Long-Tracking at Upgrade 2

Renato Quagliani (CERN)

Using parameterizations developed by

Manuel Schiller (Glasgow)

Gluing in the reco sequence the 'new' pix-fibre TTracks finding by

Abhijit Mathad (CERN)

Using .sim files inputs generated by

Christoph Lagenbruch and Lennart Uecker (Heidelberg)

Disclaimer

- A python-only framework developed to run over any *.sim* file to perform full-tracker pattern recognition prototyping.
- Why? Some re-understanding of detectors and revision of current existing Run3 algorithm needed.

- pyReco not aiming to replace LHCb software, but for track reconstruction (no Kalman fit) prototyping is good.

Question: can we do long tracks, which eff and which ghost rates?

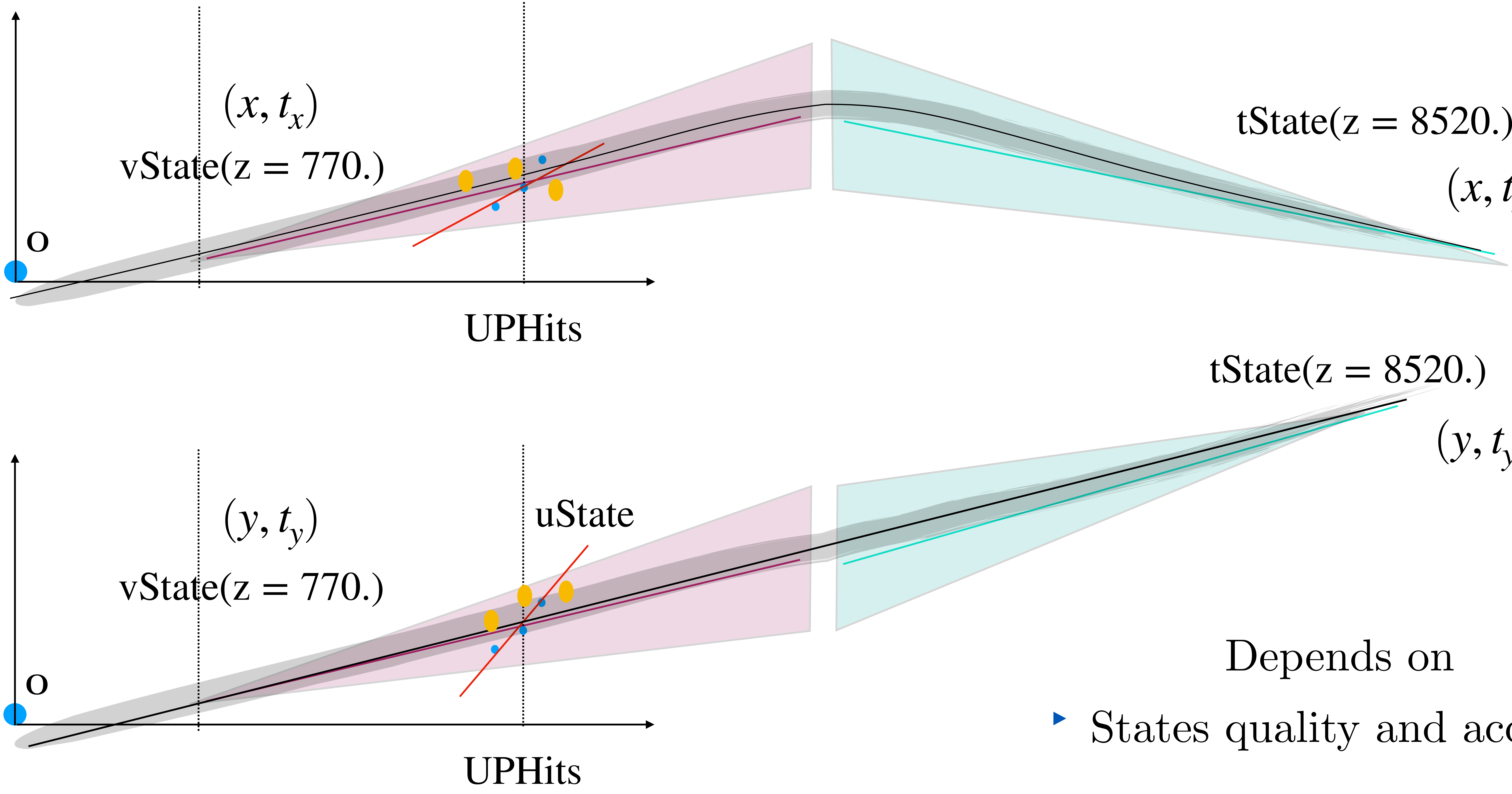
- A main driver of performance will be the presence of pixels measurements in MT and UP and how this talk with TV and MT in bending and non-bending plane.

- How to best exploit them?
  - *Strategy followed: (all of this enabling customizable tunable where possible)*
    - Build first standalone T-Tracks in MightyTracker
    - Build 'cheated' Velo segments : 100 % efficiency fitted with simply a 'line in yz/xz'
    - Match Velo-TTracks
    - Add and search UP hits in Matched Velo-TTracks [still finalizing this] and

Wrote prototypes of algorithms partly emulating Run3 , partly new

Results in the following are based on the L=1.5e34 and with FTDR MightyTracker

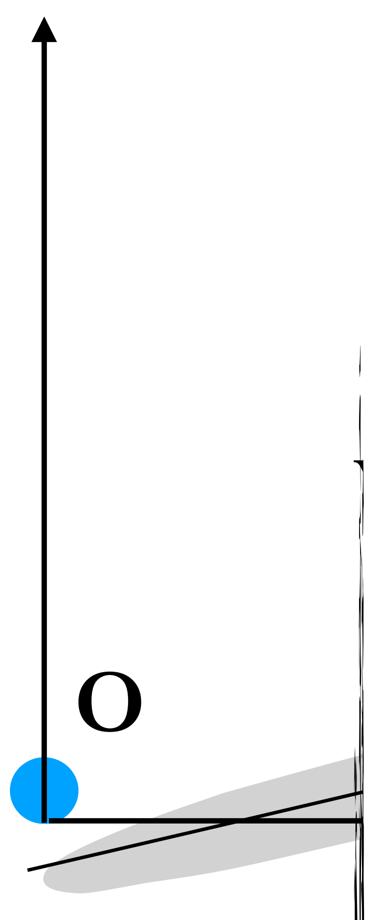
It's all a matter of states for Long-tracks making



Depends on

- States quality and accuracy'

It's all a matter of states [again]



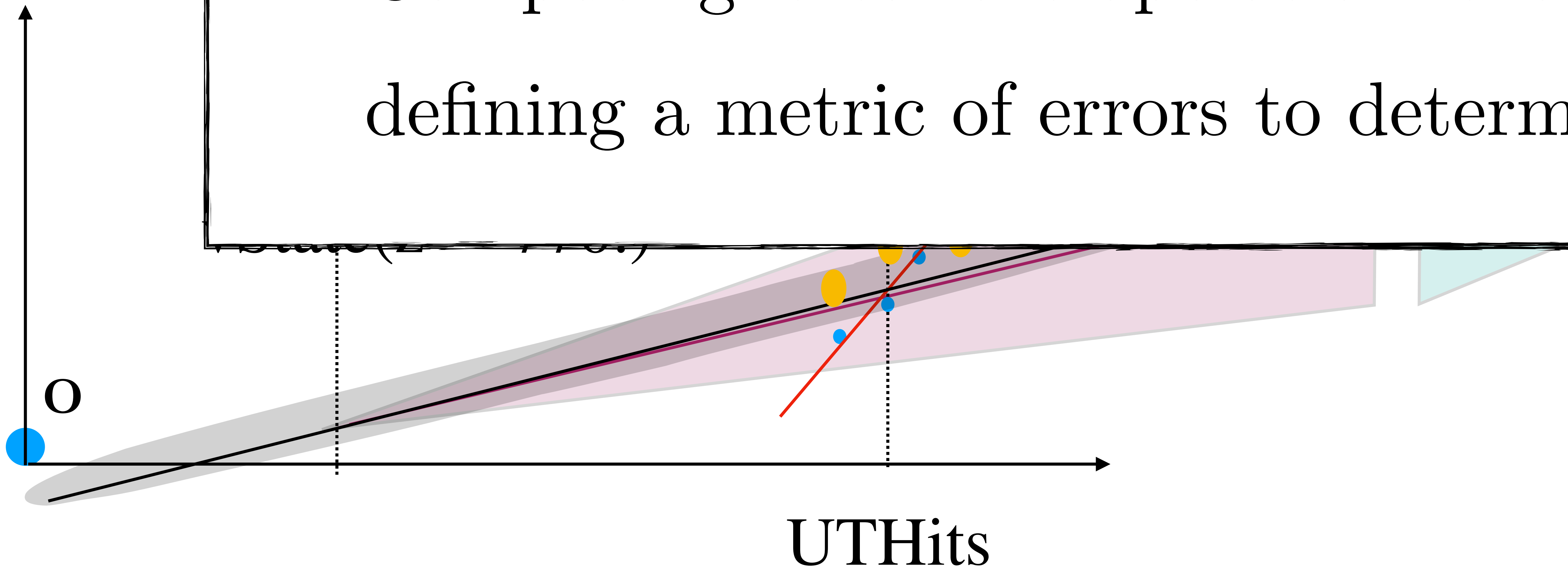
Can define a matching χ^2 based on tState and vState.

(And uState)

To do so, Manuel's tool to predict jumps between states

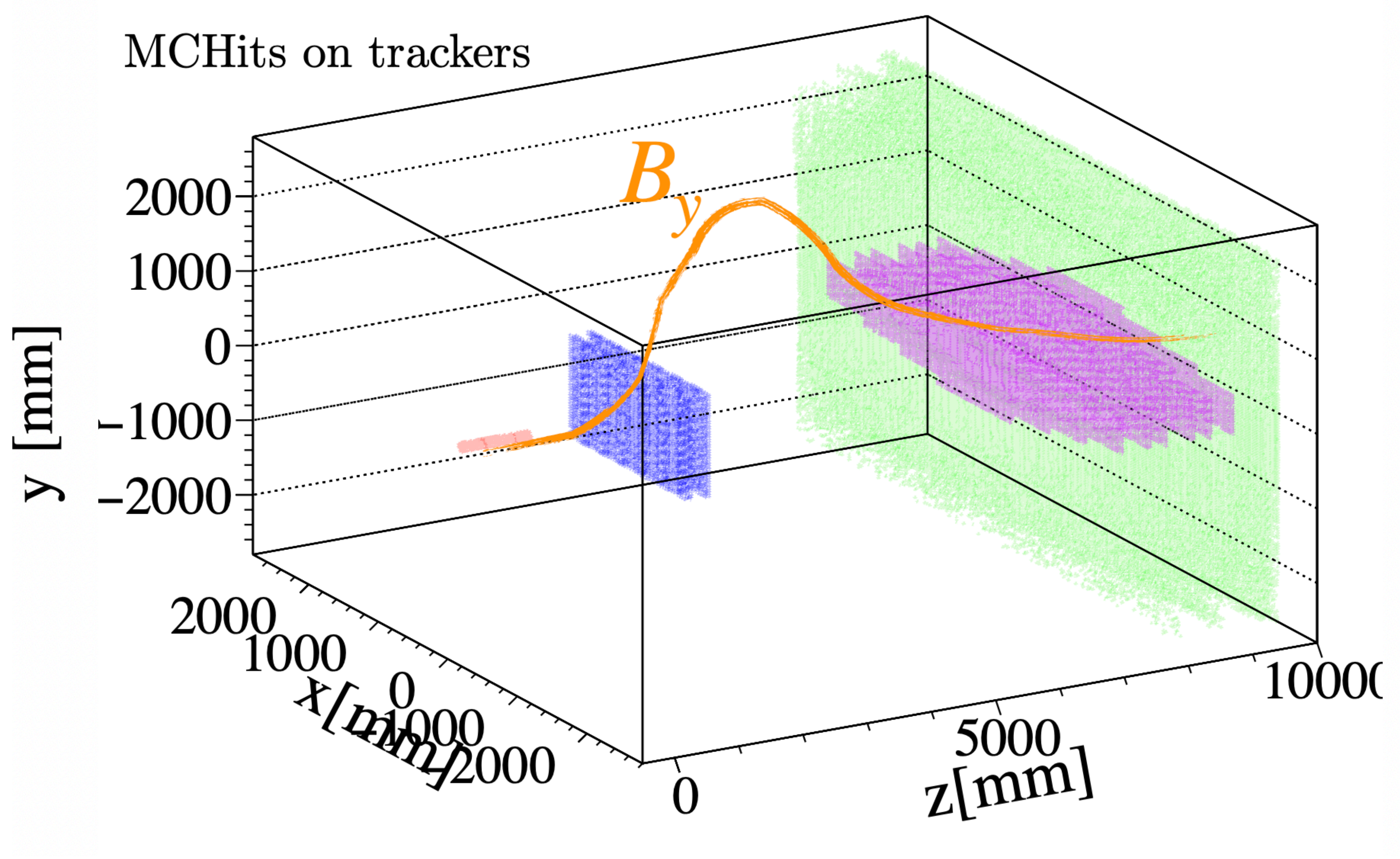
Computing 'after extrapolation' the deltas in (x,y,tx,ty) and

