Deep Learning Application in the Visible Decay Search for Dark Photons with DarkSHINE Experiment

Zejia Lu, Huayang Wang, Liang Li

Shanghai Jiao Tong University

Jan 20th , 2026

The 2nd "AI+HEP in East Asia" Workshop @ KEK

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

Content

- Introduction
- Traditional Tracking
- GNN-based Tracking
- Performance & Impact
- Summary & Outlook

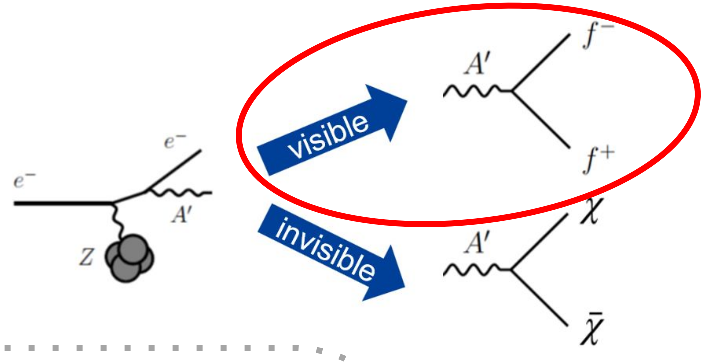
Dark Photon Search

- **Light dark matter search: dark photon**

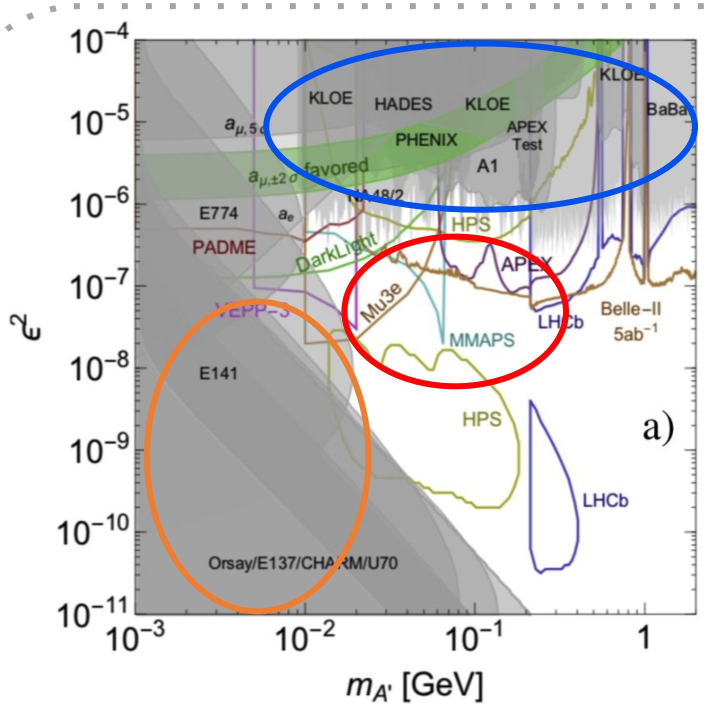
$$\mathcal{L} = \mathcal{L}_{SM} + \boxed{\frac{1}{2}\frac{\epsilon}{\cos\theta_W} F^{Y,\mu\nu}F'_{\mu\nu}} + \frac{1}{4}F'^{,\mu\nu}F'_{\mu\nu} + m_{A'}^2 A'^2$$

Two model parameters:
- Coupling constant ϵ
- Dark photon mass $m_{A'}$

We will focus on **visible decay**



According to the decay channel of dark photon, we have invisible and visible decay.



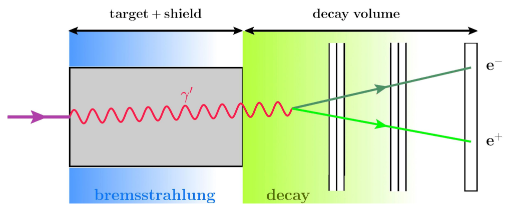
arxiv:2006.04640

Bump hunting : High production rate

Challenging region :
- Signal rate too low for bump hunting
- Lifetime too short for beam-dump experiment

-> **Displaced vertex reconstruction needed!**

Beam-dump experiment : Long decay length



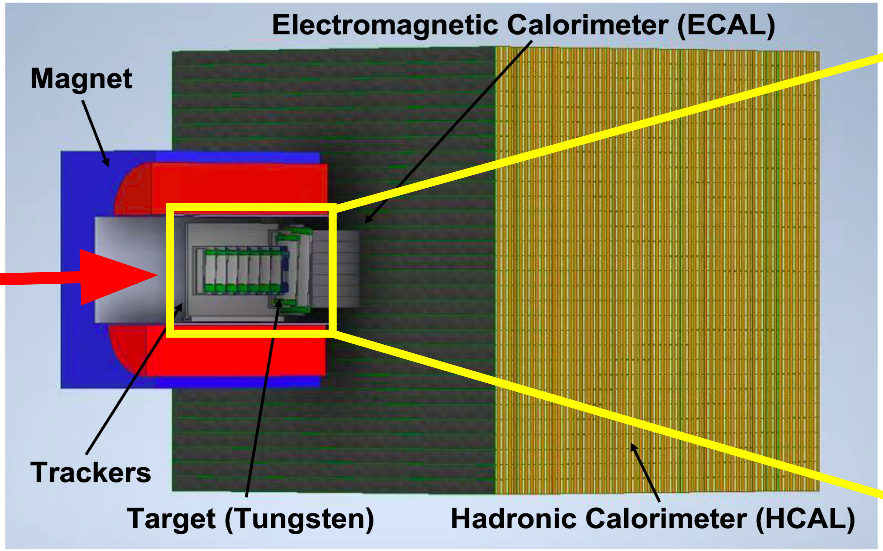
DarkSHINE Experiment

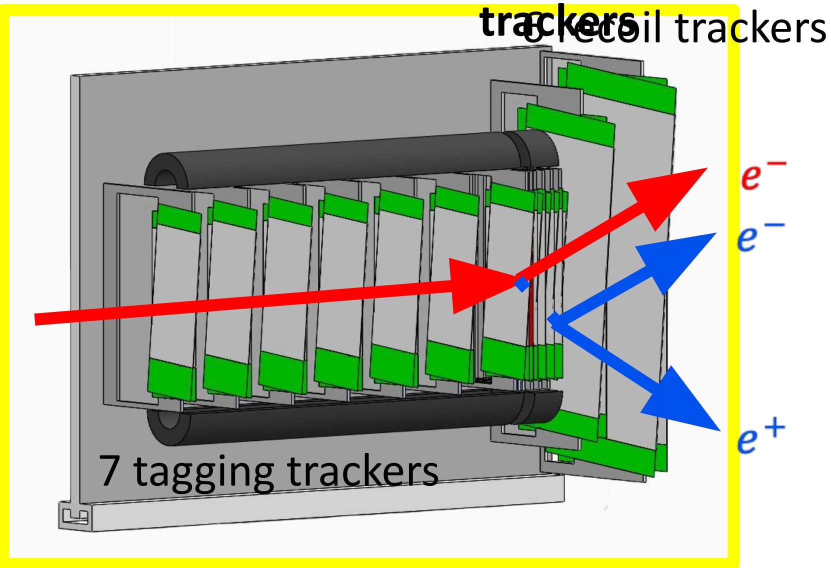
- DarkSHINE is a proposed **fixed-target** experiment that utilizing **electron beam from SHINE** aimed at searching the light dark matter.
- The DarkSHINE detector system is consisted of **tagging tracker**, **recoil tracker**, **ECAL** and **HCAL**.

