FM4NPP: Self-Supervised Pretraining for CLAS12 Tracking

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

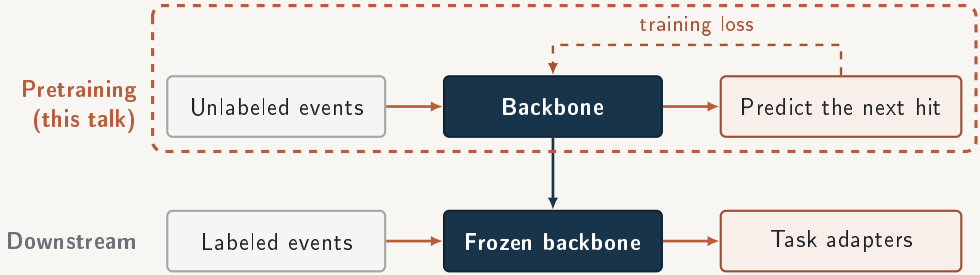
[1] University of Connecticut [2] Brookhaven National Laboratory

CLAS Collaboration Meeting

November 2026

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

FM4NPP and the Scope of This Talk

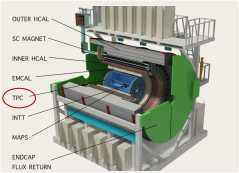


FM4NPP as published

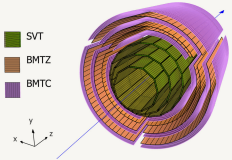
Detector	sPHENIX TPC, 3D spacepoints
Pretraining data	>11M simulated events
Backbone	Mamba state space model, up to 188M parameters
Objective	predict the k next nearest neighbours of every point

D. Park et al., arXiv:2508.14087 (2025).

From the TPC to the CVT



sPHENIX TPC



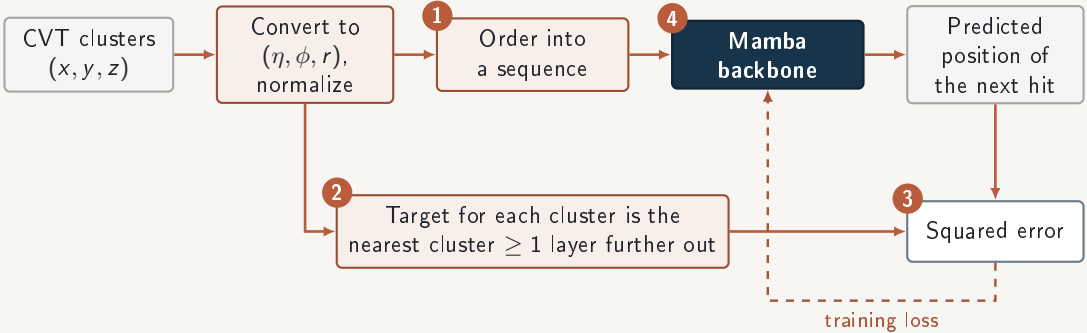
CLAS12 CVT

	sPHENIX TPC (FM4NPP)	CLAS12 CVT (this work)
Measurement	3D spacepoint	strip cluster
Radial sampling	nearly continuous	12 discrete layers
Hits per track	tens	at most 12, one per layer
Hits per event	10^2–10^3	tens
Sequence order	raster scan of a 3D grid, sorted by r inside a cell	by layer, then a Hilbert curve inside the layer
Prediction target	nearest points at larger r	nearest cluster at least one layer further out

Same pretraining task. Sequence order and prediction target are redefined for the CLAS12 detector

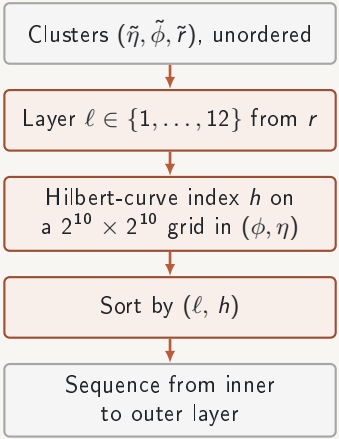
TPC image from D. Park and S. Li, NYSDS 2025. CVT image adapted from S. J. Paul et al., NIM A 1049, 168032 (2023).

