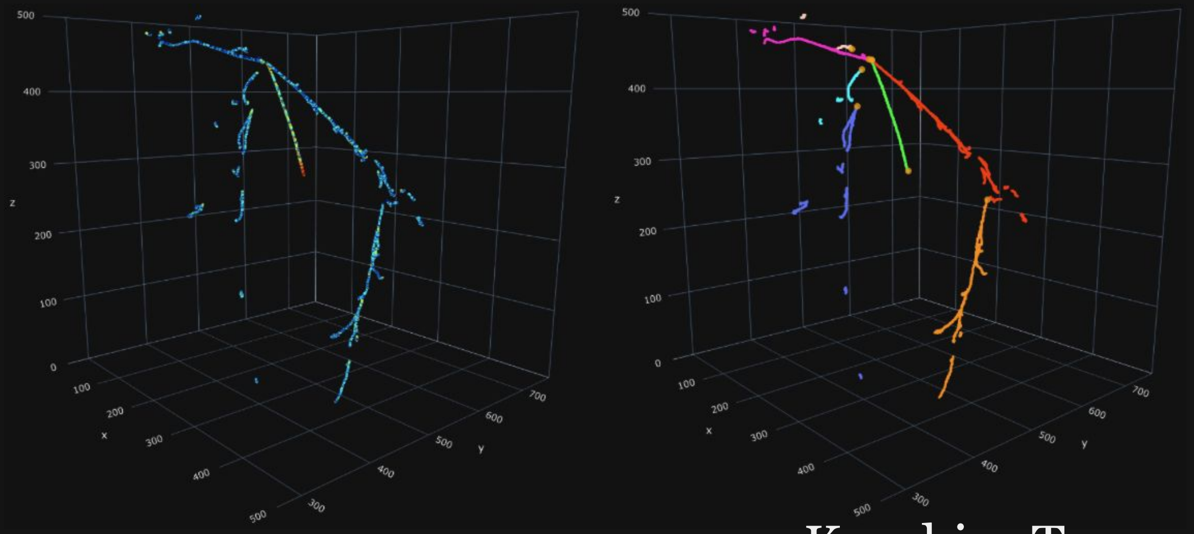
Machine Learning in Particle Physics

Original image credit: xkcd



Kazuhiro Terao
SLAC National Accelerator Laboratory

COSSURF @ SD - 2022

Outline

1. Neural Networks for Data Reconstruction
2. Likelihood-free Inference for Physics Modeling
3. Representation Learning by Foundation Models
4. Summary

Data Reconstruction in Experimental Particle Physics

Big, Monolithic Neutrino Detectors

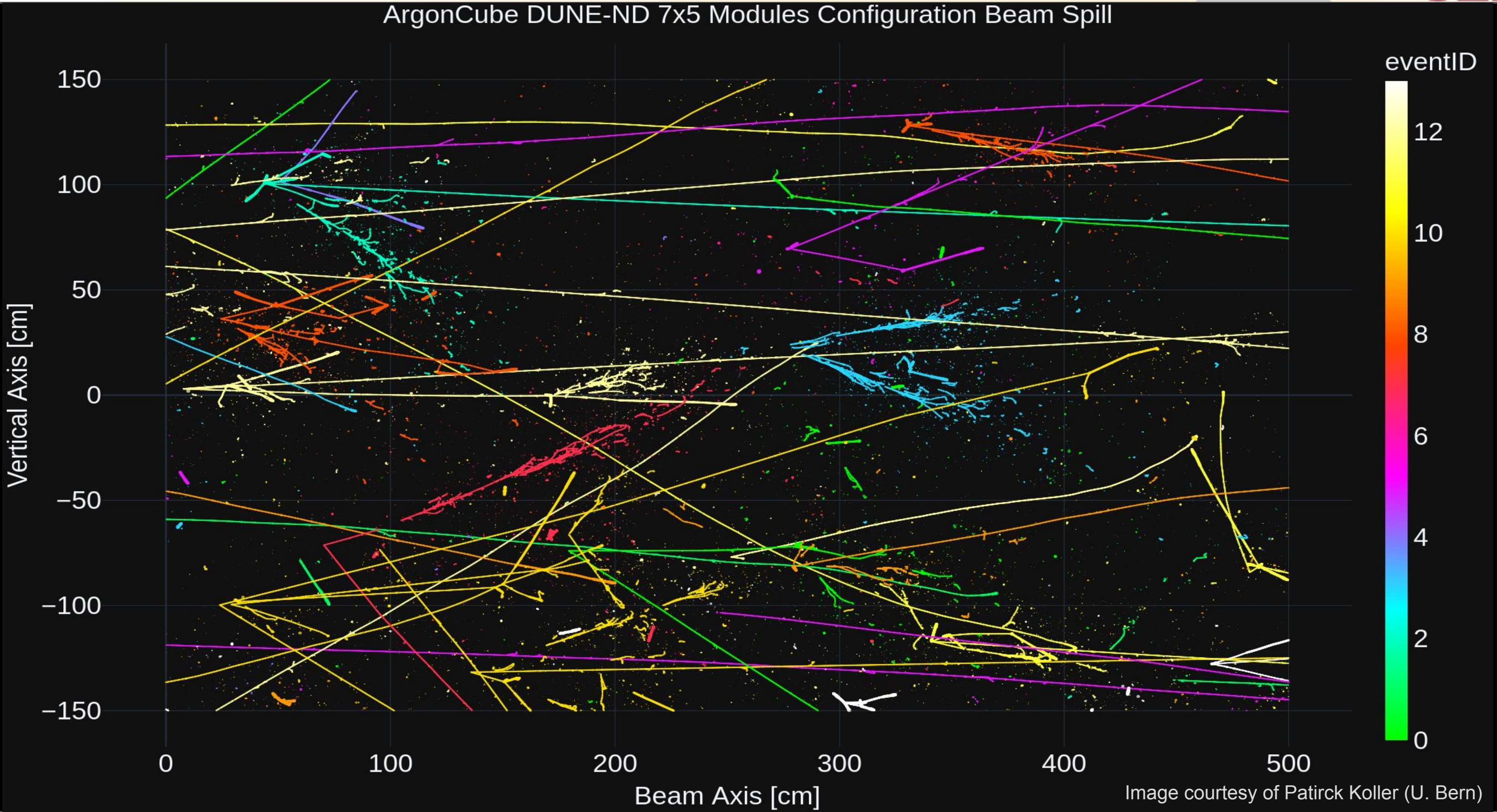


Image courtesy of Patirck Koller (U. Bern)

Data Reconstruction in Experimental Particle Physics

Multi-modal Collider Detectors

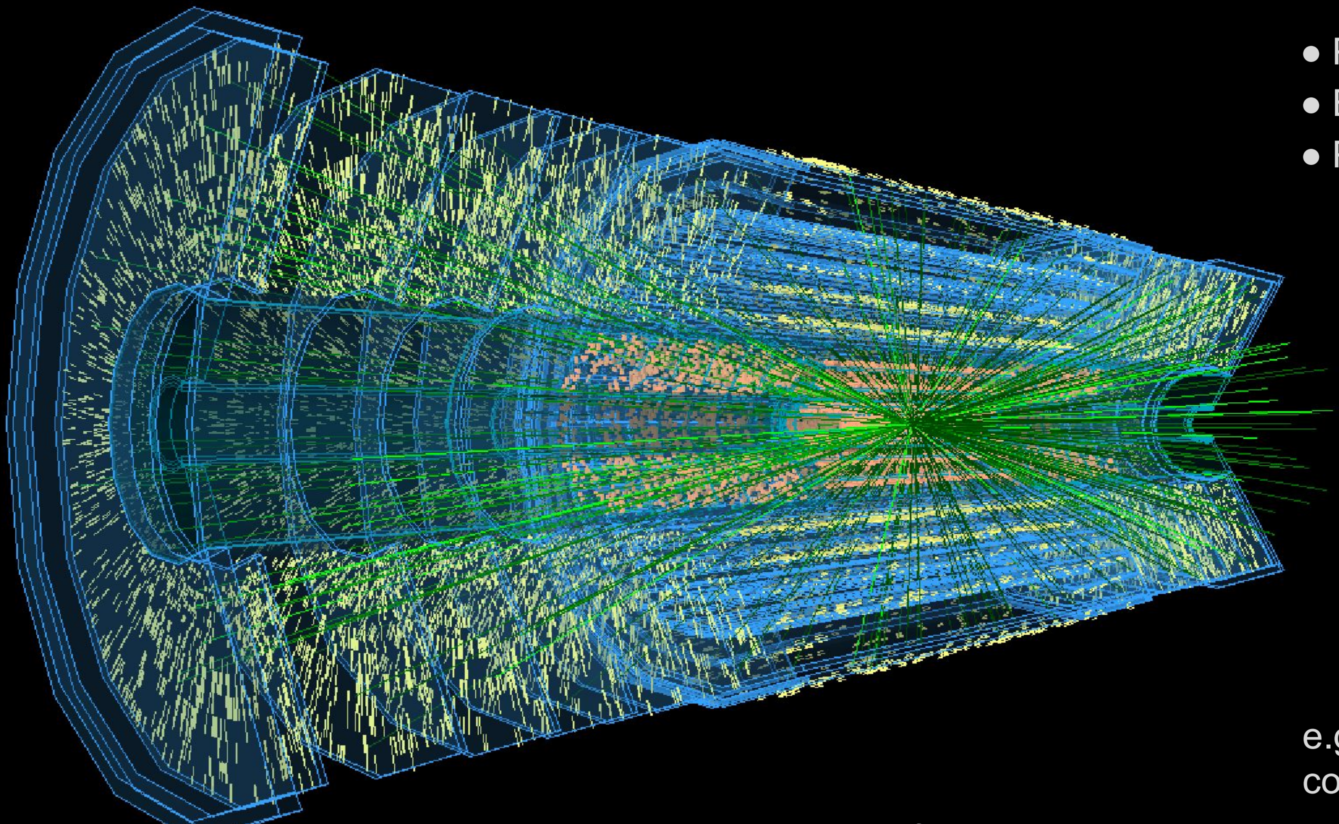
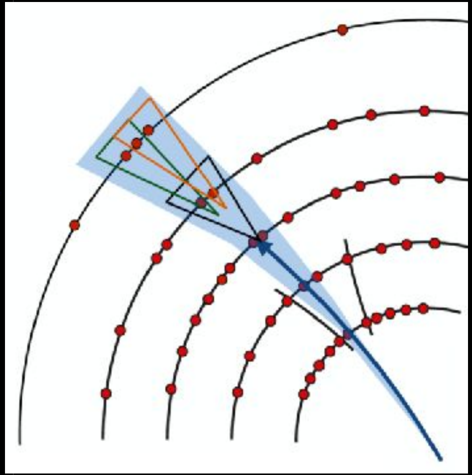


