Validation of bias weighting

A. de Wit

HELMHOLTZ RESEARCH FOR GRAND CHALLENGES

Introduction

- Bias weight function in aMC@NLO allows us to generate more events in a particular area of phase space of an inclusive sample (at LHE-level)
  - Saves having to stitch e.g. inclusive and p_T binned samples together
- We want to apply this to DY+up to 2 jets (and W+up to 2 jets)
  - Start validation with DY+ up to 1 jet as this can be run locally (ie DY +0 and DY+1)
- In these slides: validation with a total of 100k LHE events for several scenarios*:
  - **No bias weight** (unbiased, default aMC@NLO production)
  - **Biased, function 1:** $(25+(Z\ p_T)^2)*10^{njets}$ (This from the example bias weight cards)
    - As we will see this function really weights down the 0-jet events so much that the uncertainties on these events become very large. Try to mitigate this by:
  - **Biased, function 2:** $1000 + (Z\ p_T)^2*10^{njets}$
  - **Biased, function 3:** $1000 + [(Z\ p_T)^3*10^{njets}]/1000$

*cards:

unbiased: https://github.com/adewit/genproductions/tree/bias-cards/bin/MadGraph5_aMCatNLO/cards/examples/dyellell01j_5f_NLO_FXFX
bias function 1: https://github.com/adewit/genproductions/tree/bias-cards/bin/MadGraph5_aMCatNLO/cards/examples/dyellell01j_5f_NLO_FXFX_bias
bias function 2: https://github.com/adewit/genproductions/tree/bias-cards/bin/MadGraph5_aMCatNLO/cards/examples/dyellell01j_5f_NLO_FXFX_bias_2
bias function 3: https://github.com/adewit/genproductions/tree/bias-cards/bin/MadGraph5_aMCatNLO/cards/examples/dyellell01j_5f_NLO_FXFX_bias_3

Settings

- **Note:** I reduced the integration grid accuracy from the default to speed the process up a bit. Should not affect the global picture we get from these slides
- **PYTHIA fragment:**

```
 processParameters = cms.vstring(
        'JetMatching:setMad = off',
        'JetMatching:scheme = 1',
        'JetMatching:merge = on',
        'JetMatching:jetAlgorithm = 2',
        'JetMatching:etaJetMax = 999.',
        'JetMatching:coneRadius = 1.',
        'JetMatching:slowJetPower = 1',
        'JetMatching:qCut = 30.', #this is the actual merging scale
        'JetMatching:doFxFx = on',
        'JetMatching:qCutME = 10.',#this must match the ptj cut in the lhe generation step
        'JetMatching:nQmatch = 5', #4 corresponds to 4-flavour scheme (no matching of b-quarks), 5 for 5-flavour
scheme
        'JetMatching:nJetMax = 1', #number of partons in born matrix element for highest multiplicity
      ),
```

- I realise “TimeShower:mMaxGamma = 4.0” is missing from the fragment. Will fix this for future studies.

