INCLUSIVE JET NON-PERTURBATIVE CORRECTIONS

Ali Al Kadhim
Department of Physics
Florida State University
Tallahassee, FL
`aa18dg@fsu.edu`

September 11, 2022

ABSTRACT

We review the present literature, as well as present our preliminary results for the derivation of the Non-perturbative (NP) corrections for the inclusive jet cross section. All results are replicable and the code is open source on `https://github.com/AliAlkadhim/NPC_Jets`.

1 Introduction and Literature Review

The transverse momenta of the produced partons can be very small below the perturbative regime, and the total jet cross section can be dominated by non-perturbative (NP), or soft, components.

- **PRL 101, 062001 (2008), which is a D0 Measurement of Inclusive Jet Cross Section @ 1.96 TeV. They use Pythia QW tune with CTEQ6.5M PDFs. They say the following "These nonperturbative corrections to theory extend from +10% to +20% at p_T 50 GeV between $|y| < 0.4$ and $2.0 < |y| < 2.4$. The corrections are of order $+5\%$ for p_T 100 GeV, and smaller than $+2\%$ above 200 GeV."**
  Translation: They see the highest effects at very low p_T and in the lowest and highest rapidity bins, where the corrections are between 10-20 %. They show no plots

- **PRL 107, 132001 (2011), which is a CMS measurement of inclusive jet cross section @ 7TeV. They use PYTHIA 6.422 and HERWIG++ 2.4.2. The correction is defined as the average of the models, and the associated theoretical uncertainty is assumed to be half of the difference between the two predictions. For low-p T jets, the NP correction can be as large as 30%, with a relative uncertainty of 100%." They show no plots.**

- **Our analysis, from the Patrick Connor Note (AN-19-167-temp), where the corrections are shown in 2. They use PYTHIA8 with different tunes (CP1, CP5, 4C), HERWIG ++ and/or HERWIG 7, ans SHERPA. "In the context of the R scan analysis by Suman with 2016 data, NP correction factors extracted from POWHEG + PYTHIA (POWHEG + HERWIG ++) were found to be verysimilar to PYTHIA (HERWIG ++) standalone, and are not extracted here."**

- **DOI 10.1140, which is a CMS inclusive Jet cross section meassurement at 1 TeV. The plots post fitting are shown in 1, the paper is on `https://link.springer.com/content/pdf/10.1140/epjc/s10052-016-4286-3.pdf`.**

- **The familiar paper "Measurement and QCD analysis of double-differential inclusive jet cross sections in proton-proton collisions at $\sqrt{s} = 13$ TeV has the arxiv label "2111.10431". The data for this study can be found at `https://gitlab.cern.ch/fitters/xfitter-datafiles/-/tree/master/lhc/cms/jets/2111.10431` and the results are directly compared to ours.**
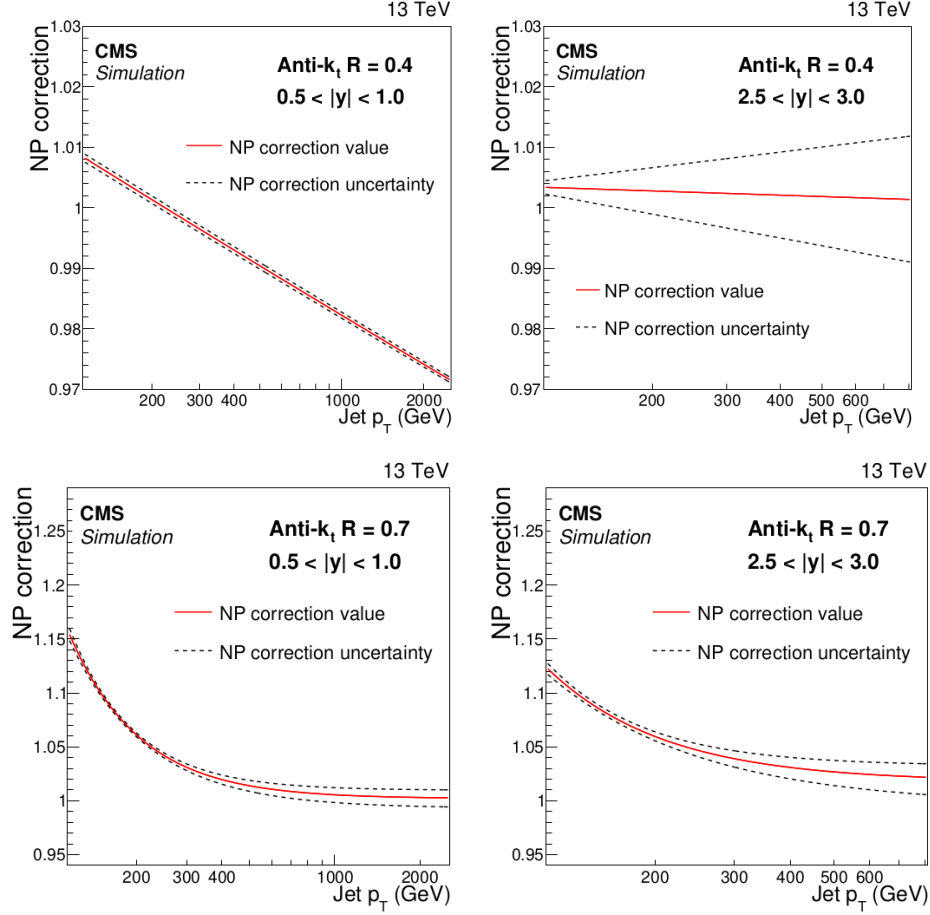
Figure 1: DOI 10.1140 (CMS Incl JEt @ 13 TeV). Pythia8 CUETS1-CTEQ6L1, CUETS1-HERAPDF, and CUETM1. Fit is power-law functin of jet p_T. `https://link.springer.com/content/pdf/10.1140/epjc/s10052-016-4286-3.pdf`

2 POWHEG-BOX

Dijet production is by far the most frequently-occuring of all hard scattering processes in hadronic collisions. We can use the threshold for the Dijet production at the Born level in POWHEG-BOX with the varibale `bornktmin` , which is the generation cut of the dijet system at the Born level, i.e. before the third parton (synce it's at NNLO) is emmitted. This is relevant to the measurement of the inclusive jet cross section since it allows one to make a cut on the dijet system, say at 100 GeV, and make an efficient estimation of the total cross section at, sat, 200 GeV.

