Status of pixel-only tracking and b-tagging studies

M. Aldaya, J. Olzem (DESY Hamburg)
Tk Upgrade Simulations Technical meeting, 23.1.2012

Setup & input samples

stdgeom

- CMSSW_423_SLHC4
- CMSSW_4_2_3_patch3/ RelValTTbar_Tauola/GEN-SIM/ DESIGN42_V11_110612_special-v1
- 5000 events **PU50**

phase1

- CMSSW_423_SLHC2, R39F16
- CMSSW_4_2_3_SLHC2/ RelValTTbar_Tauola/GEN-SIM/ DESIGN42_V11_110603_special-v1
- 5000 events **PU50**

- samples with data loss: simSiPixelDigis.AddPixelInefficiency = 20

HLT setup

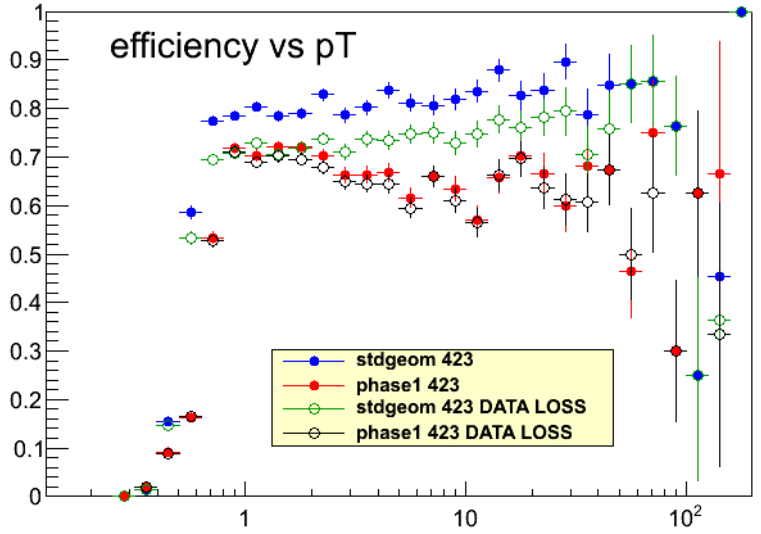
- simplified L2.5 HLT schedule with BTagIP sequence only:

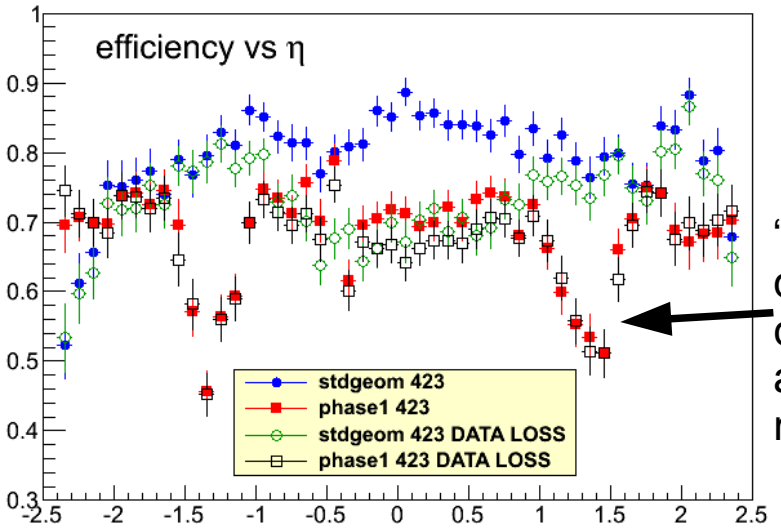
```
process.hltBtagIPPath = cms.Path(
      process.HLTBeginSequence
    + process.HLTRecoJetSequenceAK5Corrected
    + process.HLTBTagIPSequenceL25
)
```

- jet p_T cut: 30 GeV for the b-tag analysis

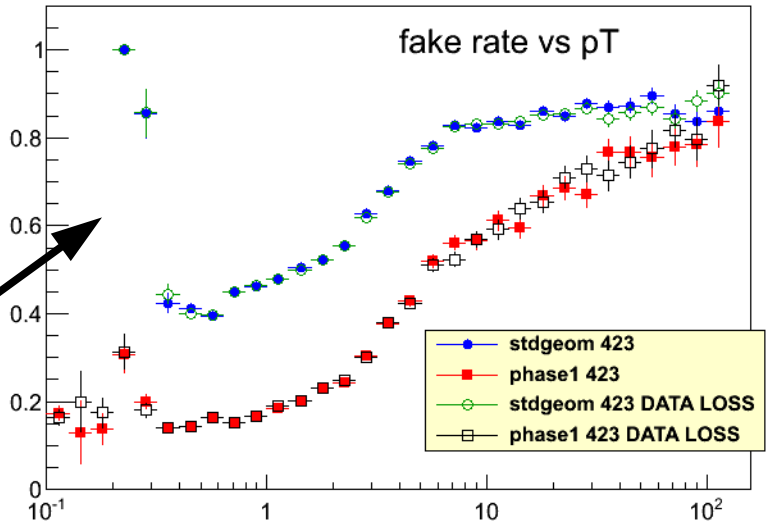
Pixel-only quadruplet tracking performance