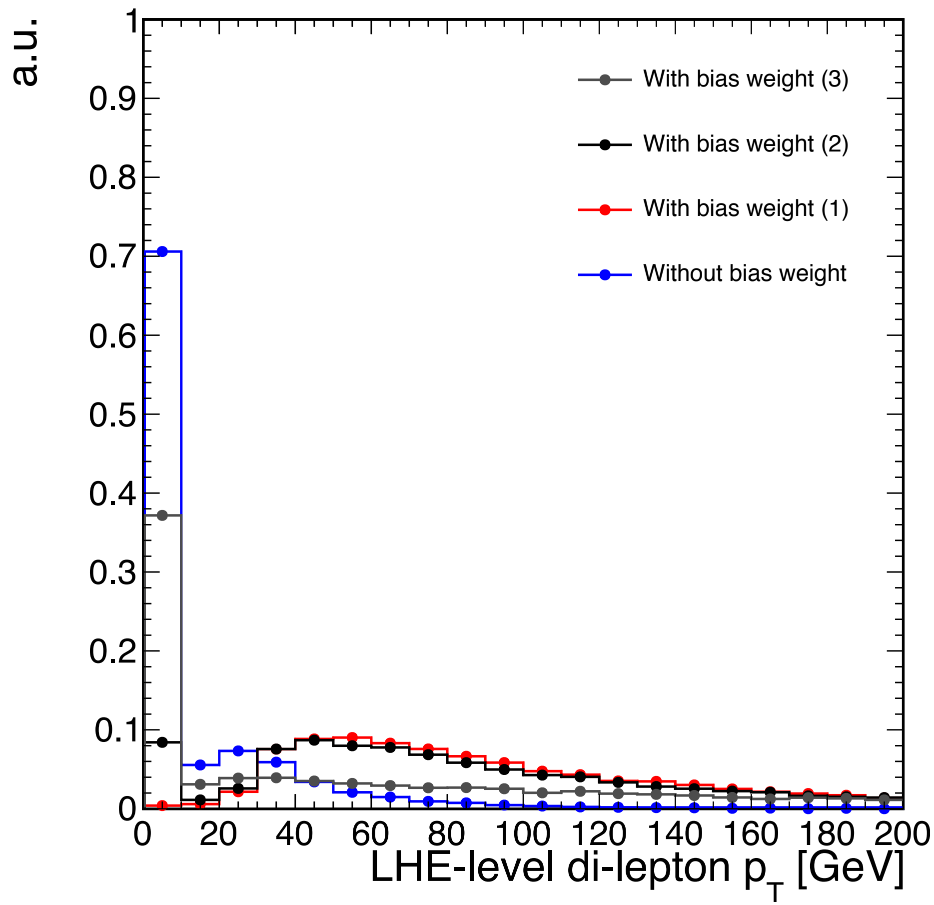
PYTHIA shower matching efficiency

- Before comparing distributions, have a look at the matching efficiency reported by PYTHIA

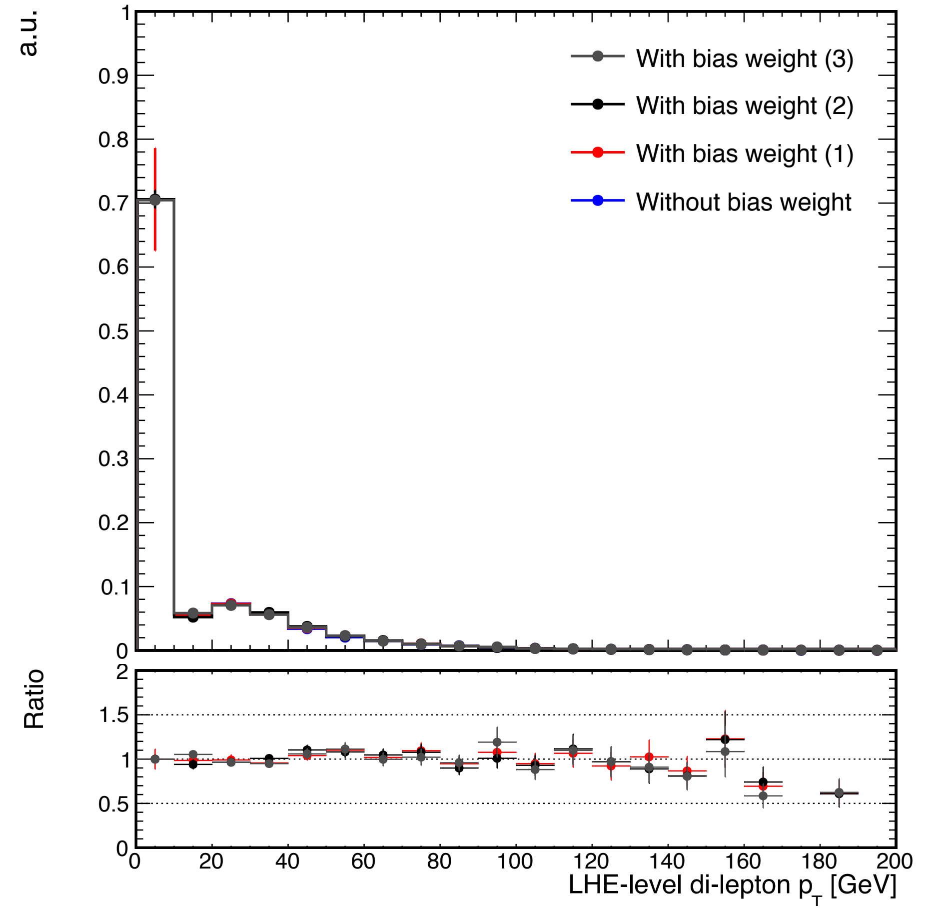
	N_{evts} tried	N_{evts} accepted	N_{evts} tried, 0-jet	N_{evts} accepted, 0-jet	N_{evts} tried, 1-jet	N_{evts} accepted, 1-jet
No bias weight	100000	51836	42230	36901	57770	14935
Function 1	100000	74147	3239	635	96761	73512
Function 2	100000	72419	7945	4981	92055	67438
Function 3	100000	66347	26580	22117	53130	44230

Function 1 leads to a vast reduction in the number of 0-jet events at LHE level. The other two functions also reduce the number of 0-jet events (expected as LHE Z p_T is 0 in such events) - but due to larger constant term in the function the reduction isn't as pronounced. This is important for the uncertainties on 0-jet events.