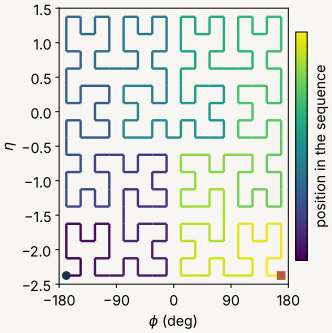
Pretraining Pipeline for the CVT



Inputs and targets contain no truth information, only clusters

1 Ordering the Clusters into a Sequence





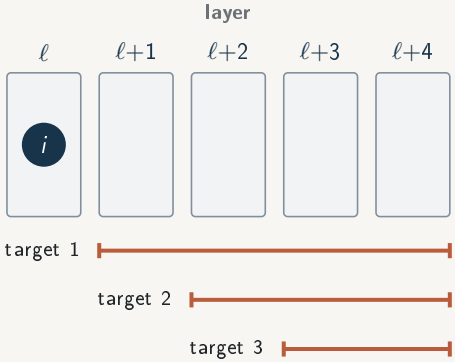
The curve within one layer at reduced resolution

Order is needed
An event is an unordered set of clusters. The same rule orders every event to ensure similar events give similar predictions.

Inner layers first
The model must only see earlier clusters, so next hit of any track must come later in the sequence.

Hilbert curve inside each layer
A layer has no intrinsic radial order. Hilbert ordering is chosen so clusters close in (ϕ, η) generally stay close in the sequence.

2, 3 The k Next Nearest Neighbours Task (kNNN)



Origin i: the cluster being predicted from
Target m: the nearest cluster to i in target range

Targets of cluster i, $m = 1, \ldots, k$

$$t_m(i) = \underset{j\,:\,\ell_j \geq \ell_i + m}{\arg\min} \left\| \tilde{\mathrm{x}}_j - \tilde{\mathrm{x}}_i \right\|$$

Loss

$$\mathcal{L} = \left\langle \frac{1}{3} \left[(\hat{\eta} - \tilde{\eta}_t)^2 + (\hat{r} - \tilde{r}_t)^2 + \frac{(\hat{s} - \sin\phi_t)^2 + (\hat{c} - \cos\phi_t)^2}{(2\pi)^2} \right] \right\rangle$$

- $\tilde{\mathrm{x}} = (\tilde{\eta}, \tilde{\phi}, \tilde{r})$ is the position scaled to $[0, 1]$
- k predicted positions per cluster averaged over all clusters and targets in the event
- FM4NPP uses $k > 1$ in the TPC. Most CVT studies here use $k = 1$

4 Model Sizes, Training Budget, Hyperparameters

Model	Width	Param. N	Clusters D	Steps
m1	64	0.54M	21M	3.2k
m2	128	1.7M	67M	10k
m3	256	5.7M	229M	35k
m4	512	21M	841M	127k
m5	1024	80M	3.2B	485k
m6	1536	180M	7.2B	1.1M

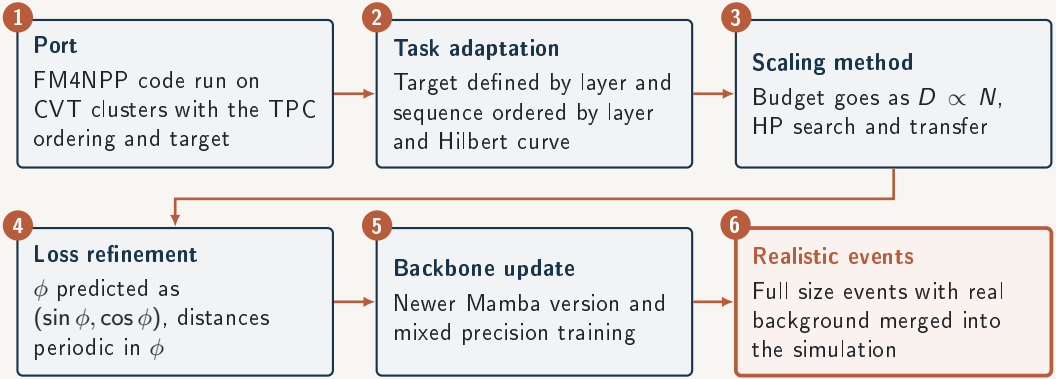
Training budget ("Chinchilla" rule)
Data grows with the model at $D \approx 20\,N$ clusters seen. A fixed number of steps over trains small models and under trains large ones.

