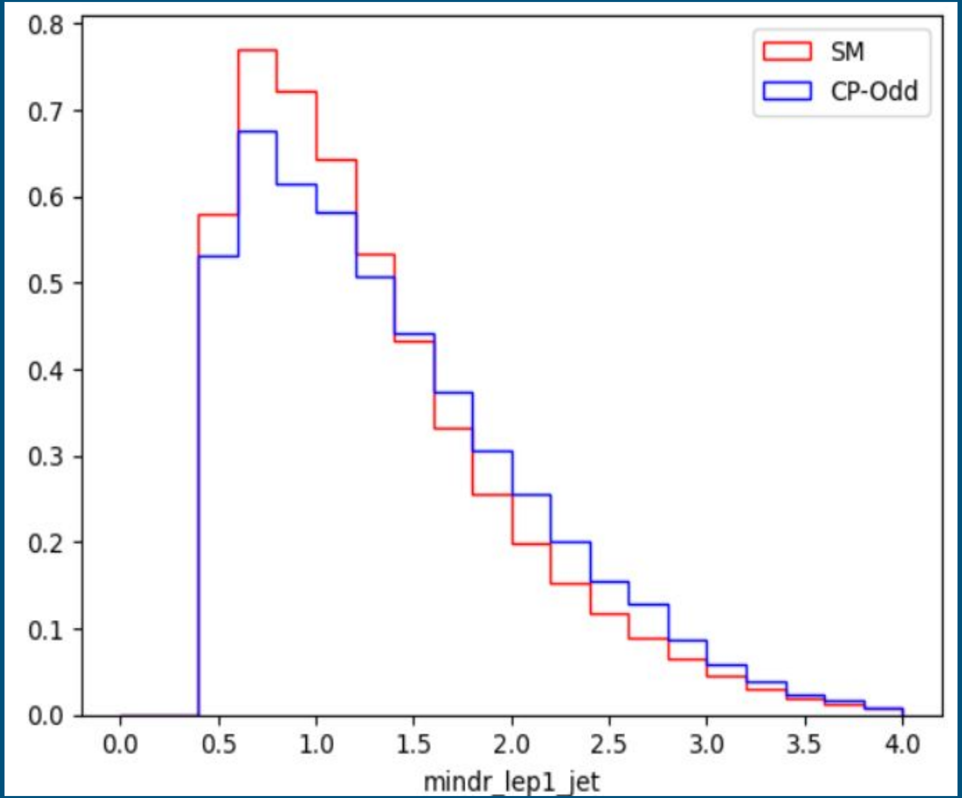
2lss0tau Input variables comparison: Run2 vs Run3



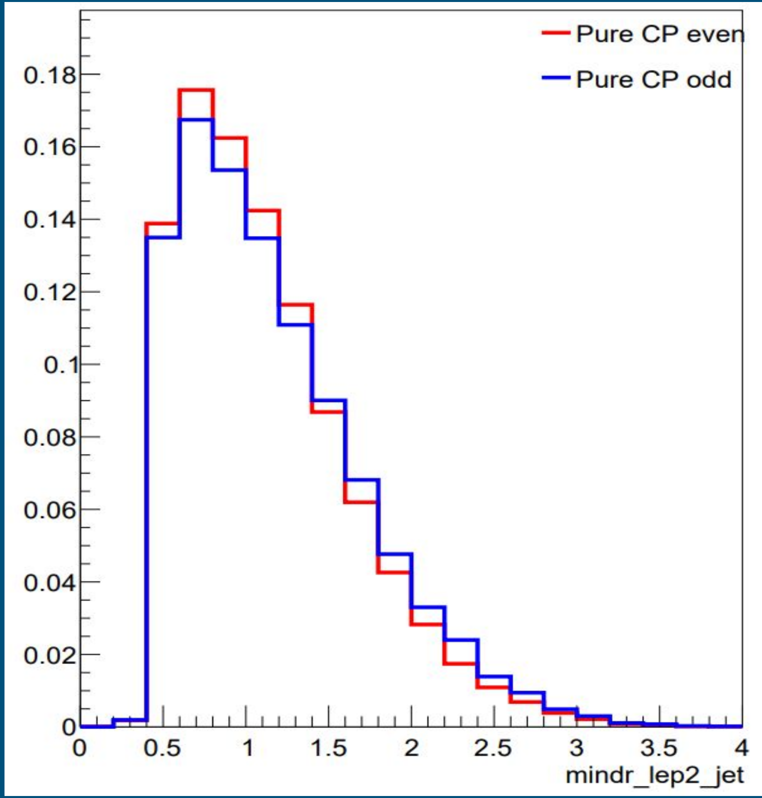


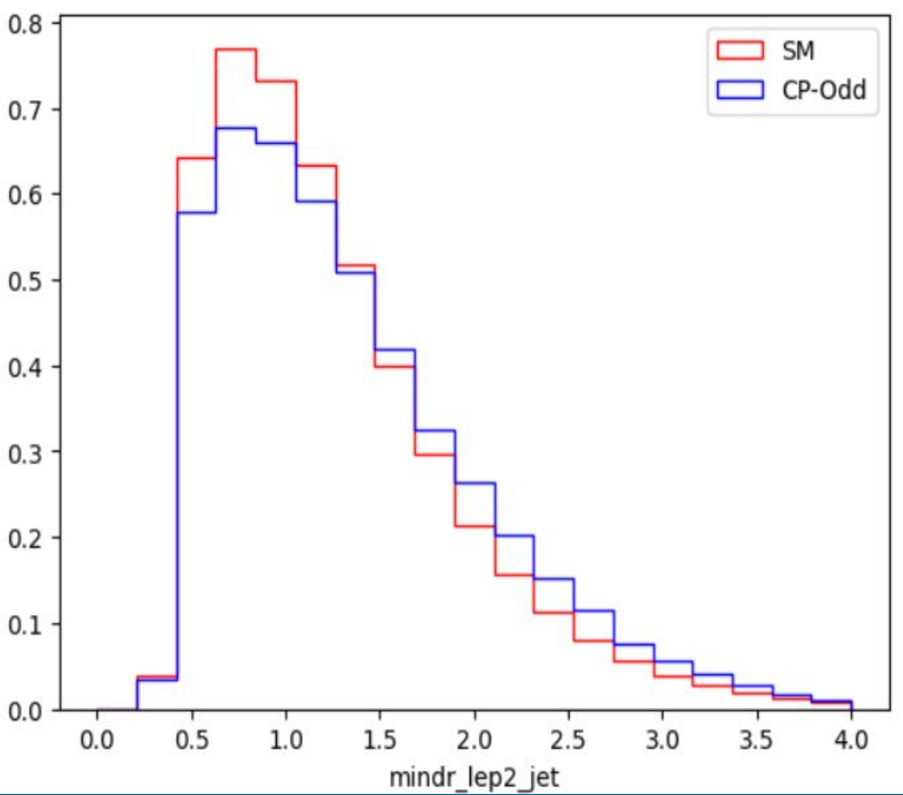
2lss0tau Input variables comparison: Run2 vs Run3



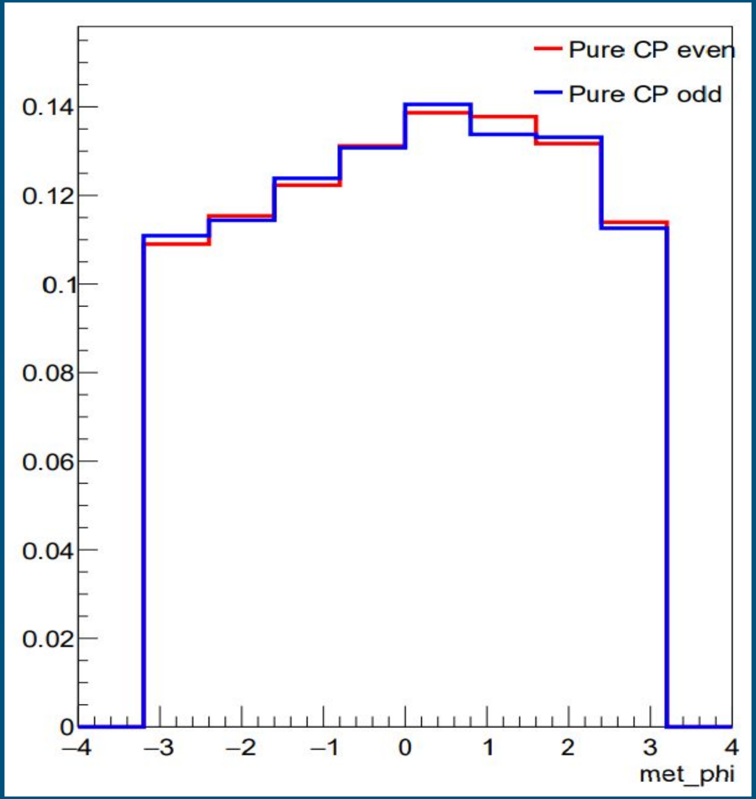


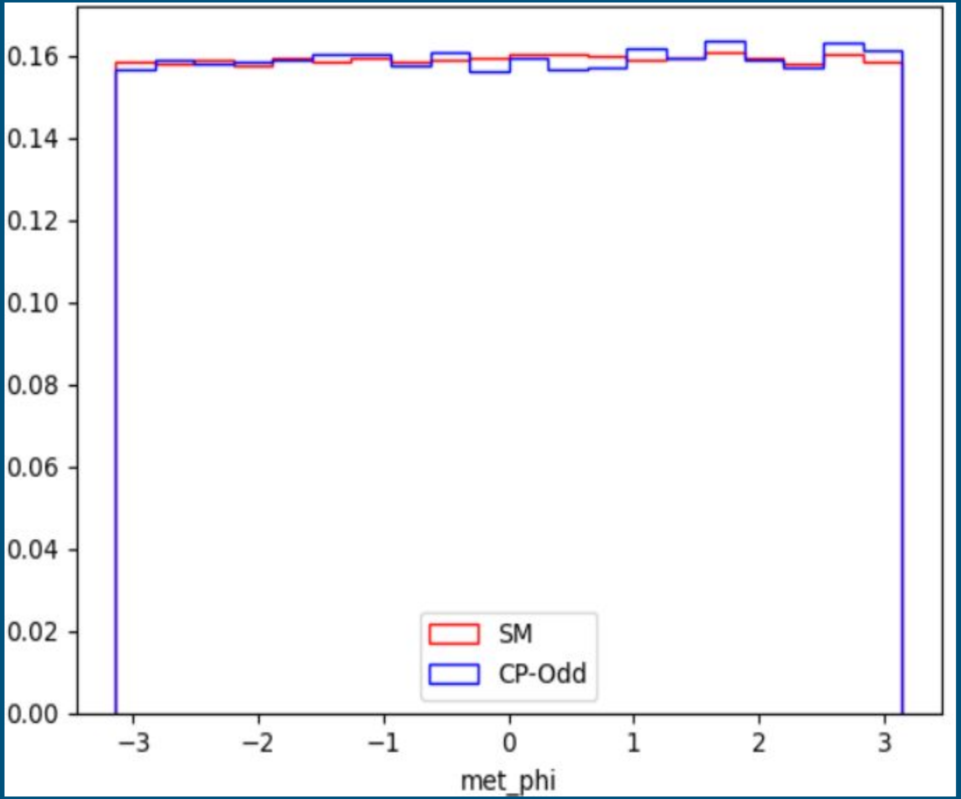
2lss0tau Input variables comparison: Run2 vs Run3



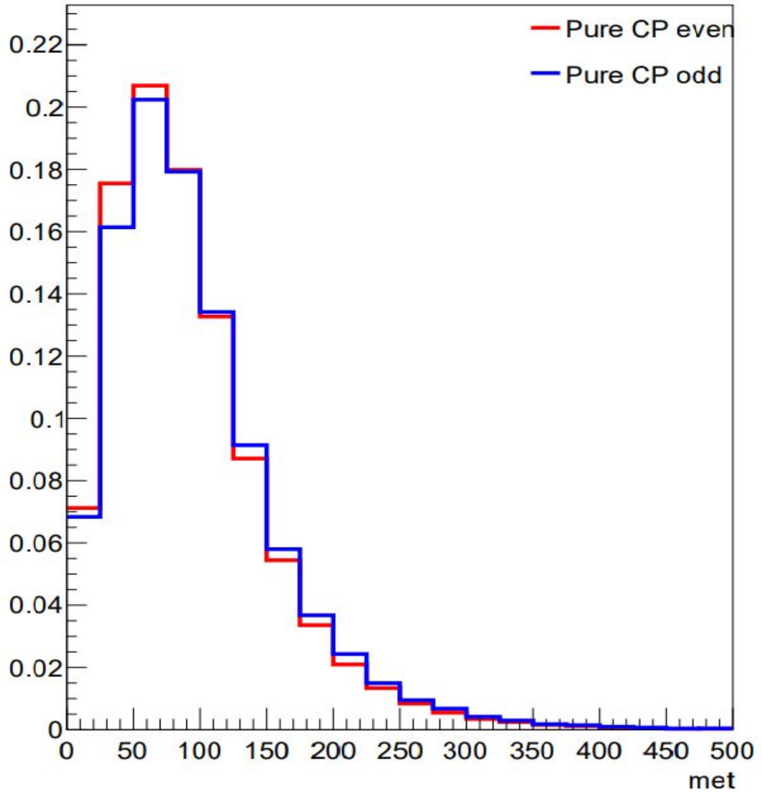


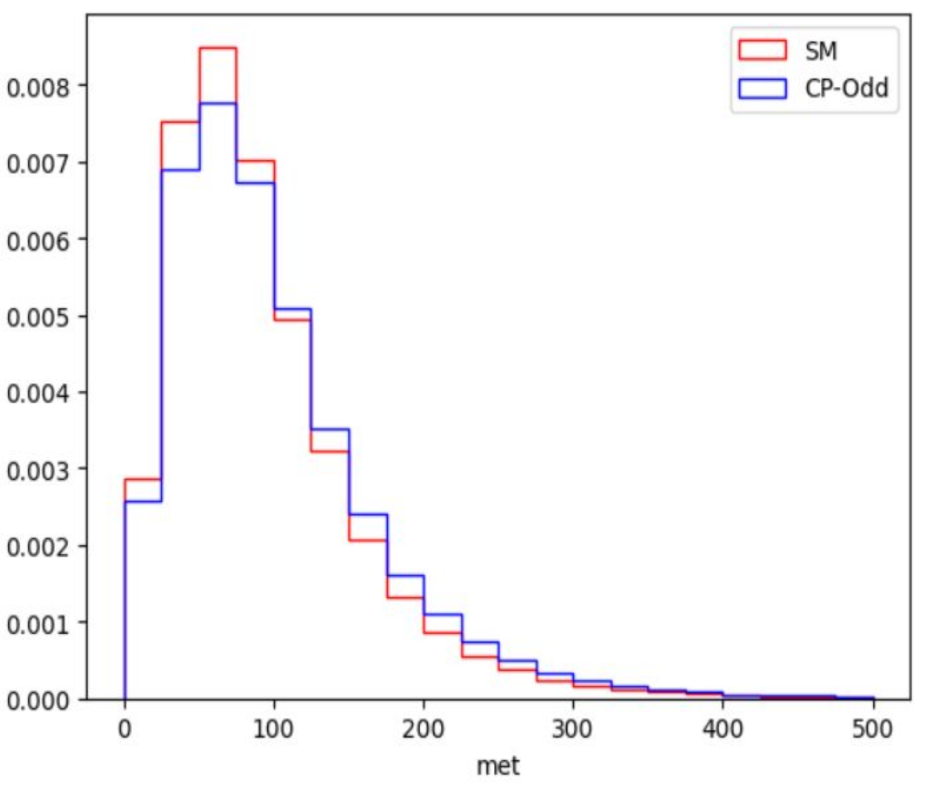
2lss0tau Input variables comparison: Run2 vs Run3