Phase1 (quad) has lower efficiency, but drastically better fake rate

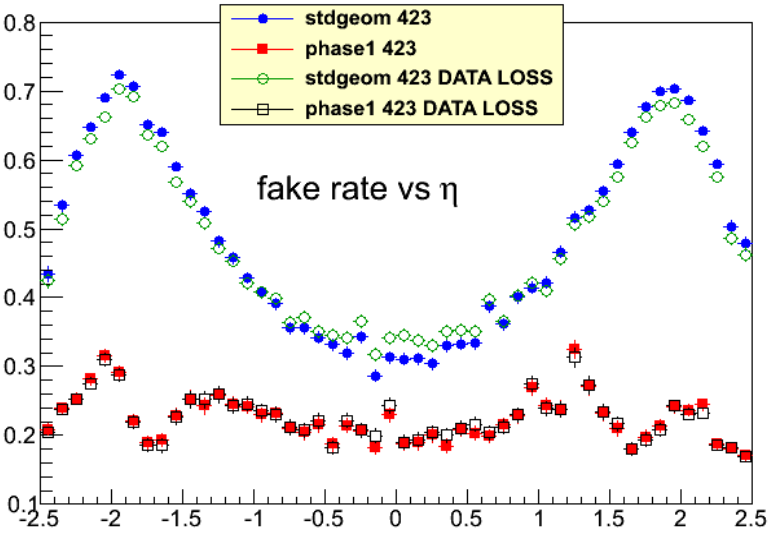




"holes" in quadruplet coverage around η = 1.4

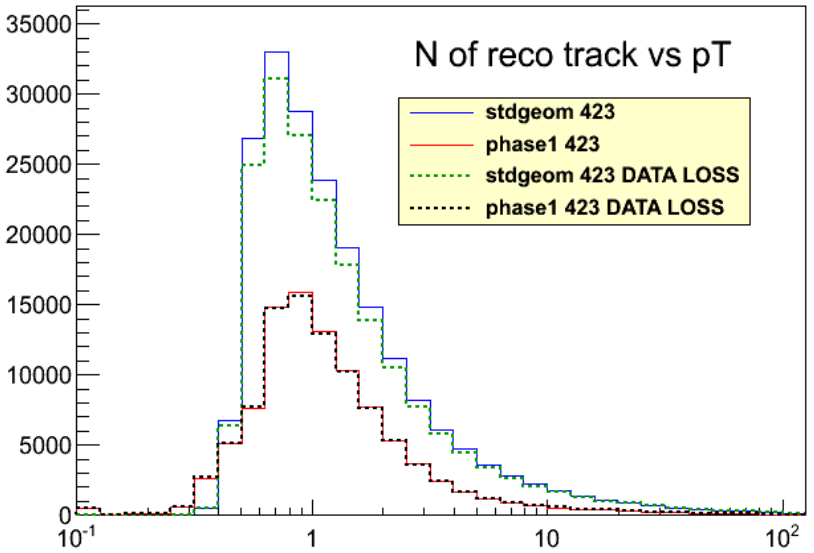


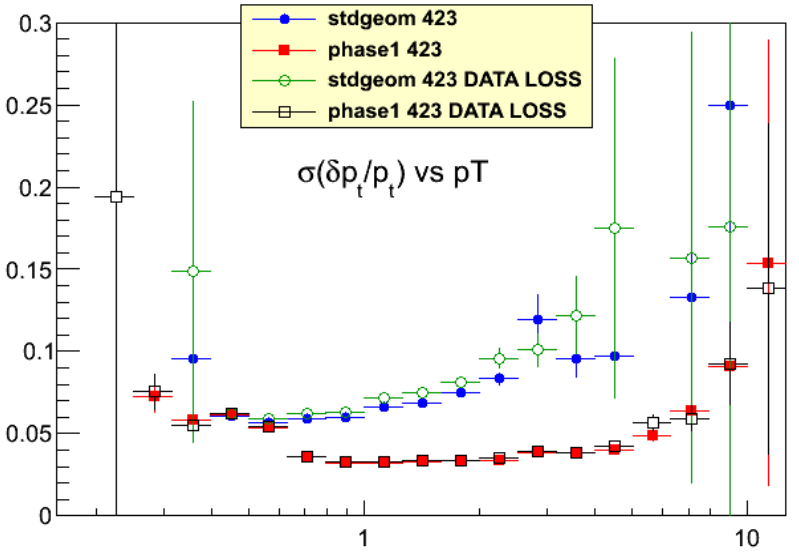
particularly strong effect at low p_T

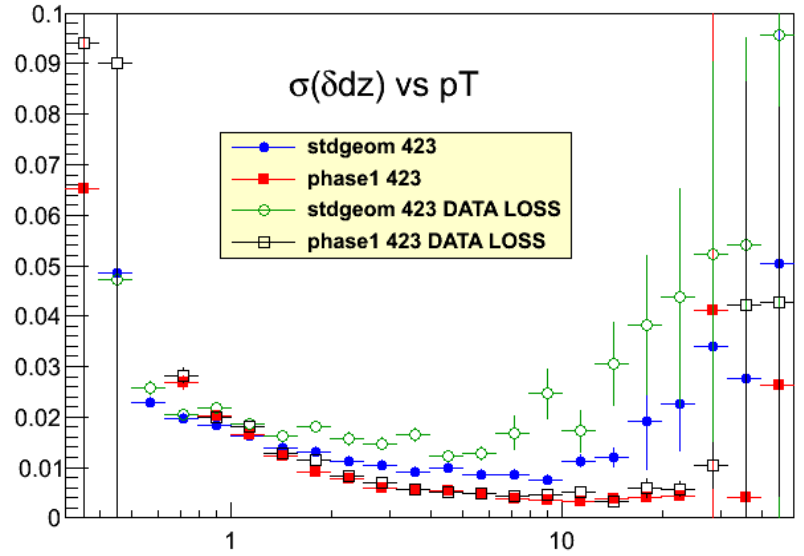


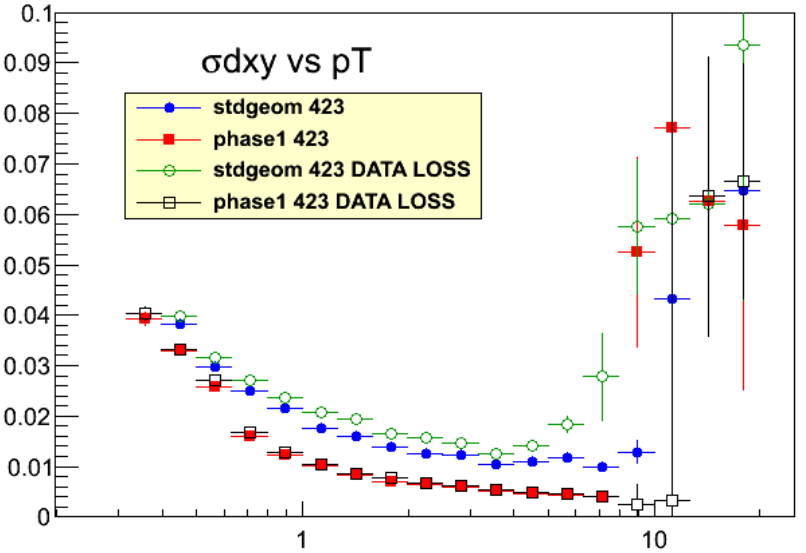
Pixel-only quadruplet tracking performance

