Tuning Herwig 7 with Lund String Model

Pratixan Sarmah

JAGIELLONIAN UNIVERSITY IN KRAKÓW

UCZELNIA BADAWCZA INICJATYWA DOSKONAŁOŚCI

NARODOWE CENTRUM NAUKI

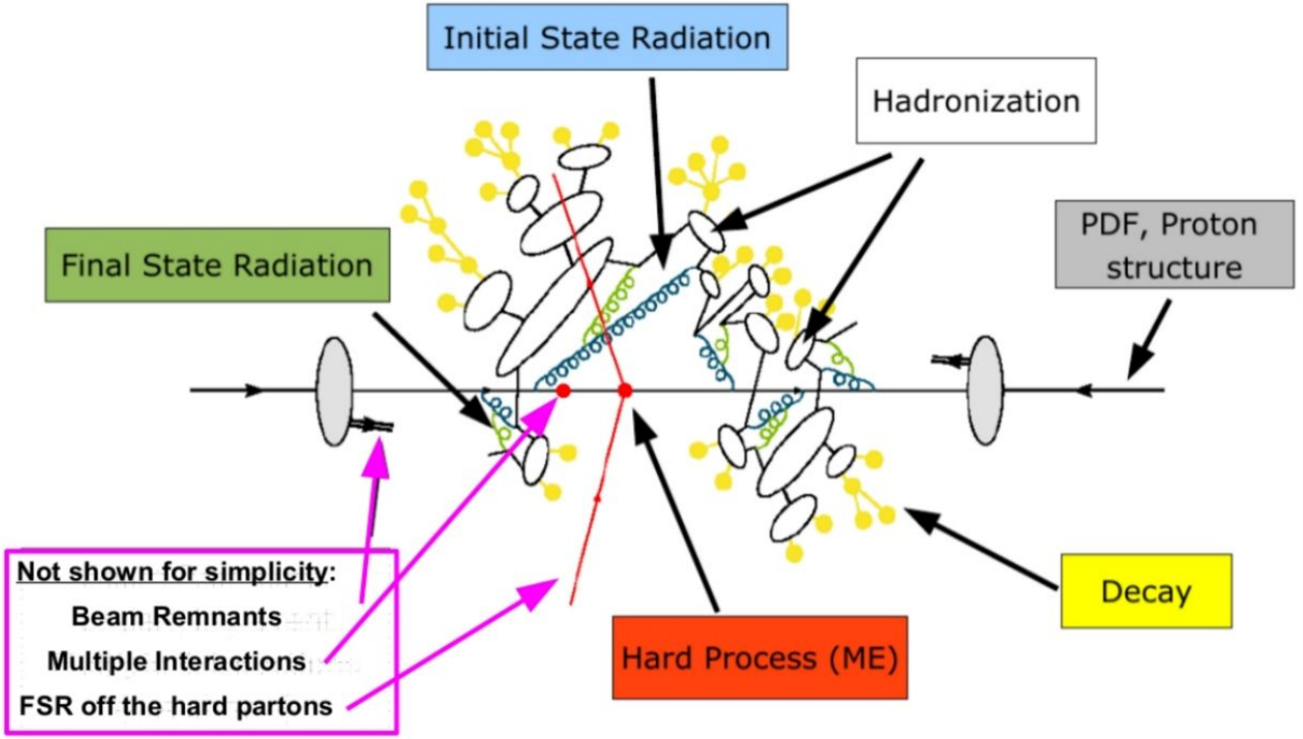
NCN: 2019/34/E/ST2/00457

in collaboration with

Miroslav Myska and Andrzej Siódmok

Introduction

- **Monte Carlo generators** are tools that can simulate collisions and bridges the **gap** between theory and experiment. An event generation simulation with the different stages looks like:

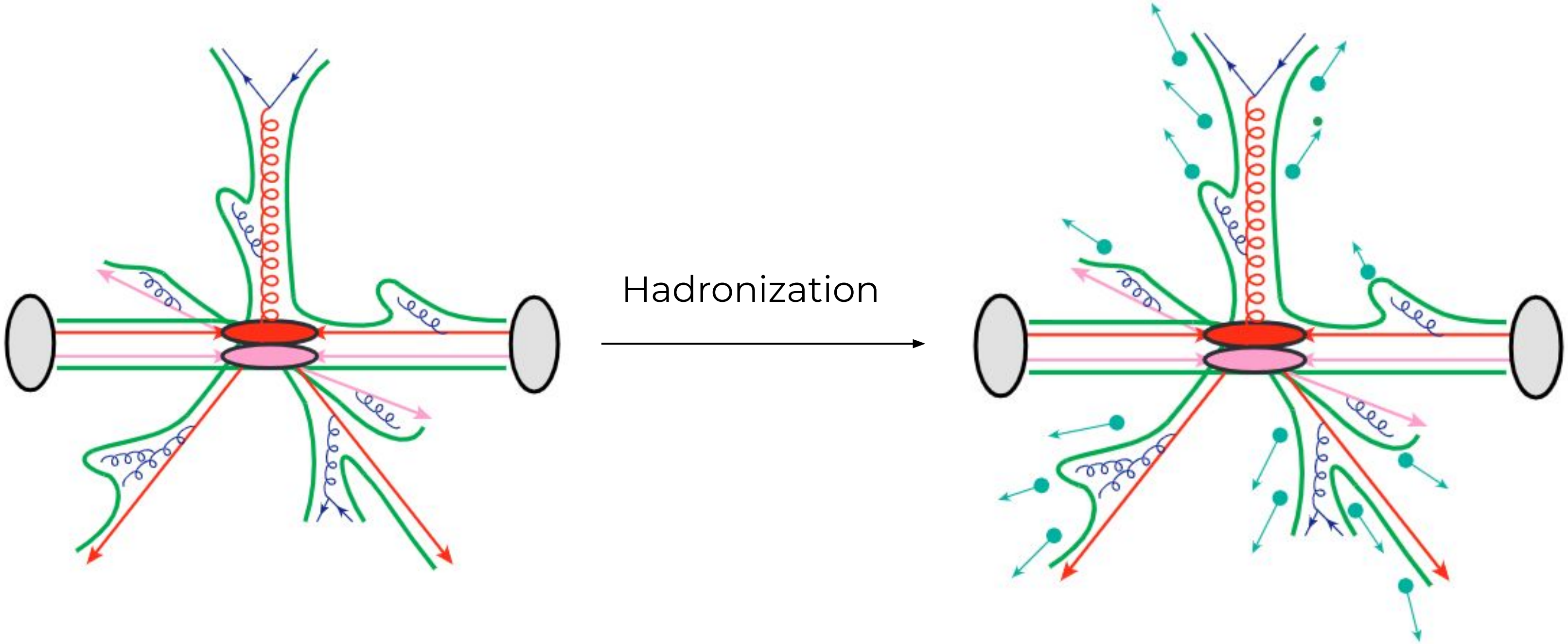


taken from Stefan Gieseke©

- **Perturbative** region is well known and accurately calculated - Hard Process (ME), Parton Shower resummation

Introduction

- **Non-perturbative** region requires phenomenological models to describe the processes in that region
- One such process is **Hadronization** - two most commonly used models are **String Model** in Pythia and **Cluster Model** in Herwig and Sherpa



- Such models generally contain several free parameters that cannot be inferred from first principles and require **tuning** to experimental data.