Image courtesy of Exa.Trk. collaboration

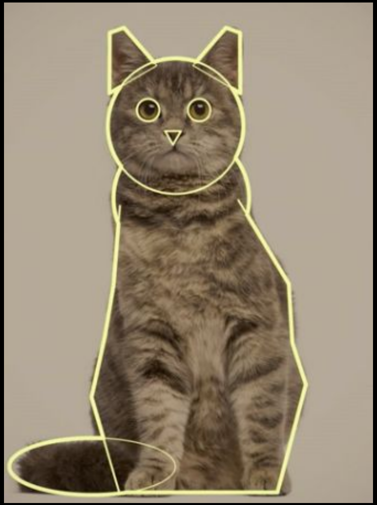
- Particle tracking (tracker)
- Energy clustering (calorimeter)
- Particle flow

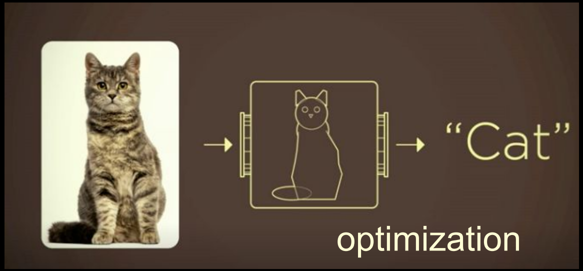


e.g.) Tracking = finding the right combination of sampled points

Primary goals in my view:

- Fast, accurate, and precise
- Automation of algorithm tuning (optimization)
- … and more (re-usability, scalability, extensibility, etc.)

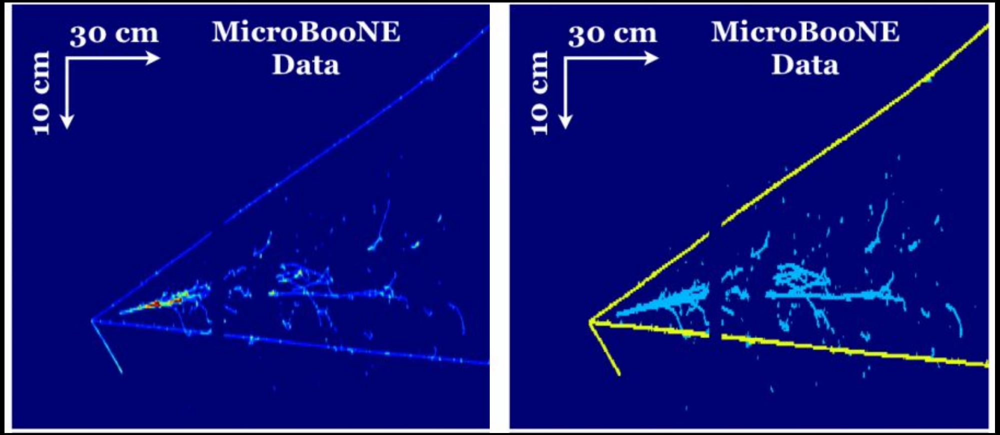


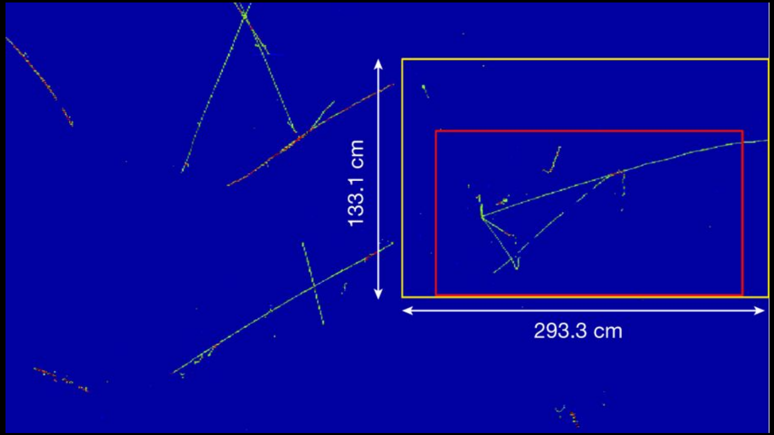


… cat?

Convolutional neural network (CNN)

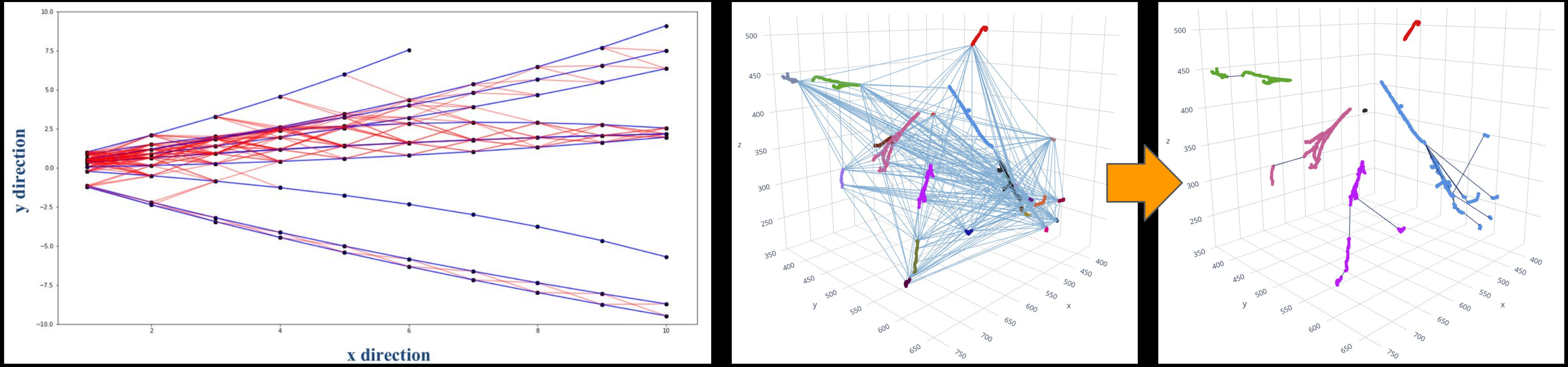
- Primarily aimed at image data
- Learns spatially local features of various size
- Translation invariant (target feature can be anywhere in image)
- Image/Pixel level classification/regression, object detection





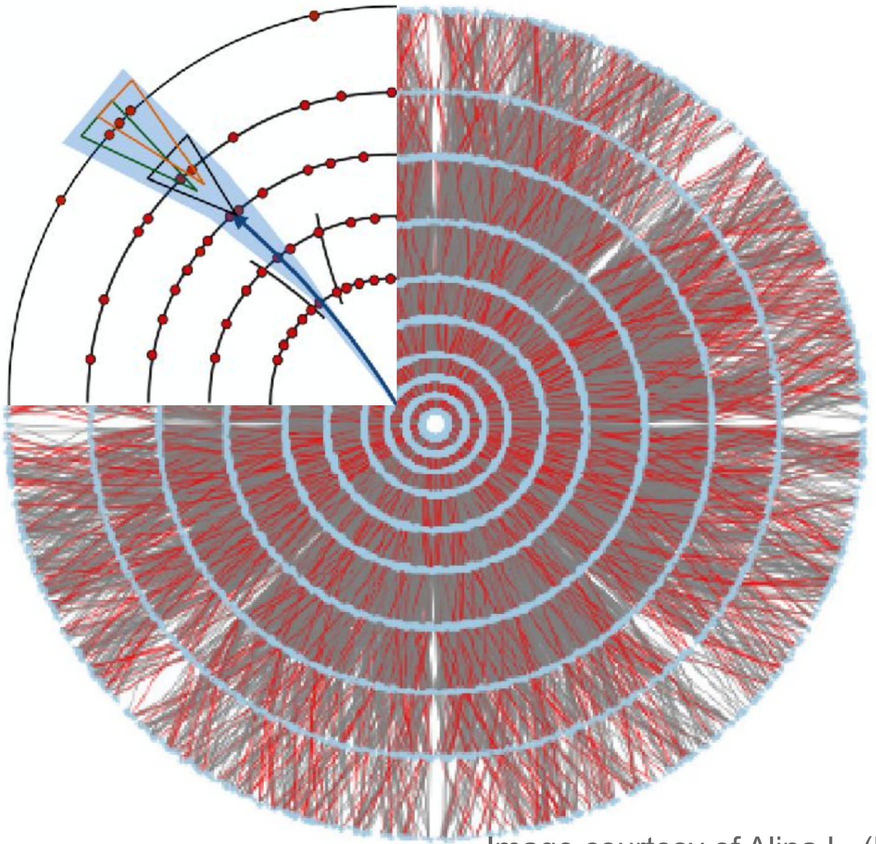
Graph neural network (GNN)