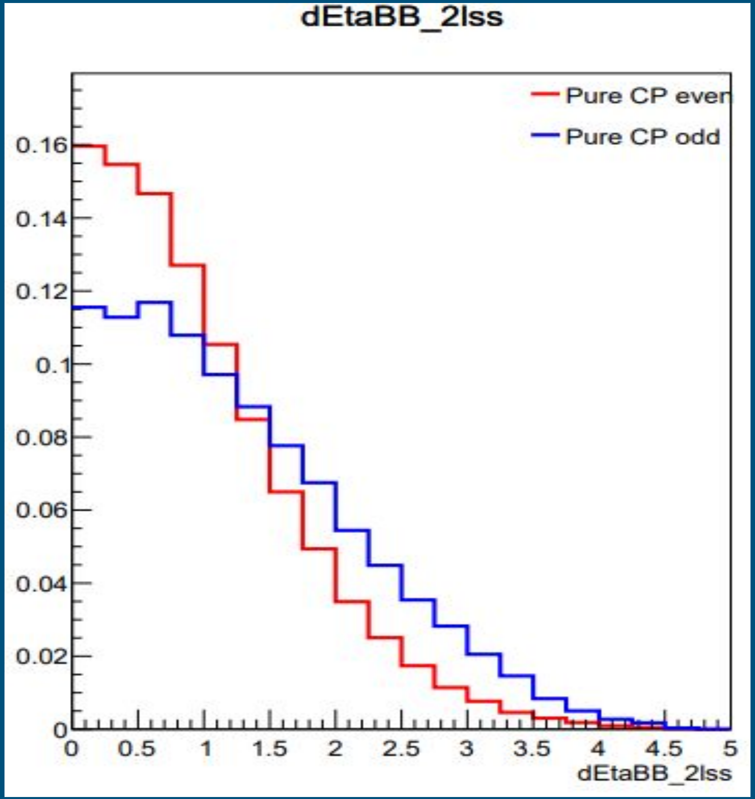


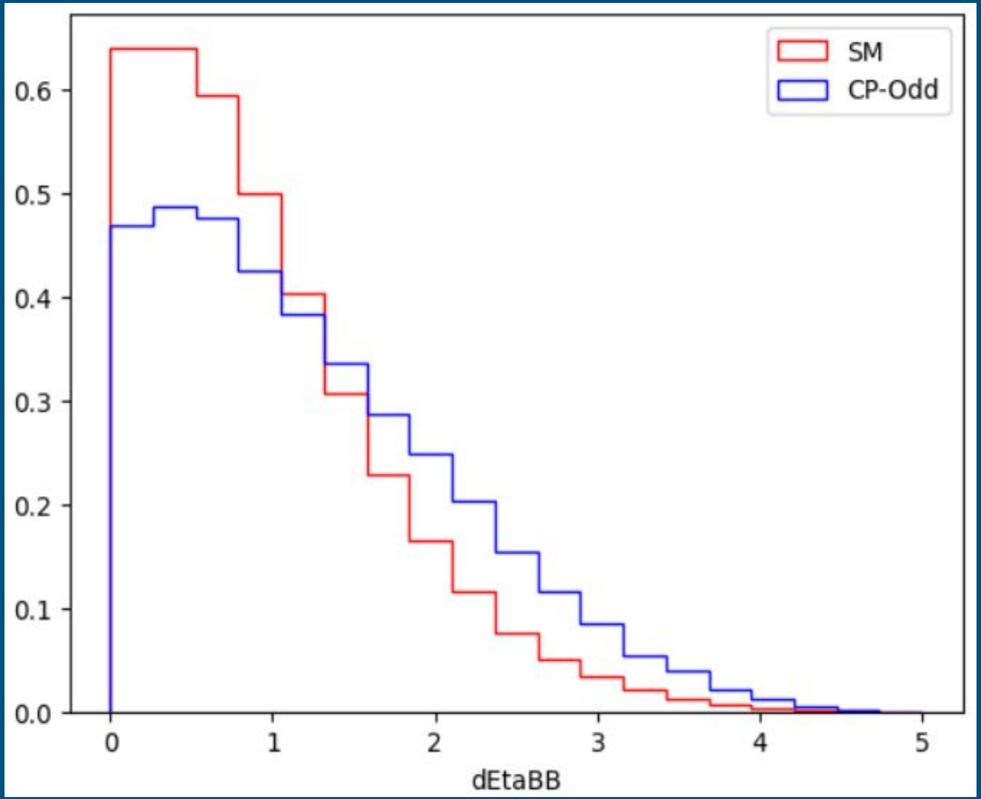
2lss0tau Input variables comparison: Run2 vs Run3



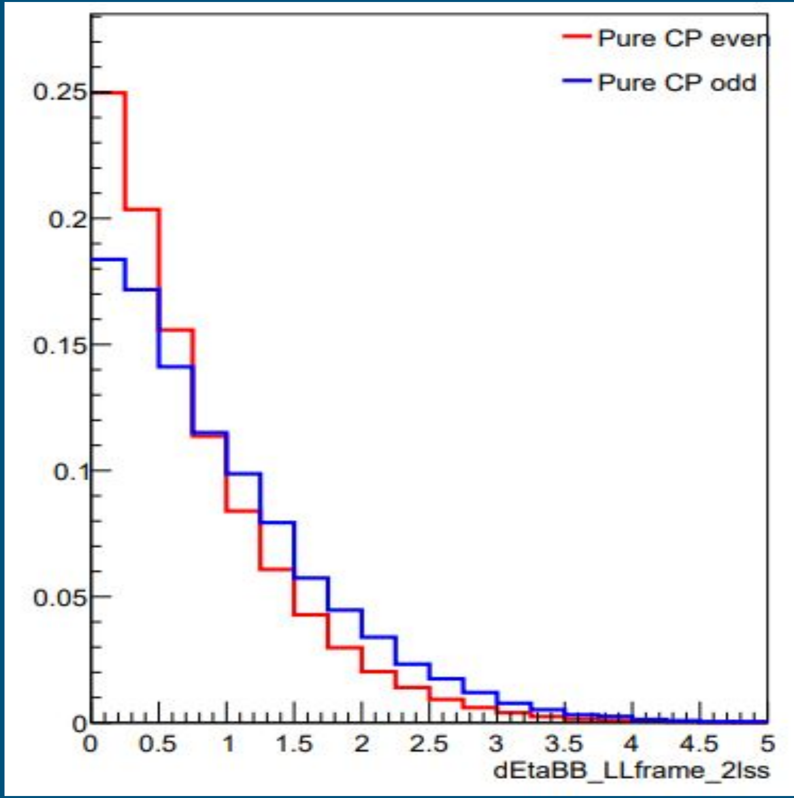


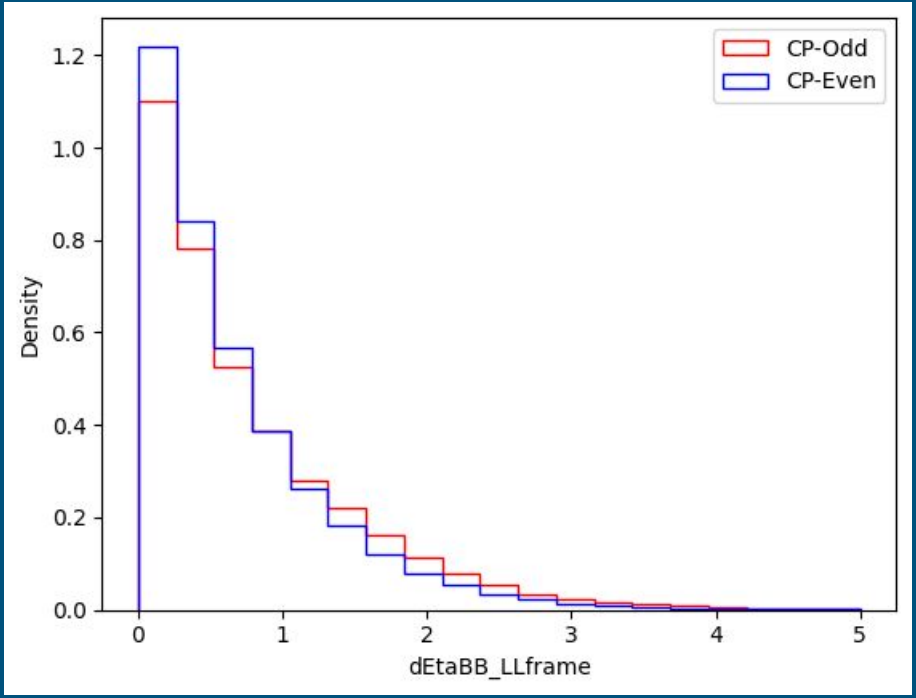
2lss0tau Input variables comparison: Run2 vs Run3



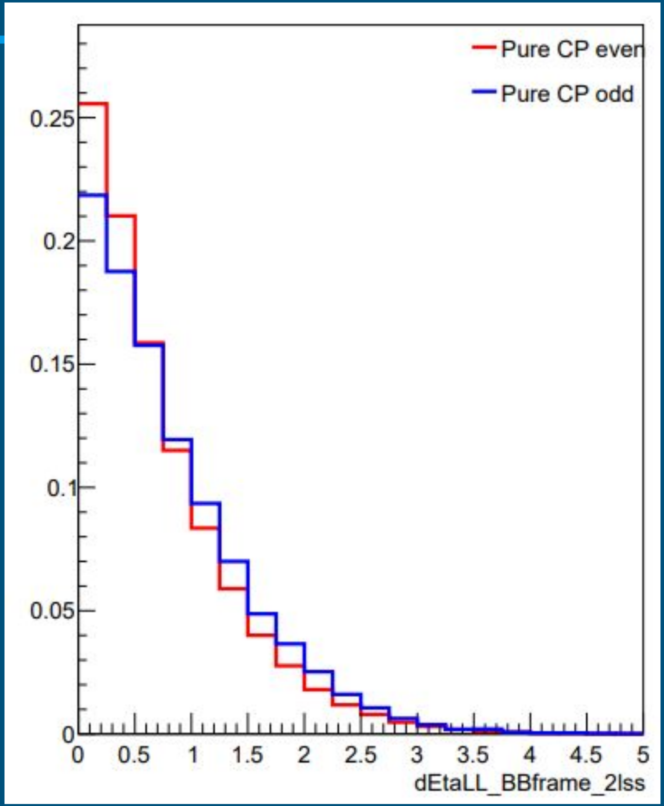


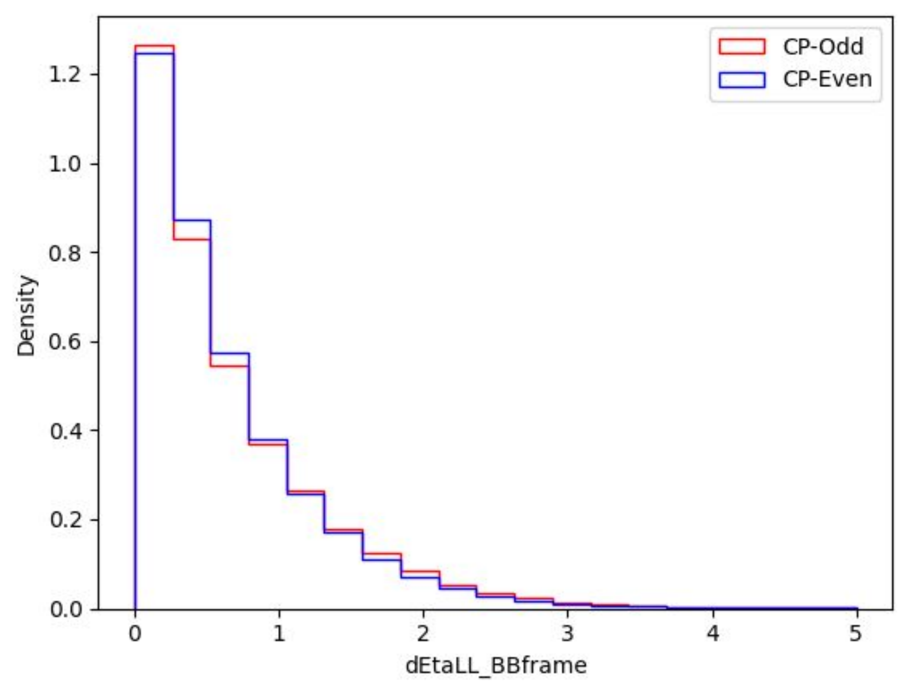
2lss0tau Input variables comparison: Run2 vs Run3