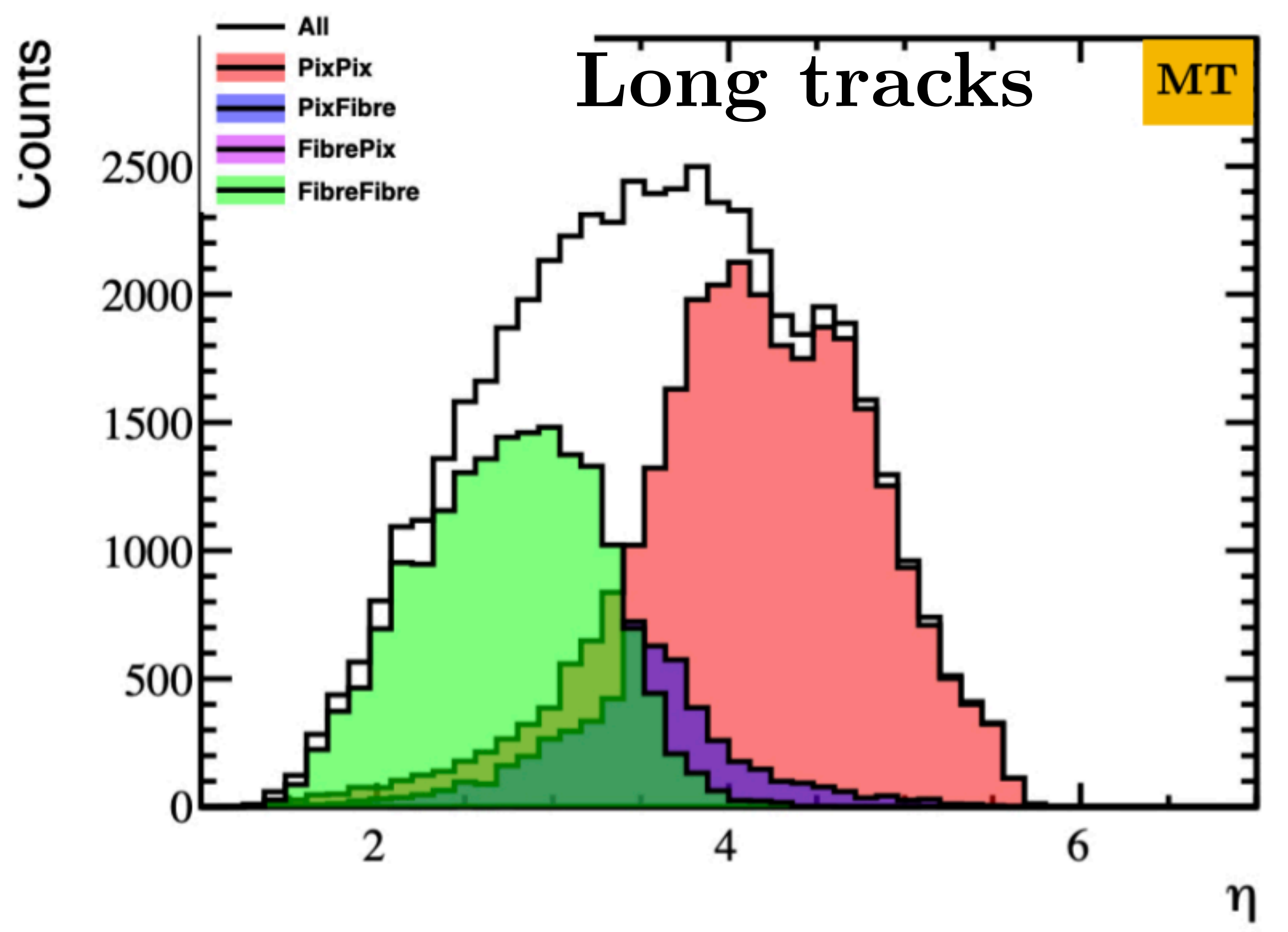
defining a metric of errors to determine a match-χ^2



(1) Build T-Track States

- In Mighty Tracker we have different acceptances, this will imply different reconstruction and track qualities. **PixPix, FibFib, PixFib** [see Abhijit talk]
- PixPix algorithm prototype presented at MT workshop, already ported in LHCb
- FibFib algorithm inspired to existing run3 one but rewrote with different logics
- PixFib, Abhijit's work as preliminary used





T-Tracks finding sets (in absence of forward tracking) the **upper limit** for long-track making (and downstream)

Note on (1) : TTracks finding and interplay with measurements

$y(z) = y_0 + t_y \cdot (z - z_{ref})$

$x(z) = x_0 + t_x \cdot (z - z_{ref}) + c_x \cdot (z - z_{ref})^2(1 + \boxed{\text{dRatio}}(z - z_{ref}))$

$\sigma_x^{fibre} \sim 100\mu\text{m}$

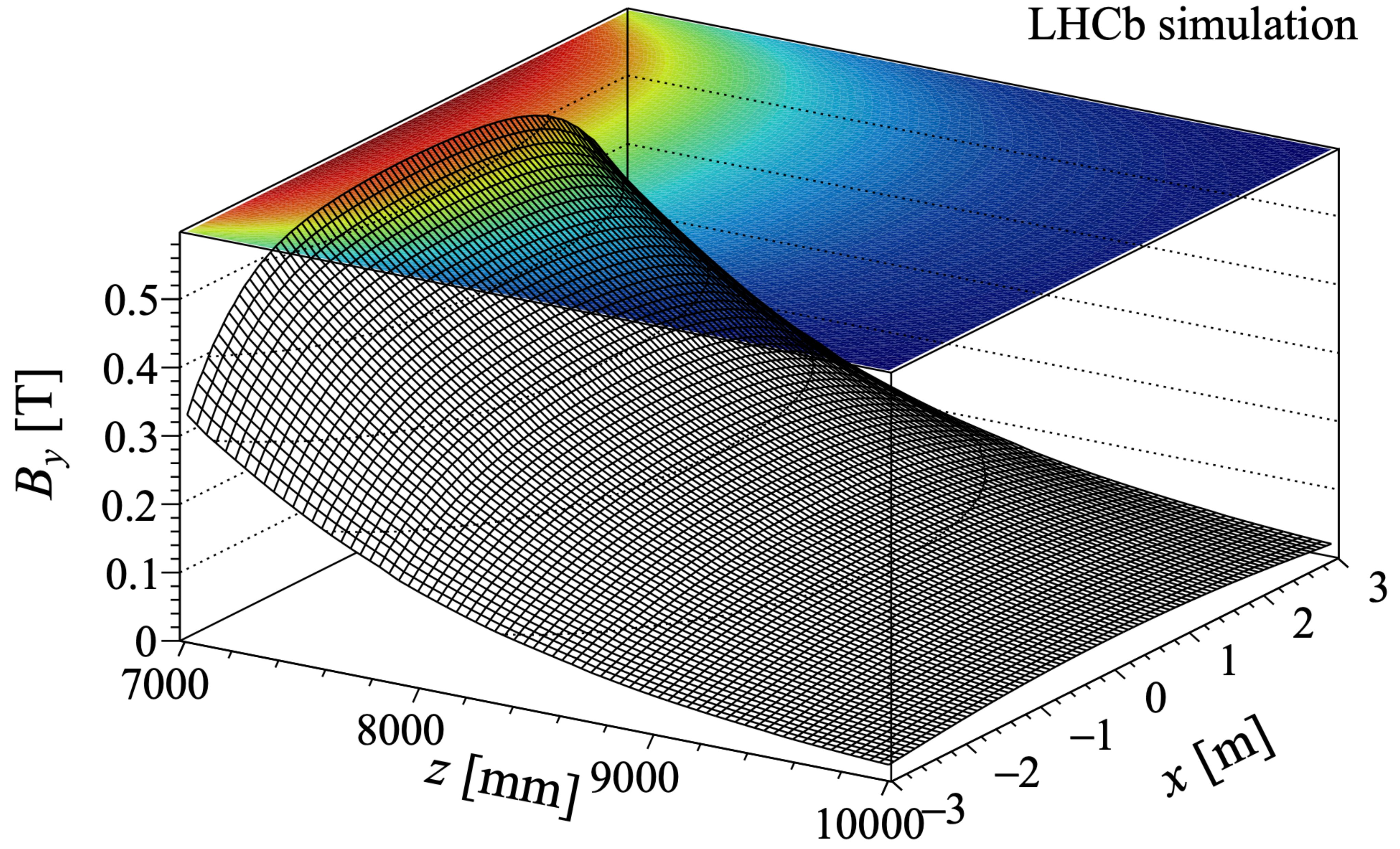
$\sigma_y^{fibre} \sim 1.15\text{mm} \quad (\sigma_x/\sin(\alpha))$

$\sigma_x^{pix} \sim \frac{50\mu\text{m}}{\sqrt{(12)}}$

$\sigma_y^{pix} \sim \frac{150\mu\text{m}}{\sqrt{(12)}}$

$B_y = B_0 + B_1 \cdot z$

$B_y(x, y) \sim B_0(x, y) + B_1(x, y) \cdot z$



- **IF** $\frac{B_1}{B_0}(x, y)$ = **Constant for any (x,y) 3 hits enough to make a good stub**
- Currently (and Run3) set to Constant : effectively it tells the Track to bend more between T1 and T2 than T2 to T3
- The more precisely we get the T-Track state the better the predictions for matching χ^2 but also $\frac{x}{X_0}$ in MT can corrupt
- Interplay on nHits [i.e fit model nDOF] to use and actual resolution and multiple scattering between layers
- I think it's good that we revisit this given improved resolutions in pixel, can we do better , likely for high momentum no.
- [this do not link to the actual fake rate one would get, but more on the accuracy of our track states at pattern reco level]