Phase1 (quad) has better track quality & resolutions above ~ 1-2 GeV







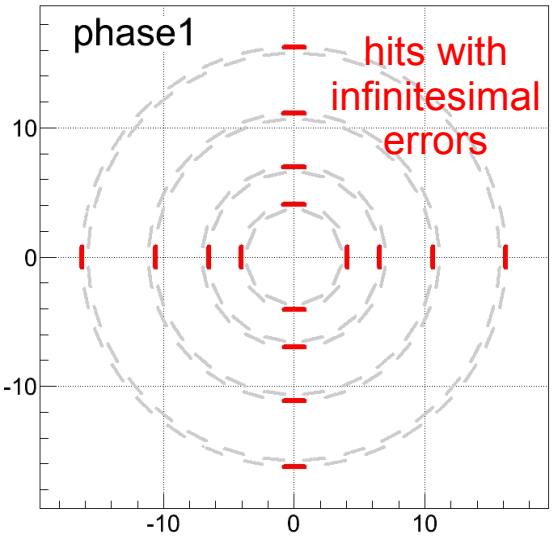


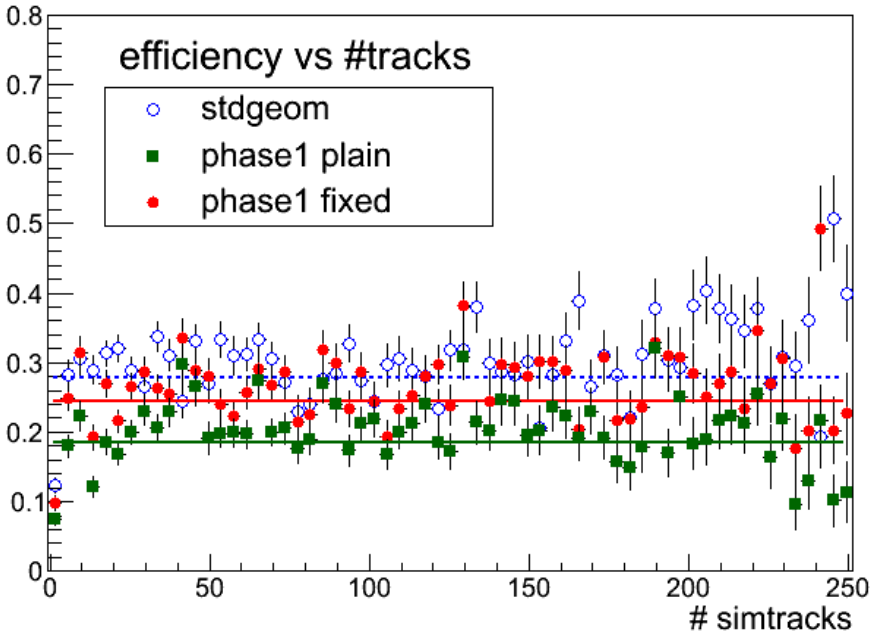
Fixes in vertexing code (default DVF)

In Phase1, pixel hits on certain modules have infinitesimal x or y position errors –

- some code projects localPositionErrors on x and y axes via sinφ and cosφ → **track dz = nan**

- routine in DVF algorithm **rejects a vertex** if it contains such a track → supresses hard interaction vertices & thus devastates the b-tagging performance