Images taken from S.Prestel

Motivation

- Several tunes have been performed to LHC data over the years for event generators Pythia and Herwig

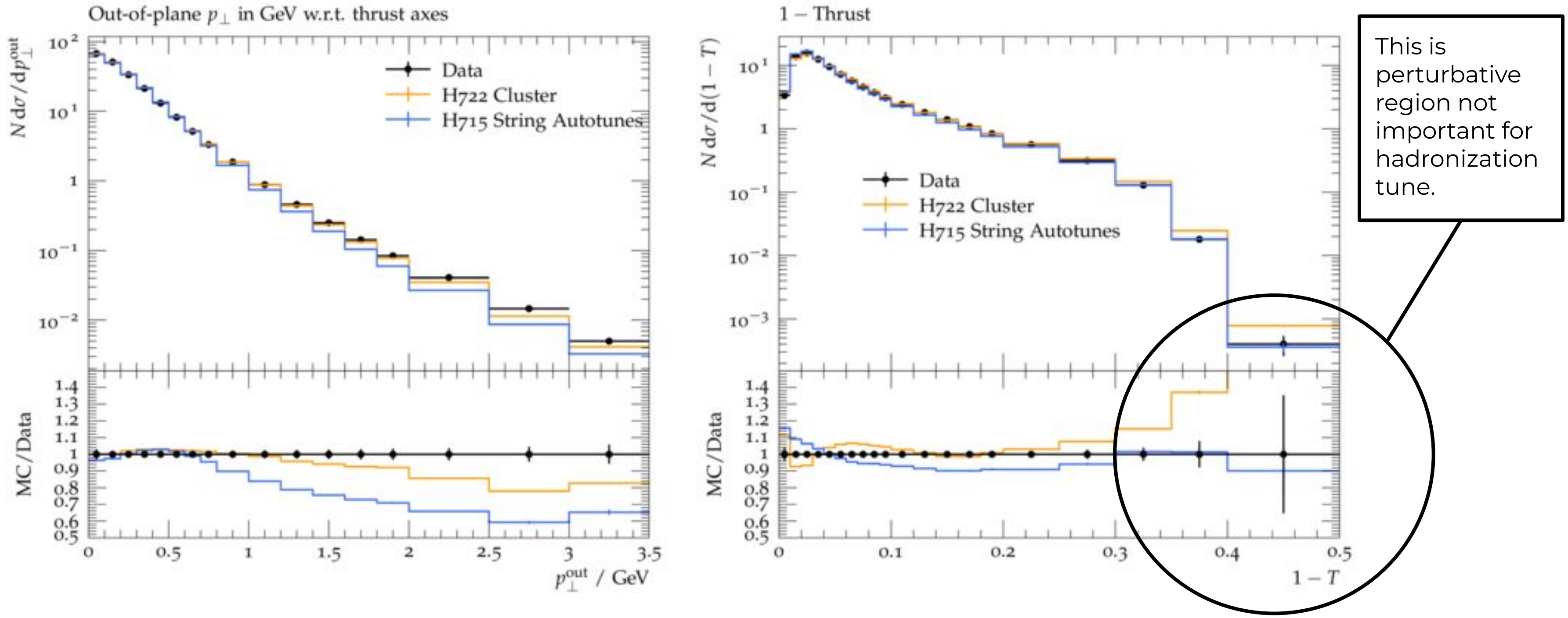
- It has been seen, the **Lund String** model in Pythia does better describing data sensitive to hadronization process and the **Angular Ordered (AOPS)** shower in Herwig7 does better in regions sensitive to perturbative calculations

- A first attempt was made to tune Herwig7 with Lund String in a new framework named AutoTunes [J.Bellm, L.Gellersen, EPJC 2020]

- **TheP8I**, written by L. Lonnblad allowed the internal use of Pythia 8 strings with Herwig 7 events

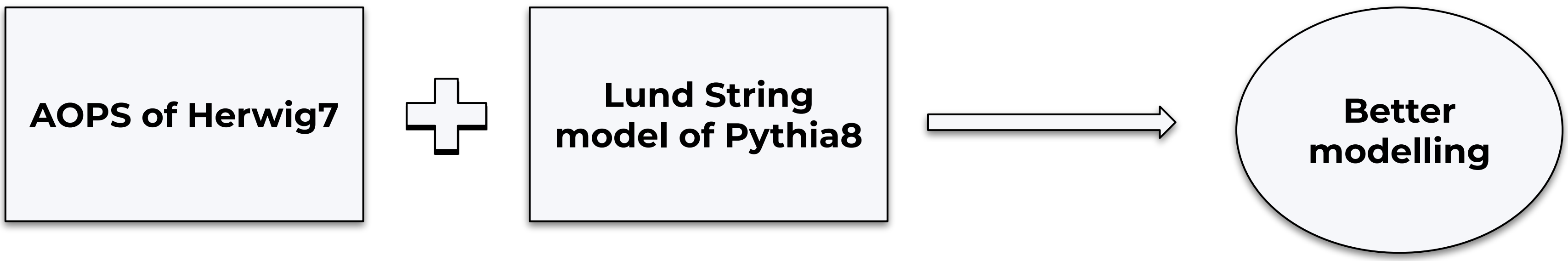
Motivation

- However, the AutoTunes paper focused on a new automatized tuning procedure than getting the best tune.
- Thus, we reproduced the results based on their tuned parameters
  [there were no plots comparing the tune to the data in the paper]
- The results didn't show much improvement and it also performed worse in many regions when compared to the default Herwig7 + cluster tune.



Motivation

- We would naively expect:



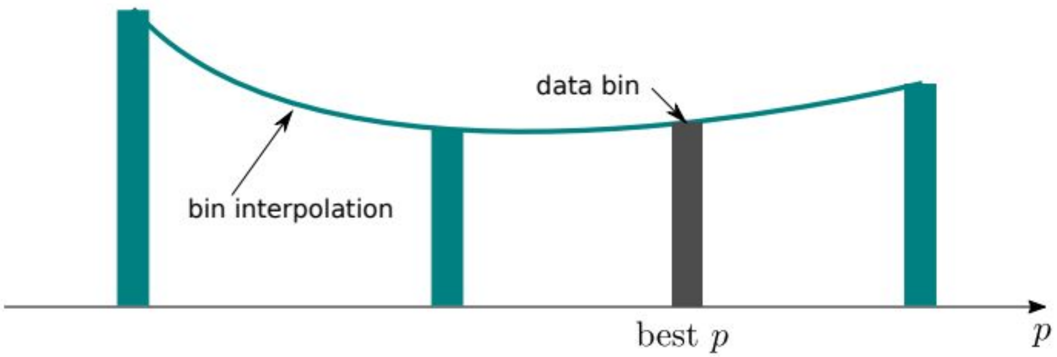
- Thus, we want to understand if the gap between our expectation and previous results are due to the tuning setup or physics understanding of the NP* region.

- A successfully combined model would also provide a setup for a dedicated study of the systematics of NP effects.

