Data and MC Modeling Improvements

HPS Collaboration Meeting

Matthew Gignac, Elizabeth Berzin

2026/10/20

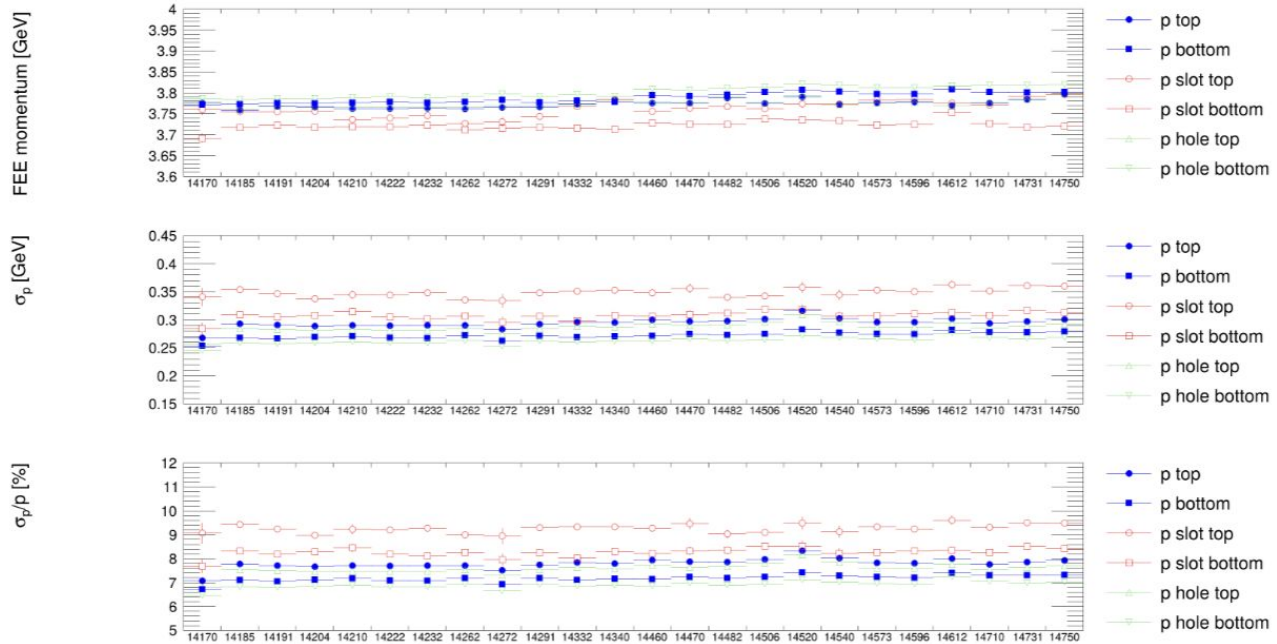
Introduction

- Several issues identified in the MC setup:
  - Low efficiency for low momentum electrons (hps-java #1175)
    - Will talk about this more tomorrow
  - Magnetic field map was incorrect
  - Beamspot was far too narrow (hps-mc #454)
  - Radiative MC did not contain the recoil electron (hps-mc #455)
- Overall, fixes to these issues have lead to an improved description of the data across many observables that were previously mis-modelled
- Slides will report on the improvements with all these points addressed

Field map fixes

Field map studies

- Prompted from the systematically high momentum scale in the 2021 dataset, I began an investigation into field maps in for the 2021 dataset

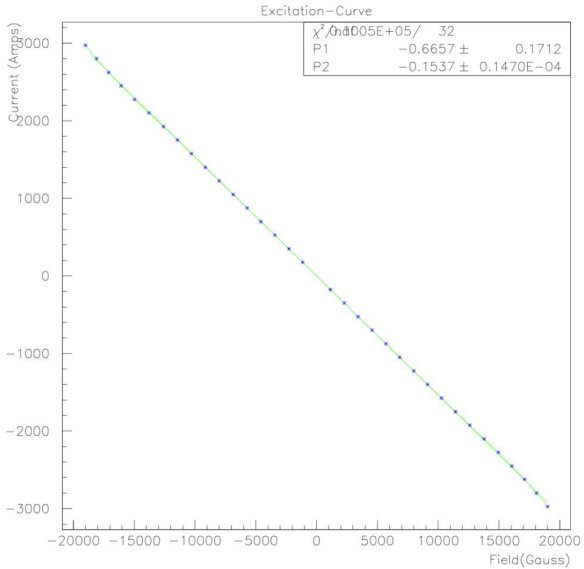


Field map studies

- Prompted from the systematically high momentum scale in the 2021 dataset, I began an investigation into field maps in for the 2021 dataset
- Field is set by scaling the beam energy: $B_0 = A \cdot E$, A = 0.22727 T/GeV); the excitation curve converts that to a magnet current: $I = a \cdot B_0 + b$, where a = 1537 A/T, b = 0.666 A (note)

Field map studies

- Prompted from the systematically high momentum scale in the 2021 dataset, I began an investigation into field maps in for the 2021 dataset
- Field is set by scaling the beam energy: $B_0 = A \cdot E$, A = 0.22727 T/GeV); the excitation curve converts that to a magnet current: $I = a \cdot B_0 + b$, where a = 1537 A/T, b = 0.666 A (note)
- ***The issue:*** The excitation curve is calibrated against the *setpoint* current, but the field maps look to be built from readback currents instead.
- Readback runs consistently a few percent higher than setpoint across all years

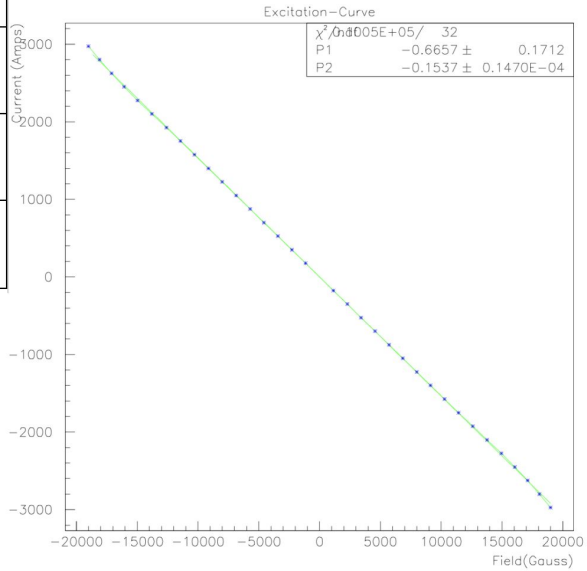


Field map studies

- Prompted from the systematically high momentum scale in the 2021 dataset, I began an investigation into field maps in for the 2021 dataset

	Energy	Setpoint	Readback	Ratio
2015	1058.57	369.6	372.7	0.9917
2016	2306.64	806.4	814.5	0.9901
2019	4556.66	1592.4	1609.5	0.9894
2021	3741.97	1307.8	1321.6	0.9896

- Readback runs consistently a few percent higher than setpoint across all years
- However, 2019 maps used setpoint already! Issue only in 2021 maps