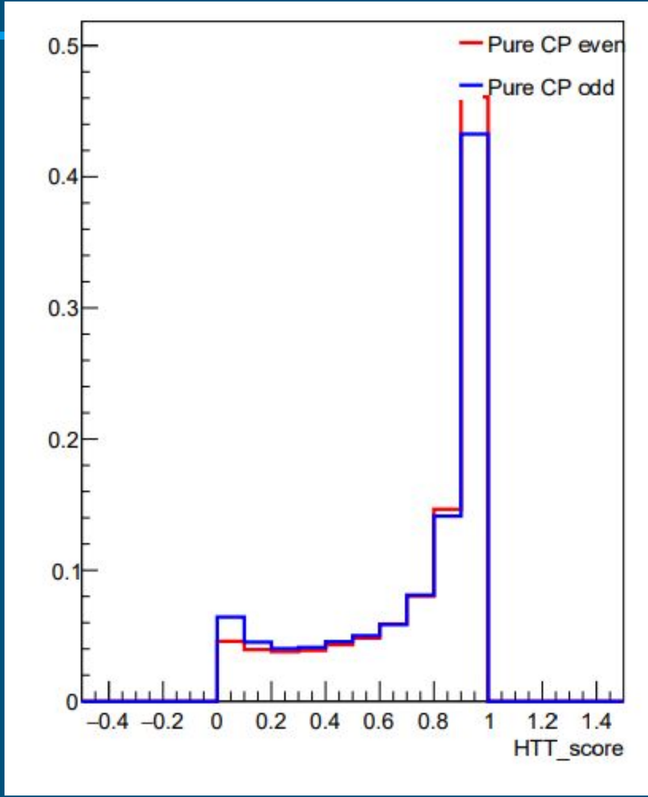


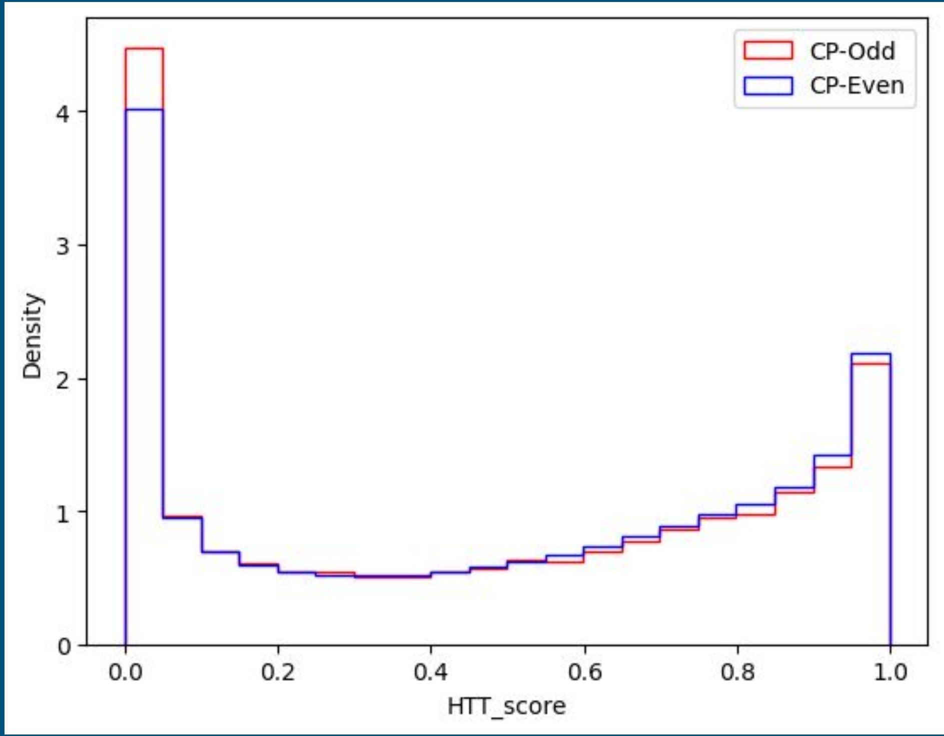
2lss0tau Input variables comparison: Run2 vs Run3



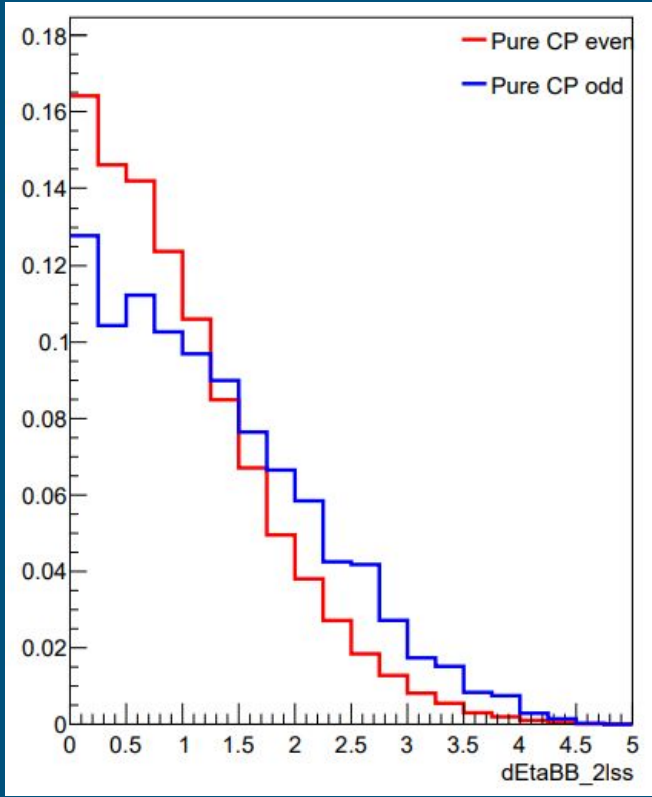


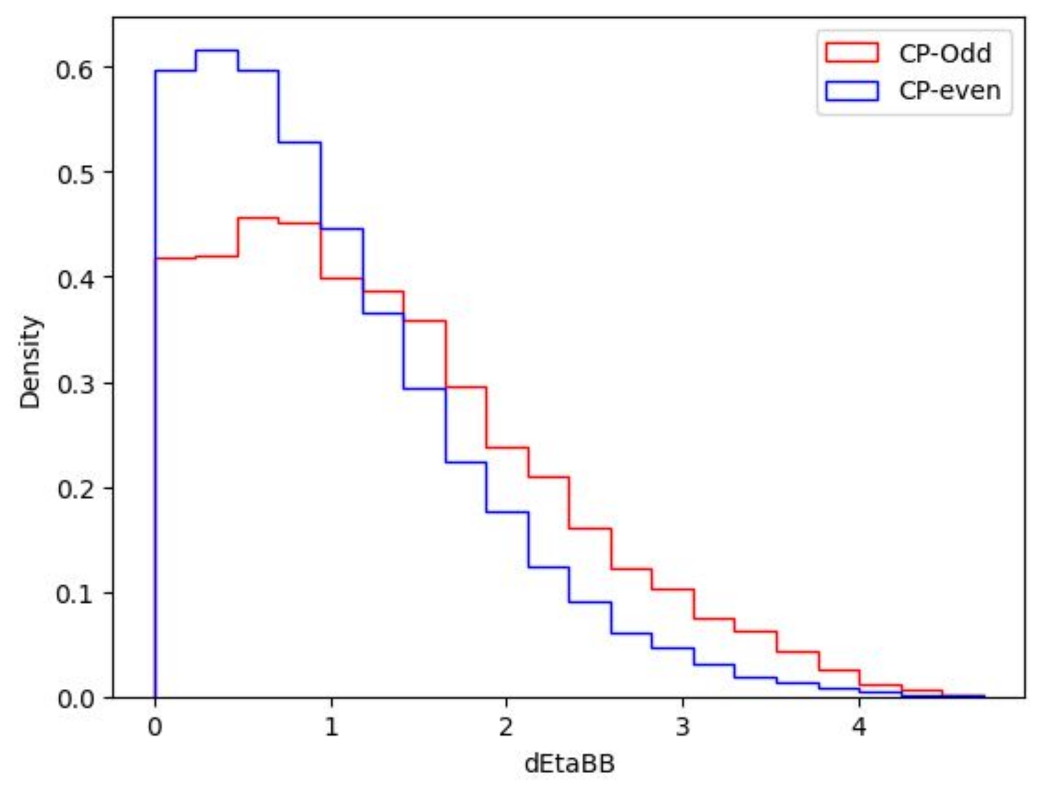
2lss0tau Input variables comparison: Run2 vs Run3



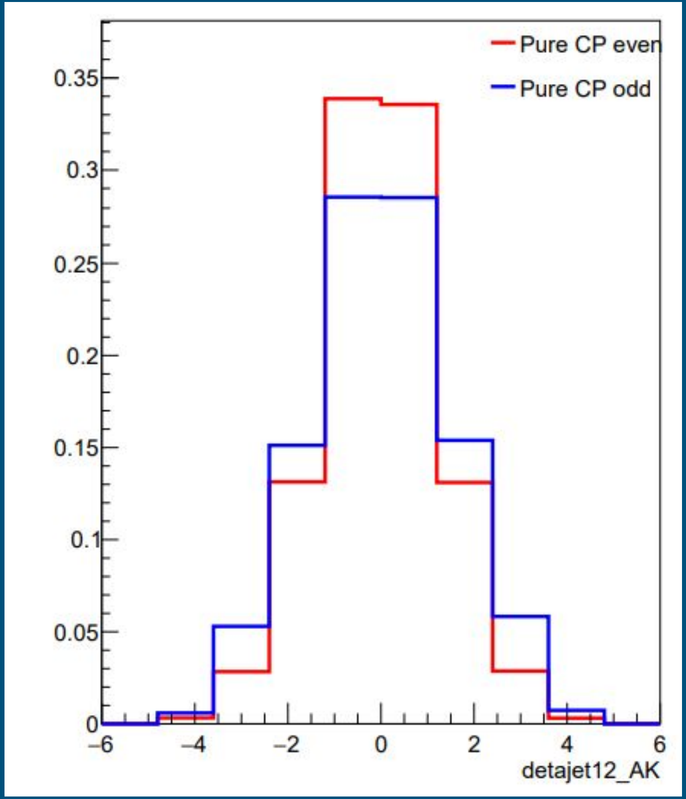


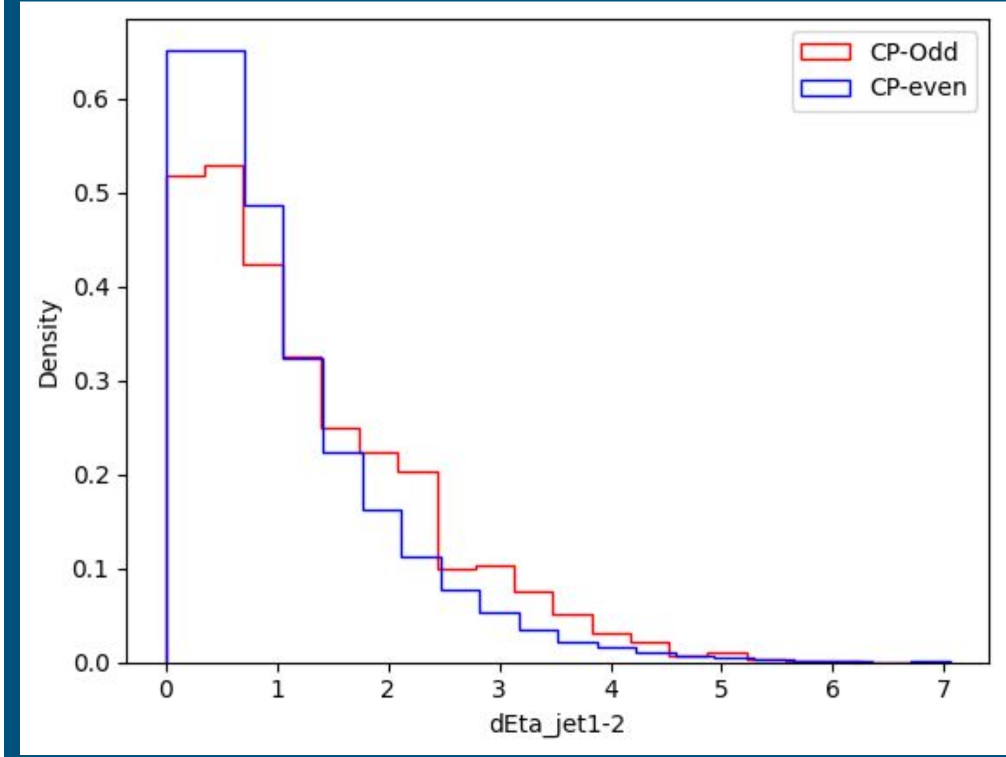
3l0tau Input variables comparison: Run2 vs Run3



