中国科学院高能物理研究所
Institute of High Energy Physics
Chinese Academy of Sciences

Progress in Drift Chamber Software

Wenxing Fang (IHEP)

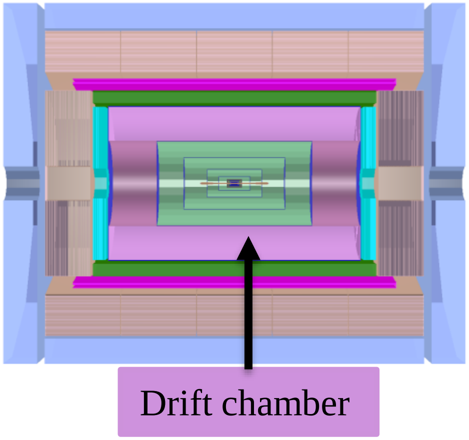
on behalf of the CEPC DC software working group

The CEPC day (2022.12.19)

Introduction

- The CEPC experiment aims to measure the property of the Higgs boson precisely
  - Requirements: high track efficiency (~100%), momentum resolution (<0.1%), PID (2σ p/K separation at P < ~ 20 GeV/c), …
  - Precise simulation is needed as it gives precise results

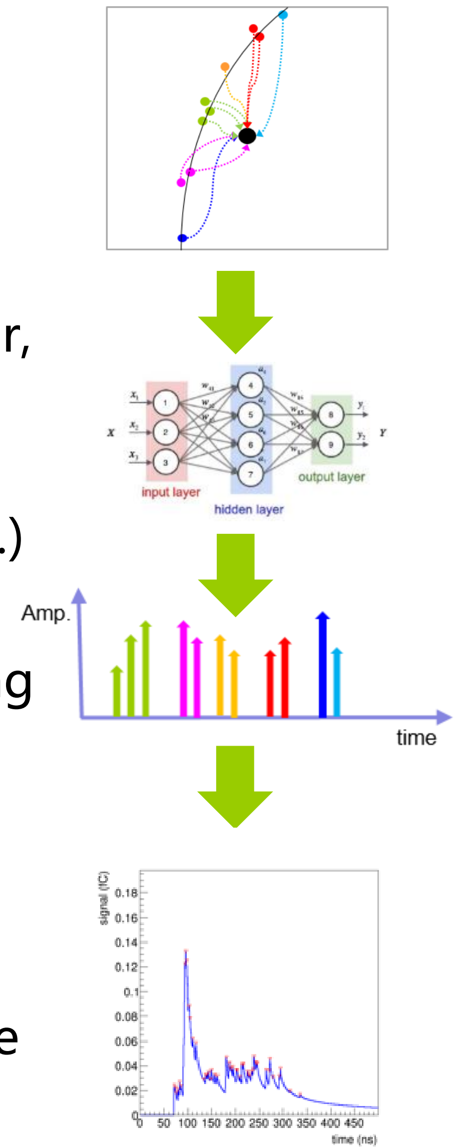
- The 4th conceptual detector design adopts silicon + drift chamber (DC) for tracking system
  - The DC: track reconstruction and PID (dN/dx+dE/dx)



- Performance of the DC needs to be studied with precise simulation

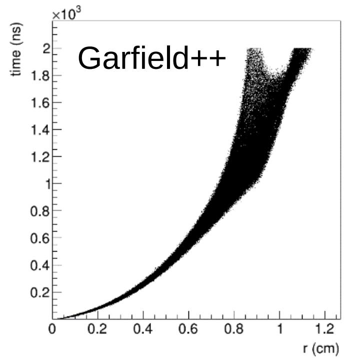
Towards precise DC simulation

- Originally driven by the dN/dx study:
  - Needs precise ionization simulation, waveform simulation
  - Geant4 + TrackerHeed + NN model is adopted
  - Geant4: for particle propagation (decay) in the detector, interaction with detector material, …
  - TrackerHeed (from Garfield++): used for ionization process simulation for charged particles (e, μ, π, K, p, …)
  - NN model: is used for fast pulse simulation for each ionized electron (primary particle independent), training data is from Garfield++
  - More details in this [talk]
  - Shortcoming, the space charge effect can not be simulated
    - arXiv:2211.06361, RPC simulation with space charge effect (dynamic update of the electric field)



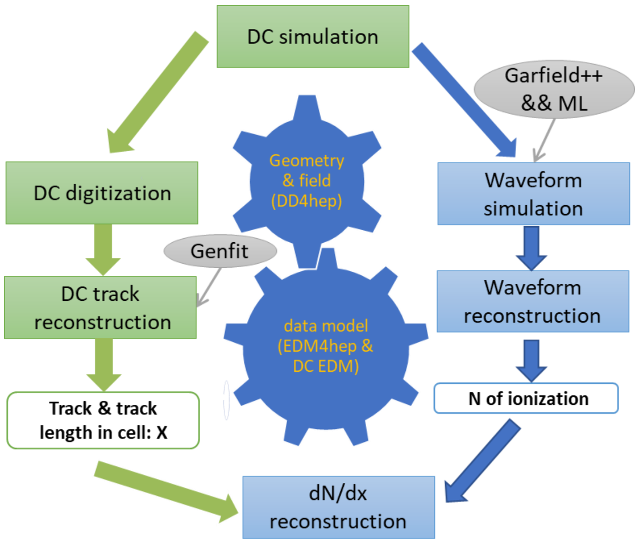
Towards precise drift time simulation

- The measurement of the earliest drift time for a fired DC cell is important for track reconstruction, as it will be converted to closet distance (between track and signal wire, via X-T relationship) and be input for tracking
- In previous: constant drift velocity V_{drift}=40μm/ns & fixed spatial resolution: σ=110μm
- Although the Garfield++ has some difficulties in space charge effect simulation, the earliest drift time simulation is almost independent of the space charge effect and can be simulated precisely
- Idea: learn drift time distribution from Garfield++ and apply it in the simulation



Overview of the DC software

- The drift chamber software has been developed from scratch
- The CEPCSW
  - Gaudi-based framework
  - External libraries and tools (Geant4, ROOT, …)
- Geometry and field map
  - DD4hep, supports a non-uniform magnetic field
- Data model
  - EDM4hep and FWCore
  - dN/dx event model extension

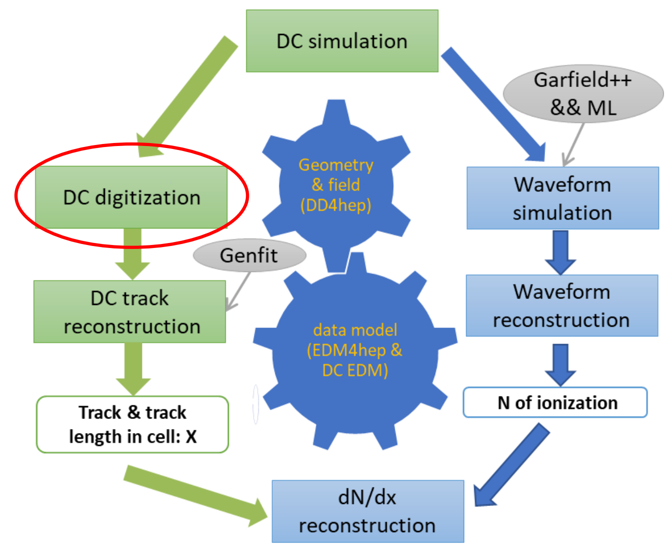


Drift chamber simulation and reconstruction flow

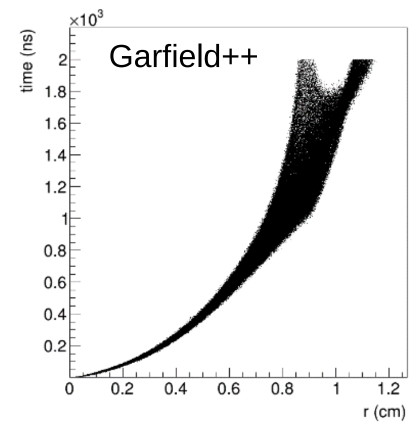
Half length	2980 *mm*
Inner and outer radius	800mm to 1800 *mm*
# of Layers	100/55
Cell size	~10mmx10mm/18mmx18mm
Gas	He:iC_4H_{10}=90:10
Single cell resolution	0.11 *mm*
Sense to field wire ratio	1:3
Total # of sense wire	81631/24931
Stereo angle	1.64~3.64 *deg*
Sense wire	Gold plated Tungsten ϕ=0.02*mm*
Field wire	Silver plated Aluminum ϕ=0.04*mm*
Walls	Carbon fiber 0.2 *mm*(inner) and 2.8 *mm*(outer)

