AI/ML-Based Geometric Alignment of the ALICE ITS2 for LHC Run 3

June 11th, 2026

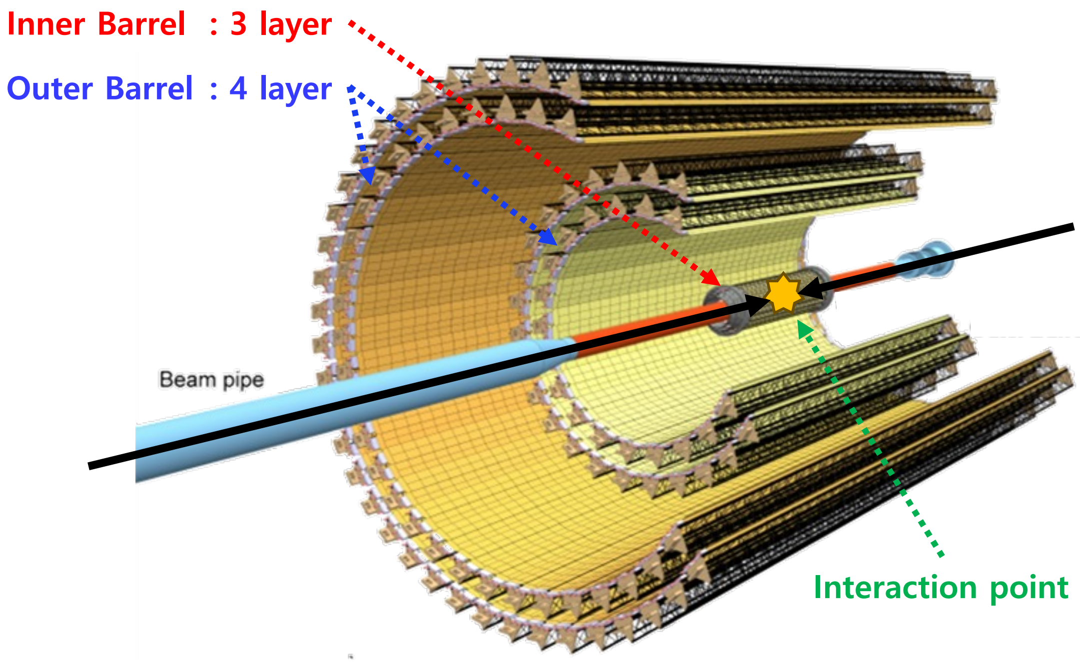
Presenter: Jaehyun Kim

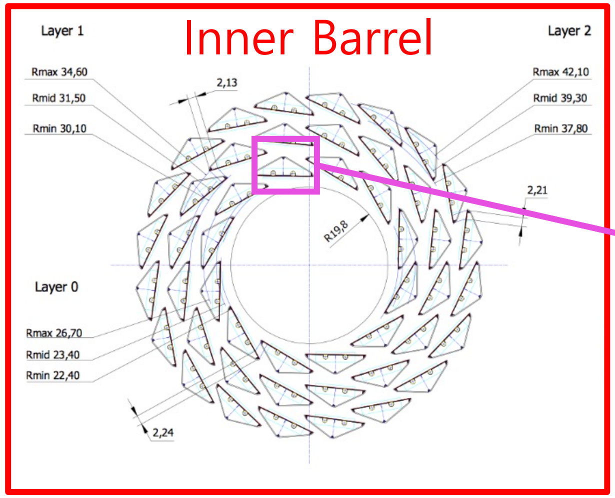
Department of Physics, Yonsei University

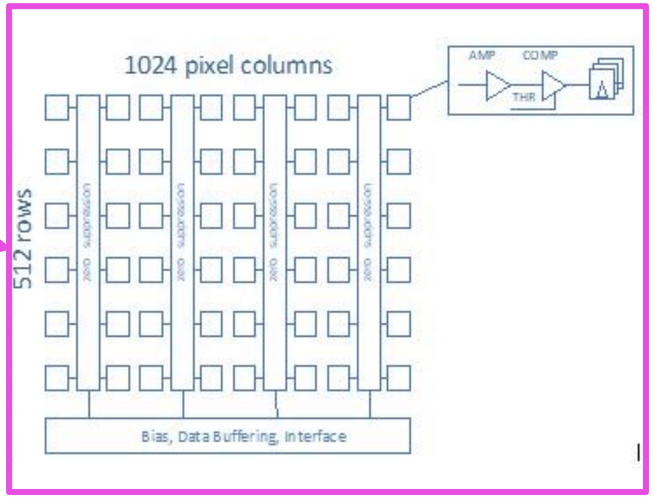
Content

1. Introduction
2. Motivation & Problem Statement
3. Research Objectives
4. Methodology
5. Results
6. Discussion & Future Work

