

1. Introduction

Drift chamber: gas ionization detector for charged particles tracking

Baseline tracking: current FDC tracking method* at RNC

FDC: forward drift chamber, vertical type, 14 layers, 3 wire orientations (x, u, v)

*M. Kurata-Nishimura et al., RIKEN Accel. Prog. Rep. 47 (2014) 176

a) Hit candidate selection (per orientation)

1D linear fit over all combinations

b) Track parameter fitting (X, A, Y, B)

Use hit candidates from all orientations

Tracking speed (CPU-based)

- Baseline: ~ 1 s / 10k events (exp. data, ~ 17 hits/event)

- Raw data decoding $\sim 75\%$, fitting $\sim 25\%$

Challenge

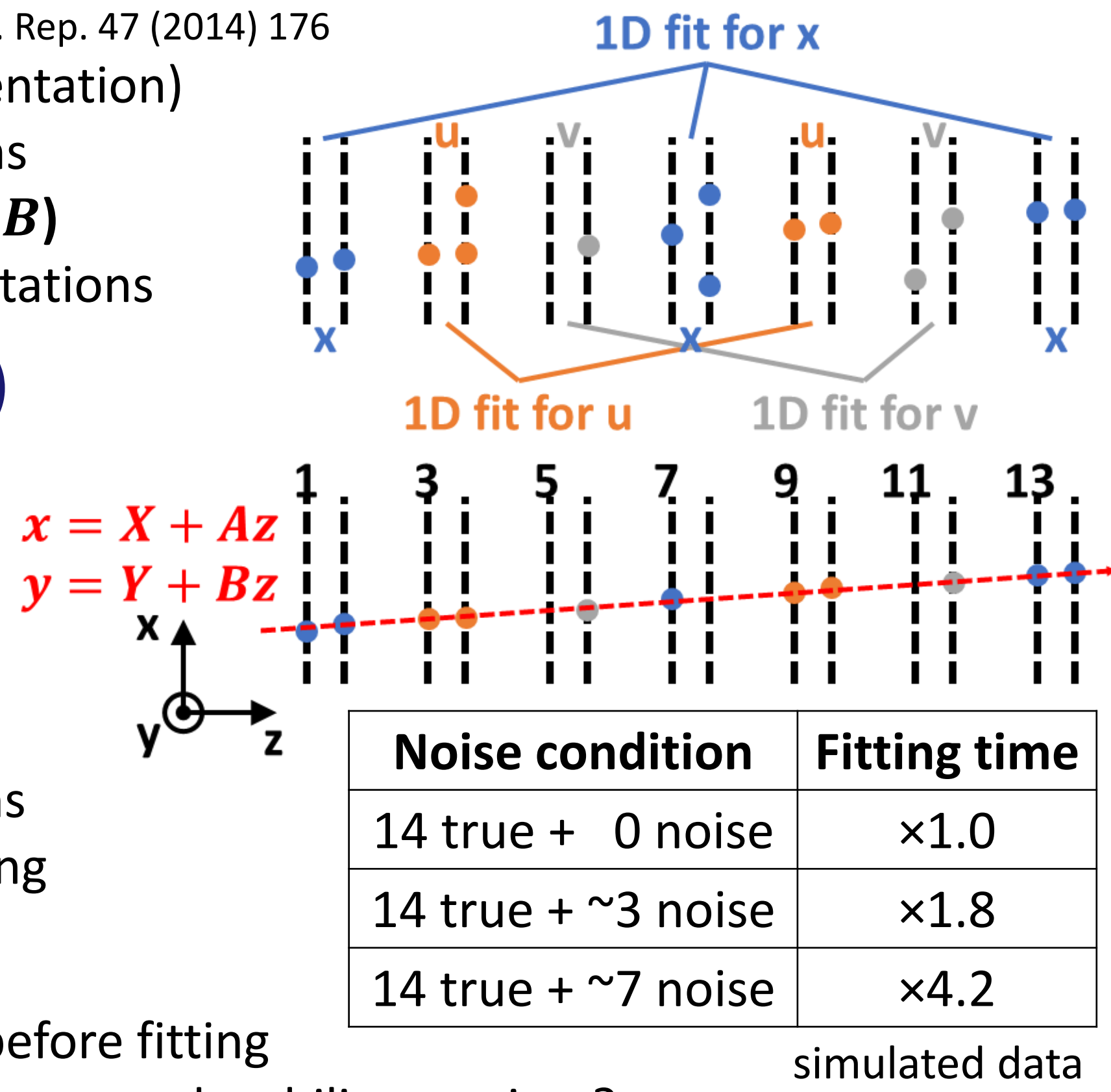
Noise hits increase hit combinations

\rightarrow slow down CPU-based track fitting

Goal

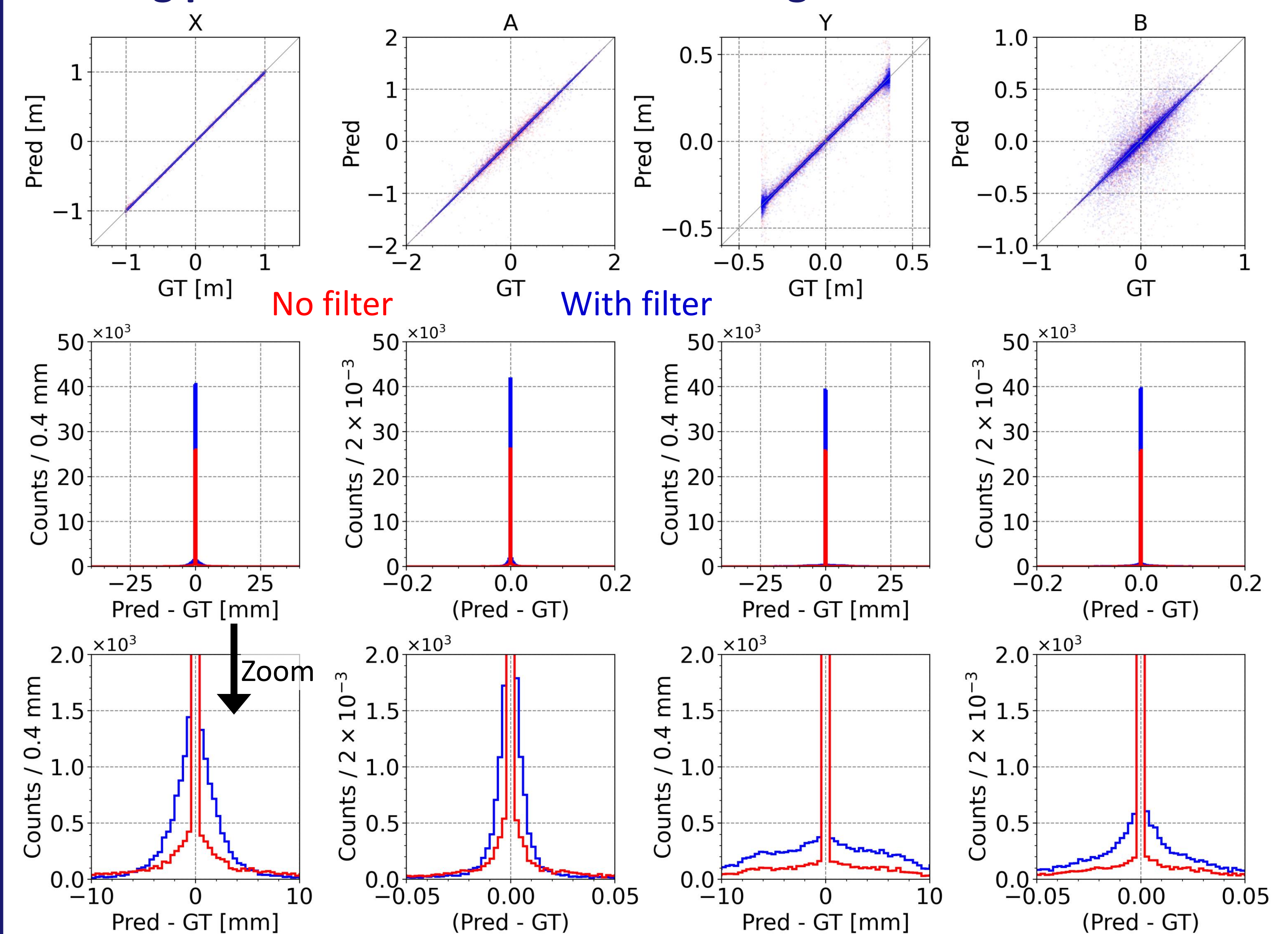
Apply hit filtering to reduce noise before fitting

\rightarrow improve reconstruction performance and stabilize runtime?



