CEPC 2026

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

CEPC TPC Track Finding: A GNN-DBSCAN Pipeline and Autonomous ML Agent

Xiaoqian Jia, Ziyue Chen, Ke Li, Chengdong Fu

Institute of High Energy Physics

The 2026 International Workshop on the High-Energy Circular Electron-Positron Collider

Oct. 24, 2026 · Shanghai

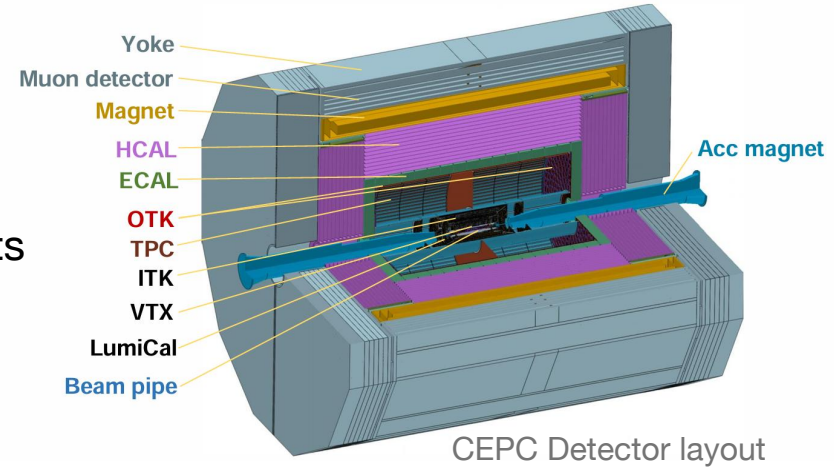
Outline

- CEPC TPC tracking
- Density-based clustering for CEPC TPC tracking
- AIDE: AI-Driven Exploration in the Space of Code
- Knowledge Base of HEP Machine Learning for Agent
- Summery and Outlook

CEPC Time Projection Chamber

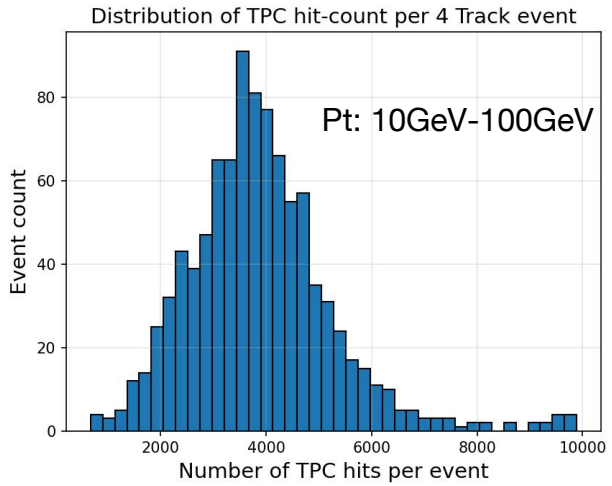
The detector component of the charged particle tracking system for the CEPC

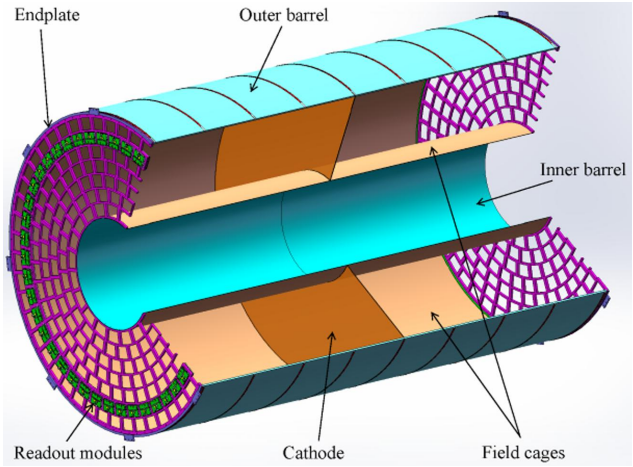
- **A pixelated Time Projection Chamber (TPC)**
  - The main gaseous tracker
  - Positioned between the ITK and OTK
  - Provide 3D space points and PID information through ionization measurements



CEPC Detector layout

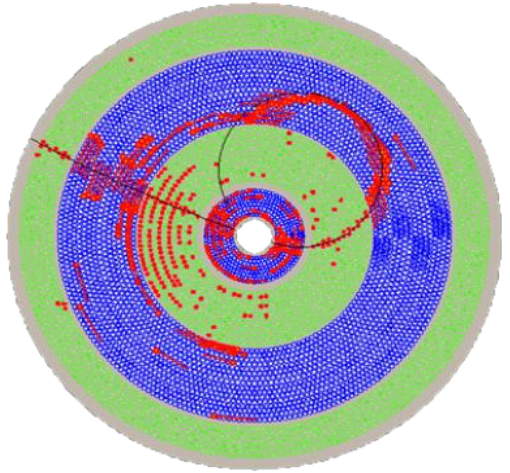
- **TPC track-finding challenges**
  - A single piece geometry with full-volume pixel readout
  - 10^7 readout channels
  - 10^3 hits per nearly straight track
  - Multi-turn tracks below 1 GeV