The DC digitization

- In DC digitization
  - Time and charge measurements will be given for fired DC cells
  - The earliest drift time will be assigned to the measured time of this DC cell
  - Drift time simulation plays a role
- In DC track reconstruction:
  - Calibrated X-T relationship is used, which converts measured time to closet distance (between track and signal wire) which will be used for track reconstruction
- Full working flow:
  - 1, learn drift time from Garfield++ (NN) (this talk)
  - 2, do X-T calibration
  - 3, check the track reconstruction performance



Drift chamber simulation and reconstruction flow



Proof of the feasibility

- To proof of the feasibility of NN can learn complex drift time distribution, real data from the BESIII experiment is used
- Dataset: radiative Bhabha event
  - Detailed event selection [link]

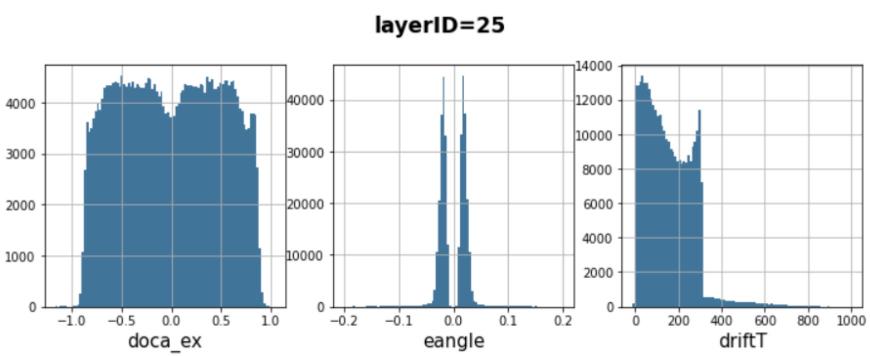
Prepare experimental data for training

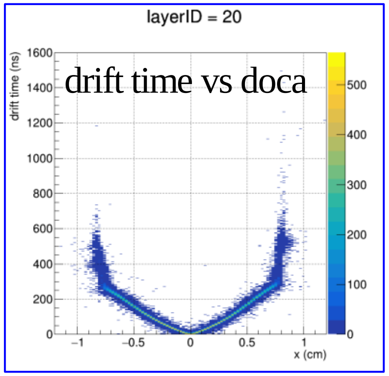
Learn drift time distribution: $D_{drift\ time} = f(doca, eangle, layer)$

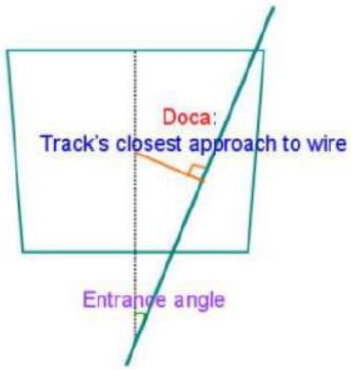
Check $D^{data}_{drift\ time}$, $D^{NN\ sim.}_{drift\ time}$

check the X-T cure: C^{data}_{X-T} , $C^{NN\ sim.}_{X-T}$

check spatial resolution: using C^{data}_{X-T} , $C^{NN\ sim.}_{X-T}$

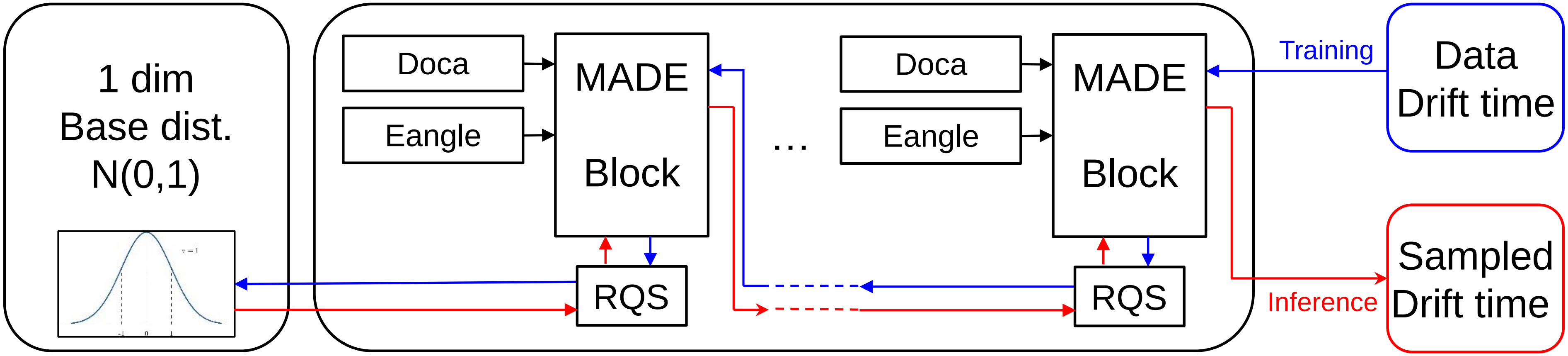




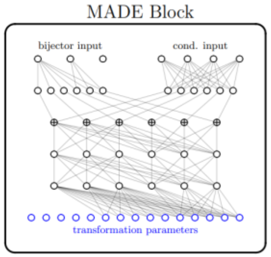


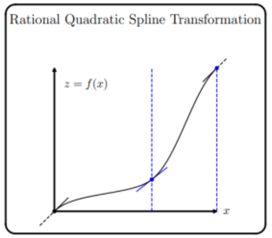
The NN model

- The Normalizing Flow is adopted:
  - Learning bijective transformation between two distributions(e.g. drift time ~ N(0,1))
  - Compared to GAN, it is much easier to train (stable and convergent)
  - Reference to the CaloFlow, a similar model is used, RQS (for transformation)+MADE block (for the parameters of RQS)



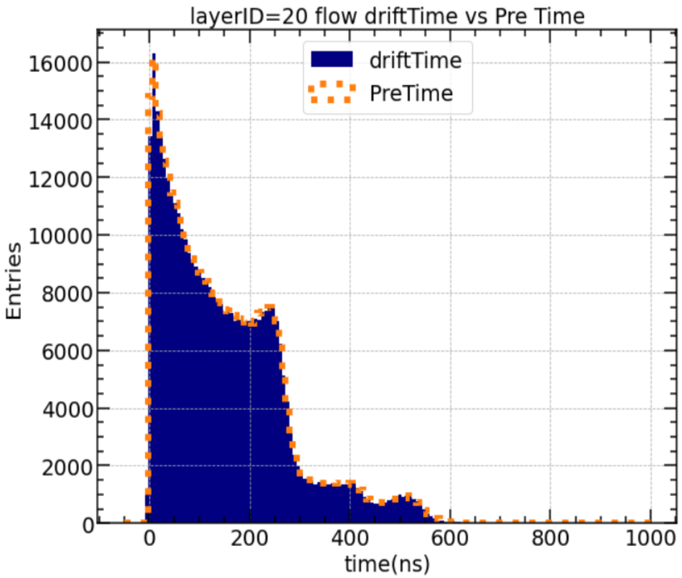
Base distribution	Number of MADE blocks	Layer sizes			Number of RQS bins
		input	hidden	output	
1-dim Standard Normal	6	64	3×64	23	8