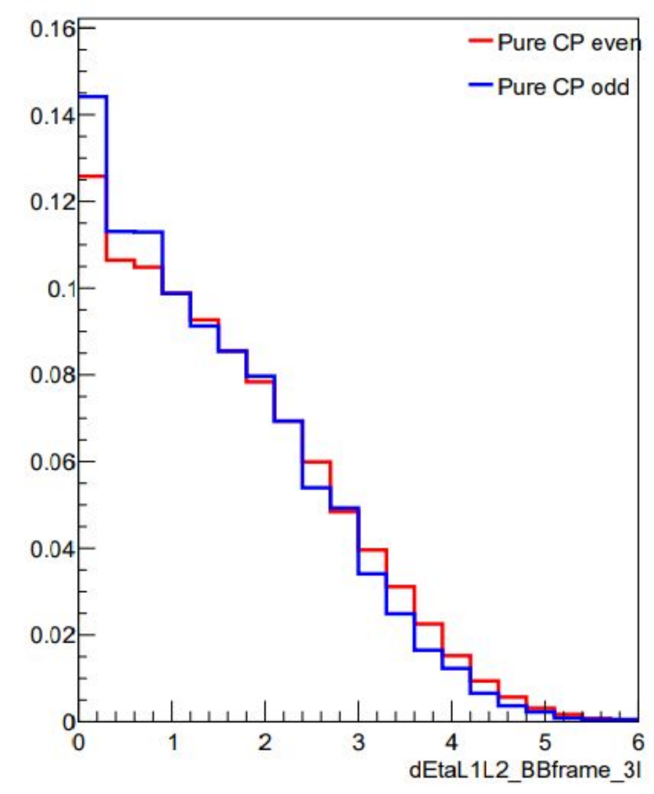


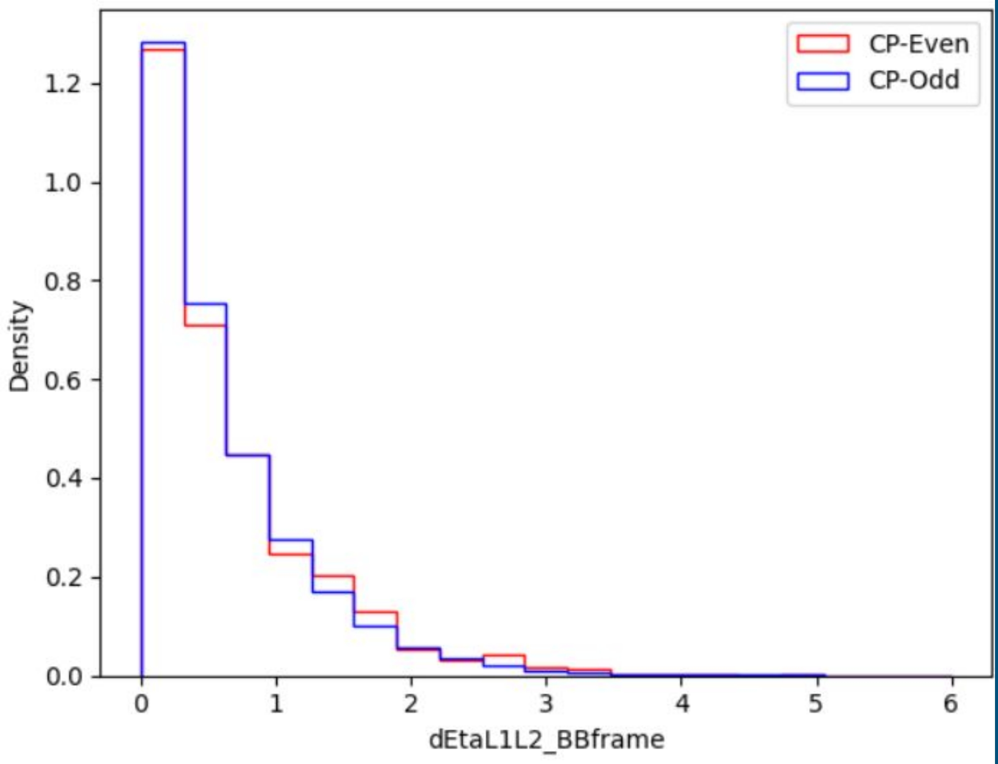
3l0tau Input variables comparison: Run2 vs Run3



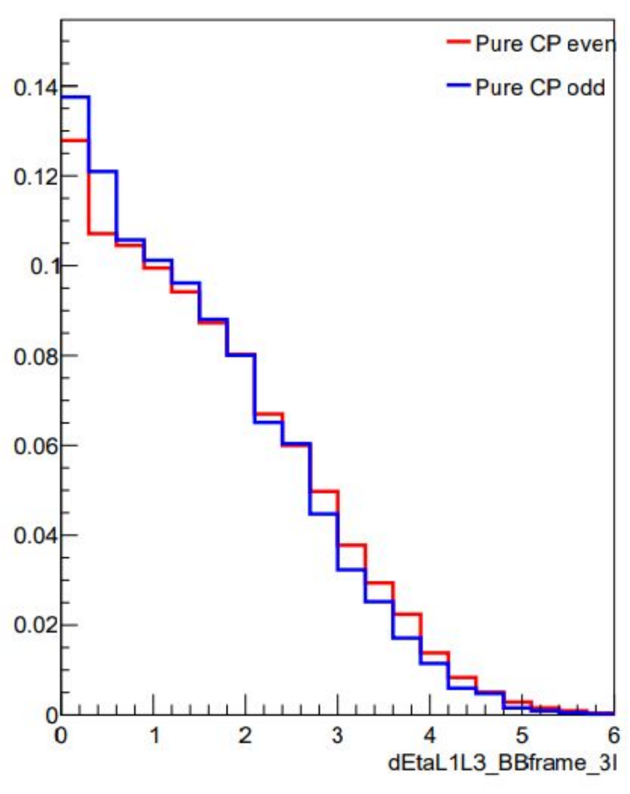


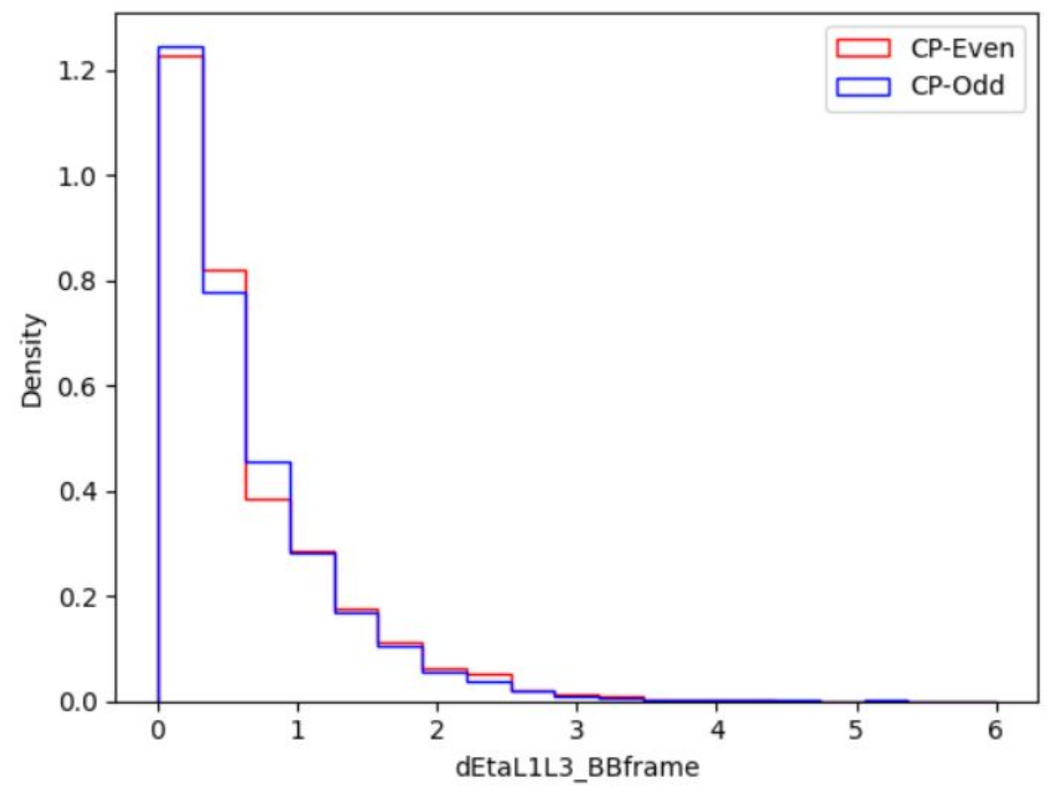
3l0tau Input variables comparison: Run2 vs Run3



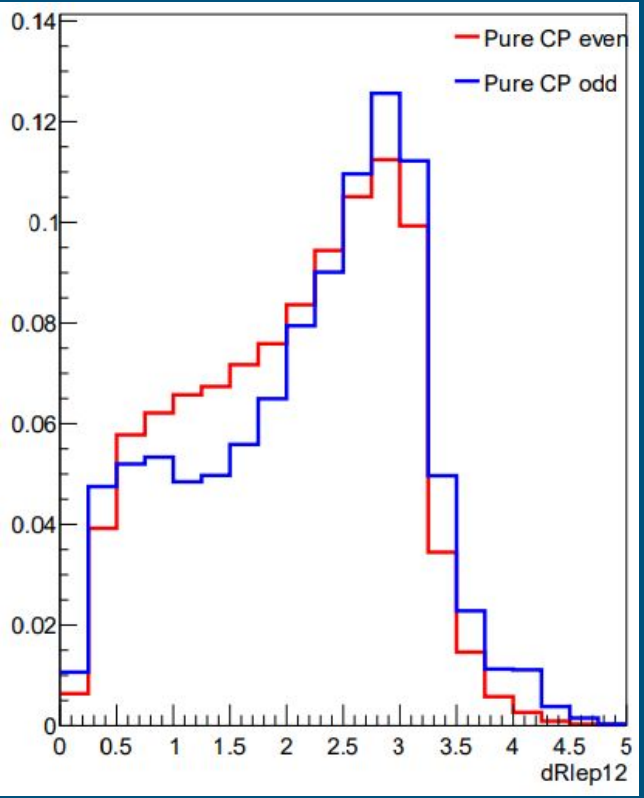


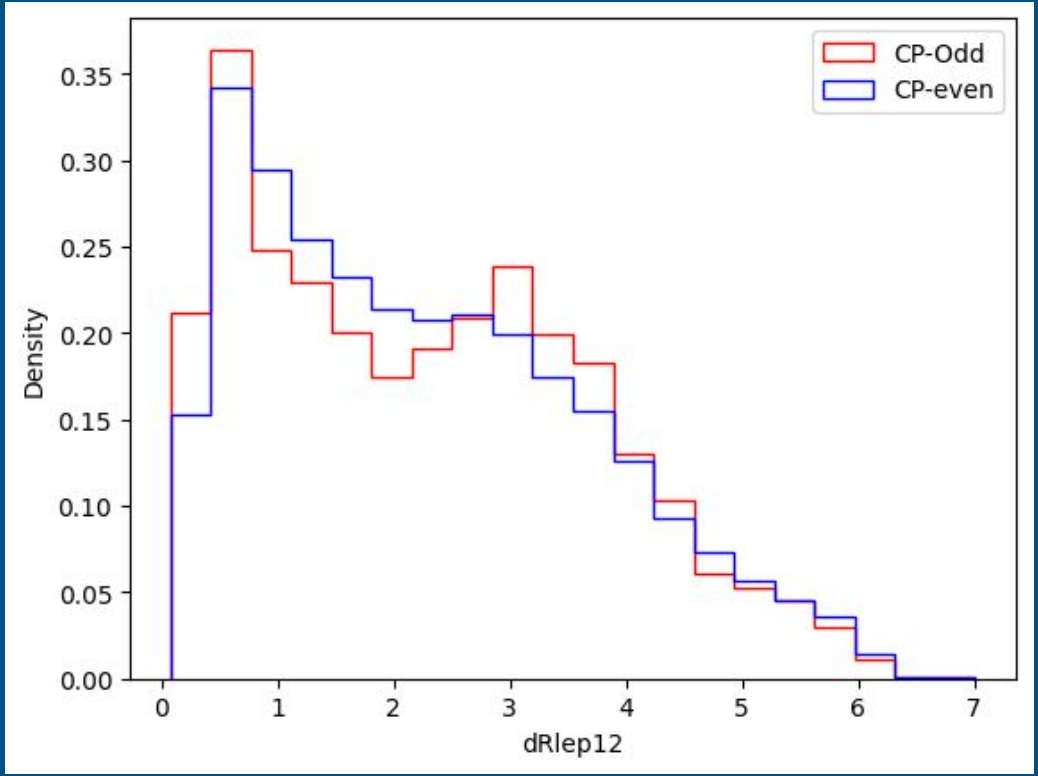
3l0tau Input variables comparison: Run2 vs Run3



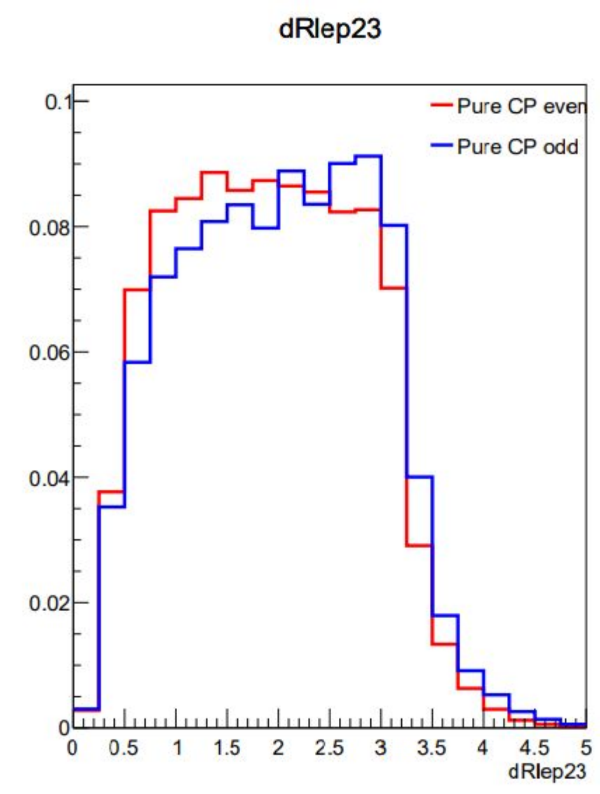


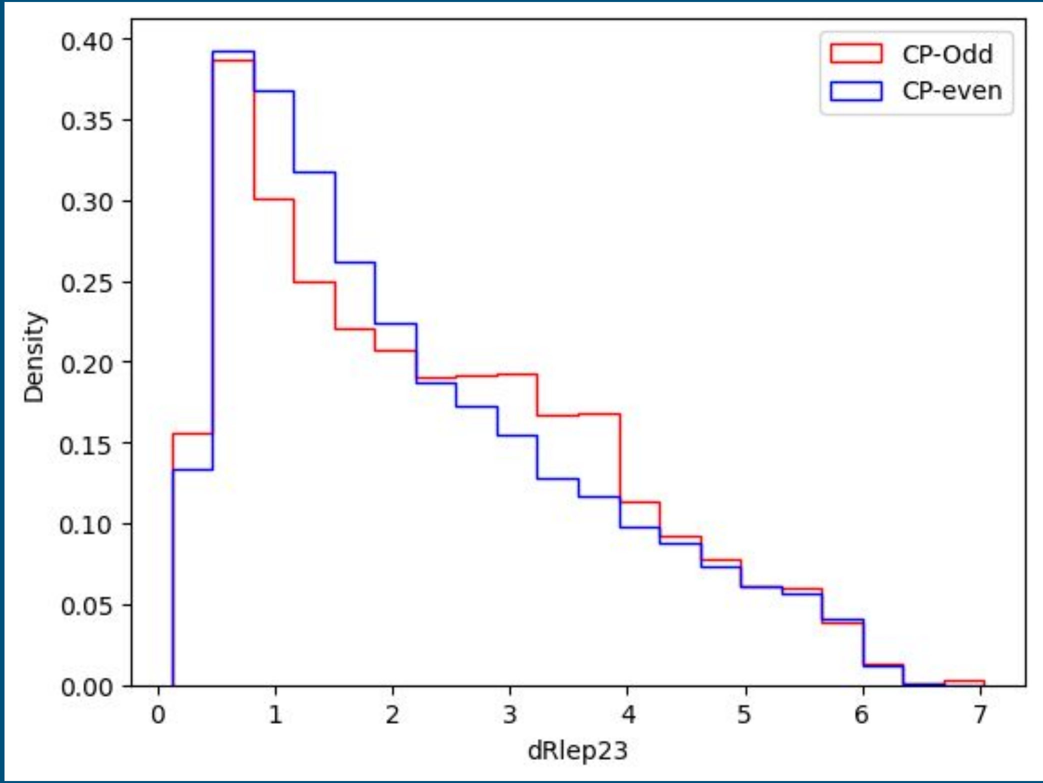
3l0tau Input variables comparison: Run2 vs Run3



