Z b-mass Uncertainty Study

3-Slide Status Summary + Backups

Luca

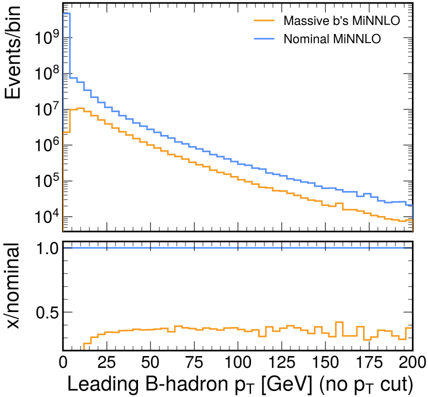
February 16, 2026

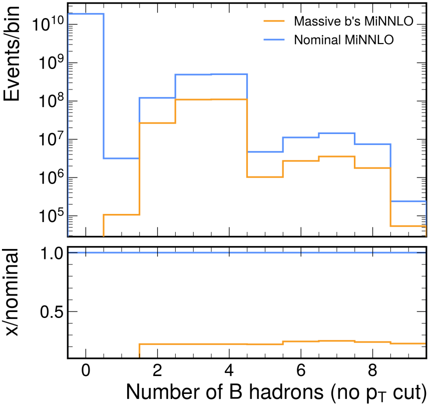
Overview: Samples, Goal, Methodology

- Samples: 5FS nominal = Zmumu MiNNLO (massless b), 4FS alternate = Zbb MiNNLO (massive b).
- Goal: build a b-mass uncertainty nuisance by replacing a selected 5FS heavy-flavor component with the 4FS component.
- Method: define a swap region using B hadrons from GenPart (status 1 or 2), compare object-level and inclusive-shape impacts, and assess unnormalized swap behavior first.

B-Hadron Diagnostics (No B-hadron p_T Cut)

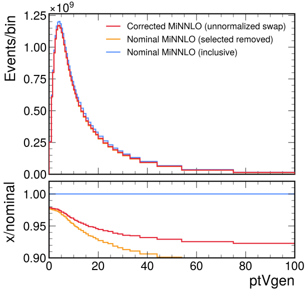
- B hadrons are identified from GenPart using PDG-ID hadron-content logic (contains bottom flavor) and requiring GenPart status 1 or 2; no additional B-hadron p_T threshold is applied for these two observables. Exact implementation is shown in backup slide ‘Backup: Exact B-hadron Selection Code‘.





Unnormalized Swap Result and Definition

- B-hadron definition: GenPart object passing the same bottom-hadron PDG-ID logic with status 1 or 2. Swap definition: select events with at least one such B hadron ($nBhad \geq 1$); corrected distribution is built as 5FS (unselected) + 4FS (selected), with no normalization factor.

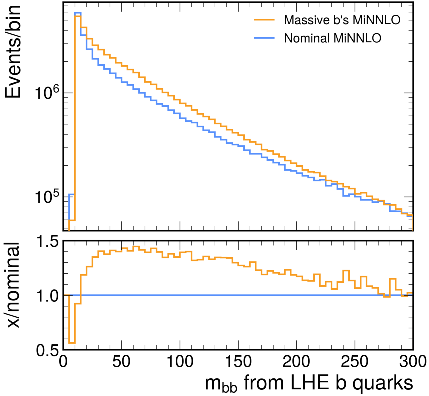


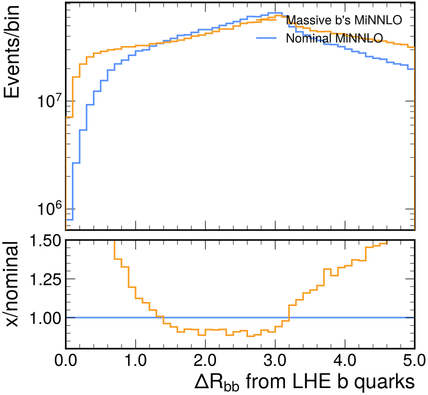
- Result: unnormalized swapping gives a sizable normalization shift (not shape-only), so it is too aggressive as-is.

Backup

Backup: LHE bb Kinematics

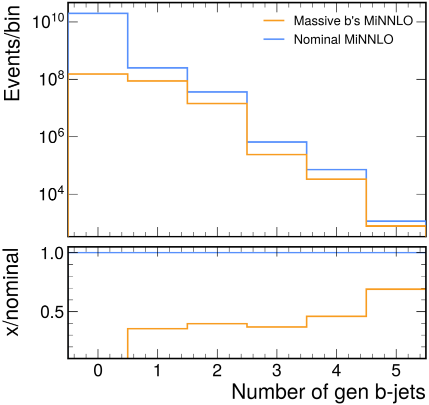
- LHE b-quark observables built with $|\mathrm{pdgId}| = 5$ and final-state status criteria in the histmaker.

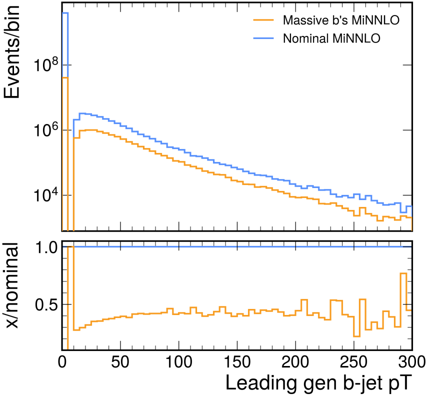




Backup: GenJet b-jet Observables

- Gen b-jets are defined with hadronFlavour = 5; multiplicity uses the standard jet-threshold selection in this study setup.





Backup: Exact B-hadron Selection Code

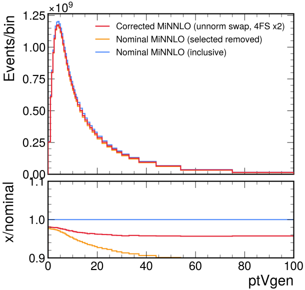
Cleaned Implementation

```
// C++ helper (theoryTools.hpp)
if (status[i] != 1 && status[i] != 2) continue;
const int apdg = std::abs(pdgId[i]);
if (!isBHadron(apdg)) continue;
idx.push_back((int)i);

// Histmaker usage (w_z_gen_dists.py)
df = df.Define("bHadIdx", "wrem::finalStateBHadronIdx(GenPart_pdgId,
df = df.Define("bHad_pt", "Take(GenPart_pt, bHadIdx)")
df = df.Define("nBhad", "static_cast<int>(bHad_pt.size())")
df = df.Define("bottom_sel", "(bHad_pt.size() >= 1)")
```

Backup: Unnormalized Swap with 4FS Scaled by 2

- Same unnormalized swap definition as main slide, but with an extra factor-2 scaling applied to the selected 4FS component before replacement.


×10^9
Events/bin
Corrected MiNNLO (unnorm swap, 4FS x2)
Nominal MiNNLO (selected removed)
Nominal MiNNLO (inclusive)
x/nominal
ptVgen