- **Stagger strip trackers**

8 GeV e^- beam

from Shanghai High Repetition-Rate XFEL and

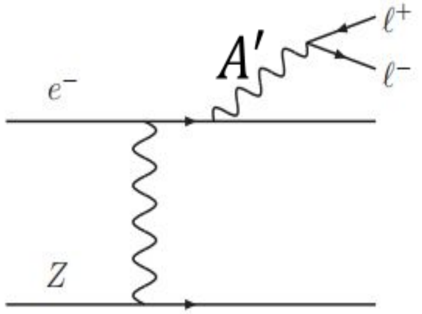




Tracker parameter	Strip width	Sensor thickness
	30 µm	150 µm

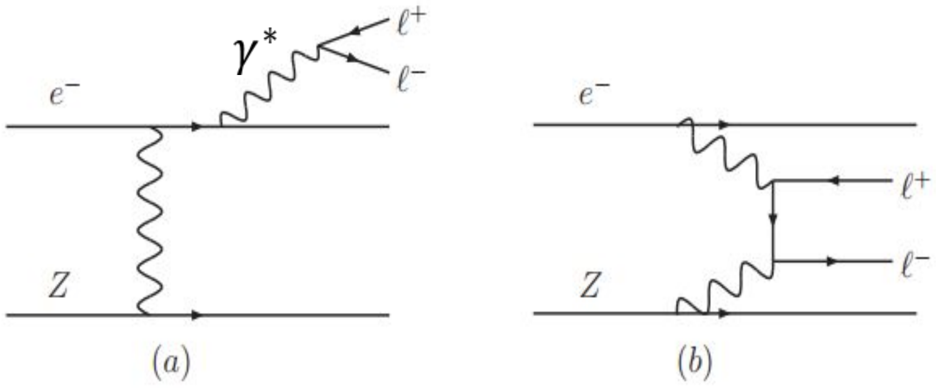
Signal and Background

Signa
l

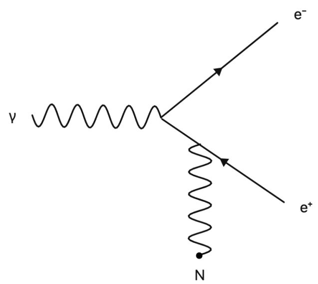


A' production and decay

Main background

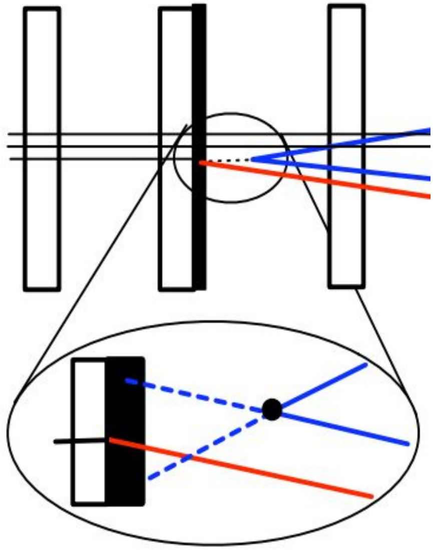


electron pair production



γ conversion followed by bremsstrahlung

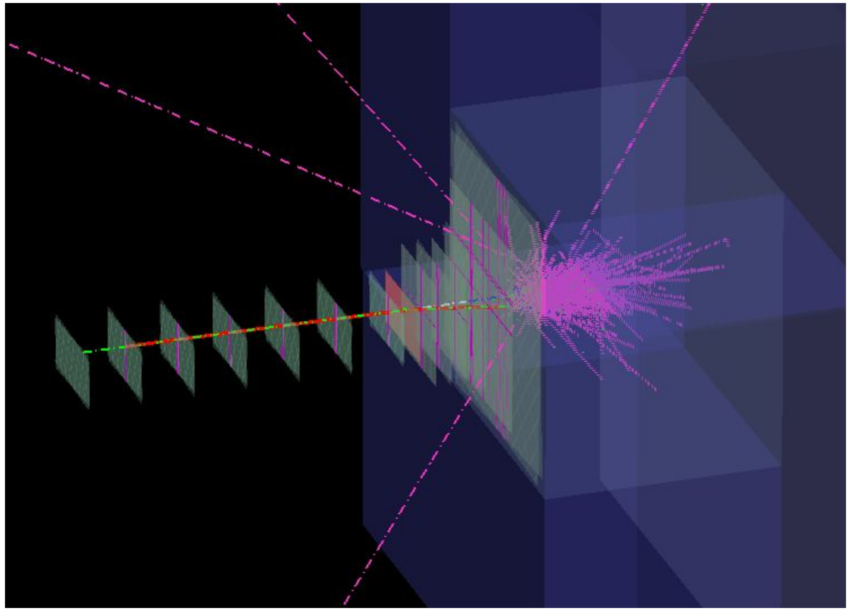
Displaced vertex

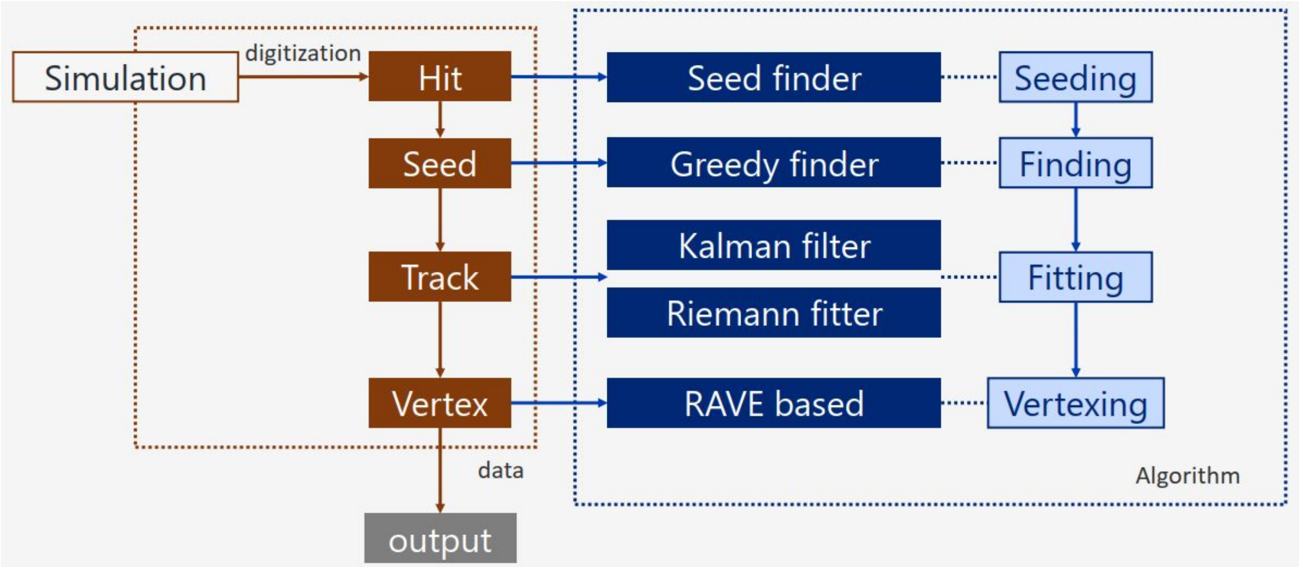


- To differentiate signal from background, the key is to **reconstruct the displaced vertex through tracking and vertexing.**

Simulation and Reconstruction

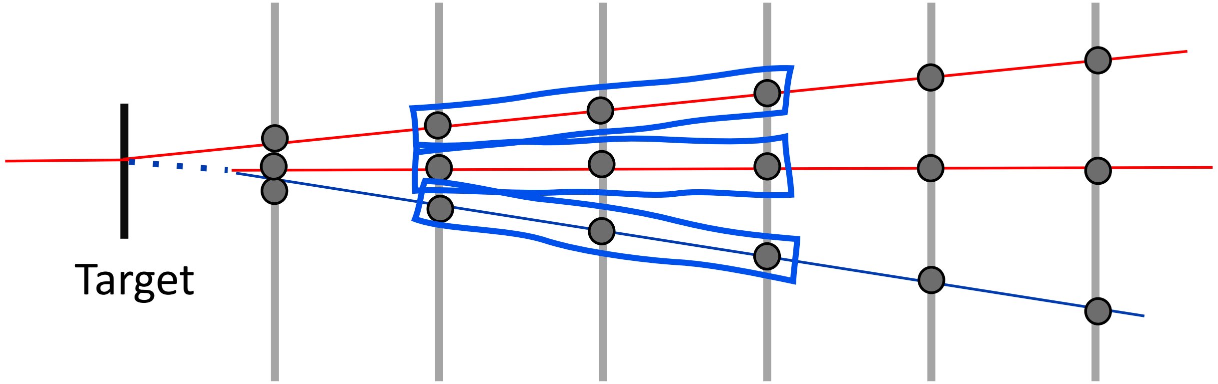