*NP - Non-Perturbative

Tuning strategy

- We follow the PROFESSOR approach (as done in Pythia 6 tune) to tune the AOPS of Herwig7 + Lund String model to **LEP data** [A.Buckley et al, EPJC 2010].
  Two stages of tuning:
  - Fragmentation parameters and
  - Flavour parameters



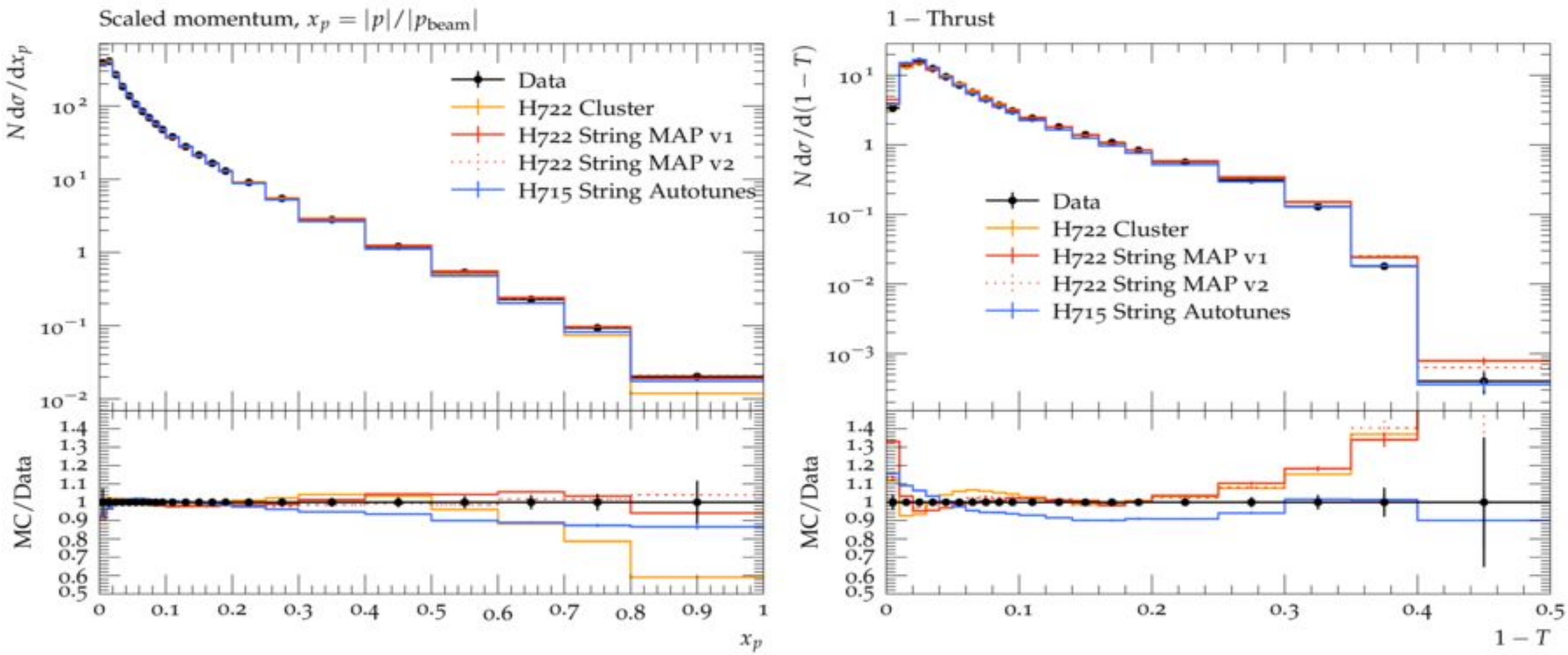
- The **PROFESSOR** framework parameterizes the generator response to a n-order polynomial and maximizes a Goodness-of-Fit function to get the set of best parameter values.

- Two different sets of input weights were used to obtain two versions of our **MAP** (Mira, Andrzej, Pratixan) tune v1 and v2, and compare them with the default Herwig7.2.2 + cluster tune and the AutoTunes tune.

