Using the FM4NPP for the CLAS12 CVT

Alessio Illari[1], Aleksandr Bulgakov[1], Richard Capobianco[1], Michael Frank[1], Andrey Kim[1]
Áron Kripkó[1], David Park[2], Huan-Hsin Tseng[2], Shinjae Yoo[2], Kyungseon Joo[1]

[1] University of Connecticut [2] Brookhaven National Laboratory

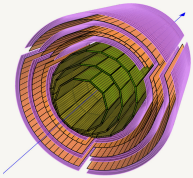
CLAS Collaboration Meeting

November 17, 2026

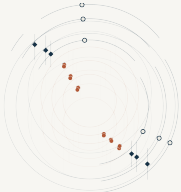
UCONN UNIVERSITY OF CONNECTICUT Jefferson Lab Brookhaven National Laboratory U.S. DEPARTMENT of ENERGY | Office of Science

One CVT event, three reconstruction tasks

CLAS12 CVT



Silicon + Micromegas

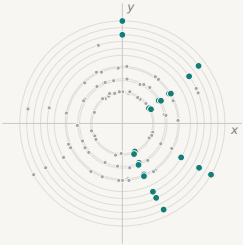


Sparse measurements on separated layers

Illustrative event — MC truth

Non-signal tagging

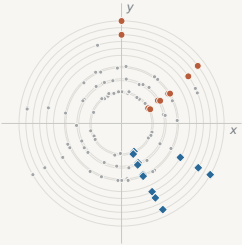
Which clusters are signal?



22 signal clusters
86 non-signal clusters

Track finding

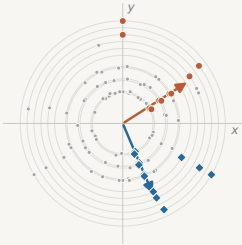
Which clusters belong together?



Track 1: 10 clusters
Track 2: 12 clusters

Momentum

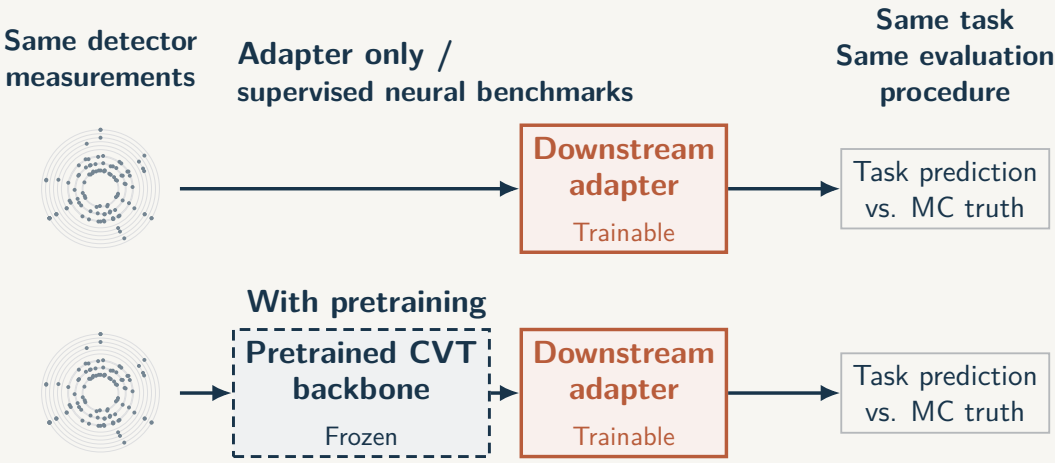
What momentum does a track carry?



$p_1 = 0.90\ \mathrm{GeV}/c$
$p_2 = 1.35\ \mathrm{GeV}/c$

CVT artwork: S. J. Paul et al., NIM A 1049, 168032 (2023).

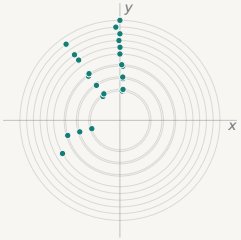
Does pretraining improve reconstruction?



Simulated particles, measured background and MC truth

RGA Fall 2018 conditions; one p, one π^+ and one K^+ per generated event

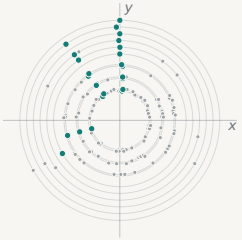
Without overlay



29 clusters

Earlier DNP benchmark
Tracking / momentum

With 45 nA background



165 clusters

Newer benchmark
Tracking / tagging

● Signal ● Non-signal

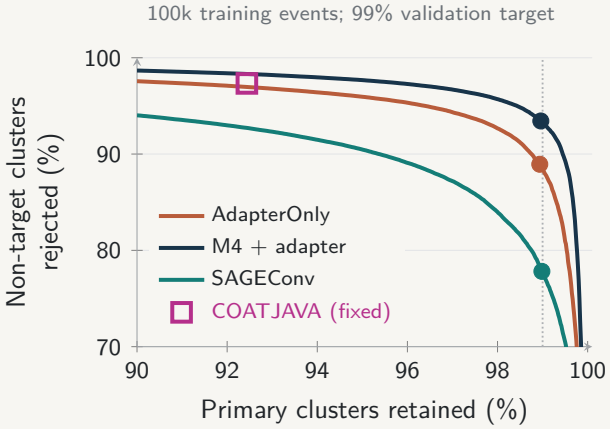
Truth labels from `MC::True`

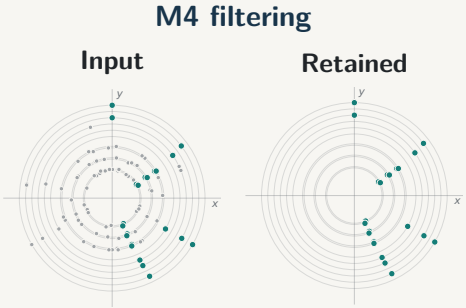
Track finding & noise tagging
Hit-level particle associations define truth tracks and primary-signal labels.

Momentum regression
Truth momentum at the innermost associated CVT measurement.

Train / validation / test
Disjoint background source files across splits; no background files shared between training and evaluation.