3. Results & Discussions (cont.)

Tracking performance with DL hit filtering



n_{event}	Passed fitting χ^2 cut	Correctly reconstructed
No filter	$\sim 60\%$	$\sim 52\%$
With filter	$\sim 95\%$	$\sim 80\%$

Correctly reconstructed: within two central bins around zero

Runtime impact of DL hit filtering

Noise condition	Hits / event	Fitting time
14 true + 0 noise	$\mu = 14, \sigma = 0$	$\times 1.0$
14 true + ~ 3 noise	$\mu = 17, \sigma = 3$	$\times 1.8$
14 true (eff=0.9) + ~ 5 noise	$\mu \approx 16.7, \sigma \approx 3.1$	$\times 2.0$
14 true (eff=0.9) + ~ 5 noise	$\mu \approx 11.7, \sigma \approx 2.1$	$\times 0.9$

Analysis scheme	Events	Runtime component [s]				Total
		Fixed overhead	Raw decoding	DL hit filtering	Track fitting	
CPU baseline	10k	CPU: ~ 0.5	~ 1.0 (decode + fit)			~ 1.5
DL filter + Opt. CPU impl.*	10k	CPU: ~ 0.5 GPU: ~ 0.15 (ref.†)	~ 0.15 (ref.†)	~ 0.8	~ 0.02	~ 1.62
DL filter (batched) + Opt. CPU impl.	100k	CPU: ~ 0.5 GPU: ~ 0.15 (ref.)	~ 1.5 (ref.)	~ 2.9	~ 0.2	~ 5.25

Large-batch parallel processing: 10 \times events \rightarrow $\sim 3.6\times$ filtering time

*optimized CPU implementation
†reference from experimental data

2. Methods & Setups

Graph neural network (GNN)

- Graph: per event

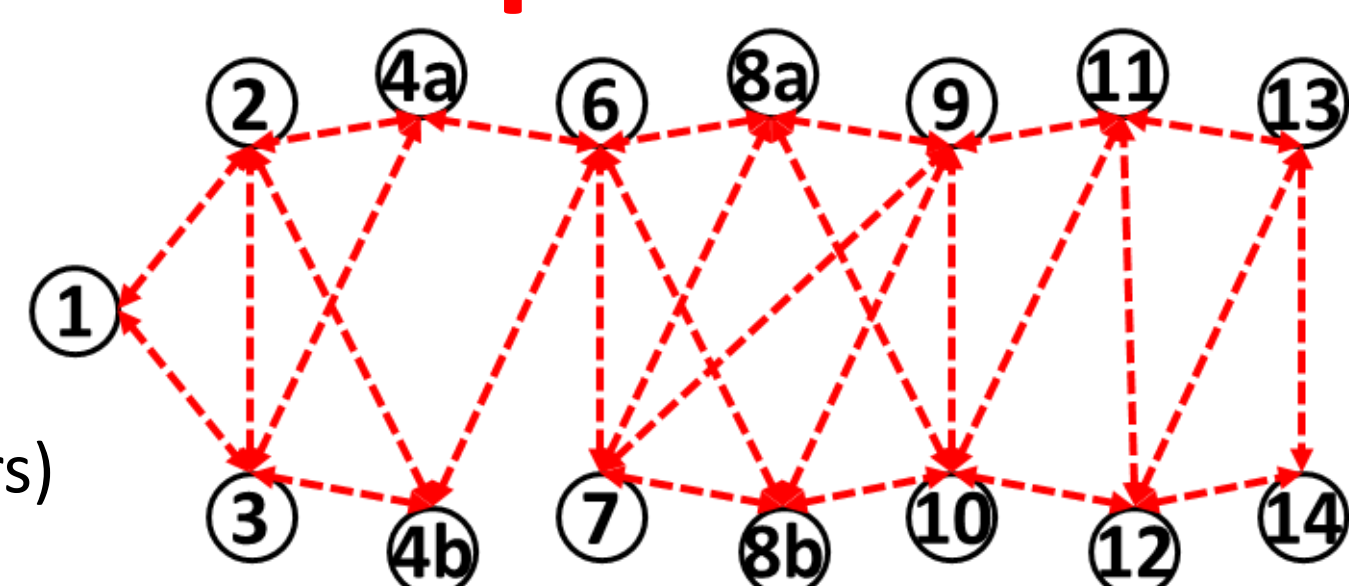
- Node: each drift-chamber hit

- Edge: connect neighboring hits (≤ 2 layers)