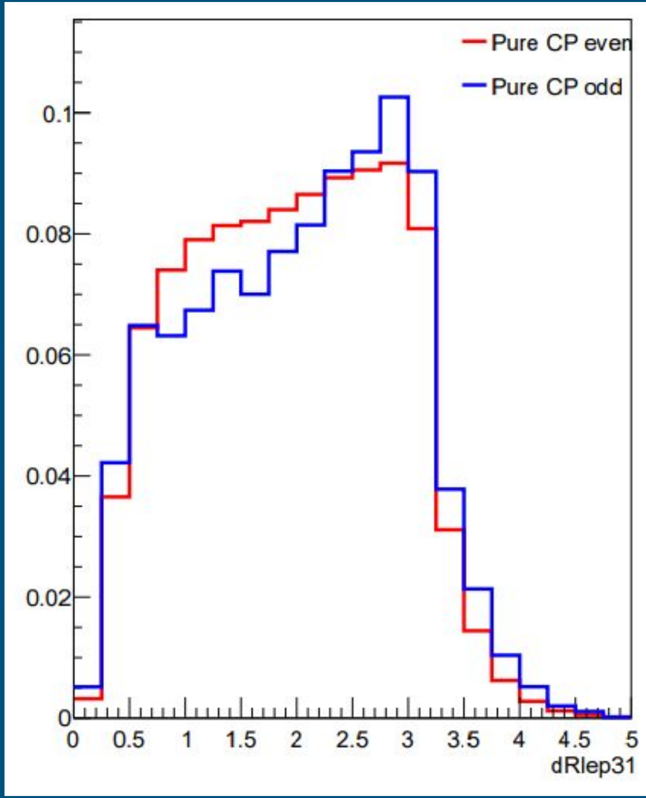


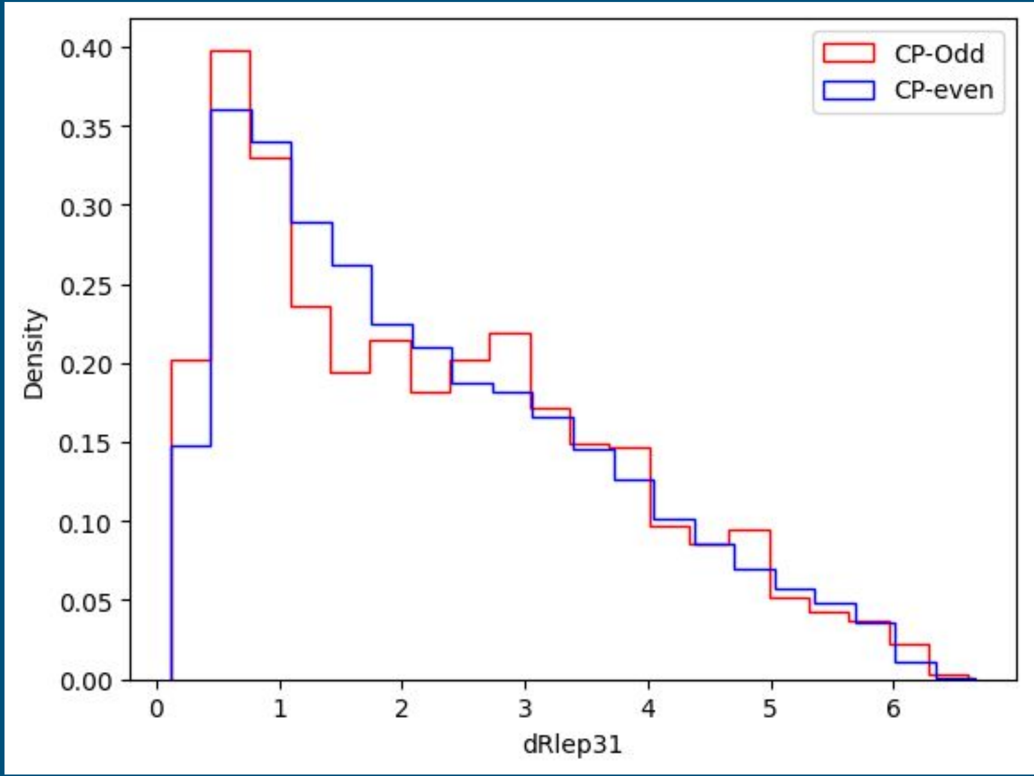
3l0tau Input variables comparison: Run2 vs Run3



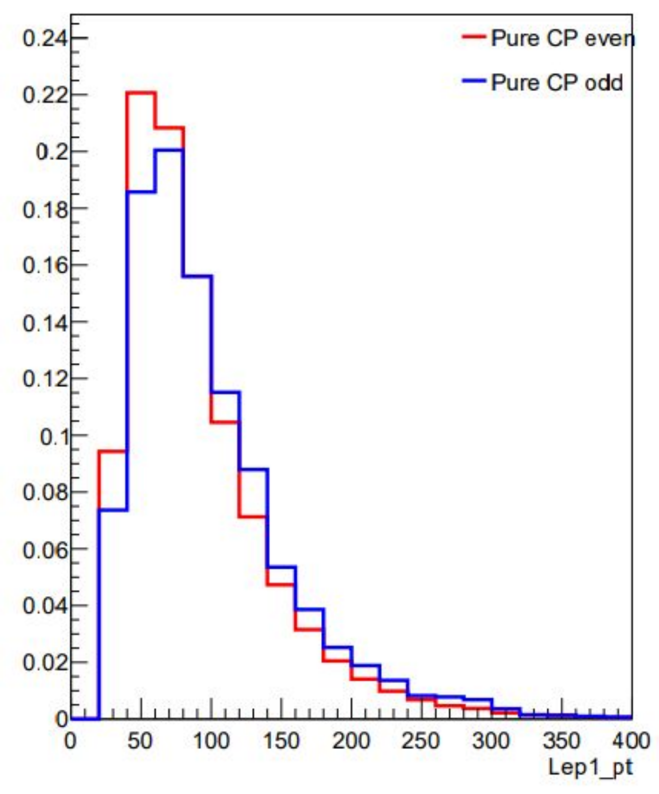


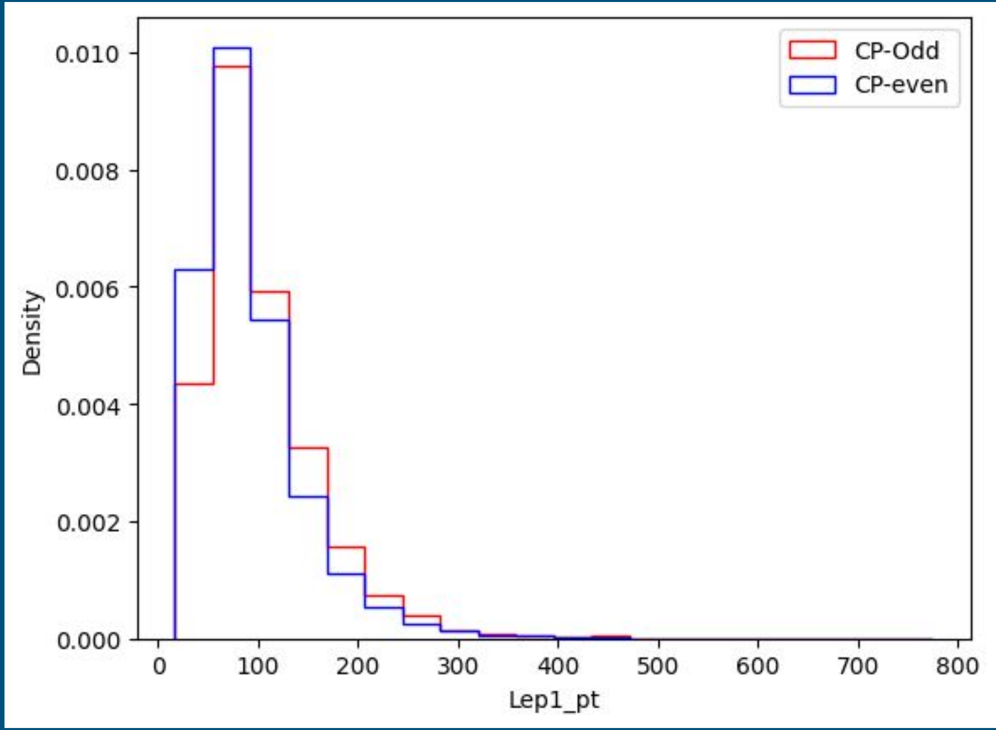
3l0tau Input variables comparison: Run2 vs Run3



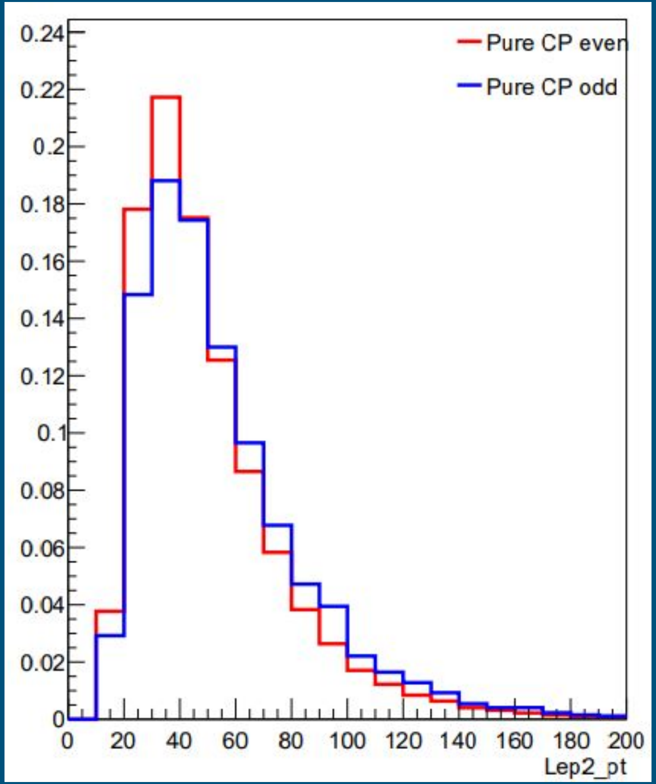


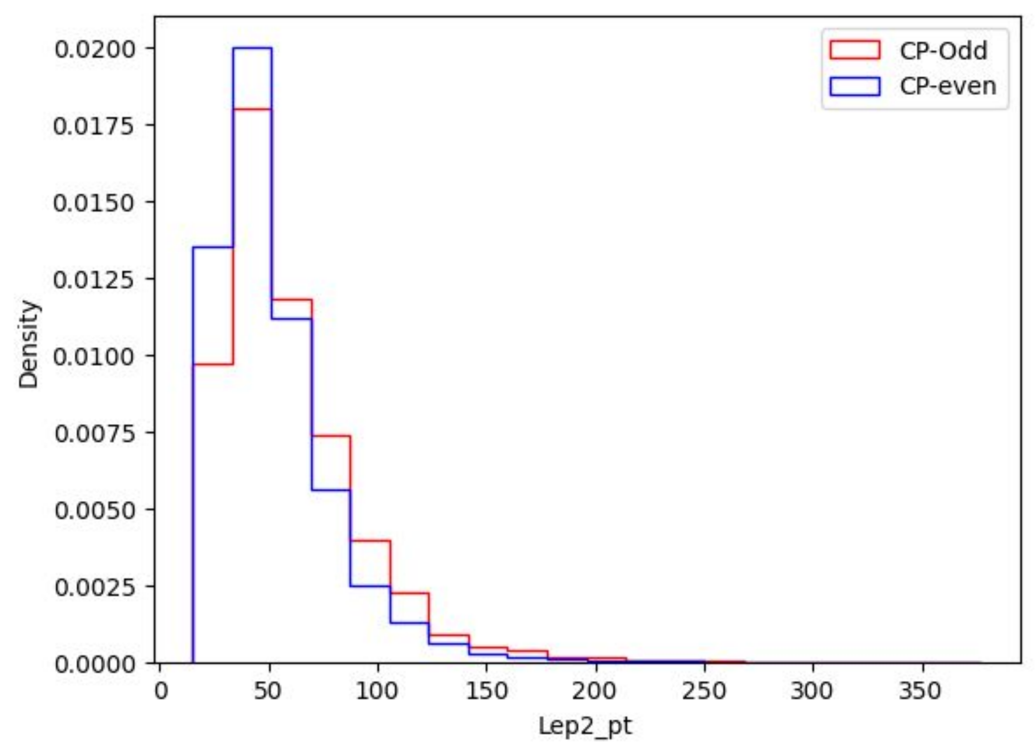
3l0tau Input variables comparison: Run2 vs Run3