The ALICE Experiment and ITS2



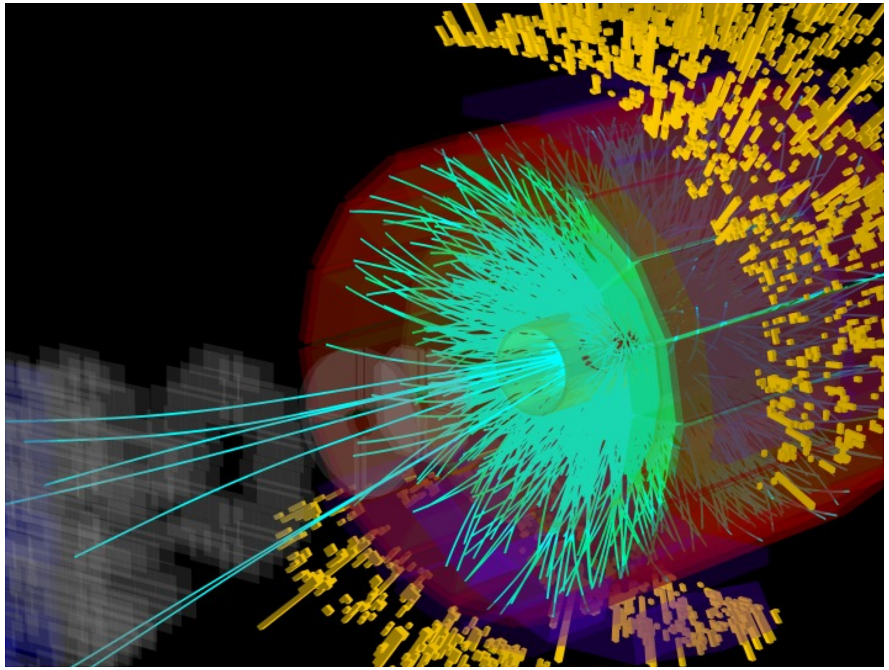


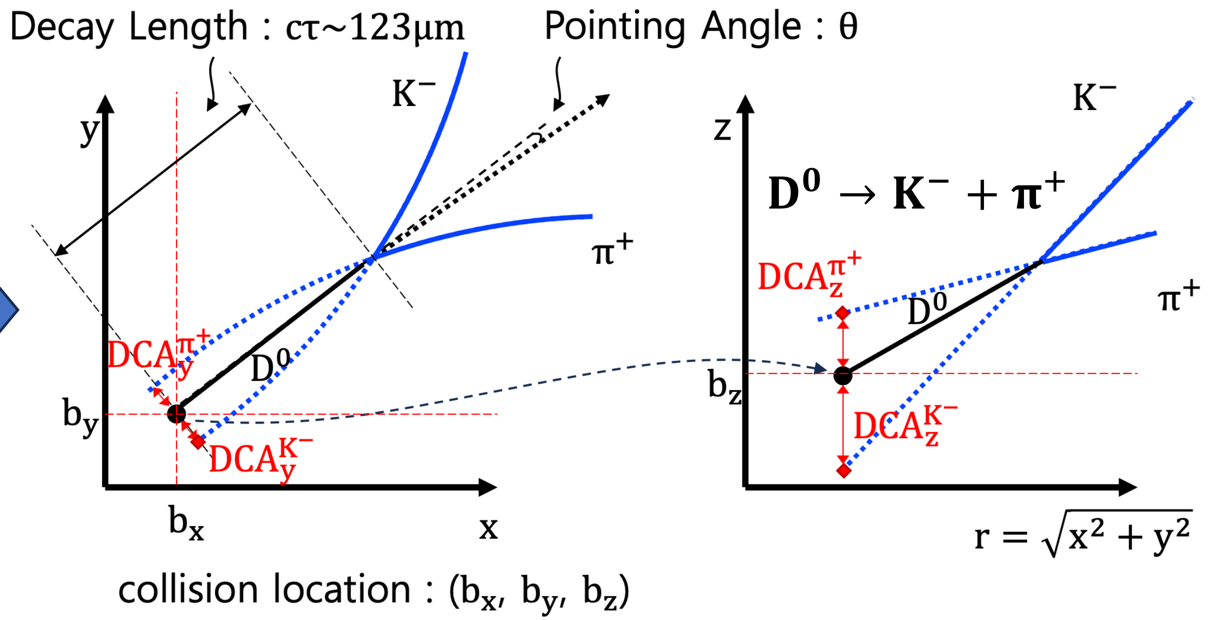


The new ITS is an all-pixel silicon detector based on CMOS monolithic active pixel sensor (MAPS).
https://ep-news.web.cern.ch/content/alice-its-upgrade-pixels-quarks

- **ALICE (A Large Ion Collider Experiment)**: a dedicated heavy-ion experiment at the CERN LHC, designed to study the Quark-Gluon Plasma (QGP).
- **The Inner Tracking System 2 (ITS2)**: upgraded in LHC Run 3, the largest-scale application of MAPS technology in a HEP experiment.
- **7 layers (3 Inner Barrel + 4 Outer Barrel), 24,120 ALPIDE sensors, 12.5 billion pixels.**
- Enables high-precision vertexing and tracking, critical for heavy-flavour hadron measurements near **the interaction point**.

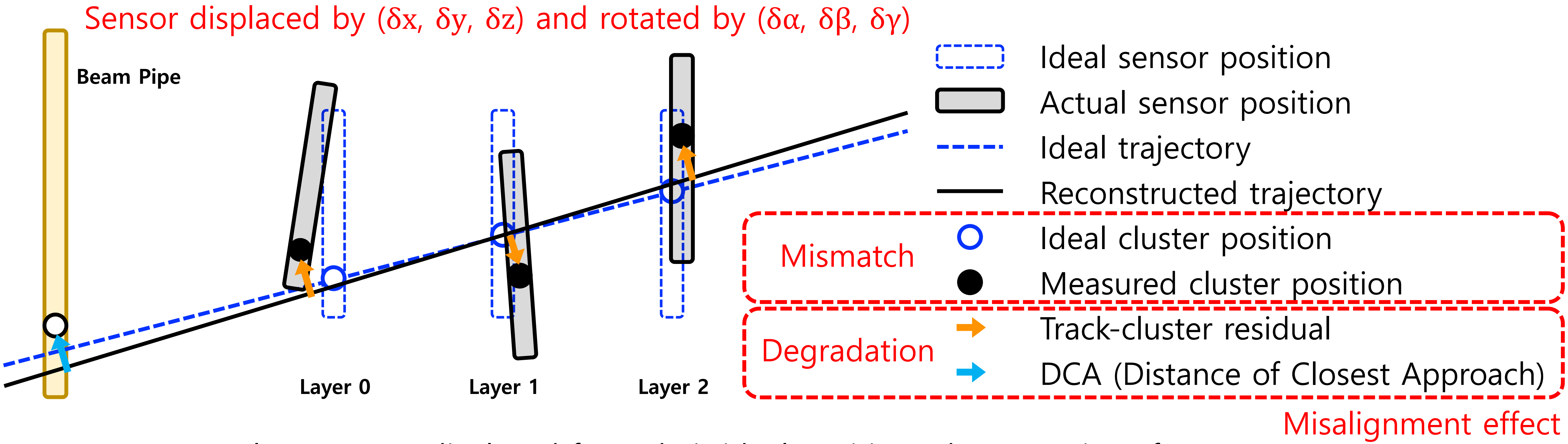
Why Precision Matters: Heavy-flavour Reconstruction





- DCA (Distance of Closest Approach): the closest distance between the extrapolated track trajectory and the primary vertex position.
- D^0 meson decays within ~123μm from the collision point .
- Reconstruction requires separating the secondary decay vertex from the primary collision vertex.
- Precise DCA measurement is essential to distinguish signal from combinatorial background.
- Any misalignment of the inner tracking layers directly degrades DCA resolution, reducing the ability to identify short-lived particles.

Geometric Alignment in Precision Tracking



- Detector elements are displaced from their ideal positions due to various factors: assembly tolerances, thermal variations, magnetic field effects, gravity sagging, etc.
- The exact contribution of each factor is difficult to isolate, but the cumulative effect degrades tracking performance at the micrometer level.
- Misalignment directly impacts: track-cluster residuals, DCA resolution, track reconstruction, and vertex reconstruction.
- ITS2 scale: 24,120 sensors × 6 DoF = ~145,000 alignment parameters to be determined.