- We use **Geant4-based** simulation to study the signal and background processes. **CalcHEP** generator is used for signal production.
- We apply **full chain reconstruction** from hits to tracks and to vertexes. We adopt Kalman Filter algorithm for both tracking (**GenFit**) and vertexing (**Rave**).

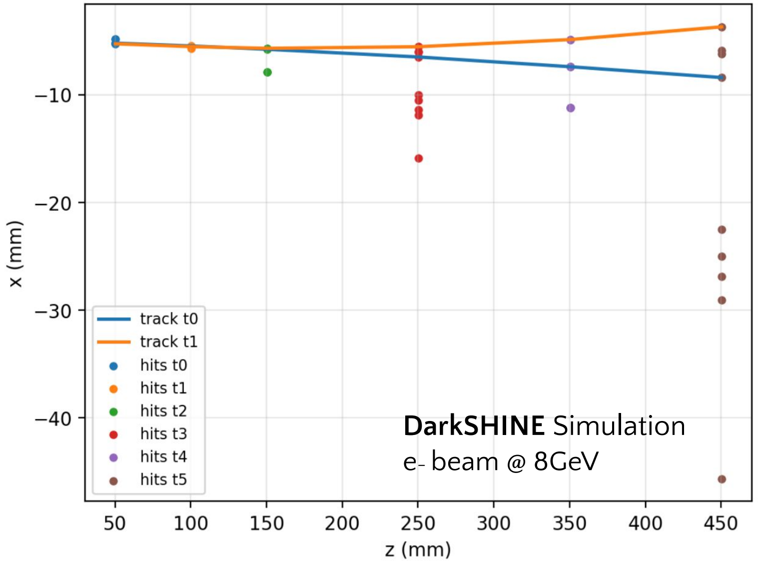




Challenge in Track Finding

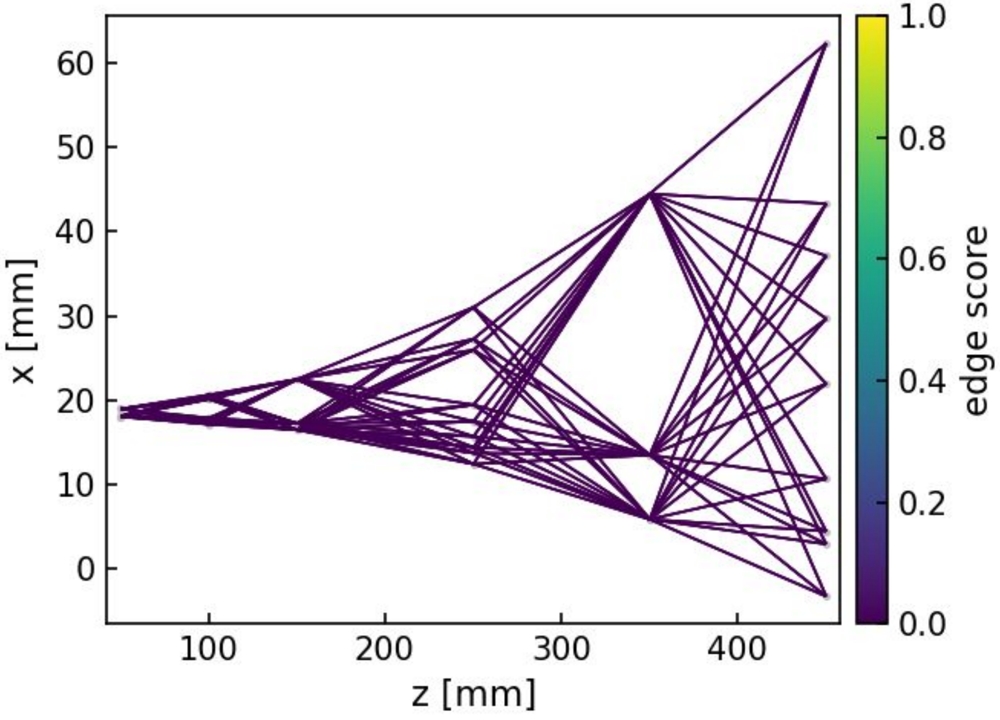
- Track finding
  - Greedy algorithm + helix fit.
  - Start from seed layers and extend to other layers.
  - Good tracking efficiency for single-track
    - 97% in tag-track case
    - 60% in 3-track case
- Challenge in Track Finding
  - Low momentum tracks suffer from multiple scattering effect, hard to find.
  - In fixed-target experiment, most tracks is highly forward with small separate angles. It is easy to misassign hits.