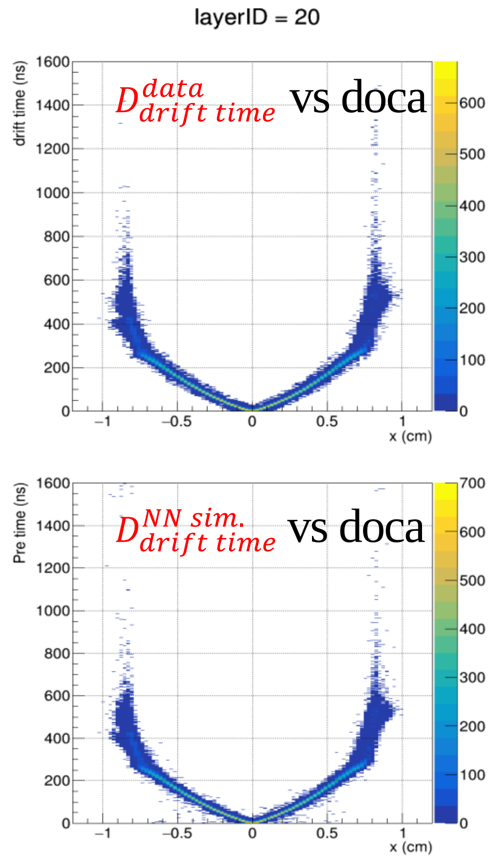




NN simulation performance

- Compared the drift time distribution between data and NN simulation





Prepare experimental data for training

Learn drift time distribution: $D_{drift\ time} = f(doca, eangle, layer)$

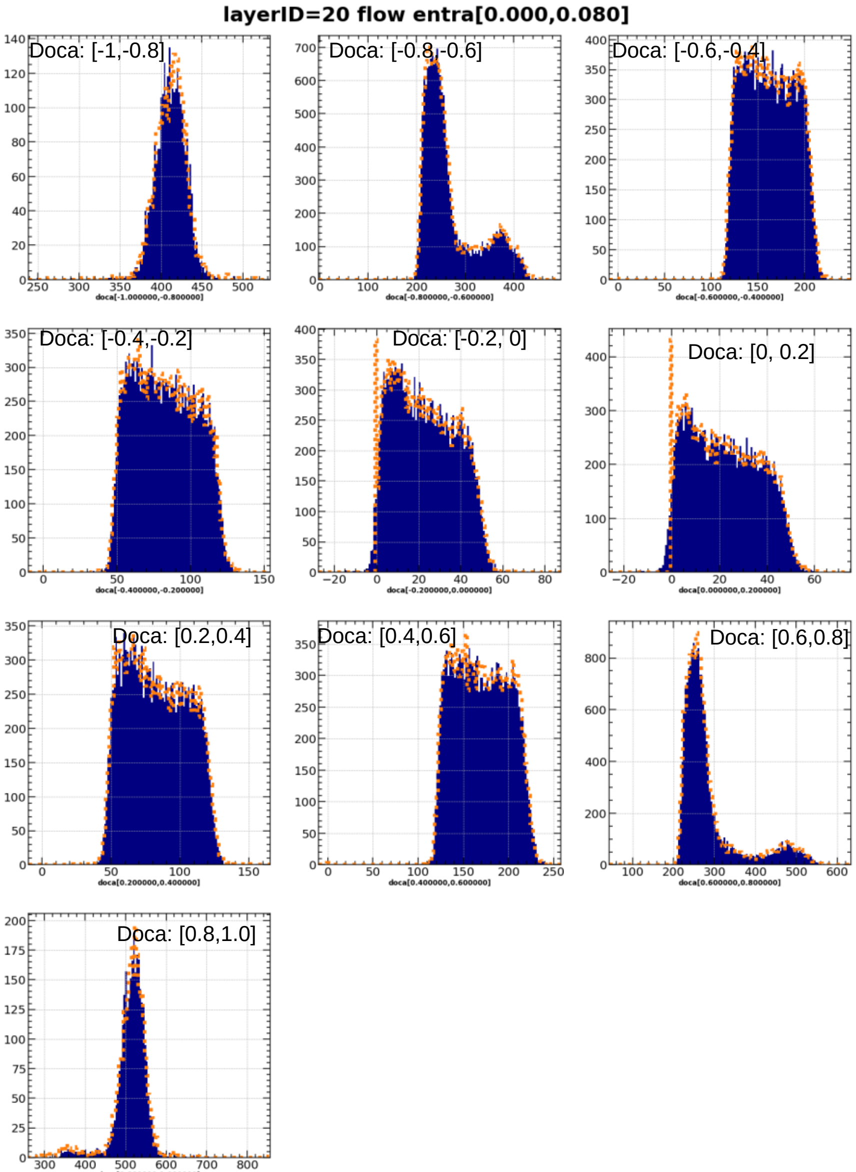
Check $D_{drift\ time}^{data}$, $D_{drift\ time}^{NN\ sim.}$

check the X-T cure: C_{X-T}^{data}, $C_{X-T}^{NN\ sim.}$

check spatial resolution: using C_{X-T}^{data}, $C_{X-T}^{NN\ sim.}$

- Consistent between data and NN simulation

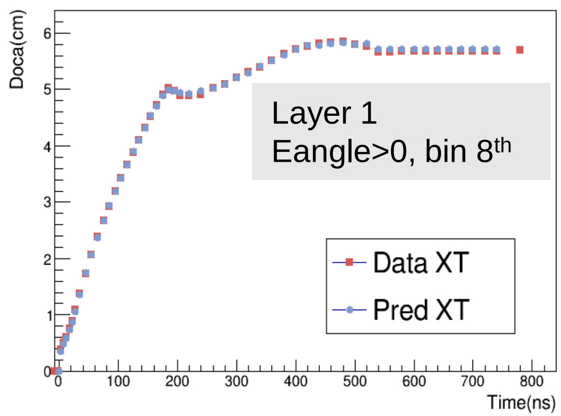
NN simulation performance

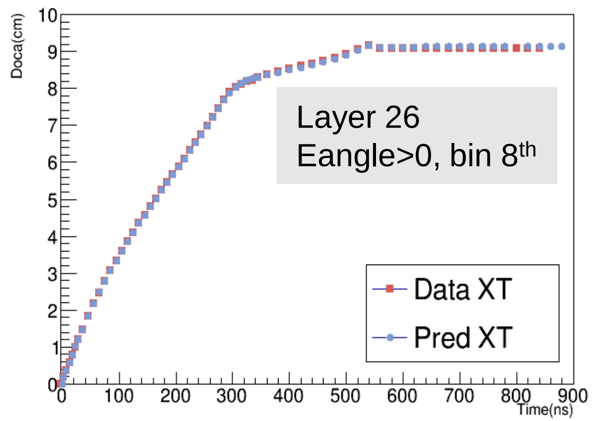


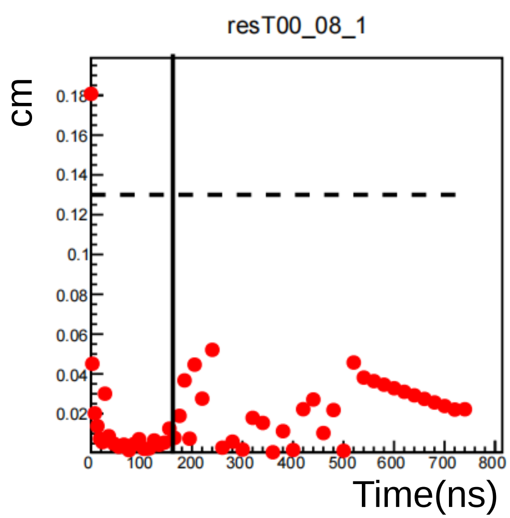
- A detailed check, in different entrance angles and Doca region.
- In general, good agreement between data and NN simulation
- Drift time distributions are flatter in small Doca regions (|Doca|<0.6) , complex for the large Doca regions (close to cell edge)