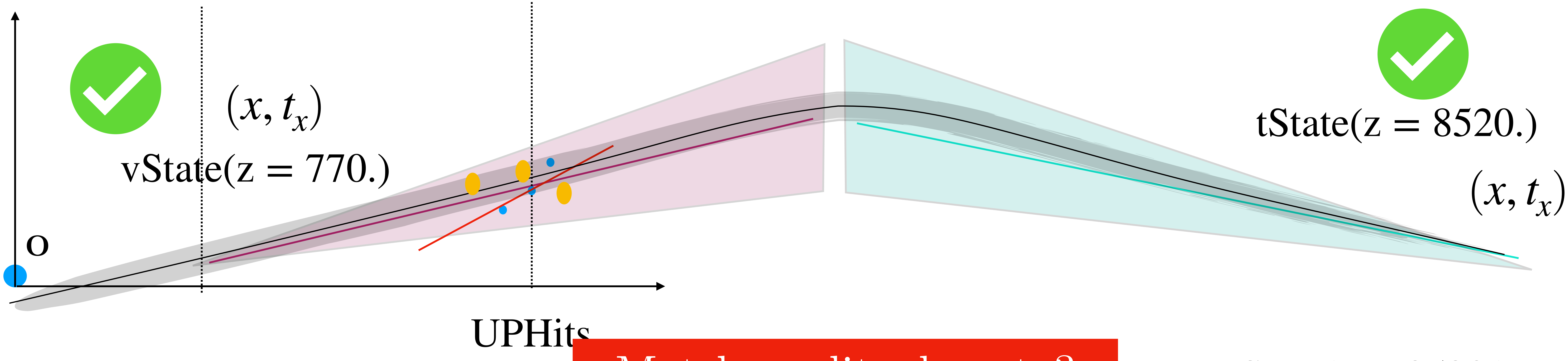
TTracks making results on long-segments

# Events	# Recoed Tracks	# Fake Tracks	Ghost Rate (evt-avg)
100	76936	8468	11.01 % (9.01)

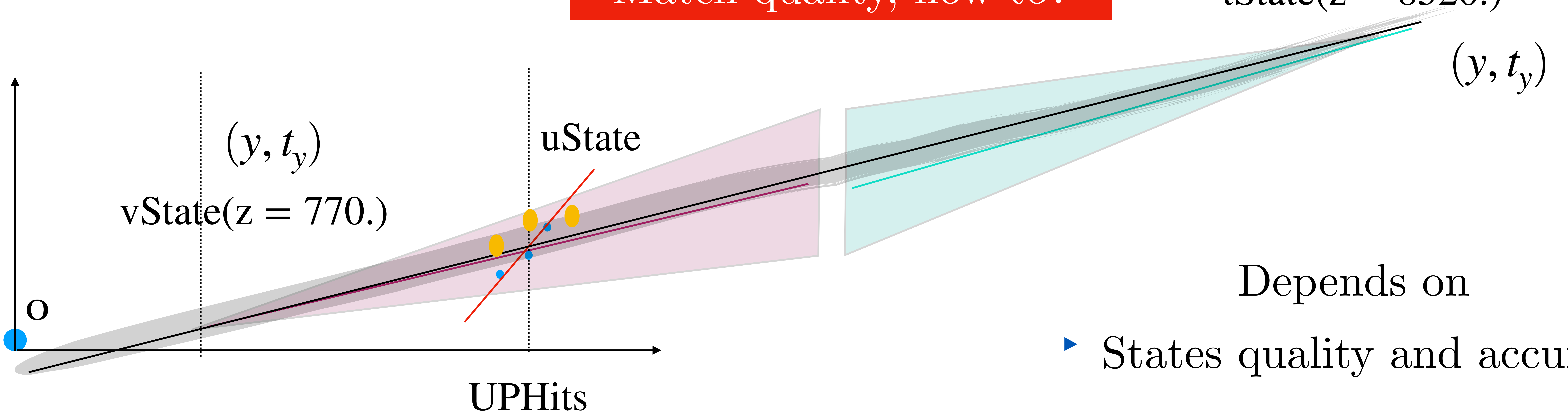
Efficiency is wrt the FULL reconstructed long tracks container.

Efficiency is a convolution of acceptance and efficiency here

It's all a matter of states for Long-tracks making



Match-quality, how to?



Depends on

- States quality and accuracy'

Match χ^2

- The match χ^2 is a primary observable we construct able to distinguish signal to background when finding pairs of segments upstream and downstream
- Defines the 'goodness' of a match found and how 'smooth' the track behaves
- At the pattern recognition level, we can't do a Kalman fit with material and B field, neither navigate the B field.
  - RTA costs will explode if so
  - We rely on those parameterizations of 'how a perfect track' would behave in the field

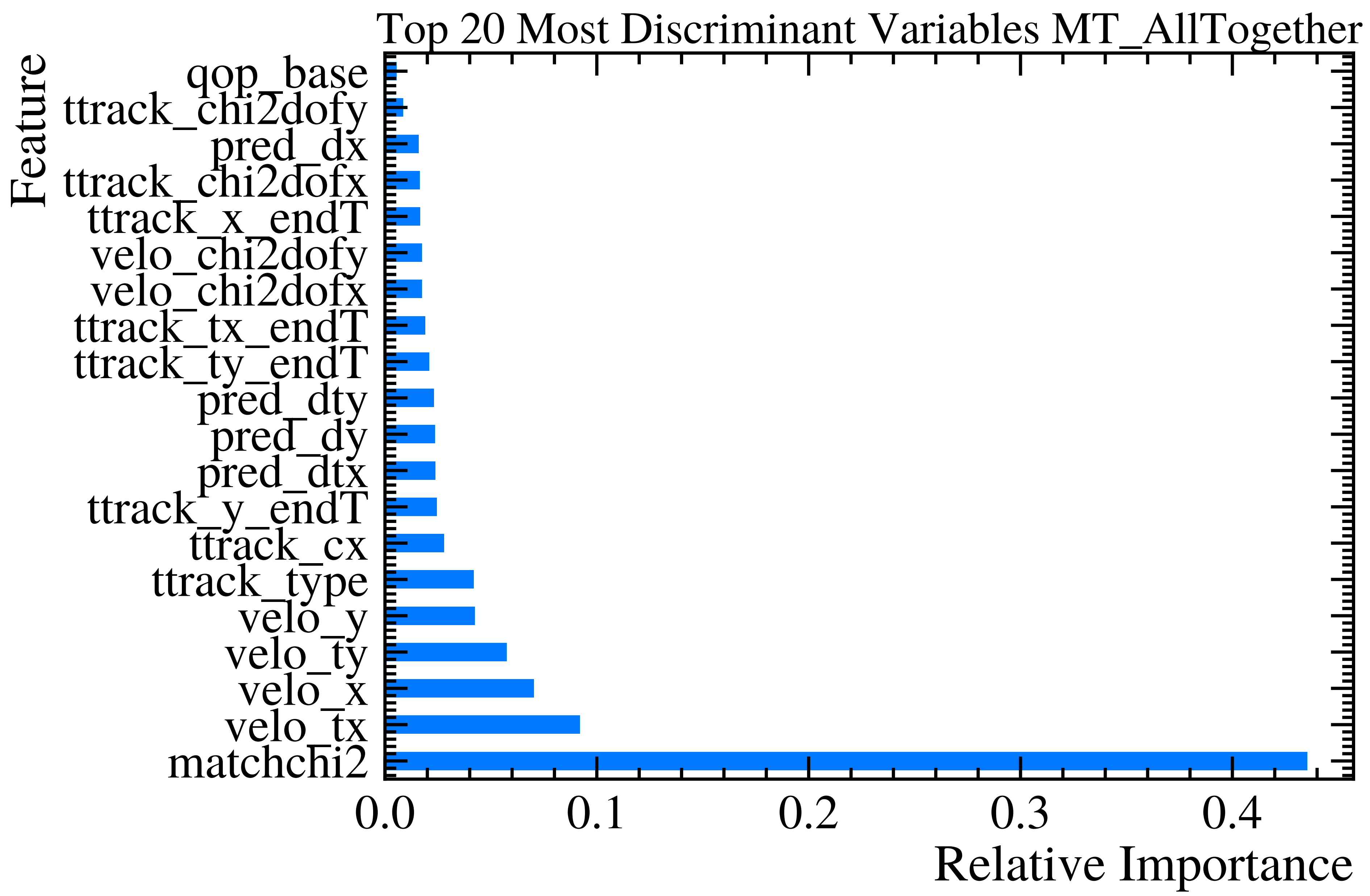
$$\chi^2_{match}[\text{Velo} - \text{TTrack}] = \left(\frac{x_{pred} - x_{vstate}}{\sigma_x^{pred}}\right)^2 + \left(\frac{y_{pred} - y_{vstate}}{\sigma_y^{pred}}\right)^2 + \left(\frac{ty_{pred} - ty_{vstate}}{\sigma_{ty}^{pred}}\right)^2 + \left(\frac{tx_{pred} - tx_{vstate}}{\sigma_{tx}^{pred}}\right)^2$$