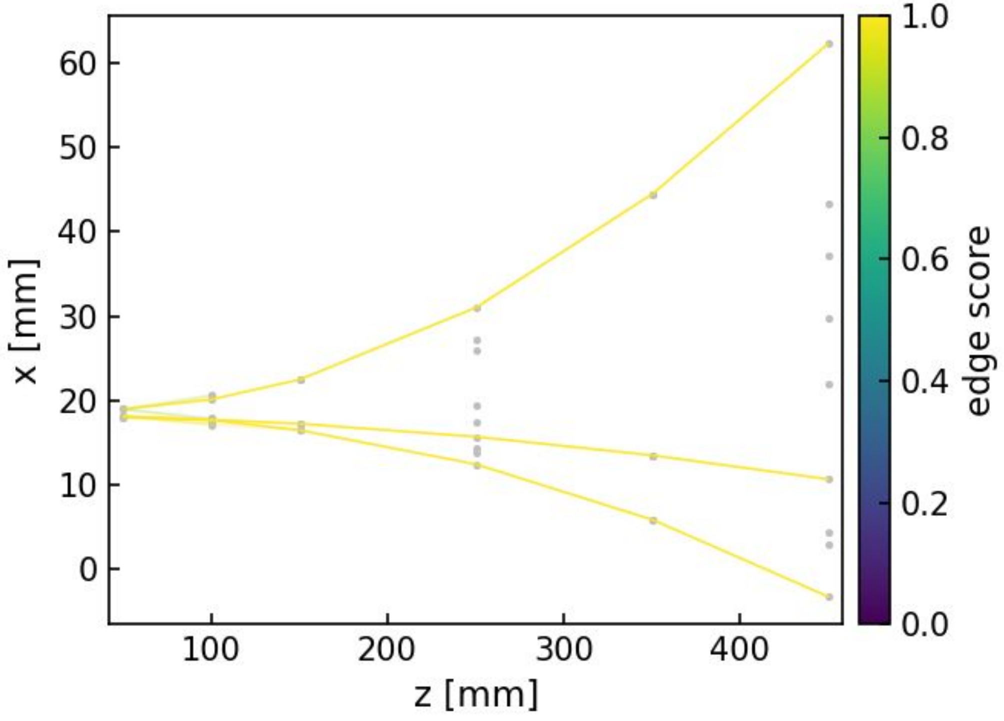




GNN Track Finding: Strategy

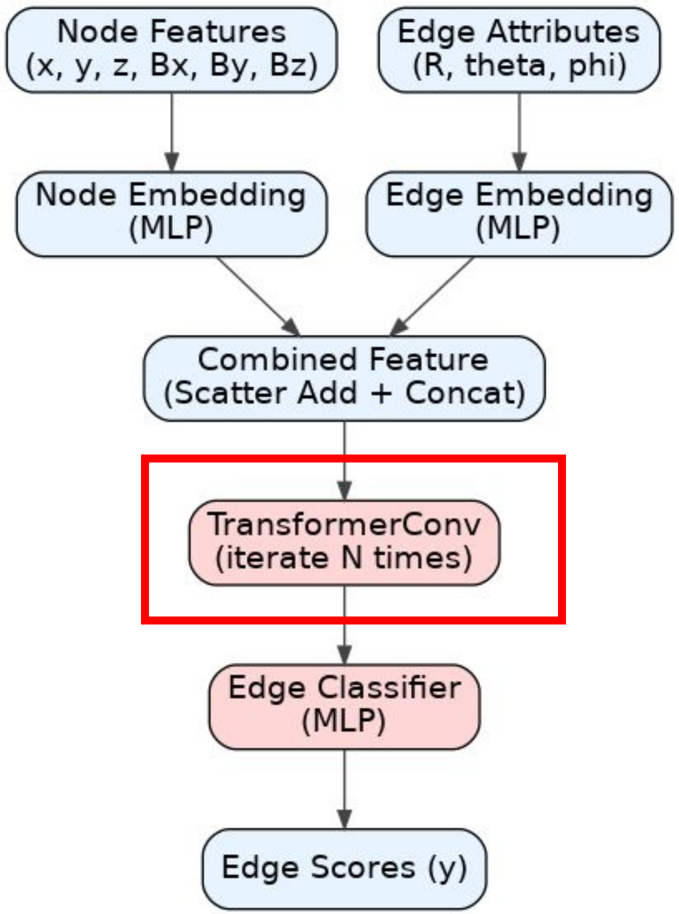
- Build a graph that connects every hits in the adjacent tracker layers, predict the score for every edges.
- After the prediction of edge scores, use the clustering algorithm to form hits to tracks.