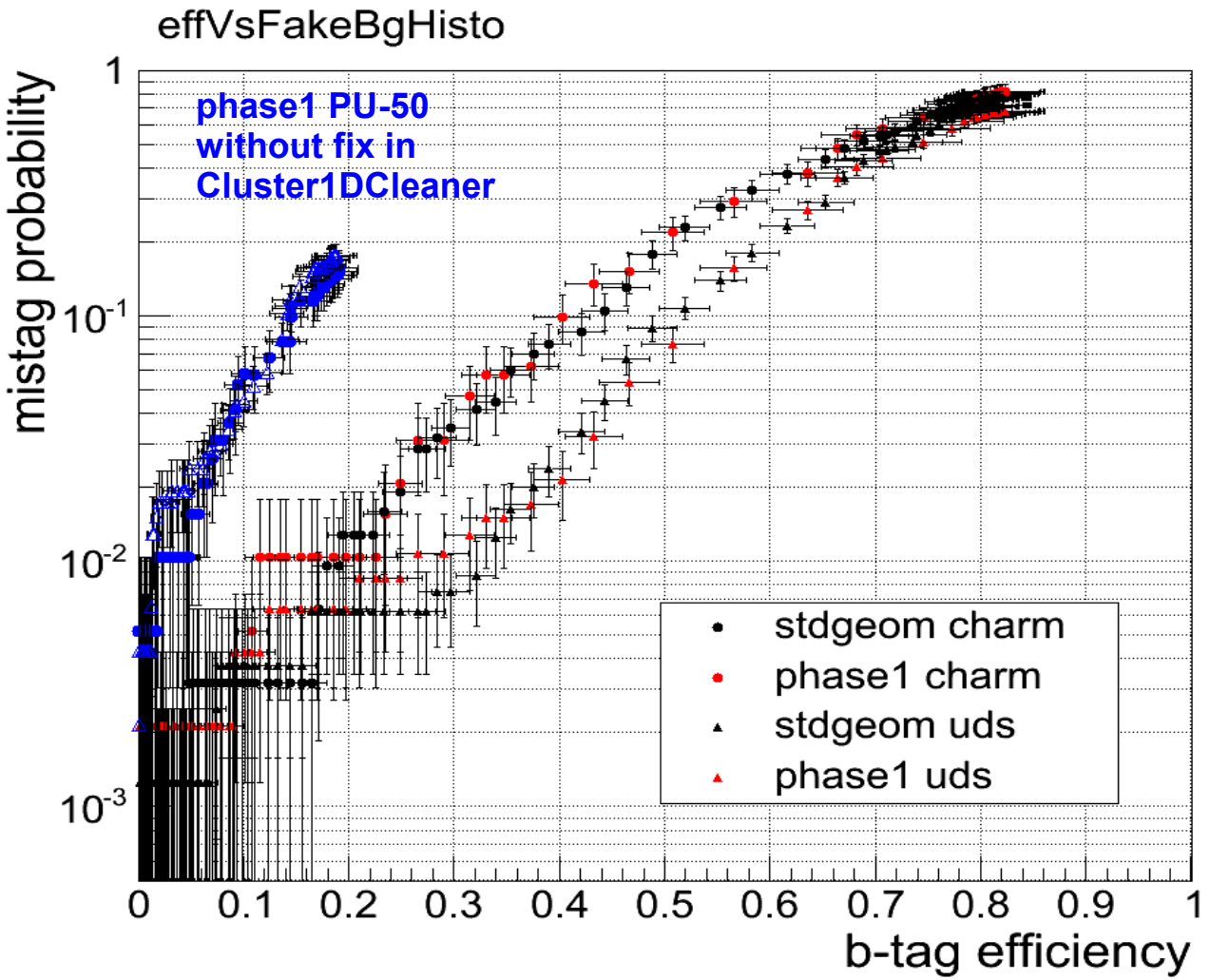


→ implemented a nan check as workaround in DVF

Vertexing efficiency raised by approx. 5% – almost level of stdgeom

(gained vertices are mostly tt, not minbias)

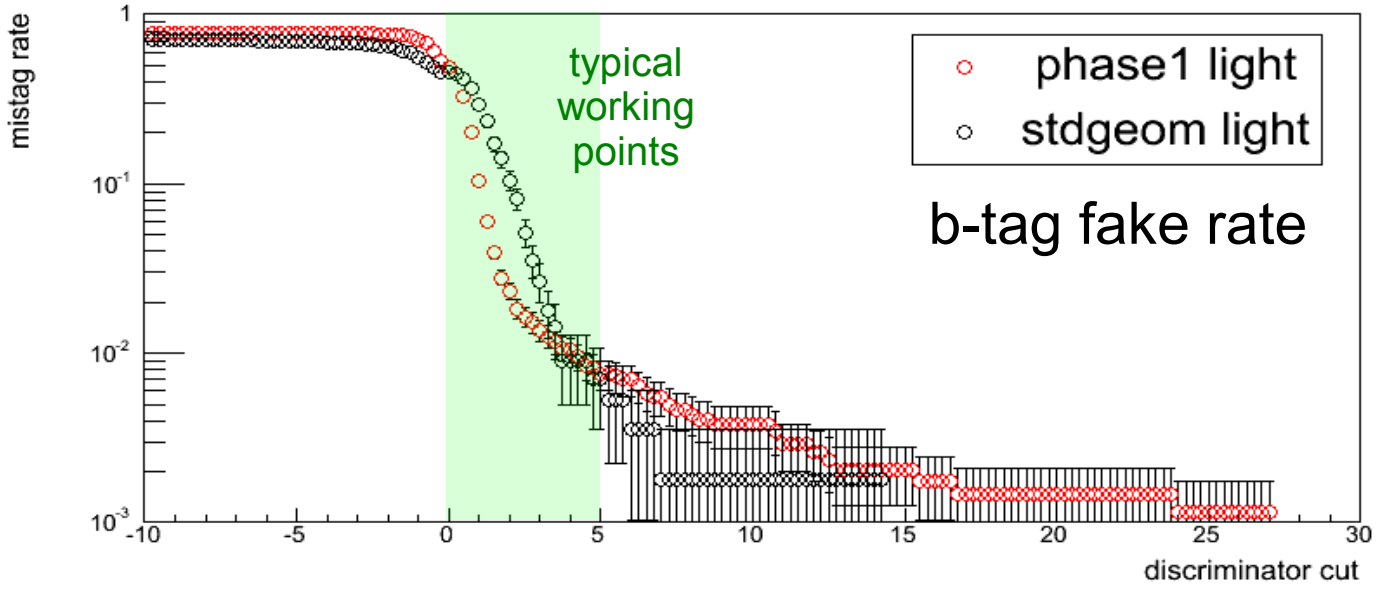
b-tagging efficiency: **quadruplets** (no data loss)