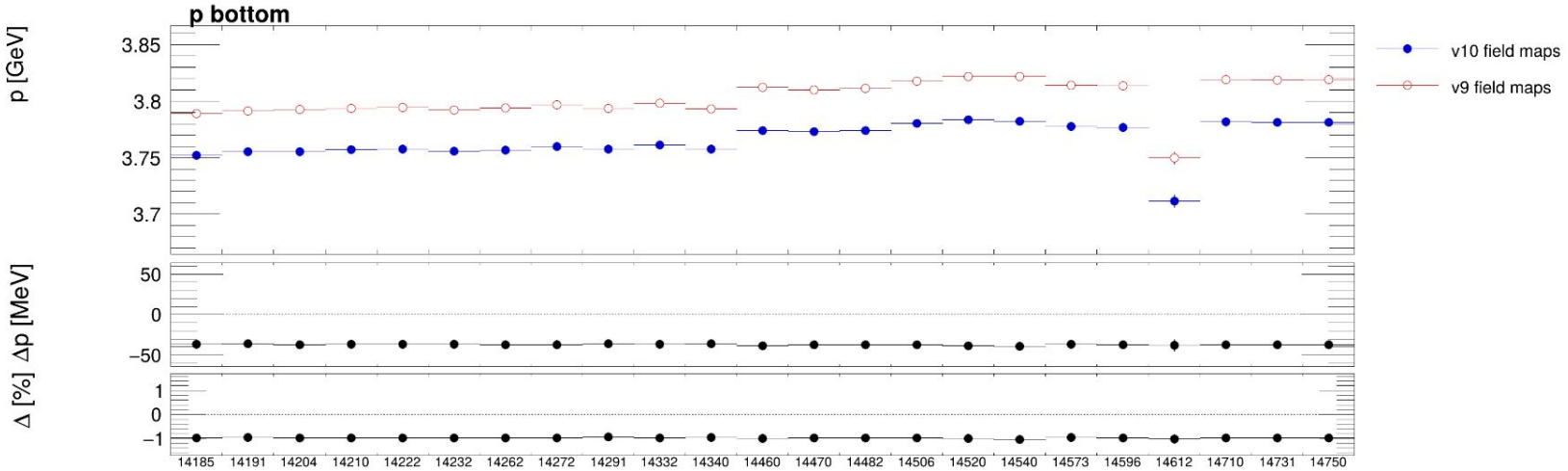


Field map studies: studies

- Re-generated the field maps, following exact steps in notebook here but using the setpoint instead of the readback current
- Studied **momentum scale**, vertex position, calorimeter matching

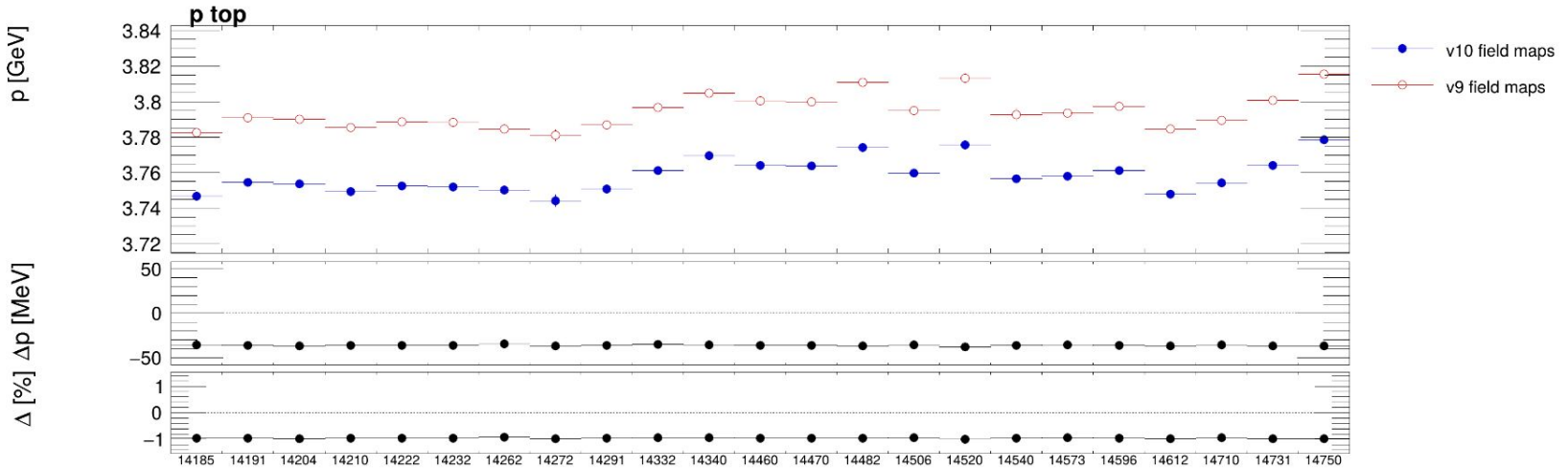
Field map studies: studies

- Re-generated the field maps, following exact steps in notebook here but using the setpoint instead of the readback current
- Studied **momentum scale**, vertex position, calorimeter matching



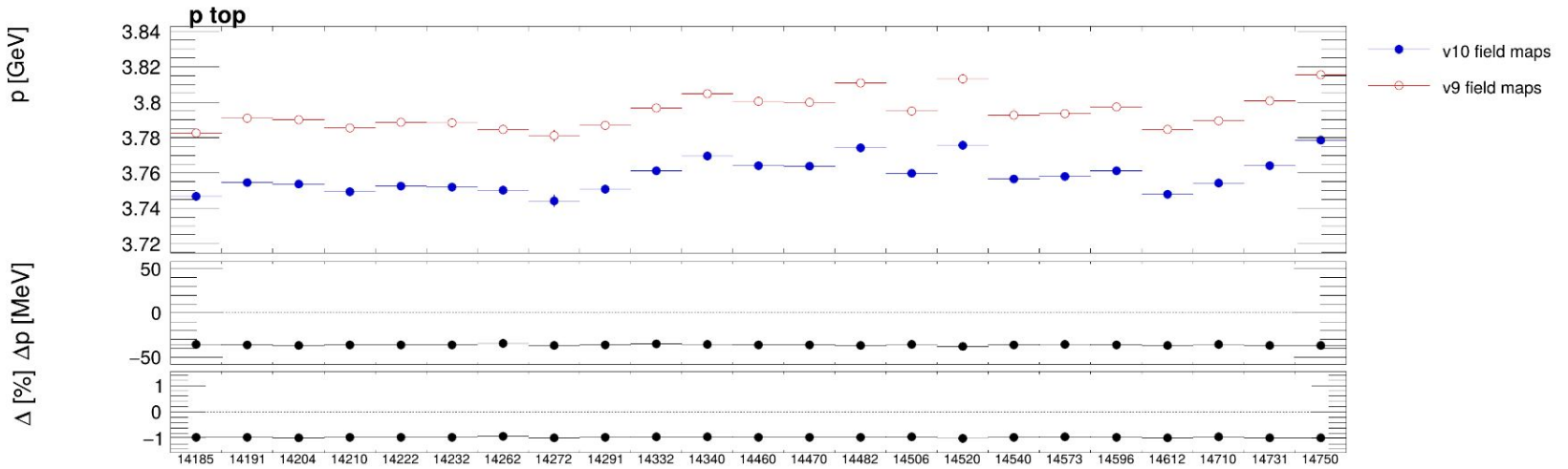
Field map studies: studies

- Re-generated the field maps, following exact steps in notebook here but using the setpoint instead of the readback current
- Studied **momentum scale**, vertex position, calorimeter matching



Field map studies: studies

- Re-generated the field maps, following exact steps in notebook here but using the setpoint instead of the readback current
- Studied **momentum scale**, vertex position, calorimeter matching



Small run-by-run variation – plan to perform scale correction to scale FEE events to beam energy

Other checks & plans

- Vertex positions & ECal matching, all seem unaffected (backup slides)
- Plans going forward:
  - Detectors have been regenerated with tag v10 – suggest we move the next production to this include these changes
  - This only changes the overall momentum scale, resolution is largely unaffected – smearing parameters derived in v9 still valid in v10.
  - Residual FEE scale differences from beam energy to be corrected for on a run-by-run basis. Implemented an FEE skimmer that will be run as part of the v10 production now (hps-java #1182)