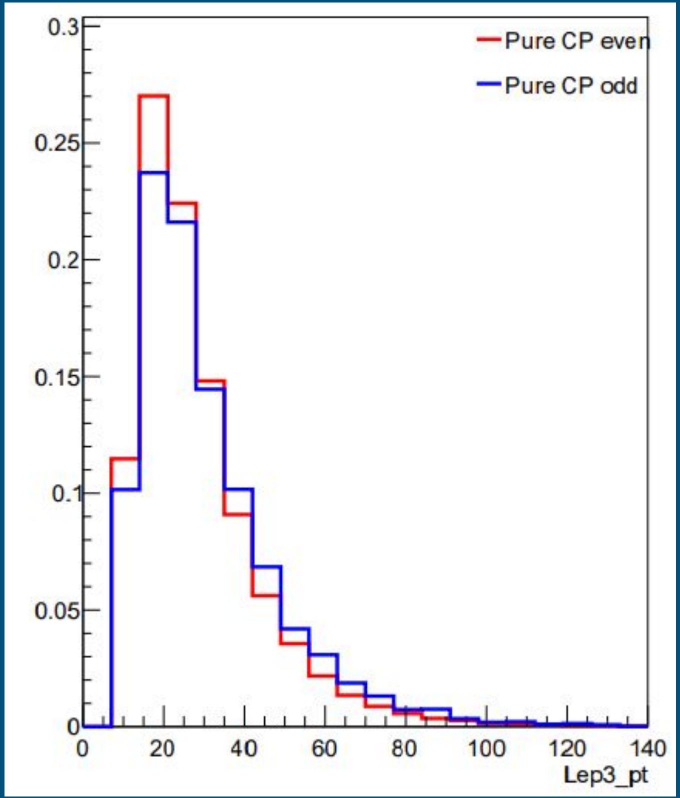


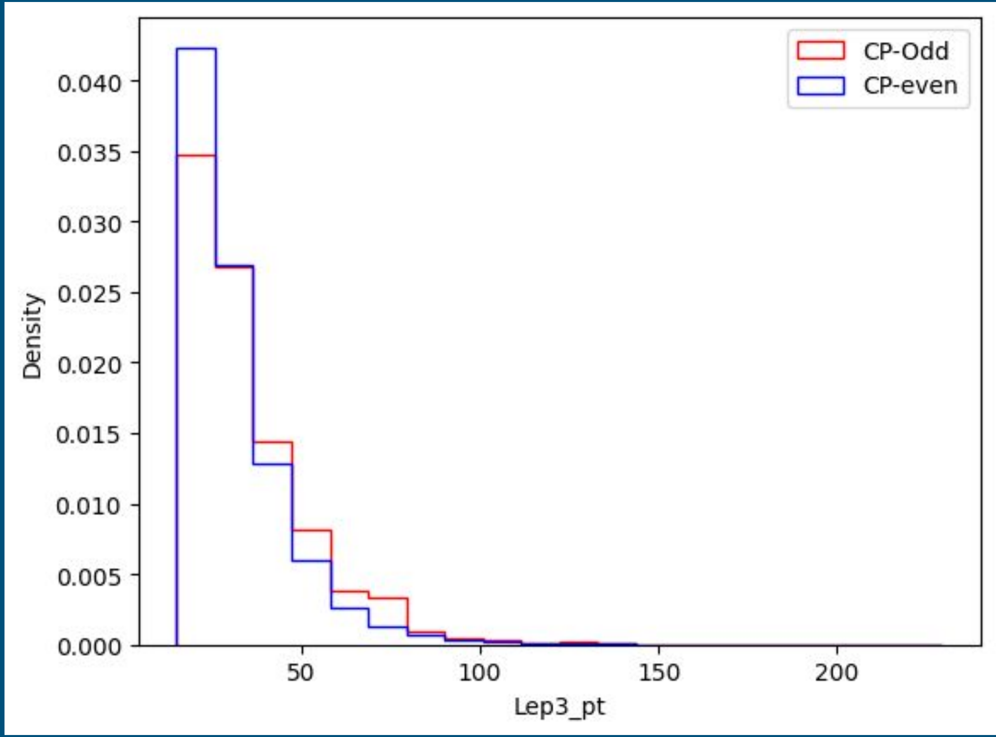
3l0tau Input variables comparison: Run2 vs Run3



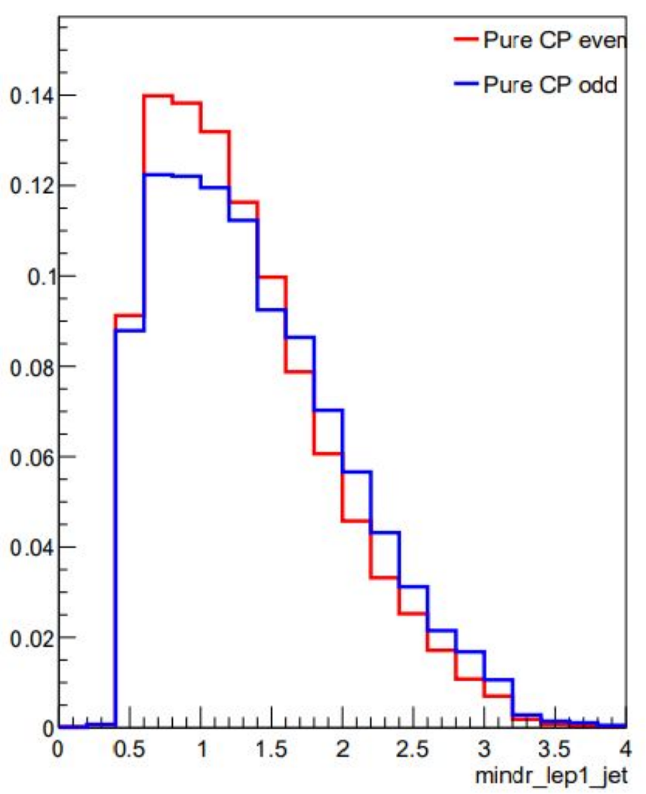


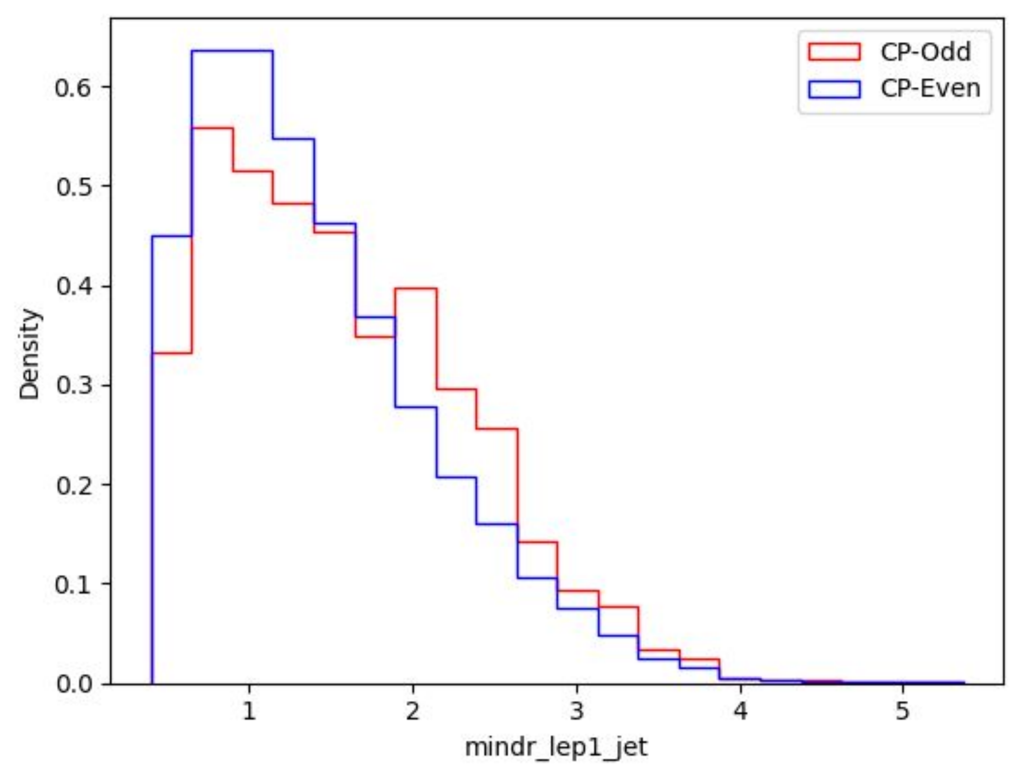
3l0tau Input variables comparison: Run2 vs Run3



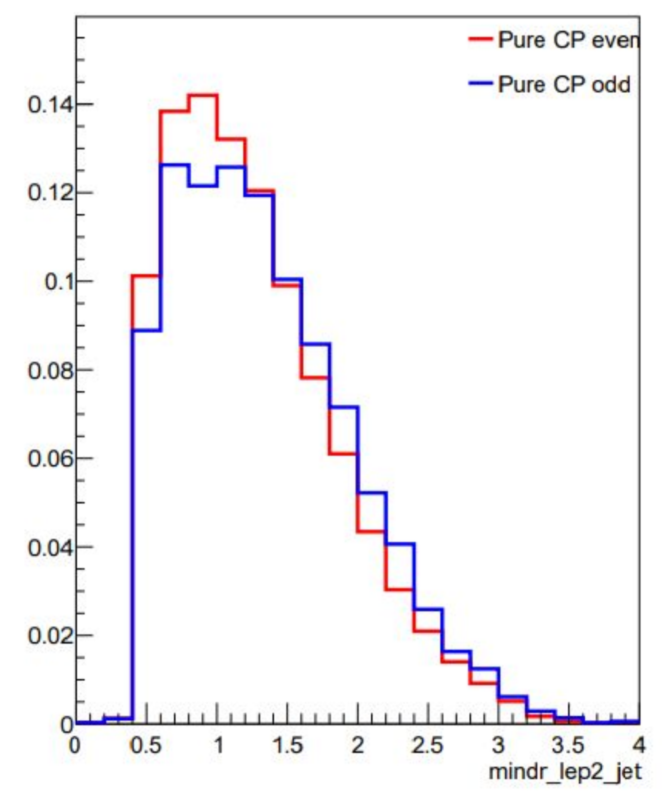


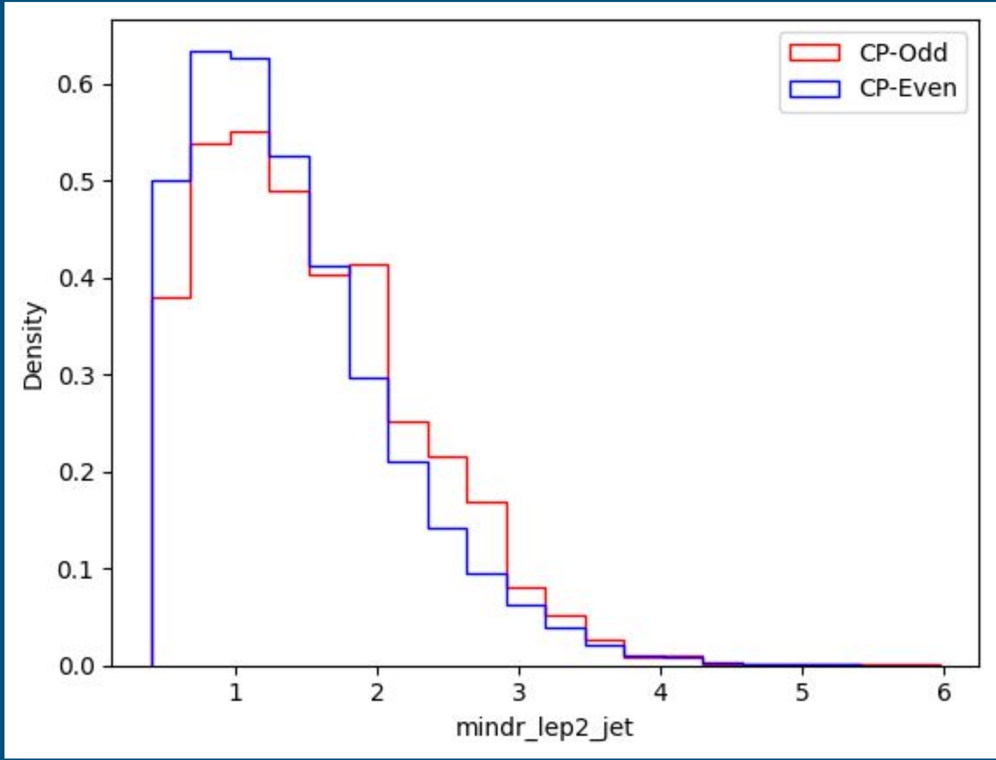
3l0tau Input variables comparison: Run2 vs Run3