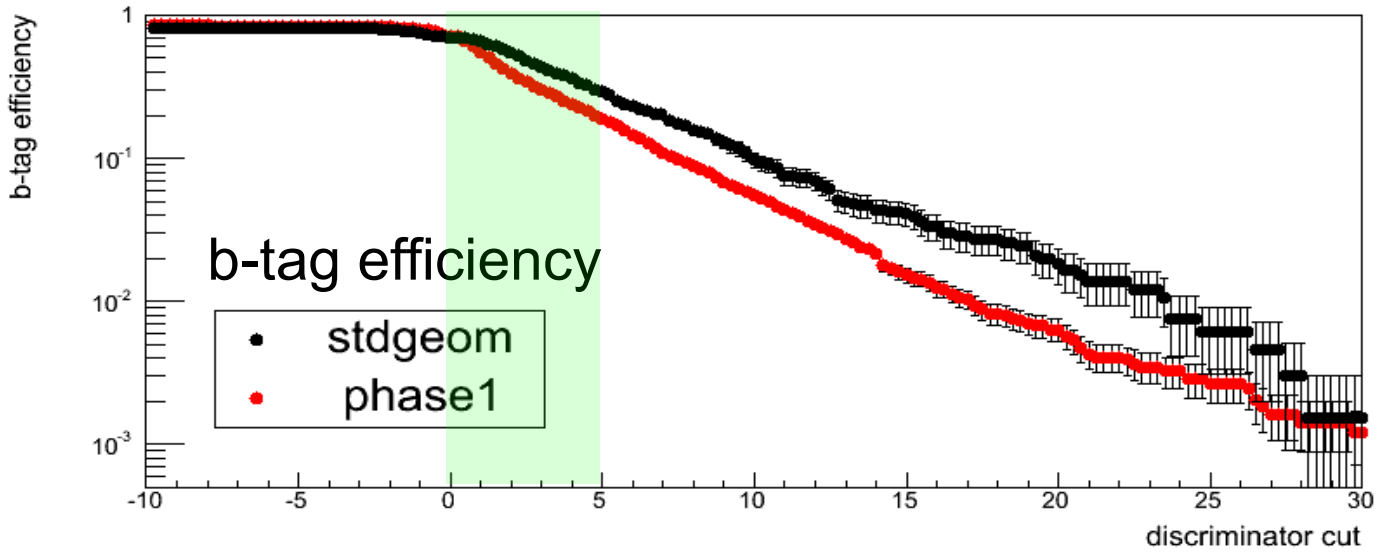


- fixing the vertexing brings phase1 b-tag performance from almost 0 to same level as stdgeom

- no improvement of Phase1 (quads) wrt. stdgeom

b-tagging efficiency: **quadruplets** (no data loss)