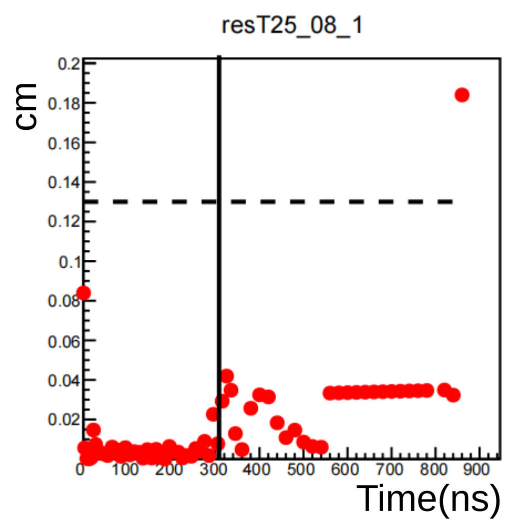
X-T relationship curve

- Check the X-T curve from the experimental data and NN simulation









Prepare experimental data for training

↓

Learn drift time distribution: $D_{drift\ time} = f(doca, eangle, layer)$

↓

Check $D^{data}_{drift\ time}$, $D^{NN\ sim.}_{drift\ time}$

↓

check the X-T cure: C^{data}_{X-T} , $C^{NN\ sim.}_{X-T}$

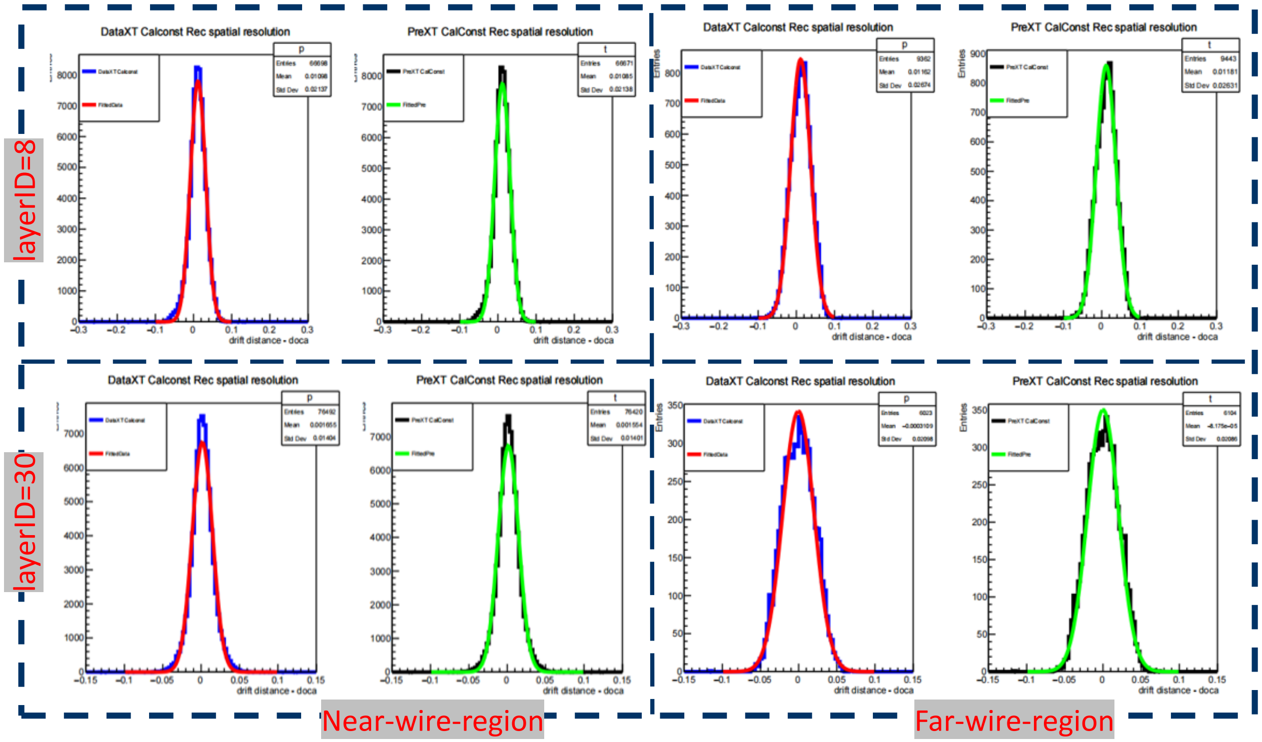
↓

check spatial resolution: using C^{data}_{X-T} , $C^{NN\ sim.}_{X-T}$

- The mean Docas versus time are obtained for different layers and entrance angle
- red for C^{data}_{X-T}, blue for $C^{NN\ sim.}_{X-T}$
- Difference is < 40 μm

Spatial resolution check

- Checked the spatial resolution of reconstructed track, using C_{X-T}^{data} and $C_{X-T}^{NN\ sim.}$



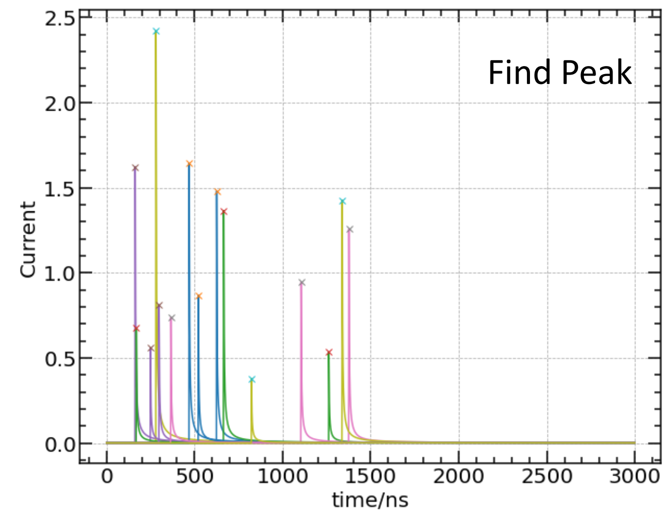
Prepare experimental data for training
↓
Learn drift time distribution: $D_{drift\ time} = f(doca, eangle, layer)$
↓
Check $D_{drift\ time}^{data}$, $D_{drift\ time}^{NN\ sim.}$
↓
check the X-T cure: C_{X-T}^{data} , $C_{X-T}^{NN\ sim.}$
↓
check spatial resolution: using C_{X-T}^{data} , $C_{X-T}^{NN\ sim.}$

layerID	Near-wire-region (C_{X-T}^{data} v.s. $C_{X-T}^{NN\ sim.}$)		Far-wire-region (C_{X-T}^{data} v.s. $C_{X-T}^{NN\ sim.}$)	
	mean	sigma	mean	sigma
8	0.01096/0.01085	0.02137/0.02138	0.01162/0.01181	0.02674/0.02631
30	0.001055/0.001564	0.01404/0.01401	-3.1e-4/-5.1e-5	0.02060/0.02086

