Time-of-Flight Estimation using Machine Learning Techniques

Konrad Helms

ML4Jets, November 6, 2023

HELMHOLTZ

DESY.

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN IN PUBLICA COMMODA SEIT 1737

Overview

Time-of-Flight Measurements at the International Large Detector

Data & Preprocessing

Benchmark Time-of-Flight Estimator

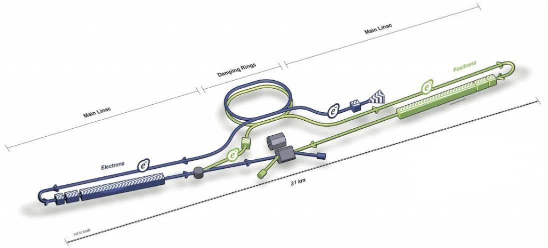
Machine Learning-Based Time-of-Flight Regression

Summary & Outlook

Time-of-Flight Measurements at the International Large Detector

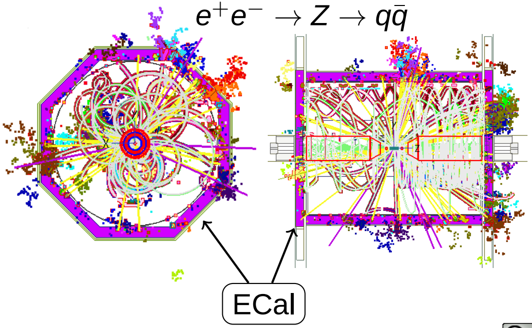
The Setup: ILC and ILD

- International Linear Collider (ILC)
- proposed, but not build yet
- colliding polarised e^+e^- @ 250 GeV centre-of-mass energy
- designed for precision electroweak and Higgs measurements



- International Large Detector (ILD)
- highly granular electromagnetic and hadronic calorimeters (ECal & HCal)
- optimised for Particle Flow calorimetry

$e^+e^- \rightarrow Z \rightarrow q\bar{q}$

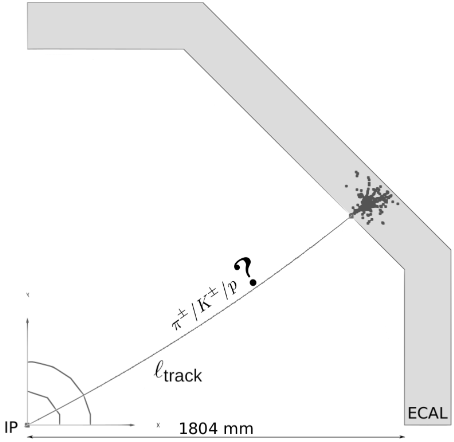


TOF at ILD

- time-of-flight (TOF) estimation using ML
- used in particle identification of particles with $p \lesssim \mathcal{O}(10\,\text{GeV})$
- particle identification reduces to $\pi^{\pm}$ vs. $K^{\pm}$ vs. $\overset{(-)}{p}$
- particle identified by mass and electric charge

$$\Rightarrow m_0 = \frac{p \cdot \text{TOF}}{\ell_\text{track}} \sqrt{1 - \left(\frac{\ell_\text{track}}{(\text{TOF} \cdot c)}\right)^2}$$

- need to know p (from curvature in magnetic field), ℓ_track (from tracker), TOF (now from ML)

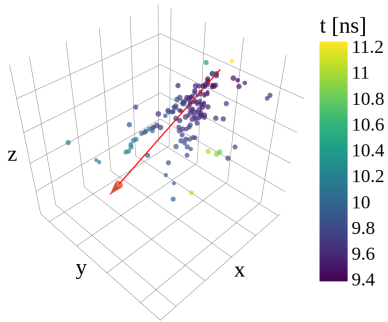


Data & Preprocessing

Data & Rotations

- data preprocessing essential, since we follow a data-centric ML approach
- MC generated unordered 5D point cloud $\oplus$ track information:

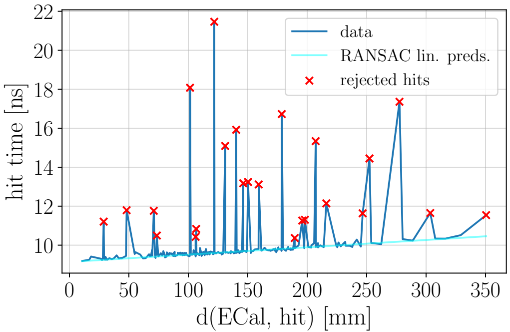
$$\Big\{ \underbrace{[x, y, z, e, t]_i}_{\substack{\text{coordinates,}\\ \text{per hit}}} \oplus \underbrace{[\ell_{\text{track}}, p, p_{\text{T}}, p_x, p_y, p_z, \text{ECal}_x, \text{ECal}_y, \text{ECal}_z]}_{\substack{\text{reconstructed track information,}\\ \text{per shower}}} \;\Big|\; i = 1 \ldots n_{\text{hits}} \Big\} \times n_{\text{showers}}$$

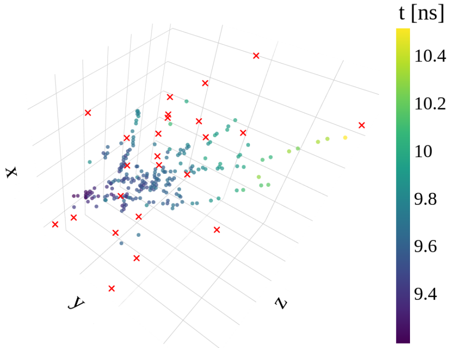


- assumed hit time t_i resolution ± 50 ps in all ECal layers
- define distances using 3D euclidean metric + two extra channels: $(x, y, z)^{\text{T}} \oplus (e, t)$
- order hits by distance to ECal surface
- transform barrel and endcap showers differently: rotations, scaling, ...

RANSAC

- outliers in space and time through backscattering, delayed nuclear reactions, ...
- random sample consensus (RANSAC) algorithm used for outlier rejection
- estimate parameters of an affine function: t_i vs d(ECal, hit$_i$)
- if $(t_i -$ affine pred.$_i) \leq 0.5$ ns: keep hit, else: reject $\rightarrow$ cuts away $\simeq 10\%$ of hits





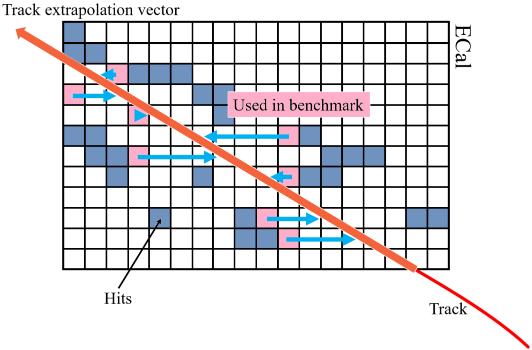
Benchmark Time-of-Flight Estimator

Benchmark TOF Estimator

- select hits closest to momentum extrapolation into ECal in the first 10 layers
- correct measured hit time by travel time to detection position, assuming velocity c

$$t_{\text{corrected},i} = t_i - \frac{\text{d}(\text{ECal}, (x, y, z)_i)}{c}$$

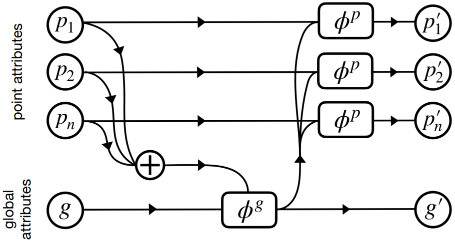
$$\text{TOF} = \frac{1}{n_{\text{hits}}} \sum_{i=1}^{n_{\text{hits}}} t_{\text{corrected},i}$$