Beam-spot

Beam spot size bug

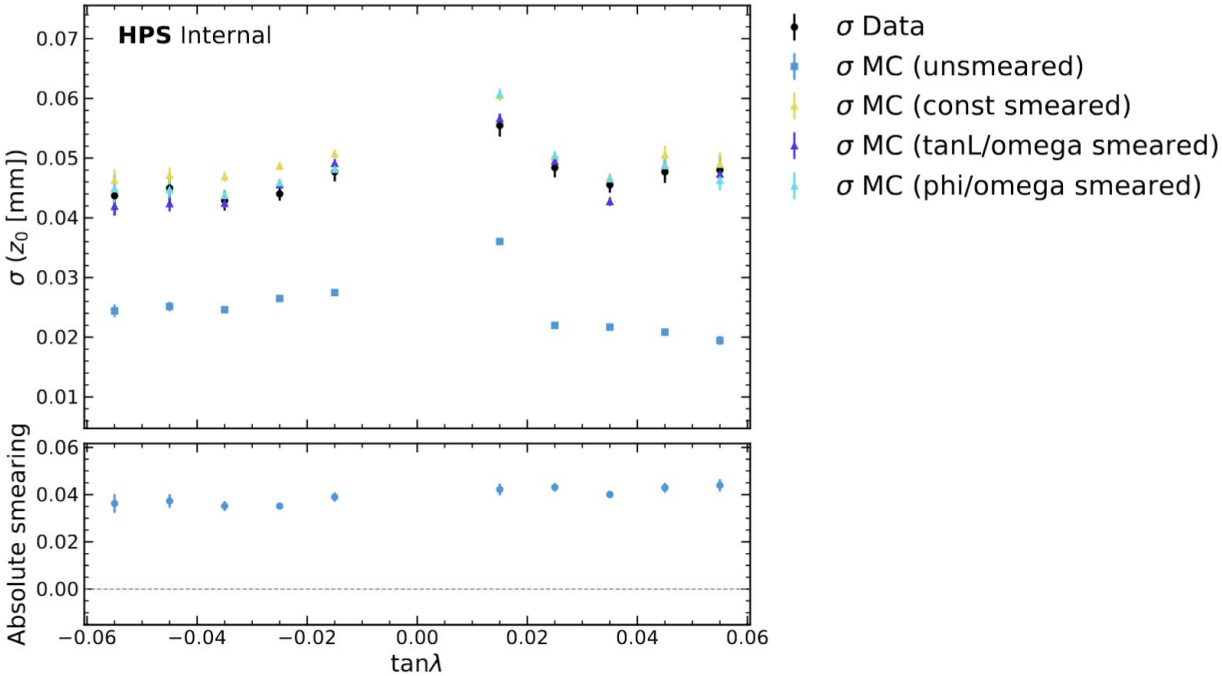
- In studying TC vertex fits, noticed that the beam spot size was sigma^2 rather than sigma – much much too narrow!

tools/stdhep-tools/src/beam_coords.cc

```
@@ -147,8 +147,8 @@ int main(int argc, char** argv)
147  147      nevhep = read_stdhep(&new_event);
148  148
149  149      double shift_x = 0.0, shift_y = 0.0;
150       -    if (sigma_x > 0) shift_x = sigma_x * gsl_ran_gaussian(r, sigma_x);
151       -    if (sigma_y > 0) shift_y = sigma_y * gsl_ran_gaussian(r, sigma_y);
     150  +    if (sigma_x > 0) shift_x = gsl_ran_gaussian(r, sigma_x);
     151  +    if (sigma_y > 0) shift_y = gsl_ran_gaussian(r, sigma_y);
152  152
```

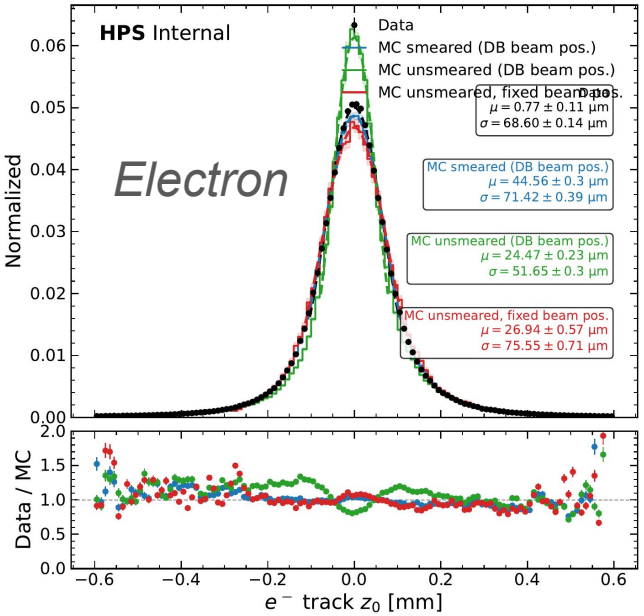
Impact of changes

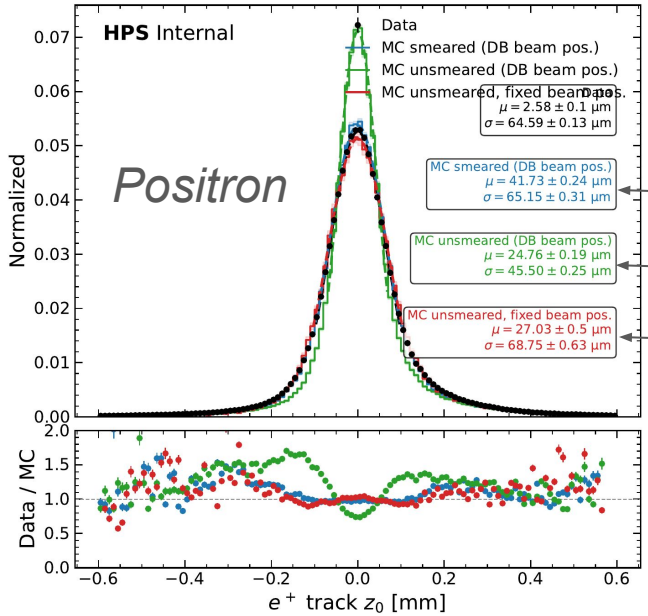
- We had been previously smearing the MC z0 (y0) distribution to match data – this worked very well, but always wondered why this was needed



Impact of changes

- We had been previously smearing the MC z0 (y0) distribution to match data – this worked very well, but always wondered why this was needed
- With the fixed beam spot size, un-smeared MC agrees well with data!