- Consistent spatial resolution results by using C_{X-T}^{data} and $C_{X-T}^{NN\ sim.}$
- The NN can learn the drift time from the data

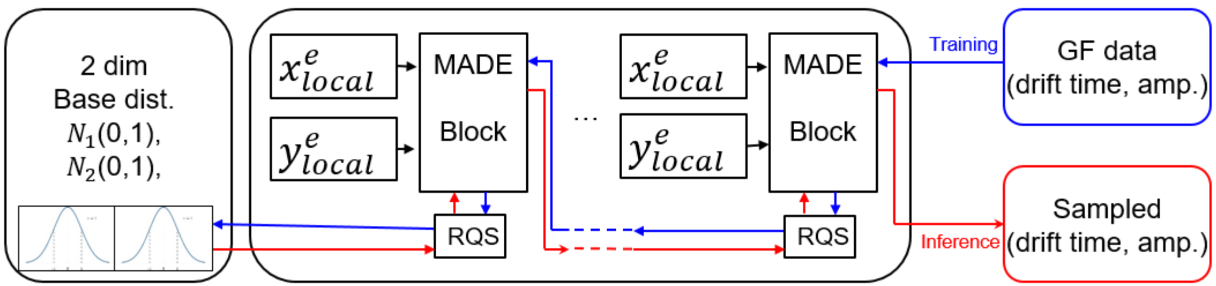
Drift time simulation in CEPC

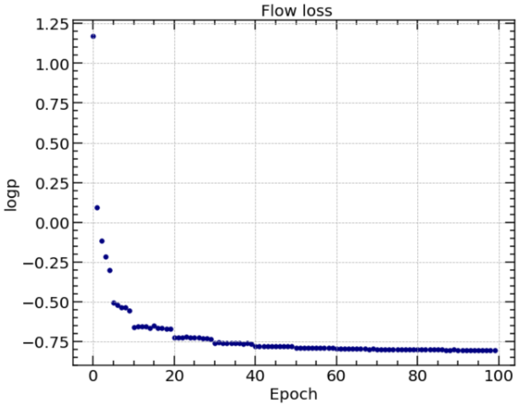
- Using a similar method for the CEPC drift time simulation
- Produce Garfield++ simulation data for training:
  - DC Cell size is 18 mm* 18 mm
  - Gas: 90%He+10%C_4H_{10}
  - Signal wire voltage: 1630 V
  - For each event, an ionized electron is uniformly generated in the cell (x,y) and the pulse is simulated
  - Then peak finding algorithm (scipy.signal.find_peaks()) is used to get pulse time (drift time) and amplitude (used for threshold cut)

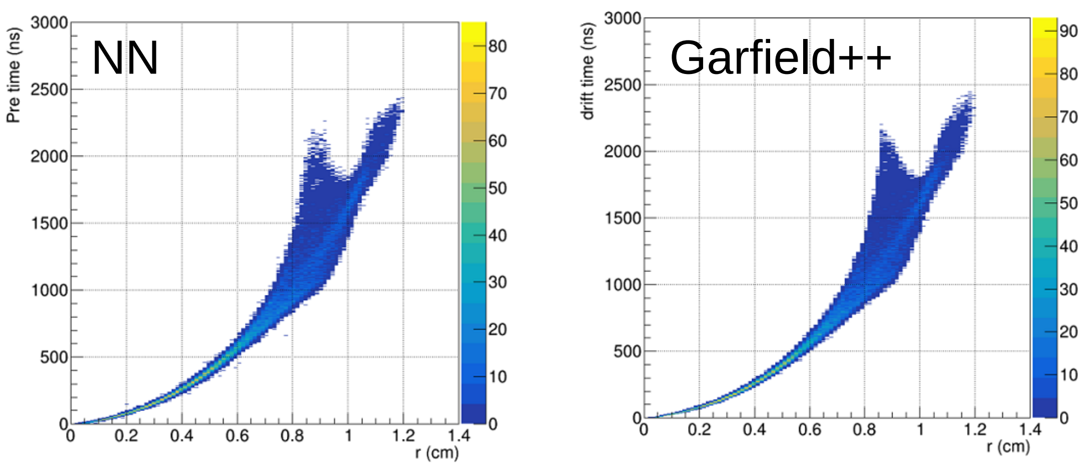


Drift time simulation in CEPC

- Similar NF model is used for training







- By using the NN model, one can simulate the drift time and pulse amplitude according to the cell local x,y position of the ionized electron

Summary and plan

- Precise simulation is important for the CEPC, the DC software is now moving on precise simulation
- Similar to dN/dx simulation, a precise drift time simulation is performed by learning Garfield++ data sample
- To prove the feasibility of the method, real data from the BESIII experiment is used. The results show the NN is able to learn the drift time precisely
- Next step:
  - the X-T calibration should be performed. An iterative process between track reconstruction and X-T calibration
  - Using cell local positions of ionized electrons (simulated by TrackerHeed) as input of the NN

Thanks for your attention!

Back up

Prepare experimental data for training

↓

Learn drift time distribution: $D_{drift\ time} = f(doca, eangle, layer)$

↓

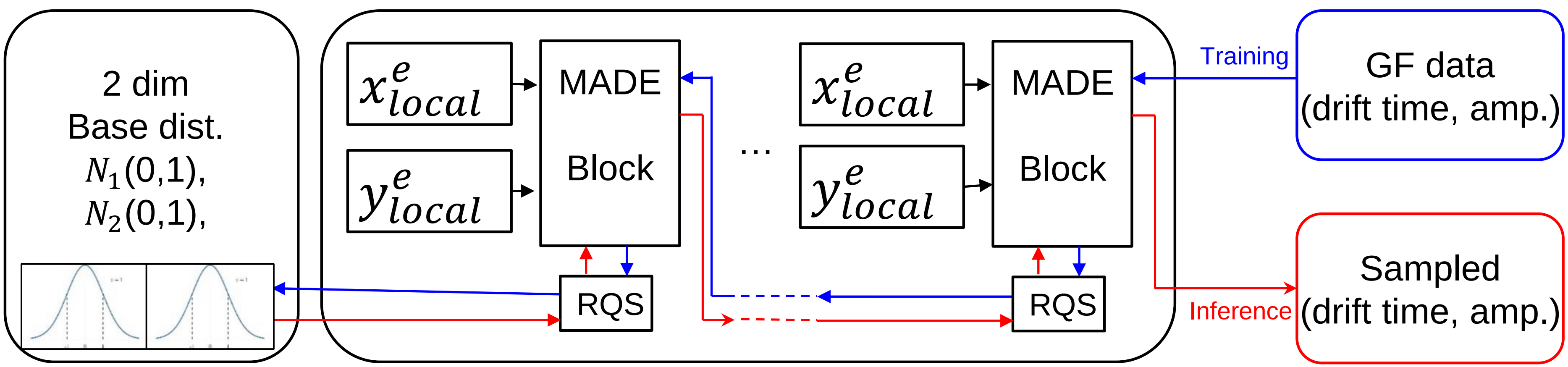
Check $D^{data}_{drift\ time}$, $D^{NN\ sim.}_{drift\ time}$

↓

check the X-T cure: C^{data}_{X-T} , $C^{NN\ sim.}_{X-T}$

↓

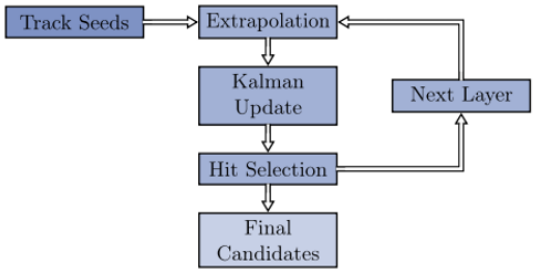
check spatial resolution: using C^{data}_{X-T} , $C^{NN\ sim.}_{X-T}$