Image taken from L.Gellersen

Event Shape Observables

- Plots from the analysis DELPHI_1996_S3430090

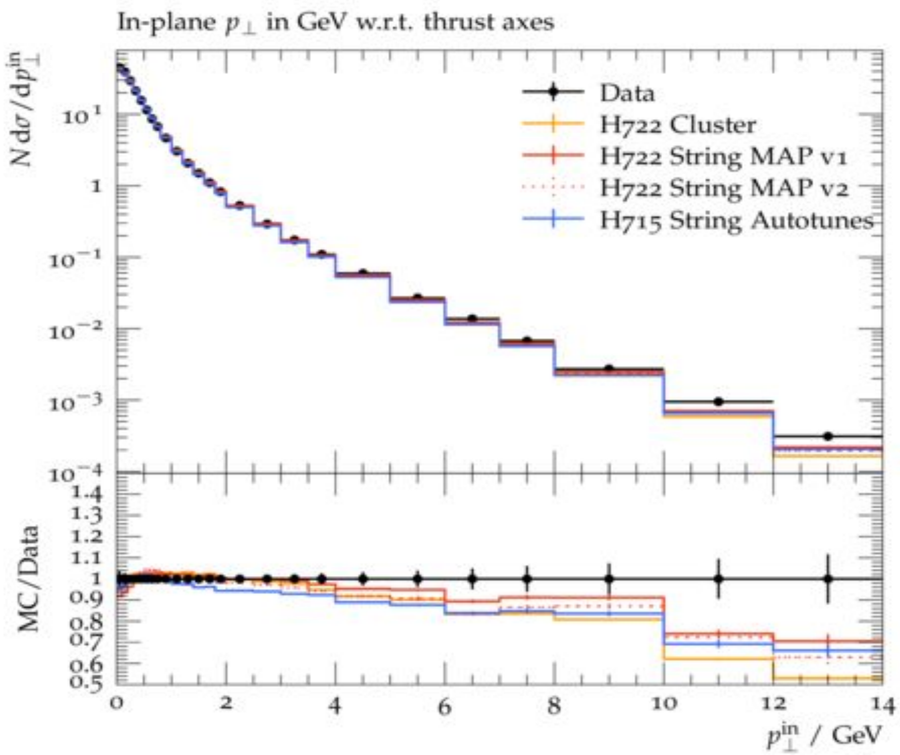


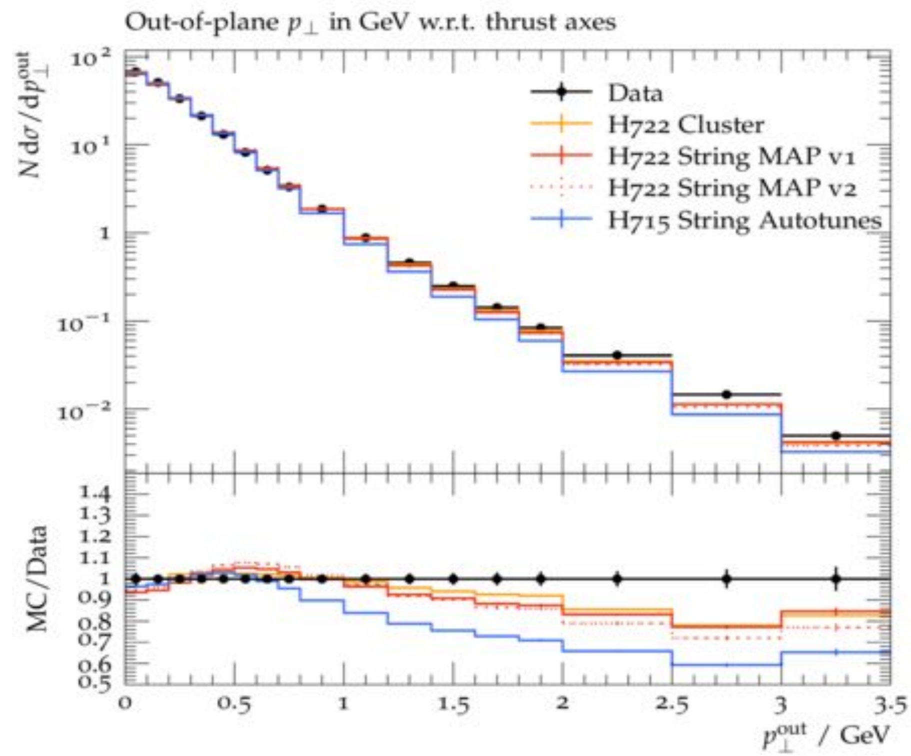
- Both these distributions were used in tune.

- Regions of the distribution sensitive to non-perturbative region are significantly better modelled by MAP tune.
- for x_p distribution, string model does better than the cluster

Event Shape Observables

- Plots from the analysis DELPHI_1996_S3430090

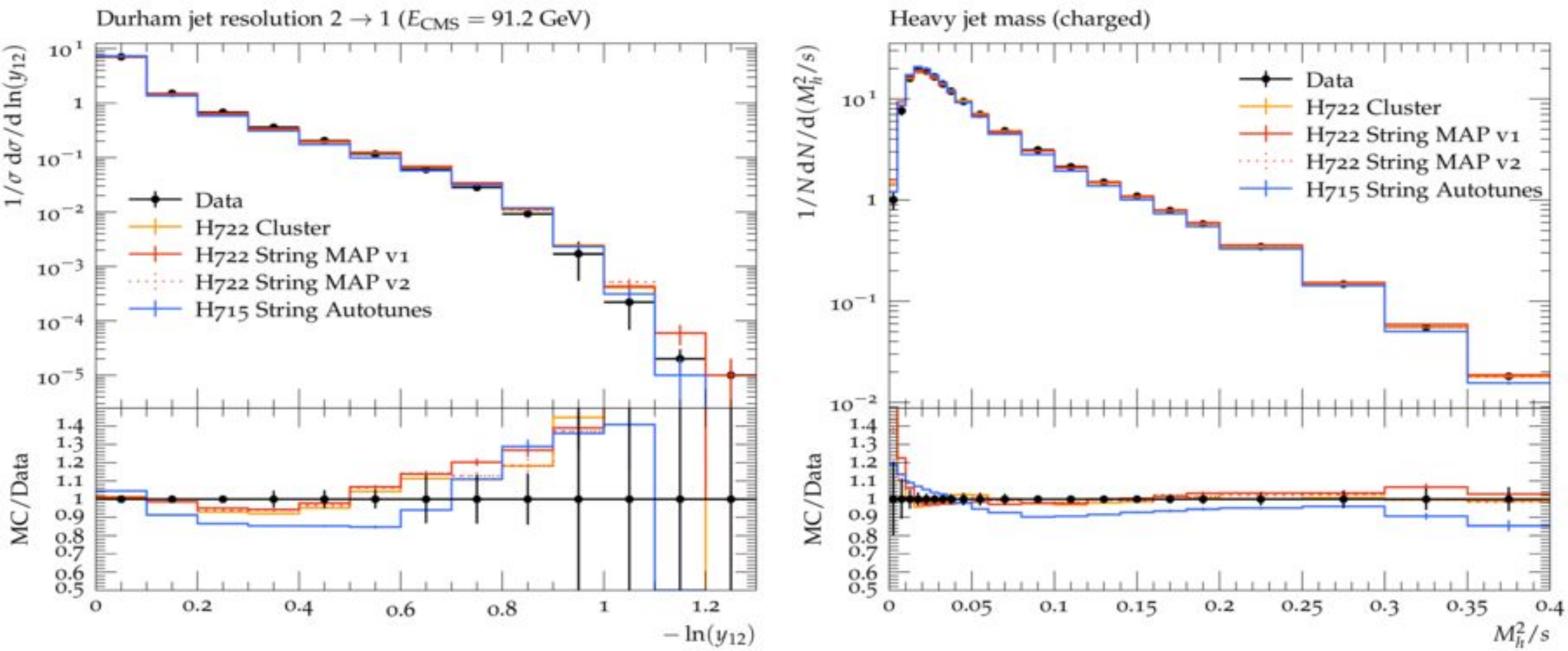




- Both these distributions were used in the tune.

- MAP tune shows small improvement towards the tail of these distributions but is significantly better than AutoTunes tune in the Out-of-plane pT distribution.

- Plots from the analysis ALEPH_2004_S5765862 (left) and ALEPH_1996_S3486095 (right)