Hyperparameter search
Learning rate, batch size, warmup, weight decay and gradient clipping, searched at m3 only.

Transfer across widths (μP)
In a wider network the same update changes the output more. The learning rate is divided by (width / 256), so the m3 settings stay valid at every width. This was confirmed by HP searching other model sizes and achieving similar rates as predicted.

Budget rule from J. Hoffmann et al., arXiv:2203.15556 (2022). μP from G. Yang et al., arXiv:2203.03466 (2022).

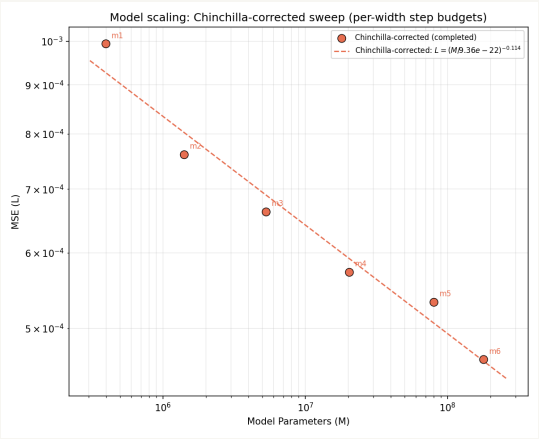
Development of the Methods



Problems Encountered

Problem	Test	Finding	Outcome
Largest model stopped improving	Learning rate and training length across widths	Far too little training	**Chinchilla budget adopted**
Order inside a layer looked arbitrary	Clusters per cell of the Hilbert grid	Grid too coarse	**Grid rescaled**
Loss barely improves with size once background is added	Loss split by cluster type	Background dominates the average	**Signal loss reported**
Is reused background memorized?	Loss on seen against unseen background	No difference	**Reuse accepted**
Does predicting several layers ahead help?	$k = 5$ against $k = 1$ at equal budget	Small models much worse	$k = 1$ **kept**

Larger Models Predict the Next Hit Better



What is plotted

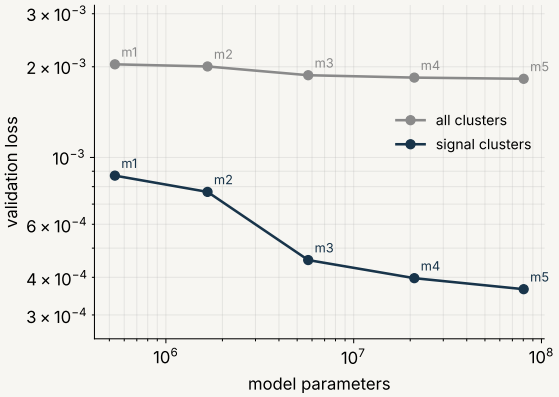
One point per model size, m1 to m6, each trained on its own budget.

The vertical axis is the error of the predicted next hit position on events not used in training set.

Every increase in size lowers the error, with no sign of levelling off at 180M parameters.

Numerical values depend on how the loss is defined, the trend is most important

Current Study with Realistic Background



Test

Real background merged into the simulated events with ∼140 clusters per event, most of them background. Five sizes retrained, m1 to m5 (m6 is very expensive to run).

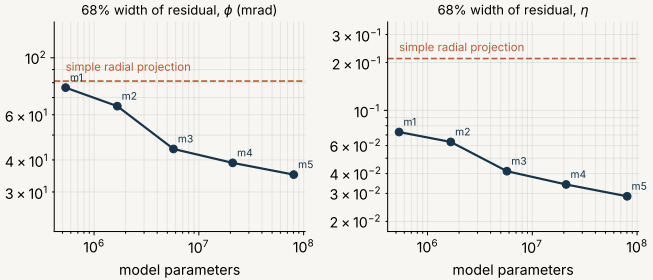
Result

Averaged over all clusters the loss falls by only 10% from m1 to m5. This comes from mostly unpredictable background

For loss from track signal clusters it falls by **58%**

The separation uses simulation truth and is done after training. Training uses no labels.