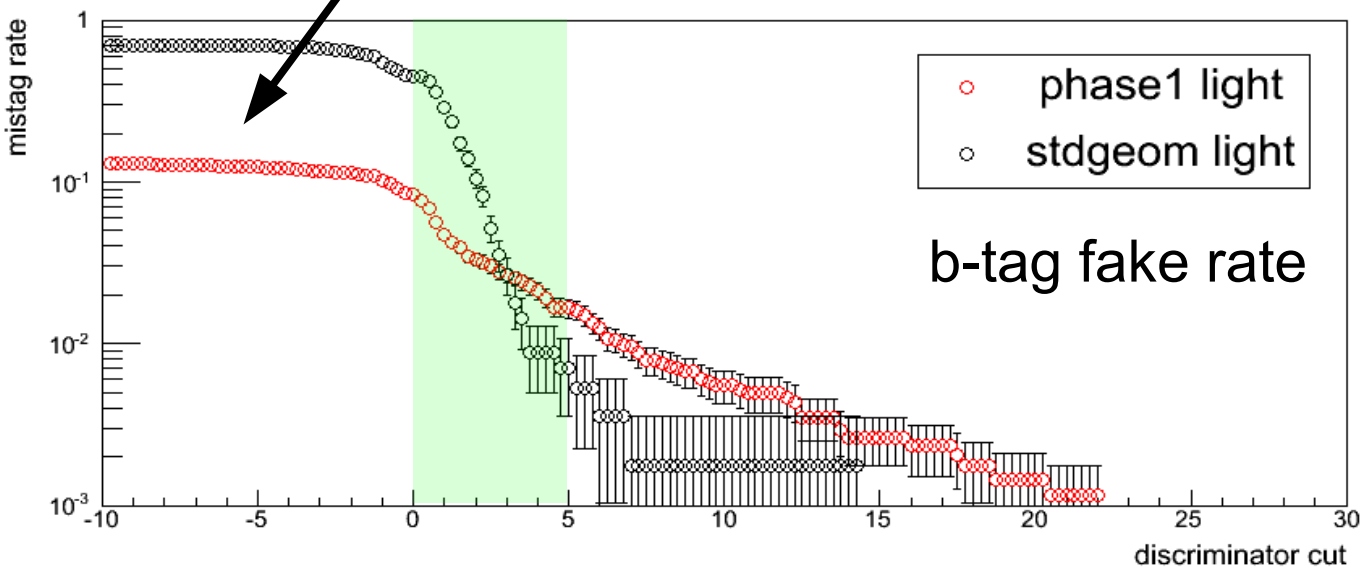


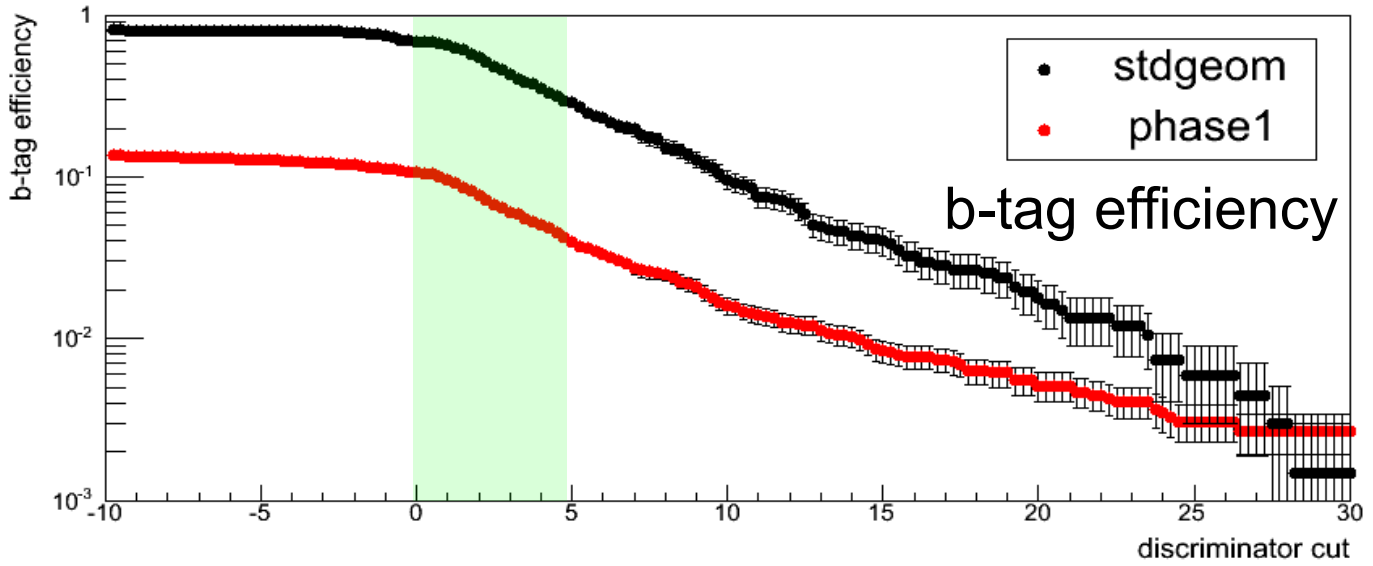


- Phase1 shows better fake rate for discriminators of 0..5, **but:**
- also lower b-tag efficiency due to fewer tracks
- track counting HE algorithm is very **sensitive to the tracking efficiency**
- improved IP resolutions have less impact
- **other algorithms not feasible** (e.g. SSV has very low efficiency with pixel tracks

b-tagging efficiency: **quad+trip** (no data loss)

lower fake rate @ small discrim.: most fakes from triplets are created within b-jets due to the large hit multiplicity – fakes typically have large IP → **b-tag**





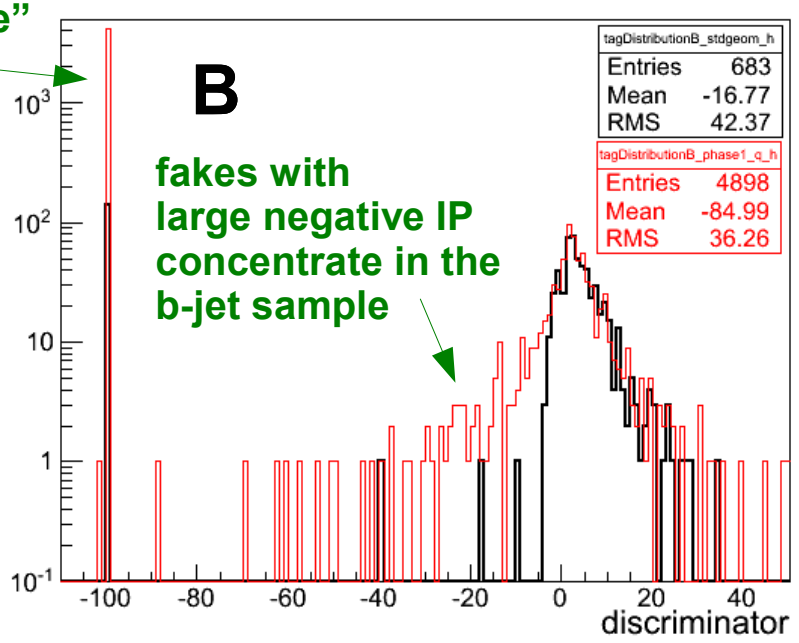
• QuadrupletSeedMerger can add **remaining (non-mergeable) triplets**, but:

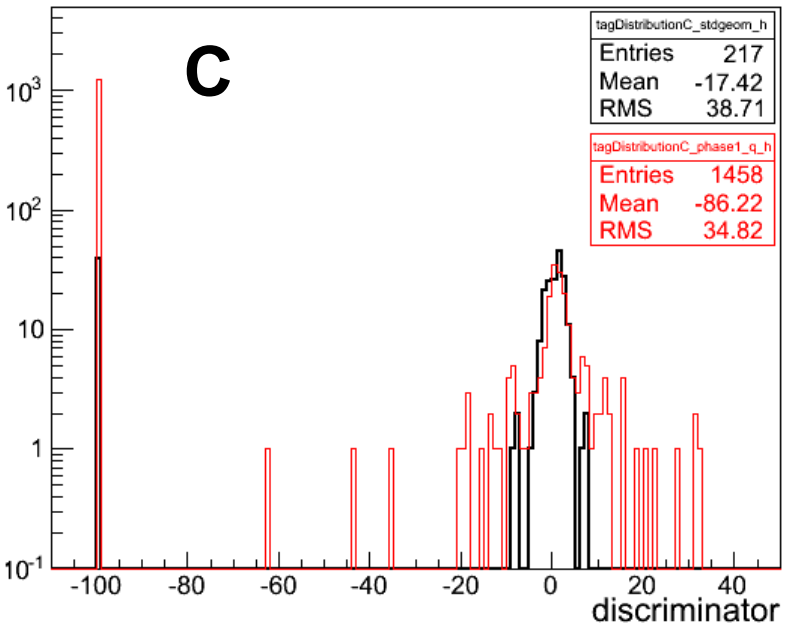
• remaining triplets seem to be mostly **junk**

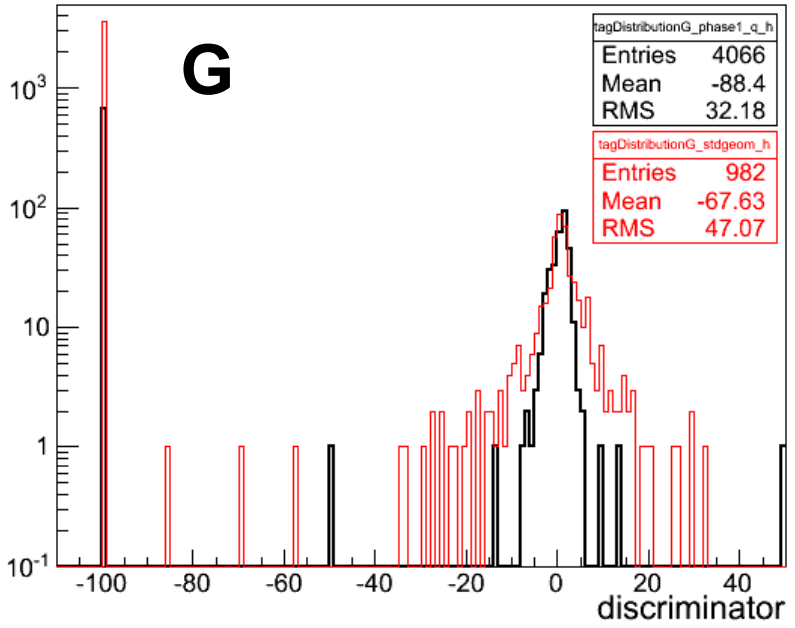
• doesn't help with b-tagging since the track/vertexing algorithms get **swamped with low p_T fake seeds,** many tracks don't pass quality cuts