In Powheg each event is built first by producing the underlying Born configuration, here a QCD 2-> 2 scattering. They use the p_T of the underying Born configuration as the $\mu_R = \mu_F$ in obtaining fixed order NLO predictions.

The cross section falls very sharply as the transverse momentum of the leading jet increases

$$\sigma(p_{T\ jet\ 1}) \propto p_{T,\ jet\ 1}^{-6} \tag{1}$$

Therefore if we generate events with eual weights, the high p_T regions will not be populated. With a positive $k_{T,supp}$ factor, the cross section is weighted such that

$$\sigma_{weighted}(k_T) = \mathcal{S}(k_T)\,\sigma(k_T) = \left(\frac{k_T^2}{k_T^2 + k_{T,supp}^2}\right)^3 \sigma(k_T) \tag{2}$$

(a) $R = 0.4$

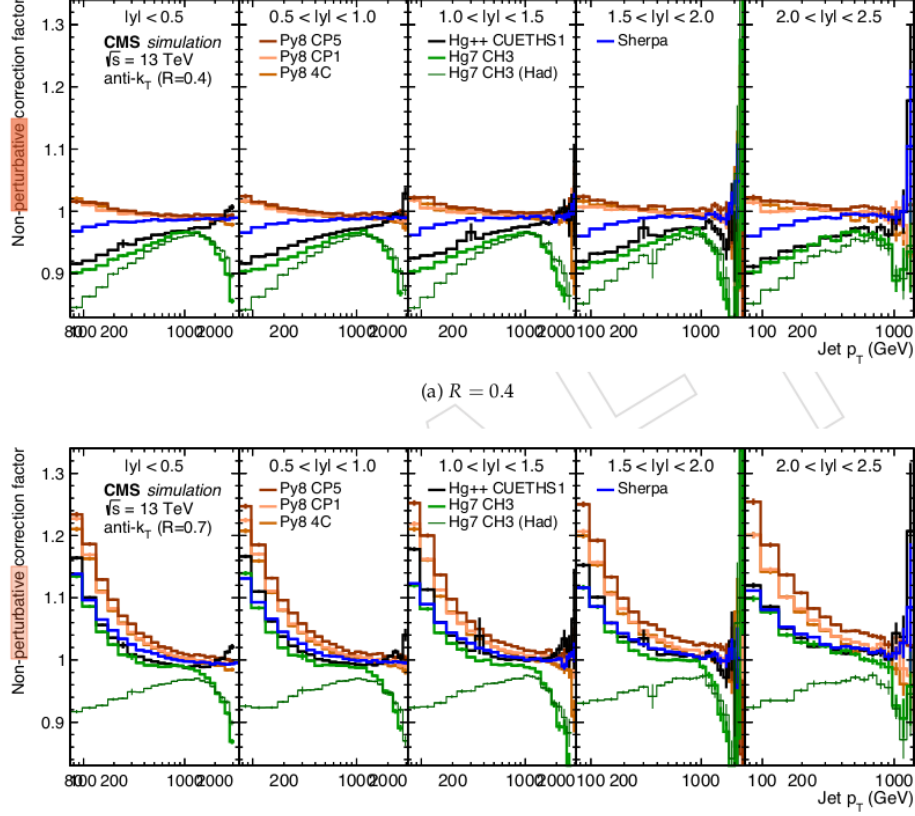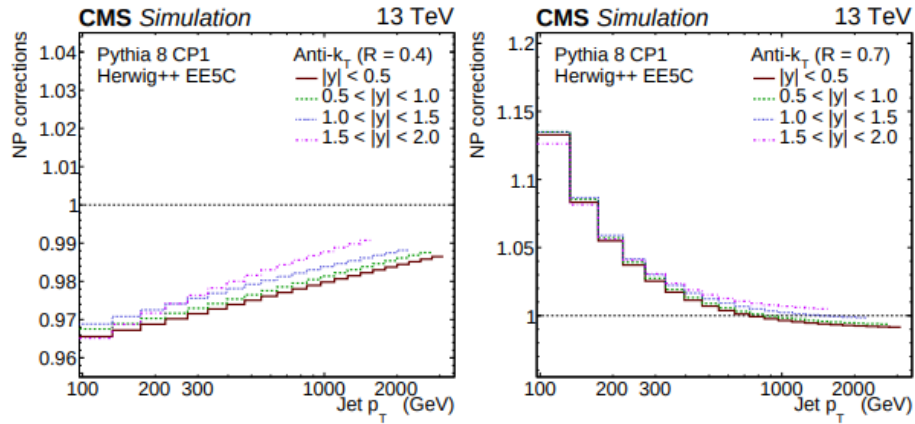

Figure 2: Our analysis, from the Patrick Connor Note (AN-19-167-temp)

Figure 3: The familiar CMS paper https://arxiv.org/pdf/2111.10431.pdf. They use Pythia Tune: CUETP8M1-NNPDF2.3LO, and fit a smooth function $a_0 + a_1/p_T^{a_2}$.

Where bornsuppfact=$k_{T,supp}$. Essentially events are weighted according to weight. To learn more about this, please visit the POWHEG-BOX Dijet paper `https://link.springer.com/content/pdf/10.1007/JHEP04(2011)081.pdf`.[1]

The k T dependence of the suppression factor in the above equation is such that the generation of low transverse momentum events is relatively damped, while the whole transverse momentum region is nearly uniformly populated up to momenta of the order of k T,supp .