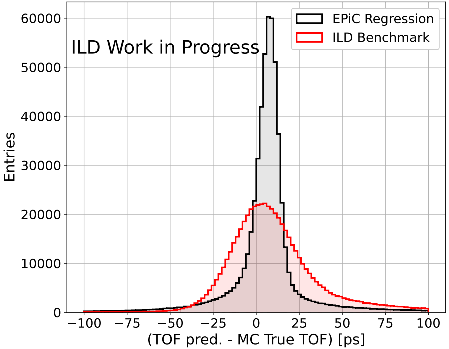


Machine Learning-Based Time-of-Flight Regression

The First Shot: EPiC Regression

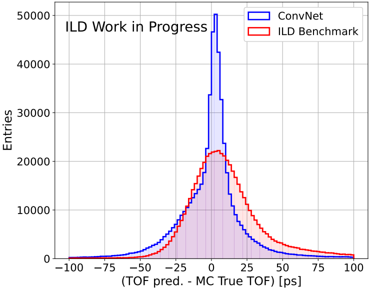
- 3 equivariant point cloud (EPiC) layers from EPiC-GAN, Buhmann, Kasieczka, Thaler for encoding of input
- EPiC layers connect global information with local shower information
- 4 layer MLP for TOF regression based on encoded input
- results - very good! Architecture too overkill?

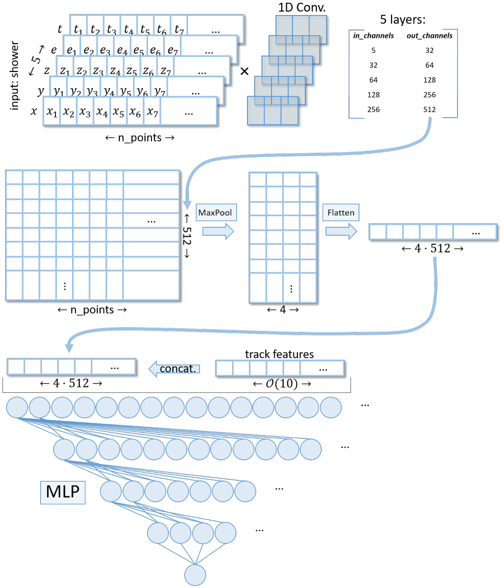




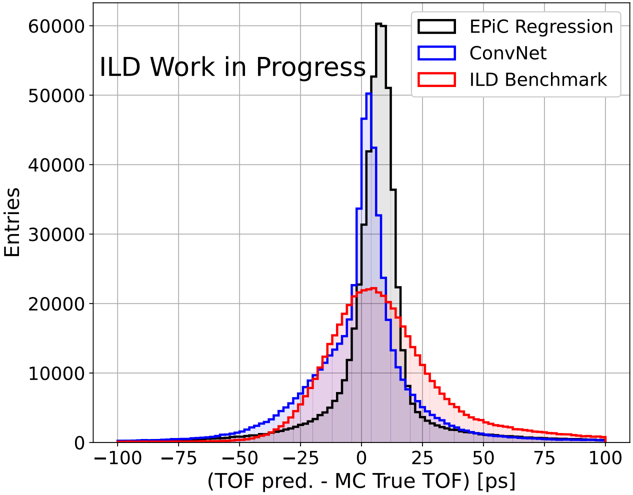
Simpler: ConvNet

- 1D convolutions along x, y, z, e, t separately
- (adaptive) max pooling to deal with varying numbers of hits in the shower + encoding focus on only most important features
- 4 layer MLP for TOF regression





Results



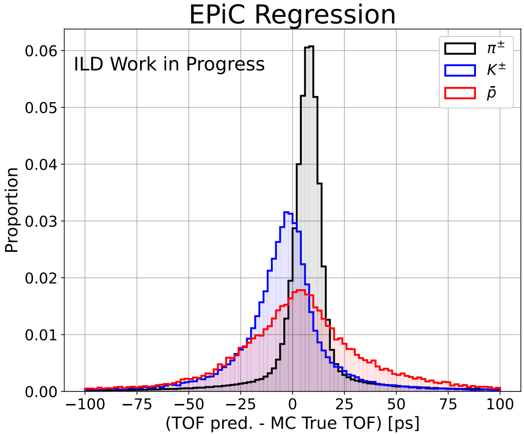
Network	RMS90	RMS90 mean
Benchmark	23.62 ± 0.02	9.73 ± 0.03
EPiC	15.00 ± 0.02	6.17 ± 0.01
ConvNet	18.18 ± 0.02	-0.48 ± 0.02