LHE-level comparisons: Z p_T



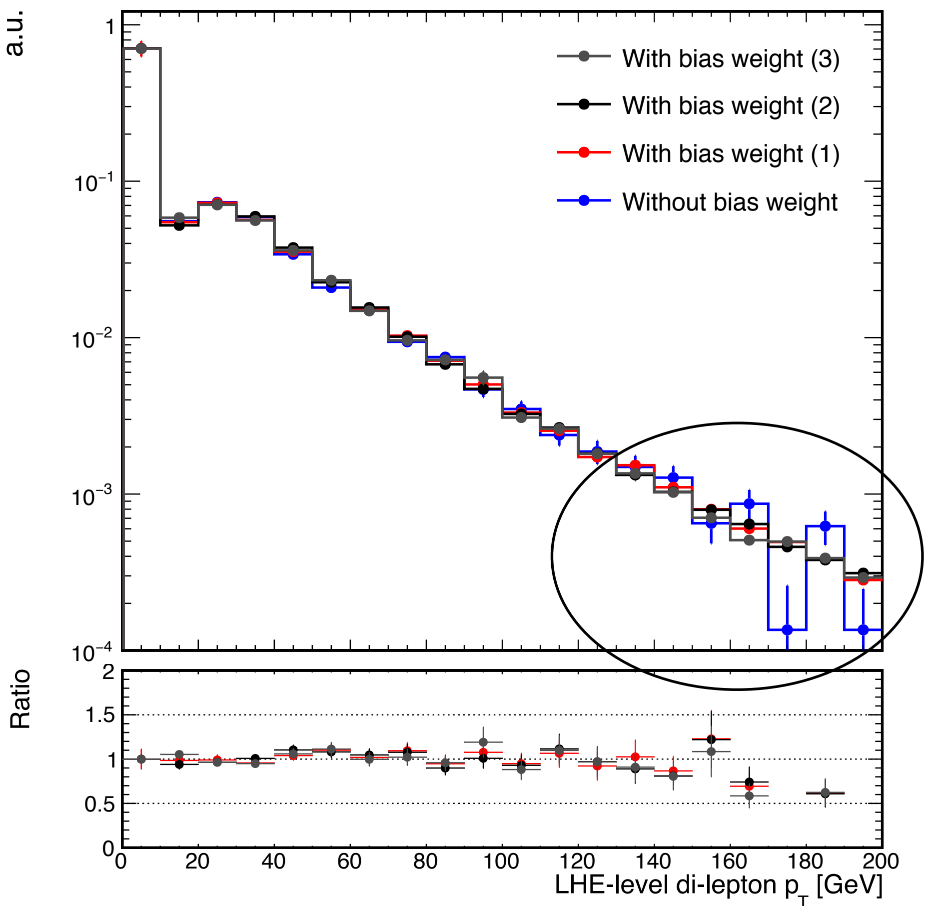
Only the sign of the weight applied (gives ~effective distribution)



Full weight applied (ie including bias weights)
Note: ratio taken with respect to default, unbiased sample (in blue)

Here we see the behaviour that function 1 leads to very large weights at very low LHE p_T (0 jet evts)