Signal preservation versus non-target rejection



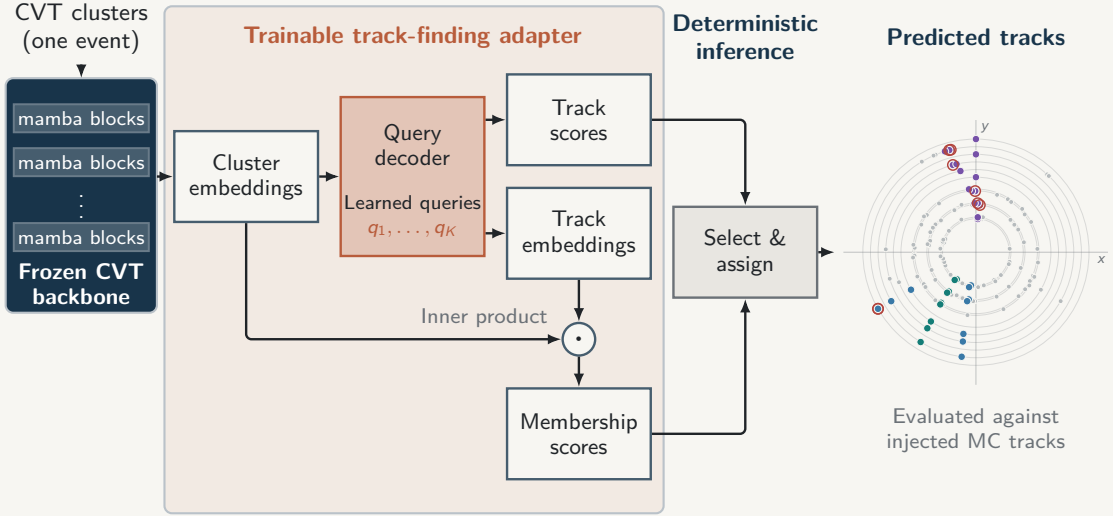


108 → 23 clusters

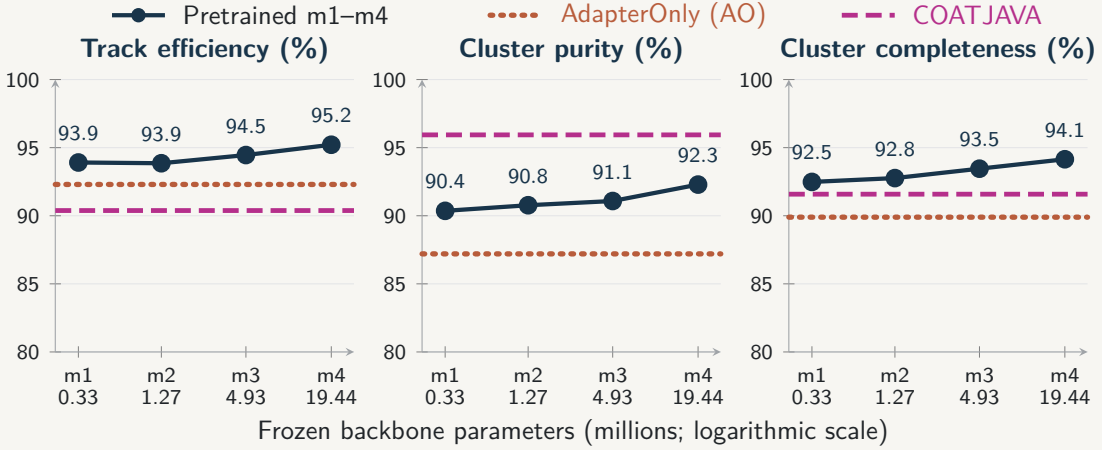
22/22 primary clusters retained

M4: 99.0% primary retention with 93.4% non-target rejection.

Track finding: recovering injected tracks

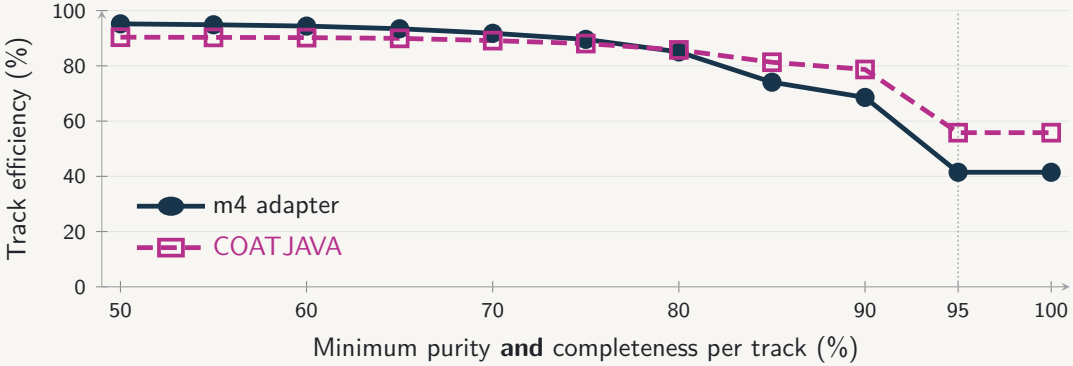


Pretraining improves track finding



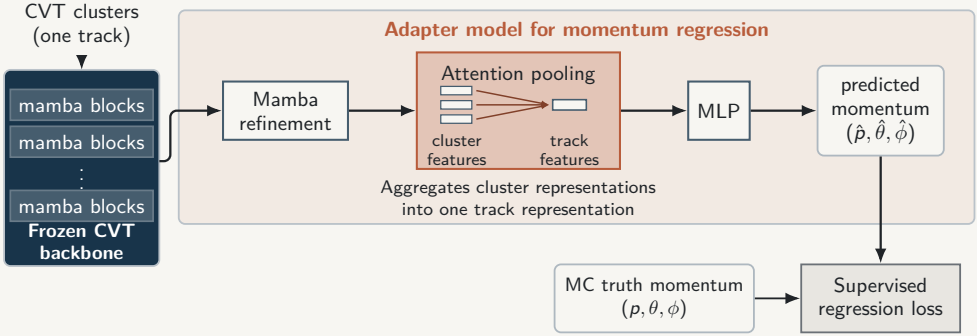
Same 10k test events. Efficiency: IoU, purity, completeness $\geq 50\%$; cluster metrics: MC associations.
AO and m1–m4: 100k training events, shared 60k-update recipe, seed 23, Dice-selected checkpoints.

But what happens when we demand high-quality reconstructed tracks?



At 95% purity and completeness: 55.8% COATJAVA vs. 41.5% m4

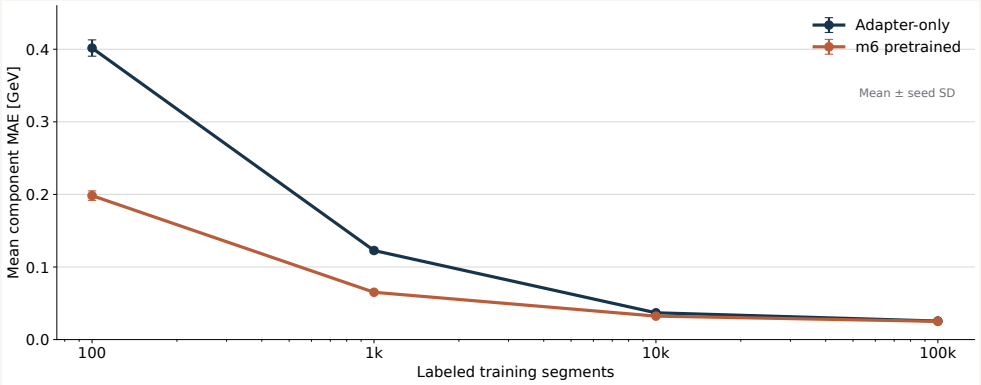
Momentum reconstruction requires track-level adaptation



Truth-associated clusters supply one track at a time.
This benchmark isolates regression from association failures.