- Primarily aimed at relational data
- Learns relations between “nodes” connected by “edges”
- Can be permutation invariant
- Node, edge, or a (sub/whole) graph level classification and regression



Data Reconstruction in Experimental Particle Physics

Tracking @ Colliders



Charged particles sampled over ~10 layers.

Find a track = figure out combination of points.

Tracking @ HL-LHC = E5 per second!

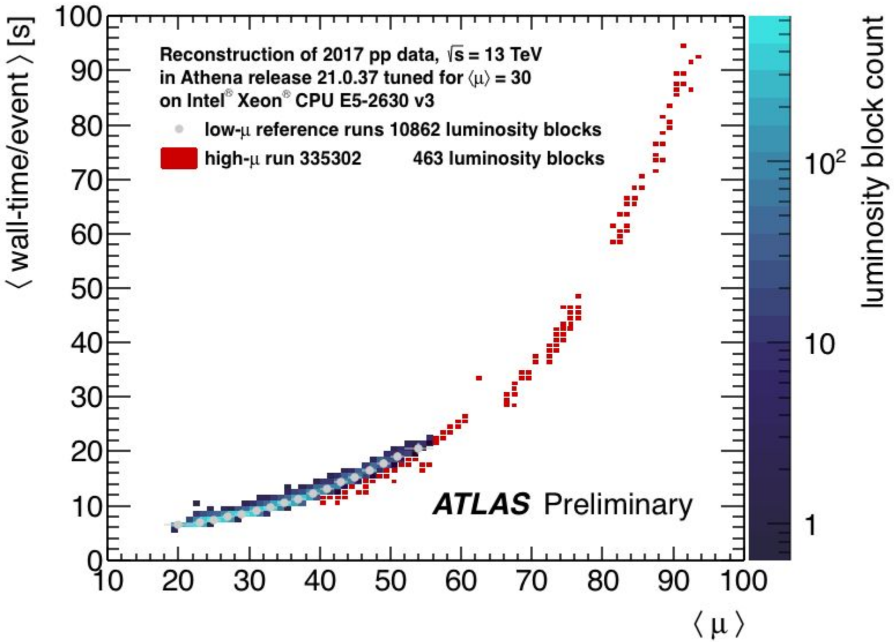
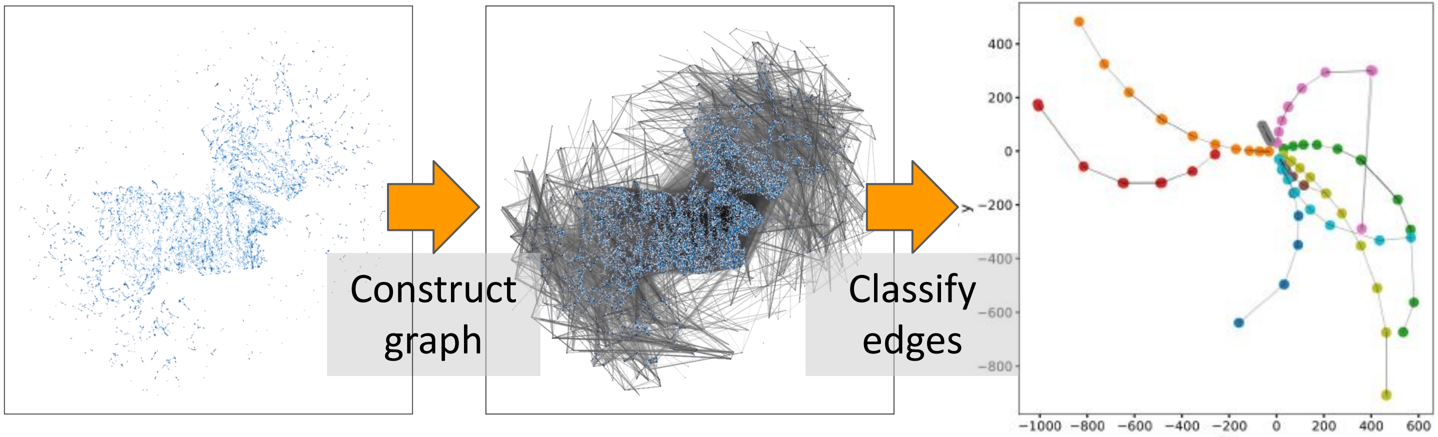


Image courtesy of Alina L. (Exa.Trk. collab.) @ ACAT2021

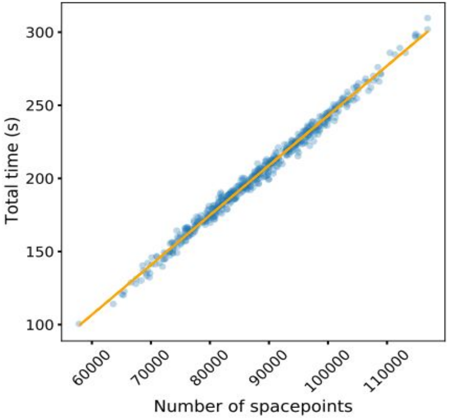
Data Reconstruction in Experimental Particle Physics
Tracking @ Colliders

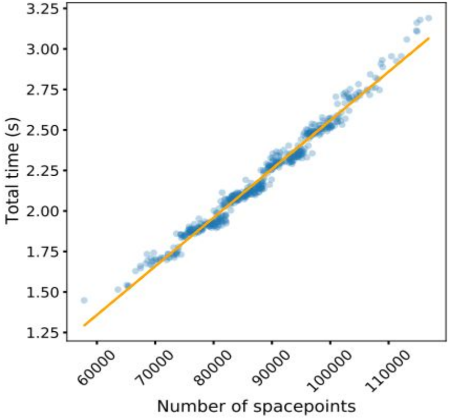
GNN for
scalable particle
tracking



Approximately linear scaling
with respect to the number
of input point

(HL-LHC by Exa.Trk.)



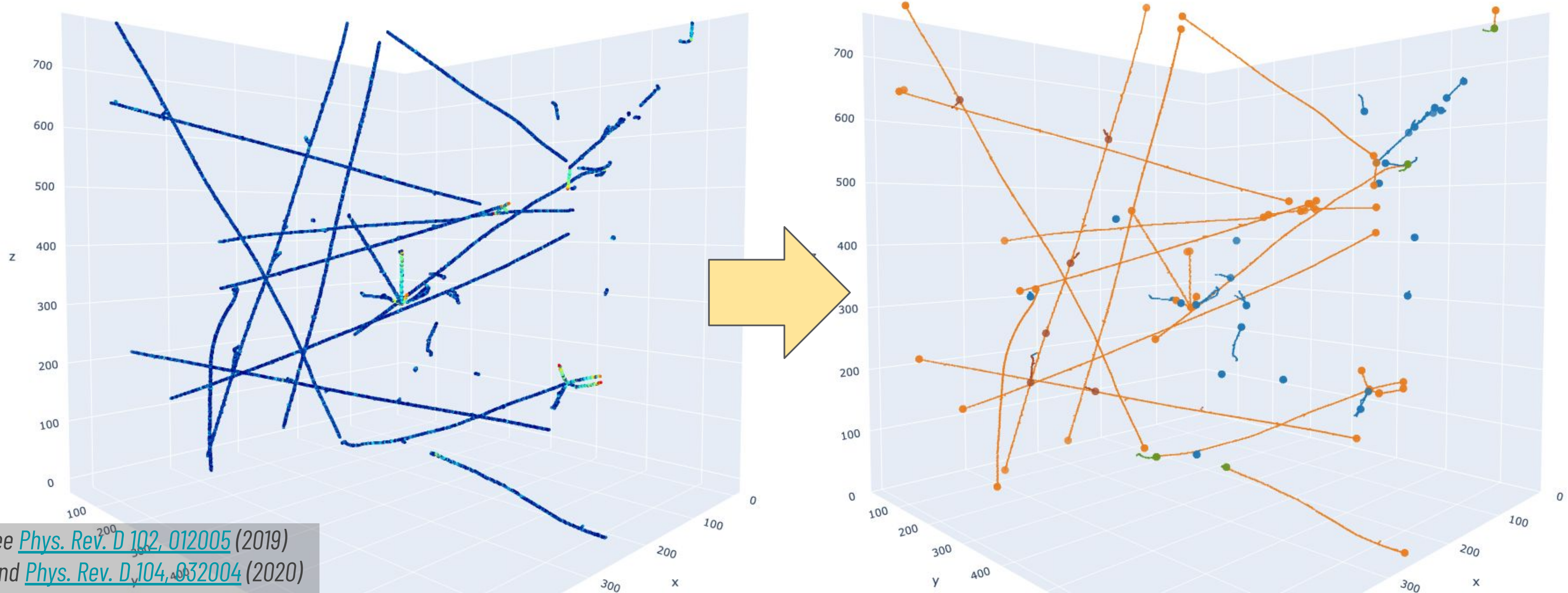


The European Physical Journal C, 81(10), pp.1-14.