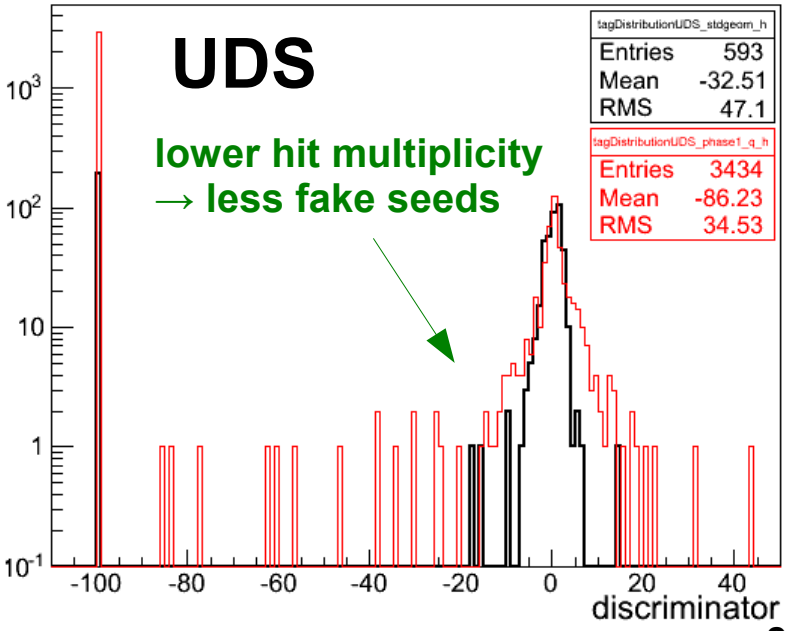
b-tagging discriminators: **quad+trip** (no data loss)

"untaggable" jets with < 2 tracks









Pixel-only b-tagging with merged quadruplets ?

Phase1 quadruplets significantly improve the pixel-only track quality –

- better IP and p_T resolutions, drastically decreased fake rate
- **but also lower track efficiency** (wrt. stdgeom triplets):
  - 4-hit requirement is more demanding
  - quadruplet coverage is not fully hermetic ("holes" around η = 1.4)

Improvements from IP resolution are compensated by efficiency losses –

- trackCounting HE algorithm is efficiency sensitive (and there's no alternative to it)

→ need to recover the track efficiency!

Possible approach: create "mixed" seed collections in 1-2 steps –

- Start from pairs → create triplets → expand to quadruplets (if matching hit)
- not possible with QuadrupletSeedMerger

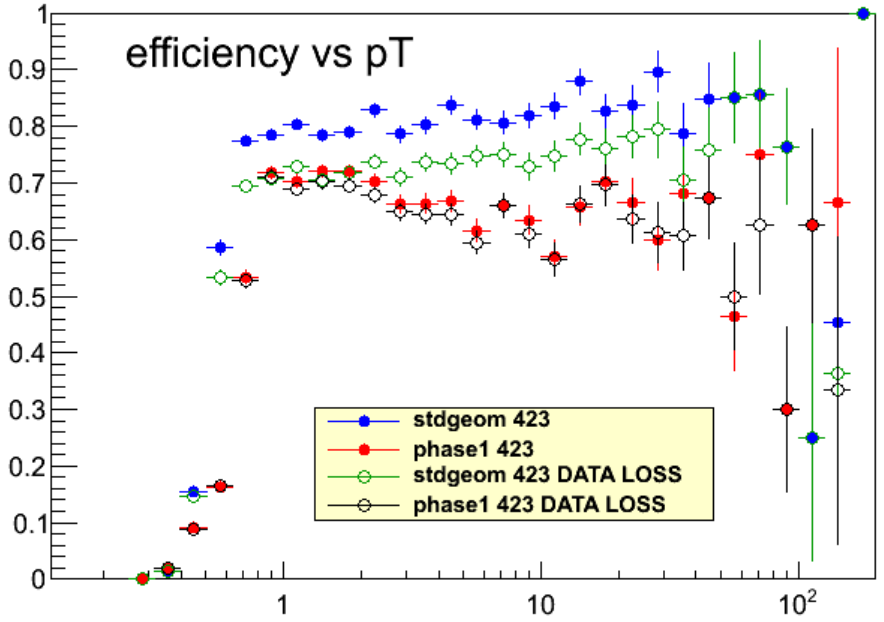
Currently evaluating possibilities of upgrading the seeding code –

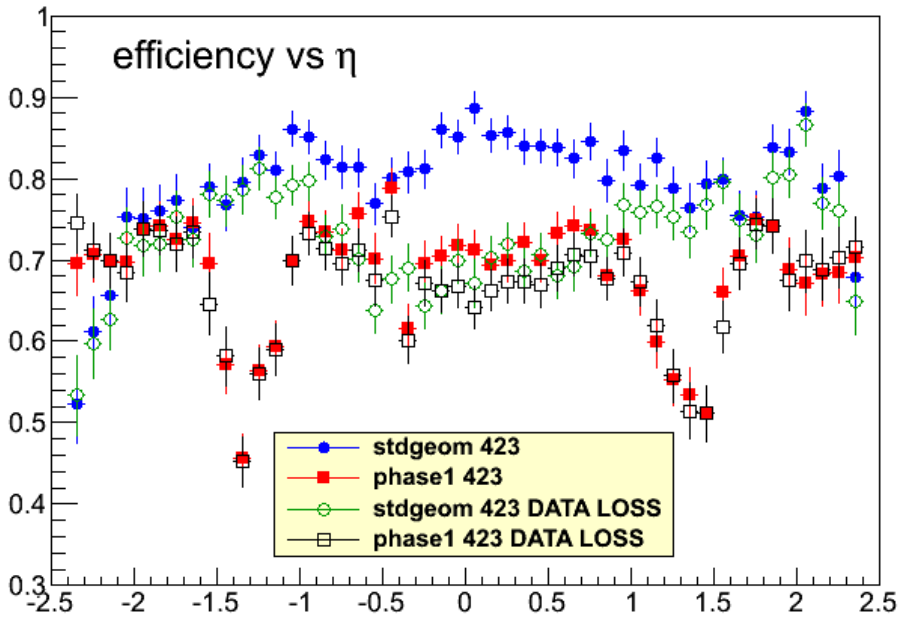
- Complex inheritance chains already in pixel-only, all-silicon code not yet looked at
- PF KDTree code might be very helpful, we would like to get involved!