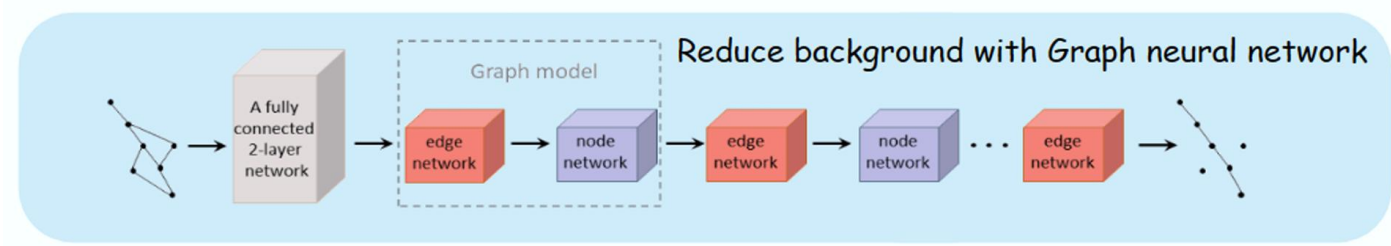


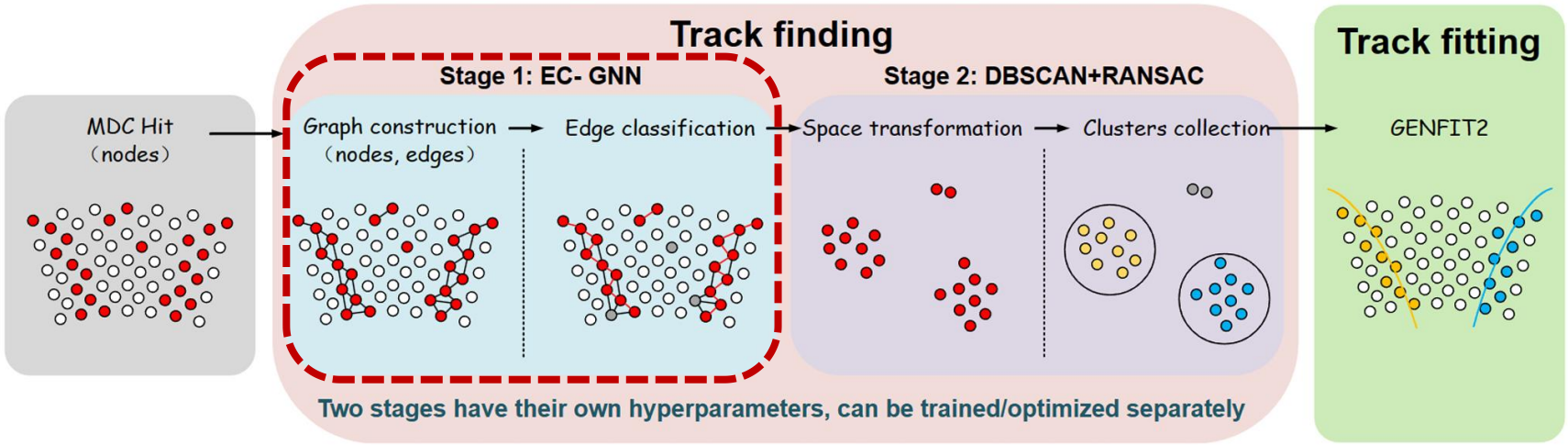
Hybrid Cascaded Track Finding for BESIII MDC

GNN noise filter + DBSCAN clustering + RANSAC fitting



- **Edge-classifying Graph Neural Network**
  - Distinguishes signal hits from noise hits before track finding
  - Reduce the computational load of downstream tracking algorithms
  - Performed well on the MDCs of STCF and BESIII

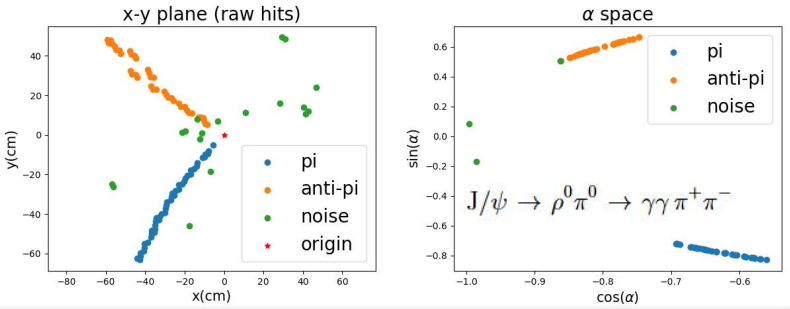




Hybrid Cascaded Track Finding for BESIII MDC

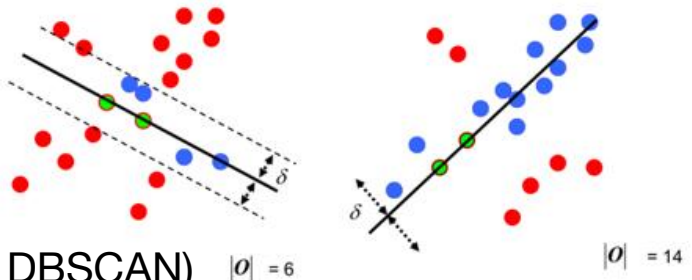
Density-based clustering (DBSCAN)