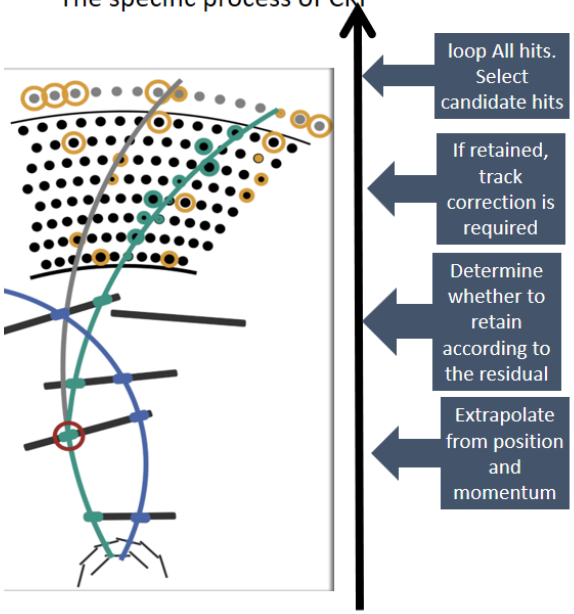
Track Finding by CKF

- Combinatorial Kalman Filter (CKF)
  - A tracking concept that combines track finding and track fitting in a search-tree-based algorithm.
  - Used by many high energy physics experiments
- Track finding using CKF in Drift chamber
  - Methods: The reconstructed SiTrack is selected as seed to select DC hits belonging to the same track
- Current progress:
  - Reusing CKF from Belle II and running it in CEPCSW

Basic procedure behind the CKF

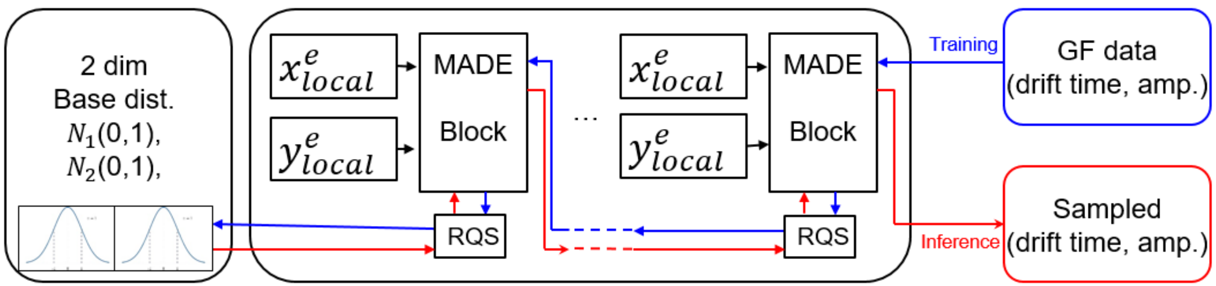


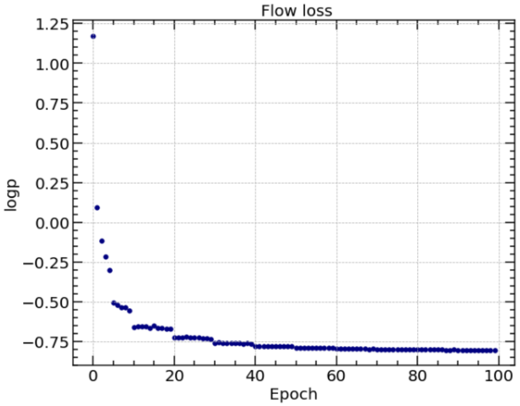
The specific process of CKF



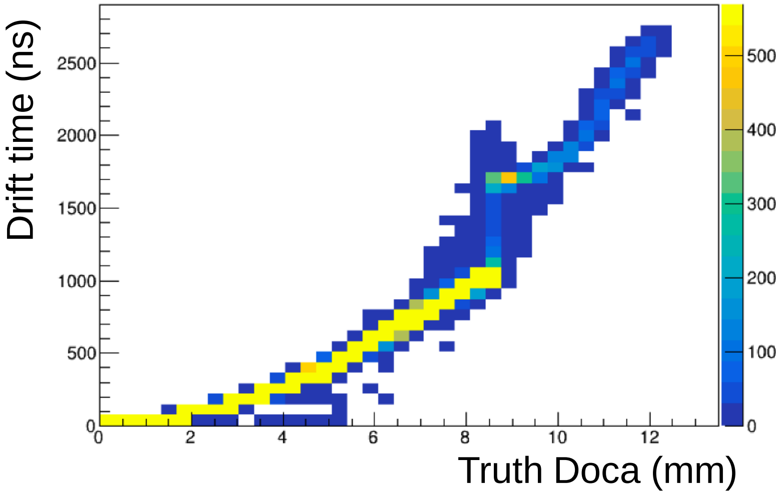
Drift time simulation in CEPC

- Similar NF model is used for training



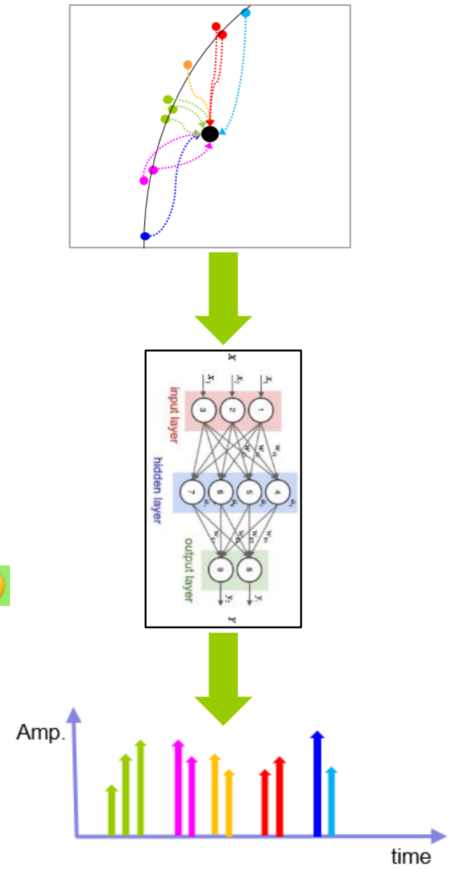


- After obtaining the NN model, one can simulate the drift time and pulse amplitude according to the local x,y position of the ionized electron



Ionization simulation

- The ionization simulation is done by combining Geant4 and TrackerHeed
  - TrackerHeed (from Garfield++) used for ionization process simulation
  - Geant4 for particle propagation (decay) in the detector, interaction with detector material, …
- Pulse simulation for each ionized electron
  - The Garfield++ simulation takes a long time 😟
  - NN is used for fast simulation, simulating the time and amplitude of each pulse, more details in this talk 😊



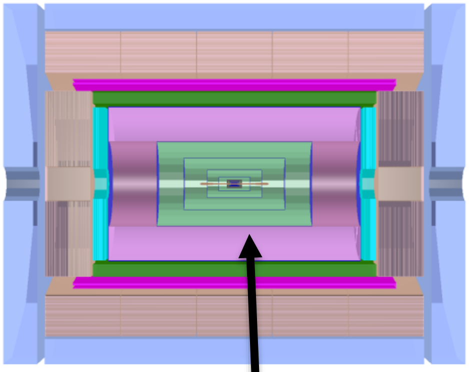
Summary

- The DC simulation and track reconstruction in the CEPCSW is presented
- The multi-track fitting has been developed with Si+DC measurements and the performance is reasonable
- The chain of dN/dx study in CEPCSW is presented
- The preliminary results for dN/dx PID performance in CEPCSW are checked, they are in good agreement with the results from the standalone Garfield++ simulation
- Future plan:
    - Continue working on track finding
    - Track reconstruction with background mixing and using more realistic X-T relation
    - More detailed dN/dx performance check

Thank you for your attention!