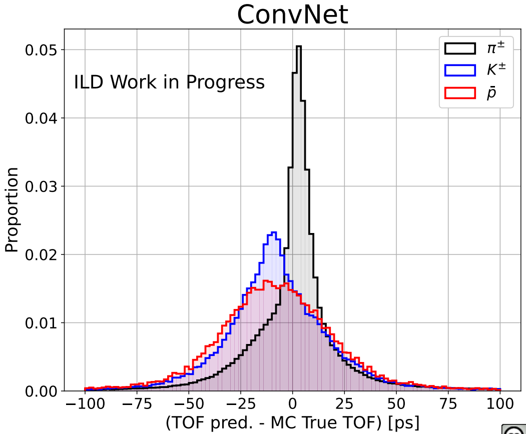
A Closer Look at the Results ... #1

$m_{\pi^\pm} \simeq 140\,\mathrm{MeV}$
$m_{K^\pm} \simeq 490\,\mathrm{MeV}$
$m_{\overset{(-)}{p}} \simeq 940\,\mathrm{MeV}$

> particle abundance in $e^+e^- \to Z \to q\bar{q}$ @ $250\,\mathrm{GeV}$:

$85\%\ \pi^\pm \quad 10\%\ K^\pm \quad 5\%\ \overset{(-)}{p}$





A Closer Look at the Results ... #2

$m_{\pi^\pm} \simeq 140\,\text{MeV}$
$m_{K^\pm} \simeq 490\,\text{MeV}$
$m_{\overset{(-)}{p}} \simeq 940\,\text{MeV}$

- pay attention to the location of the distributions
- network learned that the dataset has two 'mass classes':

  $m_1 \simeq m_{\pi^\pm}$ and $m_2 \simeq m_{\overset{(-)}{p}}$
- $\{\, [x, y, z, e, t]_i \oplus [\ell_{\text{track}}, p, p_{\text{T}}, p_x, p_y, p_z, \text{ECal}_x, \text{ECal}_y, \text{ECal}_z] \mid i = 1 \dots n_{\text{hits}} \,\} \times n_{\text{showers}}$
- it learned the two mass classes $m_{1,2}$ and c, then:

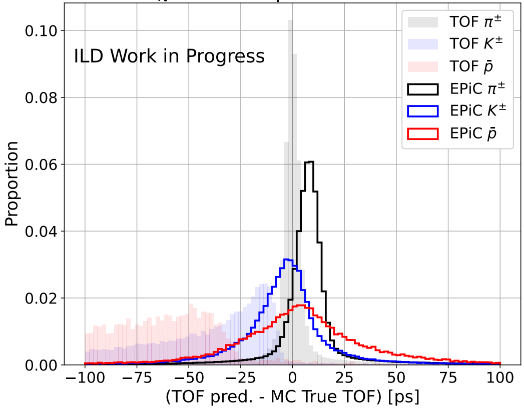
1. estimated the TOF roughly, based on t_i, $i = 1, \dots, n_{\text{hits}}$
2. estimated the corresponding mass class via: $m_{1,2} = \frac{p \cdot \text{TOF}}{\ell_{\text{track}}} \sqrt{1 - \left(\frac{\ell_{\text{track}}}{(\text{TOF} \cdot c)}\right)^2}$
3. corrected the estimated TOF, based on: $\text{TOF} = \sqrt{\left(\frac{m_{1,2}\ell_{\text{track}}}{p}\right)^2 + \left(\frac{\ell_{\text{track}}}{c}\right)^2}$

A Closer Look at the Results ... #3

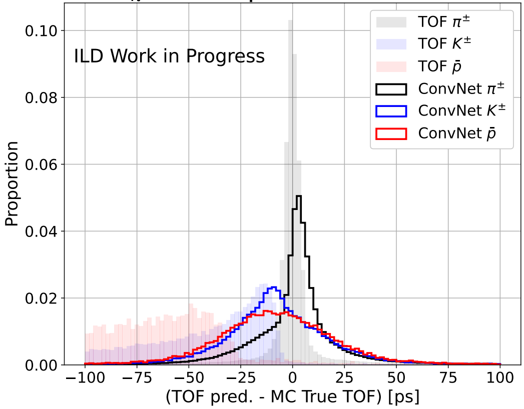
Assuming all Particles are $\pi^\pm$

$m_{\pi^\pm} \simeq 140$ MeV
$m_{K^\pm} \simeq 490$ MeV
$m_{\overset{(-)}{p}} \simeq 940$ MeV