- Model:

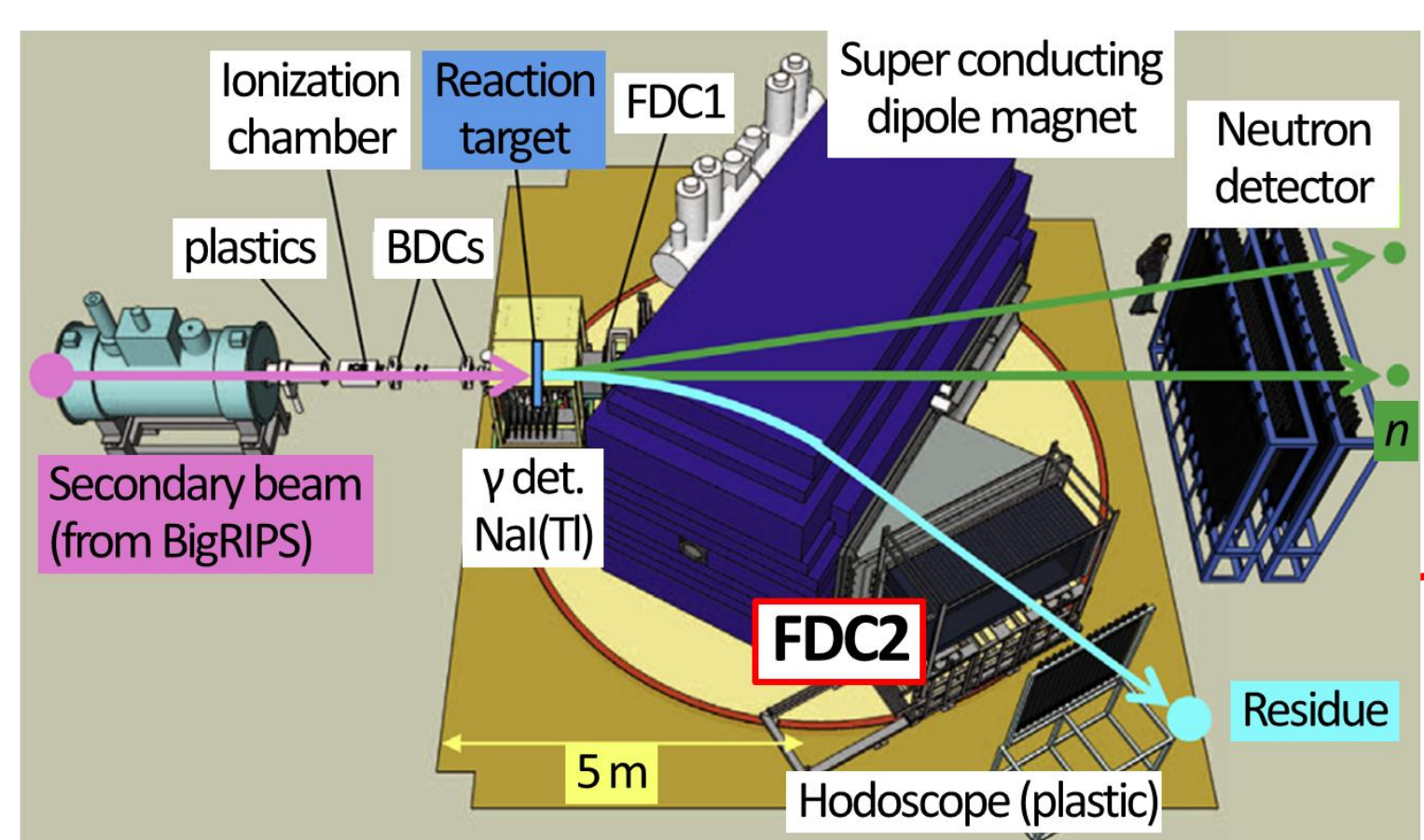
3-layer graph attention network (GAT) + fully connected layers (MLP)