Motivation: Beyond Millepede (golden standard)

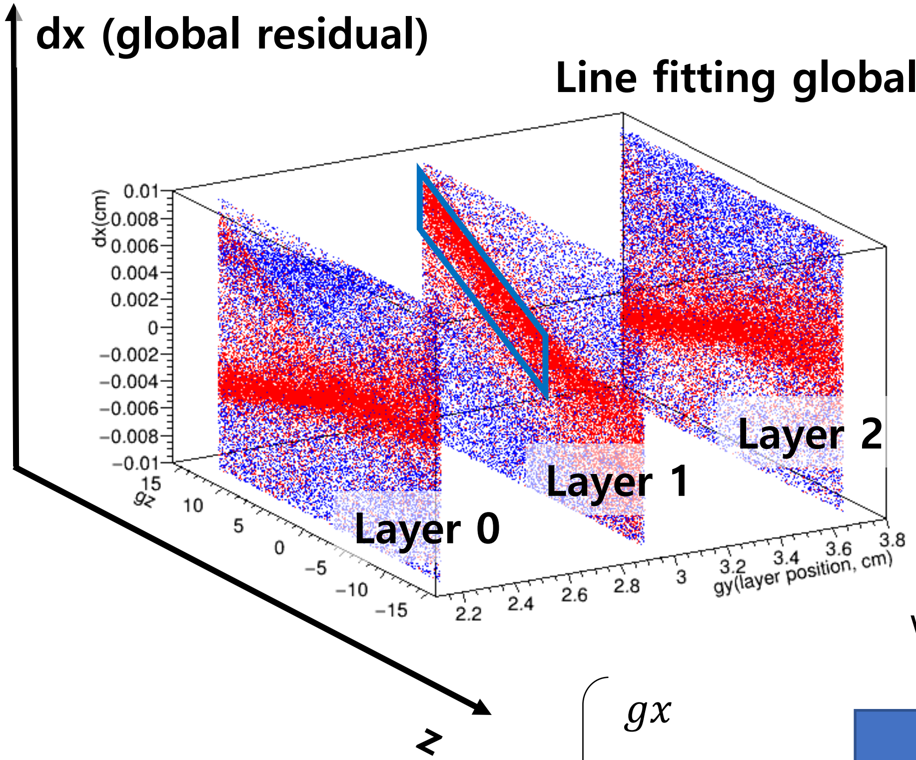
Experiment	Run Period	Subsystem	Pixel Size (μm×μm)	ROC Count	Total Module Count	Alignment Approach
ALICE	Run 2	SPD	50 ×425	1,200 (9.8M pixels)	240	**Millepede**
		SDD	202 ×294	- (133k anodes)	260	
		SSD	95 ×40,000	- (2.6M strips)	1,698	
	Run 3	**ITS2**	**26.88 ×29.24**	**24,120 (12.5G pixels)**	**1,740 (HIC)**	
ATLAS	Run 2-3	Pixel	50 ×400	27,904 (~70M pixels)	1,744	χ^2 minimization
		IBL	50 ×250	448 (~12M pixels)	280	
CMS	Run 2	Phase-0	100 ×150	15,840 (66M pixels)	~1,440	**Millepede**
	Run 2-3	Phase-1	100 ×150	29,696 (124M pixels)	1,856	
LHCb	Run 2	VELO	– (Strip)	- (Analog)	42	**Millepede**
	Run 3	VELO	55 ×55	624	52	

- **Massive chip counts** have led to a **dimensionality explosion** ($O(10^5 – 10^6)$ DoF), reaching the limits of computational costs, particularly for matrix inversion.

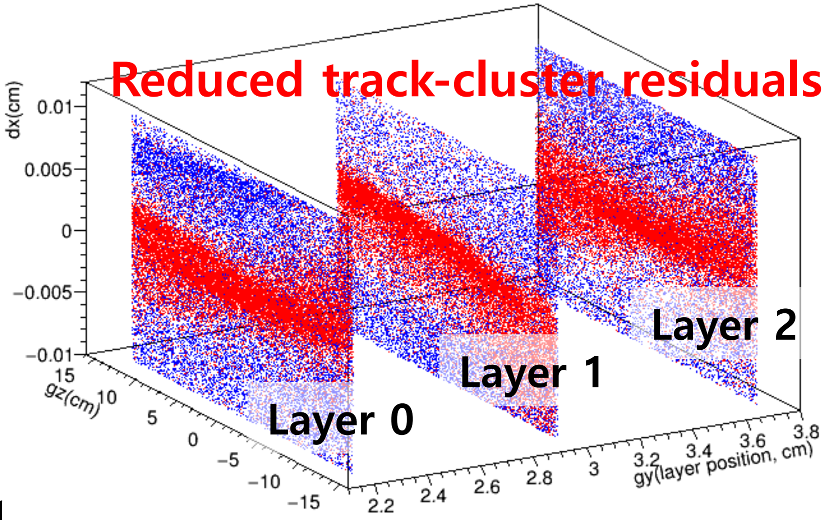
- Alignment performance is heavily dependent on manual geometry parametrization by experts, limiting the system's **flexibility and scalability**.

Correction Function

Line fitting global offset in x direction(dx) plot, cosmic muon



Apply Manual Correction



$$\mathrm{F_C} : gx \rightarrow gx' = gx + C_x,$$

$$\text{where } C_x = \left(\frac{0.01}{15}\right) \cdot gz + 0.001$$

Global hit position $\left\{ gx,\ gy,\ gz \right.$ → **Correction Function** $\mathrm{F_C}$ →

$$gx' = gx + C_x$$

$$gy' = gy + C_y$$

$$gz' = gz + C_z$$

- Designing $\mathrm{F_C}$ manually becomes increasingly challenging as detector complexity grows.
- A **data-driven approach** can advance the existing method for large-scale alignment.

Key Questions

1. Can a **Neural Network (NN)** learn geometric alignment corrections directly from data without manual parametrization?

→ **Results: Base Methodology**

2. Does the **ML-based alignment** improve tracking and vertexing performance beyond the conventional method?

→ **Results: Official Adoption in ALICE**

Artificial Intelligence model: Neural Network

A Neural Network can approximate any function!