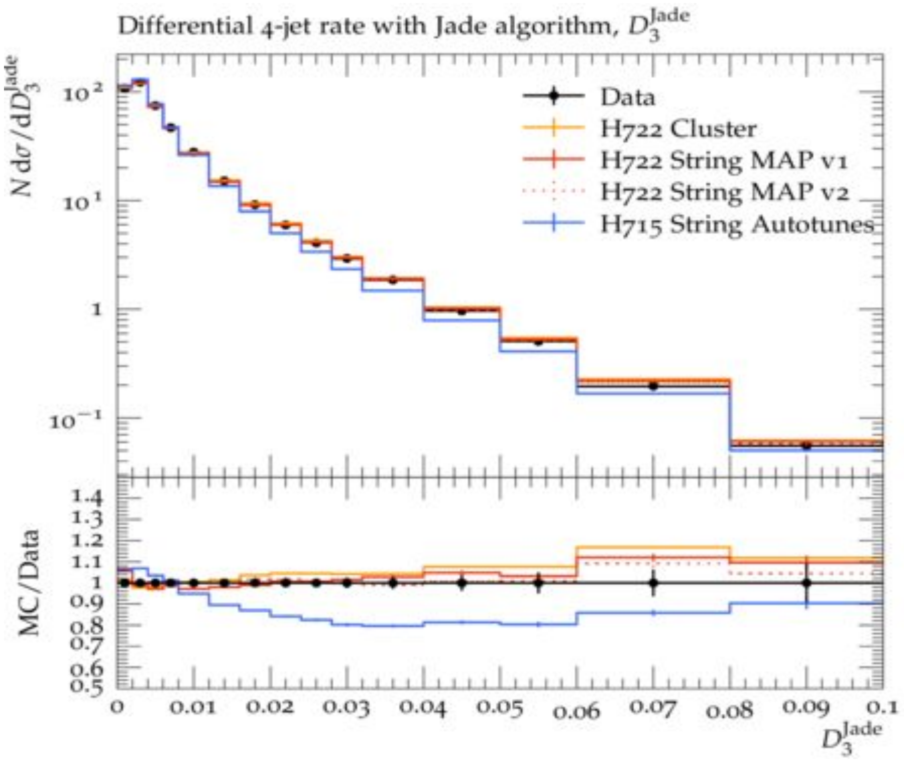


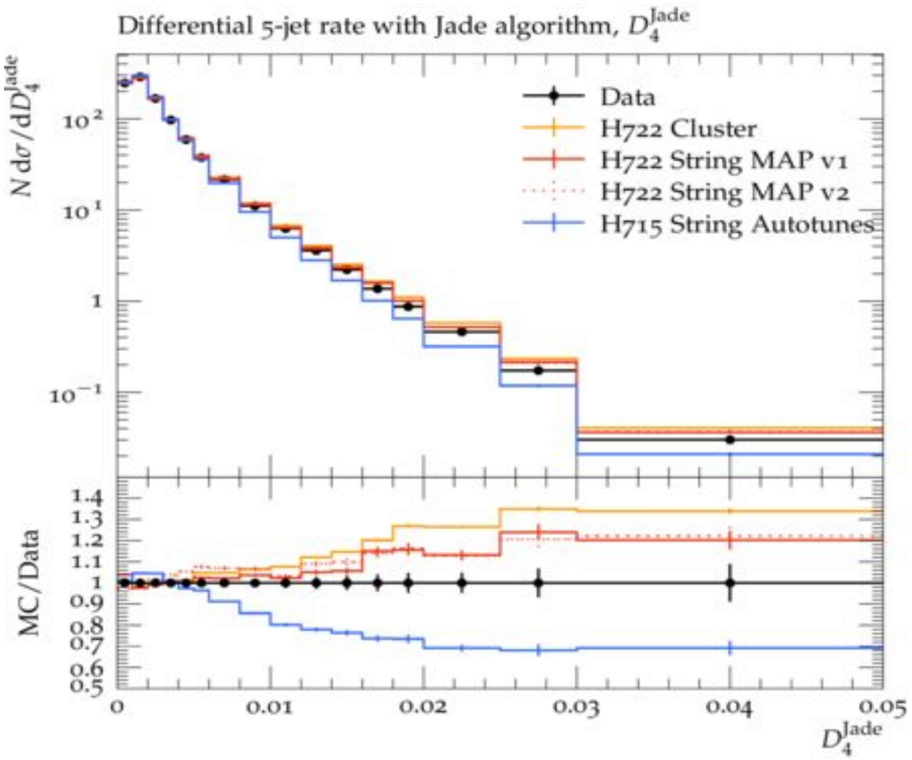
- Both these distributions were **not** used in the tune.

- MAP tune still does better than AutoTunes tune in these distributions and does similar to the default tune - shows robustness of our tune.

Jet Observables

- Plots from the analysis DELPHI_1996_S3430090



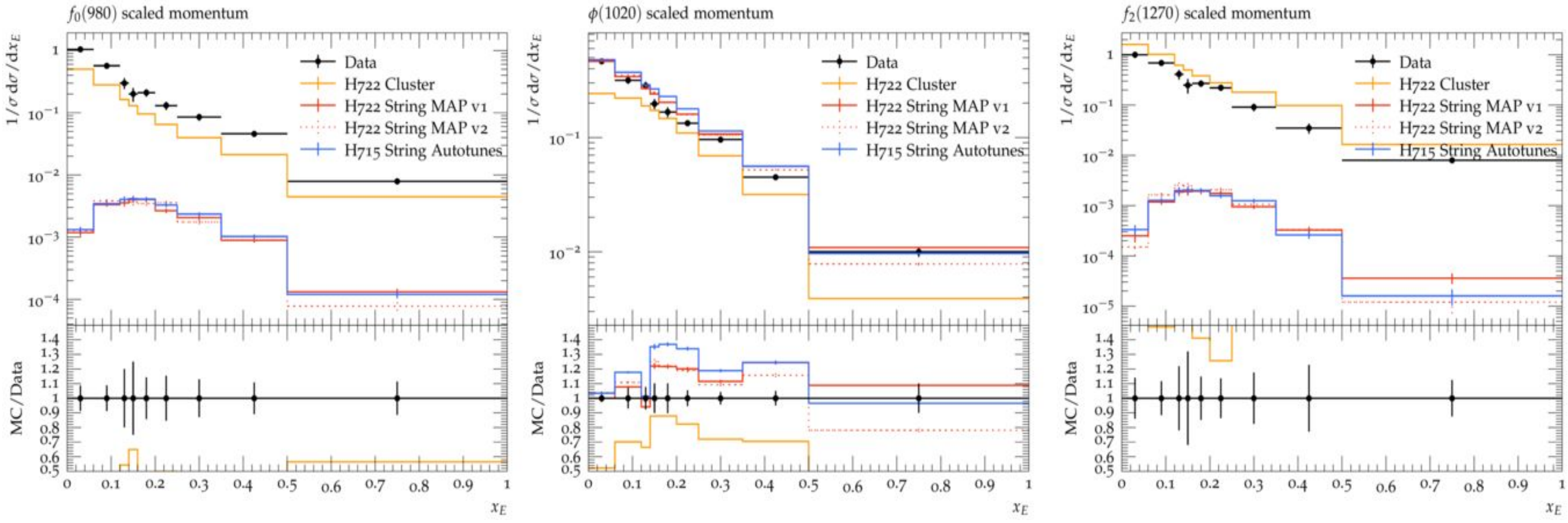


- Both these distributions were **not** used in the tune.

- MAP tune again does better even though jet observable distributions were not weighted in the tuning.

Flavour sensitive Observables

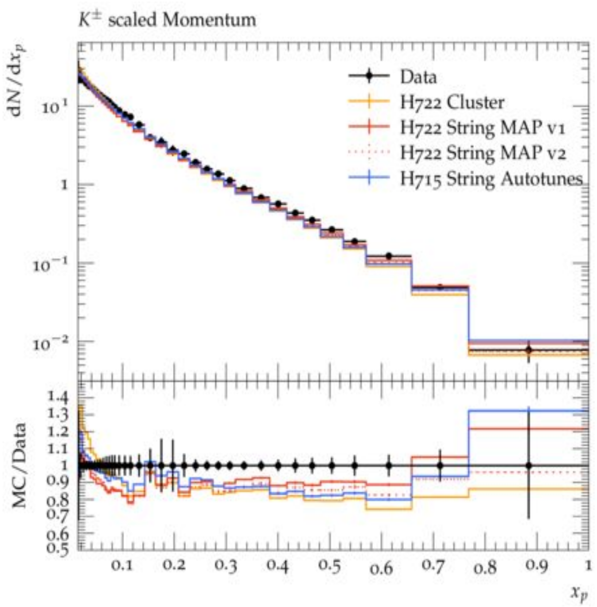
- Plots from the analysis OPAL_1998_S3702294