In PYTHIA, the p_T of MPI scattering is set to be quite high, mich above the scale of the hard process under examination. In jet production this may lead to overcounting. PYTHIA inhibits this behavior when *it* generates jets, by limiting the MPI scale to that of the jets of the primary interaction. But when interfacing with pythia through POWHEG, there is no way for pythia to know that it is showering a jet process, therefore the user should tell PYTHIA to limit the scale of the MPI to the hardness of the primary interaction: this can be done with `MSTP(86)=1`.

For the POWHEG-BOX generation of les Houches events, we use PDF set `NNPDF31_nnlo_as_0118` .

2.1 Showering, Pythia, Tunes

The next step in the chain of derivation of NP corrections is using parton showering programs. We use Pythia 8.306, and later we plan to use Herwig7 for the showing, since they have different showring models, and it might be good to compare them as other papers have done. As of Pythia 8.2, the default Pythia tune is Monash 2013.

Different tunes can be changed in pythia by simply doing "TUNE:pp =N", where N corresponds to the tune, which can be found in the "Tunes" page of `https://pythia.org/latest-manual/Welcome.html`. See a discussion on the CMS Pythia tunes here `https://arxiv.org/pdf/1512.00815.pdf`.

- We use Pythia Tune `CUETP8M1-NNPDF2.3LO`, since that is the standard one for CMS that others use, so that we can directly compare with the NPCs plotted in the literature review. Our results for one of the slices with this Tune is in Fig 6.
- We should also use We use Tune `CUETP8S1-CTEQ6L1`, but somehow that is currently not working on naf.
- We should also use `CUETP8S1-HERAPDF1.5LO`
- We should also shower with Herwig 7.

3 Results

$$\mathrm{C}^{\mathrm{NP}} = \frac{d^2\sigma_{\mathrm{MC}}^{\mathrm{PS+MPI+HAD}}/dp_T dy}{d^2\sigma_{\mathrm{MC}}^{\mathrm{PS}}/dp_T dy} \tag{3}$$

Or, for simplicity,

$$CNP(p_T^{jet}) = \frac{\sigma_{PS+MPI+HAD}}{\sigma_{PS}}. \tag{4}$$

The uncertainty in the NP correction factors $\Delta_{C_{NP}}$ were propagated from the uncertainties in the post-hadronization and pre-hadronization cross sections, assuming the post and pre-hadronization cross sections are uncorrelated (i.e. their covariance matrix has zero off-diagonal elements)

$$\Delta_{NP} = \sqrt{\left(\frac{\partial CNP}{\partial \sigma_{PS}}\Delta_{\sigma_{PS}}\right)^2 + \left(\frac{\partial CNP}{\partial \sigma_{PS+MPI+HAD}}\Delta_{\sigma_{PS+MPI+HAD}}\right)^2} \tag{5}$$

$$= \frac{\sigma_{PS+MPI+HAD}}{\sigma_{PS}}\sqrt{\left(\frac{\Delta_{\sigma_{PS+MPI+HAD}}}{\sigma_{PS+MPI+HAD}}\right)^2 + \left(\frac{\Delta_{\sigma_{PS}}}{\sigma_{PS}}\right)^2}$$

3.1 Individual Slices

Many Individual "slices" were computed, where a slice denotes a choice of the pair (k_T^{supp}, k_T^{min}).

A few individual slices are presented in the figure, for illustration purposes. Notice that some (k_T^{supp}, k_T^{min}) combinations yield very unreasonable resolts, as in the $(0, 10)$ combination, whereas others yield more uniform factors around 1.0, such as the $(11000, 1250)$ slices. and others fluctuate around between 0.9 and 1.1, such as the $(800, 600)$ slices.
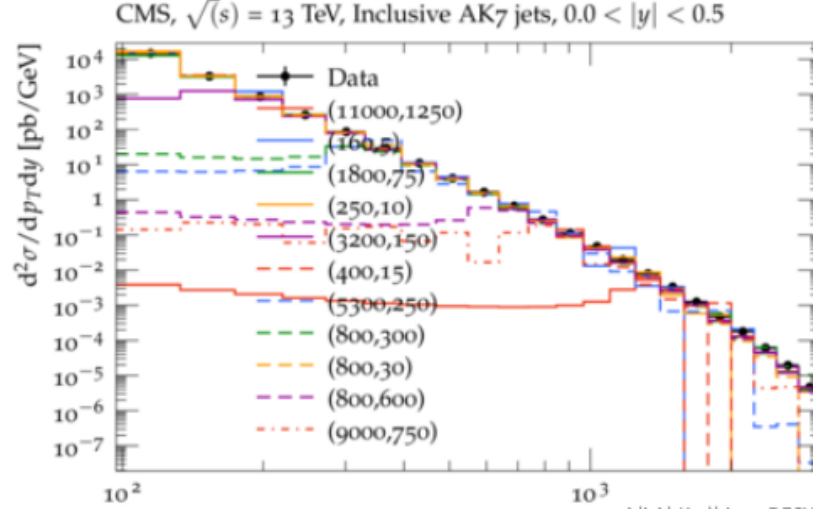
Figure 4: The differential cross section evaluated with 12 total (k_T^{supp}, k_T^{min}) "slices". We see that the slices are mostly doing what they are supposed to do, i.e. populate the spectrum at different ranges in p_T^{jet}. This figure is also useful in knowing which ranges in p_T^{jet} the different slices are useful. For example, it is clear to see that the $(k_T^{supp}, k_T^{min}) = (11000, 1250)$ slice is flat in p_T^{jet} until a very high value, ≈ 2 TeV, where it starts being very close to data. This means that this slice is only useful in the range $p_T^{jet} \geq 2$ TeV.