Pretraining improves momentum reconstruction when labels are scarce

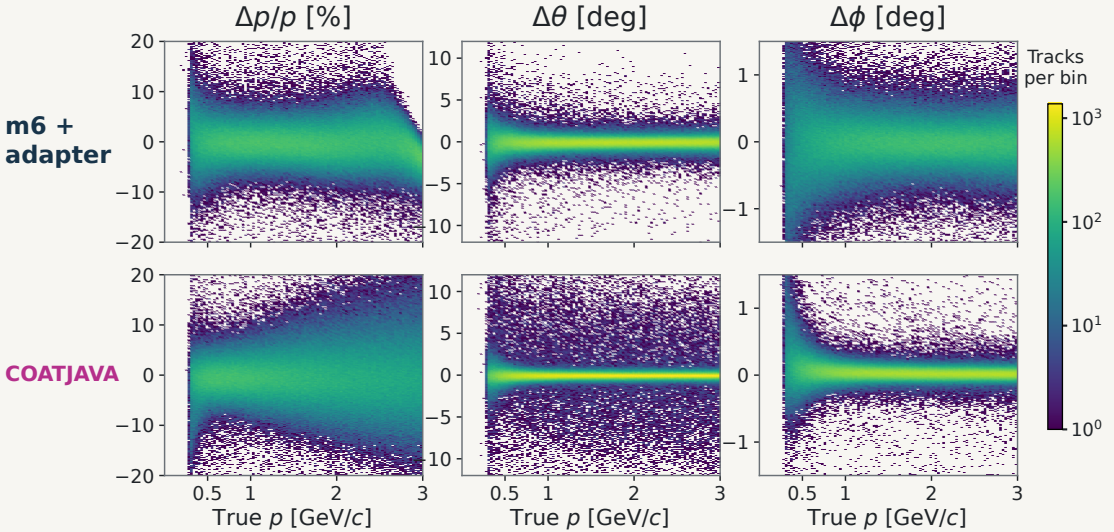
Earlier CVT simulation — no measured-background overlay



Largest advantage at low labels; convergence at 100k.

Momentum residuals: cores and tails

$\Delta = \text{reco} - \text{MC}$



Toward physics-quality CVT reconstruction

Noise tagging
Primary retention with non-signal rejection

Track finding
Improved recovery of injected primary tracks

Momentum
Gains from pretraining when labels are scarce

We are still establishing the **best reconstruction performance** achievable with our current data and methods.

In parallel, we are exploring new data representations, including **strip hits**, and new learning techniques.

This work was supported by the U.S. Department of Energy, Office of Science, under Contract No. DE-FG02-04ER41309.

EXTRA SLIDES

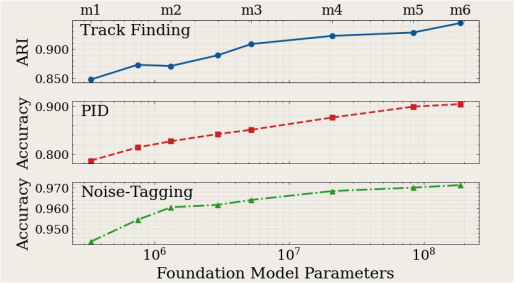
CLAS12 work

CVT simulation ▪ backbone ▪ track finding

FM4NPP transfers across tasks and scales

Metric	AdapterOnly	Best baseline	FM4NPP
Track finding ARI	0.7243	0.8765	**0.9448**
PID accuracy	0.6631	0.7701	**0.9039**
Noise-tagging acc.	0.9111	0.9646	**0.9713**

Baselines: Exa.TrkX (track); OneFormer3D (PID/noise).



FM4NPP result: Self-supervised pretraining produced a representation useful across tasks; downstream performance improved as backbone size increased.

[2] Adapted from D. Park et al., arXiv:2508.14087v1 (2025), Table 2 and Fig. 6.

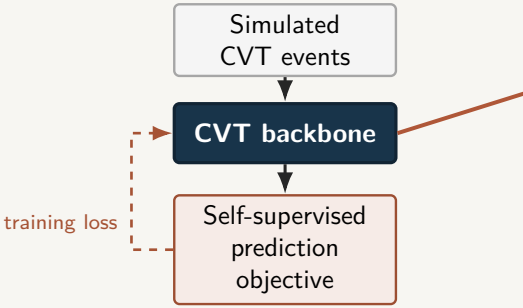
Adapting FM4NPP requires rethinking detector-specific choices

FM4NPP principle		CLAS12 CVT adaptation
Self-supervised geometric pretraining	MODIFIED	Adapt the prediction objective to sparse, layered CVT measurements
Detector-aware serialization	REWORKED	Replace TPC-specific ordering with a CVT-specific, layer-aware serialization
Geometry enters through coordinates and ordering	EXTENDING	Explore explicit detector-geometry information in the pretraining objective
Frozen backbone + lightweight downstream adapters	EXPANDED	Retain the paradigm, but extend it to track-level momentum reconstruction

Methodological transfer, not checkpoint transfer:
the CLAS12 model is pretrained from scratch on CVT data.

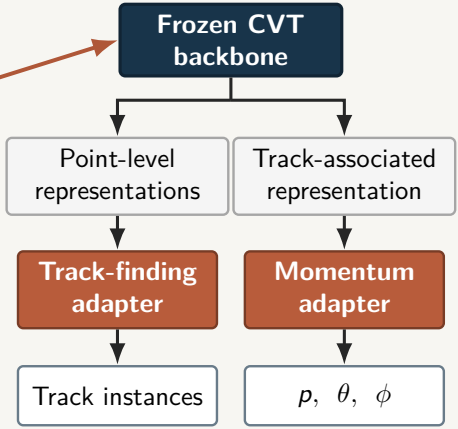
From detector structure to particle physics

1. Self-supervised pretraining



Learn detector representations without downstream reconstruction labels.

2. Downstream reconstruction



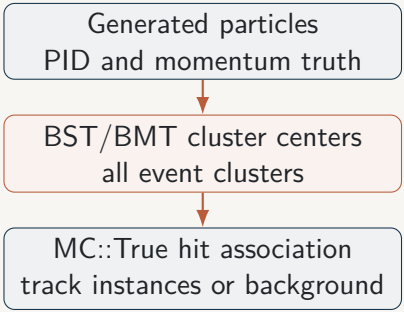
Probe the frozen representation with task-specific supervised adapters.

CLAS12 CVT simulation and skim

Three-particle PID phase-space simulation

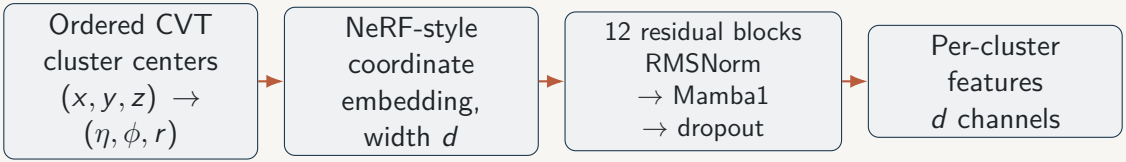
Generated charged particles are transported through the CLAS12 central detector and reconstructed into BST and BMT clusters.

The model sees an event as a variable-length sequence of cluster-center coordinates (x, y, z), including clusters with no reconstructed track ID.



Canonical event skim: cluster centers from BSTRec::Clusters/BMTRec::Clusters; truth groups use primary-origin MC::True hit matches. Groups below eight clusters are labeled background.

CLAS12 backbone architecture



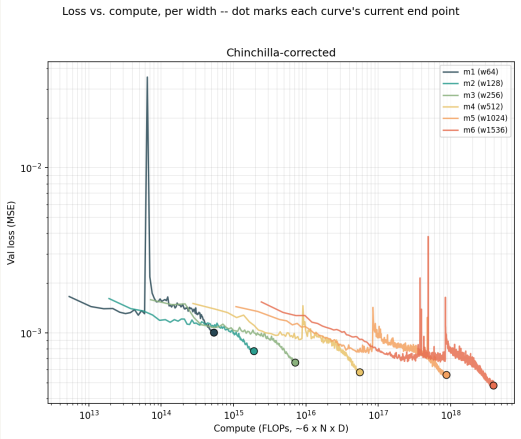
Width sweep

$d = 64, 128, 256, 512, 1024, 1536$; depth stays 12. Mamba state dimension 16, convolution width 4, expansion 2.