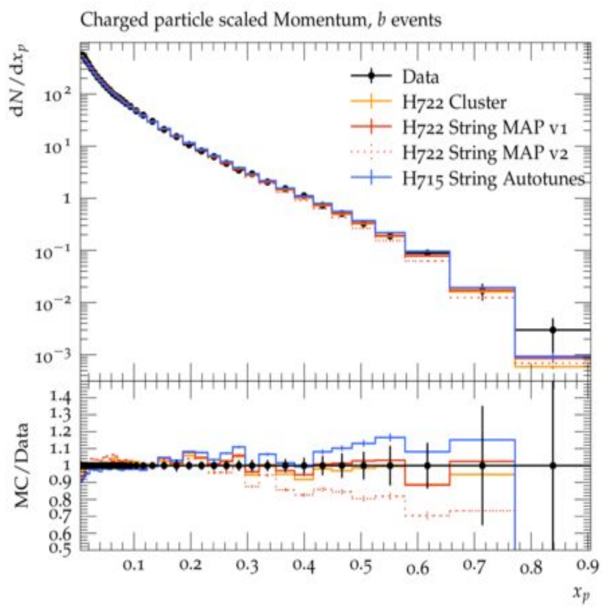


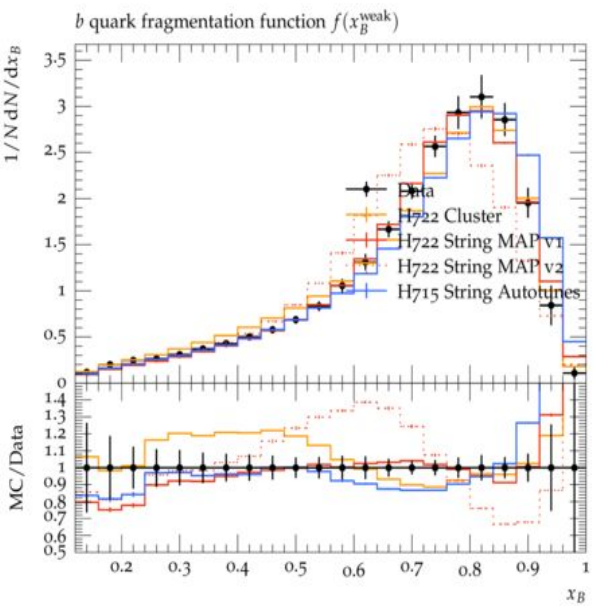
- These are light unflavoured probes of flavour dependent production.
- These distributions were **not** used in the tune (we will check if including them helps, if not need for model development?)
- It would also be interesting to see how Pythia does for the observables.
- The cluster model could be insensitive to the flavour as it does reasonably well in these distributions.

Flavour sensitive Observables

- Plots from the analysis SLD_2004_S5693039 (left and center) and SLD_2002_S4869273 (right).







- MAP tune does much better in these distributions than in previous flavour sensitive distributions

- MAP tune v1 and v2 shows much differences with v1 performing better over a wider range

Summary

- MAP Tune in general is better than AutoTunes tune for event shape distributions.

- MAP Tune does better in Jet observable distributions even though these are not weighted (used) in the tune.

- MAP tune v1 does better than MAP tune v2 in most regions and larger differences can be seen in flavour sensitive distributions therefore we recommend MAP tune v1 to use.

- Tensions in some flavour sensitive distributions may need model development rather than tuning systematics.

- The robustness of MAP tune now gives us a setup to use both strings and cluster for systematics studies of NP effects for e+e-
  - Quantify systematics EigenTune

- Next we would like to tune to LHC data - that would need some more developments to include Color Reconnection (work in progress)

Thank you!

Backup Slides

Parameter Values (MAP Tune v1)

Pythia8 parameters:

Fragmentation related Parameters

StringZ_aLund	0.750043
StringZ_bLund	0.898493
StringZ_sigma	0.309940
StringZ_aExtraSQuark	0.175699
StringZ_bExtraDiquark	0.053576
StringZ_rFactC	0.680046
StringZ_rFactB	1.265325

Flavour related Parameters

StringFlav_probStoUD	0.190037
StringFlav_probQQtoQ	0.079770
StringFlav_probSQtoQQ	0.998577
StringFlav_probQQ1toQQ0	0.022243
StringFlav_etaSup	0.512356
StringFlav_etaPrimeSup	0.184657
StringFlav_popcornRate	0.734028
StringFlav_mesonUDvector	0.329842
StringFlav_mesonSvector	0.676458
StringFlav_mesonCvector	1.065635
StringFlav_mesonBvector	1.849178

Herwig7 parameters:

/Herwig/Shower/AlphaQCD:AlphaIn	0.125772
/Herwig/Shower/NLOAlphaS:input_alpha_s	0.125772
/Herwig/Shower/PTCutOff:pTmin	1.027660

Parameter Values (MAP Tune v2)

Pythia8 parameters:

Fragmentation related Parameters

StringZ_aLund	0.394348
StringZ_bLund	0.688822
StringZ_sigma	0.308203
StringZ_aExtraSQuark	0.863354
StringZ_bExtraDiquark	1.901871
StringZ_rFactC	0.522541
StringZ_rFactB	1.577757