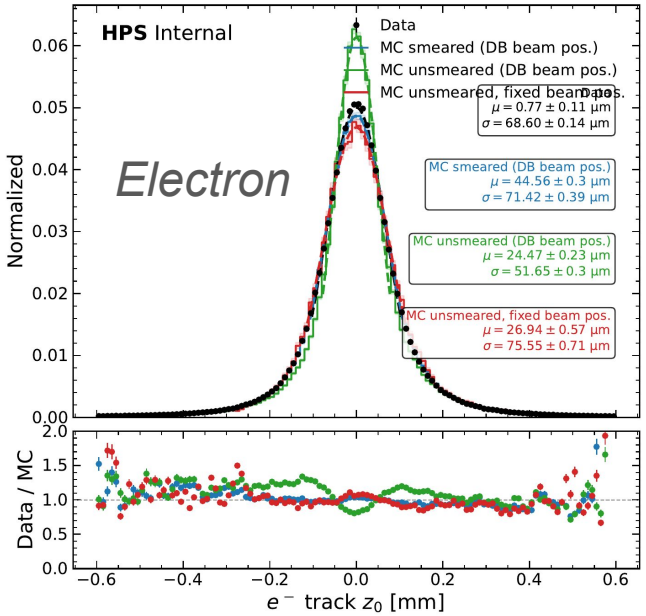
Old MC, z0 smeared

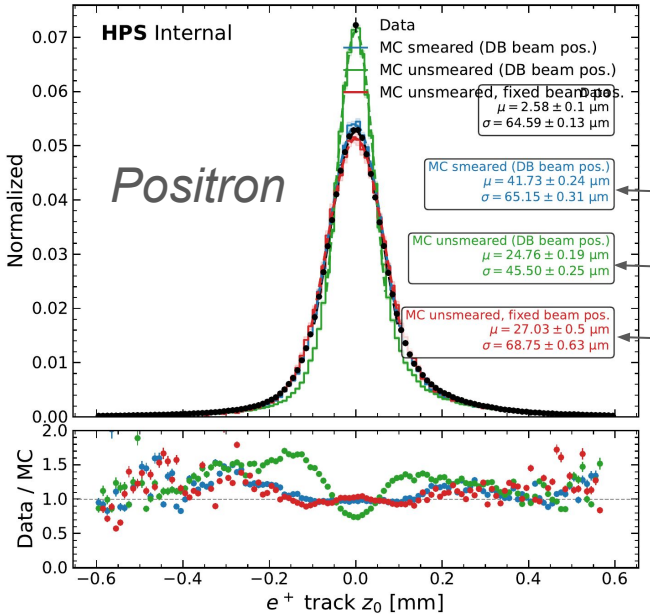
Old MC, no smearing

New MC, no smearing

Impact of changes

Implications: we no longer need to smear z0 (y0) at analysis level when we transition to the fixed MC inputs!





Old MC, z0 smeared

Old MC, no smearing

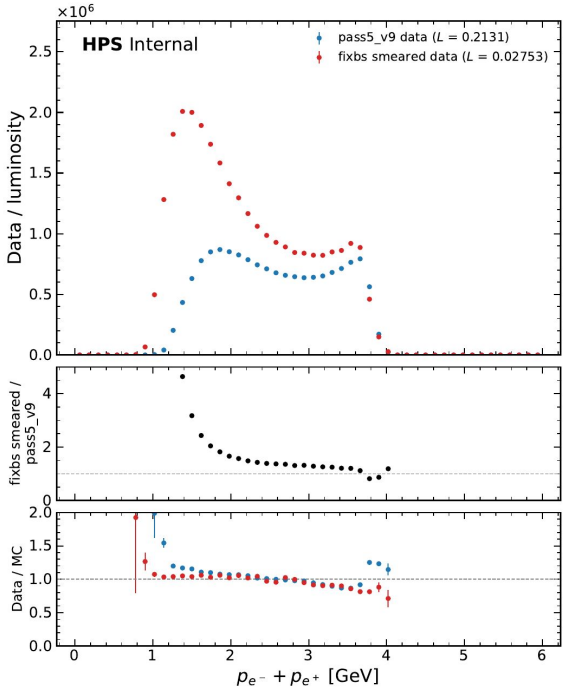
New MC, no smearing

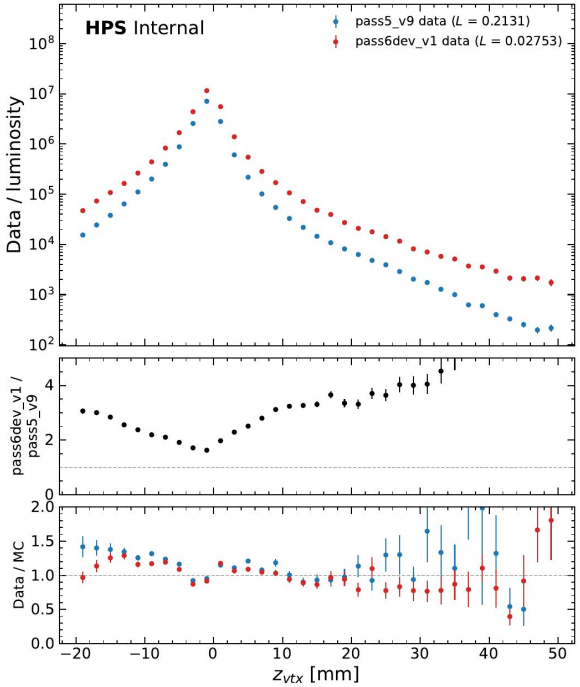
Data/MC agreement

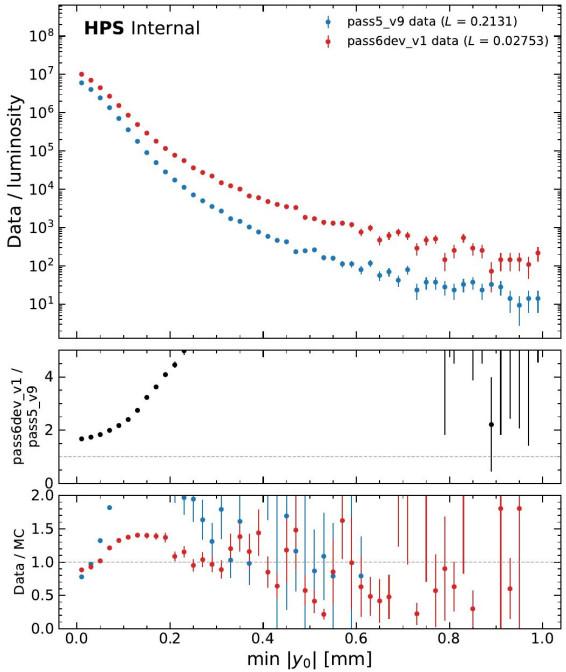
- Data & MC reproduced over the last several weeks with: fixed field maps, correct MC BS conditions, and the low momentum reconstruction
- Overall, significant improvements in the description of the data! Most notably the variables that improved:
  - d0 distribution is modelled much better (largely from lower momentum reconstruction)
  - Resolution of the z0 improved (largely from correct beamspot)
  - Impact parameter uncertainties on electron
- Next plots are all at preselection level, with number of hits relaxed to 6/8 for electrons/positrons, given the low momentum reconstruction pass

Plots

- Largest increase at low p-sum; tails in z0 and min_y0 increase by approximately an order of magnitude