Pretraining and reuse

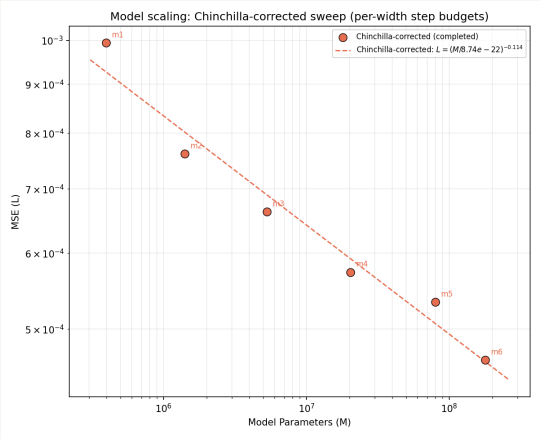
A per-token linear head predicts the next spatial point ($k = 1$). Downstream adapters use frozen intermediate features.

Pretraining loss across compute budgets



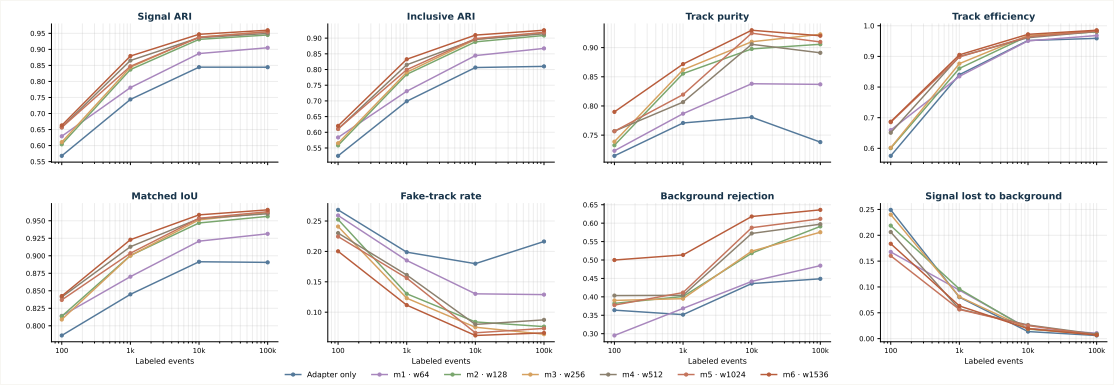
The endpoint of each width's run moves to higher compute as model size grows; validation loss generally falls. Spikes in the traces are transient.

Model size and the fitted scaling trend



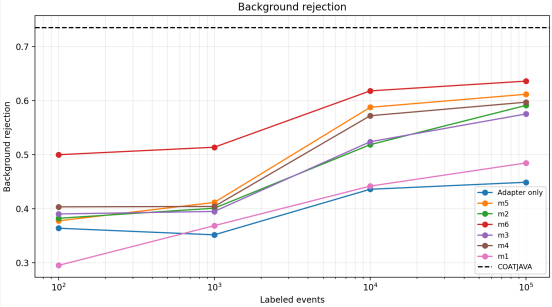
Completed width-sweep endpoints; the dashed line is an empirical fit within this sweep, not an extrapolated compute optimum.

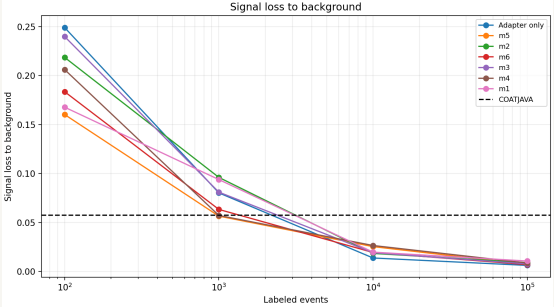
Track finding: all evaluation metrics



Campaign 4 signal-only label sweep v3; 500,000 test events per run, one run per width and label count. Track matching uses IoU, purity, and efficiency thresholds of 0.5.

Track finding: signal and background behavior





The two background decisions have different denominators.

Background rejection asks how many true background clusters stay out of tracks; signal loss asks how many true signal clusters are sent to background.

Track-finding metric definitions

Clustering and matching

Signal ARI: adjusted Rand index over truth-signal clusters only.

Inclusive ARI: adjusted Rand index over signal and background, with predicted background as one label.

Matched IoU: mean $|P \cap T|/|P \cup T|$ for matched predicted and true tracks. A match also needs purity and efficiency ≥ 0.5.

Track and background counts

Track efficiency: matched true tracks / all true tracks.

Track purity: matched predicted tracks / all predicted tracks; fake rate is its complement.

Background rejection: background clusters predicted as background / true background clusters.

Signal loss: signal clusters predicted as background / true signal clusters.

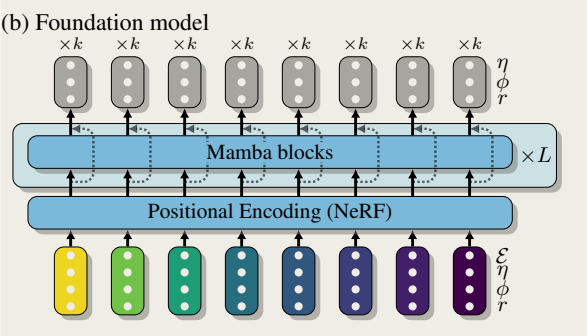
Matches are one-to-one Hungarian assignments maximizing IoU; reported global track ratios pool counts across events.

REFERENCE SLIDES

Original FM4NPP work

sPHENIX TPC ▪ pretraining ▪ benchmarks

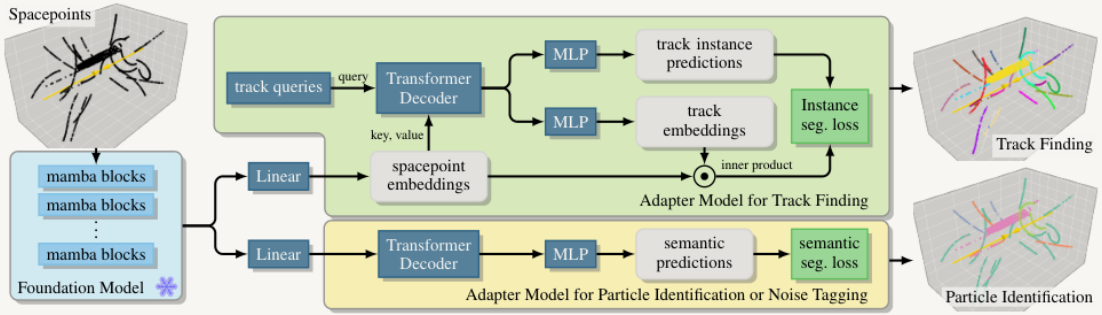
FM4NPP backbone and pretraining objective



- Each TPC spacepoint is a token: energy plus (r, ϕ, η) position.
- Energy and NeRF-style positional encodings feed a stack of Mamba2 blocks.
- For each point, predict the k spatially nearest points at larger radius.
- Train with squared coordinate error, with event-difficulty reweighting.

