FastDIRC Status Report

Yunjie Yang

DIRC workfest
April 21, 2020

Outline

- A proper (re-)introduction to FastDIRC
- Current status: reconstruction and observable studies
- Moving forward

Outline

- **A proper (re-)introduction to FastDIRC**
- Current status: reconstruction and observable studies
- Moving forward

The many meanings of “FastDIRC”

Depending on the context, the term “FastDIRC” can refer to:

- A reconstruction philosophy/approach
- A way of doing fast photon generation & propagation
- A specific implementation, i.e. a code repo on GitHub, of this reconstruction approach

Some charged track hits a DIRC bar

track's extrapolated $\overrightarrow{p}, \overrightarrow{x}, t$

↓

A "black box" that we try to model as best as we can: geometry, photon generation and propagation

↓

O(50) observed PMT hits

each hit: PMT channel and time

FastDIRC reconstruction philosophy

Some charged track hits a DIRC bar

track's extrapolated $\vec{p}, \vec{x}, t$

A "black box" that we try to model as best as we can: geometry, photon generation and propagation

O(50) observed PMT hits

each hit: PMT channel and time

What if…

O(100k or 1M) hits that could have been generated by this track, according to our model of the "black box"

Let's call them "**support points**"

FastDIRC reconstruction philosophy

track's extrapolated $\overrightarrow{p}, \overrightarrow{x}, t$

O(1M) support points under **pion** hypothesis

O(1M) support points under **kaon** hypothesis

Given the O(50) observed hits and the O(1M) support points under each hypothesis,
how likely do the observed hits come from each hypothesis?

FastDIRC reconstruction philosophy: a 3D-PDF approach

Answer: Hypothesis testing.
Compute the likelihood under each hypothesis, and form delta log-likelihoods

How to compute the likelihood under each hypothesis?

Pseudocode*:

likelihood $= 1$
for every observed hit O_i **do**
  for every support point S_j **do**

$$r_{i,j}^2 = \frac{(x_{O_i} - x_{S_j})^2}{s_x^2} + \frac{(y_{O_i} - y_{S_j})^2}{s_y^2} + \frac{(t_{O_i} - t_{S_j})^2}{s_t^2}$$

$$\text{likelihood} \mathrel{*}= \exp\left(-\frac{r_{i,j}^2}{s_b^2}\right)$$

  end for
end for

*In the real code, cut-offs, logs and additions are used in intermediate steps to make computation tractable.

FastDIRC reconstruction philosophy: a 3D-PDF approach

$$r_{i,j}^2 = \frac{(x_{O_i} - x_{S_j})^2}{s_x^2} + \frac{(y_{O_i} - y_{S_j})^2}{s_y^2} + \frac{(t_{O_i} - t_{S_j})^2}{s_t^2}$$

$$\text{likelihood } *= \exp\left(-\frac{r_{i,j}^2}{s_b^2}\right)$$

- s_x, s_y, s_t, s_b are simply free parameters that represent some sort of scaling and they do *not* correspond to resolutions on those quantities
- This is simply *one* way of computing likelihoods from the given support points and observed hits, i.e. different kernels can be used
- At the present stage, it's likely more productive to focus on the question of *whether the support points represent the 3 dimensional probability density function from which the observed points are drawn*

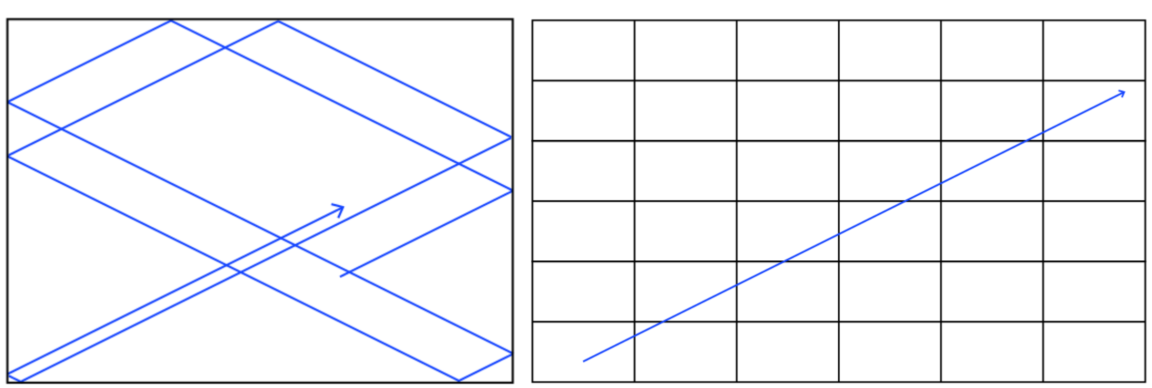
How do you obtain the O(1M) support points per charged track?