Tracking efficiency: quadruplets (**with data loss**)

Introducing data loss has strong impact on stdgeom tracking efficiency –

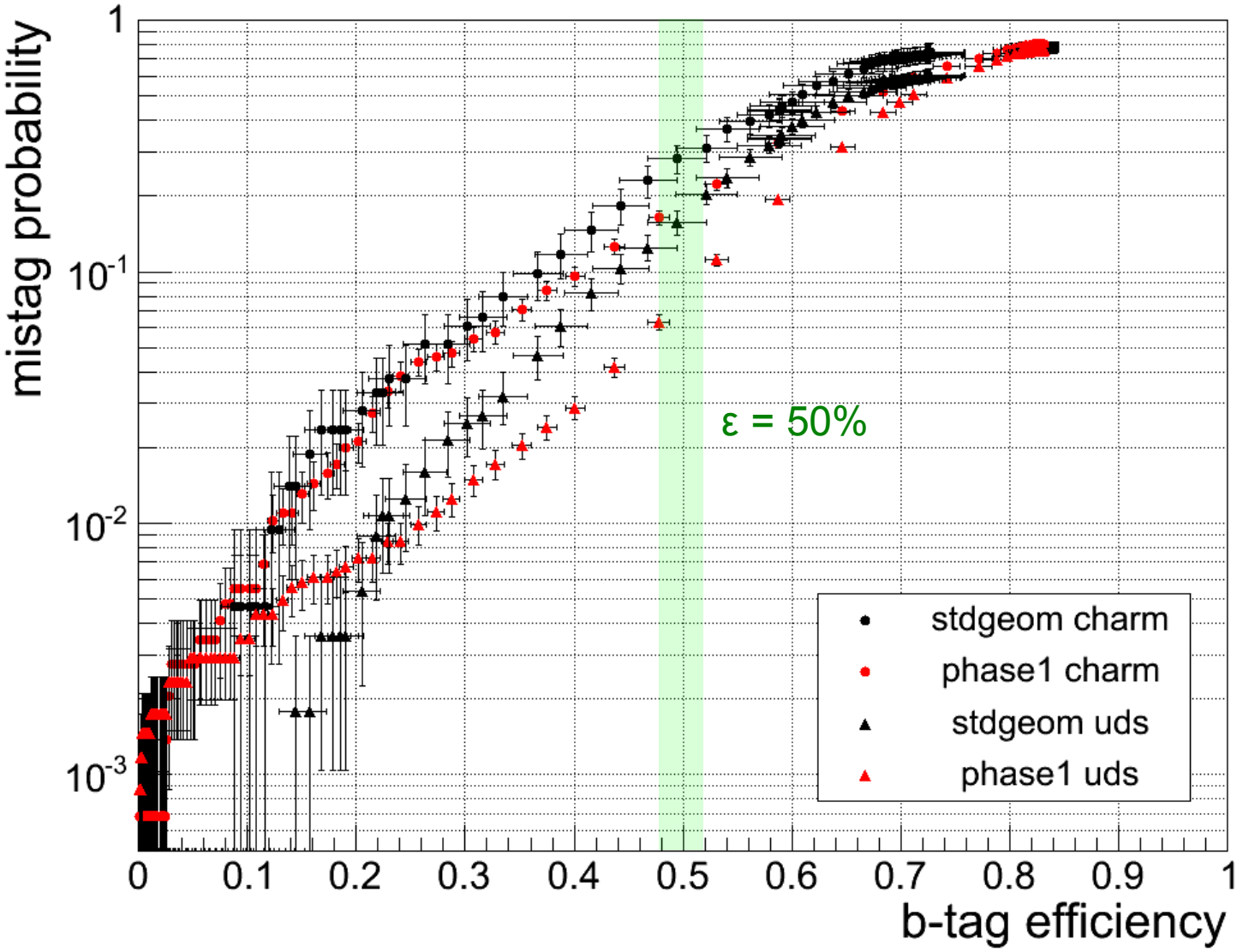
- BPix1: 16% (stdgeom), 4.7% (phase1)
- stdgeom efficiency loss in barrel up to ~ 15% (compare blue / green)
- phase1 efficiency (quadruplets) not much affected (compare red / **black**)





TrackCounting algorithm already known to be strongly affected by tracking efficiency

b-tagging efficiency: quadruplets (**with data loss**)



- altered tracking efficiencies have direct impact on b-tag performance
- for ε ≈ 50%, gain in fake rate is approx. a factor of 2
- effect is exclusively due to simulated BPix data loss

Summary

Phase1 pixel-only tracking with (merged) quadruplets is well understood –

track quality and efficiency, vertexing performance (DVF and DA),
b-tagging algorithms and performance, high pile-up, simulated data loss

Pure quadruplet seeds don't improve the pixel-only b-tag performance –

better IP resolution compensated by lower efficiency
(feature of the trackCounting algorithm)

Need for “genuine” quadruplet seed creation (avoid merging) –

- “mixed” triplet/quadruplet seed collections for higher efficiency
- extensive upgrade or replacement of the complex code seems unavoidable
- also important for speeding up the seeding with high PU

Pixel-only seeding code – inheritance & calling scheme