Approximation by Superpositions of a Sigmoidal Function*

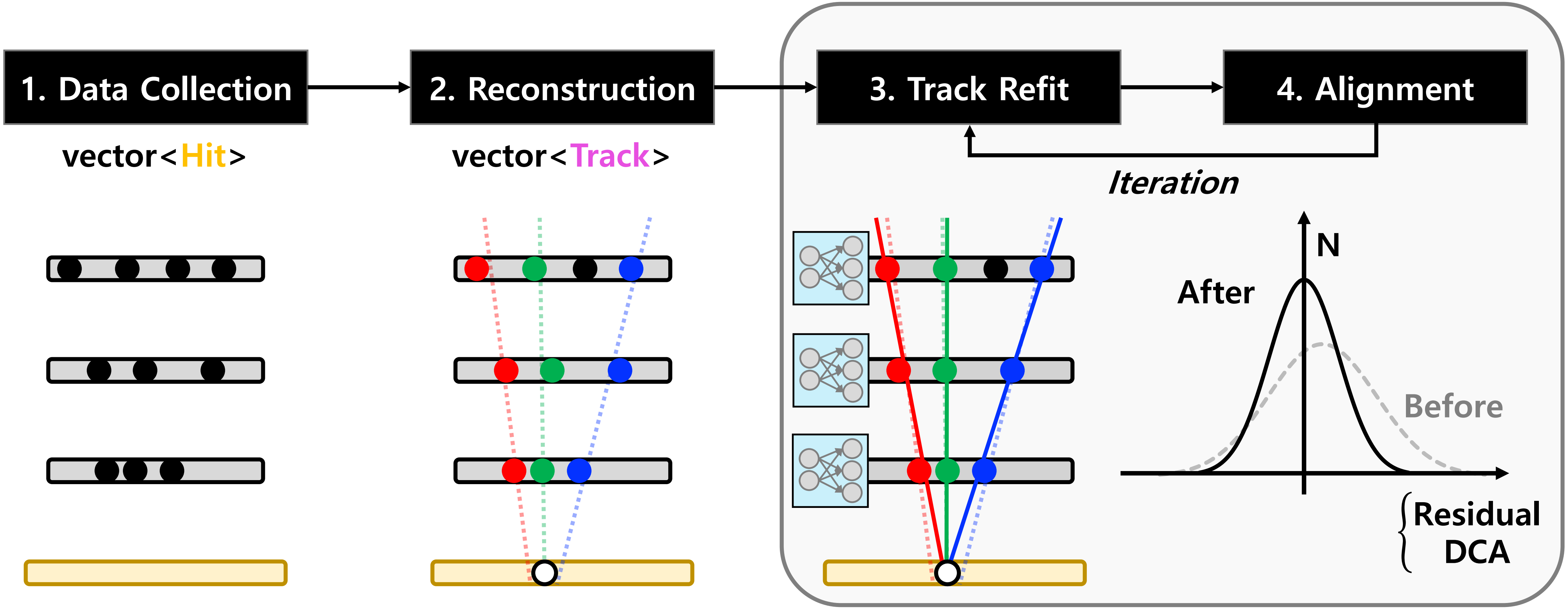
G. Cybenko†

Abstract. In this paper we demonstrate that finite linear combinations of compositions of a fixed, univariate function and a set of affine functionals can uniformly approximate any continuous function of *n* real variables with support in the unit hypercube; only mild conditions are imposed on the univariate function. Our results settle an open question about representability in the class of single hidden layer neural networks. In particular, we show that arbitrary decision regions can be arbitrarily well approximated by continuous feedforward neural networks with only a single internal, hidden layer and any continuous sigmoidal nonlinearity. The paper discusses approximation properties of other possible types of nonlinearities that might be implemented by artificial neural networks.

Key words. Neural networks, Approximation, Completeness.

Cybenko, G. (1989) "Approximation by superpositions of a sigmoidal function", *Mathematics of Control, Signals, and Systems*, 2(4), 303–314. doi:10.1007/BF02551274

Workflow for AI-ML based Alignment



2D cluster position	3D track trajectory	3D alignment correction
$(\mathrm{row}, \mathrm{col})$	$\left(X; Y, Z, \sin\phi, \tan\lambda, \frac{q}{p_T}\right)$	$(\mathrm{row}, \mathrm{col}) \rightarrow (s_1, s_2) \rightarrow$ [NN] $\rightarrow (cs_1, cs_2, cs_3)$
by sensor	by particle	by sensor

Formalism: Network Parameters to Alignment Parameters

1. Network Parameters

- Key objects: $(w_{11}, w_{21}, w_{21}, w_{22}, w_{13}, w_{23}, b_1, b_2, b_3)$, double;

$$\begin{pmatrix} g_x \\ g_y \\ g_z \end{pmatrix} = R_{GEOM} PO \begin{pmatrix} \mathbf{w_{11} + d_1} & \mathbf{w_{21}} & \mathbf{0} \\ \mathbf{w_{12}} & \mathbf{w_{22} + d_2} & \mathbf{0} \\ \mathbf{w_{13}} & \mathbf{w_{23}} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \widetilde{\mathbf{s_1}} \\ \widetilde{\mathbf{s_2}} \\ \mathbf{0} \end{pmatrix} - R_{GEOM} POD^{-1} \Delta + R_{GEOM} PO \begin{pmatrix} \mathbf{b_1} \\ \mathbf{b_2} \\ \mathbf{b_3} \end{pmatrix} + T_{GEOM}$$

normalized local sensor position

2. Alignment Parameters (ALICE)

- Key objects: $(\alpha, \beta, \gamma, \tau_1, \tau_2, \tau_3)$, std::vector<std::array<double, 6>>;

$$\begin{pmatrix} g_x \\ g_y \\ g_z \end{pmatrix} = \mathbf{a_R} \left(R_{GEOM} \begin{pmatrix} \mathbf{l_x} \\ \mathbf{l_y} \\ \mathbf{l_z} \end{pmatrix} + T_{GEOM} \right) + \mathbf{a_T} = \mathbf{a_R} R_{GEOM} POD^{-1} \begin{pmatrix} \widetilde{\mathbf{s_1}} \\ \widetilde{\mathbf{s_2}} \\ \mathbf{0} \end{pmatrix} - \mathbf{a_R} R_{GEOM} POD^{-1} \Delta + \mathbf{a_T} + \mathbf{a_R} T_{GEOM}$$

local sensor position

- $R_{GEOM} \in \mathbb{R}^{3\times3}$, $T_{GEOM} \in \mathbb{R}^3$: Baseline Geometry
- $\mathbf{a_R} \in \mathbb{R}^{3\times3}$, $\mathbf{a_T} \in \mathbb{R}^3$: alignment parameter
- PO : coordinate determination matrix, e.g. $(l_x, 0, l_z) \rightarrow (\tilde{s_1}, \tilde{s_2}, 0)$
- D, Δ : constant matrix related to sensor length