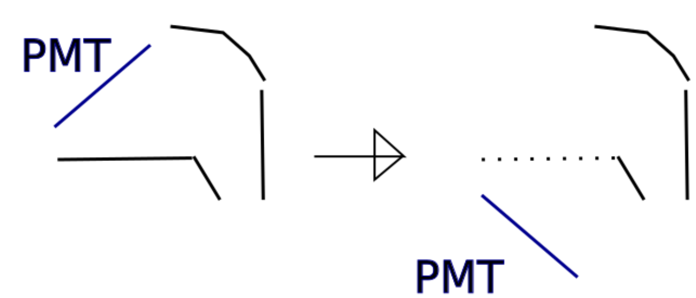
- **Monte Carlo approach**, i.e. Geant
  - pros: reasonably model all the effects that you put in
  - cons: slow, cannot do this for every track — naive implementation would render this reconstruction not viable
- **Analytical approach**
  - pros: fast enough to make this reconstruction approach feasible
  - cons: cannot model everything — could miss subtle effects

FastDIRC simulation: a fast analytical ray-tracing modeling

- Generation: only generate those that will be detected
  - Sample a wavelength spectrum: intrinsic $1/\lambda^2$, materials' λ dependence, PMT Q.E. etc.
  - Determine the index of refraction: $n(\lambda)$
  - Throw the photons around the Cherenkov cone

- Propagation: purely geometrical
  - A reflection: e.g. dirVec_x = -dirVec_x
  - Reflections are perfect
  - Gaussian smearing can be added





FastDIRC simulation: a fast analytical ray-tracing modeling

- The modeling of generation+propagation is certainly not comprehensive
- It can serve as an independent way of checking data, e.g. photon yield, timing
- It *enables* the 3D-PDF reconstruction approach, but in principle it is also *independent* of the reconstruction approach used