Data Reconstruction in Experimental Particle Physics
Tracking/Clustering @ Calorimetric Neutrino Detector

SLAC

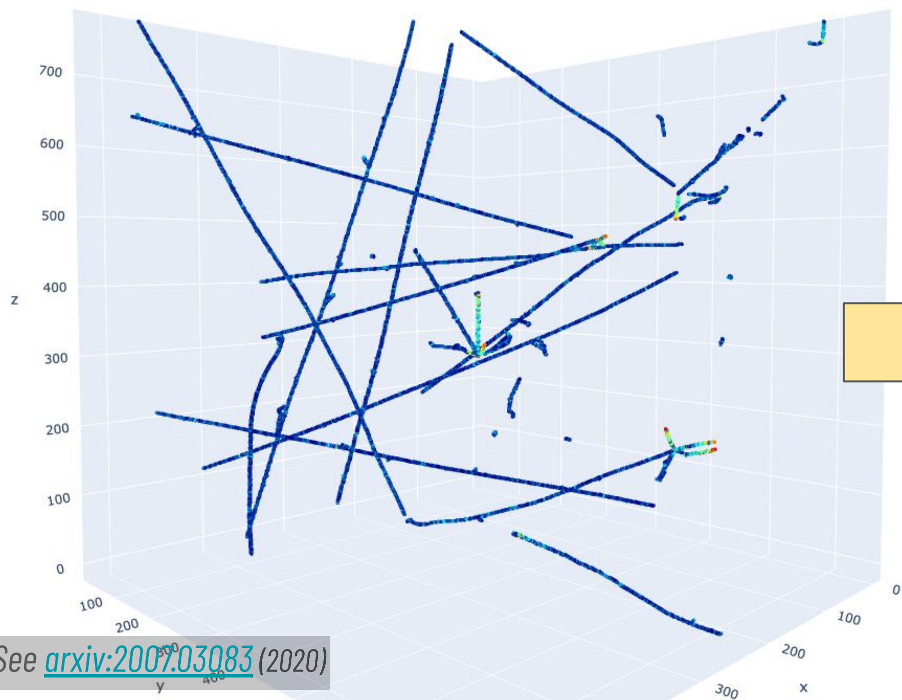
CNN for pixel-level classification and key point detection
(DeepLearnPhysics for DUNE)

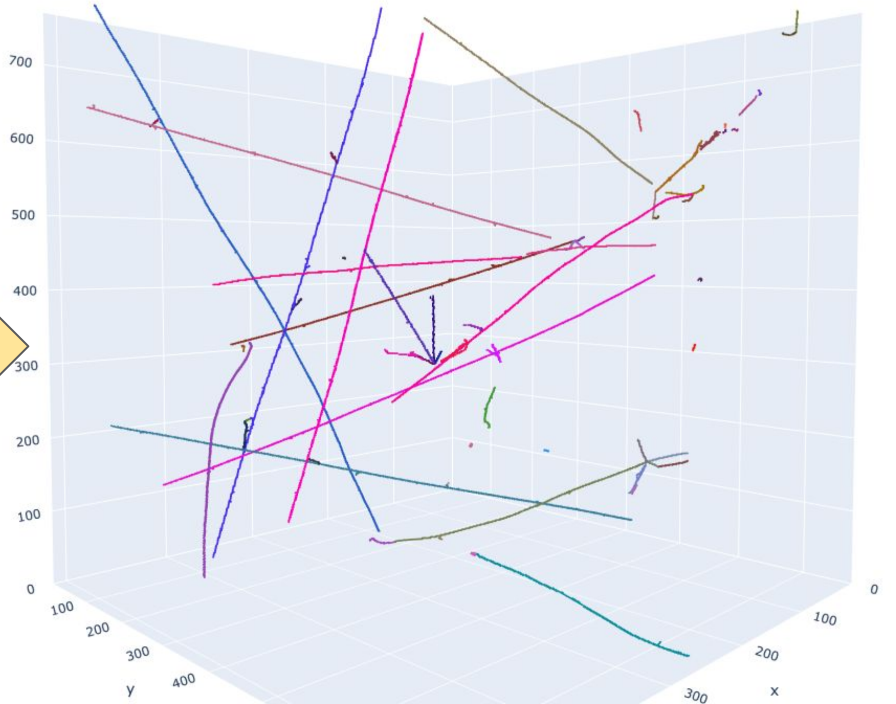


See Phys. Rev. D 102, 012005 (2019)
and Phys. Rev. D 104, 032004 (2020)

Data Reconstruction in Experimental Particle Physics
Tracking/Clustering @ Calorimetric Neutrino Detector

CNN for pixel-level regression dense clustering
(DeepLearnPhysics for DUNE)



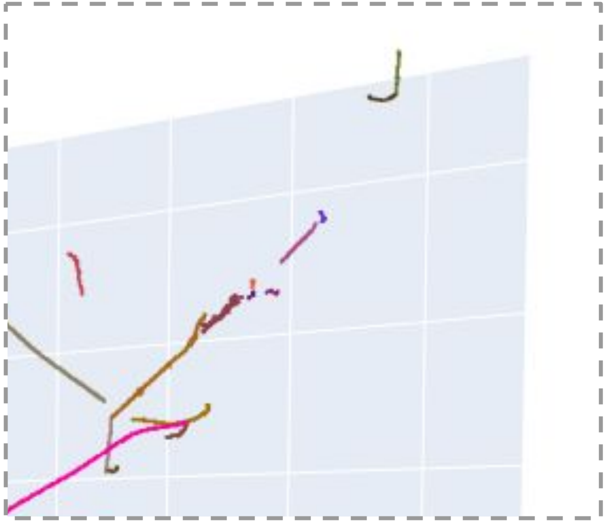


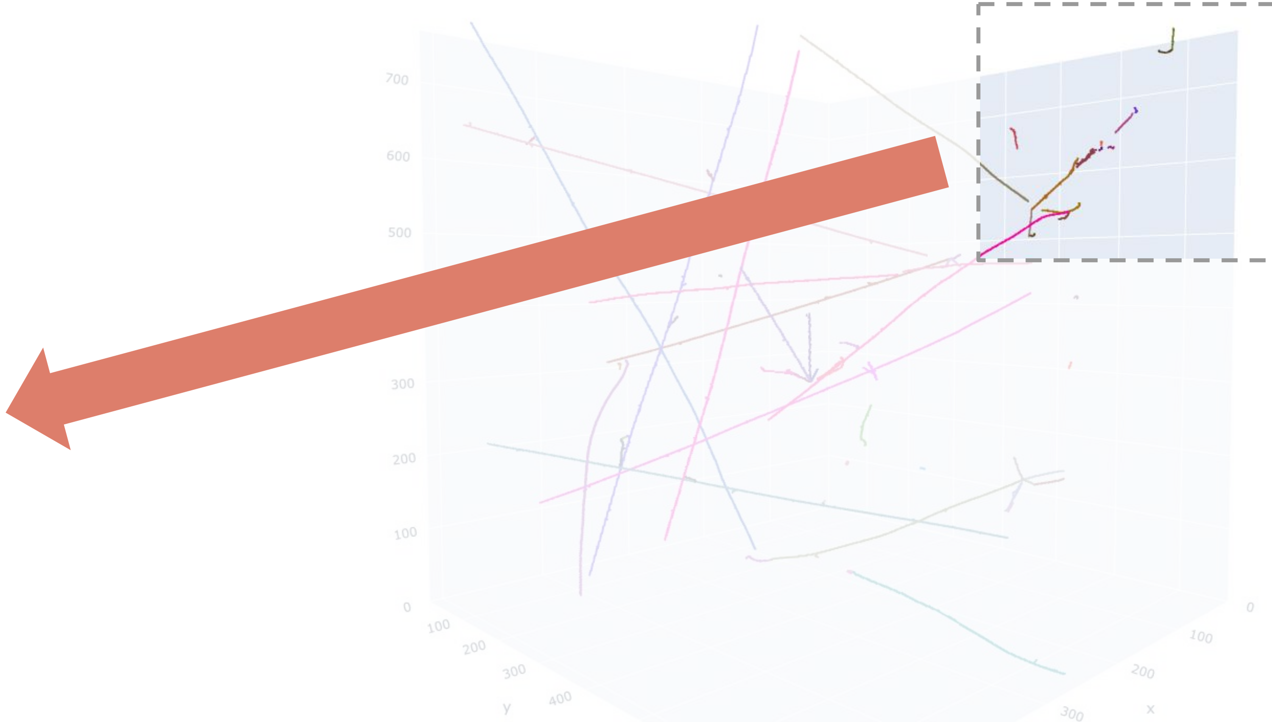
See arxiv:2007.03083 (2020)