$m_{\pi^\pm}$ assumption vs. EPiC



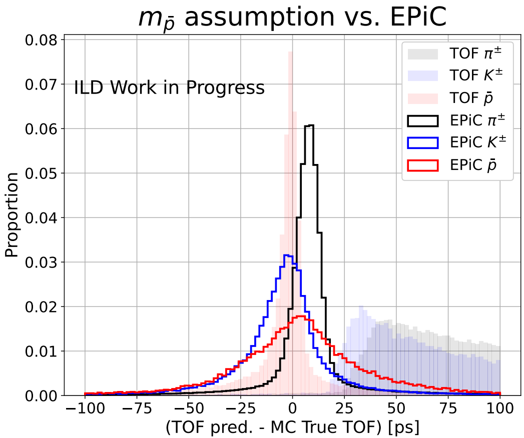
$m_{\pi^\pm}$ assumption vs. ConvNet

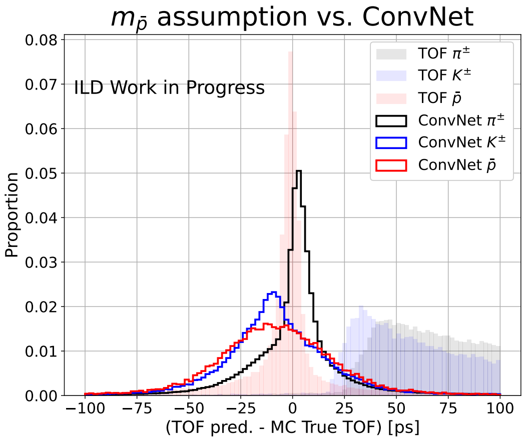


A Closer Look at the Results ... #4

Assuming all Particles are $\overset{(-)}{p}$

$m_{\pi^\pm} \simeq 140\,\text{MeV}$
$m_{K^\pm} \simeq 490\,\text{MeV}$
$m_{\overset{(-)}{p}} \simeq 940\,\text{MeV}$





Summary & Outlook

Summary & Outlook

Summary:

- idea: build a TOF regression network

EPiC Regression, ConvNet

$\Downarrow$

- predict TOF

$\Downarrow$

- use TOF + p + ℓ_{track} for particle identification

Outlook:

- investigate which and how many masses the networks identify
- train on an equal ratio dataset (we're on it)
- remove track length or normalise momentum

$\vdots$

- skip TOF regression, classify $\pi^{\pm}$ vs. $K^{\pm}$ vs. $\overset{(-)}{p}$ directly

Thank you!

Contact

Deutsches Elektronen-
Synchrotron DESY

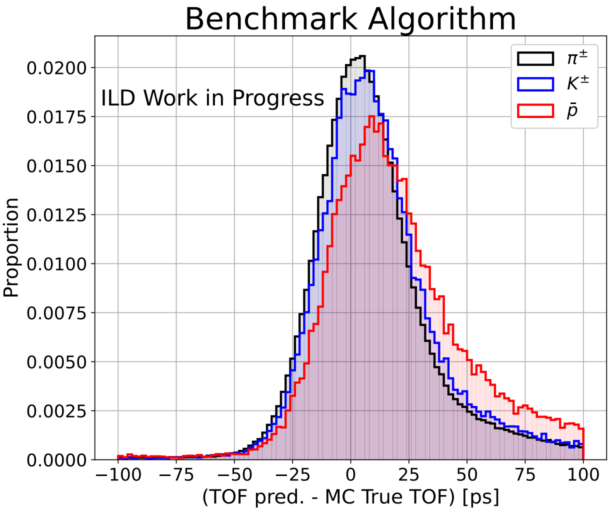
www.desy.de

Konrad Helms
0009-0008-7200-7670
DESY FTX / Georg-August-Universität Göttingen
konrad.helms@desy.de