Teaching the AI What "Better Alignment" Means

$$C(\Theta) = \frac{1}{N_{event}} \sum_{i}^{N_{event}} \frac{\chi^2}{\nu_i},$$

$$\chi^2 = \sum_{j}^{N_{track}} \left(\chi^2_{track} + \chi^2_{vertex}\right),$$

$$\chi^2_{track} = \sum_{k}^{N_{layer}} \left(\chi^2_{meas,xy} + \chi^2_{meas,z}\right)_k,$$

$$\left(\chi^2_{meas,xy}\right)_k = \left\{\frac{1}{\sigma^2_{meas,xy}}\left(\left(r_{xy} + \Delta r_{xy}\right) - \bar{r}_{xy}\right)^2\right\}_k,$$

$$\left(\chi^2_{meas,z}\right)_k = \left\{\frac{1}{\sigma^2_{meas,z}}\left(\left(r_z + \Delta r_z\right) - \bar{r}_z\right)^2\right\}_k,$$

$$\chi^2_{vertex} = \chi^2_{vertex,xy} + \chi^2_{vertex,z},$$

$$\chi^2_{vertex,xy} = \left\{\frac{1}{\sigma^2_{vertex,xy}}\left(r_{origin,xy} - \bar{r}_{vertex,xy}\right)^2\right\}_k,$$

$$\chi^2_{vertex,z} = \left\{\frac{1}{\sigma^2_{vertex,z}}\left(r_{origin,z} - \bar{r}_{vertex,z}\right)^2\right\}_k,$$

$$\sigma^2 = \sigma^2_{IPR} + \Sigma_q \sigma^2_{SPP,q}$$

AI-ML alignment = minimization of $C(\Theta)$

The optimization step at iteration t :

$$\Theta^{(t+1)} = \Theta^{(t)} - \eta \cdot \left(\nabla_\Phi C(\Theta) \cdot \frac{\partial \Phi}{\partial \Theta}\right),$$

where η learning rate.

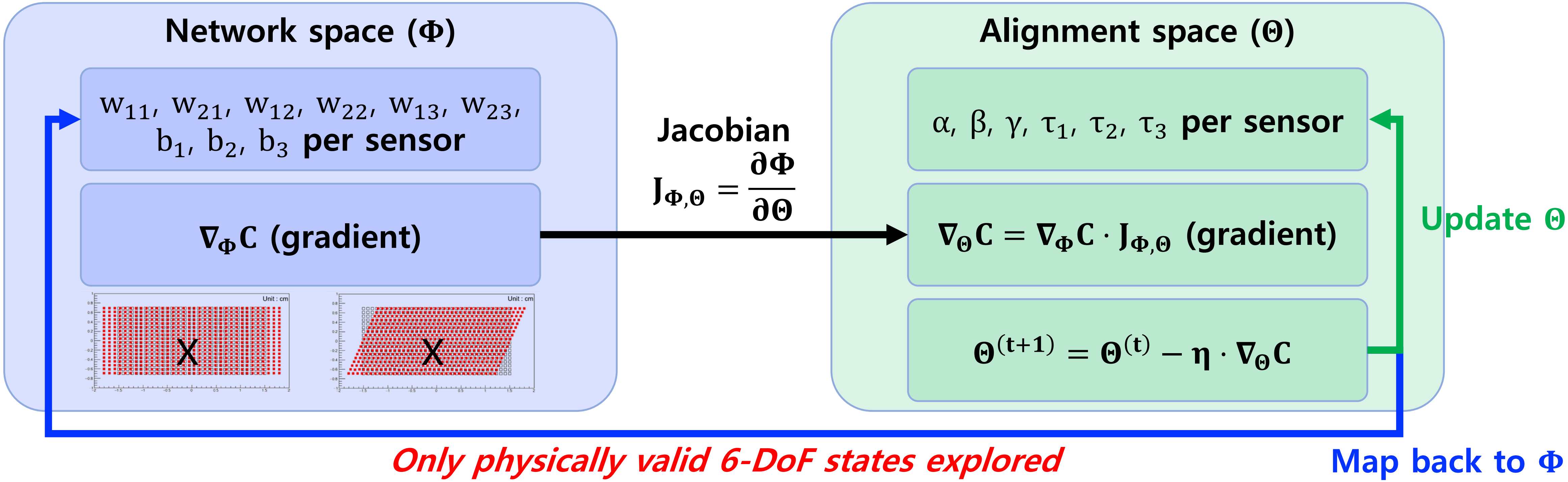
$$\Phi = \{(w_{11}, w_{21}, w_{12}, w_{22}, w_{13}, w_{23}, b_1, b_2, b_3)_s\}_{s=1}^{N_{sensor}}$$

$$\Theta = \{(\alpha, \beta, \gamma, \tau_1, \tau_2, \tau_3)_s\}_{s=1}^{N_{sensor}}$$

- $r + \Delta r$: corrected position with alignment parameters
- $\bar{r}$: projected trajectory position on the sensor plane
- r_{origin} : common event origin (primary vertex)
- $\bar{r}_{vertex}$: point on the fitted trajectory closest to r_{origin}
- σ_{IPR} : intrinsic pixel resolution
- σ_{SPP} : stochastic physical process (e.g. multiple scattering, energy-loss fluctuations

Real-world sensor behavior encoded in σ,
only physically valid alignments ***are permitted***.

Update Procedure for 6-DoF Alignment



- **Gradient Computation in Φ-space**: compute $\nabla_\Phi C$ using track-cluster residuals in the χ^2 cost.
- **Transformation via Jacobian ($J_{\Phi,\Theta}$)**: transform gradients to 6-DoF alignment parameter space Θ.
- **Update in Θ-space**: update ($\alpha, \beta, \gamma, \tau_1, \tau_2, \tau_3$) with Euler angles following the ALICE convention, orthogonality of the rotation matrix guaranteed by construction.
- **Mapping back to Φ**: updated **Θ** mapped back to network parameters **Φ**.