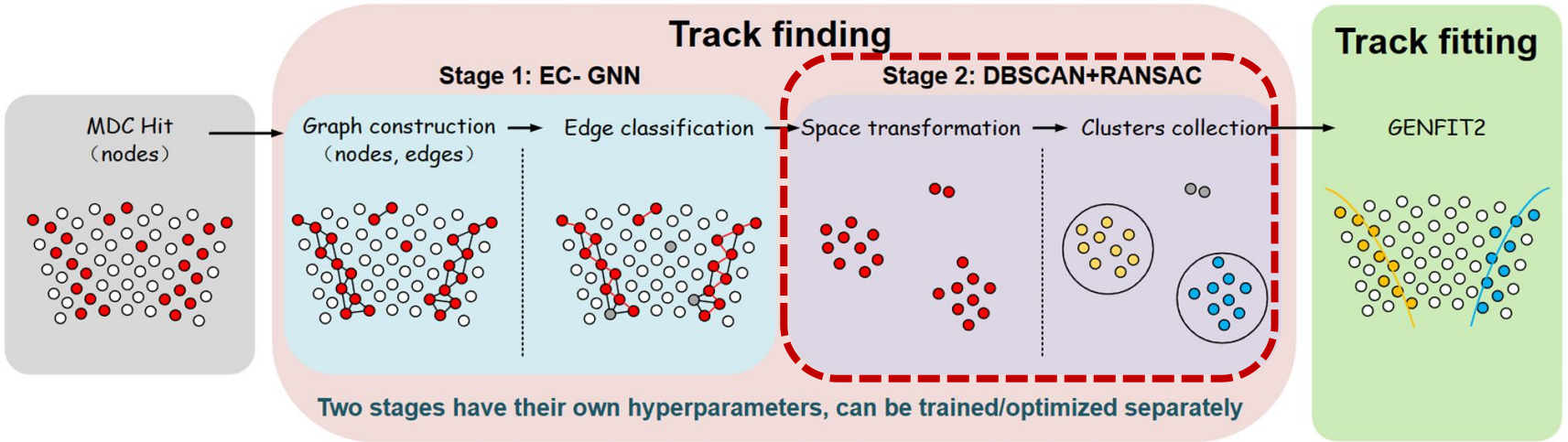
- Transform MDC 2D physical space into a custom parameter space, making tracks cluster more closely together
- Adaptively select parameters for each event



Random Sample Consensus（RANSAC)

- Fitting a regression model from noisy data, the inliers selected at one time are the hits of the same track
- Dealing with nearby or crossing tracks (densities connected, hard to separate with DBSCAN)



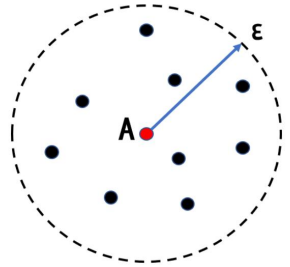


Clustering for CEPC TPC

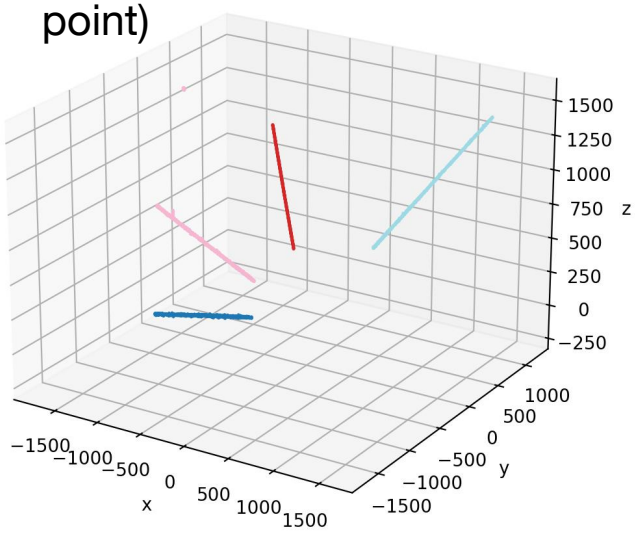
- **Custom 3D parameter space:**
  - sin (α), cos (α)

    α : the angle between the line through the data point in the conformal plane and the origin, and the positive X-axis
  - z/r