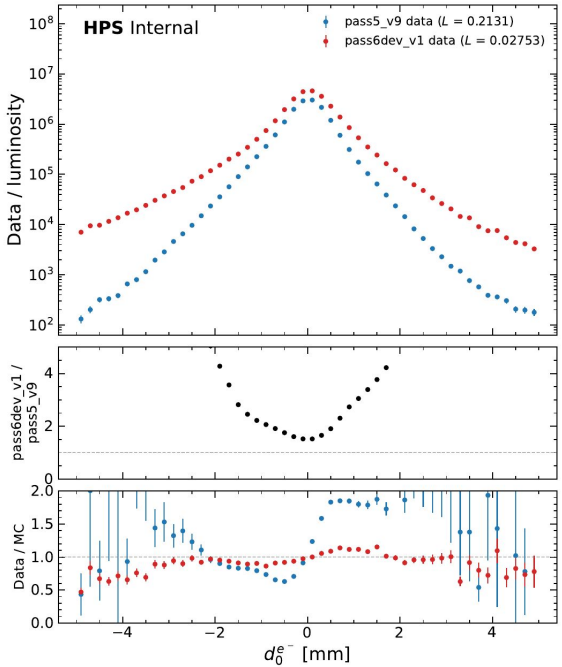


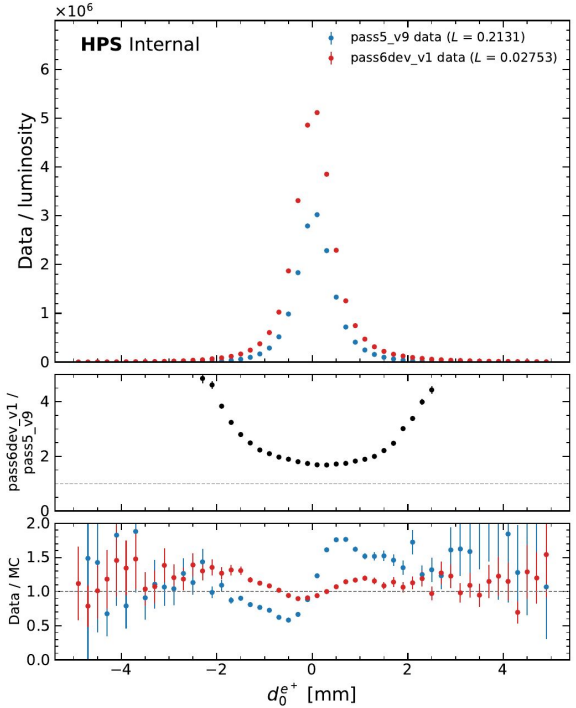


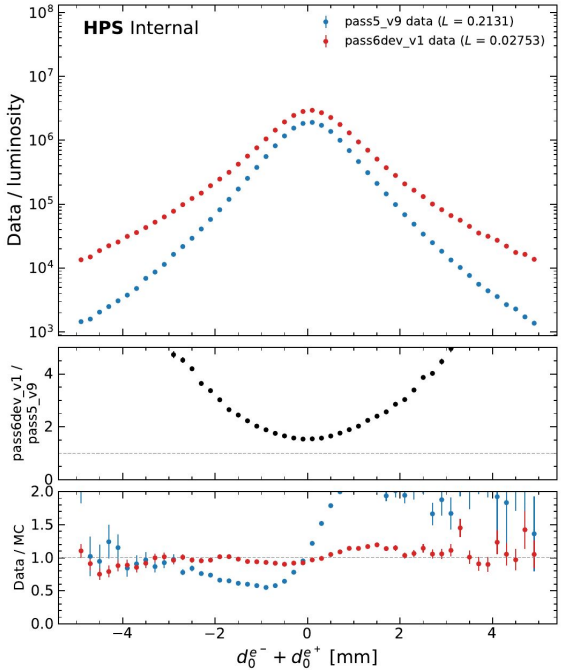


Impact parameters: d0

- Significant improvement in d0 (mostly due to low p tracking)! We should explore this as a discriminating signal vs background variable!

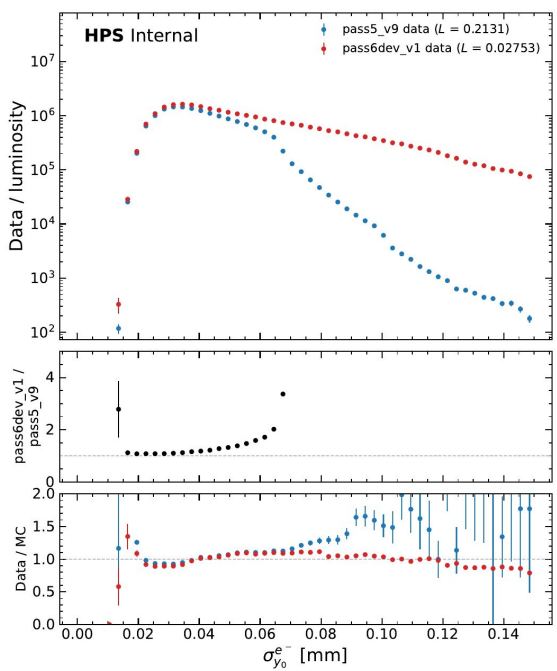


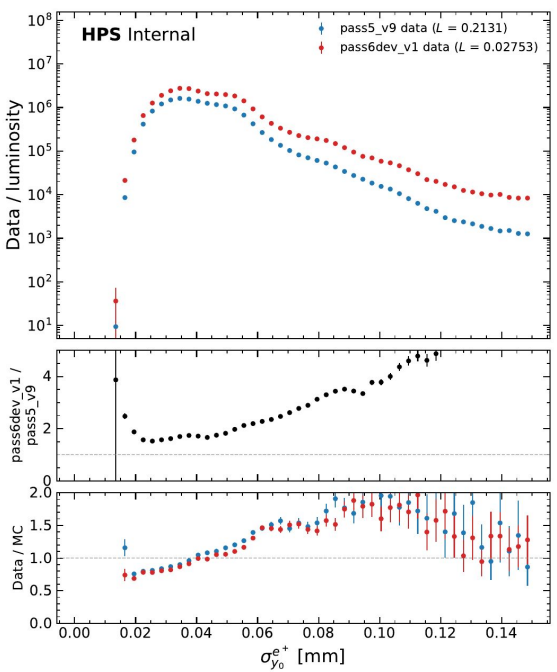


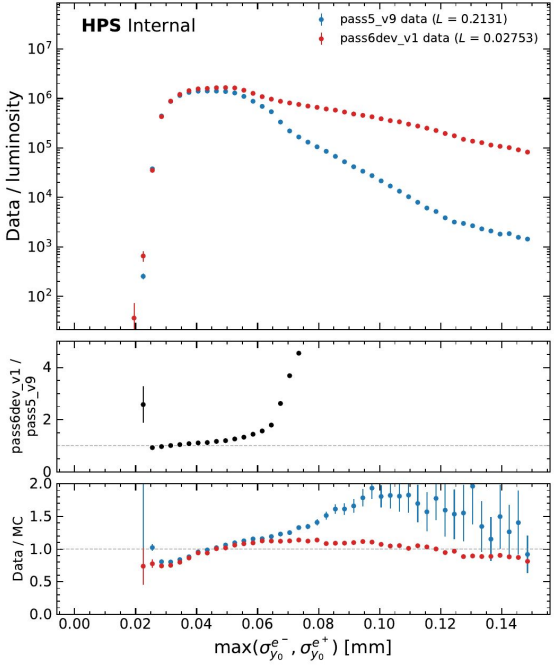


y0 errors

- Significant improvement in electron y0 error, but not positron?