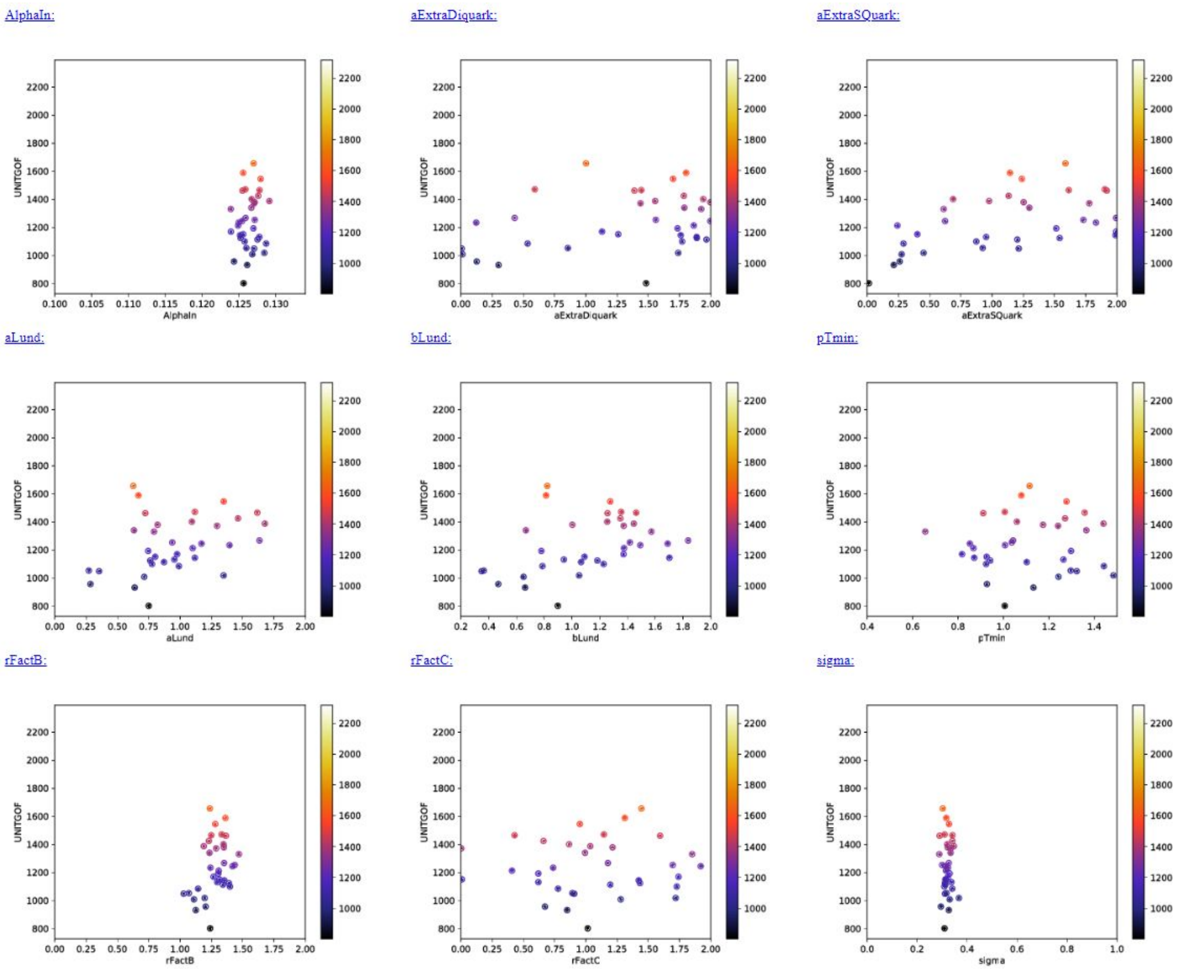
Flavour related Parameters

StringFlav_probStoUD	0.185213
StringFlav_probQQtoQ	0.076194
StringFlav_probSQtoQQ	0.998577
StringFlav_probQQ1toQQ0	0.063338
StringFlav_etaSup	0.537677
StringFlav_etaPrimeSup	0.135651
StringFlav_popcornRate	0.002145
StringFlav_mesonUDvector	0.330021
StringFlav_mesonSvector	0.688189
StringFlav_mesonCvector	1.151696
StringFlav_mesonBvector	2.257771

Herwig7 parameters:

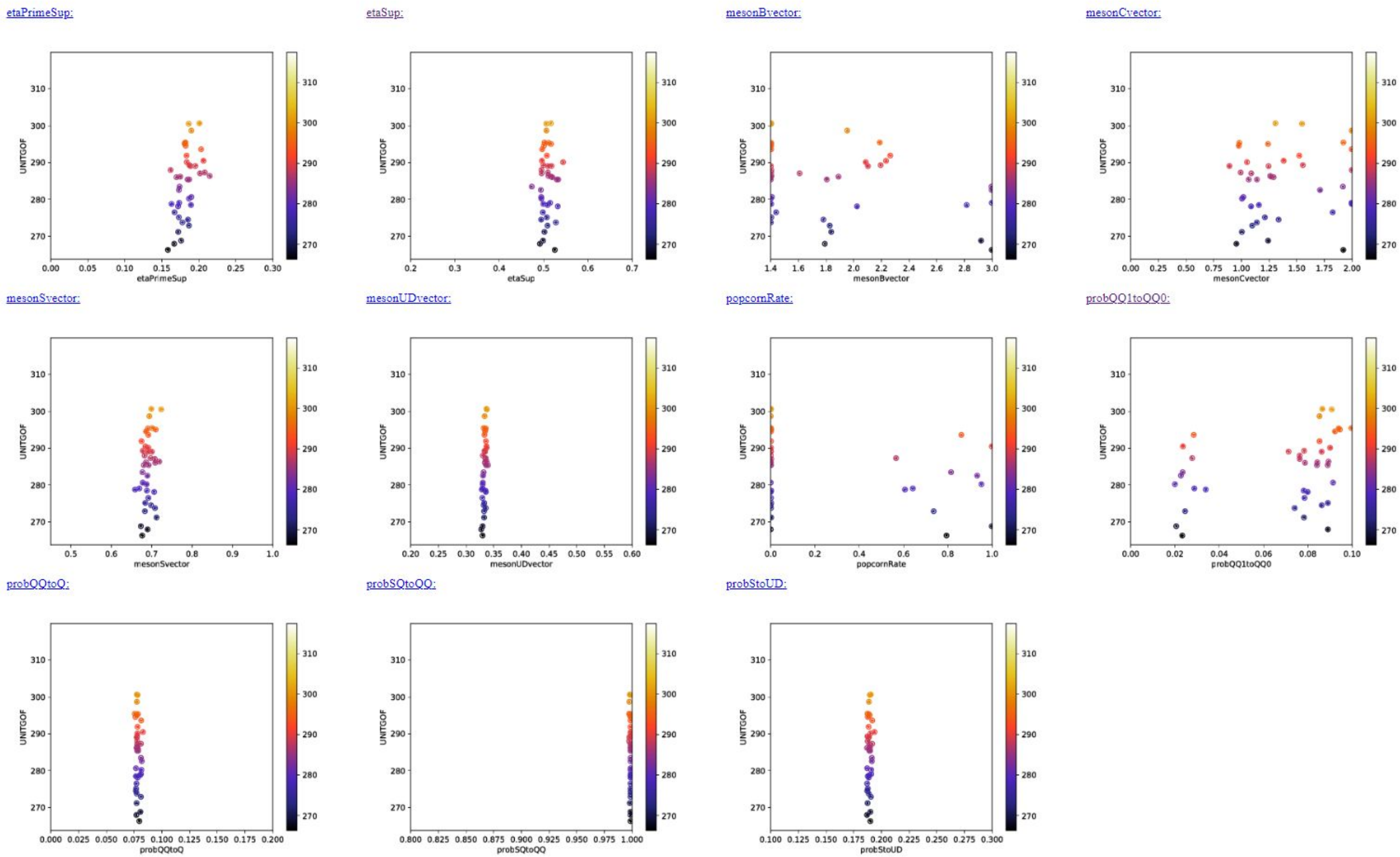
/Herwig/Shower/AlphaQCD:AlphaIn	0.124523
/Herwig/Shower/NLOAlphaS:input_alpha_s	0.124523
/Herwig/Shower/PTCutOff:pTmin	0.894570

Scatter plots (Fragmentation parameters)



- Parameters which are constrained in a narrow range are fixed and the tune is run again until maximum number of parameters are fixed.
- This reduces the number of parameters to be tuned after each step.

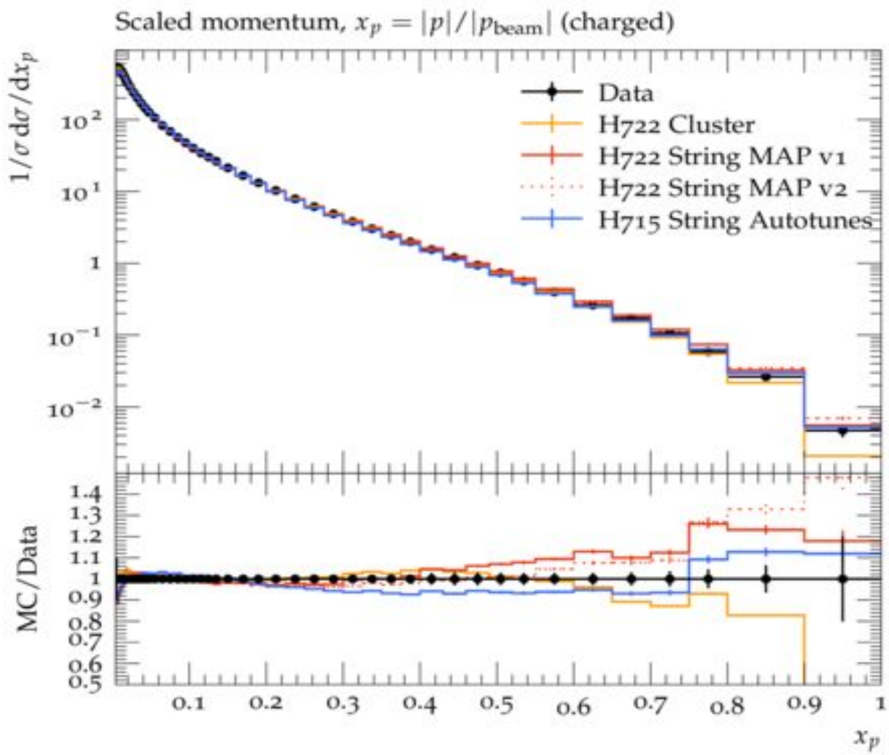
Scatter plots (Flavour Parameters)