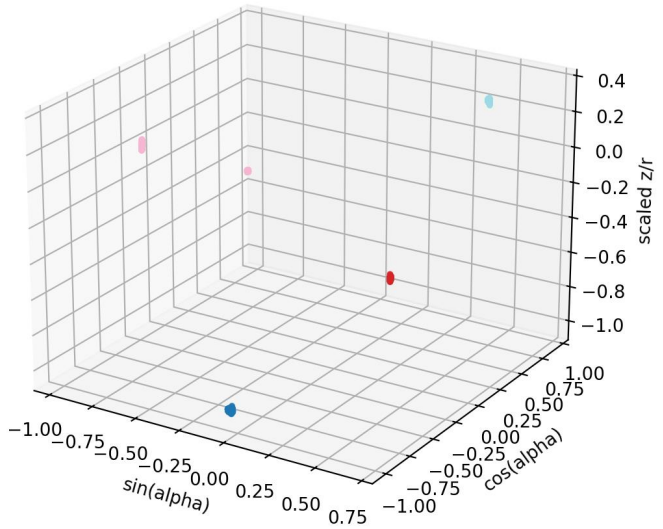
    r : $\sqrt{x^2 + y^2}$



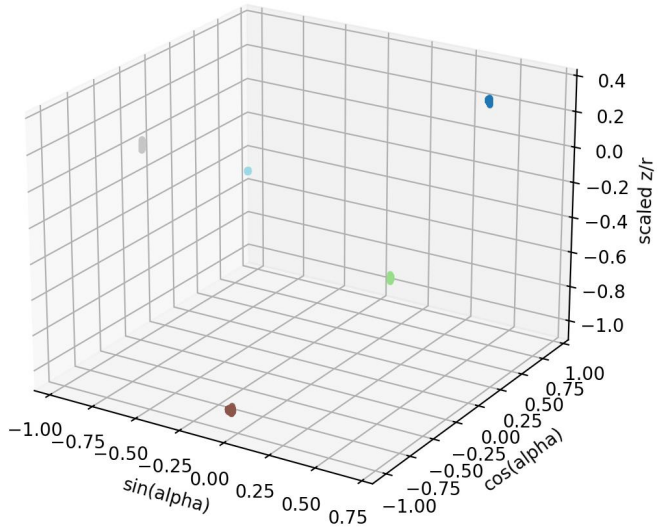
- **Parameter optimization：Grid search**
  - Epsilon (radius of the circle to be created around each data point)
  - MinPoints (the minimum number of data points required inside that circle for that data point to be classified as a Core point)



Original space



Parameter space

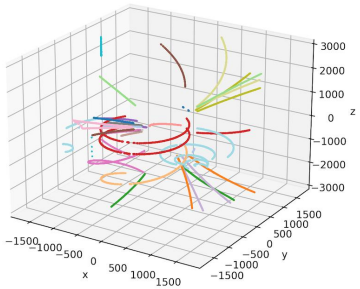


Custering result, n_cluster=5

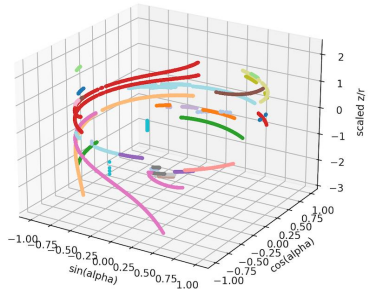
Clustering performance

Data sets

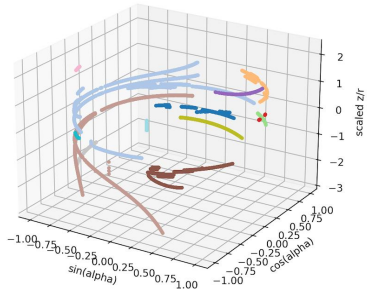
- 4 track: $\pi^+\pi^-\mu^+\mu^-$(Particle Gun), 10GeV~100GeV
- $e^+e^- \rightarrow ZH$, $Z \rightarrow q\bar{q}$, $H \rightarrow b\bar{b}$ (~ 49 Track)