3.2 Patched NPC

The $CNP(p_T^{jet})$ was made by patching together different C_{NP} based on different values of (k_T^{supp}, k_T^{min}) to target different regions of the p_T^{jet}, and combined based on the the correction factors that yielded the lowest uncertainty in that p_T range (i.e. $C_{NP} = \sum_i \sim_{(k_T^{supp}, k_T^{min})} w_i^{(k_T^{supp}, k_T^{min})} C_{NP,i}^{(k_T^{supp}, k_T^{min})}$ where $w_i^{(k_T^{supp}, k_T^{min})} \propto 1/\Delta_{C_{NP},i}$ and i indexes the p_T bins).

4 The problem of large fluctuations

As you can see in the patched NPCs above, there is large fluctuations (esp at low p_T) even with 1B events. Furthermore, there is much ambiguity about why certain slices yield large fluctuations, why some slices result is something completely unreasonable, and how best to combine all these samples to one spectrum. One should note that my slices with the least uncertanites on the NPCs are plotted in Fig., and we note that they still show considerable fluctuations (even with 10^9 events!!). Furthermore, other groups have noted that you may need several billion events in order to get reasonable results. There are two possibilities here: either

1. Case 1: This is not a problem, and it is expected. One could argue that this should not seem like such an unexpected problem, since we have little published data to go off of, as most publications report the NP corrections after they fit it with a smooth function.

   Case 2: This is a legitimate problem. In that case, we should try to find the root of the problem, by isolating each part of the chain, and running tests. Since there are so many steps and software components to this measurement, the source of the issue could come from one of the steps in the chain. The cause can also be something more fundamental. Each possibility of failure along this chain should be investigated. The possible causes enumerate and manifest themselves themselves in the following manner:

   (a) **It is a problem with the (bornsuppfactor, bornktmin). A sample in Fig. 5 was done to address this. As we see, defining them as zero exaserbates the problem.**

   (b) **It is a problem with POWHEG alltogether. A sample to address this, was generated with Pythia standalone to address this. This was done in "two runs", once with MPI/HAD on and once again with them off. (`https://github.com/AliAlkadhim/NPC_Jets/blob/master/pythia_tutorials/pythia_stanalone_NP_tworuns_pre.cc`).**

(c) **It is a problem with the yoda/rivet chain. The workflow is: rivet -> Fifo -> yoda. and the yoda files are then combined with `yodamerge_tmp.py` to two `.yoda` files, one fore post and one for pre hadronization. Then `rivet-cmphistos --mc-errs <file1.yoda> <file2.yoda>` was called to compare the two files and generate a `.dat` file for each rapidity bin. However, the numbers in the individual .dat files often differ than the numbers assoiated with the same histograms in the `.yoda` file. There is then ambiguity whether one should use the `.yoda` file or the `.dat` files.**

(d) **It relates to calculation of errors and their propagation to the ratio. A calculation was done with ROOT, and `yoda2root` executable.**

This implies that we are all doing something fundamentally wrong, because we shouldn't need more than hundreds of thousands of counts per bin in order to get good fluctuations/uncertainties.

4.1 ROOT and yoda2root

see the `.yoda` files were converted to ROOT files with `yoda2root` such that the histograms can be used right a way from the root file.

4.2 Weighted Average of Slices

A note that was raised by Harrison Prosper is that We should treat this as analogous to the problem of triggers, where we have no biasing of the data at all. It is analogous in the sense that the data has different thresholds, all the events have different weights, and we just patch them together. One way could be to use a weighted average, such that the complete $CNP(p_T; y)$ is the a weighted sum of the correction factors of each p_T bin, such that there is no biasing of the different slices. The correction factor at one rapidity bin y and a P_T bin can be written as

$$\mathbf{CNP}^{p_T \ bin}(p_T^{jet}) = \sum_{i=1}^{N_{slices}} w_i \ \mathbf{CNP}_i^{p_T \ bin}(p_T^{jet}) \tag{6}$$

Where the weight of slice i at the same p_T and y bins is

$$w_i = \frac{1/\sigma_i^2}{\sum_{i=1}^{N_{slices}} 1/\sigma_i^2} \tag{7}$$

where σ is the variance, which can be approximated as the relative uncertainty of the correction factor Δ_{NP} (at a given p_T and y bin). Although this would not bias any slice more than another, there is ambiguity on which slices to include or exclude and why.

4.3 Pythia Standalone

To assess whether the problem is with POWHEG alltogether, as outlined in the possible sources of error, the NP corrections were derived with Pythia8 standalone and is shown in Fig. 10, using 10^6 events, via the "two runs", i.e. once with MPI and Hadronization turned off and once again with them turned on. The samples were generated with `https://github.com/AliAlkadhim/NPC_Jets/blob/master/pythia_tutorials/pythia_stanalone_NP_tworuns_pre.cc` and analyzed with `https://github.com/AliAlkadhim/NPC_Jets/blob/master/pythia_tutorials/analyze_tworuns_NP.py`.

4.4 Effective Count

Suppose we have a histogram of m bins. If it is an unweighted histogram, the distribution of the number of events for bins of the histogram is multinomial and the probability for a random vectory $n_1, ..., n_m$ is

$$P^{uw}(n_1, \ldots, n_m) = \frac{n!}{n_1! n_2! \ldots n_m!} p_1^{n_1} \cdots p_m^{n_m} \tag{8}$$

And satisfying $\sum_{i=1}^{m} p_i = 1.$, and p_i is the probability that a random event belongs to bin i for a given PDF $p(x)$:

$$p_i = \int_{S_i} p(x) dx, i = 1, \ldots, m \tag{9}$$

This is because each bin has a count that is poisson-distributed $Poiss(n; \lambda) = \frac{e^{-\lambda}\lambda^n}{n!}$, (and $\mu_{poiss} = E[n] = \lambda$ and $\sigma^2_{poiss} = E[(n - \mu_{poiss})^2] = \lambda$).