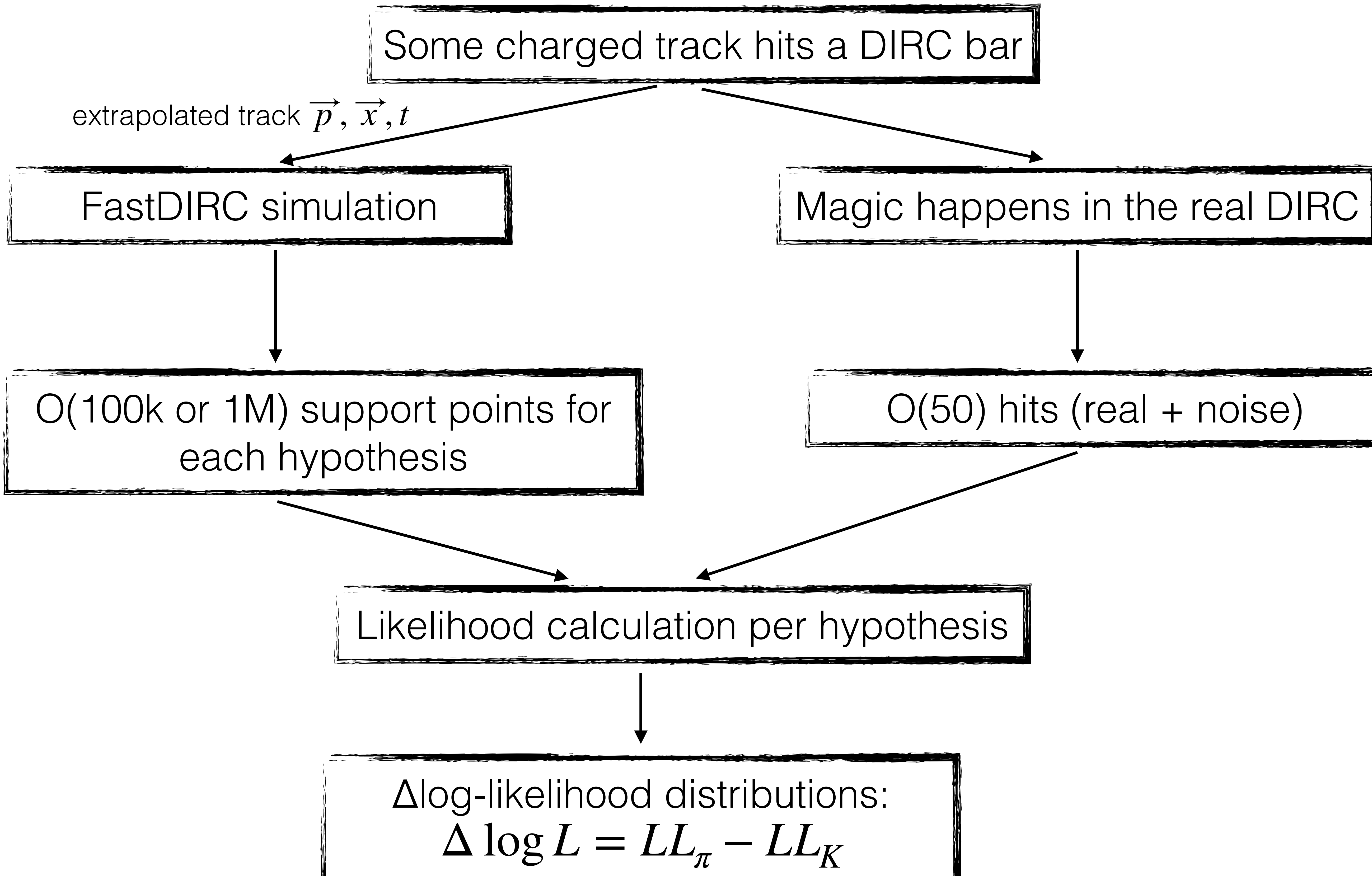
Key questions I am focusing on:

- The ability to quickly simulate a large number of potential PMT hits should be a powerful tool at our disposal. How can we make the best use of it?
- The correct modeling of the p.d.f that the observed hits are drawn from is critical. How can we ensure that?