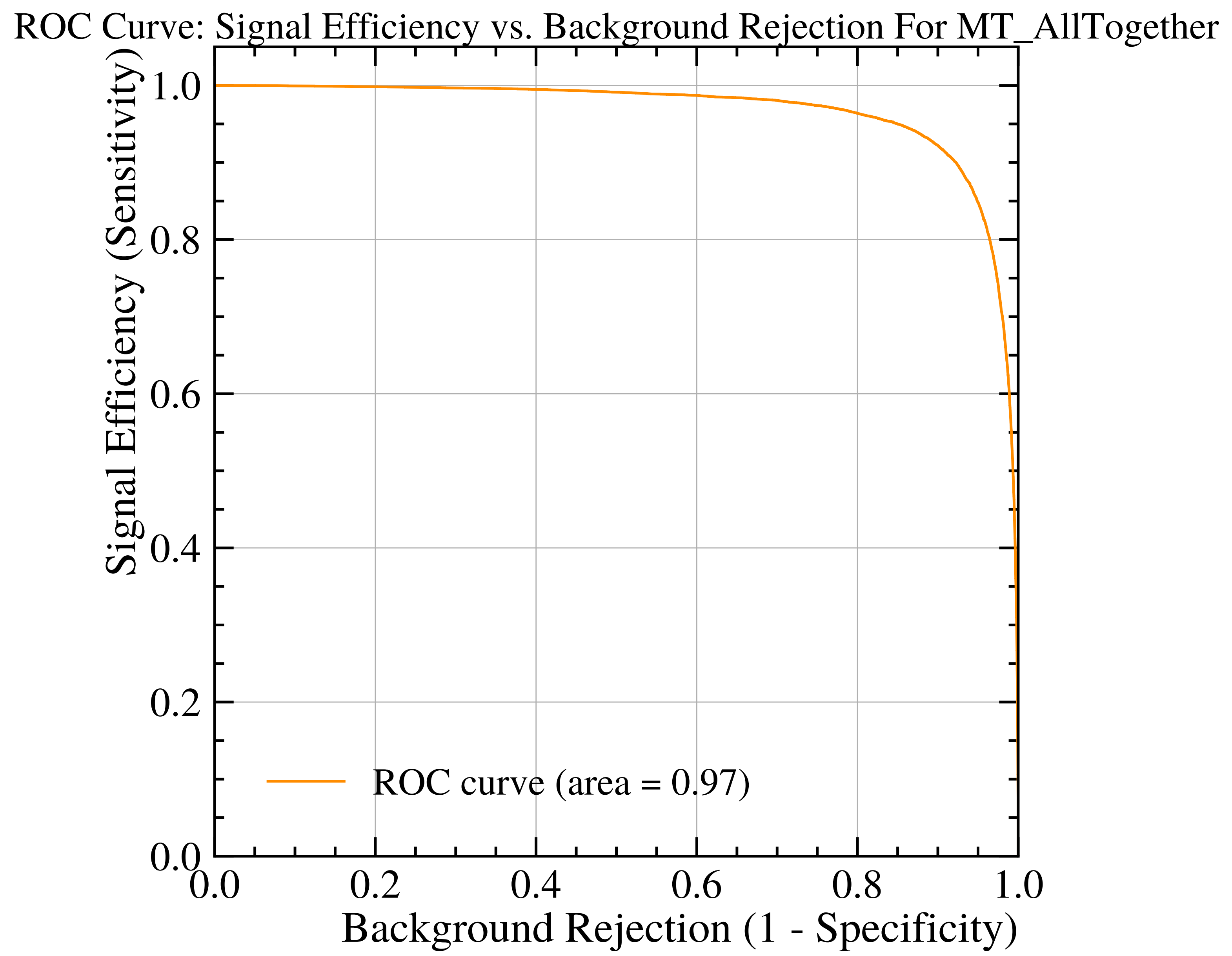
Selection based on χ^2_{Match}

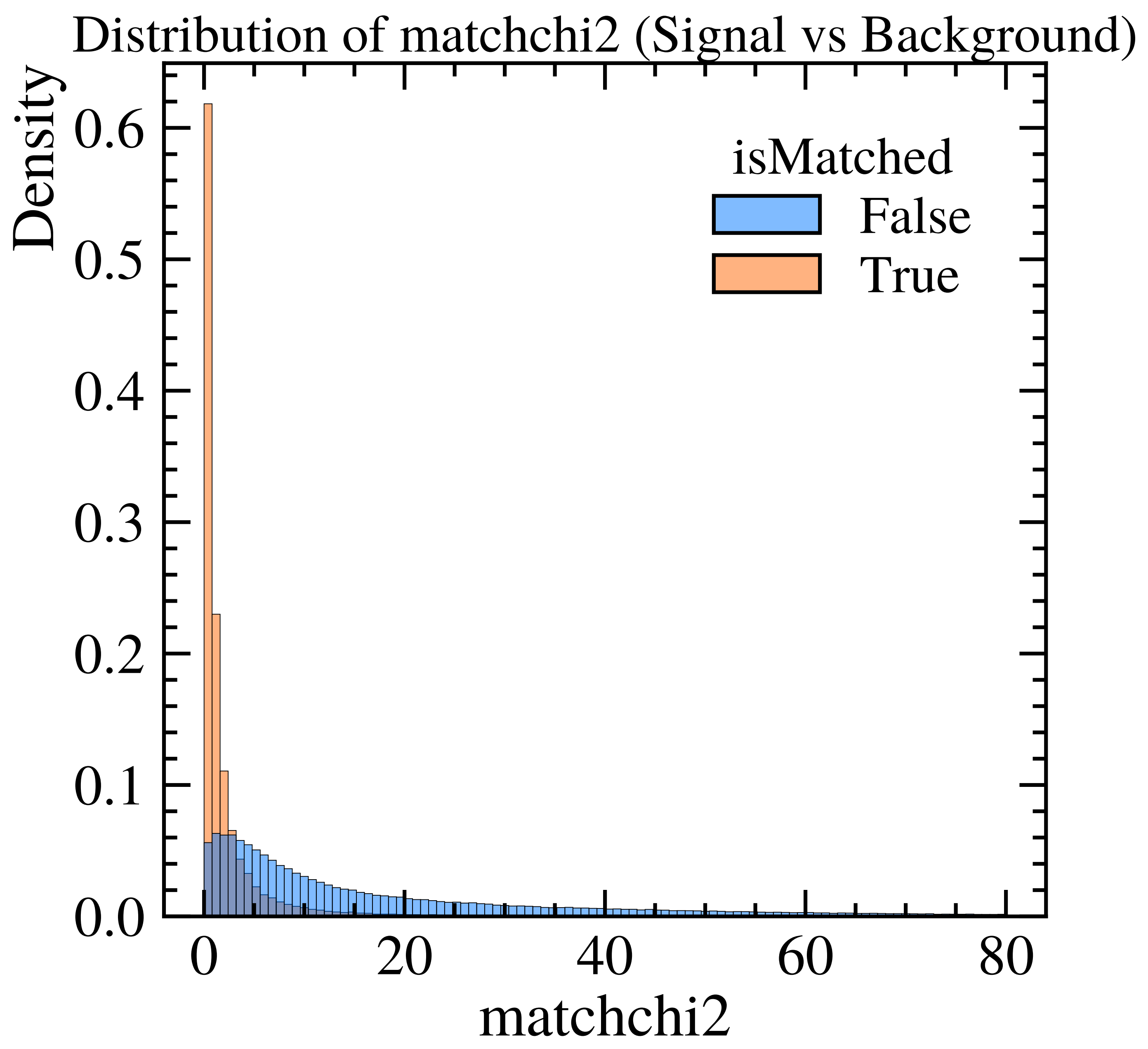
- For each **v** in Velo:
  - For each **t** in TTrack:
    - [**v**,[**t**.....]] χ^2 match, pick the first best 5 (tunable)
    - With the truth and failed matched with best 5 a `xgboost` MVA is trained here.
    - A lot of variables included, need further polishing, but good enough for a first go.
    - Point is: TTracks has new 'resolution' states, the NN must be retuned

NN training [for TTrack-Velo only matches]

- Tried, a splitted or not by TTrack type NN , found not a big difference, to be exploited more.

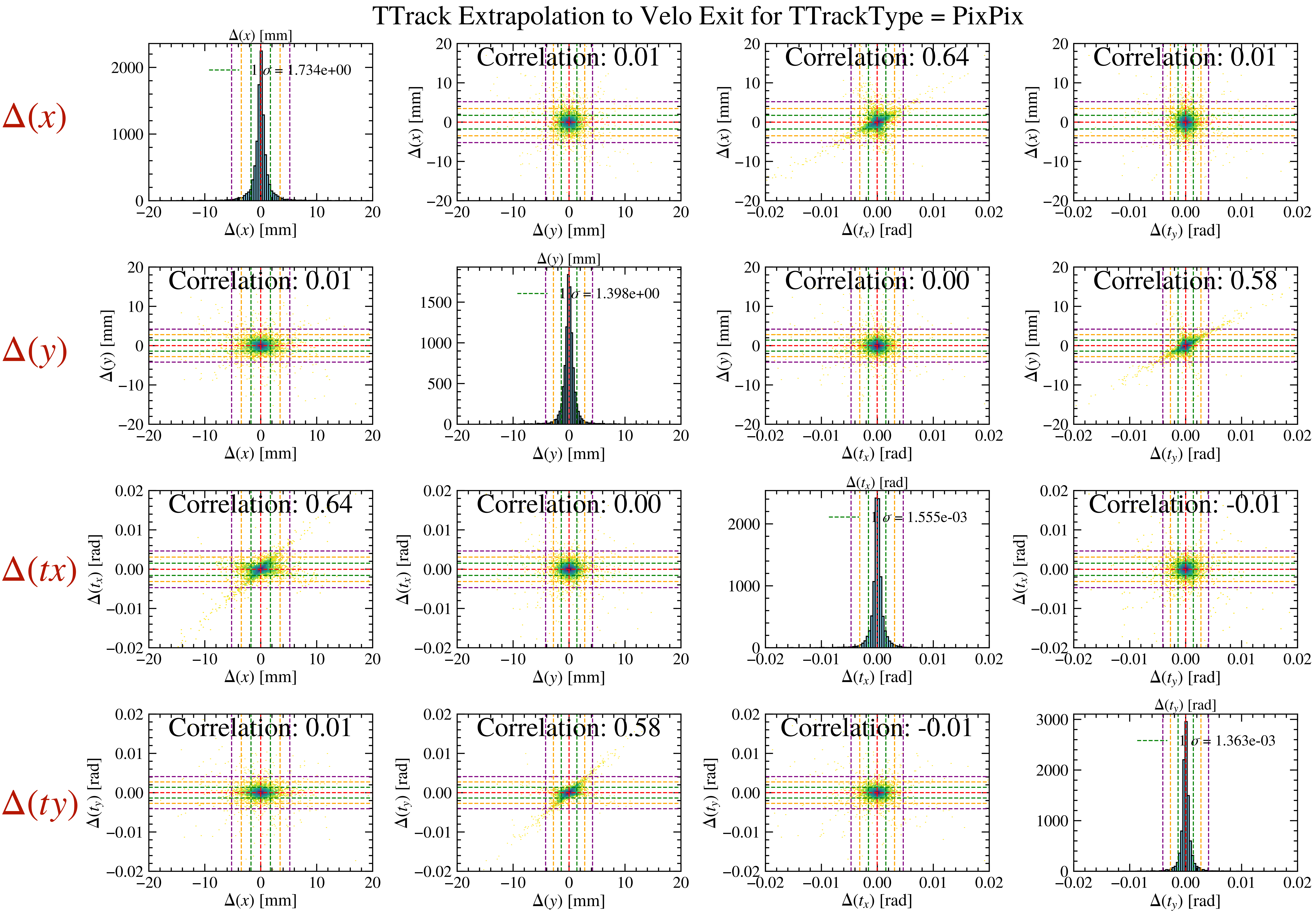






- The best 5 chi2 match get assigned this NN score, and a cut at 0.005 is applied [very loose]. Found a non negligible loss in performance if tighter, fakes go down but also efficiencies (from 90% to 70% efficiency with ghost from 50% to 30 %)