If a histogram is weighted, then the distribution for the number of entries follows another multinomial

$$P(n_1, \ldots, n_m) = \frac{n!}{n_1! n_2! \ldots n_m!} g_1^{n_1} \cdots g_m^{n_m} \tag{10}$$

Satisfying $\sum_{i=1}^{m} g_i = 1$ where g_i approximates another PDF

$$g_i = \int_{S_i} g(x) dx, i = 1, \ldots, m \tag{11}$$

then the total sum of weights of events (entires) in the ith bin, W_i, $i = 1, ..., m$ is

$$W_i = \sum_{j=1}^{n_j} w_i(j), \tag{12}$$

where n_i is the number of entires in bin i.

The number of counts in bin i is the same as the sum of weights

$$counts_i = W_i = \sum_{j=1}^{n_j} w_i(j) \tag{13}$$

From here we can derive the uncertainty, or variance for bin i (which has $count_i$)

$$Var(count_i) = var\left(\sum_{j=1}^{n_j} w_i(j)\right)$$

$= \sum_{j=1}^{n_j} var\left(w_i(j)\right)$

$= \sum_{j=1}^{n_j} var\left(w_i(j) \times Poiss(\lambda = 1)\right) \quad Because we have one measurement with constant weight$

$= \sum_{j=1}^{n_j} w_i^2(j) var\left(Poiss(\lambda = 1)\right) \quad because var(const.Xx) = xonst^2 var(X)$
$= \sum_{j=1}^{n_j} w_i^2(j)$

(14)

One of the problems that could give rise to this is that the number of counts of the cross sections in each bin is not what we expect them to be. Since we are using 10^9 events for each slice, we expect to see hundreds of thousands of counts in each bin, which means that the relative uncertainty at low p_T should not be as high as the plots in Fig. 8 indicate. We can approximate the effective counts per bin by a consideration of the weights and uncertainties per bin. We can also simply plot a distribution of the weights. If it has a long tail (ie large variance), then one could easily destroy the precision of the data. This is because the standard deviation per bin, σ, is given as the sum of weights for that bin $\sigma^2 = \sum w_i^2$ and the mean per bin, $\mu = \sum w_i$. We can then effective count N_{eff} from

$$\frac{\sigma}{\mu} = \frac{1}{\sqrt{N_{effective}}} \Rightarrow N_{eff} = \frac{\mu^2}{\sigma_2} = \frac{\sum w_i^2}{\sum w_i}. \tag{15}$$

The effective counts based on the equation above can be seen in Fig. 11.

5   Presentations

- https://indico.cern.ch/event/1184925/contributions/4978573/attachments/2484854/4266334/NPC_Update_Jul25.pdf

6 Planed Studies

6.1 Seeing the impact of the NP Correction Variations on xFitter Fits of α_S

### 6.2 The corrections as a 2-d transfer matrix

6.3 Defining our own Pythia Tune, which targets hadronization parameters, and doing variations on the parameters

6.4 Complete Likelihood fit of pythia parameters $\vec{\theta}$ and propagating them to the cross section $\sigma(\vec{\theta})$
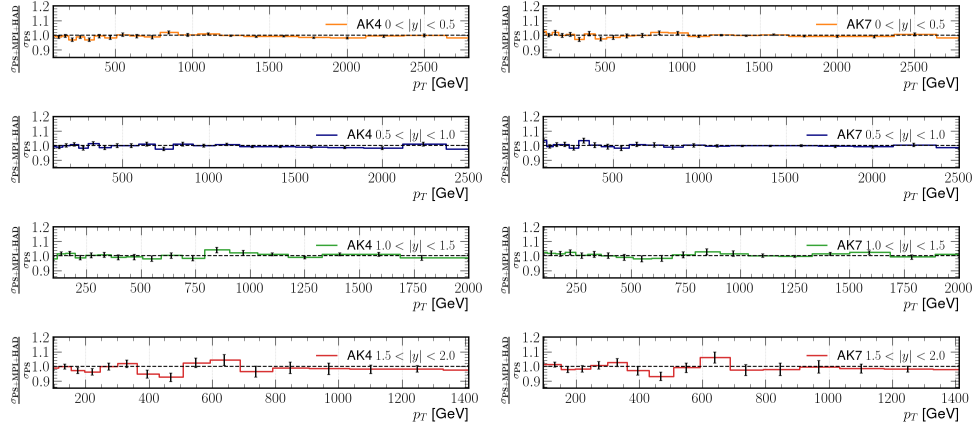
References

[1] Alioli, S., Hamilton, K., Nason, P. et al. Jet pair production in POWHEG. J. High Energ. Phys. 2011, 81 (2011). https://doi.org/10.1007/JHEP04(2011)081

suppr=0, ktmin=10, 500M events, Monash2013
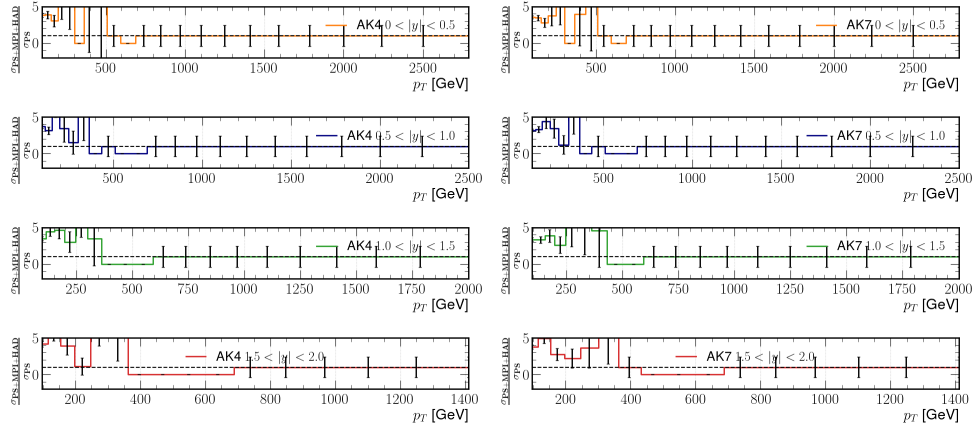
bornktmin 1,250, bornsuppfact 11,000, MSTP(86)=1, 1B events