[2] Adapted from D. Park et al., arXiv:2508.14087v1 (2025), Fig. 3(b) and Sec. 4.

FM4NPP adapters for downstream tasks

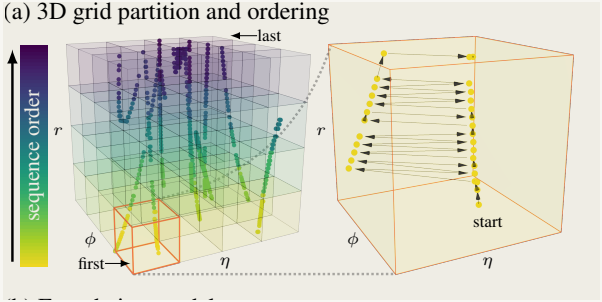


The backbone stays frozen while each task adapter is trained.

PID and noise tagging share the same architecture but are trained separately.

[2] Adapted from D. Park et al., arXiv:2508.14087v1 (2025), Fig. 4 and Sec. 4.

Hierarchical raster scan



1. Transform (x, y, z) to (r, ϕ, η).
2. Partition the TPC into a $6 \times 8 \times 8$ grid in (r, η, ϕ).
3. Order boxes by center (r, ϕ, η); within each box, sort points by radius.

The sequence moves outward globally while keeping nearby spacepoints together locally.

[2] Adapted from D. Park et al., arXiv:2508.14087v1 (2025), Fig. 3(a) and Sec. 4.

Full downstream benchmark

Track finding

Model	Trainable	ARI	Efficiency	Purity
EggNet	0.16M	0.7256	74.19%	75.14%
Exa.TrkX	3.86M	0.8765	91.79%	66.42%
AdapterOnly	2.39M	0.7243	78.01%	64.54%
FM4NPP (m6)	2.39M	**0.9448**	**96.08%**	**93.08%**

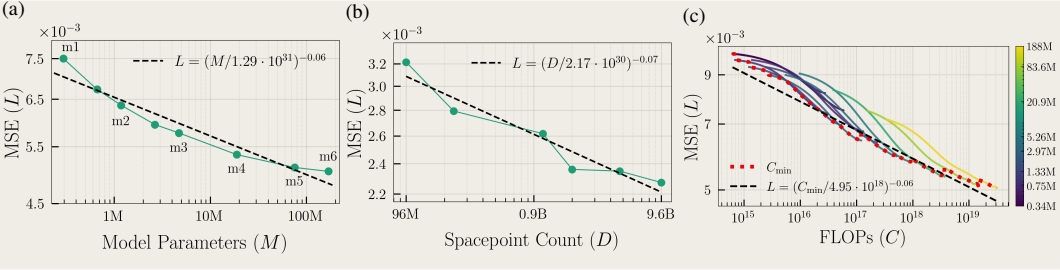
Particle identification and noise tagging

		PID			Noise tagging		
Model	Trainable	Acc.	Recall	Prec.	Acc.	Recall	Prec.
SAGEConv	0.91M	0.7262	0.4563	0.6502	0.9174	0.7227	0.8165
OneFormer3D	44.95M	0.7701	0.4897	0.5767	0.9646	**0.9404**	0.8948
AdapterOnly	0.74M	0.6631	0.3387	0.6111	0.9111	0.6215	0.8359
FM4NPP (m6)	0.74M	**0.9039**	**0.7652**	**0.8782**	**0.9713**	0.9367	**0.9190**

Trainable counts exclude the frozen FM4NPP backbone.

[2] Adapted from D. Park et al., arXiv:2508.14087v1 (2025), Table 2.

Neural scaling during pretraining



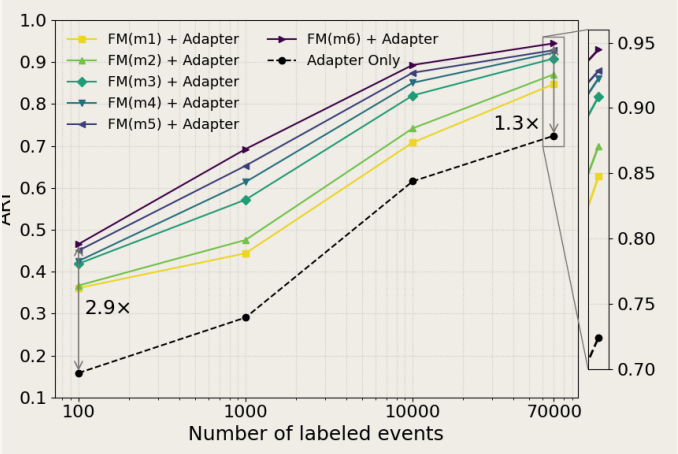
Model size
Validation loss falls as parameters increase.

Training data
The m3 model improves with more spacepoints.

Compute
Larger models become advantageous at higher FLOPs.

[2] Adapted from D. Park et al., arXiv:2508.14087v1 (2025), Fig. 5.

Larger pretrained models need fewer labels

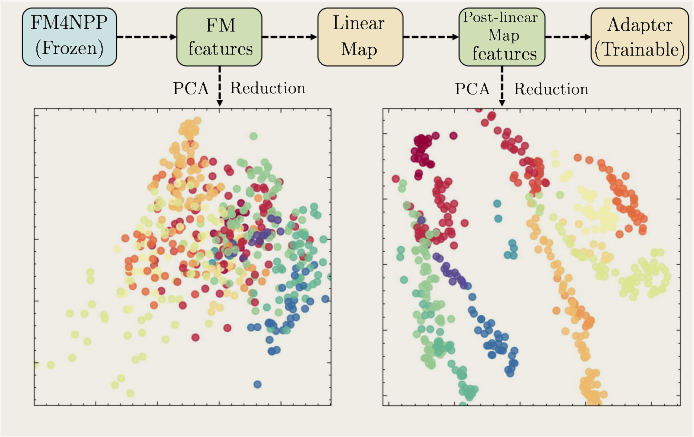


Track-finding ARI rises with labeled events for every model.

The paper reports a 2.9× relative gain over AdapterOnly at 100 labeled events, versus 1.3× at 70,000 events.

[2] Adapted from D. Park et al., arXiv:2508.14087v1 (2025), Fig. 7 and accompanying text.

A linear map specializes FM features

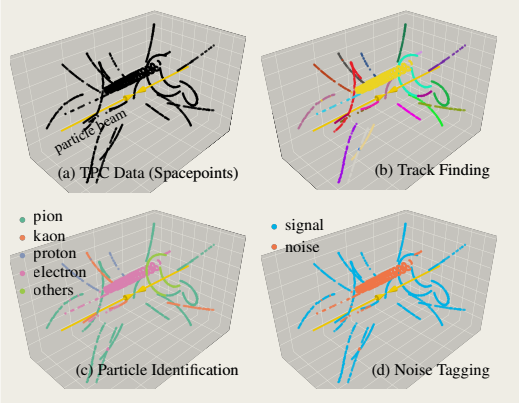


In this representative track-finding example, raw frozen features overlap in PCA.

After one learned linear projection, points group by truth track identity (color).