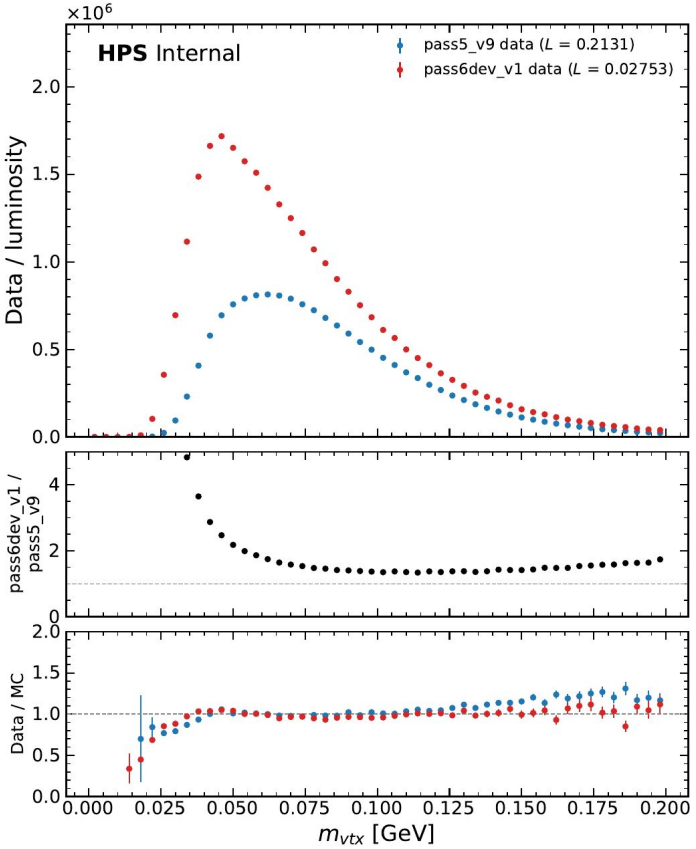






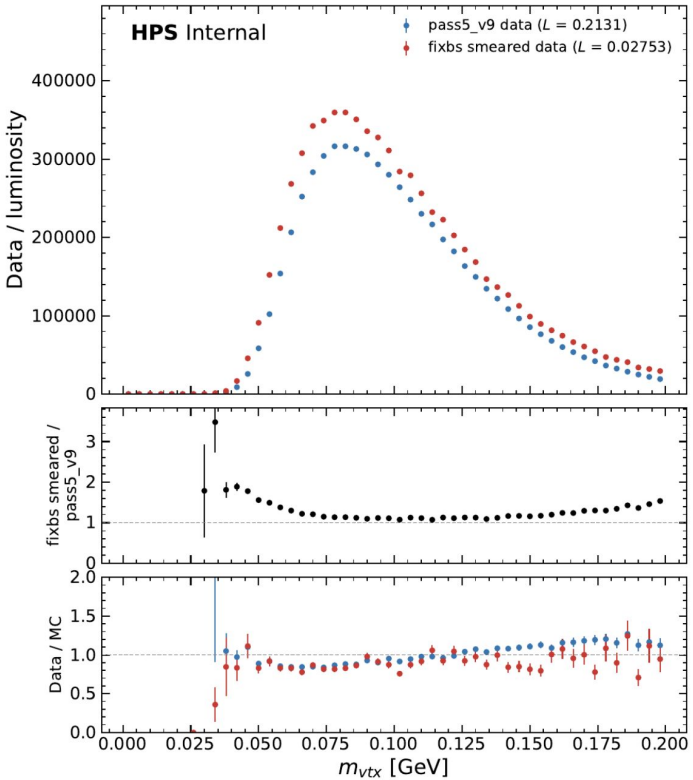
Invariant mass

- Slight improvement to invariant mass modelling, and at a first glance looks like it'd change the shape significantly for the prompt search above 50 MeV



Invariant mass

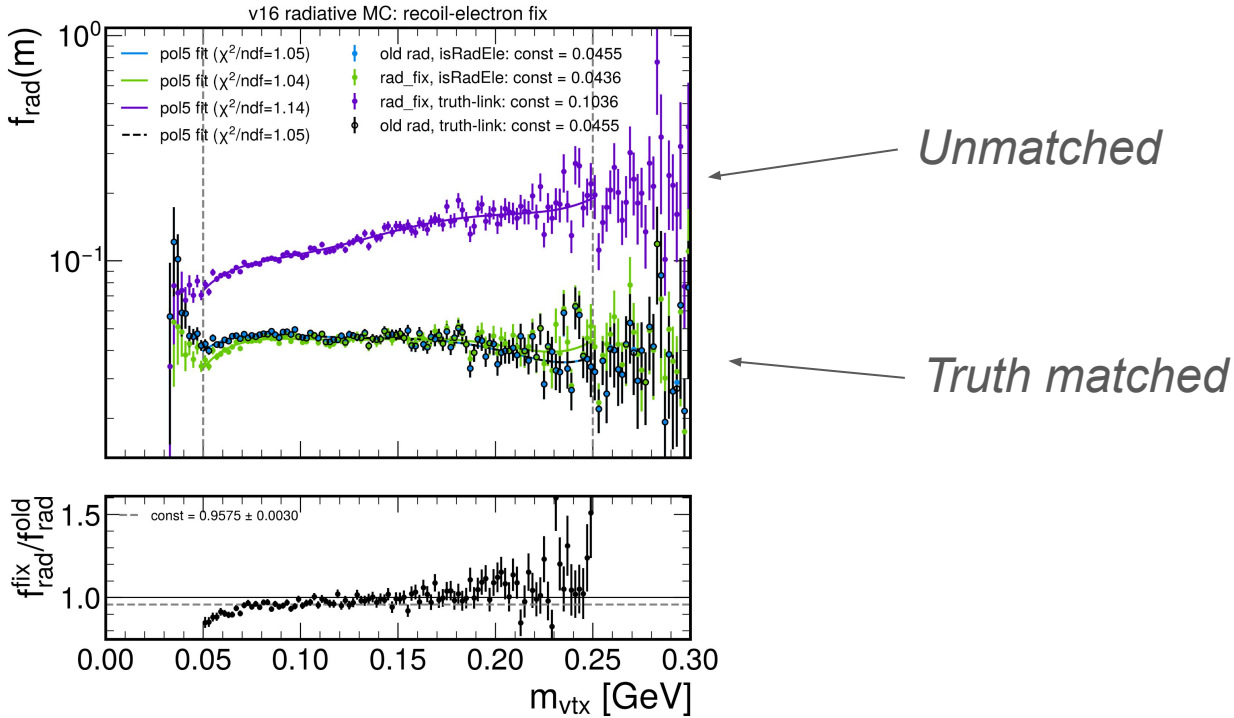
- Slight improvement to invariant mass modelling, and at a first glance looks like it'd change the shape significantly for the prompt search above 50 MeV
- However, with psum>2.8 GeV shapes are quite similar and just a modest increase in data yields – don't expect huge changes for prompt search (noting these are UC vertices)



Radiative Fraction

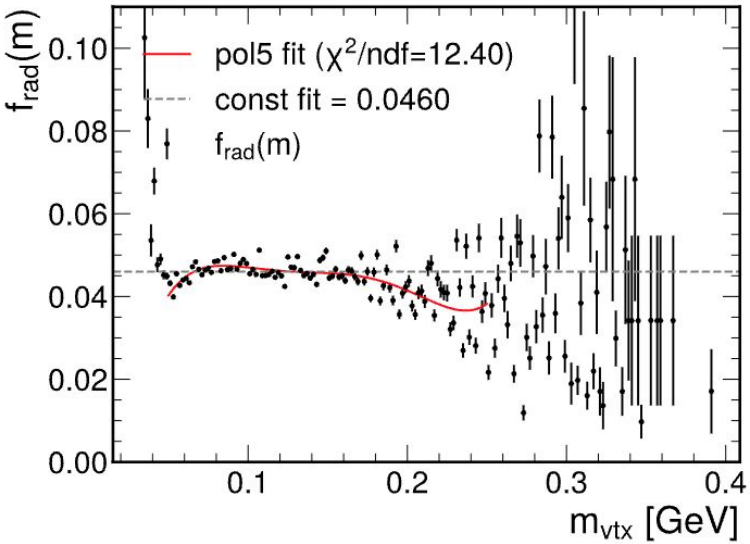
Radiative Fraction

- Elizabeth noticed that the recoil was not included in the simulation
  - Essentially no effect on the radiative fraction once truth matching is applied