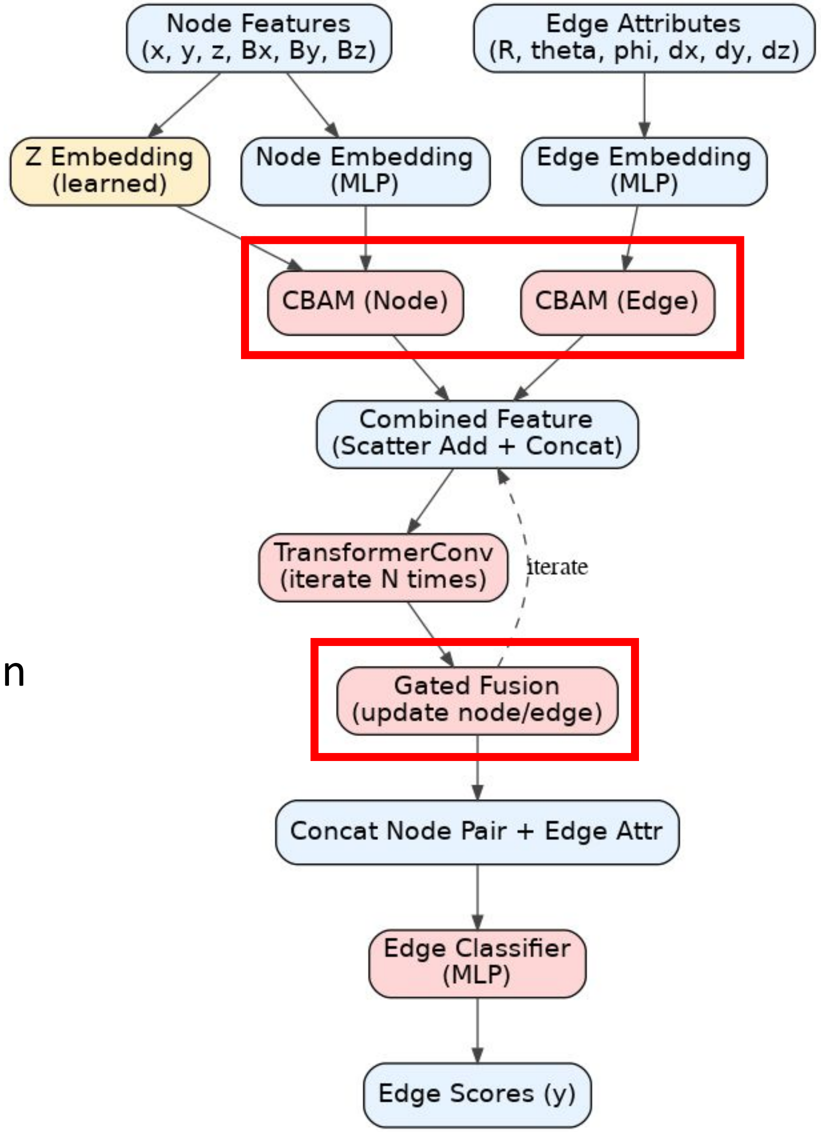
GNN Tracking: Build the Network

LinkNet



- Convolutional Block Attention Module
  - Channel dimension
  - Spatial dimension
- Gated Fusion

TrackNet



GNN Tracking: Training

Dataset

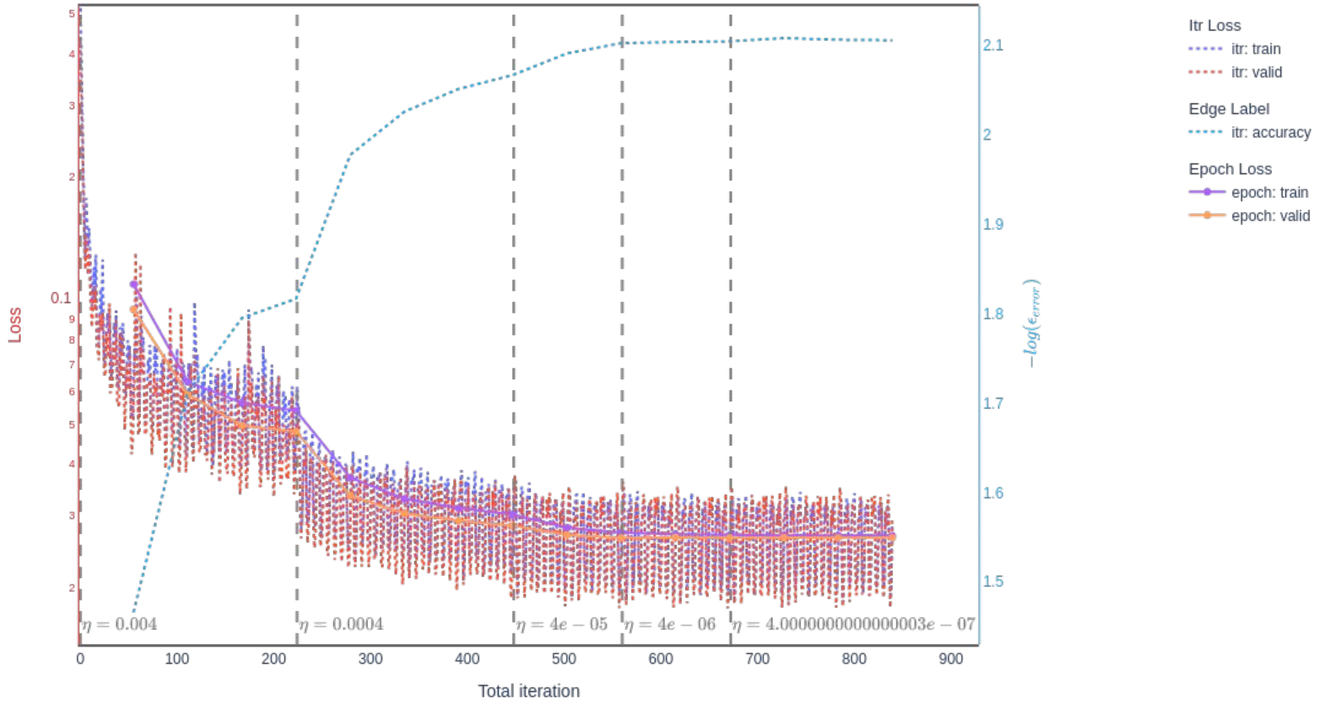
- gamma conversion, e pair production, signal with masses (20, 50, 100, 200, 500 MeV, $\epsilon = 10^{-4}$). Each process 10^5 events.
- train: validation: test $= 3:1:1$

Loss function

- Binary cross-entropy with logit
- $L = \sum_i [-\hat{y}_i \log(\sigma(y_i)) - (1 - \hat{y}_i) \log(1 - \sigma(y_i))]$