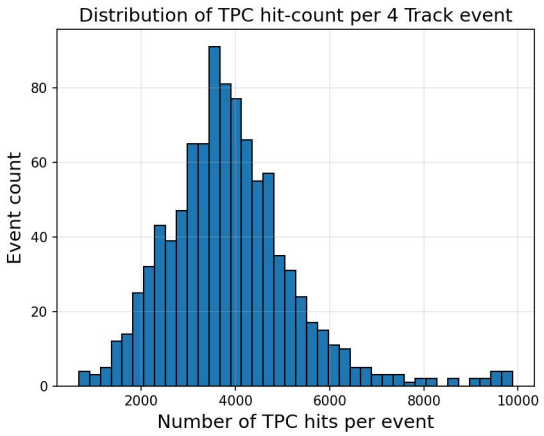
Original space

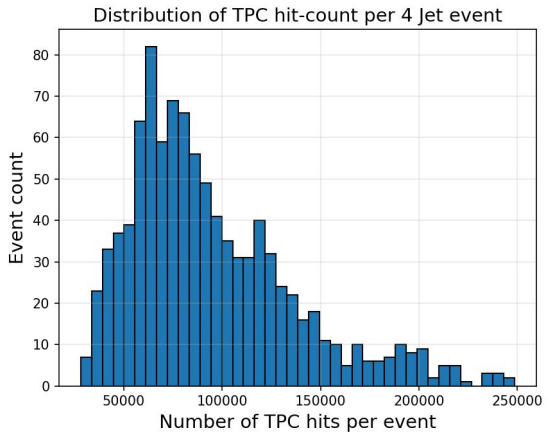


Parameter space



Custering result, n_cluster=13





Hit efficiency

$$\epsilon_{hits} = \frac{N_{hit}^{matched}}{N_{hit}^{MC\ truth}}$$

True hits of a particle that are included in the reconstructed cluster

Hit purity

$$p_{hits} = \frac{N_{hit}^{matched}}{N_{hit}^{rec}}$$

Hits inside the reconstructed cluster that belong to the same true particle

Track efficiency

$$\epsilon_{track} = \frac{N_{track}^{matched}}{N_{track}^{truth}}$$

Matched: $\boldsymbol{p_{hits} > 90\%,\ \epsilon_{hits} > 50\%}$

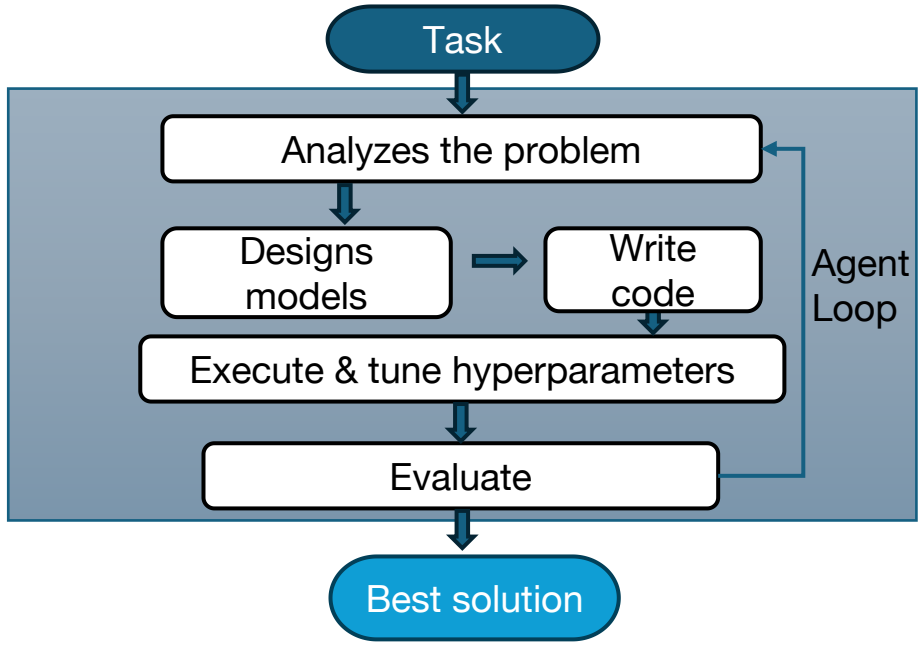
Baseline	Hit efficiency	Hit purity	Track efficiency
4 track	95.3%	98.5%	92.1%
4 jet	77.5%	75.8%	39.5%

LLM-Driven Autonomous Machine Learning

An AI agent that automatically builds ML models

- **Takes data + goal + evaluation metric as input**
- **Automates the design choices that traditionally require human ML experts:**
  - Data preprocessing and cleaning
  - Feature engineering and selection
  - Algorithm selection
  - Hyperparameter tuning
  - Pipeline construction and evaluation
- **Outputs a runnable, best-performing solution**

Agent	Role
Manager	Coordinate & judge trajectories
Designer	Search / Revise method plans
Coder	Implement designed pipeline
Tuner	Monitor long run& tune parameters
Aggregator	Ensemble & select best submission
…	…



AutoML : Not a single algorithm !
Acoordinated multi-agent system, powered by an LLM, that searches, builds, evaluates, and improves ML pipelines **end-to-end**

Why Auto ML ?

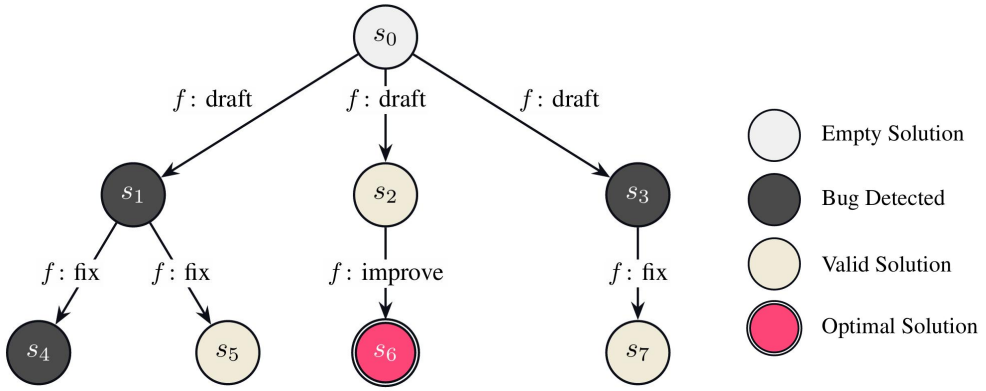
- Democratizes ML for non-experts across domains
- Reduces development time from weeks to days
- Enables reproducible, scalable ML pipelines for science