Tracking/Clustering @ Calorimetric Neutrino Detector

SLAC

CNN for pixel-level regression dense clustering
(DeepLearnPhysics for DUNE)



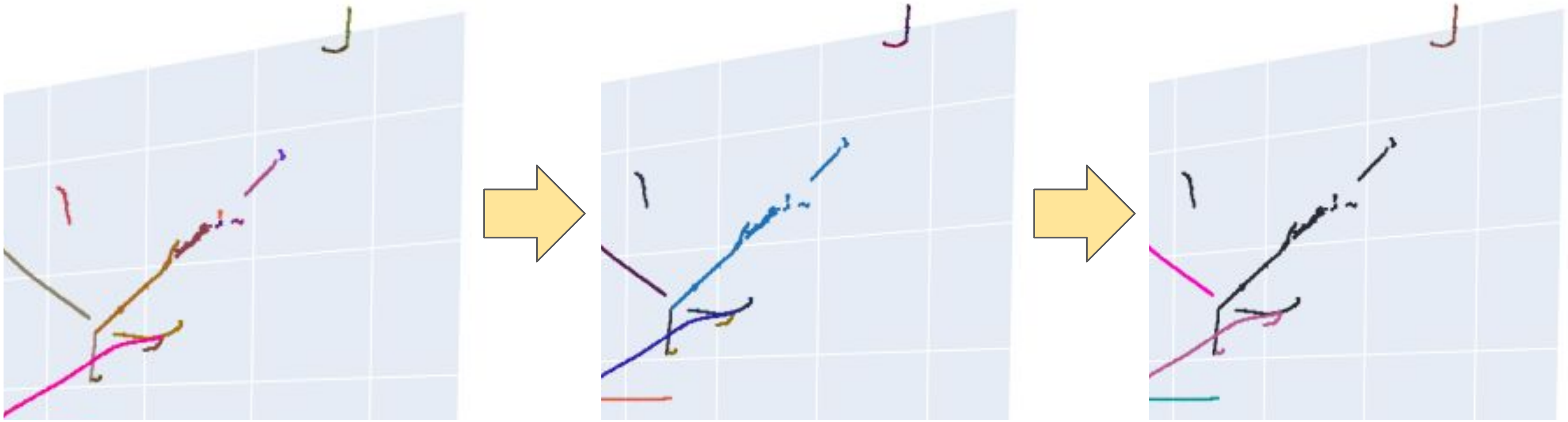


Tracking/Clustering @ Calorimetric Neutrino Detector

SLAC

GNN clustering at two levels: individual particle and interaction (DeepLearnPhysics for DUNE)

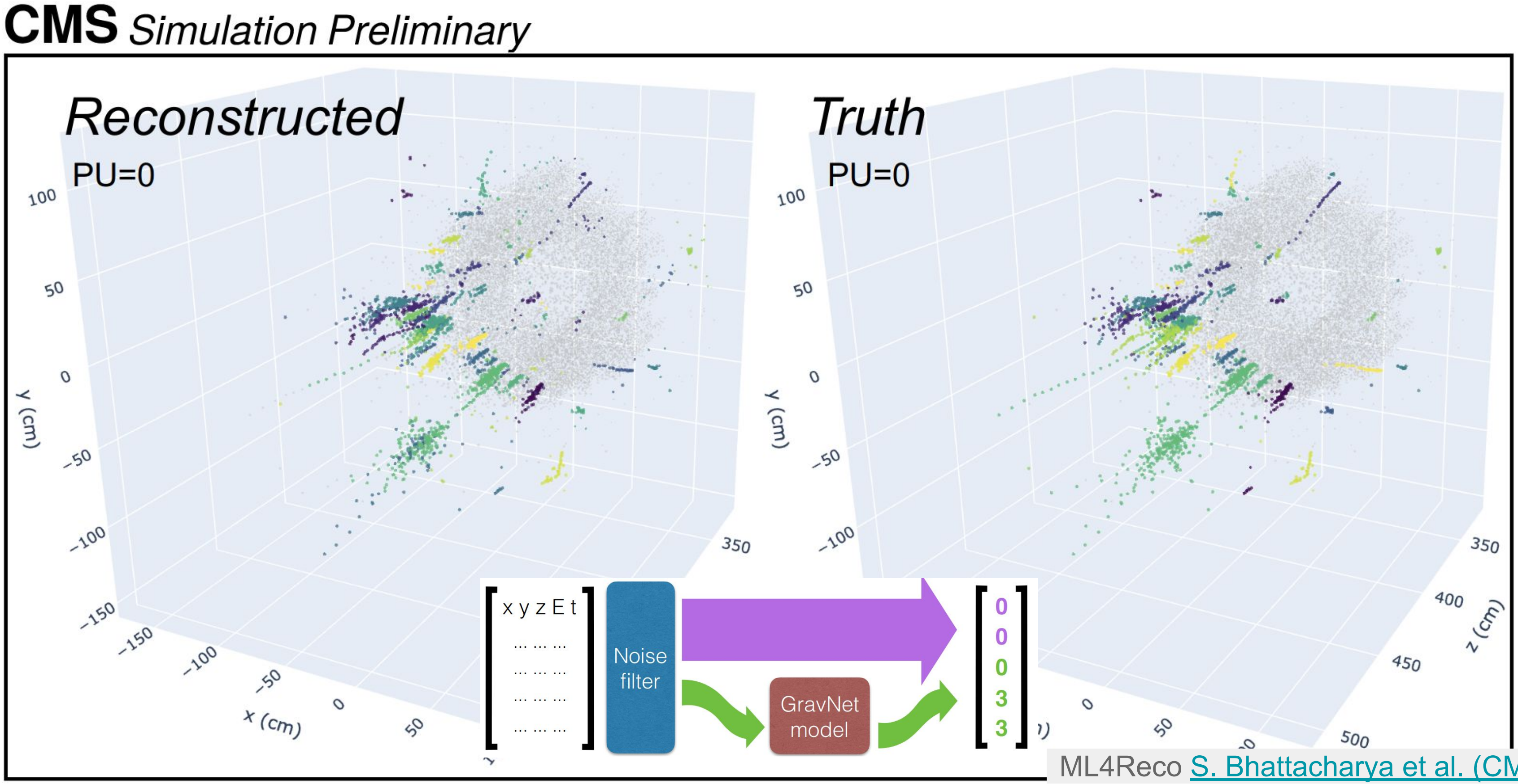
Trajectory fragments are stitched together to form a complete trajectory. Same algorithm reused to group particles into an interaction



See Phys. Rev. D 104, 072004 (2020)

Data Reconstruction in Experimental Particle Physics

GNN for Clustering in Calorimeter (CMS HGCAL Simulation)



ML4Reco S. Bhattacharya et al. (CMS)

Data Reconstruction in Experimental Particle Physics

ML Particle Flow @ Collider (Reco for Multi-modal Data)

Machine learned particle flow (MLPF)