Implementation and Dataset

- Implementation details

Parameter	Value / Specification
NN Architecture	Single Layer Perceptron, (Input: 2, Output: 3)
Activation Function	Linear
Optimizer	Stochastic Gradient Descent (SGD)
Framework	ROOT (CERN Analysis Framework), ALICE O2 (Experiment Framework)
Hardware (CPU)	Intel i7-14700K
Memory Usage	2GB (Train) / 6GB (Monitor)
Training Duration	~2 Weeks (45 Epochs, $\eta \in 10^{[-12,-9]}$)
Parallelization	8 Modules ($\mu_m = 1.0, \tau = 5$)

- Scalable Training via Parallelization

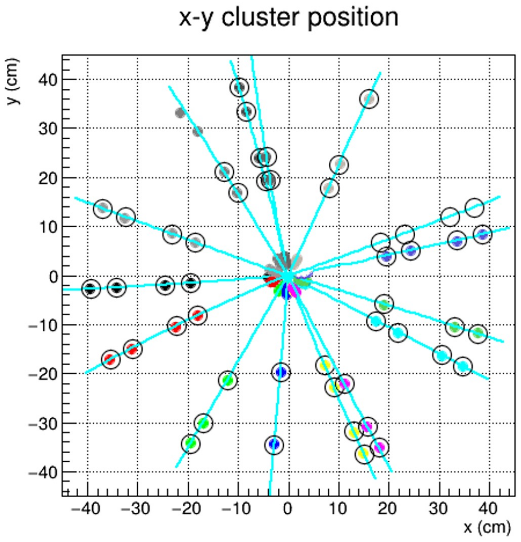
$$\boldsymbol{\Theta}^{(\tau+1)} = \frac{\Sigma_{\mathrm{m}} \mu_m^{(\tau)} \boldsymbol{\Theta}_m^{(\tau)}}{\Sigma_{\mathrm{m}} \mu_m^{(\tau)}}$$

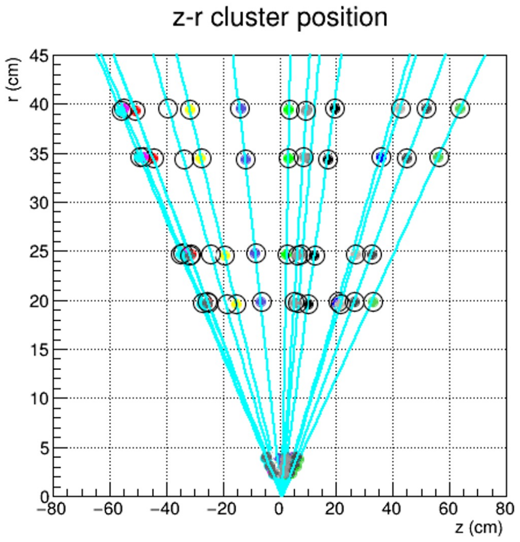
$\mathbf{m}$ the parallel module index
$\boldsymbol{\tau}$ the synchronization step
$\boldsymbol{\mu}_{\mathbf{m}}$ the weight of module $\mathbf{m}$ determined by the data partition size

- Dataset (LHC23zt/539884)

Data set	# of Events	Trajectories per Event
Train	5.0 M	8
Validation	1.5 M	8
Test	1.5 M	8

- Example of Dataset (reconstructed trajectories)