Outline

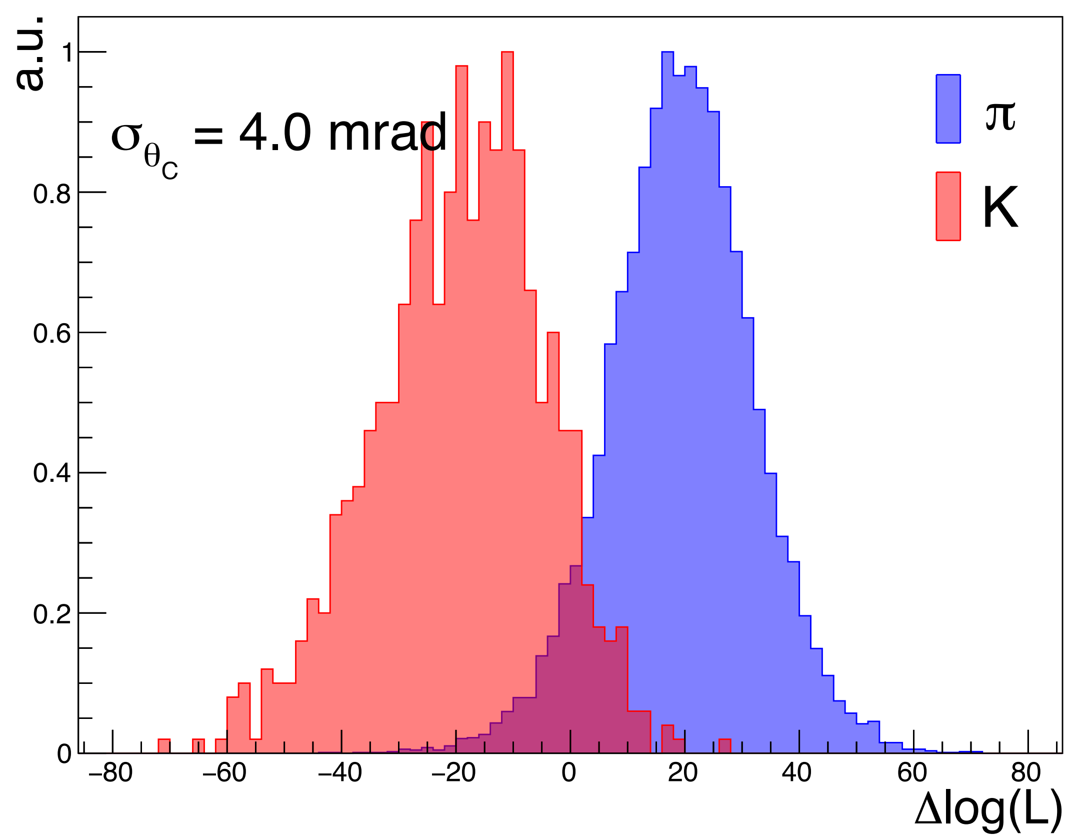
- A proper (re-)introduction to FastDIRC
- **Current status: reconstruction and observable studies**
- Moving forward

FastDIRC reconstruction



To convert from a ΔLL distribution into an equivalent angular resolution:

- From ΔLL, construct the ROC curve and compute the AUC
- In θ_C space, we know the theoretical $\Delta\theta_C$ for this momentum (~11.5 mrad in this case)
- Assume both the pion and kaon θ_C are Gaussian-distributed and the single track resolution σ_{θ_c} is the same for both, numerically compute what σ_{θ_c} value would produce the same AUC