LHE-level comparisons: Z p_T



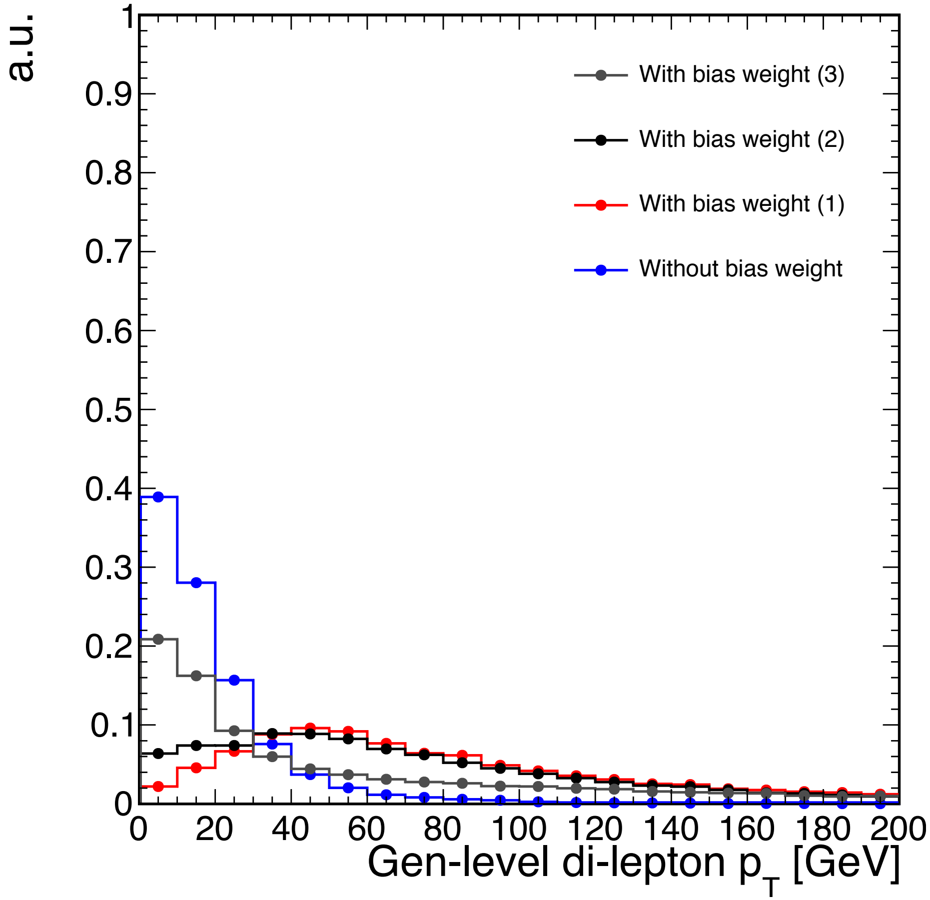
Biased events give a visibly smoother tail here

Full weight applied (ie including bias weights)
Note: ratio taken with respect to default, unbiased sample (in blue)