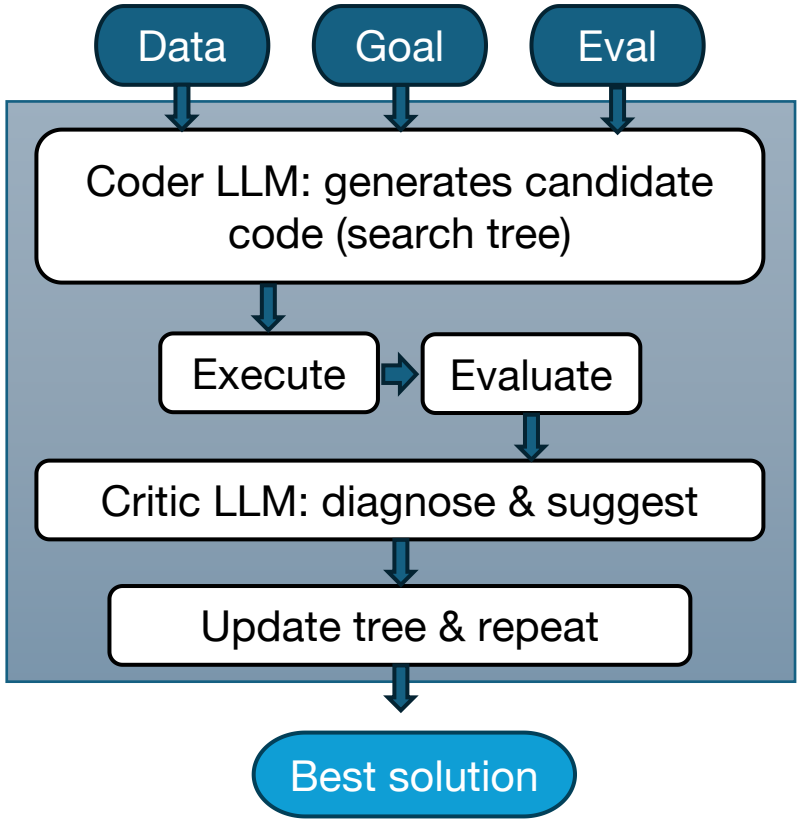
AIDE: Tree-Search-Driven Autonomous ML Agent

From “data + goal” to a runnable, optimized ML solution

- An LLM-based autonomous agent that automatically designs, writes, evaluates and iteratively improves machine learning code **end-to-end**
  - Formulates every solution as **a tree** of code nodes (not a prompt or weight update)
  - Each node samples a candidate; execution returns metric + error log
  - **A critic LLM** turns the result into a natural-language improvement instruction
  - **A coder LLM** implements that instruction as a new candidate code

- Key capabilities
  - **Search** - explores solutions by selecting and combining code components
  - **Evaluate** - scores every candidate against the task metric (e.g. AUROC)
  - **Iterate & debug** - traces failures and proposes fixes or new ideas





Reference: AIDE: AI-Driven Exploration in the Space of Code. github.com/WecoAI/aideml

AIDE configs for TPC tracking

- **Goal:**
  - Perform data points cluster on parquet data.
  - Each row is one point with columns event_id,cellID, x, y, z, time, edep, truthParticleId.
  - Group points by event_id and run clustering independently for each event. The number of clusters per event is unknown and variable.
  - Only cellID, x, y, z, time, edep can be used as model input feature. TruthParticleId is ground truth and may be used only for evaluation to show which points belong to one cluster.

A more general description can improve the AIDE's comprehension and overall performance

- **Eval:**
  - Clustering efficiency
  - Clustering purity
  - The proportion of events in the test set with a clustering purity greater than 0.9 and efficiency greater than 0.5 .

Parameter	Purpose
data_dir	Specifies the input data directory
goal / desc_file	Defines the task objective
eval	Defines the evaluation criteria
agent.steps	Number of iterative improvement steps
agent.search.num_drafts	Number of candidate solutions per step
agent.search.max_debug_depth	Maximum debugging depth for each solution
exec.timeout	Maximum runtime for each code execution, in seconds
preprocess_data	Whether to automatically preprocess or extract data
agent.code.model	Model used for code generation
agent.feedback.model	Model used for error analysis and feedback
report.model	Model used to generate the final report
generate_report	Whether to generate a final report
…	…

Tracking method provided by AIDE

- **Data set**
  - 4 track: $\pi^+\pi^-\mu^+\mu^-$(Particle Gun), 10GeV~100GeV
- **Best method**