[2] Adapted from D. Park et al., arXiv:2508.14087v1 (2025), Fig. 8 and accompanying text.

sPHENIX dataset and downstream tasks



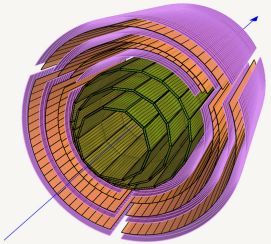
More than 11 million simulated minimum-bias $p + p$ collisions at $\sqrt{s} = 200$ GeV in the sPHENIX TPC.

Inputs are reconstructed spacepoints (E, x, y, z) with truth-particle associations.

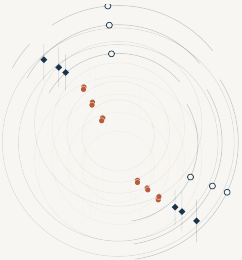
- **Track finding:** group points into tracks.
- **PID:** label each point as pion, kaon, proton, electron, or other.
- **Noise tagging:** identify points from low-momentum secondaries.

[2] Adapted from D. Park et al., arXiv:2508.14087v1 (2025), Fig. 2, Sec. 3, and Appendix A.

Many reconstruction tasks, one underlying detector representation



CLAS12 CVT [1]



Sparse CVT measurements

FOCUS OF THIS TALK

Track finding

Momentum reconstruction

FUTURE DIRECTIONS

β (particle speed)

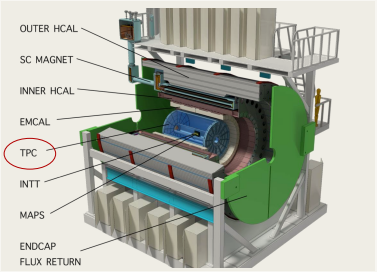
Particle identification

> Foundation-model hypothesis: learn the detector representation once, then specialize it for each task.

[1] Adapted from S. J. Paul et al., *Nucl. Instrum. Methods Phys. Res., Sect. A* **1049**, 168032 (2023).

FM4NPP in the sPHENIX TPC

sPHENIX Time Projection Chamber



More than 11 million events in the original study

One representation, several tasks

Self-supervised pretraining

Learn a detector representation from simulated spacepoints.

Freeze the backbone and adapt

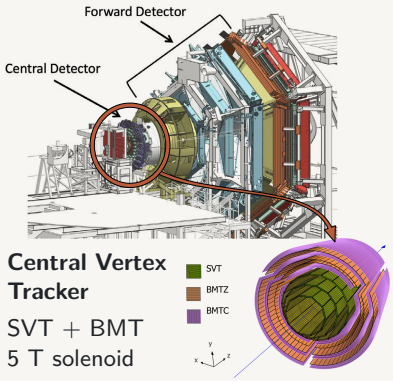
Train small supervised adapters for track finding, particle identification and noise tagging.

Original result: pretraining improved all three downstream tasks.
Larger backbones improved performance and data efficiency.

[2] Adapted from D. Park et al., arXiv:2508.14087v1 (2025), Secs. 3–5. Detector image: Park and Li, NYSDS 2025.

Why adapting FM4NPP to CLAS12 is nontrivial

CLAS12 at Jefferson Lab



Central Vertex Tracker

SVT + BMT

5 T solenoid

Original FM4NPP sPHENIX TPC		This work CLAS12 CVT
Many measurements Hundreds–thousands of spacepoints/event	→	**Few measurements** Sparse cluster measurements
3D point cloud Fine-grained trajectory sampling	→	**Discrete detector layers** Large gaps between measurements
TPC spacepoints	→	**Silicon + Micromegas clusters**

Adaptation challenge: sparse detector layers require changes to the ordering of measurements and the prediction targets.

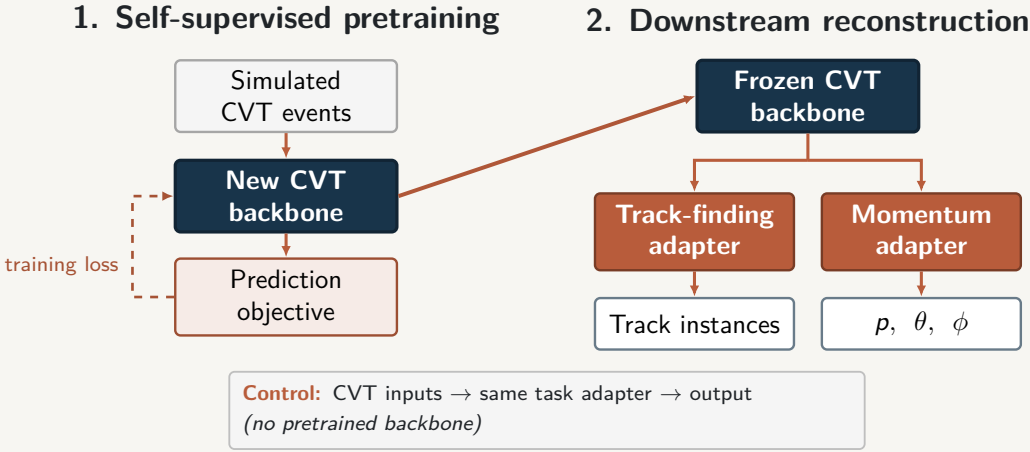
TPC: D. Park et al., arXiv:2508.14087v1, Sec. 3. CVT: S. J. Paul et al., NIM A 1049, 168032 (2023). CLAS12: supplied artwork.

Adapting FM4NPP to the CLAS12 CVT

Original FM4NPP strategy		This work: CLAS12 CVT
Self-supervised geometric pretraining	MODIFIED	Predict nearby cluster positions on later detector layers
Ordering detector measurements	REWORKED	Order clusters within each layer, then from inner to outer layers
Spacepoint coordinates as the input representation	EXPLORING	Continue testing hit representations, with strip inputs as a next step
Frozen backbone + lightweight downstream adapters	EXPANDED	Keep the backbone frozen and add a track-level momentum adapter

Methodological transfer, not checkpoint transfer:
the CLAS12 model is pretrained from scratch on CVT data.

Testing the adapted CVT backbone



Question: does pretraining improve reconstruction when labeled data are scarce?

Tokenize, serialize, predict

1. Tokenize

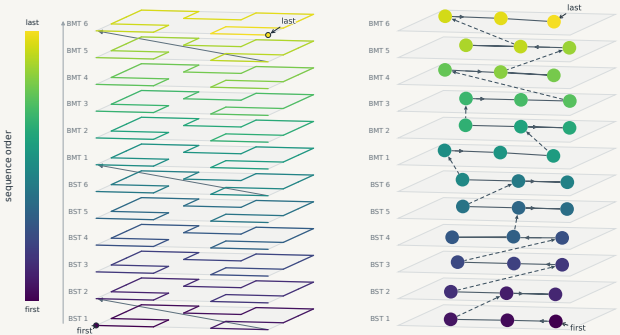
CVT cluster

↓

token $= (x, y, z)$

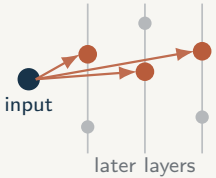
Each CVT cluster is represented by its 3D center.

2. Serialize



Hilbert within layers; inner → outer across layers.