Drift Chamber(DC) Software

- Drift chamber is the key detector in the 4th conceptual detector design to provide PID
  - Good PID ability (2σ p/K separation at P < ~ 20 GeV/c)
  - Precise momentum measurement (eff. ~100%, σ_p<=0.1%)
- Motivation of DC software project
  - Development of simulation and reconstruction for DC
  - Support the detector design, optimization and performance study
  - Support physics sensitivity study
- Requirements for DC software
  - Modular design and friendly interfaces
  - Easily integrated with common tools (ACTS, Genfit etc.)
  - Reuse existing algorithms from other experiments
  - Application of advanced technic (ML) to simulation and reconstruction
- Manpower
  - IHEP: Yao Zhang, Tao Lin, Wenxing Fang, Chengdong Fu, Ye Yuan, Weidong Li
  - SDU: Mengyao Liu, Xueyao Zhang, Xingtao Huang



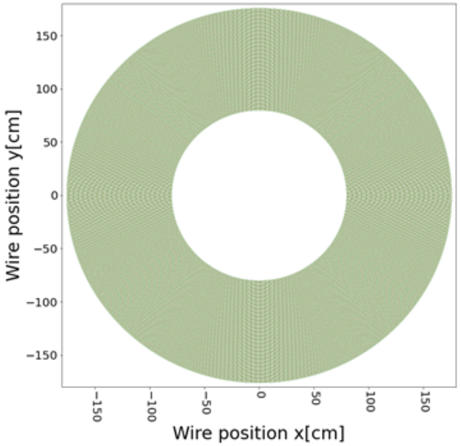
A PID drift chamber

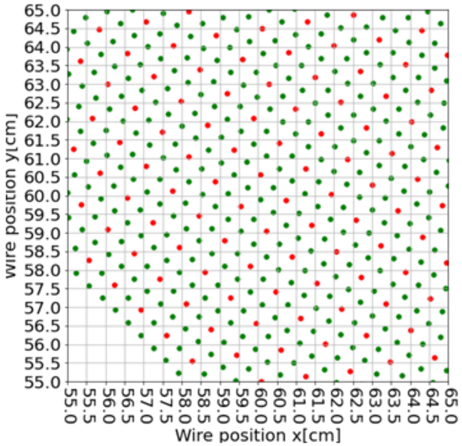
Physics process	Measurands	Detector subsystem	Performance requirement
$ZH, Z \to e^+e^-, \mu^+\mu^-$ $H \to \mu^+\mu^-$	$m_H, \sigma(ZH)$ $\mathrm{BR}(H \to \mu^+\mu^-)$	Tracker	$\Delta(1/p_T) = 2 \times 10^{-5} \oplus \frac{0.001}{p(\mathrm{GeV}) \sin^{3/2}\theta}$

Drift Chamber Parameters in CEPCSW

- The baseline configuration of DC in CEPCSW

Half length	2980 *mm*
Inner and outer radius	800mm to 1800 *mm*
# of Layers	100/55
Cell size	~10mmx10mm/18mmx18mm
Gas	He:iC_4H_{10}=90:10
Single cell resolution	0.11 *mm*
Sense to field wire ratio	1:3
Total # of sense wire	81631/24931
Stereo angle	1.64~3.64 *deg*
Sense wire	Gold plated Tungsten ϕ=0.02*mm*
Field wire	Silver plated Aluminum ϕ=0.04*mm*
Walls	Carbon fiber 0.2 *mm*(inner) and 2.8 *mm*(outer)

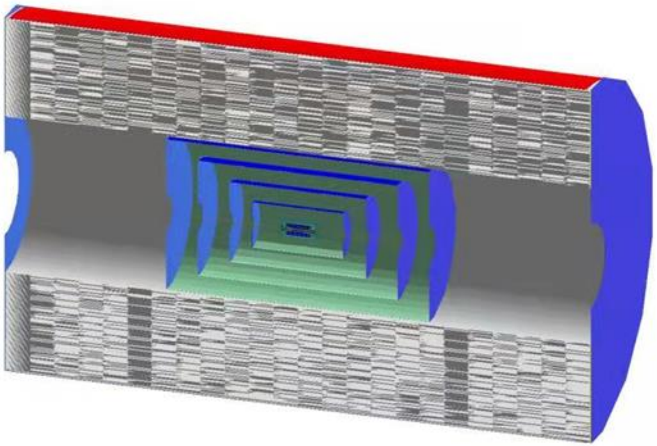




Cell structure

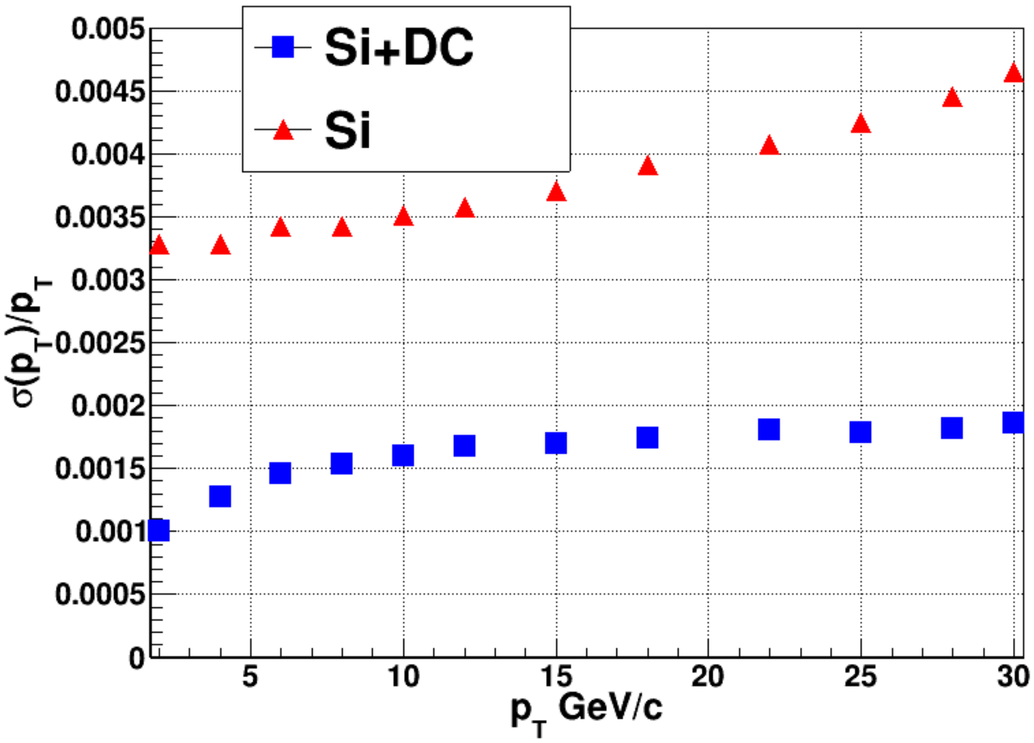
Silicon detectors Parameters in CEPCSW

Silicon tracker	*Number of layer*	*Radius(mm)*	$\sigma_U\,(\mu m)$	$\sigma_V(\mu m)$
VXD	3 double layers	16-58	2.8/6/4/4/4/4	2.8/6/4/4/4/4
SIT	4 layers	230-770	7.2	86
SOT(SET)	1 layer	1815	7.2	86