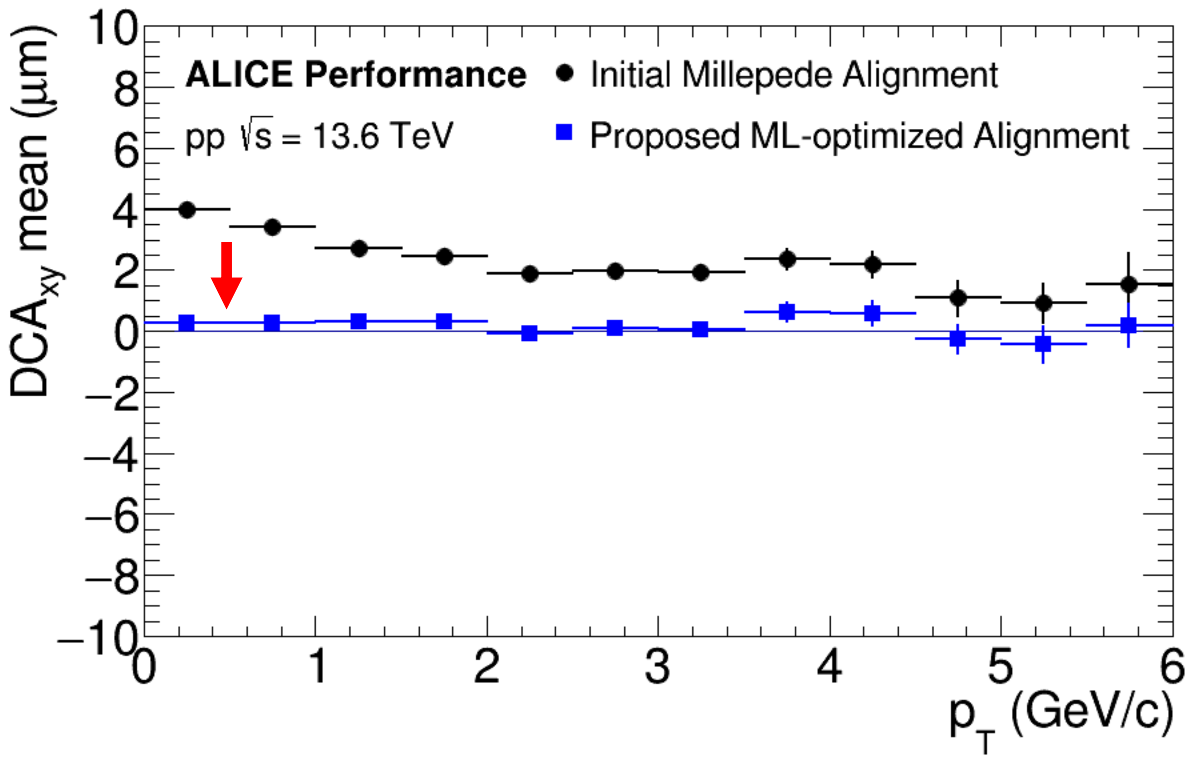


- Charged particles under active magnetic field

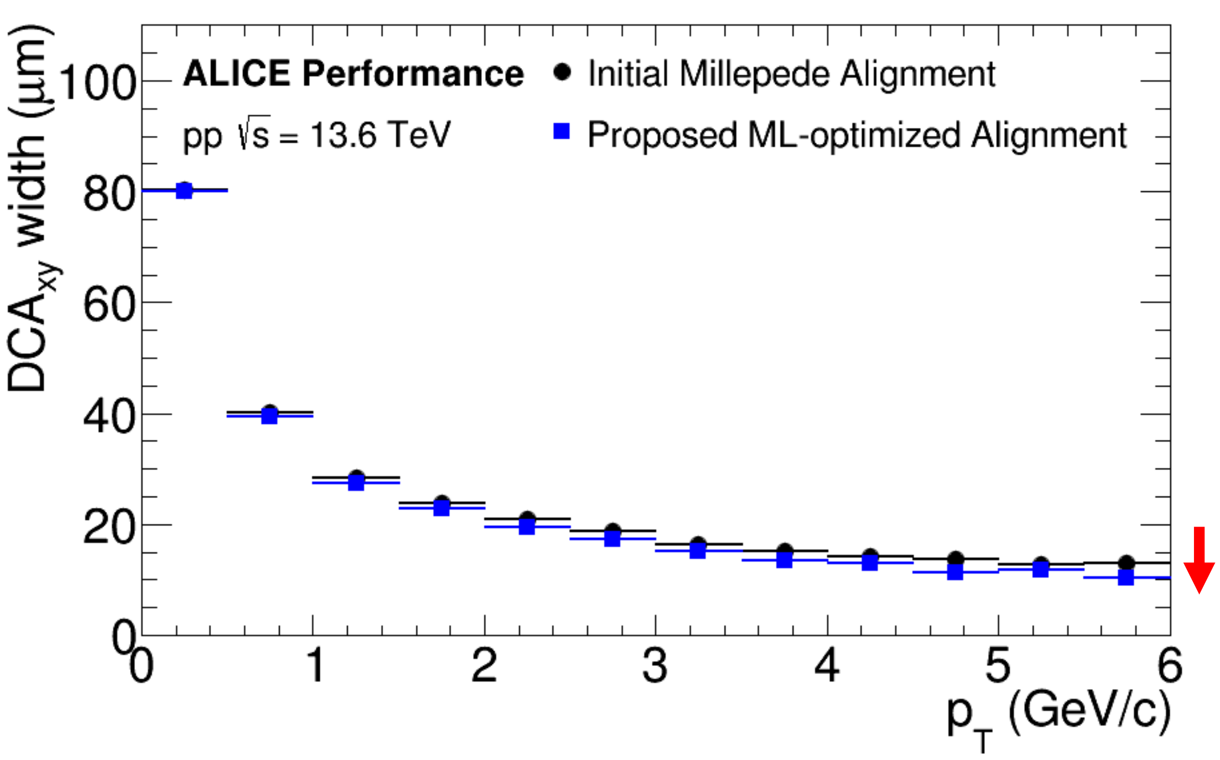
$$p_{\mathrm{T}}[\mathrm{GeV}/c] = 0.3B[\mathrm{T}]R[\mathrm{m}]$$

ML-Base Alignment on LHC23 Data

$\mathbf{DCA_{xy}}$ mean (bias)



$\mathbf{DCA_{xy}}$ width (resolution)



- ML-Base alignment **eliminates the systematic DCA bias (~2-4 μm → ~0 μm),** and **improves resolution by ~5-10% across all p_T (transverse momentum).**

Official Adoption in ALICE: LHC23 Pb-Pb apass4

https://its.cern.ch/jira/browse/O2-5169

O2 General / O2-5169

Upload of geometry with ITS ML alignment for PbPb23 (apass4)

Edit | Add comment | Assign | More | Closed

Details

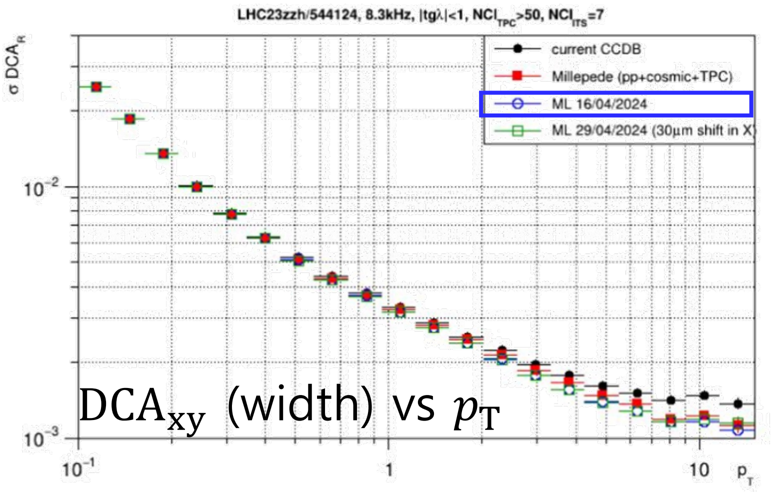
Type:	CCDB upload	Resolution:	Fixed
Priority:	Blocker	Fix Version/s:	None
Affects Version/s:	None		
Component/s:	None		
Labels:	None		
CCDB:	alice-ccdb.cern.ch		
Calibration type:	GLO/Config/GeometryAligned, ITS/Config/Geometry		

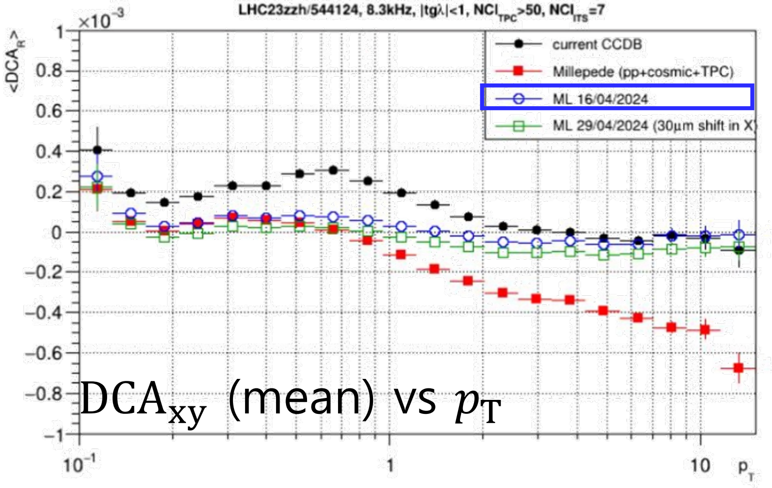
Description

The version sent by Jaehyun Kim on 16/04/24 will be used as the with global X shift of 30\mu (29/04/24) does not show any advantage but move the meanvertex, see attached `alg_comp080724.pdf`.
Upload is done for pbpb23 and preceding pp ref (LHC23zw) timespan only:
1695167249000 : 1699807883000

```
o2-ccdb-upload --host http://alice-ccdb.cern.ch -p ITS/Calib/Align -k ccdb_object -m "JIRA=[~~O2-5169~~](/jira/browse/O2-5169);comment=for runs 543107 - 545367 i.e. ref.pp and pbpb23"   -f ITSAlignment_20240416.root --starttimestamp 1695167249000 --endtimestamp 1699807883000

o2-create-aligned-geometry-workflow --configKeyValues HBFUtils.startTime=1695167449000
```