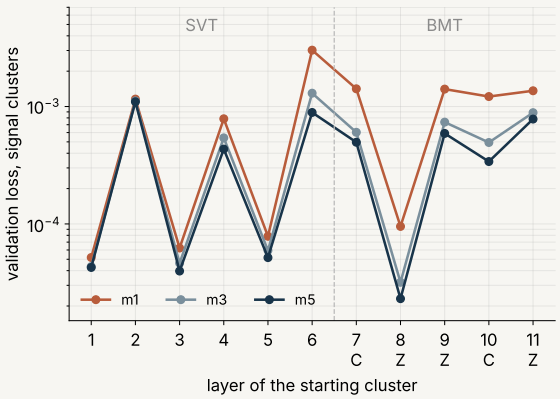
Next Hit Prediction in Detector Units



68% width	ϕ (mrad)	η
Radial projection	81	0.21
m1 (0.5M)	77	0.073
m5 (80M)	35	0.029

- Predicted minus true position of the next hit, only for signal clusters whose next hit is on the same particle
- Simple radial projection, same ϕ and η as the current cluster, moved out to the next layer
- The largest model is 2 to 3 times closer than the smallest, in both coordinates.

Where Model Size Helps



Low loss predictions

The next hit is fixed by the current cluster, the partner layer of an SVT pair, 3.5 mm away (1, 3, 5), or the next BMT layer of the same strip type (8). Every model size does well

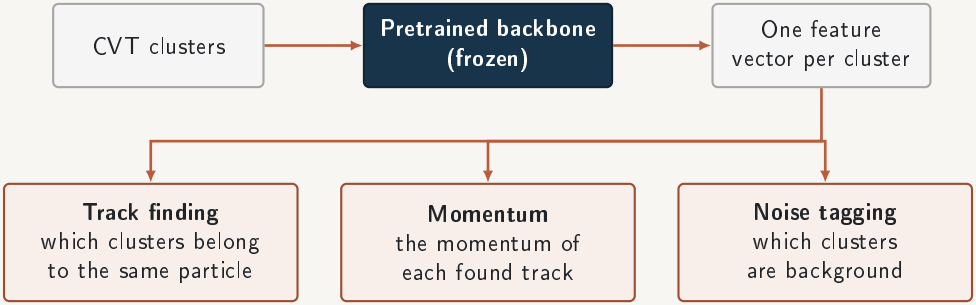
High loss predictions

If the next layer measures a coordinate the current cluster does not have (BMT C measures z, Z measures ϕ), or lies across a gap the prediction is more difficult

Take Away

As more context is given to the model, larger models can more accurately predict the next cluster

What the Pretrained Model Is For

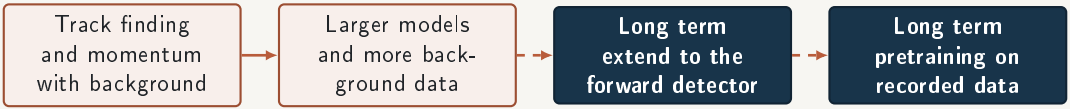


- Each task gets a small network (an adapter) trained with labels on top of the same backbone
- The backbone is trained once, without labels, and is not changed afterwards
- The pretraining is useful to the extent that its features make these tasks easier and fewer labeled events are needed for comparable performance

Summary and Outlook

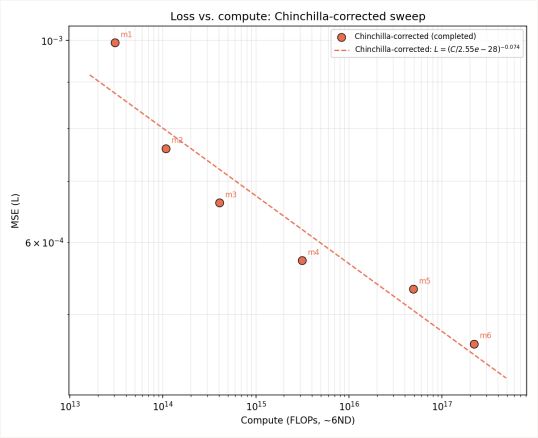
- FM4NPP's self-supervised pretraining has been adapted from the sPHENIX TPC to the CLAS12 CVT.
- A detector with 12 discrete layers needed a new sequence order and a target defined by layer.
- Models from 0.5M to 180M parameters were trained with a common method. The next hit prediction improves with every increase in size.
- With realistic background the prediction for signal tracks still improves with model size, with 58% lower error at 80M than at 0.5M parameters.