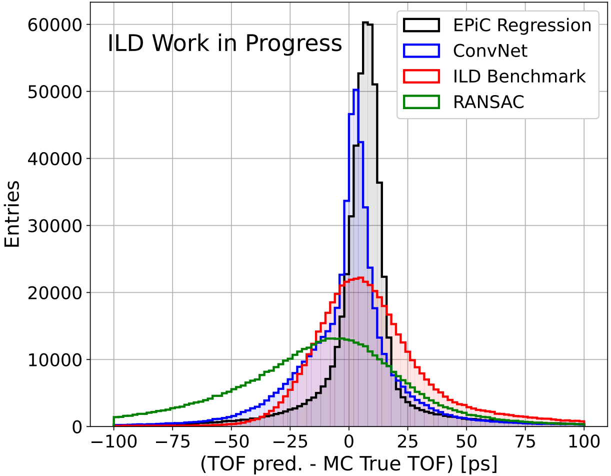
Backup

A Closer Look at the Benchmark Algorithm

$m_{\pi^\pm} \simeq 140\,\text{MeV}$
$m_{K^\pm} \simeq 490\,\text{MeV}$
$m_{\overset{(-)}{p}} \simeq 940\,\text{MeV}$



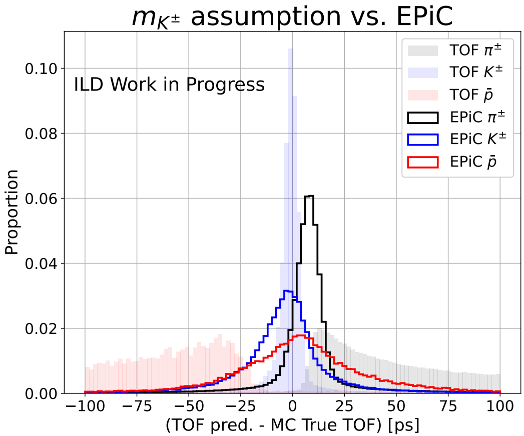
Performance Comparison to RANSAC

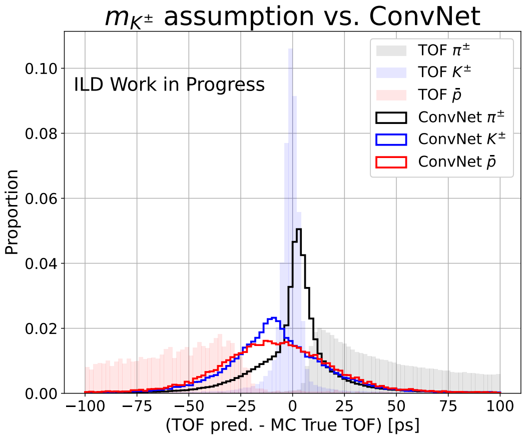


A Closer Look at the Results ... #5

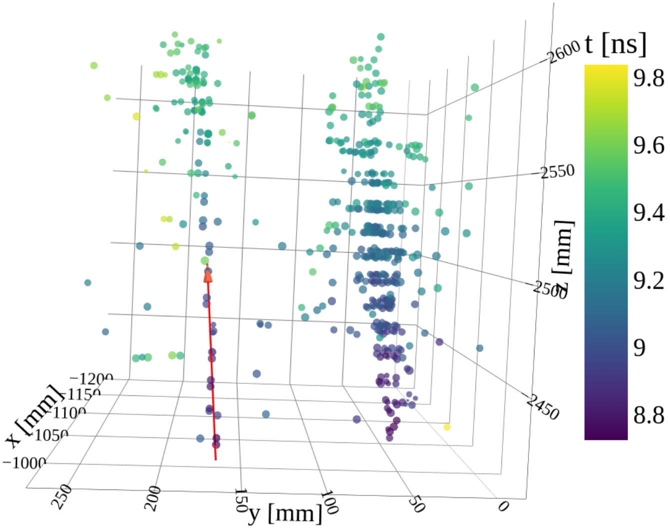
Assuming all Particles are $K^{\pm}$

$m_{\pi^{\pm}} \simeq 140\,\text{MeV}$
$m_{K^{\pm}} \simeq 490\,\text{MeV}$
$m_{\overset{(-)}{p}} \simeq 940\,\text{MeV}$





Problems


t [ns]
9.8
9.6
9.4
9.2
9
8.8
z [mm]
2600
2550
2500
2450
x [mm]
−1200
−1150
−1100
−1050
−1000
y [mm]
250
200
150
100
50
0