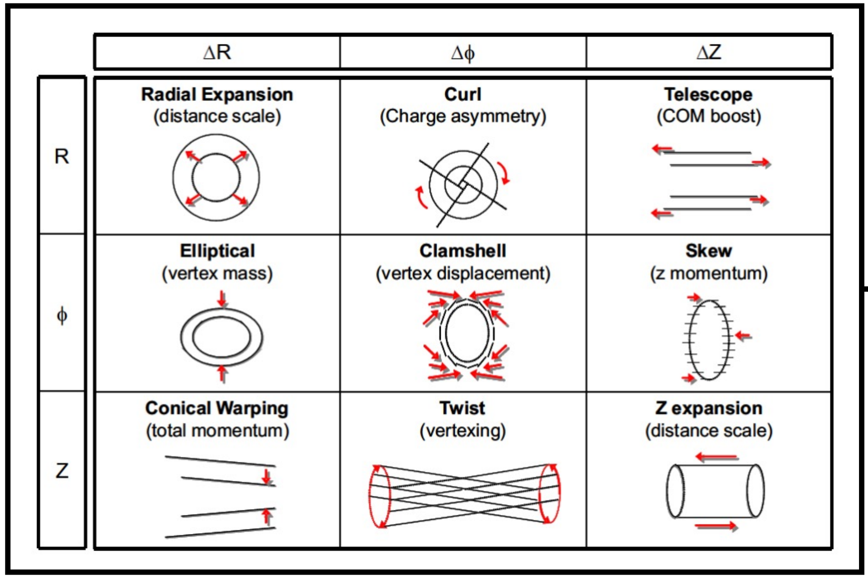




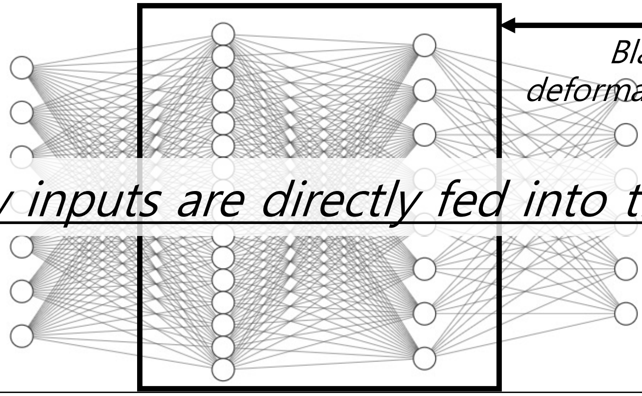
- Proposed ML-optimized alignment constants **officially uploaded to ALICE CCDB,** and **adopted in the LHC23 Pb-Pb apass4 reconstruction (JIRA O2-5169).**

Pure Data-Driven vs Physics-Informed Approach

Pure Data-Driven Approach



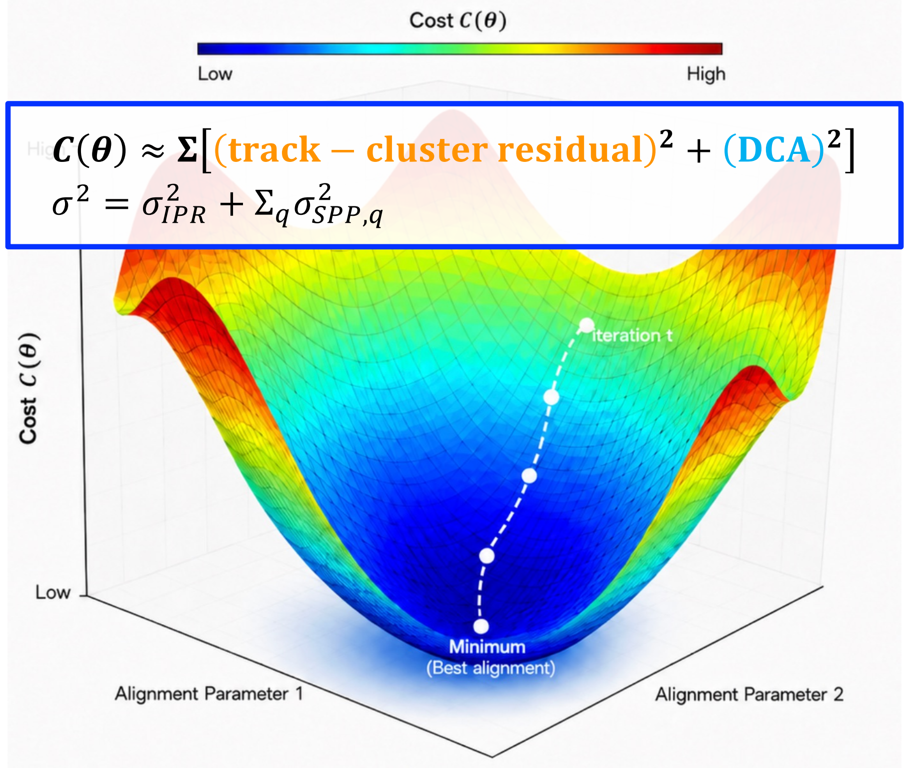
3D correction or 6-DoF parameter



Black-box deformation learning

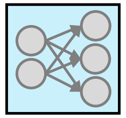
Massive raw inputs are directly fed into the network.

Physics-Informed Approach



$$C(\theta) \approx \Sigma[(\text{track} - \text{cluster residual})^2 + (\text{DCA})^2]$$
$$\sigma^2 = \sigma_{IPR}^2 + \Sigma_q \sigma_{SPP,q}^2$$

3D alignment correction by sensor

$(\mathbf{row}, \mathbf{col}) \rightarrow (\mathbf{s_1}, \mathbf{s_2}) \rightarrow$  $\rightarrow (\mathbf{cs_1}, \mathbf{cs_2}, \mathbf{cs_3})$

Explainable AI

$(\alpha, \beta, \gamma, \tau_1, \tau_2, \tau_3)$

Key Advantages and Outlook

Key Advantages

- A self-supervised learning framework for the geometric alignment of large-scale systems, requiring no labeled data while demonstrating intrinsic robustness to physics-induced noise.
- A differentiable physics-informed χ^2 cost formulation that enforces geometric correctness and enables end-to-end optimization grounded in physical principles.
- A scalable architecture capable of optimizing 24,120 sensor-wise regression models in parallel, demonstrating efficient gradient-based learning at realistic operational scales.
- A quantitative demonstration of improved geometric accuracy, including the removal of systematic DCA biases and ~5-10% enhanced resolution.

Outlook: Potential Extensions

- Application to the ALICE ITS3 upgrade (wafer-scale bent MAPS, LS3)
- Exploration of online alignment for real-time calibration
- Generalization to other large-scale tracking detectors (CMS, ATLAS, LHCb)