Match chi2 definition, shooting back TTrack to Velo



TTrack Pixel in Pixel

$\sigma_x = 1.734\text{mm}$

$\sigma_y = 1.40\text{mm}$

$\sigma_{t_x} = 1.555\text{mrad}$

$\sigma_{t_y} = 1.363\text{mrad}$

TTrack Fibre in Fibre

$\sigma_x = 2.74\text{mm}$

$\sigma_y = 3.11\text{mm}$

$\sigma_{t_x} = 2.425\text{mrad}$

$\sigma_{t_y} = 2.035\text{mrad}$

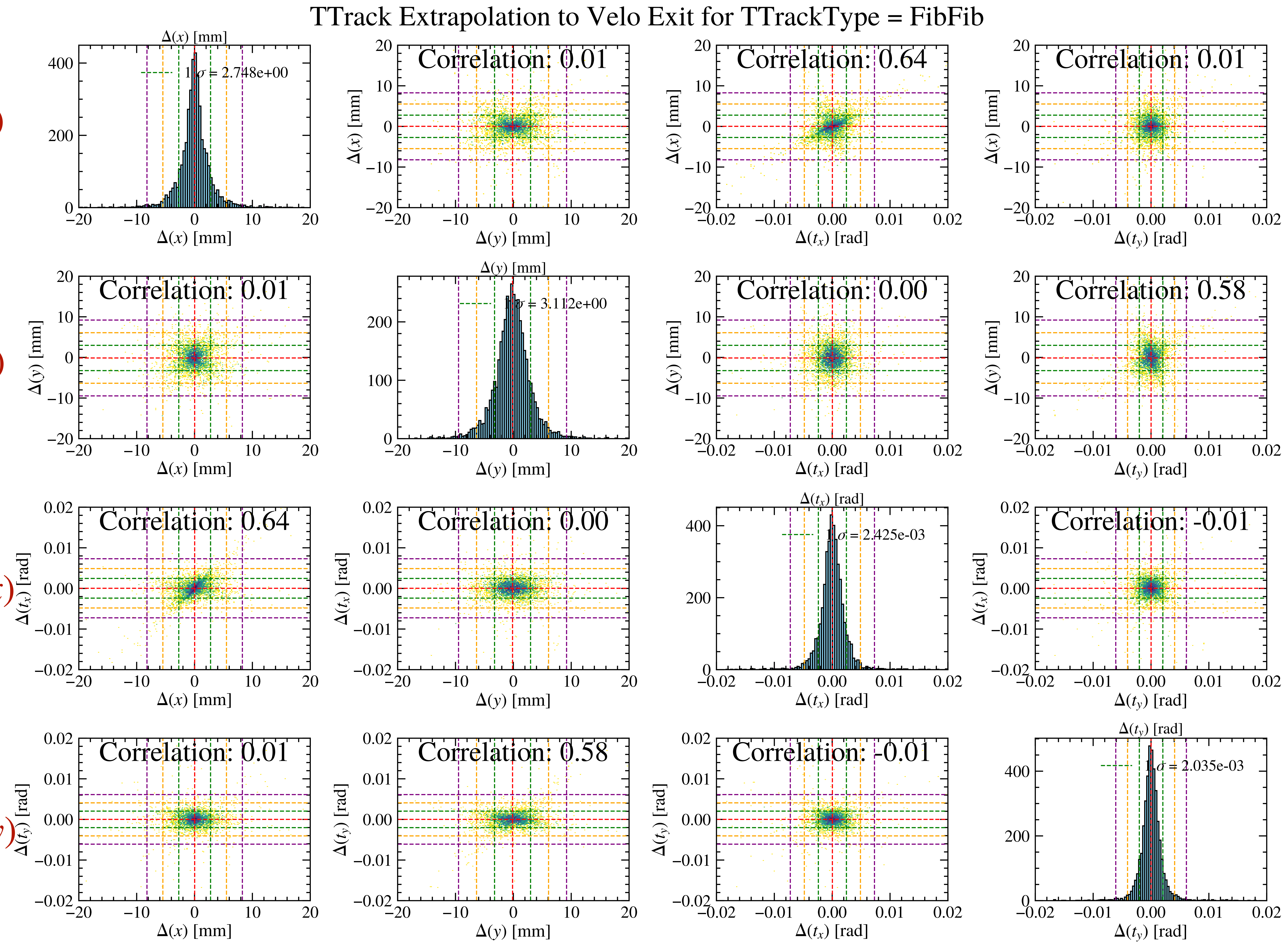
Match chi2 definition, shooting back TTrack to Velo

$\Delta(x)$

$\Delta(y)$

$\Delta(tx)$

$\Delta(ty)$



TTrack Pixel in Pixel

$\sigma_x = 1.734\text{mm}$

$\sigma_y = 1.40\text{mm}$

$\sigma_{t_x} = 1.555\text{mrad}$

$\sigma_{t_y} = 1.363\text{mrad}$

TTrack Fibre in Fibre

$\sigma_x = 2.74\text{mm}$

$\sigma_y = 3.11\text{mm}$

$\sigma_{t_x} = 2.425\text{mrad}$

$\sigma_{t_y} = 2.035\text{mrad}$

Use of NN for final tracks selection in absence of UP addition

- For each **v** in Velo:
  - For each **t** in TTrack:
    - [**v**,[**t**.....]] χ^2 match, pick the first best 5 (tunable) to train a NN
    - Assing NNscore, keep best 5 [can be tuned] NNscores
    - Construct inverse relation [t, [(v1,score),....] sorted from highest to lower
- Make matches for each single TTrack with best score velo segment
  - Keep multiple solutions if the second best score is good as well based to distance in score from best one [tunables]
- Matches Velo-TTracks found at this point, no UP added

Performance up to here (matching noUP)

- We have no UP added here which is expected to give additional discrimination power. But let's have a look to performance obtained.

LongTracks Tracks PrChecker

Category	# Recoed / # Recoble	Efficiency (%)	Clones (%)
Long (all)	39358 / 54008	72.87 %	257 (0.65 %) [nTotMatch = 39411])
Long eta[2,5]	37034 / 49379	75.00 %	257 (0.69 %) [nTotMatch = 37087])
Long eta[2,5], p>3GeV	34215 / 41517	82.41 %	230 (0.67 %) [nTotMatch = 34262])
Long eta[2,5], p>5GeV	26887 / 30935	86.91 %	116 (0.43 %) [nTotMatch = 26914])
Long_decay (all)	36492 / 47985	76.05 %	186 (0.51 %) [nTotMatch = 36532])
Long_decay eta[2,5]	34262 / 43749	78.31 %	186 (0.54 %) [nTotMatch = 34302])
Long_decay eta[2,5], p>3GeV	31691 / 37187	85.22 %	170 (0.54 %) [nTotMatch = 31728])
Long_decay eta[2,5], p>5GeV	25053 / 28139	89.03 %	84 (0.33 %) [nTotMatch = 25075])
(!e) Long_decay (all)	36351 / 47720	76.18 %	186 (0.51 %) [nTotMatch = 36391])
(!e) Long_decay eta[2,5]	34131 / 43533	78.40 %	186 (0.54 %) [nTotMatch = 34171])
(!e) Long_decay eta[2,5], p>3GeV	31691 / 37187	85.22 %	170 (0.54 %) [nTotMatch = 31728])
(!e) Long_decay eta[2,5], p>5GeV	24965 / 28016	89.11 %	84 (0.34 %) [nTotMatch = 24987])
(!e) LongUT_decay (all)	34597 / 44958	76.95 %	186 (0.54 %) [nTotMatch = 34637])
(!e) LongUT_decay eta[2,5]	34097 / 43431	78.51 %	186 (0.54 %) [nTotMatch = 34137])
(!e) LongUT_decay eta[2,5], p>3GeV	31659 / 37103	85.33 %	170 (0.54 %) [nTotMatch = 31696])
(!e) LongUT_decay eta[2,5], p>5GeV	24937 / 27957	89.20 %	84 (0.34 %) [nTotMatch = 24959])
(!e) LongUT_decay (Pix-Pix)	21363 / 24372	87.65 %	8 (0.04 %) [nTotMatch = 21364])
(!e) LongUT_decay (Pix-Pix) eta[2,5]	21110 / 23942	88.17 %	8 (0.04 %) [nTotMatch = 21111])
(!e) LongUT_decay (Pix-Pix) eta[2,5], p>3GeV	20953 / 23115	90.65 %	8 (0.04 %) [nTotMatch = 20954])
(!e) LongUT_decay (Pix-Pix) eta[2,5], p>5GeV	18785 / 20219	92.91 %	0 (0.00 %) [nTotMatch = 18785])
(!e) LongUT_decay (Fib-Fib)	11372 / 16575	68.61 %	40 (0.35 %) [nTotMatch = 11381])
(!e) LongUT_decay (Fib-Fib) eta[2,5]	11127 / 15533	71.63 %	40 (0.36 %) [nTotMatch = 11136])
(!e) LongUT_decay (Fib-Fib) eta[2,5], p>3GeV	9038 / 11093	81.47 %	34 (0.38 %) [nTotMatch = 9046])
(!e) LongUT_decay (Fib-Fib) eta[2,5], p>5GeV	5038 / 5997	84.01 %	22 (0.44 %) [nTotMatch = 5044])
(!e) LongUT_decay (Pix-Fib)	1862 / 4011	46.42 %	138 (7.29 %) [nTotMatch = 1892])
(!e) LongUT_decay (Pix-Fib) eta[2,5]	1860 / 3956	47.02 %	138 (7.30 %) [nTotMatch = 1890])
(!e) LongUT_decay (Pix-Fib) eta[2,5], p>3GeV	1668 / 2895	57.62 %	128 (7.55 %) [nTotMatch = 1696])
(!e) LongUT_decay (Pix-Fib) eta[2,5], p>5GeV	1114 / 1741	63.99 %	62 (5.49 %) [nTotMatch = 1130])

# Events	# Recoed Tracks	# Fake Tracks	Ghost Rate
100	66384	26901	40.52 %

NB: those numbers has to be compared with Lennart algorithms, before doing any kaman fit and without UP added or used for tracking

Fake rate very high due to the large combinatoric for matching

Now let's add UP

- A very preliminary code to add UP hits as pixels developed. Use parameterization from Manuel to predict the (x,y) location from Velo state and from Track state.

- A reference location in UP z = 2500 is used to determine a UP-state

$\text{TTrack}_{state} = (x, y, t_x, t_y)$

$UP_{state} = (x, y, t_x, t_y)$ $\quad \text{TTrack}^{pred}_{state} = (x, y, t_x, t_y)$ $\quad \text{Velo}^{pred}_{state} = (x, y, t_x, t_y)$

Use those info
To build a second NN

$\text{Velo}_{state} = (x, y, t_x, t_y)$

(Use Manual predictions)

Now let's add UP

- A very preliminary code to add UP hits as pixels developed. Use parameterization from Manuel to predict the (x,y) location from Velo state

5x5 mm windows from prediction in Velo , increasing to 35 mm when R(hit) goes to 1 meter
Pass over the 4-z locations in a given station and select the 1 best closest to prediction in R
Create a stub, can have 2,3,4 fired planes
A special fit is performed : always line in yz , cubic in xz if 4 planes, line if 3 or 2
Fit chi2 and outlier removal to inspect better as well windows

- Predictions to collect from velo to UT with q/p of the match using Manuel code
- Given Velo-TTrack match, a single found ***UPState*** is added (x,y,tx,ty)
- Not yet exploiting for for 3 or 4 planes the possible curvature information.
- If we have 2 planes UP, we can only make a line, with 3 do line model as well.

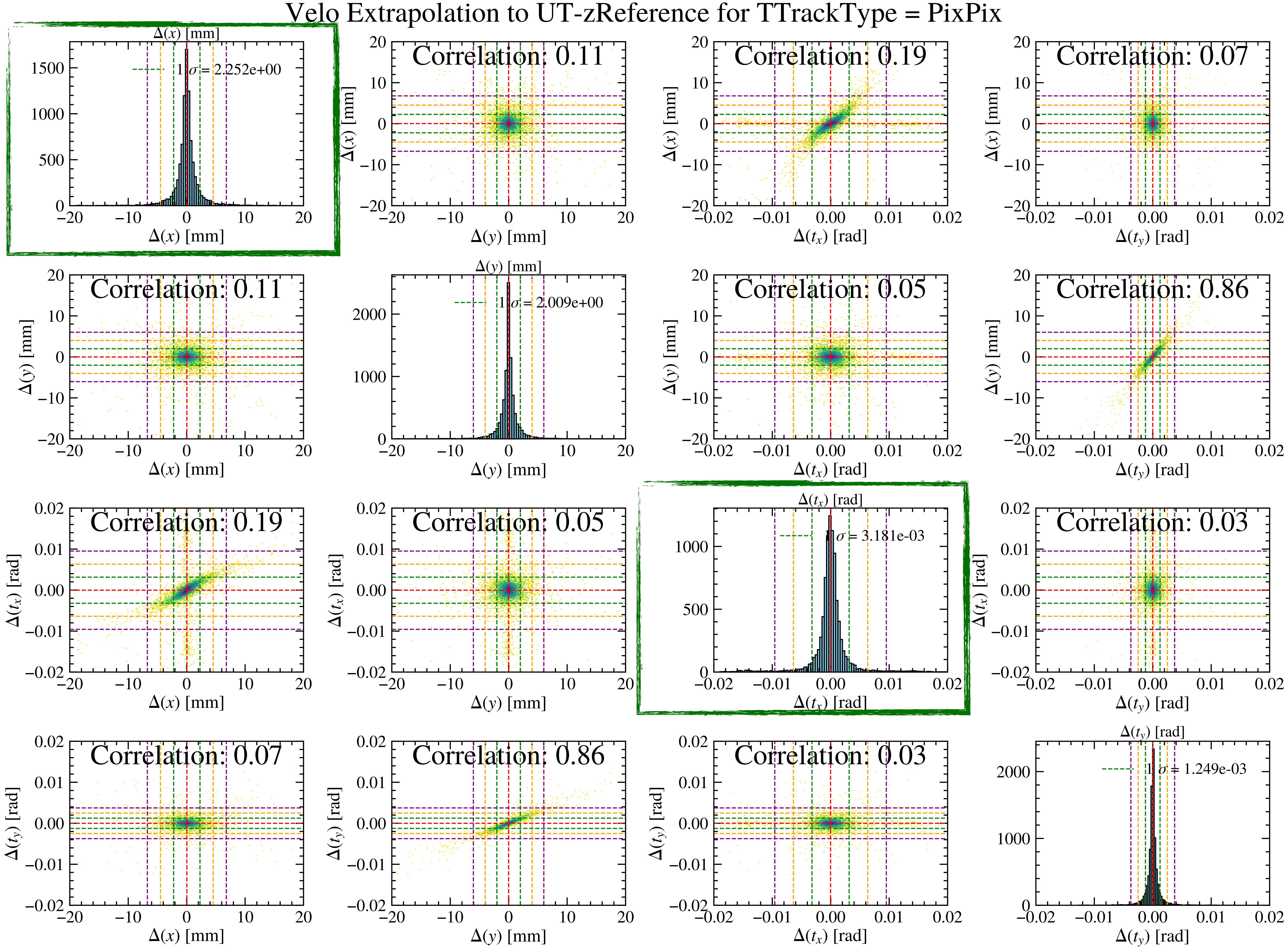
Predictions to UT from Velo side given matched TTrack-Velo