OUTLOOK



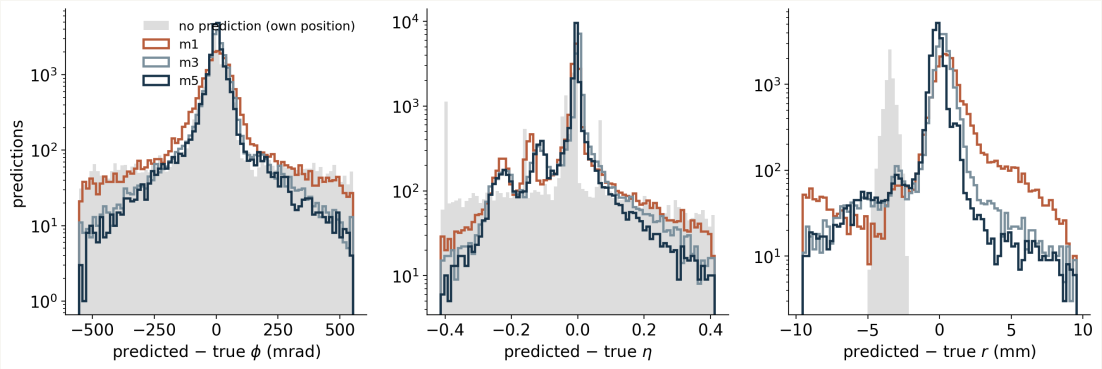
Backup

Next Hit Error Against Training Compute



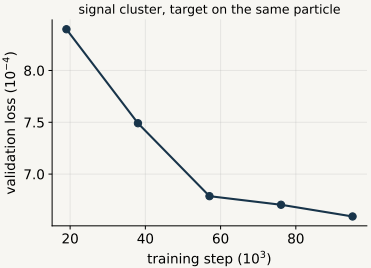
- One point per model size, m1 to m6, at the end of its training.
- Compute $\approx 6\,ND$ FLOPs (D is the number of clusters seen).
- Same runs as the main result, plotted against compute instead of parameters.

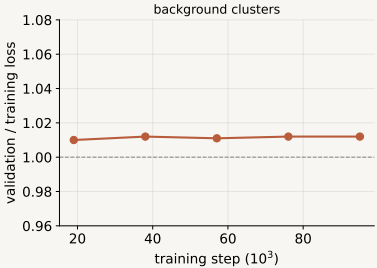
Residual Distributions



Predicted minus true next hit position for signal clusters with the target on the same particle. Grey is the cluster's own position.

Reused Background Is Not Memorized



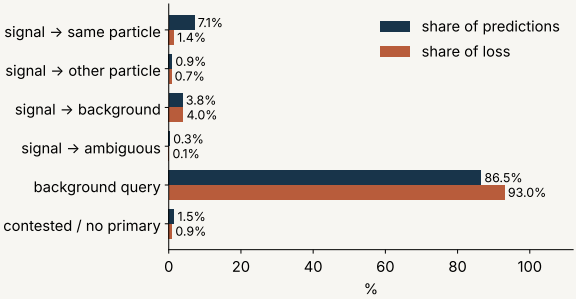


Left The error on signal clusters whose target is on the same particle keeps falling during training.

Right Background is reused between training events. The model does no better on background it has seen than on new background.

Preliminary. One model size (21M parameters), five checkpoints. Truth information is used for evaluation only.

Loss Composition in the Background Test



Training step (10^3)	19	38	57	76	95
Signal → same particle, loss (10^{-4})	8.40	7.49	6.79	6.70	6.59
Background, validation / training	1.010	1.012	1.011	1.012	1.012

m4, five targets per cluster, end of training (chart). Categories from simulation truth, used for evaluation only.