- Output: true hit or noise

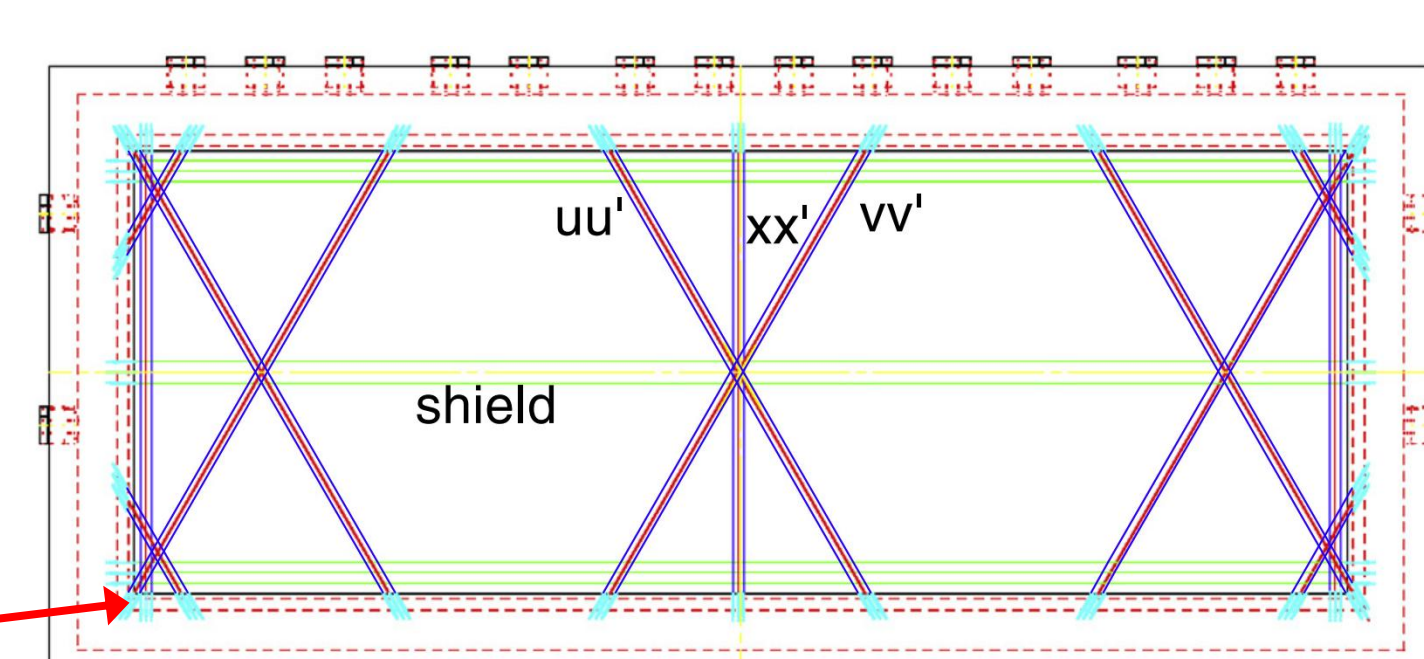


Drift chamber configuration & simulation

Train DL models: simulated FDC2 hits with true/noise labels



H. Otsu et al. / NIM B 376 (2016) 175–179



Simulation setup

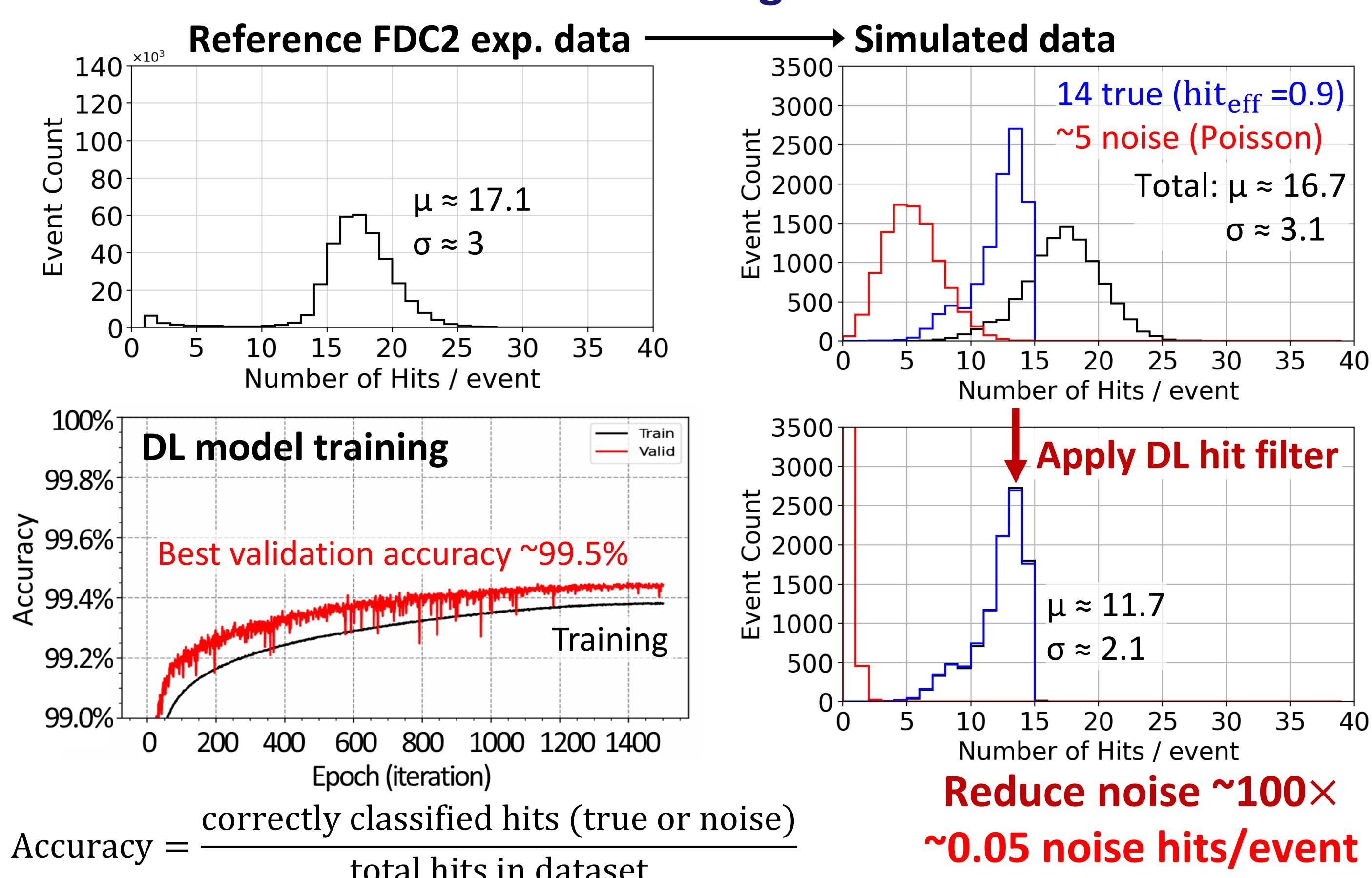
X, A, Y, B	within detector acceptance
True hit	nearest wire to track-layer intersection
Noise	Random to non-true wires
Drift length	True: track-wire distance Noise: uniform 0–10 mm

FDC2 specifications

Size	2.2 m \times 0.8 m \times 0.8 m
Layer config	14 layers (7 pairs) xx' uu' vv' xx' uu' vv' xx' xx': 0°, uu': +30°, vv': -30°

3. Results & Discussions

Noise reduction with DL hit filtering



4. Summary

DL hit filtering was explored to reduce noise before CPU-based drift chamber track fitting.

- GNN-based hit filtering reduced noise hits by nearly 100 \times

- Filtered hits improved CPU track fitting:

Fitting χ^2 passed events: 60% \rightarrow 95%

Correctly reconstructed events: 52% \rightarrow 80%

- Filtering mitigated noise-induced fitting slowdown

- Large-batch parallel processing improved DL-filtering throughput

Outlook

- Explore lower-cost DL architectures, e.g., CNN?

- Study CPU/GPU inference choices for small-batch processing.

- Extend toward a full DL-based tracking pipeline.