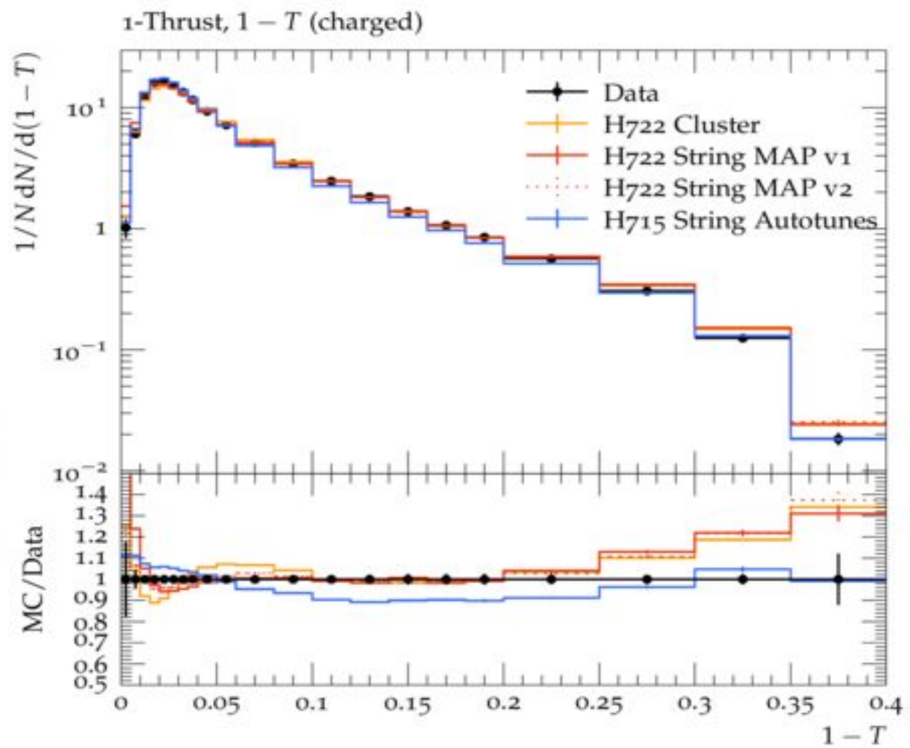


- Many of the flavour parameters are strongly constrained which are fixed and tuned.
- The probQQ1toQQ0 parameter shows interesting constraints near two values, but converges to the lower value for MAP tune v1 and the higher value for MAP tune v2 when other parameters are fixed.

Event Shapes (not used in tuning but doing well)

- Plots from the analysis ALEPH_1996_S3486095

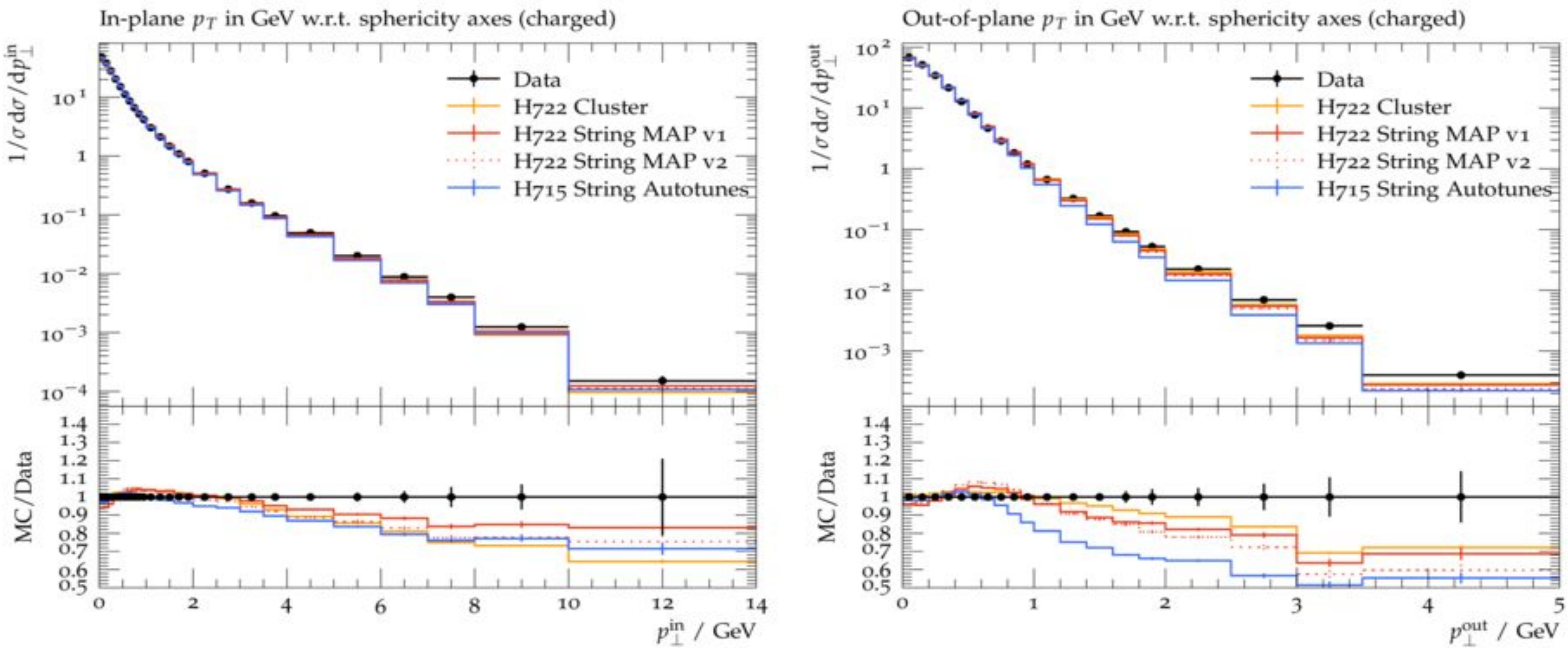




- Both these distributions were **not** used in tune.
- MAP tune does relatively well even though these distributions are no weighted in the tune
- For x_p distribution, MAP tune does better than the default but the AutoTunes is better near the tail.

Event Shapes (not used in tuning but doing well)

- Plots from the analysis ALEPH_1996_S3486095



- Both these distributions were **not** used in the tune.

- MAP tune shows small improvement towards the tail of these distributions (similar to distributions from DELPHI experiment in slide 9) even though these are not weighted in the tune.