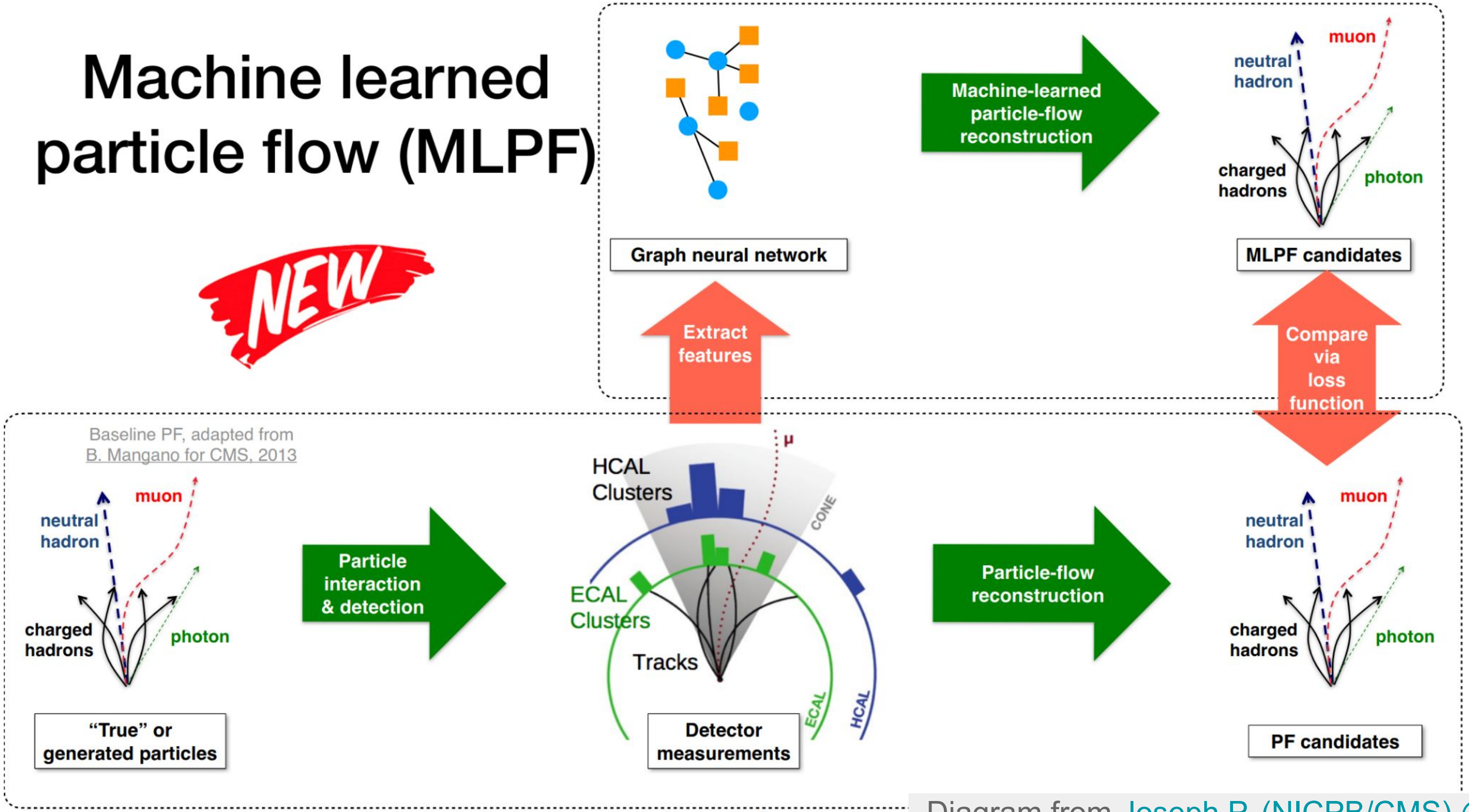
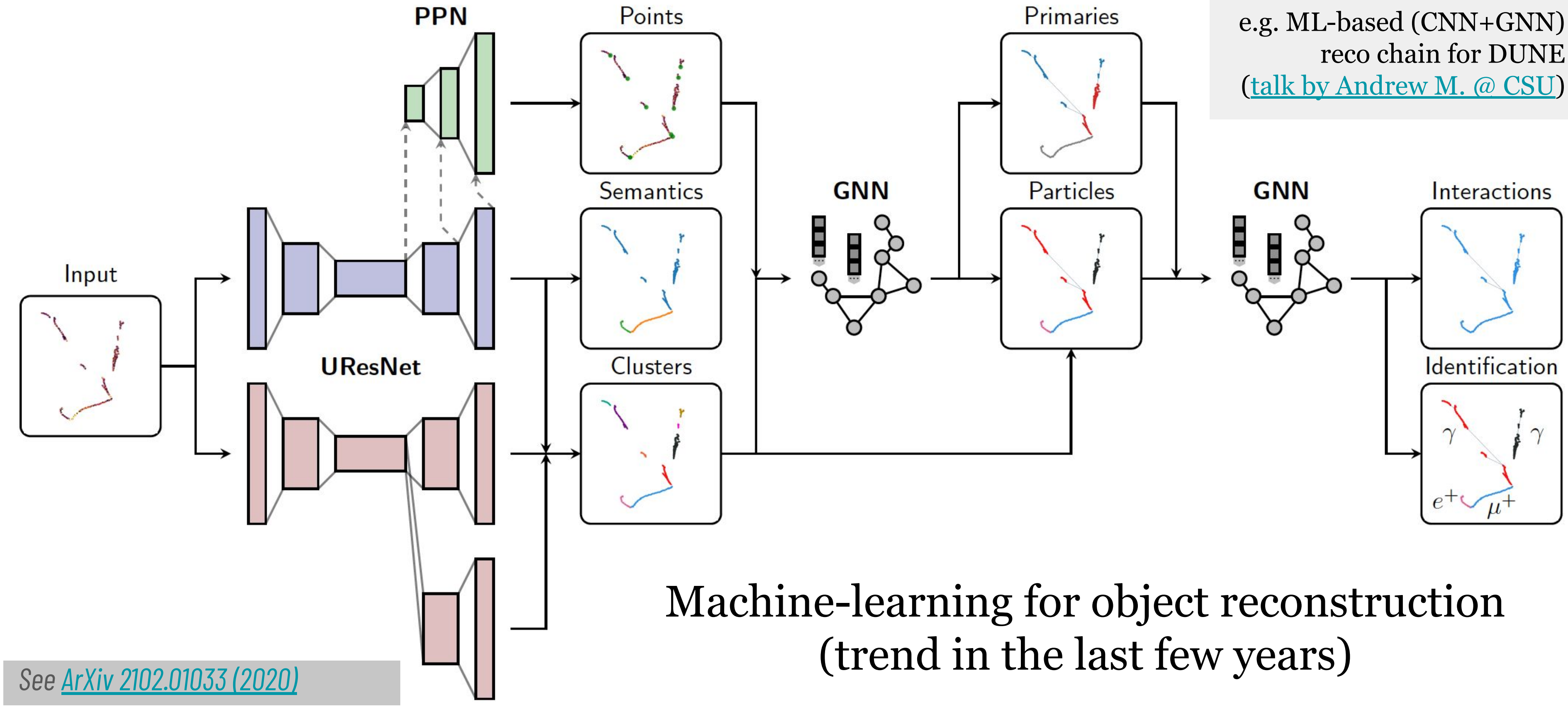


Diagram from [Joseph P. (NICPB/CMS) @ ACAT2021]

Data Reconstruction in Experimental Particle Physics

Automated optimization for an end-to-end reconstruction

e.g. ML-based (CNN+GNN) reco chain for DUNE (talk by Andrew M. @ CSU)



Machine-learning for object reconstruction (trend in the last few years)

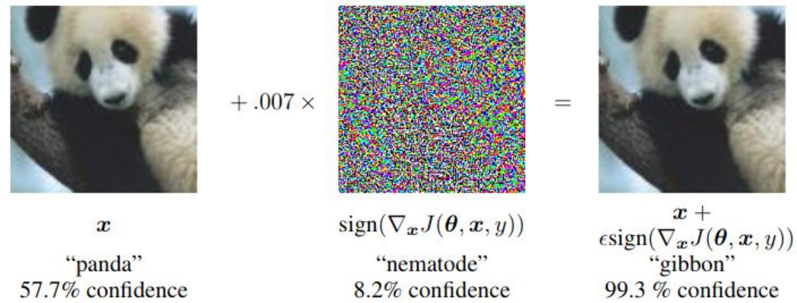
See ArXiv 2102.01033 (2020)

ML for Detector Physics Modeling
Automation of physics model tuning

ML for Tuning Physics Models

Physics model tuning

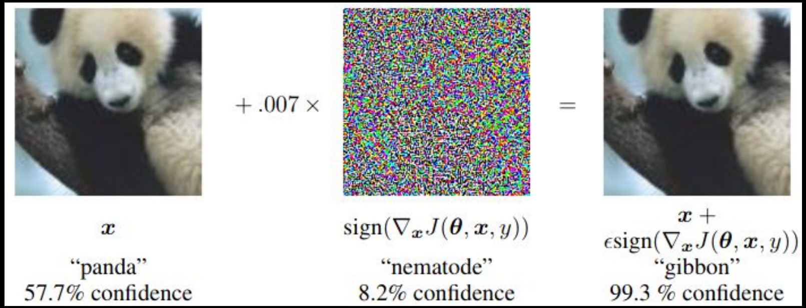
[Explaining and harnessing adversarial examples]



The Catch

Used supervised optimization with simulated particle interactions, which may be imperfect (i.e. domain shift)

[Explaining and harnessing adversarial examples]()

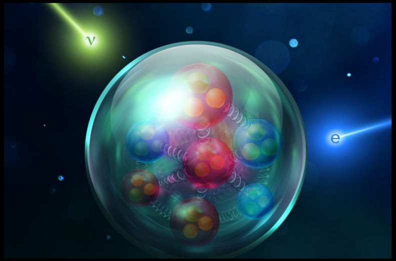


The Catch

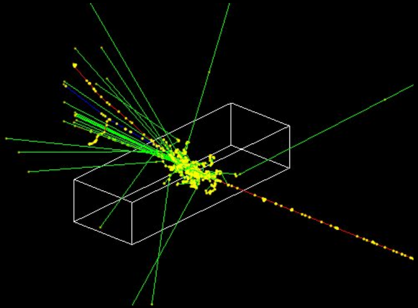
Used supervised optimization with simulated particle interactions, which may be imperfect (i.e. domain shift)

= multiple iterations of manual tuning

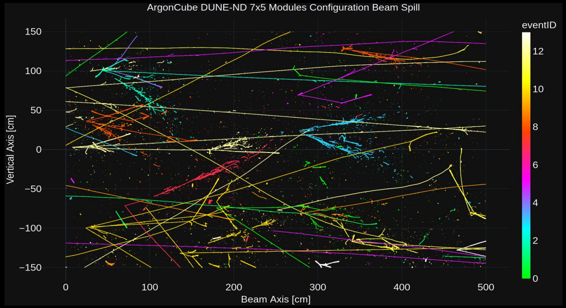
Example: detector physics modeling



Fundamental particle interactions



Interaction with the detector volume

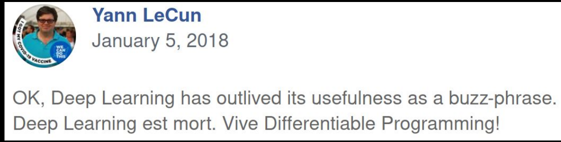


Detector response simulation

ML for Detector Physics Modeling
Physics model tuning

Recent success in machine learning … much are backed by **deep learning**

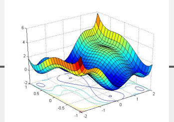
… for which, one key success is **gradient-based optimization**


Yann LeCun
January 5, 2018

OK, Deep Learning has outlived its usefulness as a buzz-phrase.
Deep Learning est mort. Vive Differentiable Programming!