3. Predict

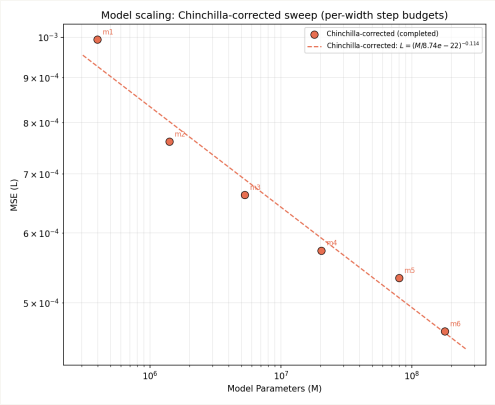


Modified kNNN prediction

Predict geometrically nearby measurements on later detector layers.

The backbone learns by predicting cluster positions on later layers.
Track and momentum labels are only used to train the task adapters.

Model and compute scaling in CVT pretraining



CVT pretraining comparison

Each point is a completed CVT pretraining run. m1–m6 are increasingly large backbones.

- **Horizontal:** model size, measured by the number of parameters.
- **Vertical:** mean squared prediction error (MSE). Lower is better.

Each model has its own training budget. The dashed line is an empirical fit to these six runs.

Why it matters: sparse CVT measurements support gains as models and compute scale up. We next test whether these features improve track finding.

Takeaways

- FM4NPP can be adapted to the sparse, layered CLAS12 CVT geometry.
- Self-supervised pretraining gives a clear benefit for track finding.
- Momentum pretraining improves low-label regression and resolution; high-label resolution gains remain limited.
- Current work continues to test hit representations and track-level adaptation, with strip inputs as a next step.
- The results are promising and we plan to extend this approach to the entire CLAS12 detector in future studies.

This work was supported by the U.S. Department of Energy, Office of Science, under Contract No. DE-FG02-04ER41309.

Thank you — questions?

Backup: track-finding definitions

Earlier DNP cluster-input experiment, 500,000 test events per run.

Clustering and matching

Signal ARI: adjusted Rand index over truth-signal clusters only.

Inclusive ARI: signal and background, with predicted background as one label.

Matched IoU: mean $|P \cap T|/|P \cup T|$ for accepted predicted/true track pairs.

Pooled track counts

Efficiency: accepted matches / all true tracks.

Purity: accepted matches / all predicted tracks.

Background rejection: true background clusters left unassigned / all true background clusters.

Historical evaluator: maximize total IoU with one-to-one Hungarian assignment, then require IoU, overlap/predicted size and overlap/truth size each ≥ 0.5.

The canonical earlier skim labels primary-origin groups with fewer than eight clusters as background.

Backup: momentum targets and comparison scope

Campaign 4 m6 versus AdapterOnly, as reported in the DNP source.

Item	Scope of the plotted experiment
Training labels	100, 1k, 10k or 100k truth-associated segments
Repeats	Seeds 11, 17 and 23; 500k evaluation tracks per run
Regression target	`MC::True` momentum at the inner observed hit
Component error	Mean of MAE(p_x), MAE(p_y), MAE(p_z), then mean $\pm$ seed SD
Resolution summary	Gaussian-core σ, averaged across eight true-p bins from 1 to 3 GeV

Truth membership removes association failures from the benchmark. Gaussian-core widths omit full residual tails.

COATJAVA coverage and trajectory-state conventions limit small-bias and end-to-end claims.

Backup: validation-selected noise operating points

Best validation checkpoints. The percentages below are held-out test cluster measures.

Rule / model	Threshold	Signal retained	Noise rejected
95% target, unweighted	0.838589	95.124%	95.306%
95% target, balanced	0.417096	95.123%	94.821%
99% target, unweighted	0.989786	98.992%	82.588%
99% target, balanced	0.926999	98.992%	83.263%
COATJAVA unassigned	Fixed rule	91.325%	97.258%

Reject a cluster only if its noise score is strictly greater than the threshold.

One matched seed. The common-final-update comparison preserves the 95% / 99% tradeoff.

Backup: independent neural tracking benchmarks

CLAS12 adaptations described in the November 9 implementation audit.

EggNet

Evolving graph attention produces an embedding. DBSCAN converts embeddings to track instances.

Exa.TrkX

Embedding and filtering build a graph. An Interaction Network scores edges, then connected components define tracks.

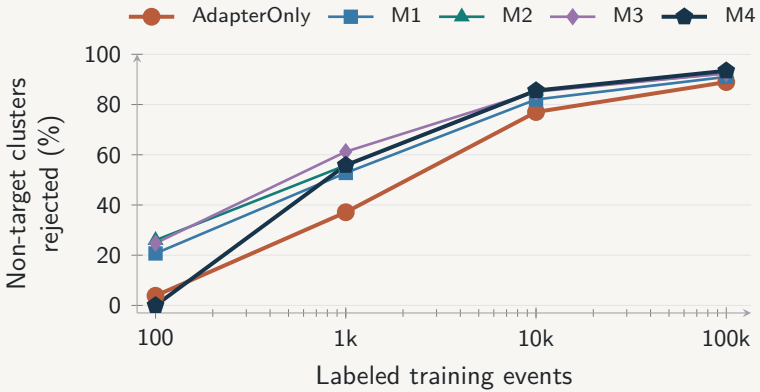
A numerical ranking needs a common comparison

Matching event and truth populations, identical eligibility and evaluator logic, validation-selected operating points, and held-out test reporting.

The inspected archived runs and newer FM evaluations do not establish that full comparison.

Equal numerical thresholds do not make different matching algorithms equivalent.

Noise tagging: label efficiency



At 1k labeled events

53–61%

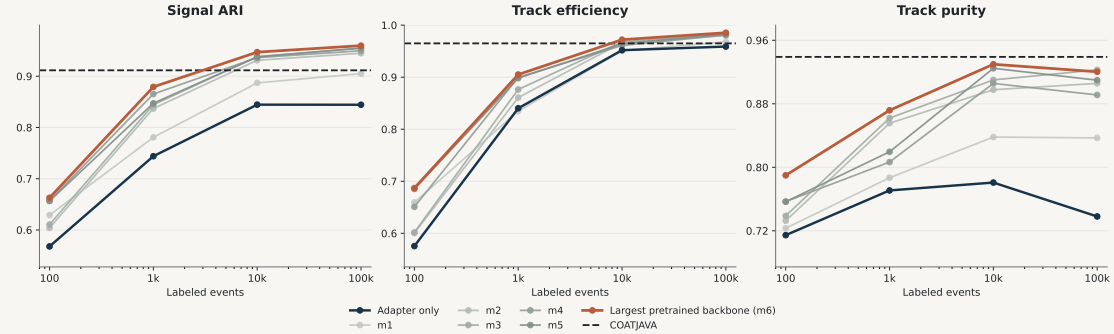
rejection with pretraining

versus **37%** for AdapterOnly

99% validation-retention target; test results shown. M4 at 100 events retains all clusters.

Pretraining strongly improves CLAS12 track finding

Earlier cluster-input experiment, Campaign 4. Original m1–m6 backbone labels.



Pretraining effect

Frozen CVT features improve all three headline metrics across the label sweep.

Conventional reconstruction

At 100k labels, m6 exceeds COATJAVA in ARI and efficiency. Track purity remains lower.

Model-size dependence

Larger backbones tend to perform best, but the ordering is not strictly monotonic.

Track-finding metric definitions

T_i: eligible truth-track clusters; P_j: all assigned prediction clusters; $n_{ij} = |T_i \cap P_j|$.
N_T counts injected-primary tracks with ≥ 4 owned clusters. Event-local matches; pooled test-set counts.