From a sample of 3 GeV π+K tracks

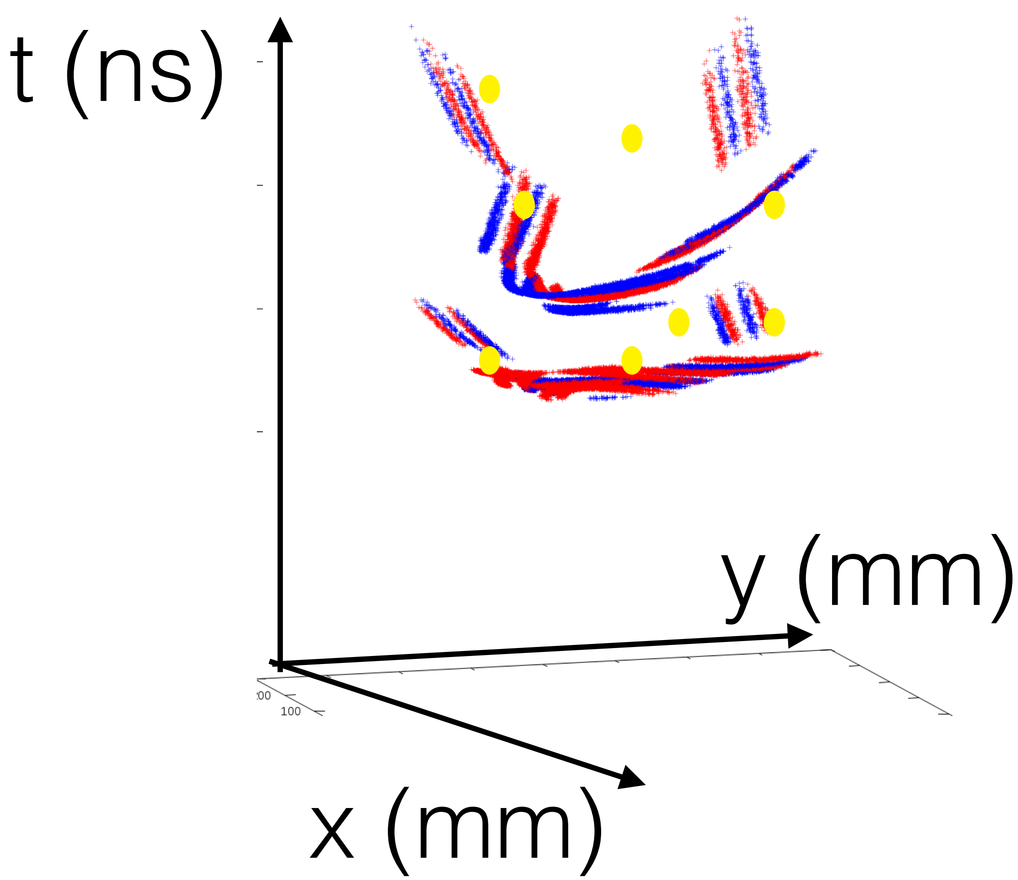
FastDIRC current status

- Adapted the original FastDIRC code with the as-built nominal geometry
- Functional as an alternative reconstruction method
- Achieved π/K separation (~3σ at 3 GeV) comparable with the without-any-correction geometrical reconstruction
- Experimented with some ideas:
  - Attempted at alignment with geometry parameters: looked at different figures-of-merit, tried a few naive overall shifts