Δ(x)

Δ(y)

Δ(tx)

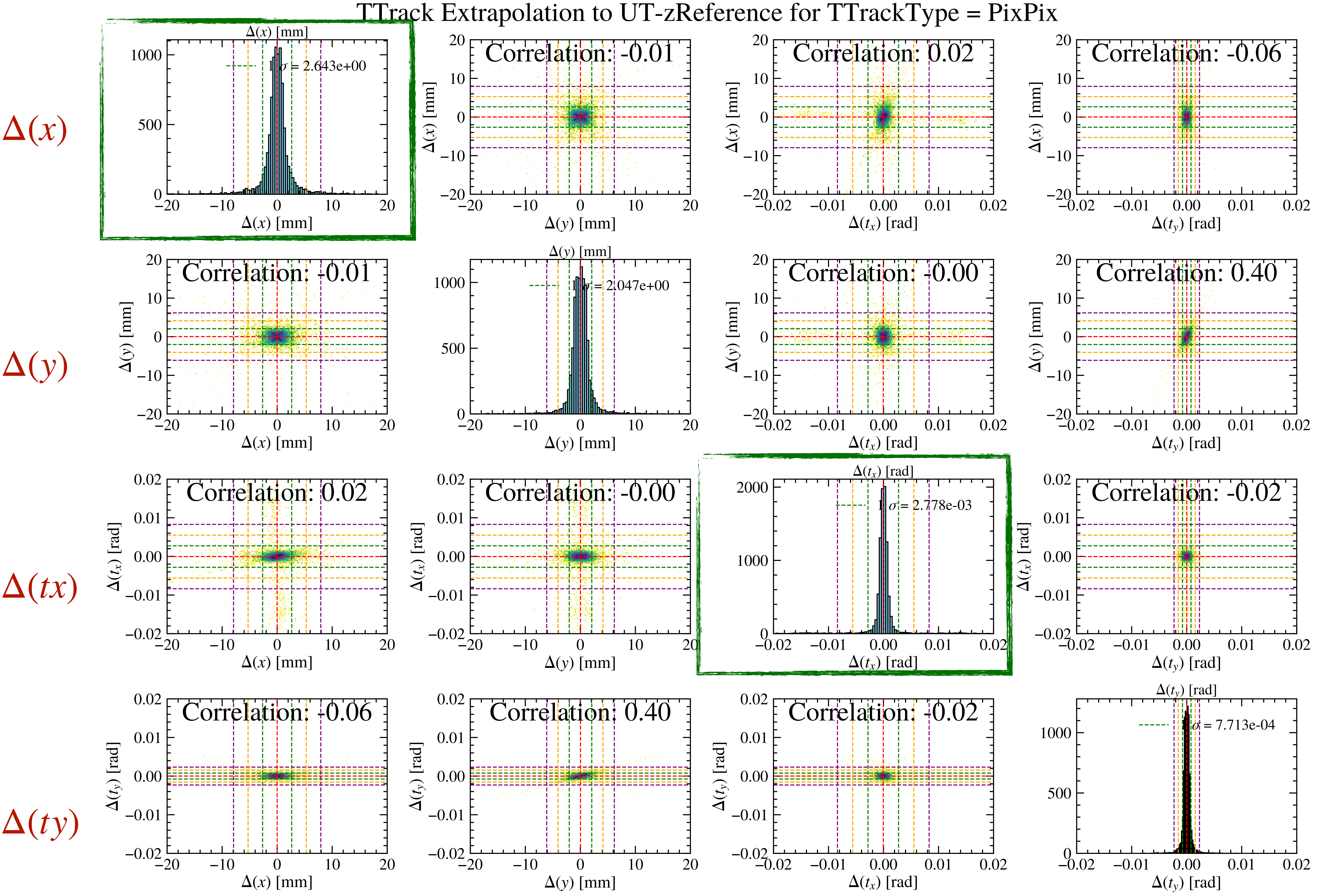
Δ(ty)



DELTAS of Cheated UPStates and States prediction from Velo to expect using Manuel parameterization

(NB: predictions Depends on the q/p uncertainty due to the TTrack quality state and momentum of those acceptances)

Predictions to UT from Velo side given matched TTrack-Velo



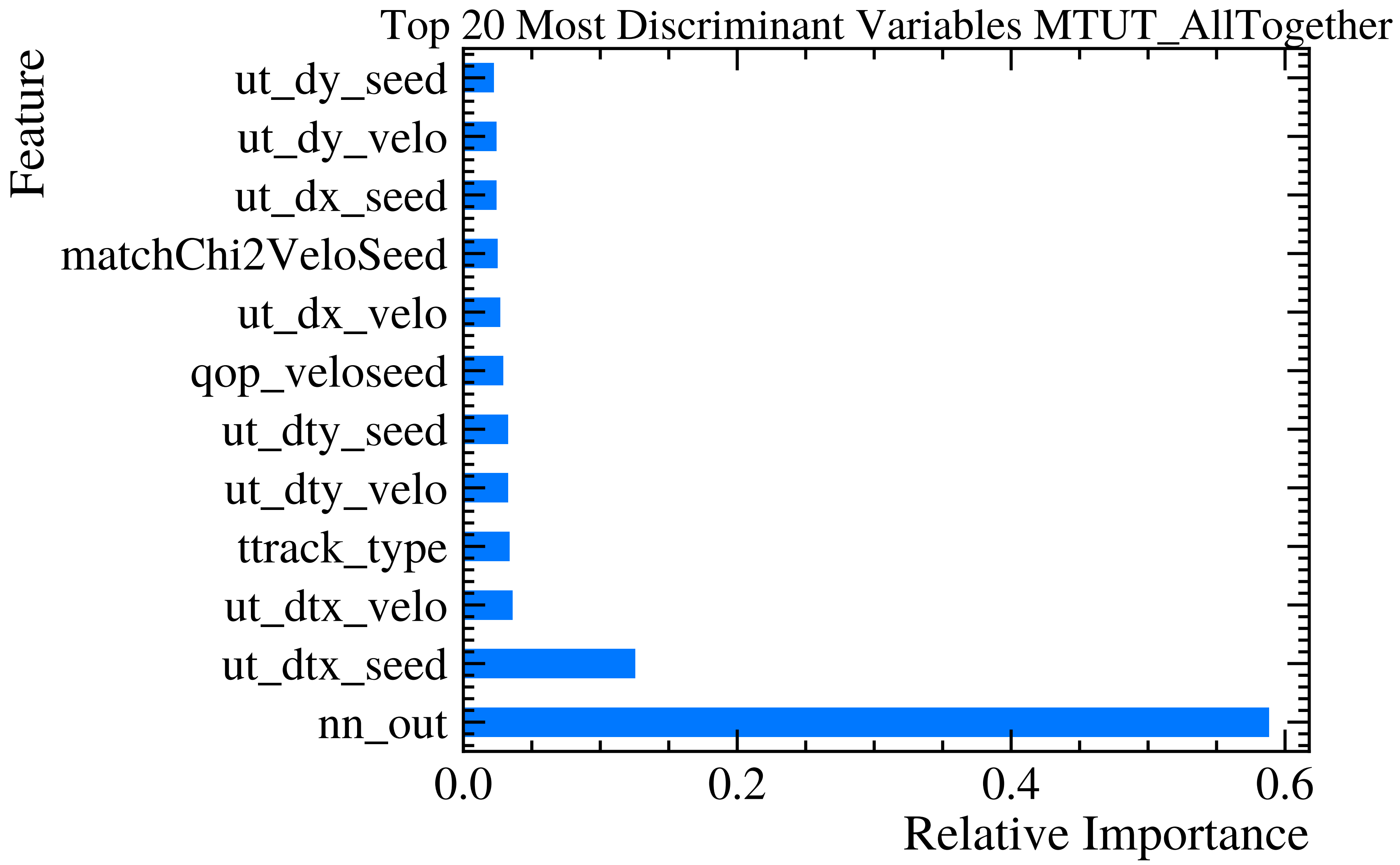
DELTAS of Cheated UPStates and States prediction from TTracks to expect using Manuel parameterization