Track efficiency

$$\epsilon_{\text{track}}(q) = \frac{|M(q)|}{N_T}$$

Fraction of eligible truth tracks recovered by an accepted one-to-one match.

Cluster purity

$$\mathcal{P} = \frac{\sum_{(i,j)\in A} n_{ij}}{\sum_{(i,j)\in A} |P_j|}$$

Fraction of correctly assigned clusters in MC-associated predictions.

Cluster completeness

$$\mathcal{C} = \frac{\sum_{(i,j)\in A} n_{ij}}{\sum_{i=1}^{N_T} |T_i|}$$

Fraction of all eligible truth clusters recovered in their associated predictions.

Efficiency matching $M(q)$

Maximize the one-to-one match count, then total IoU. Each pair must satisfy

$$\frac{n_{ij}}{|T_i| + |P_j| - n_{ij}} \geq 0.5, \quad \frac{n_{ij}}{|P_j|} \geq q, \quad \frac{n_{ij}}{|T_i|} \geq q.$$

$q = 0.5$ on slide 7; $q = 0.50, 0.55, \ldots, 1$ on slide 8.

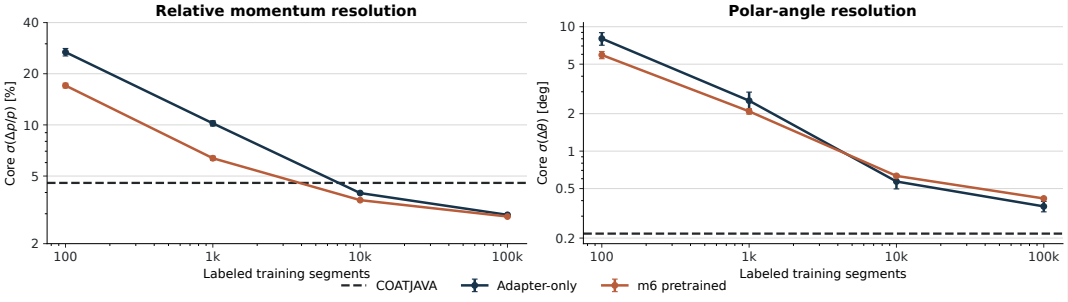
Independent cluster association A

Maximize total Dice with one-to-one pairs and positive overlap. No quality cuts are applied:

$$D_{ij} = \frac{2n_{ij}}{|T_i| + |P_j|}, \qquad n_{ij} > 0.$$

Purity excludes unassociated predictions; all eligible truth clusters enter the completeness denominator.

Resolution gains also fade as labels increase



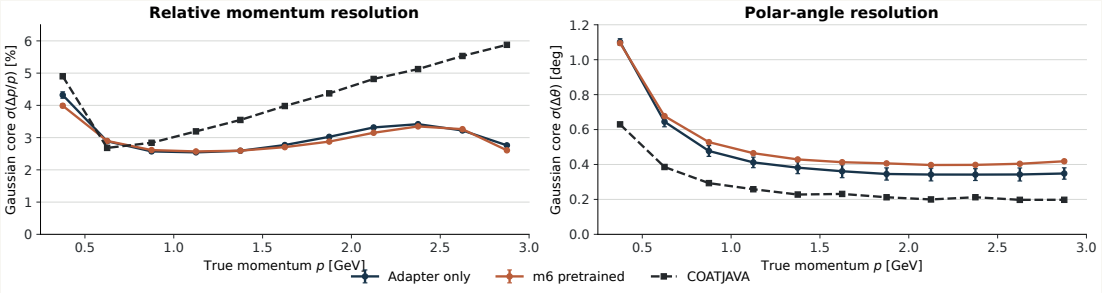
Gaussian-core widths averaged across 1–3 GeV true-p bins. Error bars show seed SD.

- At 100–1k labels, m6 narrows both cores relative to AdapterOnly.
- At 100k, momentum cores nearly coincide. m6's polar-angle core is slightly broader.
- COATJAVA retains narrower angular cores.

Earlier truth-associated cluster experiment. Core widths exclude tracking inefficiency and full residual tails.

Resolution versus true momentum at 100k labels

100k labeled segments in the same DNP campaign. Mean fitted widths with seed SD.



Momentum magnitude

The two learned curves nearly coincide through much of the displayed range.

Polar angle

The gap to COATJAVA persists at high momentum.

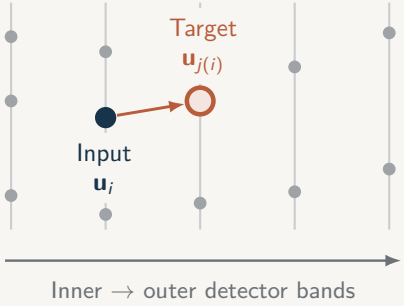
Conventional reconstruction coverage and trajectory-state differences limit end-to-end interpretations.

Pretraining: learn from CVT geometry

The training targets come from cluster positions, without track or momentum labels.

A geometric prediction target

Schematic detector bands



Target chosen by geometry;
no same-particle association is required.

Order the cluster centers

Normalize (η, ϕ, r); use Hilbert ordering within detector bands, then inner to outer.

Predict an outer-band neighbor

At each token, predict the nearest cluster in a higher band ($k = 1$), using the ordered context.

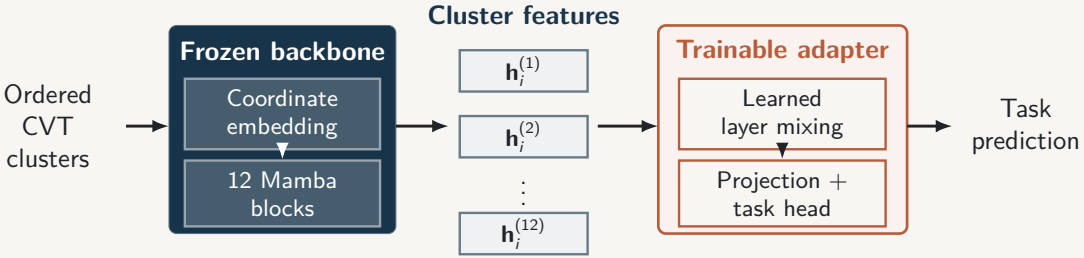
Train the backbone + prediction head

Minimize coordinate error; omit targets with no eligible outer-band neighbor.

$$\mathcal{L}_{\text{pre}} = \frac{1}{3N_{\text{valid}}} \sum_{i \in \text{valid}} \left\| \widehat{\mathbf{u}}_i - \mathbf{u}_{j(i)} \right\|^2$$

What is the frozen CVT backbone?

A pretrained feature extractor: one vector per cluster, at each depth.



Fixed weights, event-dependent features

New inputs produce new features;
the backbone receives no task updates.

MC truth trains each task adapter

Adapters learn from MC labels.
Each task is trained separately.

The pretraining prediction head is unused downstream.

Backbones used in this talk

Earlier Mamba3 campaign: 12 blocks, state size 16, $k = 1$.

Model	Feature width	Backbone parameters	Pretraining budget
m1	64	0.333 million	100k
m2	128	1.266 million	100k
m3	256	4.927 million	100k
m4	512	19.438 million	200k

Budgets in optimizer updates. Mike's later $k = 1$ sweep uses state size 128.

Tagging and tracking

m1–m4: simulation + measured background

Momentum results

Historical m6: simulation without overlay

The size comparison also varies pretraining budget: m4 receives twice as many updates.