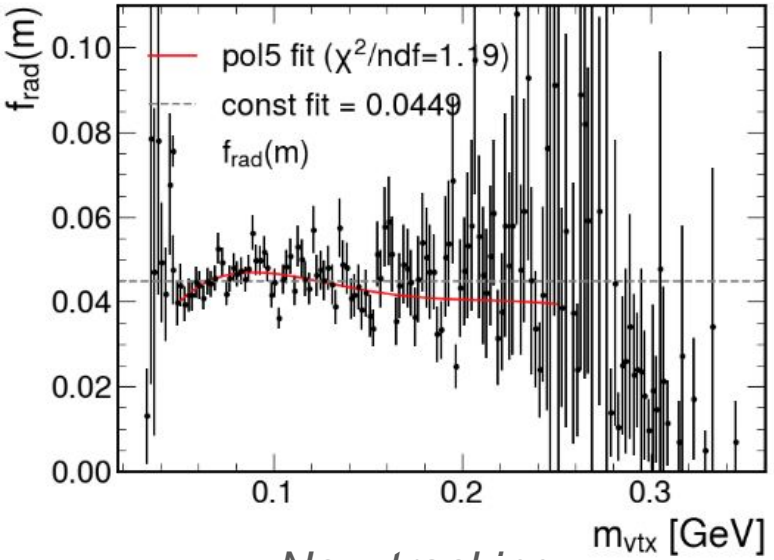


Radiative Fraction

- Elizabeth noticed that the recoil was not included in the simulation
  - Essentially no effect on the radiative fraction once truth matching is applied
- Radiation fraction checks with low momentum reconstruction



Old Tracking



New tracking

Other points worth discussion

Strategy to deal with changing conditions

Several quantities are known to have run-by-run dependencies:

- **Alignment**: run-dependent detector. MC generated with 1-2 detectors, though expected that run dependency is to bring data into (best we can do) alignment, which is what the MC is under these conditions
- **Beam position:** two MC campaigns account for most of the change in conditions. Influences track parameter cuts (min_y0); need run-by-run calibration. If we cut on vertex quantities, such as vertex_y, need run-by-run calibration of vertex positions
- **Hit layer categories**: vary run-by-run, generally degrading as a function fo run – affected selection efficiency;

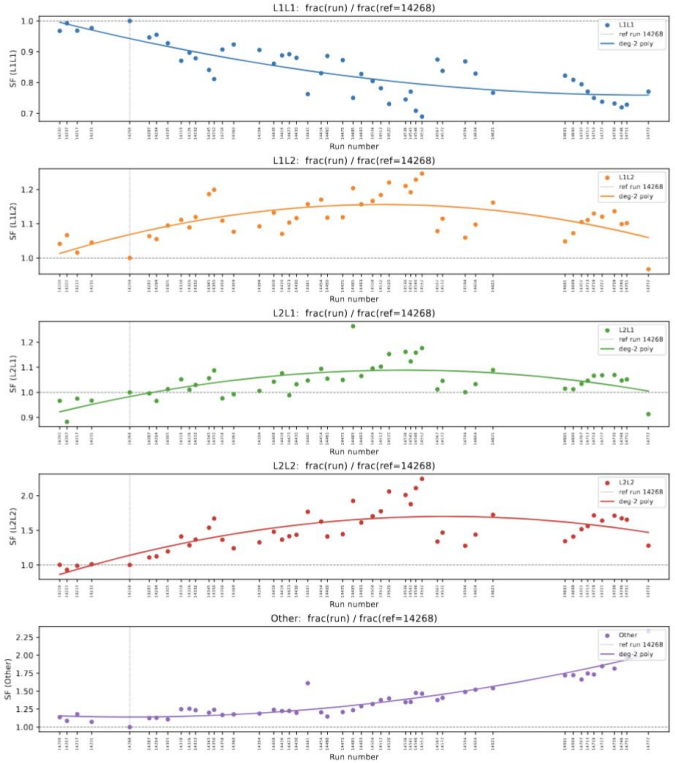
Strategy to deal with changing conditions

Several quantities are known to have run-by-run dependencies:

- **Alignment**: run-dependent detector. MC generated with 1-2 detectors, though expected that run dependency is to bring data into (best we can do) alignment, which is what the MC is under these conditions
- **Beam position:** two MC campaigns account for most of the change in conditions. Influences track parameter cuts (min_y0); need run-by-run calibration. If we cut on vertex quantities, such as vertex_y, need run-by-run calibration of vertex positions
- **Hit layer categories**: vary run-by-run, generally degrading as a function fo run – affected selection efficiency – still not resolved and important: our L1L1 signal yields will be over-estimated!

Hit layer categories

- Efficiency varies by up to 20% across the entire dataset with respect to a reference run at the start of the dataset
- Decreasing L1L1 fraction, increasing fractions for outer layer categories
- Need strategy to reflect these changing conditions in our MC, otherwise our signal yields will be incorrectly estimated



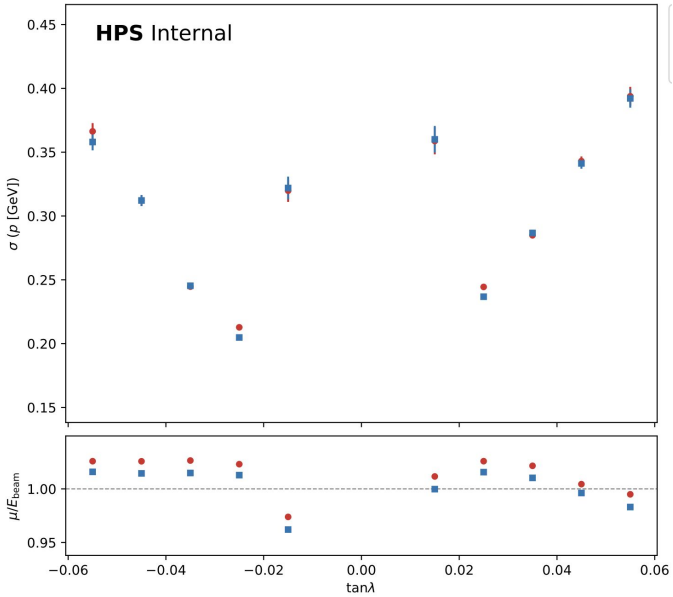
Conclusions

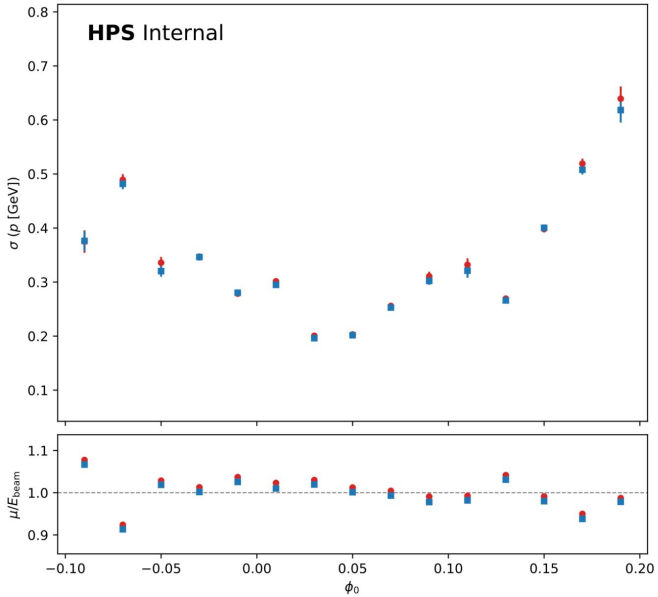
- Several issues have been identified in our MC & data setups that would need to be included into a new pass for a proper fix, but there are patches that might be fine to live with:
  - *Beam spot size* – inclusive smearing does a good job fixing issue and reproduces the data as well as fixed MC samples
  - *Field maps* – fixable with a constant scale correction. We plan to perform a scale correction anyways to fix run-by-run FEE scale variation to beam energy. Bigger issue is deriving it with pass5_v9 (would require FEE pass)
  - *Improved data/MC agreement* – particularly d0 – largely arising from low momentum tracking configuration. Only can benefit from this with pass6_v10. Inclusion of d0 might be additional handle to reject backgrounds, especially in ML techniques.