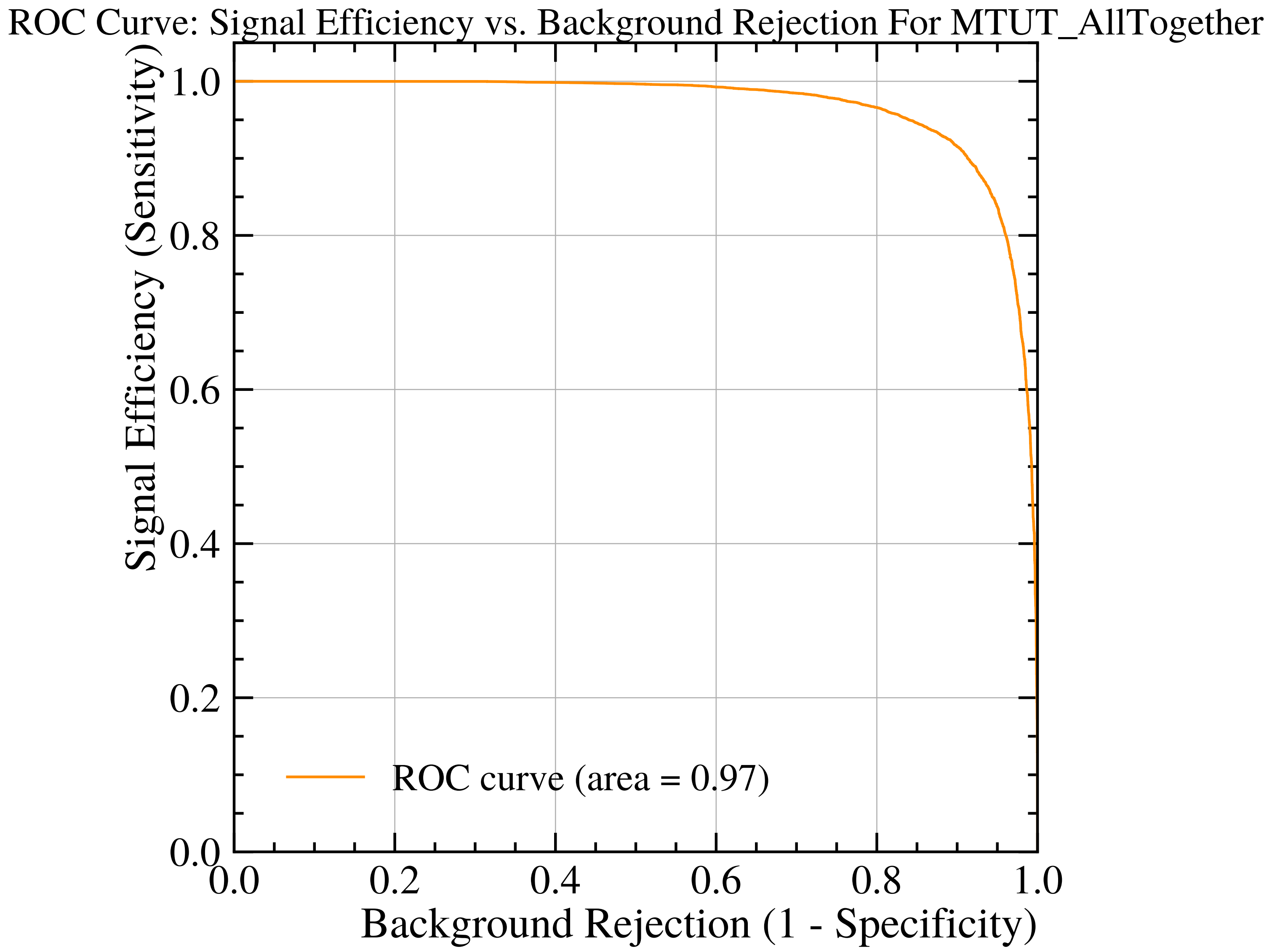
UT state can be constrained from both sides.

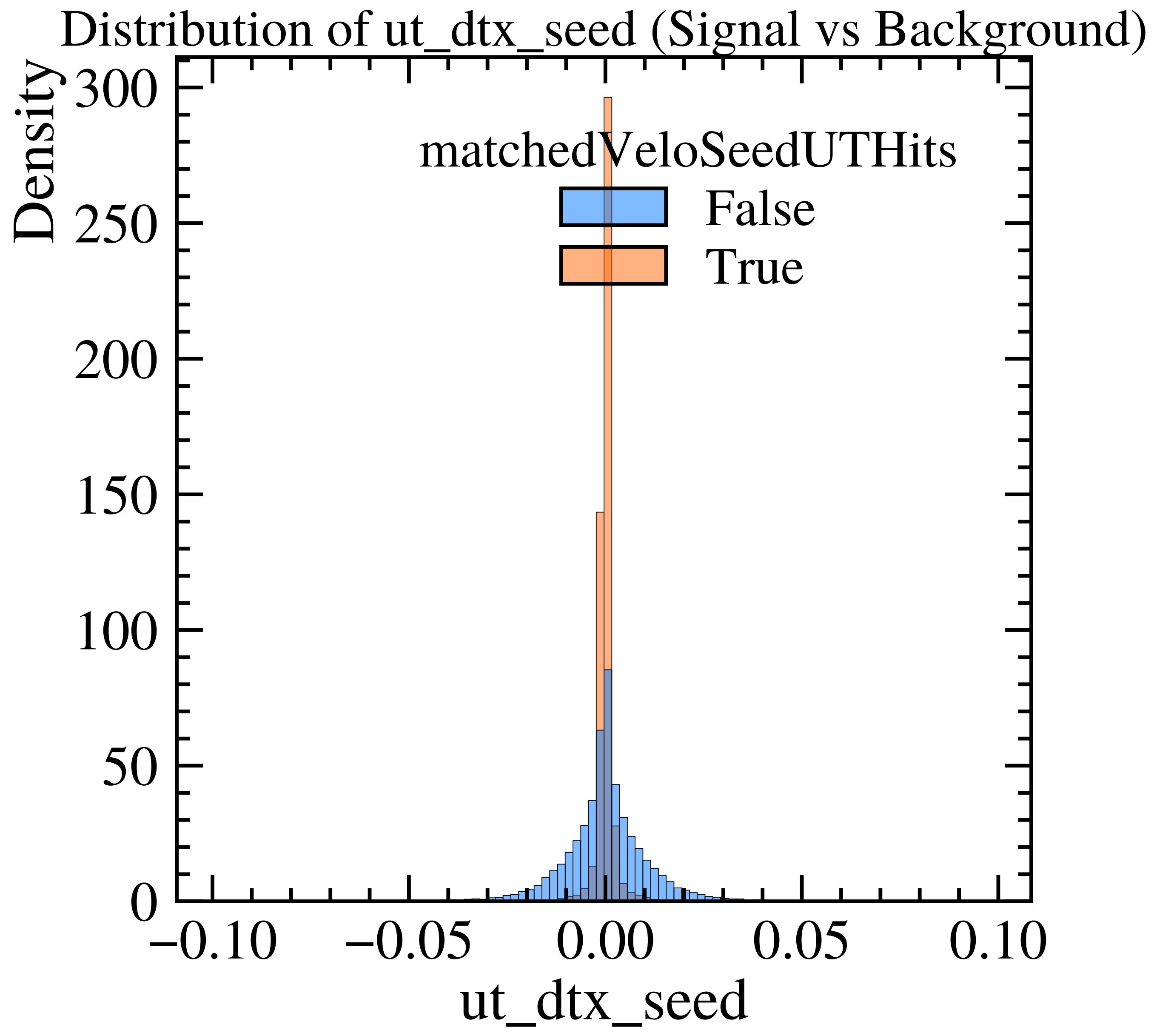
Is this too much?

Additional NN trained with deltas of states and previous NN score values

Potentially more information to introduce [from UP state fit quality/velo etc..] and improve predictions from Velo / FT







Potentially more information to introduce and improve predictions from Velo / FT

Different selection of final candidates if NN from UP available

- For each **v** in Velo:
  - For each **t** in TTrack:
    - [**v**,[**t**.....]] χ^2 match, pick the first best 5 (tunable) to train a NN
    - Assing NNscore, keep best 5 NNscores
    - **Look up for 1 UP 'line-parabola' to have UP state, reject pair if UP found, add NN with UP (score2)**
    - Construct inverse relation [t, [(v1,**score2**),....] sorted from highest to lower
- Make matches for each single TTrack with best score velo-up-ttrack segment
  - Keep multiple solutions if the second best score is good as well [tunable]

Current results [with UT addition]