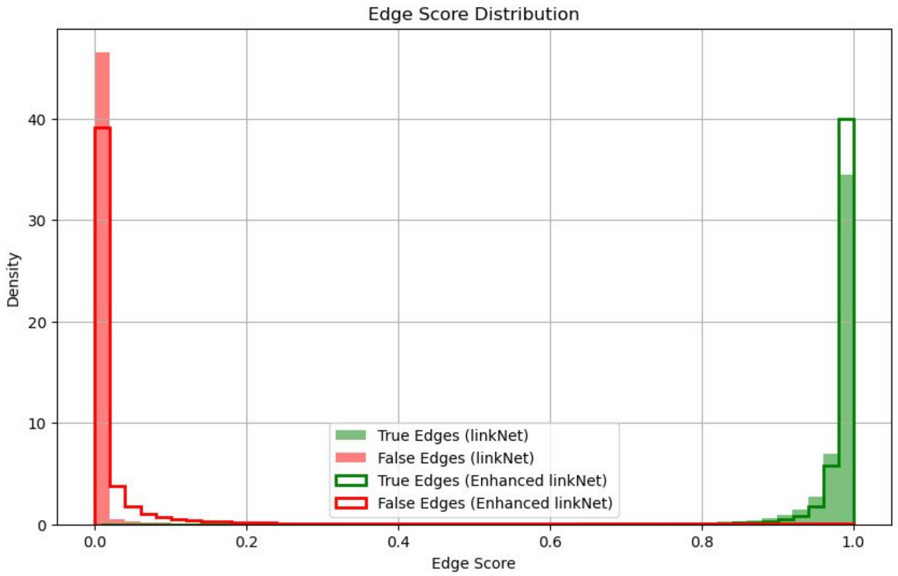
Optimization

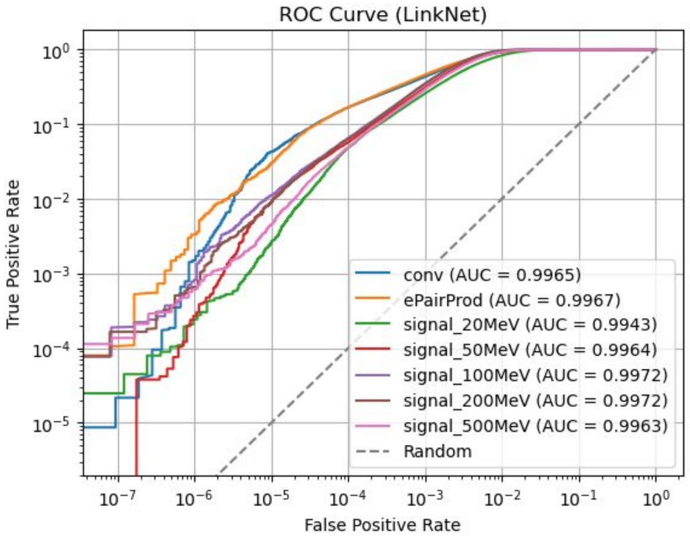
- Adam algorithm

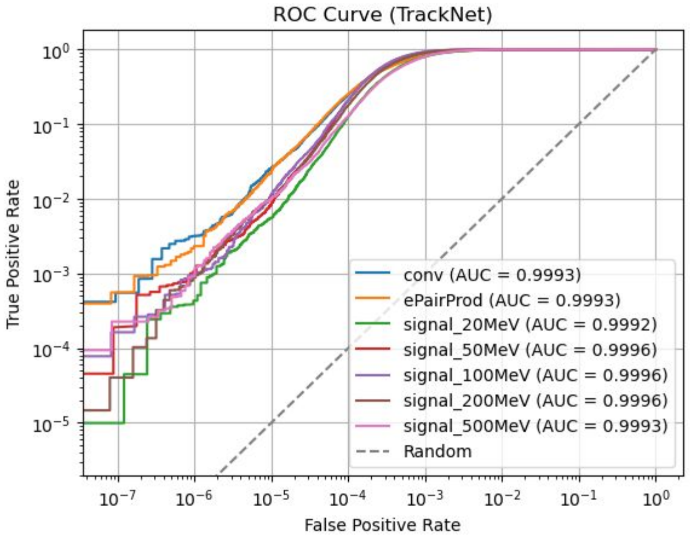


GNN Tracking: Result

- As a classification task, the scores for true and false edges are well separated.
- According to the ROC curve, TrackNet yields better classification performance with AUC larger than 99.9%.

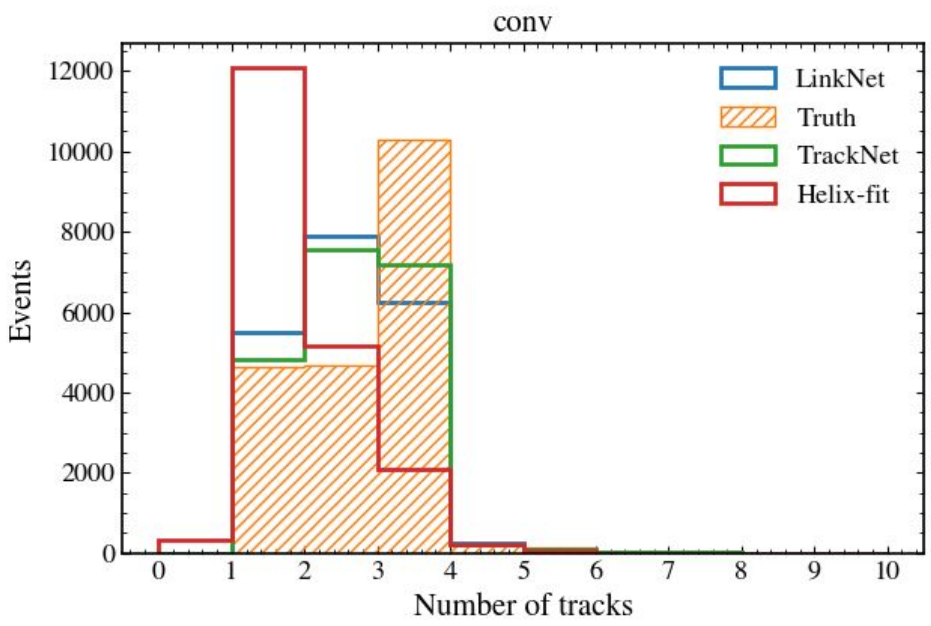


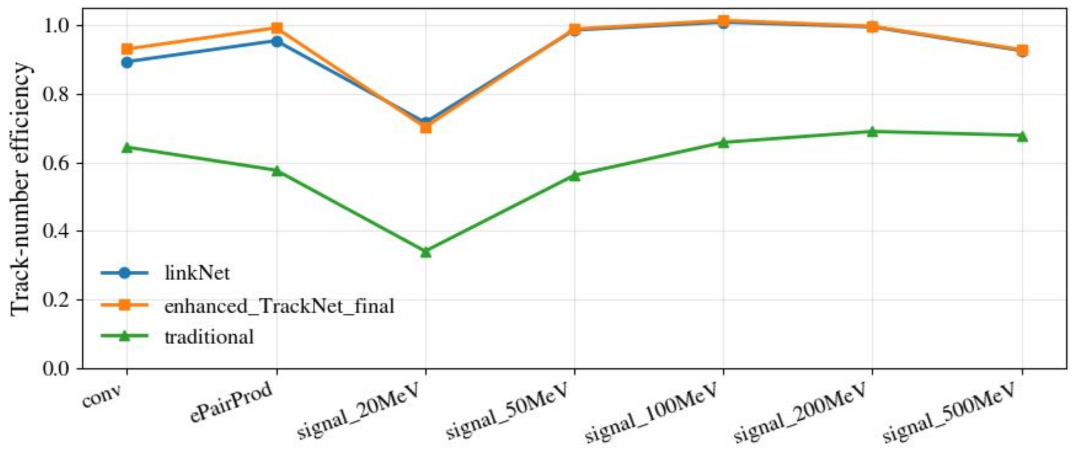


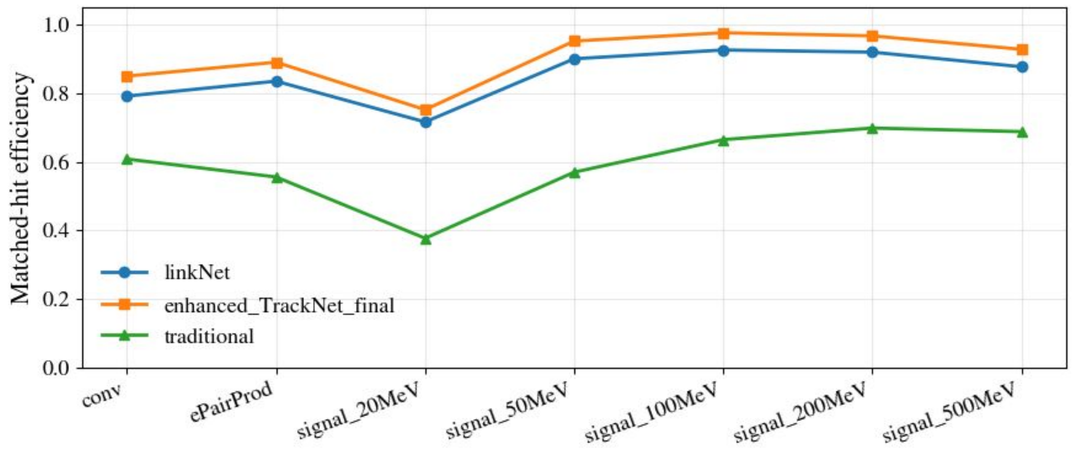


GNN Tracking: Result

- The output of GNN is the score of each edge. After cut on the score, we use the Dijkstra global search algorithm to select the tracks with highest score.