$(800, 600)$, 10^9 events, Monash2013
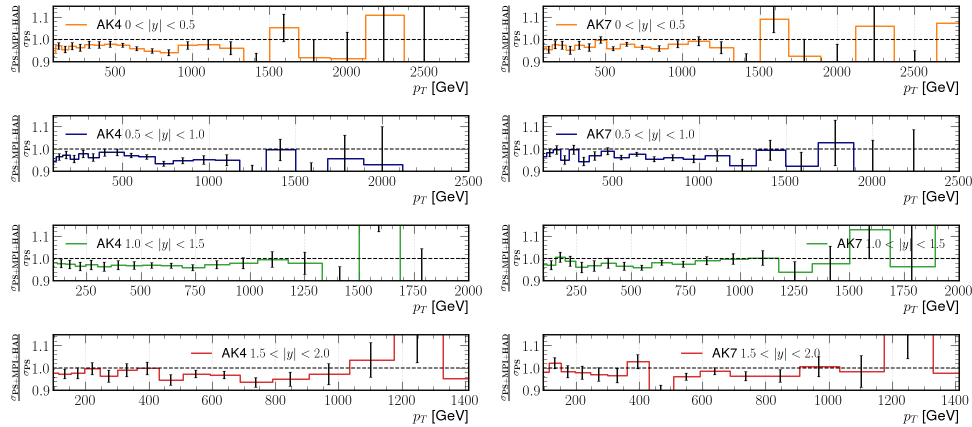
Figure 5: The NP corrections for some individual example slices using the Default Pythia Tune Monash2013. Different choices of slices lead to varying levels of fluctuations and uncertainties. With these plots, it is important to note that the bornktmin is a cut on the Dijet system, therefore in the mid plot, the cut is 1,250 GeV, meaning that even though it looks smooth at low p_T, these are not the inclusive jets we want. The cuts only work in the region where $p_T^{jet} \geq k_T^{min}$