Approximated gradient

Input
x



physics model parameters
θ

Output
$F(x|\theta)$

Optimization target
$L(F(x|\theta), y)$

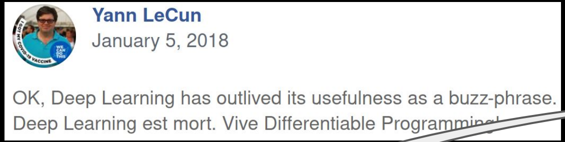
Exact gradient

ML for Detector Physics Modeling

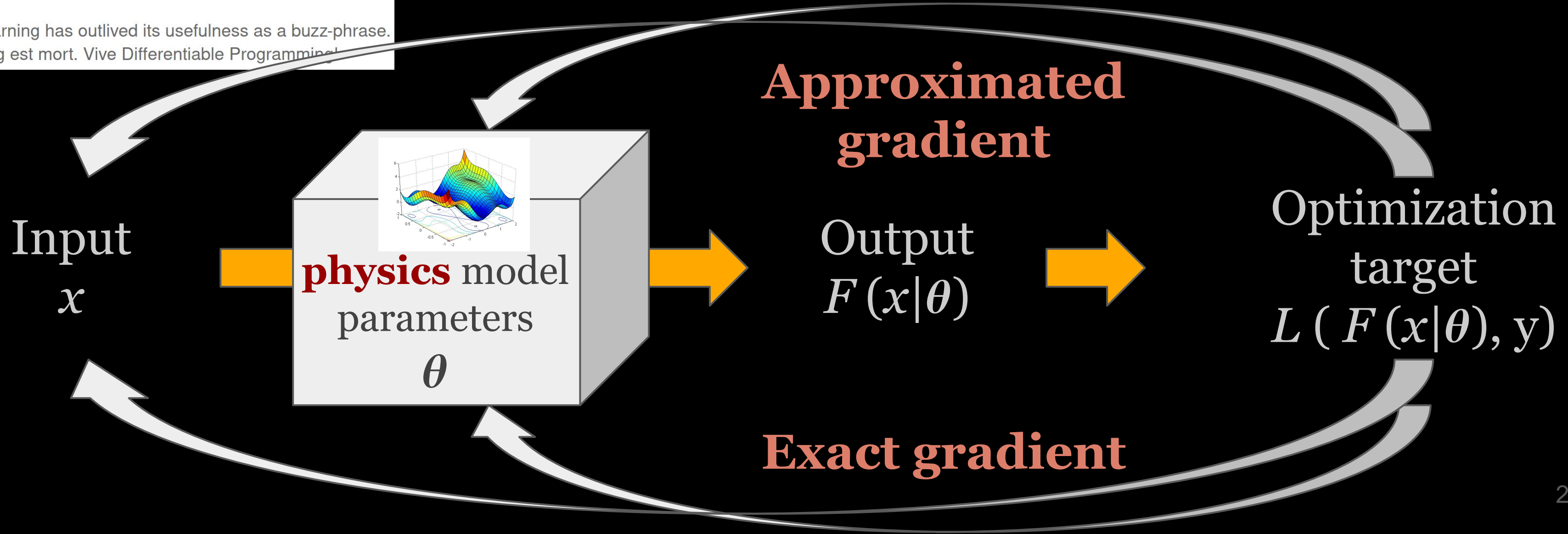
Physics model tuning

Recent success in machine learning … much are backed by **deep learning**

… for which, one key success is **gradient-based optimization**


Yann LeCun
January 5, 2018

OK, Deep Learning has outlived its usefulness as a buzz-phrase.
Deep Learning est mort. Vive Differentiable Programming!



Example Application: Modeling Optical Visibility Map

ML for Detector Physics Modeling
Differentiable detector simulator

Photo-multiplier tubes (PMTs) detect scintillation photons

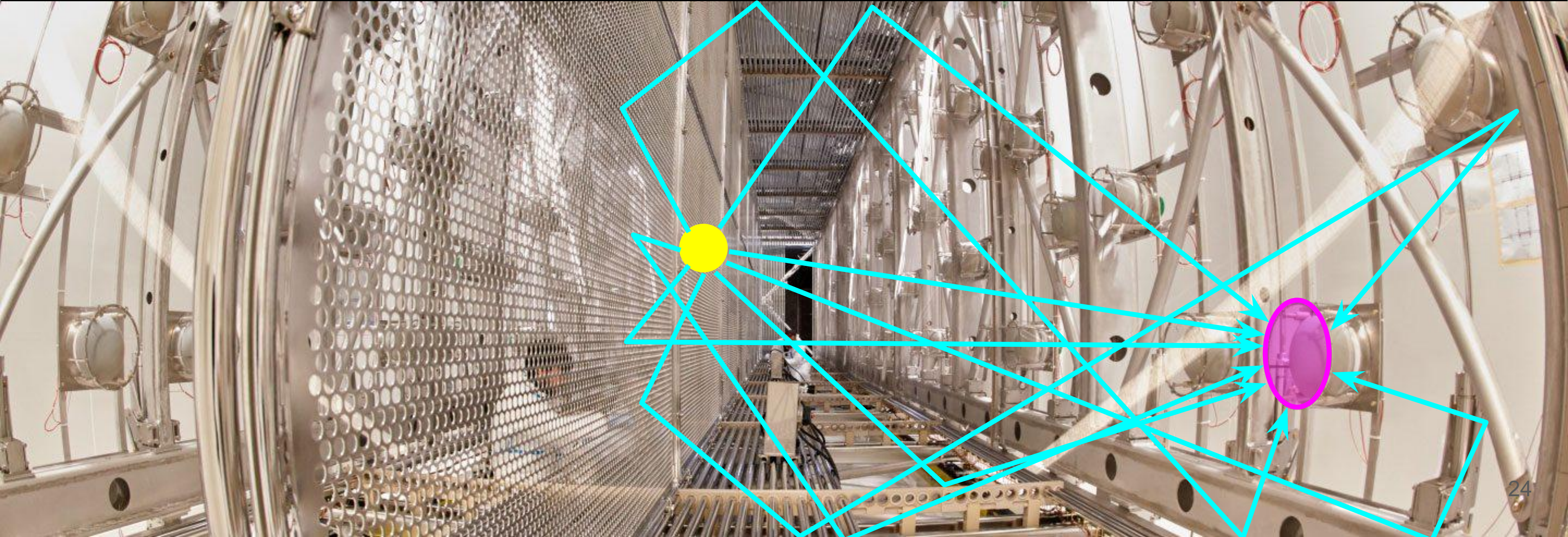
Optical Photon Transport

ML for Detector Physics Modeling
Differentiable detector simulator

Photo-multiplier tubes (PMTs) detect scintillation photons produced isotropically from an Argon atom

1 meter muon produces **> 4M photons**

Optical Photon Transport

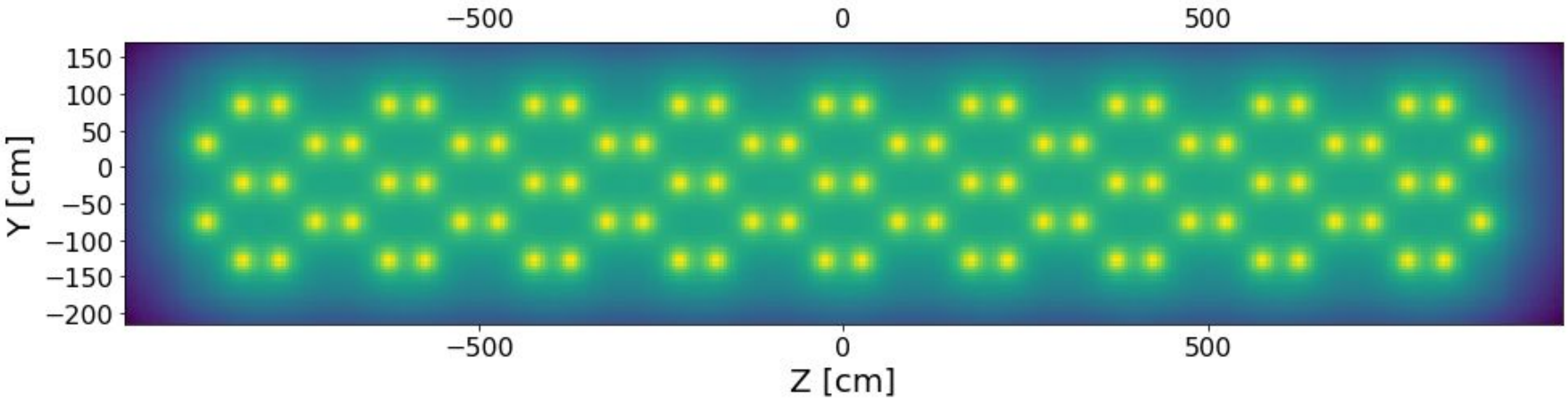


ML for Detector Physics Modeling
Differentiable detector simulator

A marginalized "**Visibility Map**" for 3D voxelized volume used to estimate photon count at each PMT