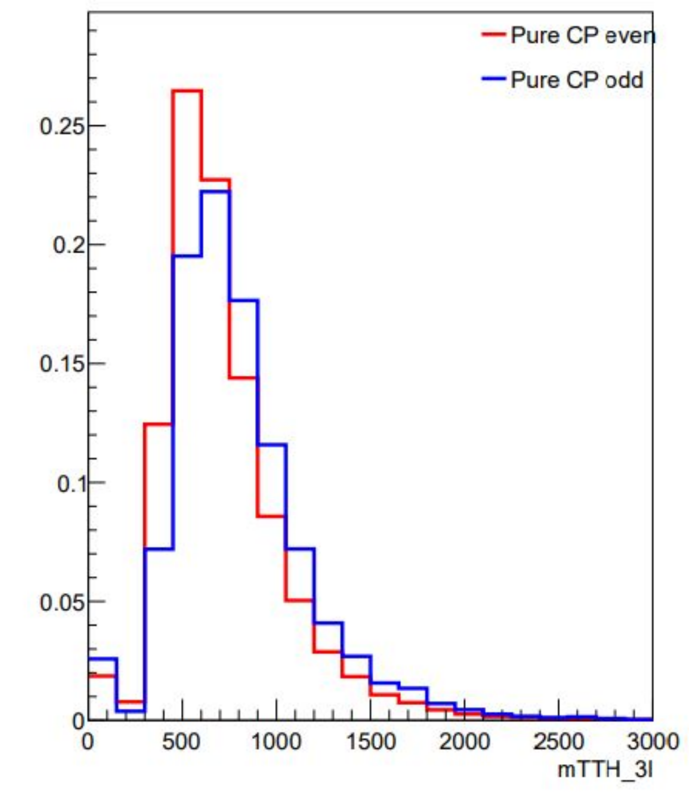


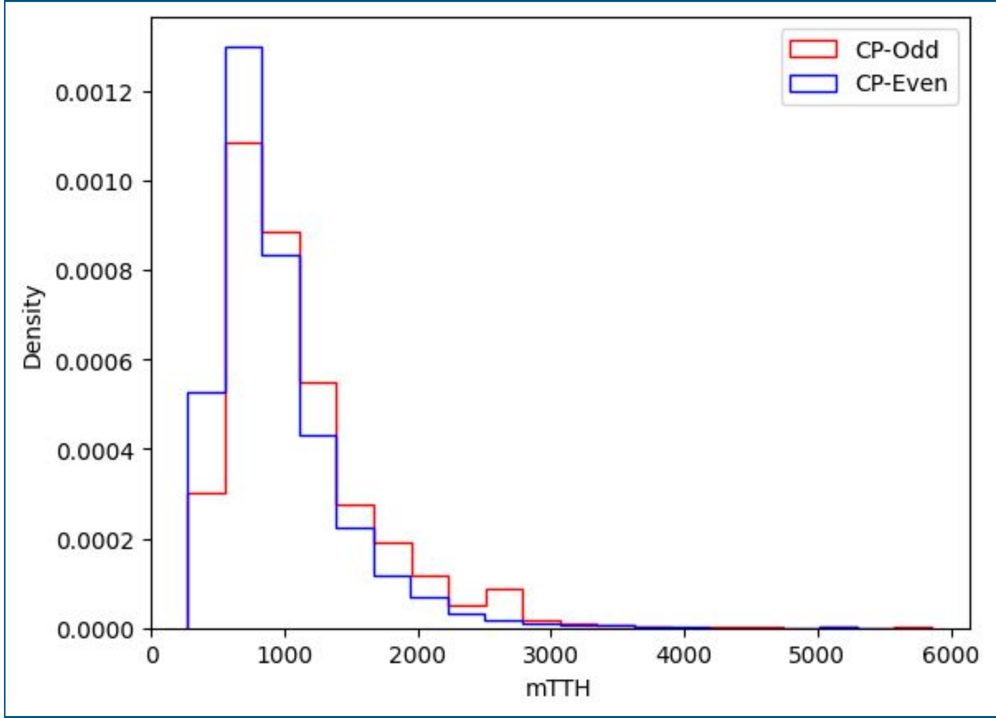
3l0tau Input variables comparison: Run2 vs Run3



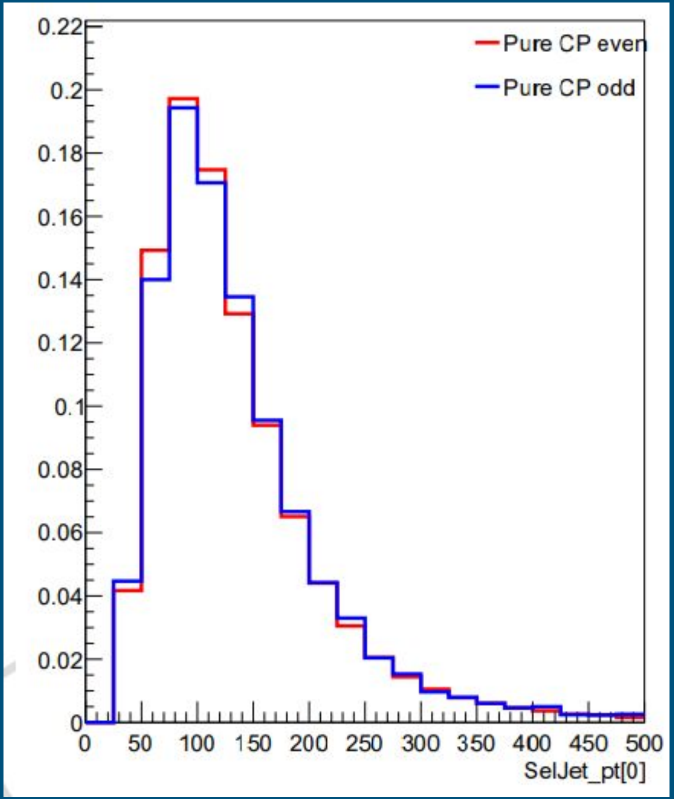


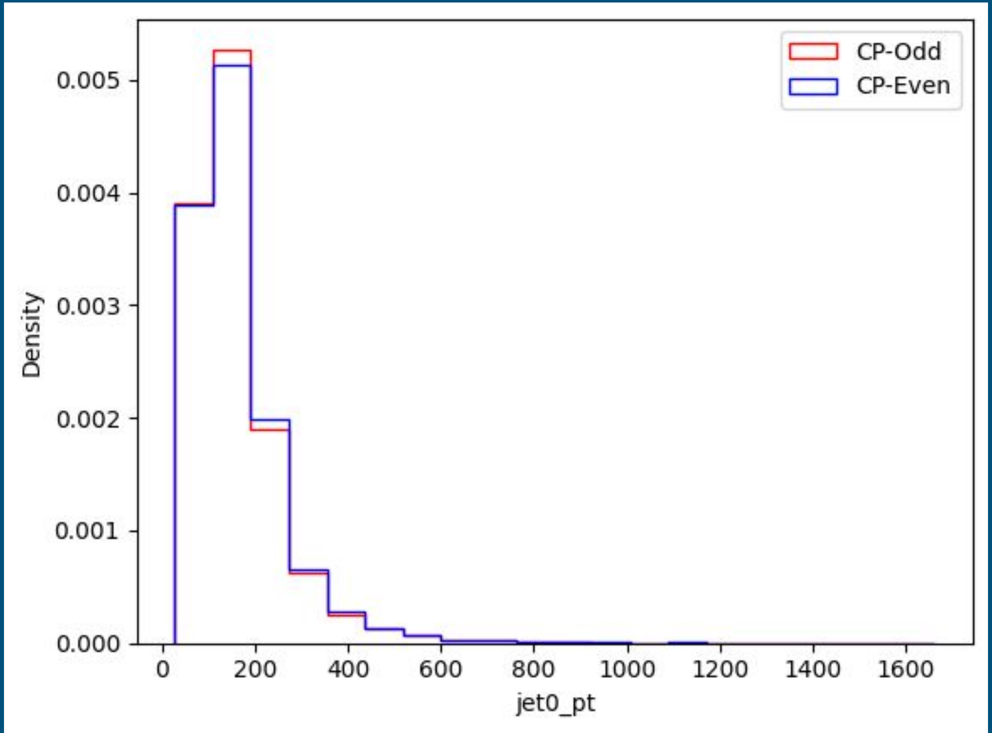
3l0tau Input variables comparison: Run2 vs Run3



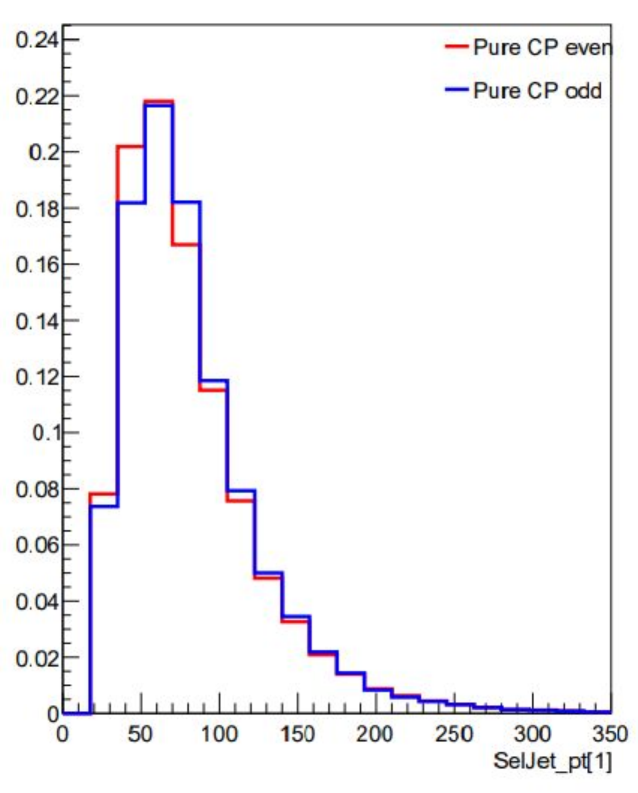


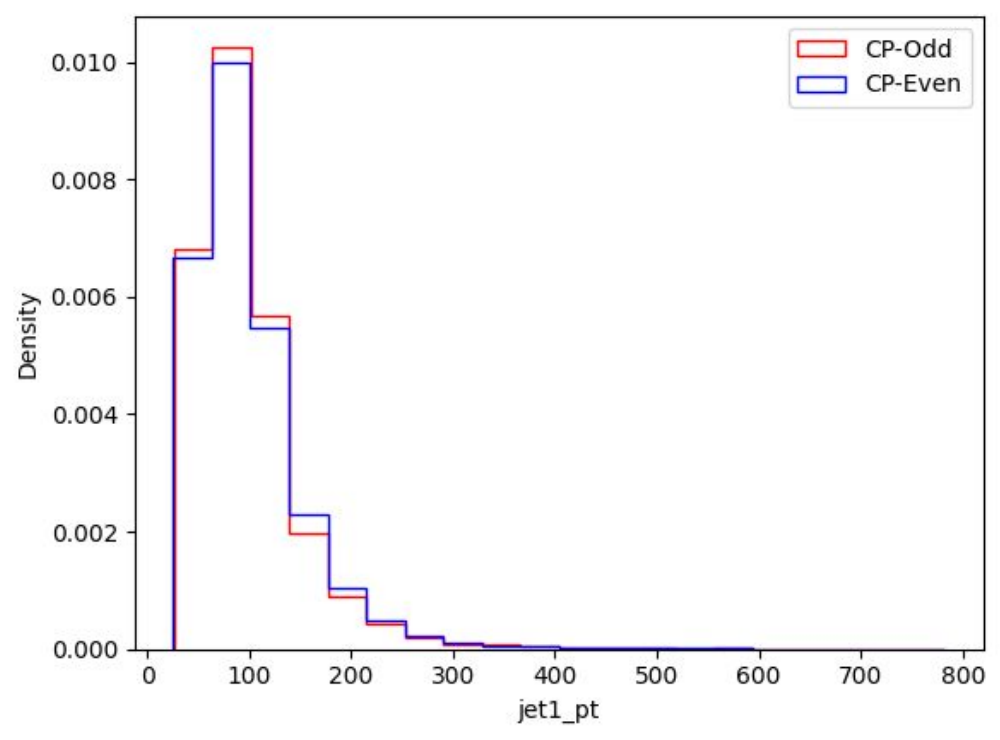
3l0tau Input variables comparison: Run2 vs Run3



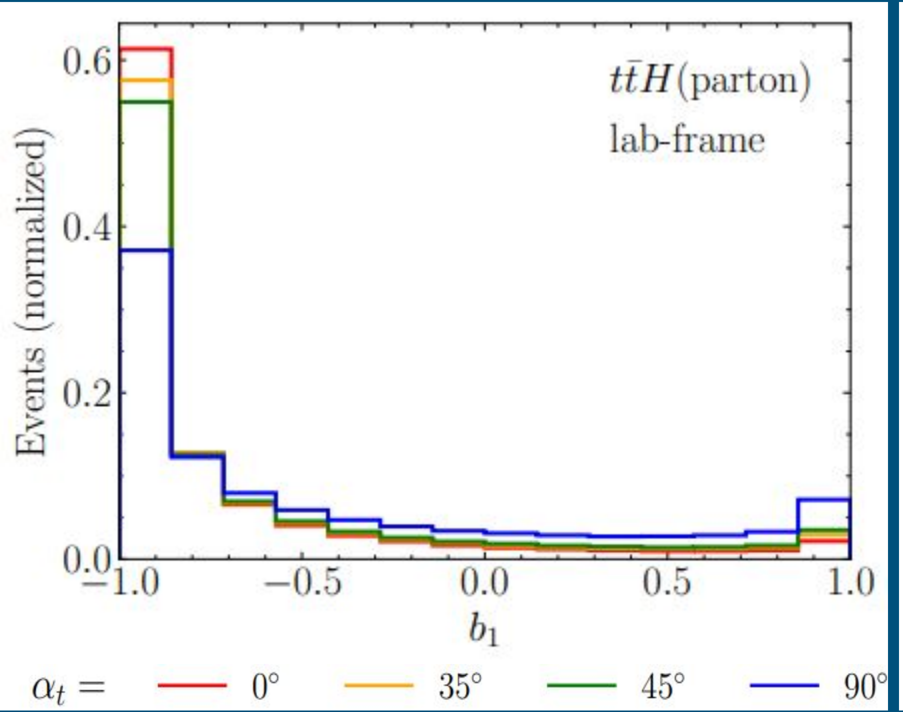


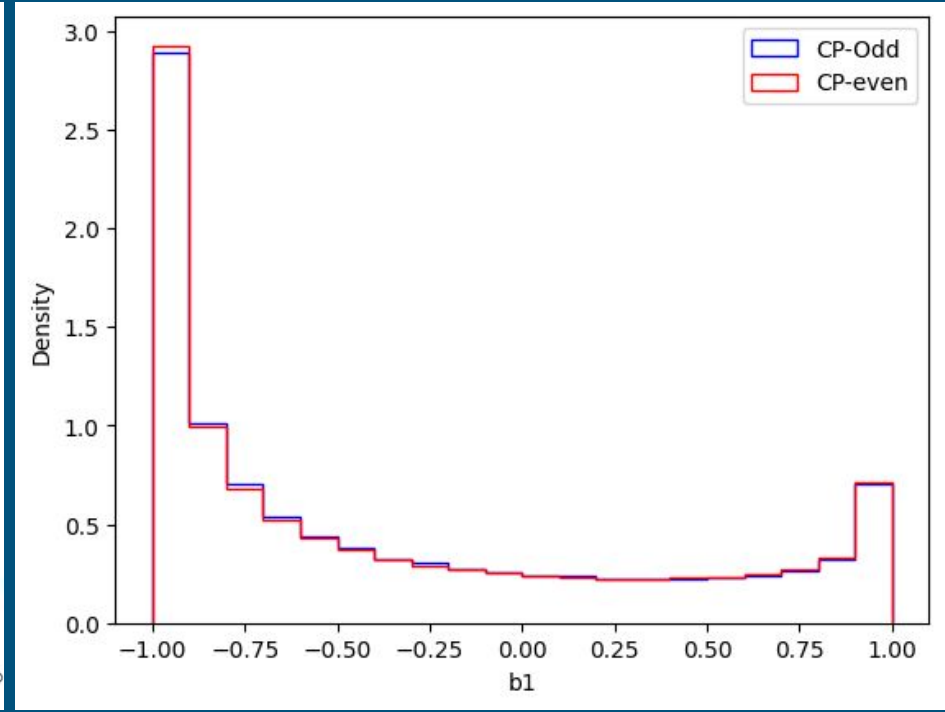
3l0tau Input variables comparison: Run2 vs Run3