Method or graph setting	Efficiency	Purity	Track efficiency
DBSCAN	0.3514	0.9987	30.8%
BIRCH	0.0182	0.9999	0.04%
Bayesian Gaussian mixture	0.8745	0.989	73.8%
Graph: 9 neighbors, radius multiplier 1.25	0.0827	0.9998	0.12%
Graph: 9 neighbors, multiplier 16	0.7973	0.9974	88.95%
Graph: 24 neighbors, multiplier 16	0.9515	0.9948	93.90%
Graph: 40 neighbors, multiplier 16	0.9568	0.9927	93.65%
Graph: 64 neighbors, multiplier 16	0.9592	0.9918	93.60%
Baseline	**0.953**	**0.985**	**92.1%**

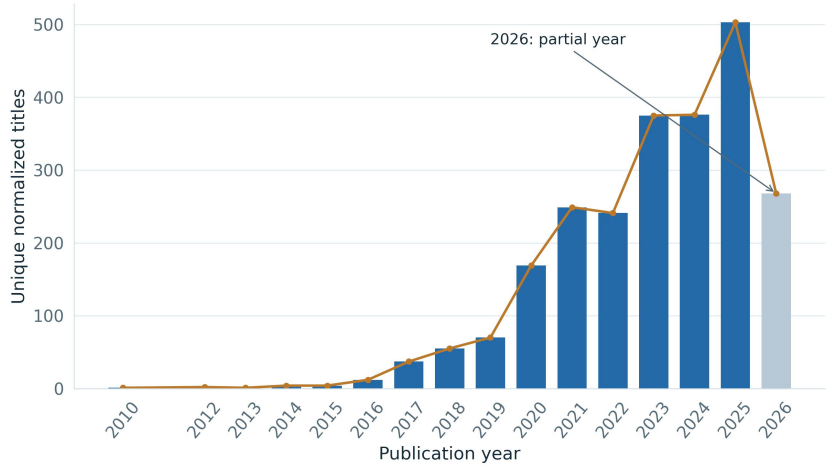
Tracking method provided by AIDE

- **Data set**
  - $e^+e^- \rightarrow ZH$, $Z \rightarrow q\bar{q}$, $H \rightarrow b\bar{b}$ (~ 49 Track)
- **Best method**

Method or graph setting	Efficiency	Purity	Track efficiency
Graph: 9 neighbors, radius multiplier 1.25	0.034	0.64	0.08%
Graph: 9 neighbors, multiplier 16	0.326	0.852	35.97%
Graph: 24 neighbors, multiplier 16	0.6843	0.8898	43.98%
Graph: 40 neighbors, multiplier 16	0.7455	0.7476	47.59%
Graph: 64 neighbors, multiplier 16	0.7723	0.6893	47.03%
Baseline	**0.775**	**0.758**	**39.5%**

Knowledge Base for Agent

- **ML4HEP literature is growing in volume, topical breadth, and overlap**
- **A dedicated HEP knowledge base:**
  - Provides a persistent and structured memory
  - Reduces hallucination, improves physics consistency
  - Retrieve relevant reconstruction experience before designing a new model
- **Build process**



From : iml-wg.github.io/HEPML-LivingReview/

arXiv ID
↓
Resolve paper identity
↓
Download abs page / HTML / PDF
↓
HTML–to–Markdown or PDF–to–Markdown
↓
Generate raw.md
↓
Split into evidence fragments
↓
Agent workers extract structured fields
↓
Generate Problem / Solution / Result cards
↓
Schema validation
↓
Evidence coverage check
↓
Quality review
↓
Human approval
↓
Promote to project knowledge index
↓
Retrieved by AutoML–Agent for future model design

Knowledge Base for Agent

Paper is organized into a one-paper-one-folder structure and represented by three main YAML cards:

- A human-readable wiki.md : support fast retrieval and inspection.
- Human review: If the extraction is incomplete or uncertain, marked as needs_review, and the user can approve or reject it

Problem

- physics problem
- detector environment
- dataset
- objective
- evaluation metrics...

Solution

- proposed method
- model architecture
- training configuration
- inference workflow
- code reference
- transferability...

Result

- quantitative results
- comparisons with baselines
- failure modes
- assumptions
- lessons learned
- provenance...

```yaml
extensions:
  proposed_benchmark: proposes τ+τ- as the new benchmark system for quantum i
    studies at the LHC, complementing tt̄
schema_version: v3
problem_id: arxiv_2504.01496
title: Entanglement and Bell Nonlocality in τ+ τ- at the LHC using Machine Le
  for Neutrino Reconstruction
physics_task:
  extensions:
    quantum_observables:
    - entanglement
    - Bell nonlocality
    system: τ+τ- two-qubit state
    process: pp → τ+ τ- X
  category: quantum_tomography
  subcategory: spin_density_matrix_reconstruction
  ml_formulation:
  - conditional_generation
  - regression
experiment_contexts:
- extensions:
    process: pp → τ+τ- X
    physics_goal: measure full spin density matrix, entanglement and Bell non
      surpassing 5σ
  experiment: τ+τ- production at the LHC
  facility: CERN Large Hadron Collider (LHC)
  detector: null
  role: quantum tomography and measurement of entanglement and Bell nonlocali
    the τ+τ- two-qubit state using machine-learning neutrino momentum reconst
  conditions: null
  evidence_ids:
  - source_section_001.part_001
  - process_001
- extensions:
    trigger_subchannels:
```

knowledge-base record: arXiv:2504.01496

Summary

- **Density-based clustering has been preliminarily validated**
  - Adapted from Drift Chamber tracking to TPC track finding
  - Good performance for simple final states with well-separated tracks
  - Dense and overlapping tracks inside jets and low-momentum curling tracksremain challenging
- **AIDE-optimized solutions show a similar trend**
  - Promising performance for simple events
  - Clear degradation in complex jet environments
  - Current results demonstrate the potential of AIDE, but also indicate the need for stronger task constraints, physics priors, and better search-space design