9
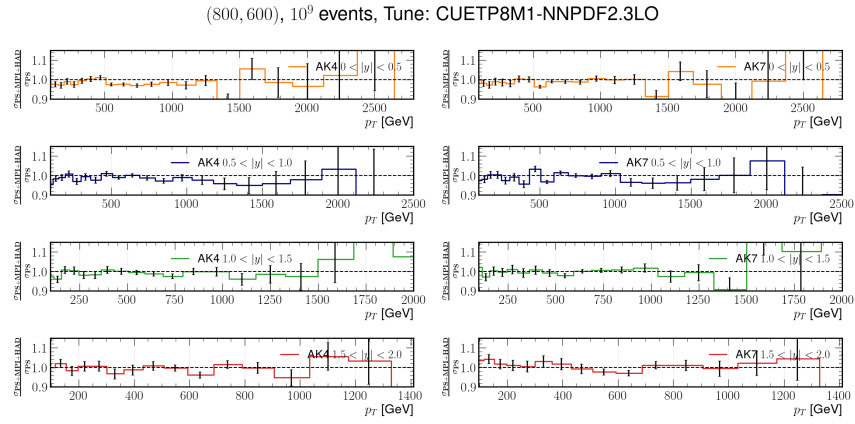
Figure 6: The $(800, 600)$ Slices, showered with Pythia's popular CMS Tune CUETP8M1-NNPDF2.3LO.
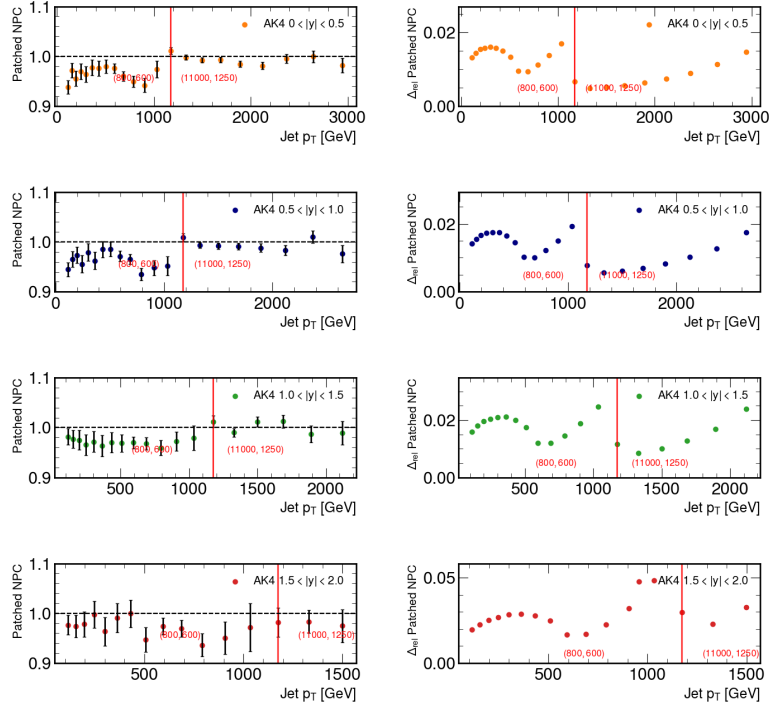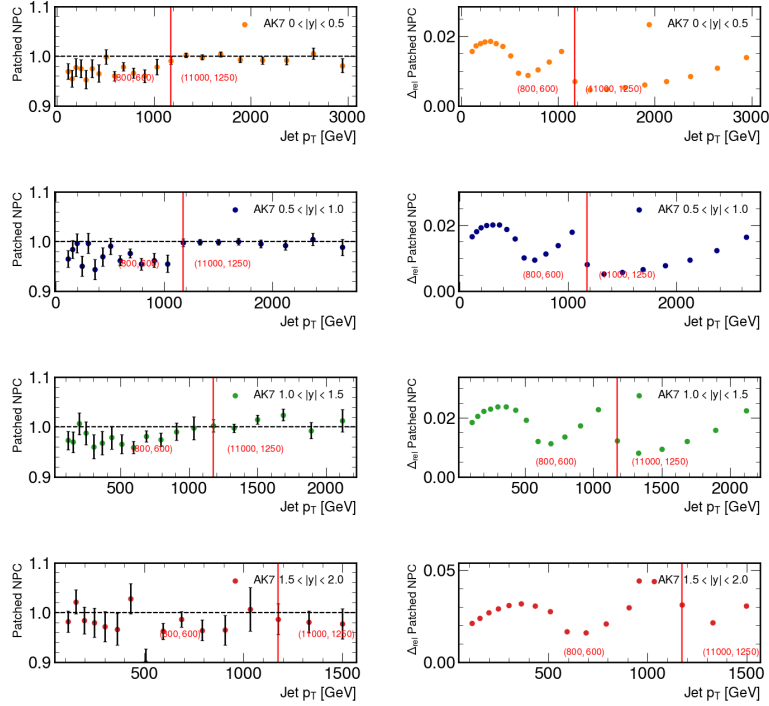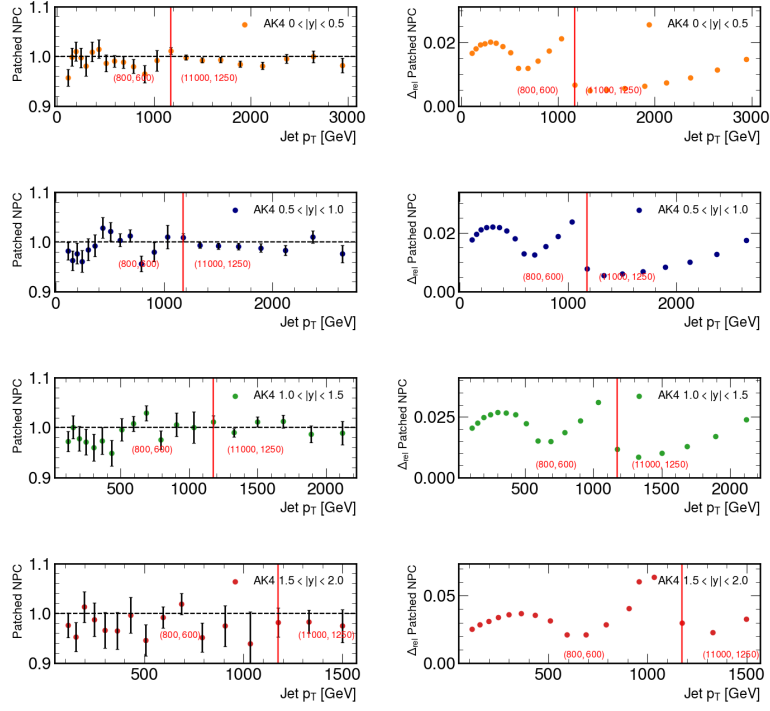
Figure 7: "Patched" Non-perturbative corrections based on two slices $(k_T^{supp}, k_T^{min}) = (800, 600)$ for low p_T and $(11000, 1250)$ for high p_T. The relative uncertainties on the right plots are simply Δ_{NP}/NPC from Eq. 5.
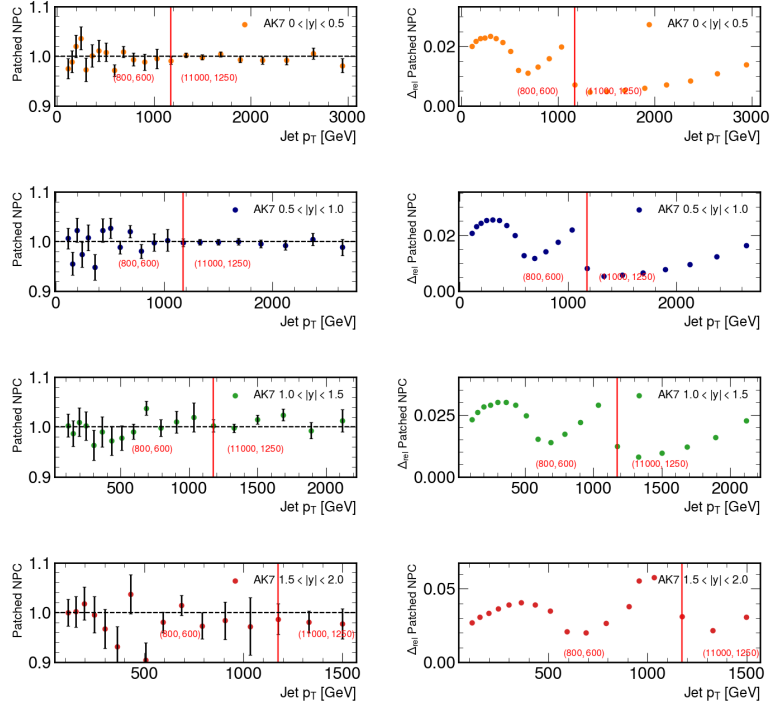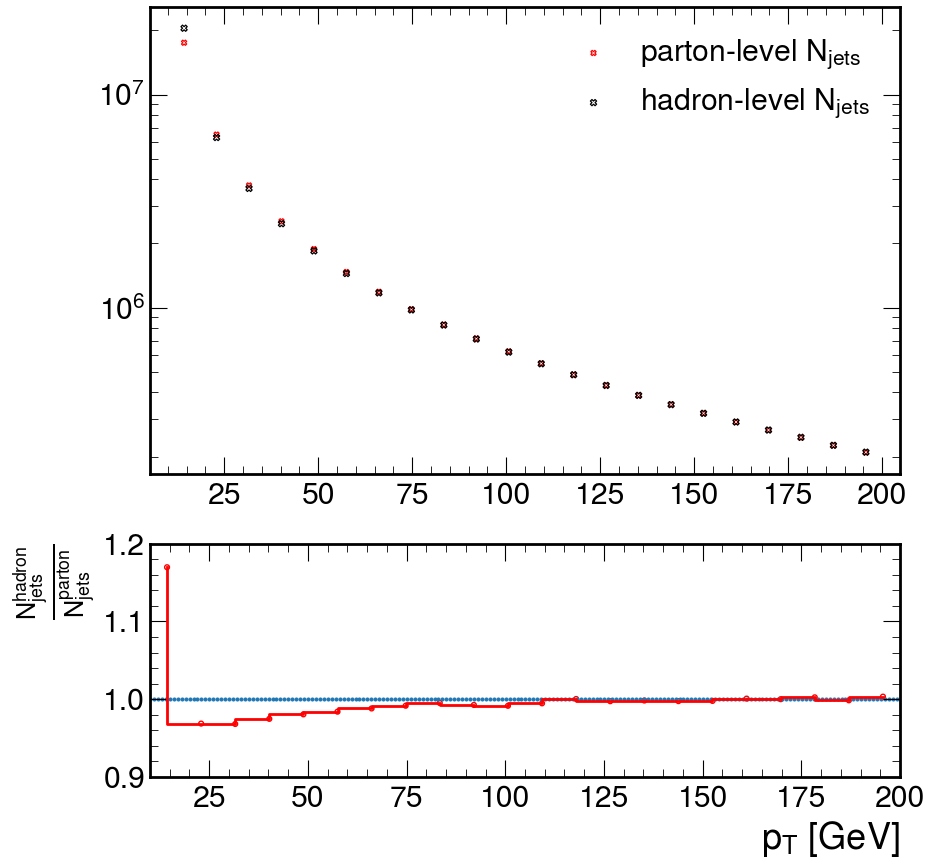
11