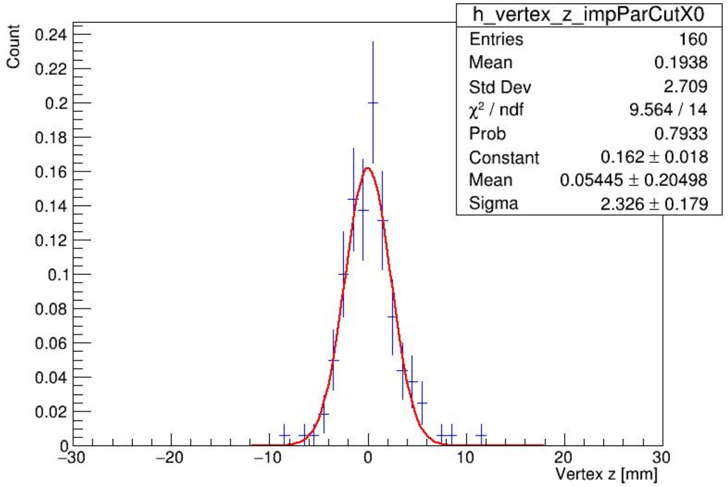


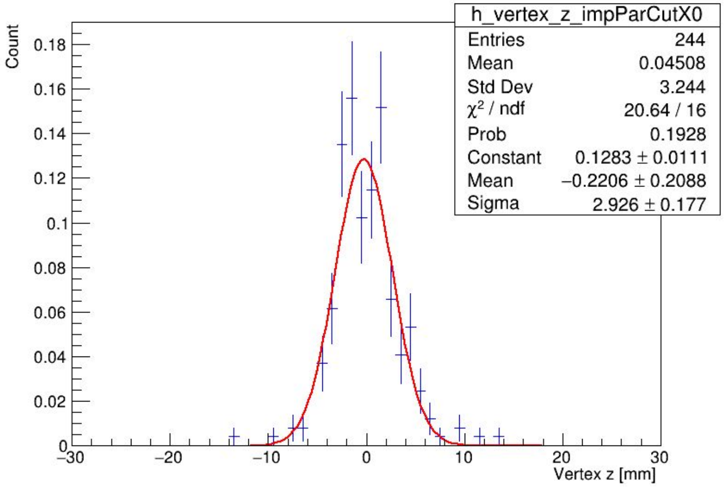


Vertexing

- The reconstructed track is further used for vertex reconstruction.
- The vertexing algorithm is minimal chi-square + Kalman filter. Additional cuts are applied to optimize the vertex resolution.

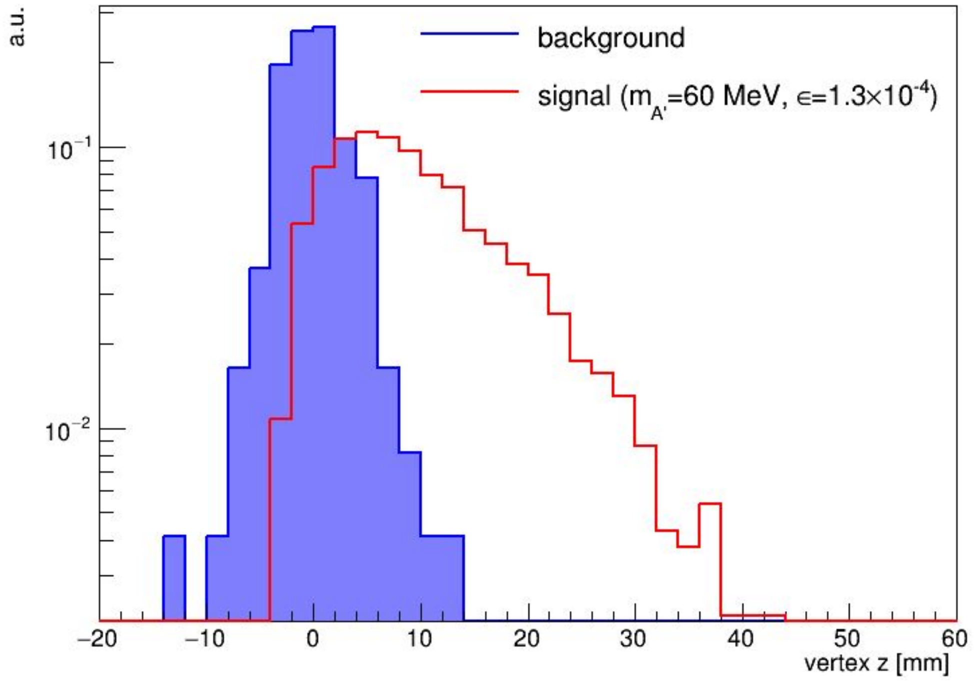
Cut summary	
Kinematics cut	Track p > 0.5 GeV
	Invariant mass > 20 MeV
Reconstruction track cut	Shared hit num = 0
	Impact parameter cut
Vertex parameter cut	Recon vertex num >= 1
	Vertex dispersion < 0.4 mm
	Vertex theta > 0.012 rad
	Vertex projection – IP distance < 0.3 mm





Vertexing for Visible Decay Search

- We rely on the reconstructed vertex position to search for visible decay signal.
- Due to the statistics limit, we assume the ideal Gaussian shape of vertex z distribution after the vertex quality cuts, so that we can define a region with no background.