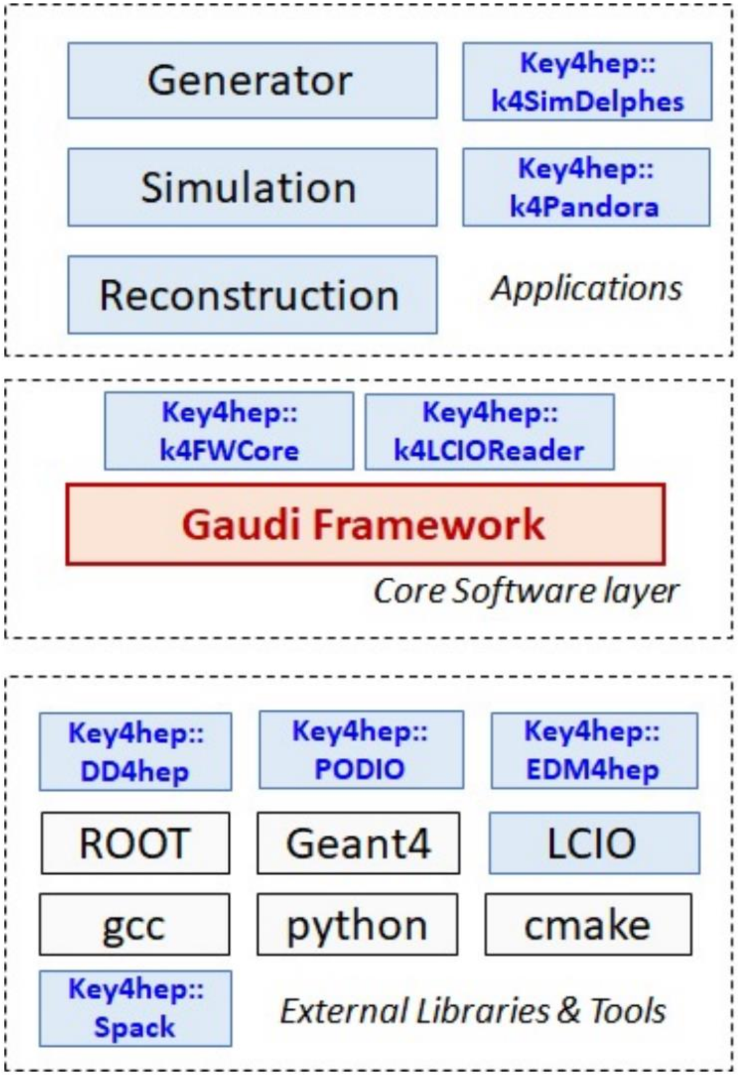
Silicon+DC vs Silicons

- Got better momentum measurement with the drift chamber



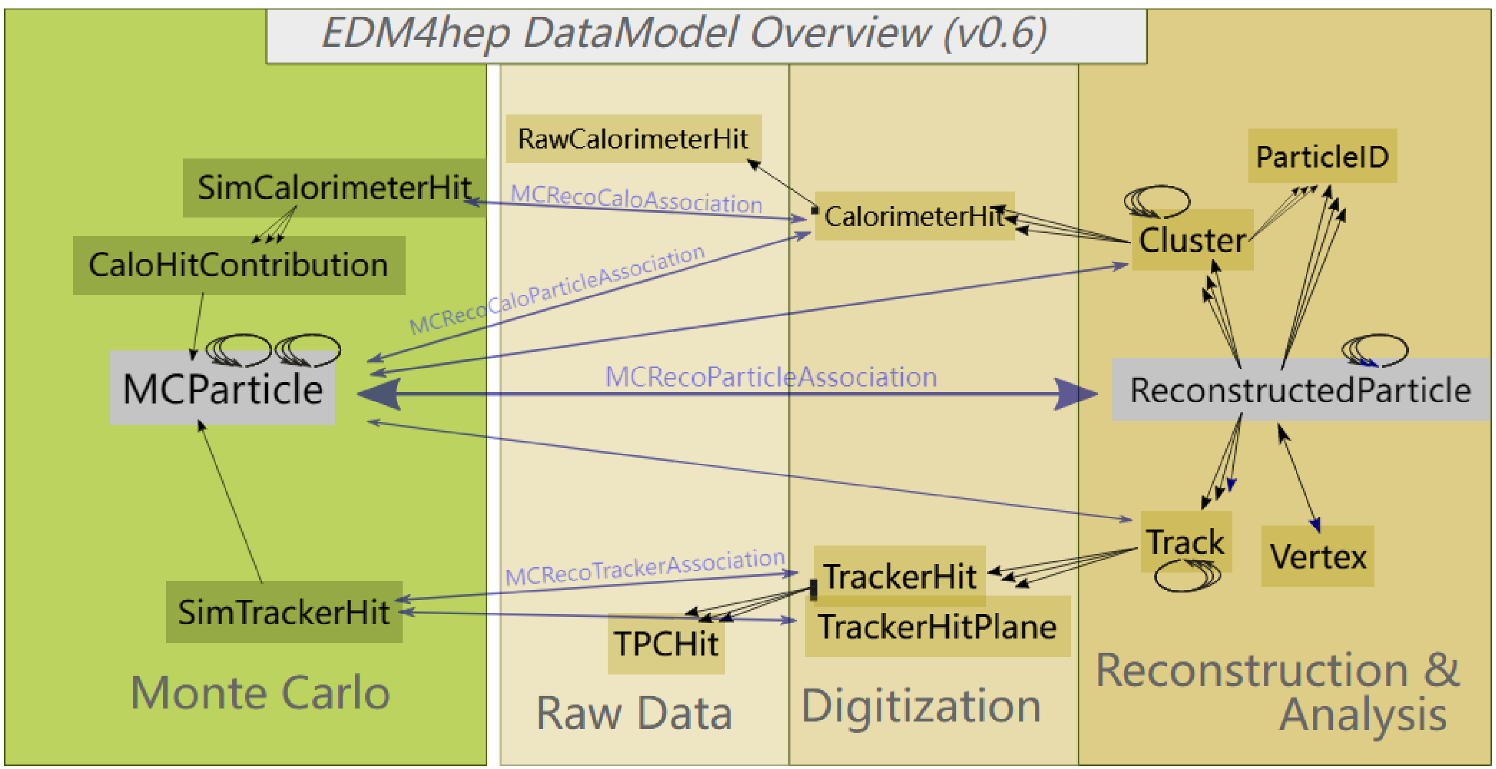
CEPCSW Software

- CEPCSW software structure
  - External libraries:
    - DD4hep: complete detector description (geometry, B field, Material, …). Consistent description (simulation, reconstruction, analysis)
    - EDM4hep: the generic event data model for HEP experiments (see next slide)
    - …
  - Core software:
    - Gaudi framework: defines interfaces to all software components and controls their execution
    - K4FWCore: data service for EDM4hep
  - Applications:
    - CEPC-specific software: generator, Gean4 simulation, reconstruction, and analysis



EDM4hep

- Common EDM: ILC, FCC, CEPC, CLIC, …
- Efficiently implemented (fast data access, efficient memory usage)
- Support multi-threading
- Potentially heterogeneous computing
- Easy to generate the C++ code from a high-level description of the desired EDM (YAML file) using the podio



EDM4hep Extension

- Currently, the EDM4hep does not include the EDM for dN/dx study, we extended it by using the extension mechanism of podio (very convenient)
- Following EDMs are extended (more details in following slides):
  - SimPrimaryIonizationCluster
  - TrackerData
  - TrackerPulse
  - RecIonizationCluster
  - RecDndx
- The extended EDM is supposed to be used both for the drift chambers and the TPC