Outlook

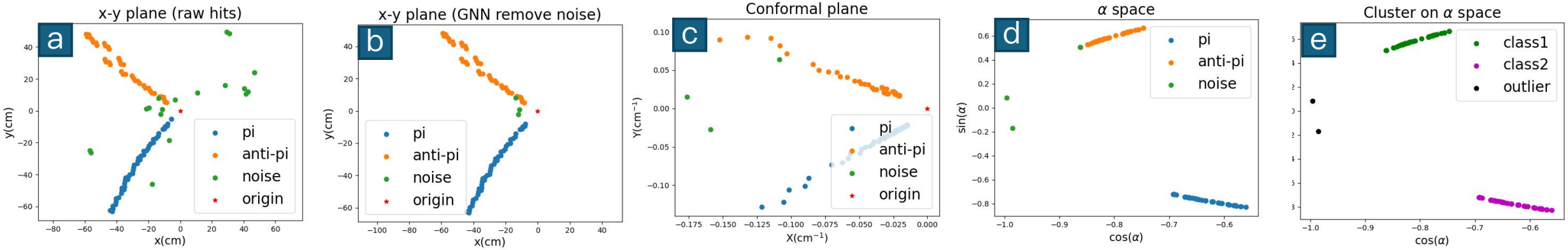
- **dedicated preprocessing for complex CEPC tracking**
  - Local structure recognition inside jets
  - Dedicated treatment of low-momentum curling tracks
- **Knowledge base move from a static retrieval system to a self-improving HEP-ML knowledge engine**
  - Introduce reinforcement learning into the knowledge-base update process
  - The agent’s retrieval and recommendation actions can be evaluated using feedback from downstream experiments

BACKUP

DBSCAN

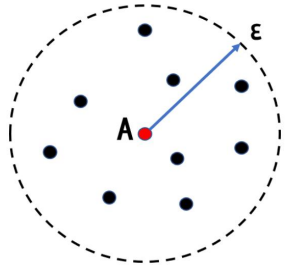
- α : the angle between the line through the data point in the conformal plane and the origin, and the positive X-axis

- BESIII : $J/\psi \to \rho^0\pi^0 \to \gamma\gamma\,\pi^+\pi^-$

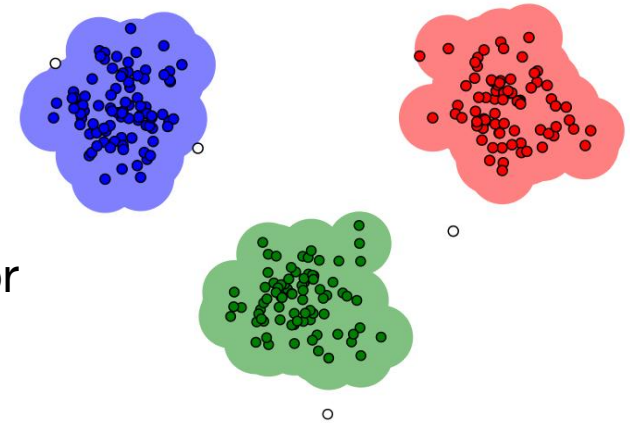


BACKUP

DBSCAN (Density-Based Spatial Clustering of Applications with Noise)



- A density-based clustering algorithm that can automatically discover clusters of arbitrary shapes and identify noise points
- Robust to outliers
- Not require the number of clusters to be told beforehand
- Parameter
  - Epsilon (radius of the circle to be created around each data point)
  - MinPoints (the minimum number of data points required inside that circle for that data point to be classified as a Core point)
  - Choose MinPoints based on the dimensionality (≥dim+1), and epsilon based on the elbow in the k-distance graph



BACKUP

RANSAC (Random Sample Consensus)

- Basic idea: randomly select a subset of data points, fit a model based on these points, and then judge whether the remaining data points belong to the inlier set by calculating their distances to the model
- Accurately estimate model parameters even in the presence of noise and outliers
- The specific steps
  - Randomly select a small subset of data, called the inlier set
  - Fit a model based on the inlier set
  - Calculate the distances between the remaining data points and the model, and classify these points as inliers or outliers based on a certain threshold
  - If the number of inliers reaches a preset threshold, the algorithm exits and the current model is considered good
  - If the number of inliers is not enough, repeat steps 1-4 until the maximum iteration times are reached
- Parameters such as threshold and iteration times need to be preset