  - Construction of low-level observables: as an independent check of data quality (during commissioning and beyond), and to identify where the inner workings of the modeling need to be improved

Input:
track kinematics,
mass hypothesis

Model:
geometry + propagation

O(1M) support points
in (x, y, t) space



red and blue bands:
support points from
two mass hypos

observed hit

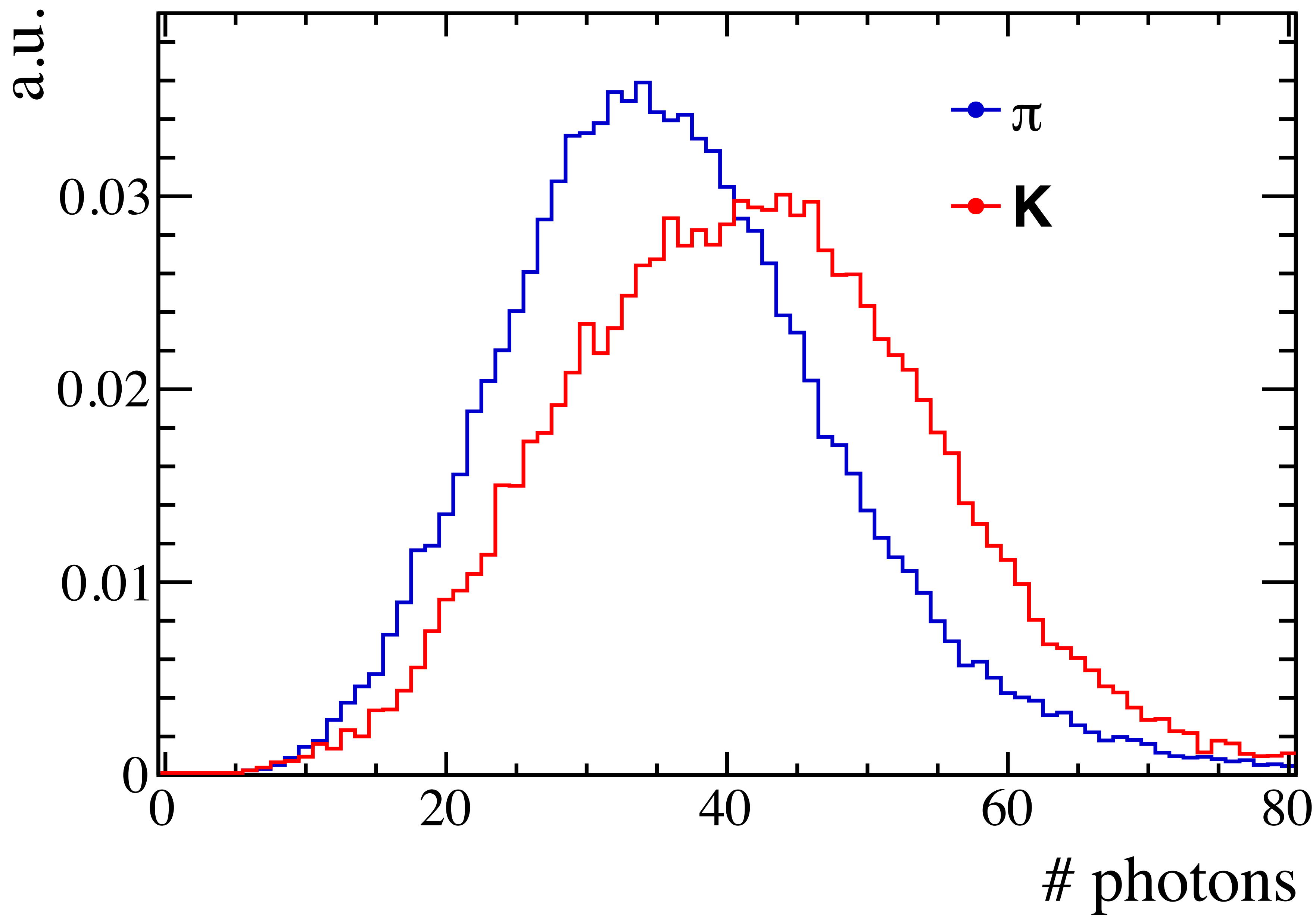
Idea: for each observed hit, look at its “neighbors” in this 3D space, and try to construct observables