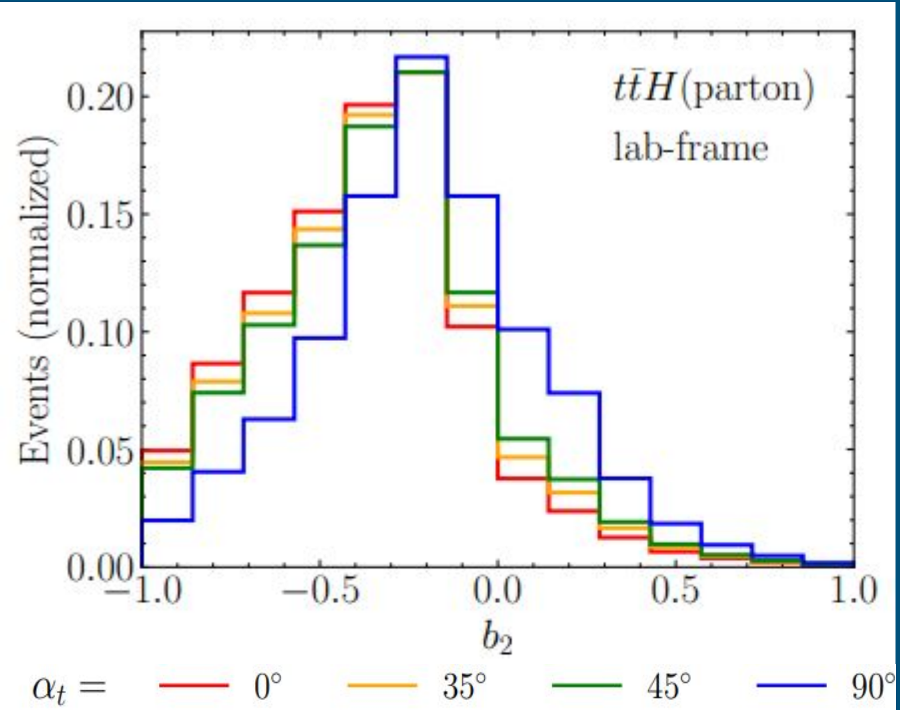


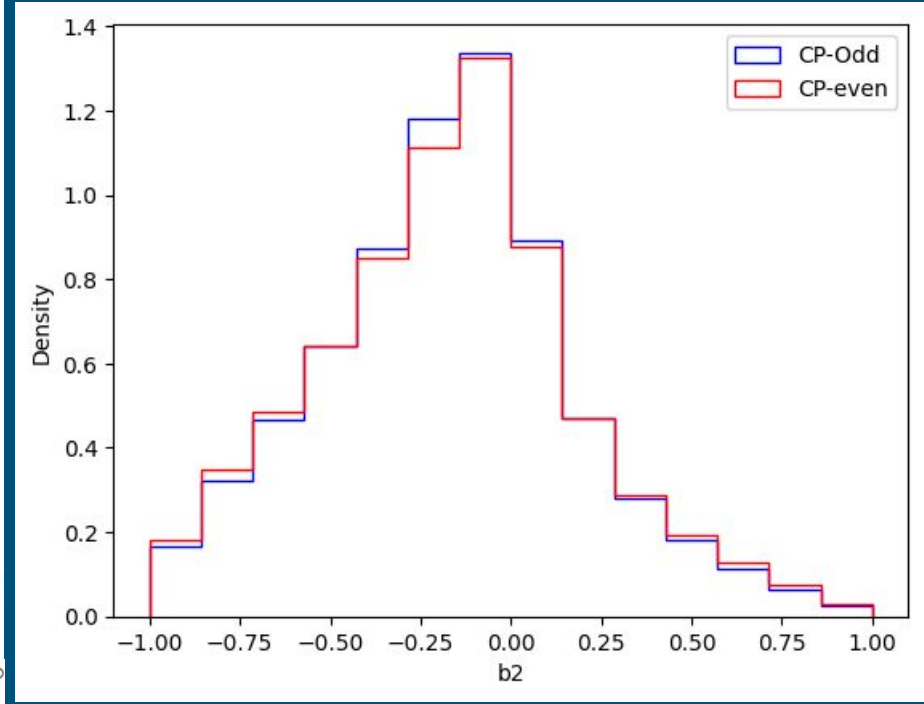
Input variables comparison: Paper vs Run3



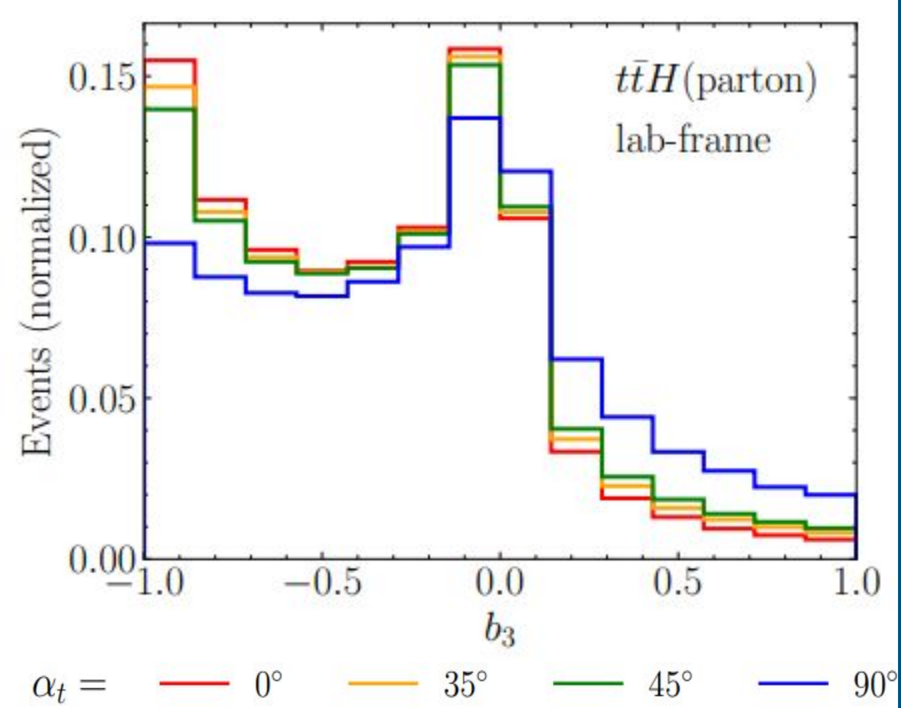


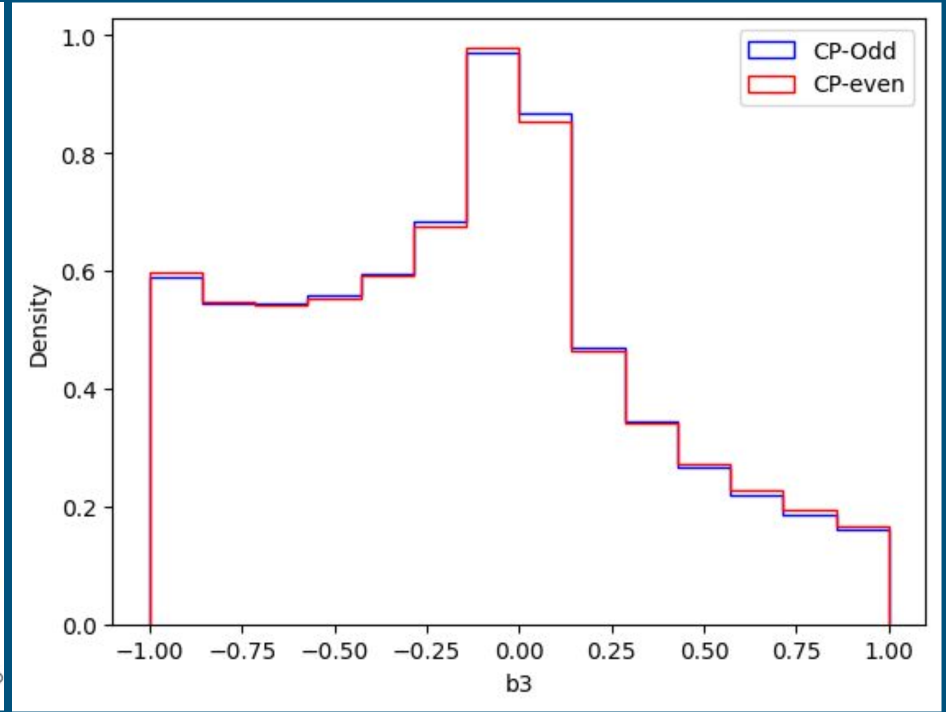
Input variables comparison: Paper vs Run3



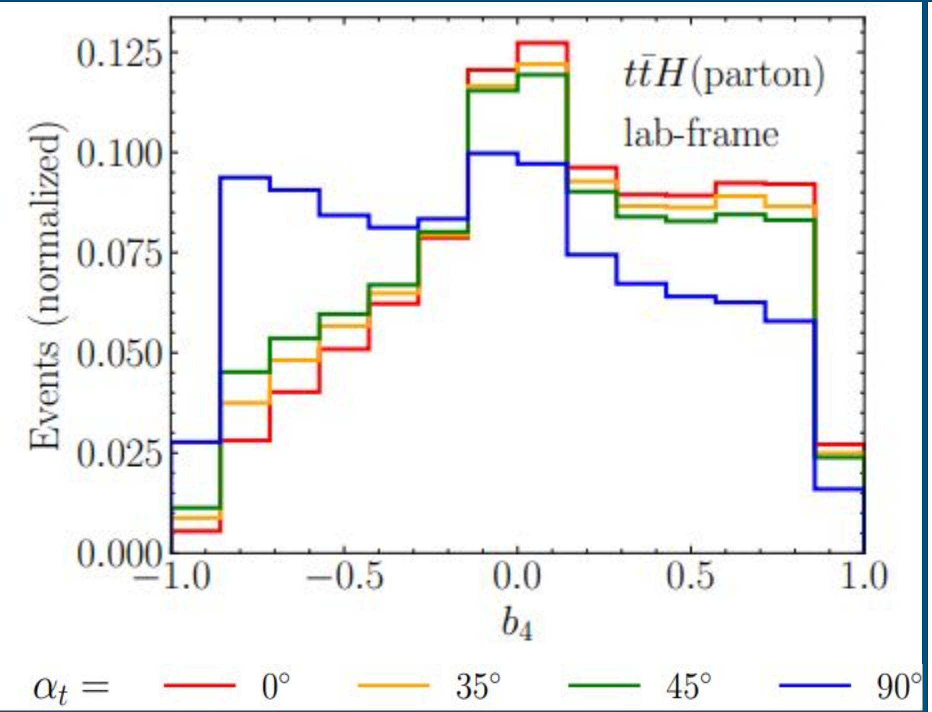


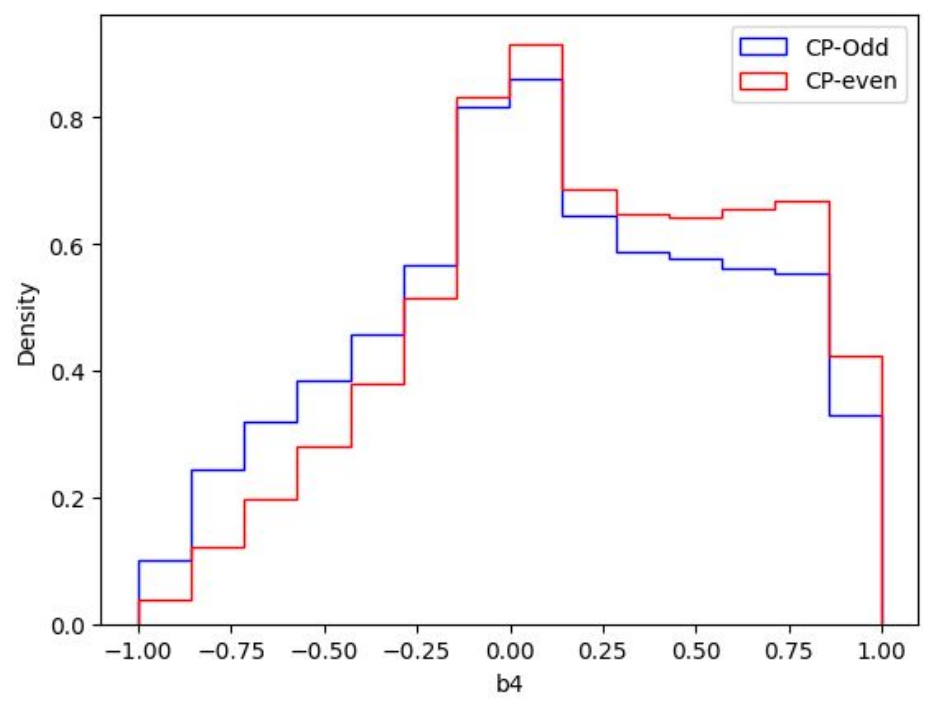
Input variables comparison: Paper vs Run3





Input variables comparison: Paper vs Run3





Input variables comparison: Paper vs Run3