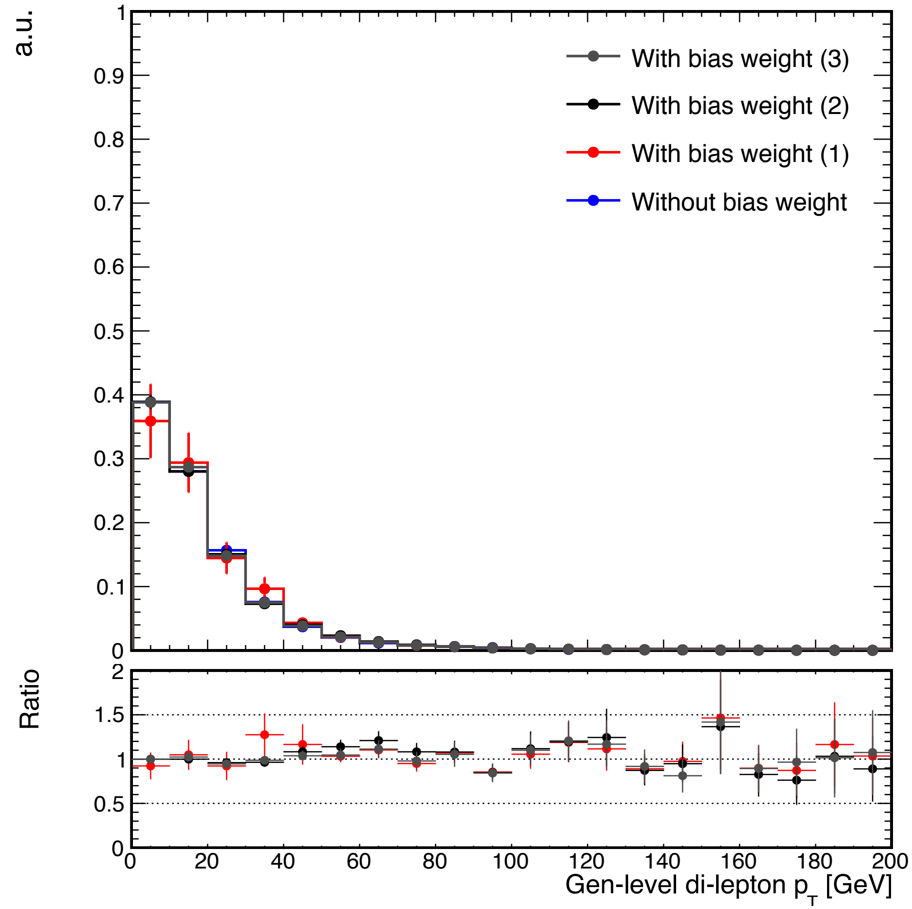
Gen-level comparisons

- Next few slides show ~same comparisons, but now using gen-level quantities (ie post-shower)
  - Di-lepton p_T and mass calculated using leptons from the genParticle collection, which satisfy:
    - **Electrons/Muons:** status flag IsPrompt **OR** IsDirectPromptTauDecayProduct, in addition status flag IsLastCopy
    - **Hadronic taus:** rebuild the hadronic taus by summing four-vectors of gen-level tau decay products (unless the tau decays leptonically).

Gen-level comparisons: Z p_T



Only the sign of the weight applied (gives ~effective distribution)

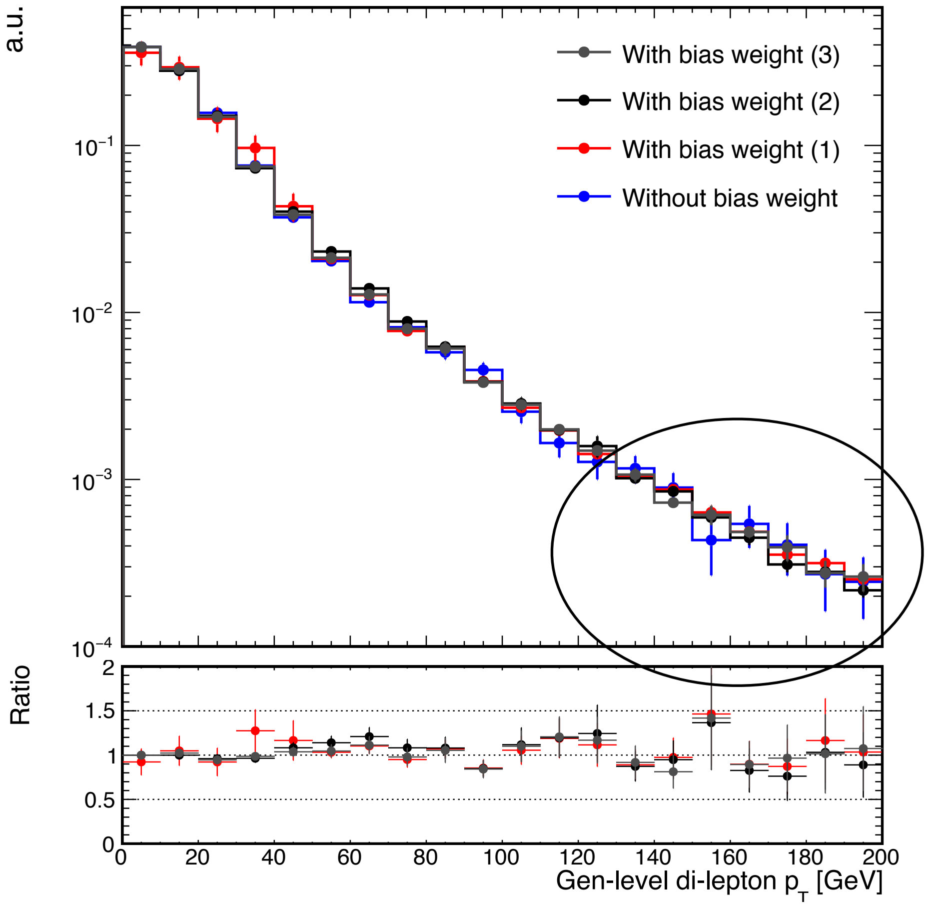


Full weight applied (ie including bias weights)
Note: ratio taken with respect to default, unbiased sample (in blue)

Here we see the behaviour that function 1 leads to very large weights at very low p_T (0 jet evts)

DESY.

Gen-level comparisons: Z p_T



Biased events have smaller uncertainties in the tail

Full weight applied (ie including bias weights)
Note: ratio taken with respect to default, unbiased sample (in blue)

Summary

- **On a technical level the bias weighting does what it promises**
- Definition of the bias function to be used should still be tuned
  - Would still like to test the functions used here without the 10^{njet} factor before going to more complicated functions
  - Should always be careful to generate enough 0-jet (@ME-level) events so as not to have huge uncertainties at low boson p_T

DESY.