For each hit, define its *neighborhood* as, e.g. within ± 5 ns and ± 8.5 mm:

- Yes/No counting <=> photon yield:
  - If there is at least one (no) support points in the neighborhood, call it a signal (noise) hit
- Distance measure <=> Delta observables:
  - For all support points in the neighborhood, construct $\Delta x = x_{\mathrm{O}} - x_{S,i}$, $\Delta y = y_{\mathrm{O}} - y_{S,i}$, $\Delta t = t_{\mathrm{O}} - t_{S,i}$

=> Showed interesting/promising signs, but a lot to understand

Example: Photon Yield

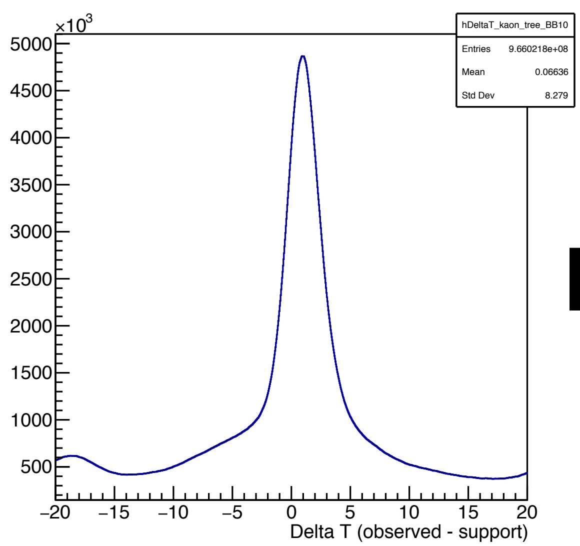


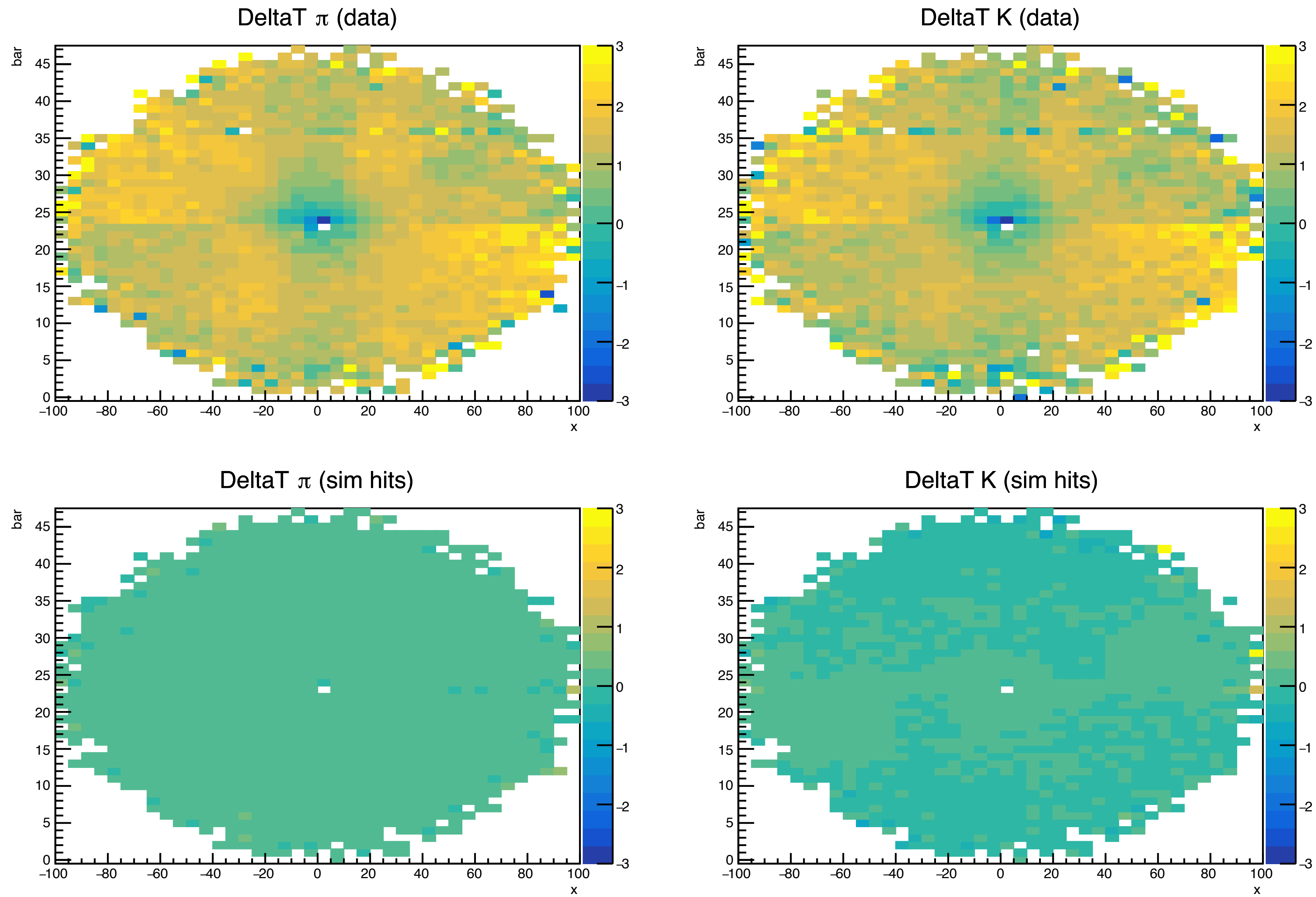
From a subset of 2019-11 dataset, integrated over 4+ GeV tracks and all locations on the DIRC

Similar behavior, but with an independent approach

Example: Δt

Per bar per x-bin





From a subset of 2019-11 dataset, 3 GeV tracks selected

Need to understand: what's the cause of overall shift?
path length dependence?

This is just the beginning of looking at real data with this reconstruction approach. There is **A LOT** to understand:

- What does it mean to apply “corrections” in this context? (maybe this is not the right question to ask)
- How to factorize the various effects at play? What observables can we construct to do that?
- Fast simulation of support points should be a powerful tool, but how can we best use it?
- …

Many aspects of the “code commissioning” still needed:

- Geometry manipulation works as intended
- Cherenkov wavelength spectrum represents our current knowledge

Outline

- A proper (re-)introduction to FastDIRC
- Current status: reconstruction and observable studies
- **Moving forward**

Future Plan

I need to graduate at some point… but before that, I hope to

- integrate FastDIRC into GlueX software
- continue exploring different ideas

Summary

- FastDIRC is still useful and relevant in the era of data

- Its potential is not well unexplored

- It is just the beginning and I hope the story continues