Backup slides

Field map studies: studies

- Re-generated the field maps, following exact steps in notebook here but using the setpoint instead of the readback current
- Studied **momentum scale**, vertex position, calorimeter matching



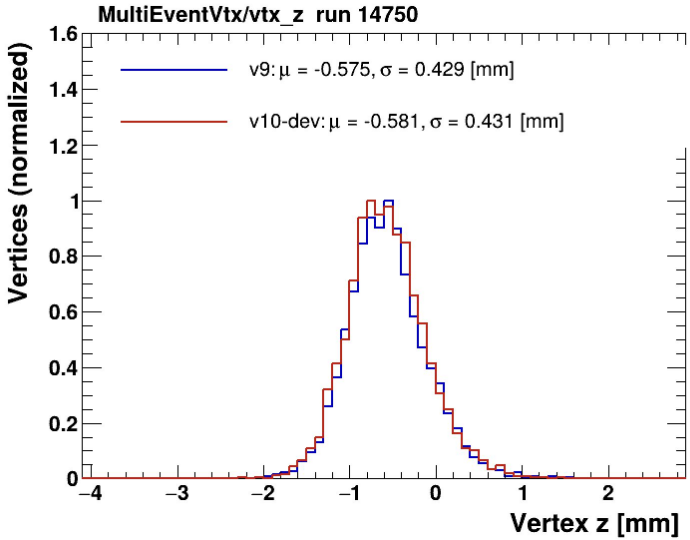




Field map studies: studies

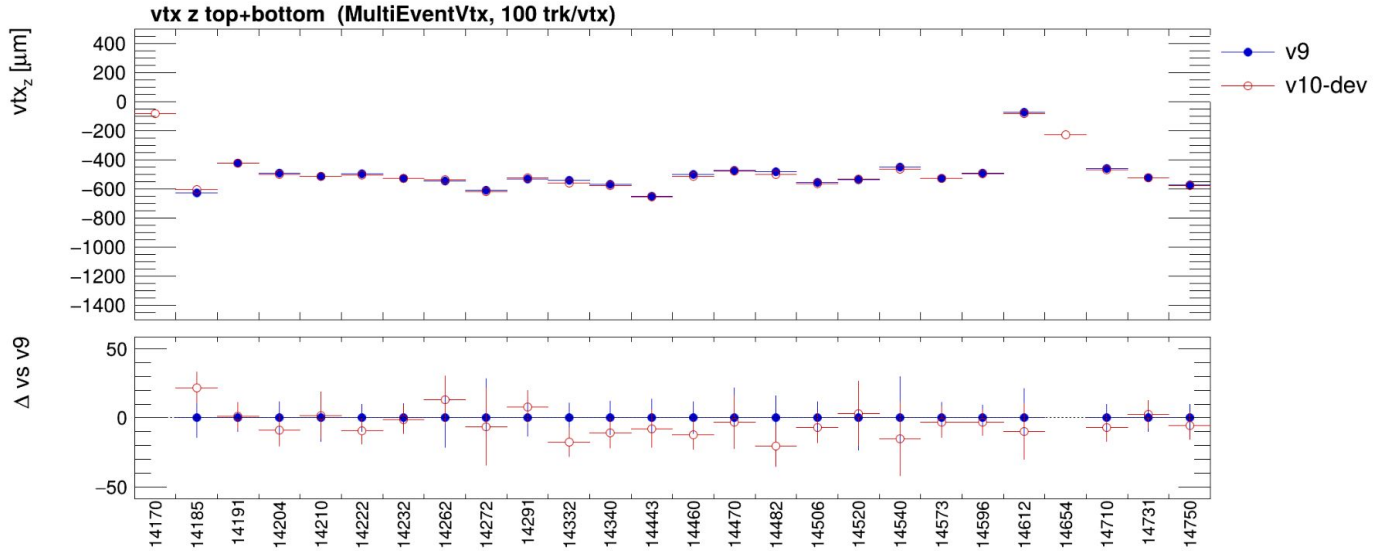
- Re-generated the field maps, following exact steps in notebook here but using the setpoint instead of the readback current
- Studied momentum scale, **vertex position**, calorimeter matching

- Vertex x and y are ~unchanged
- Changes in vertex z top/bottom, mostly small – overall across runs movements leads to improved agreement between top and bottom detectors, but overall combined top+bottom mean and resolutions are ~unchanged



Field map studies: studies

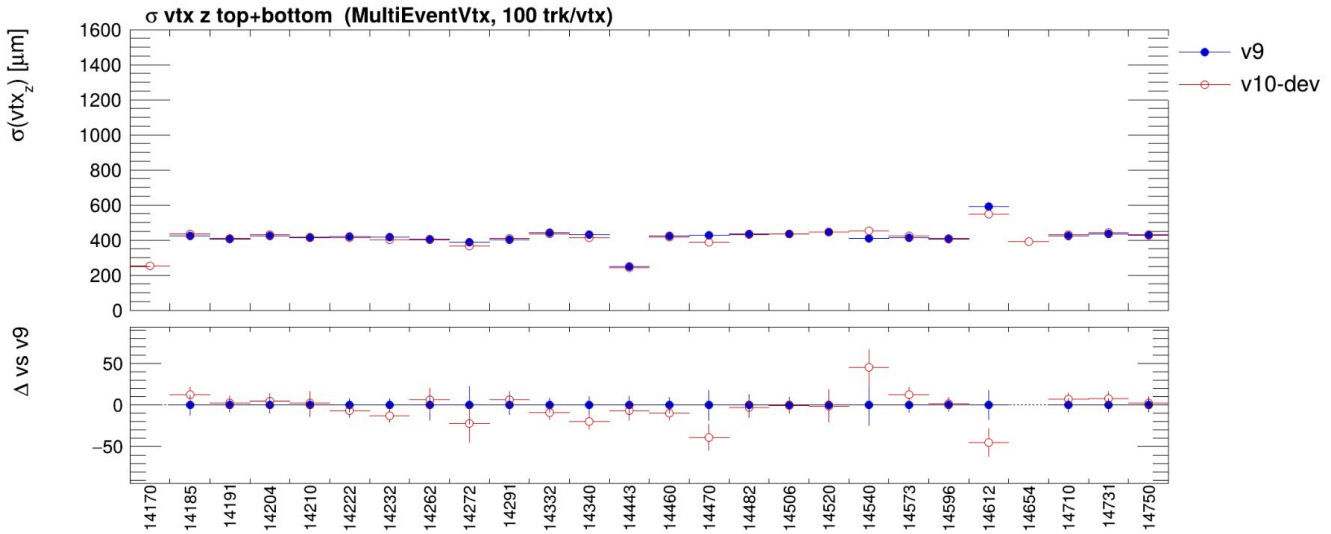
- Re-generated the field maps, following exact steps in notebook here but using the setpoint instead of the readback current
- Studied momentum scale, **vertex position**, calorimeter matching



*Vertex z **mean** unaffected*

Field map studies: studies

- Re-generated the field maps, following exact steps in notebook here but using the setpoint instead of the readback current
- Studied momentum scale, **vertex position**, calorimeter matching



*Vertex z **sigma** unaffected*

Field map studies: studies

- Re-generated the field maps, following exact steps in notebook here but using the setpoint instead of the readback current
- Studied momentum scale, vertex position, **calorimeter matching**

Similar performance