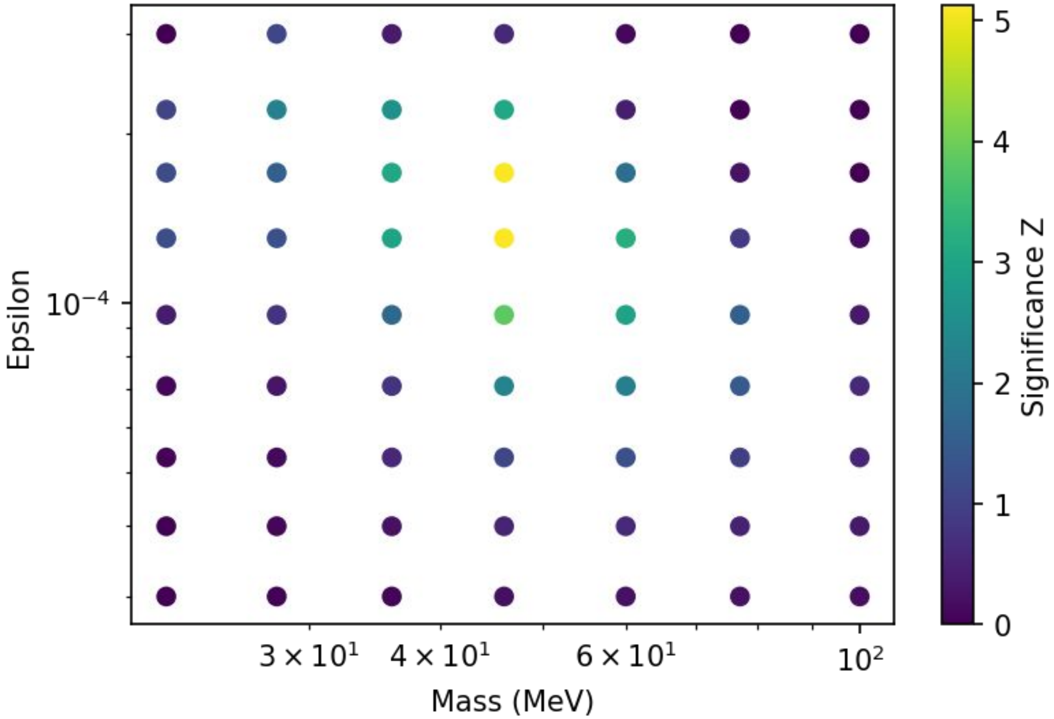


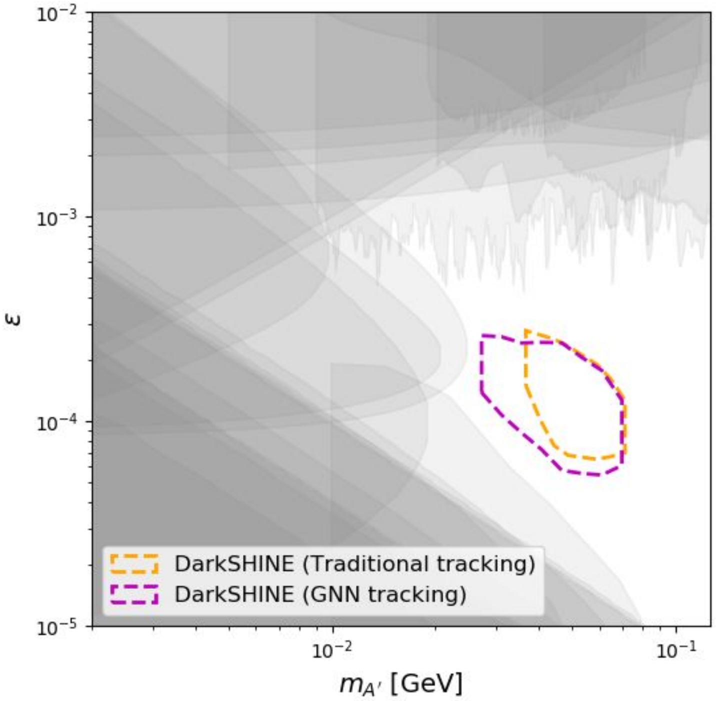
EOT: 3E14

Event type	eBrem + conv	ePairProd
Event num	7.5E12	4.3E11
Event num (after cut)	2.2E10	1.2E9
0.1 background range	**> 6.4 sigma (19mm, 50mm)**	

Exclusion Limit

- After the definition of signal region for displaced vertex, we can calculate the signal yield, efficiency and significance for each parameter space point.
- Thanks to around 2 times improvement on the signal efficiency, the result with GNN tracking can cover more parameter space with small $m_{A'}$.





Summary

- **Summary**
  - We introduce a **GNN-based track finding pipeline** (LinkNet / TrackNet with CBAM and Gated Fusion) for the DarkSHINE experiment.
  - The network achieves **excellent edge classification performance** with AUC larger than 99.9%.
  - After vertexing and physics cuts, the improved tracking yields **about a factor of two gain in signal efficiency**.
  - This directly enhances the **sensitivity to visible dark photon decays**, especially in the low-mass region.
- **Outlook**
  - Incorporate **more realistic detector effects** (noise, inefficiency, misalignment) into training.
  - Explore **end-to-end learning** from hits to vertices and joint optimization of tracking and vertexing.

上海交通大學

SHANGHAI JIAO TONG UNIVERSITY

Thank You

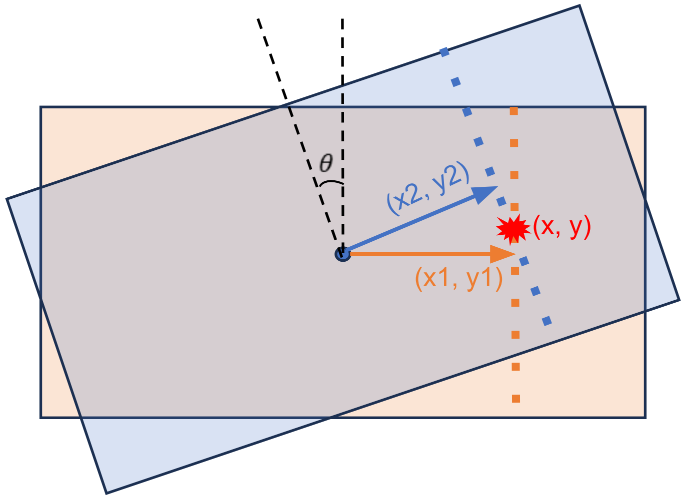
饮 水 思 源 爱 国 荣 校

Hit Reconstruction

Strip clustering

Merge u, v layer

- Cluster strip hits on single layer.
- Use mean shift clustering, based on the weighted distance (energy deposition of hits) to the center of cluster.



$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} x_1 \cos\theta_1 - y_1 \sin\theta_1 \\ x_2 \cos\theta_2 - y_2 \sin\theta_2 \end{pmatrix}$$

$$\boldsymbol{A} = \begin{pmatrix} \cos\theta_1 & -\sin\theta_1 \\ \cos\theta_2 & -\sin\theta_2 \end{pmatrix}$$

$$\begin{pmatrix} x \\ y \end{pmatrix} = \boldsymbol{A}^{-1} \begin{pmatrix} u \\ v \end{pmatrix}$$

$$\begin{pmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{pmatrix} = \boldsymbol{A}^{-1} \begin{pmatrix} \sigma_u^2 & \sigma_{uv} \\ \sigma_{uv} & \sigma_v^2 \end{pmatrix} (\boldsymbol{A}^{-1})^T$$

- Use transformation of coordinates, can handle any cases.
- With error propagation.