Figure 8:

Figure 9:

Figure 10: NP Corrections with Pythia Standalone with 10^7 events, via the "two runs" method, i.e. once with MPI and Hadronization turned off and once again with them turned on. There is no cuts in rapidity. The samples were generated with `https://github.com/AliAlkadhim/NPC_Jets/blob/master/pythia_tutorials/pythia_stanalone_NP_tworuns_pre.cc` and analyzed with `https://github.com/AliAlkadhim/NPC_Jets/blob/master/pythia_tutorials/analyze_tworuns_NP.py`.
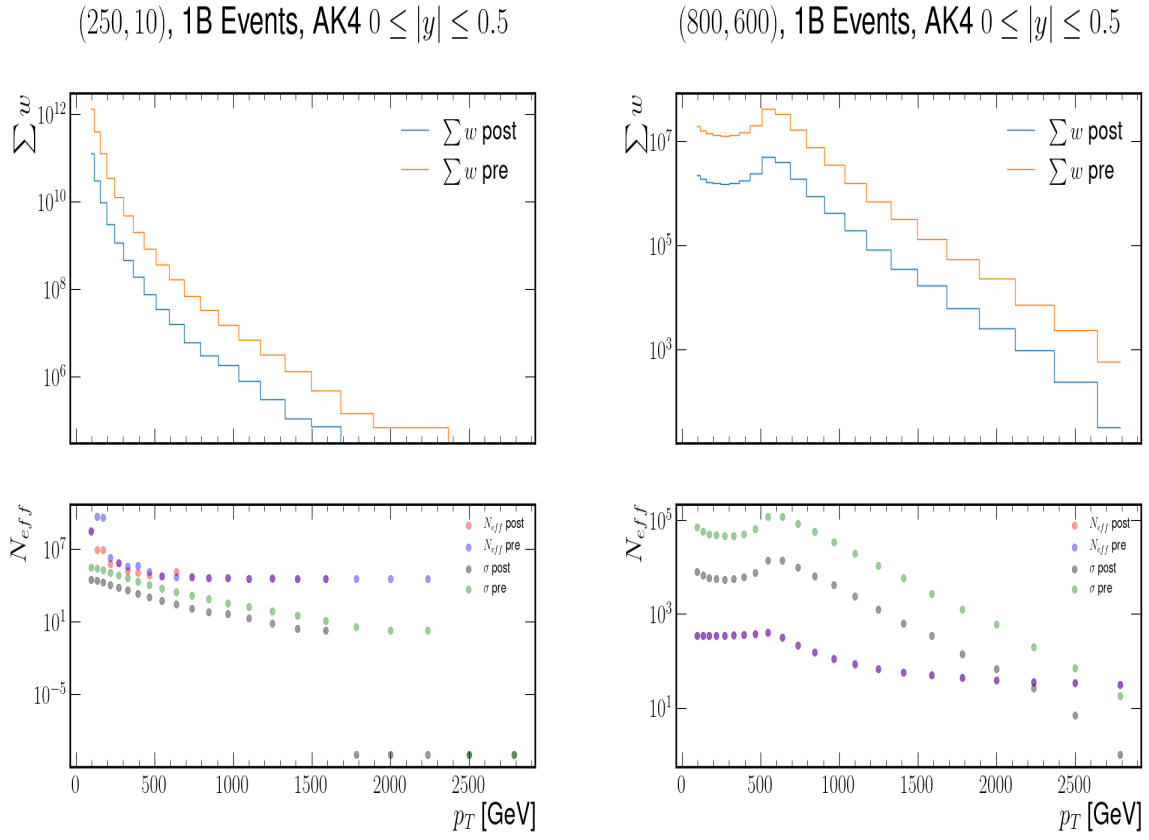
Figure 11: Sum of weights per bin $\sum w$ and effective counts N_{eff}, based on Eq. 15.