- For L = 1.5E34 , FTDR MightyTracker geometry

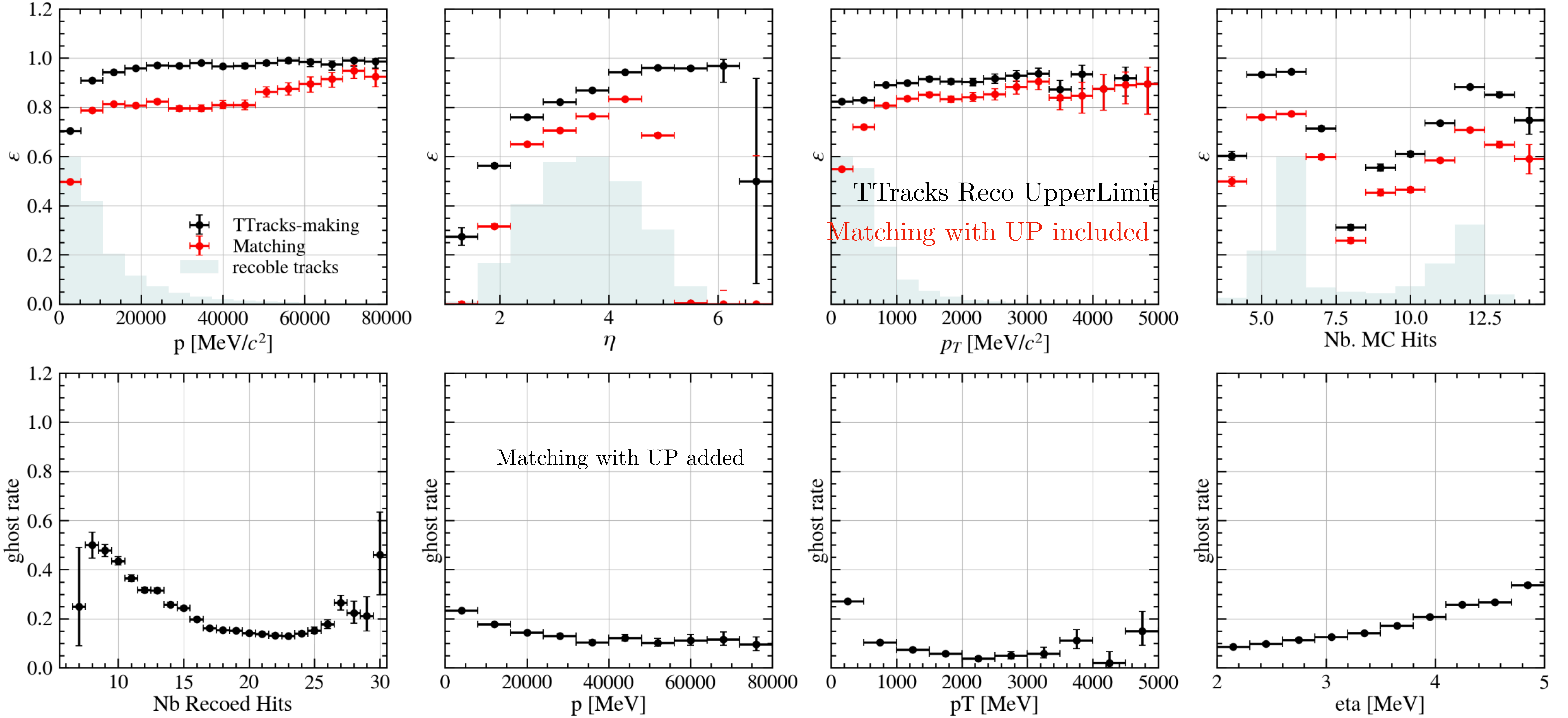
LongTracks Tracks PrChecker

Category	NO UP: # Recoed / # Recoble	NO UP: Efficiency (%)	NO UP: Clones (%)	ADD UP: # Recoed / # Recoble	ADD UP: Efficiency (%)	ADD UP: Clones (%)
Long (all)	39358 / 54008	72.87 %	257 (0.65 %) [nTotMatch = 39411]	35637 / 54008	65.98 %	220 (0.62 %) [nTotMatch = 35684]
Long eta[2,5]	37034 / 49379	75.00 %	257 (0.69 %) [nTotMatch = 37087]	35106 / 49379	71.09 %	220 (0.63 %) [nTotMatch = 35153]
Long eta[2,5], p>3GeV	34215 / 41517	82.41 %	230 (0.67 %) [nTotMatch = 34262]	32561 / 41517	78.43 %	190 (0.58 %) [nTotMatch = 32602]
Long eta[2,5], p>5GeV	26887 / 30935	86.91 %	116 (0.43 %) [nTotMatch = 26914]	26134 / 30935	84.48 %	116 (0.44 %) [nTotMatch = 26161]
Long_decay (all)	36492 / 47985	76.05 %	186 (0.51 %) [nTotMatch = 36532]	32931 / 47985	68.63 %	160 (0.49 %) [nTotMatch = 32967]
Long_decay eta[2,5]	34262 / 43749	78.31 %	186 (0.54 %) [nTotMatch = 34302]	32420 / 43749	74.10 %	160 (0.49 %) [nTotMatch = 32456]
Long_decay eta[2,5], p>3GeV	31691 / 37187	85.22 %	170 (0.54 %) [nTotMatch = 31728]	30132 / 37187	81.03 %	138 (0.46 %) [nTotMatch = 30164]
Long_decay eta[2,5], p>5GeV	25053 / 28139	89.03 %	84 (0.33 %) [nTotMatch = 25075]	24310 / 28139	86.39 %	84 (0.35 %) [nTotMatch = 24332]
(!e) Long_decay (all)	36351 / 47720	76.18 %	186 (0.51 %) [nTotMatch = 36391]	32816 / 47720	68.77 %	160 (0.49 %) [nTotMatch = 32852]
(!e) Long_decay eta[2,5]	34131 / 43533	78.40 %	186 (0.54 %) [nTotMatch = 34171]	32305 / 43533	74.21 %	160 (0.49 %) [nTotMatch = 32341]
(!e) Long_decay eta[2,5], p>3GeV	31691 / 37187	85.22 %	170 (0.54 %) [nTotMatch = 31728]	30132 / 37187	81.03 %	138 (0.46 %) [nTotMatch = 30164]
(!e) Long_decay eta[2,5], p>5GeV	24965 / 28016	89.11 %	84 (0.34 %) [nTotMatch = 24987]	24231 / 28016	86.49 %	84 (0.35 %) [nTotMatch = 24253]
(!e) LongUT_decay (all)	34597 / 44958	76.95 %	186 (0.54 %) [nTotMatch = 34637]	32787 / 44958	72.93 %	160 (0.49 %) [nTotMatch = 32823]
(!e) LongUT_decay eta[2,5]	34097 / 43431	78.51 %	186 (0.54 %) [nTotMatch = 34137]	32301 / 43431	74.37 %	160 (0.49 %) [nTotMatch = 32337]
(!e) LongUT_decay eta[2,5], p>3GeV	31659 / 37103	85.33 %	170 (0.54 %) [nTotMatch = 31696]	30128 / 37103	81.20 %	138 (0.46 %) [nTotMatch = 30160]
(!e) LongUT_decay eta[2,5], p>5GeV	24937 / 27957	89.20 %	84 (0.34 %) [nTotMatch = 24959]	24227 / 27957	86.66 %	84 (0.35 %) [nTotMatch = 24249]
(!e) LongUT_decay (Pix-Pix)	21363 / 24372	87.65 %	8 (0.04 %) [nTotMatch = 21364]	20403 / 24372	83.71 %	8 (0.04 %) [nTotMatch = 20404]
(!e) LongUT_decay (Pix-Pix) eta[2,5]	21110 / 23942	88.17 %	8 (0.04 %) [nTotMatch = 21111]	20186 / 23942	84.31 %	8 (0.04 %) [nTotMatch = 20187]
(!e) LongUT_decay (Pix-Pix) eta[2,5], p>3GeV	20953 / 23115	90.65 %	8 (0.04 %) [nTotMatch = 20954]	20048 / 23115	86.73 %	8 (0.04 %) [nTotMatch = 20049]
(!e) LongUT_decay (Pix-Pix) eta[2,5], p>5GeV	18785 / 20219	92.91 %	0 (0.00 %) [nTotMatch = 18785]	18241 / 20219	90.22 %	0 (0.00 %) [nTotMatch = 18241]
(!e) LongUT_decay (Fib-Fib)	11372 / 16575	68.61 %	40 (0.35 %) [nTotMatch = 11381]	10652 / 16575	64.27 %	38 (0.36 %) [nTotMatch = 10661]
(!e) LongUT_decay (Fib-Fib) eta[2,5]	11127 / 15533	71.63 %	40 (0.36 %) [nTotMatch = 11136]	10386 / 15533	66.86 %	38 (0.37 %) [nTotMatch = 10395]
(!e) LongUT_decay (Fib-Fib) eta[2,5], p>3GeV	9038 / 11093	81.47 %	34 (0.38 %) [nTotMatch = 9046]	8525 / 11093	76.85 %	28 (0.33 %) [nTotMatch = 8532]
(!e) LongUT_decay (Fib-Fib) eta[2,5], p>5GeV	5038 / 5997	84.01 %	22 (0.44 %) [nTotMatch = 5044]	4910 / 5997	81.87 %	22 (0.45 %) [nTotMatch = 4916]
(!e) LongUT_decay (Pix-Fib)	1862 / 4011	46.42 %	138 (7.29 %) [nTotMatch = 1892]	1732 / 4011	43.18 %	114 (6.48 %) [nTotMatch = 1758]
(!e) LongUT_decay (Pix-Fib) eta[2,5]	1860 / 3956	47.02 %	138 (7.30 %) [nTotMatch = 1890]	1729 / 3956	43.71 %	114 (6.50 %) [nTotMatch = 1755]
(!e) LongUT_decay (Pix-Fib) eta[2,5], p>3GeV	1668 / 2895	57.62 %	128 (7.55 %) [nTotMatch = 1696]	1555 / 2895	53.71 %	102 (6.46 %) [nTotMatch = 1579]
(!e) LongUT_decay (Pix-Fib) eta[2,5], p>5GeV	1114 / 1741	63.99 %	62 (5.49 %) [nTotMatch = 1130]	1076 / 1741	61.80 %	62 (5.68 %) [nTotMatch = 1092]

	# Events	# Recoed Tracks	# Fake Tracks	Ghost Rate
NO UP	100	66384	26901	40.52 %
ADD UP		44409	8662	Ghost Rate (evt-avg) 19.51 % (18.98)

Results on $B \to J/\psi\phi$ at L = 1.5e34

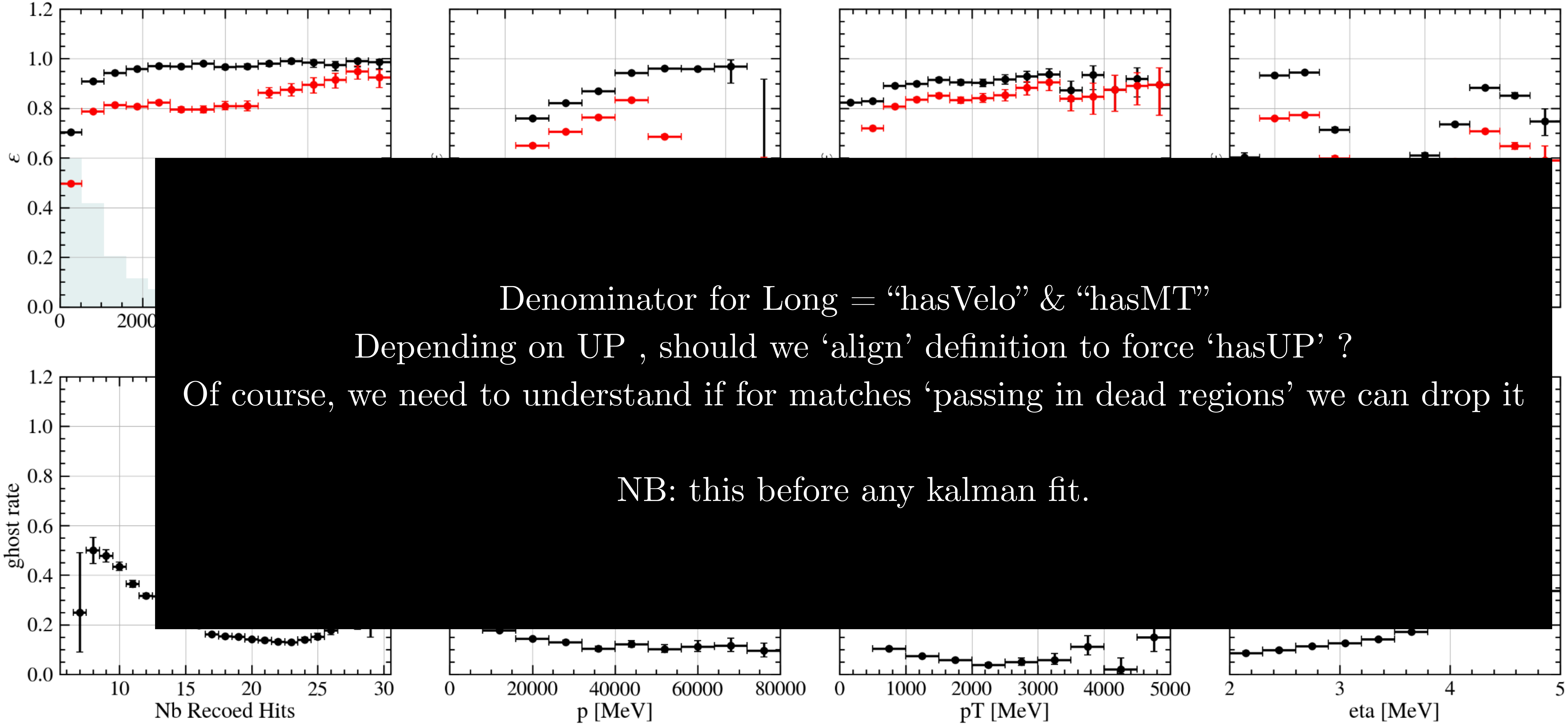
(Upper threshold from TTracks - Finding)



(Assigned p) (Assigned pT) (Assigned eta)

Results on $B \rightarrow J/\psi\phi$ at L = 1.5e34

(Upper threshold from TTracks - Finding)



(Assigned p) (Assigned pT) (Assigned eta)

Denominator for Long = "hasVelo" & "hasMT"

Depending on UP , should we 'align' definition to force 'hasUP' ?

Of course, we need to understand if for matches 'passing in dead regions' we can drop it

NB: this before any kalman fit.

Conclusions (1)

- **A full long track making using the full acceptance of the MT prototyped**
- Further optimization needed for the UP hits finding given Velo-Track window and for FT-fibre only low-p tracks
- Revisit UP states creation (aka velo-UP with precise prior p-estimation)

- Some toughs on hit-flagging in FT to do and ordering of sequence when we deal with pix/fib, some tracks are truth and easy, others are harder. “Dedicated TTrack for secondaries from beampipe to reduce occupancy?”

- The matching with 3 states has to be further explored and improved.
- Also Velo-EndState will get better due to less material (?)
- I think the outcome from the checks are good enough for now.
- Those tracks made are not yet ‘Kalman-Fitted’. p assigned based on kink

Conclusions (2)

- Within the pyReco , plan to make few fixes in PixFib tracking and some smarter UP hits selection to make candidates. Drop useless training variables in NN.

- Test over all Christoph/Lennart samples scenarios [lumis] and interface 3/4 UP layers and geometry from Benjamin/Carlos. Test the scoped scenarios we will agree on.

- Porting of algorithms/logics where updated in LHCb software has started, but the code can be easily adapted to run over U2 .sim files.

- Compiling a set of scenarios to quote efficiencies and fake rates for the matching and long-track making.

Backup

Reconstructible criteria in pyReco

- Tunables:
- With baseline 4 UT layers >=3 MCHits
  - [but has to be updated to be >=3 stations or >=2]
- For MT : rather than counting hits, count 'information'
  - nX = 1unit for pixel, 1 unit for x-layers
  - nY = 1 unit for pixel, 1 unit for u/v-layers
  - Unified definition for reconstructible in MightyTracker

MightyTracker Masks and tunings

Can study fibre length / overlap region in flexible way