Issue: static, not scalable

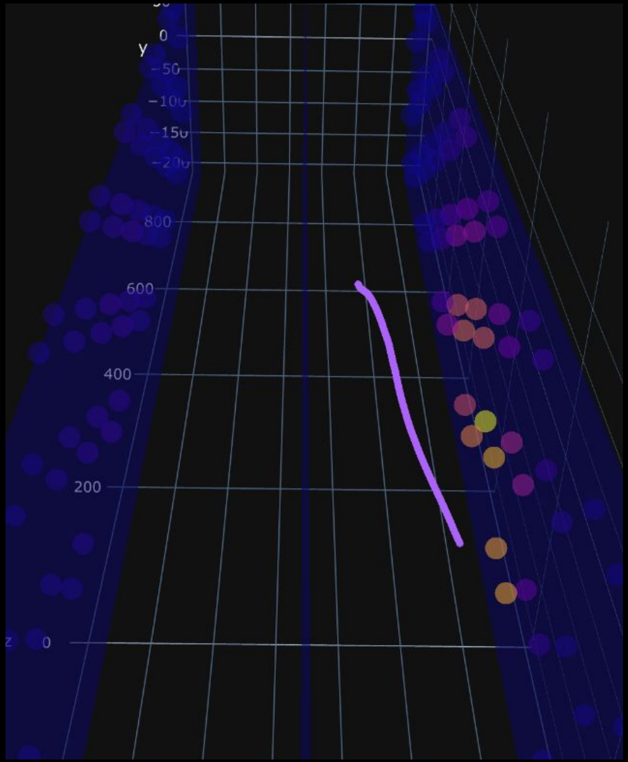
Optical Photon Transport



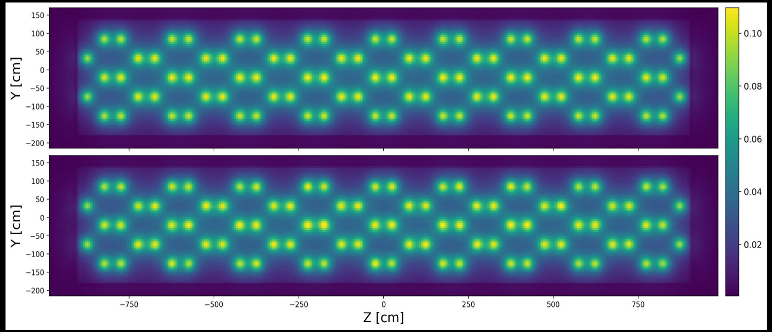
Example: ICARUS detector, 2D slice of a 3D map

ML for Detector Physics Modeling

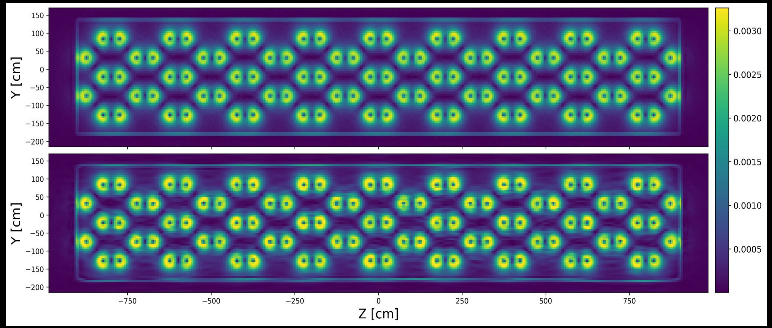
Differentiable detector simulator



Static map (top) v.s. SIREN



Gradient map (top, sobel filter) v.s. SIREN

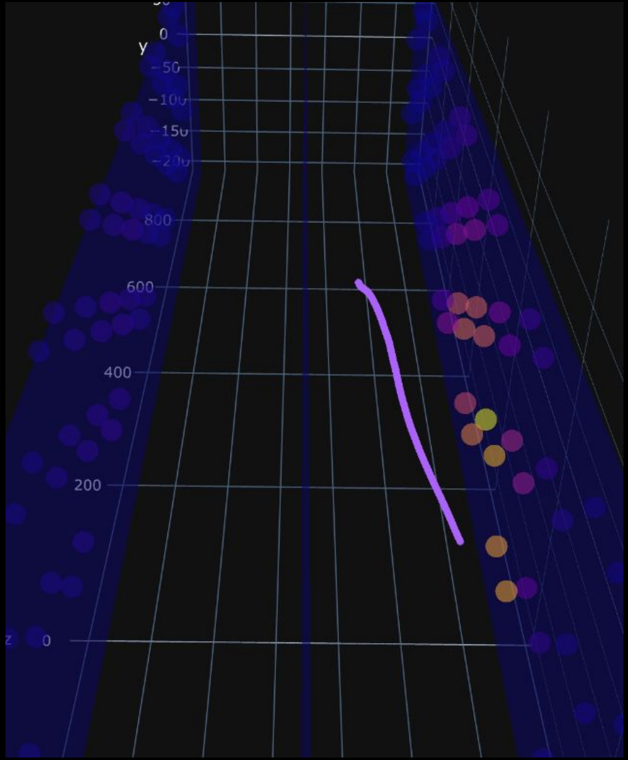


Optical Photon Transport using **Differentiable Surrogate (SIREN)**

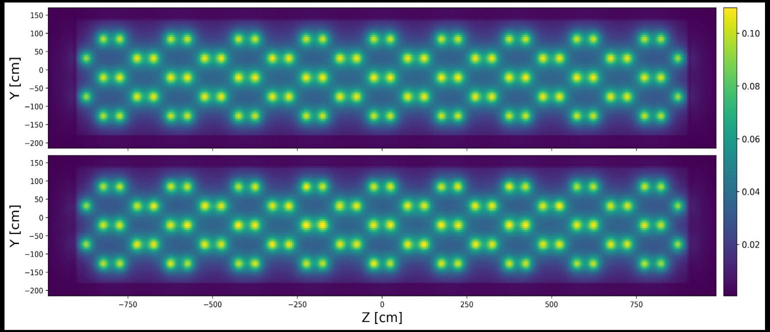
Neural scene representation (alternative: NeRF inc. differentiable rendering)

ML for Detector Physics Modeling

Differentiable detector simulator



Static map (top) v.s. SIREN



Optical Photon Transport using Differentiable Surrogate (SIREN)

SIREN enables …

- Avoid an explicit likelihood calculation which is intractable for optimization (likelihood-free inference)
- Smooth interpolation of optical visibility
- Data-driven optimization of visibility map
- Position-dependent discrepancy (error) propagation

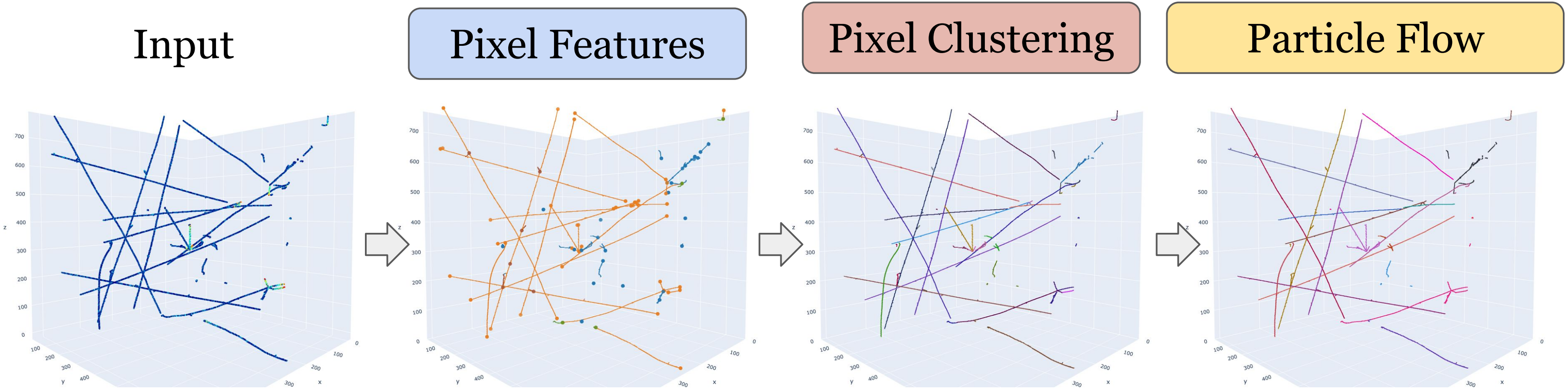
Representation Learning by Foundation Models

Research on General AI and HEP datasets

Scalable, Extensible, General AI for HEP
Cons on Composite Machine Learning Models

Challenges in extending ML for all reconstruction tasks + combining them

- **Factorization** is useful (e.g. application of domain knowledge, interpretable intermediate outputs)



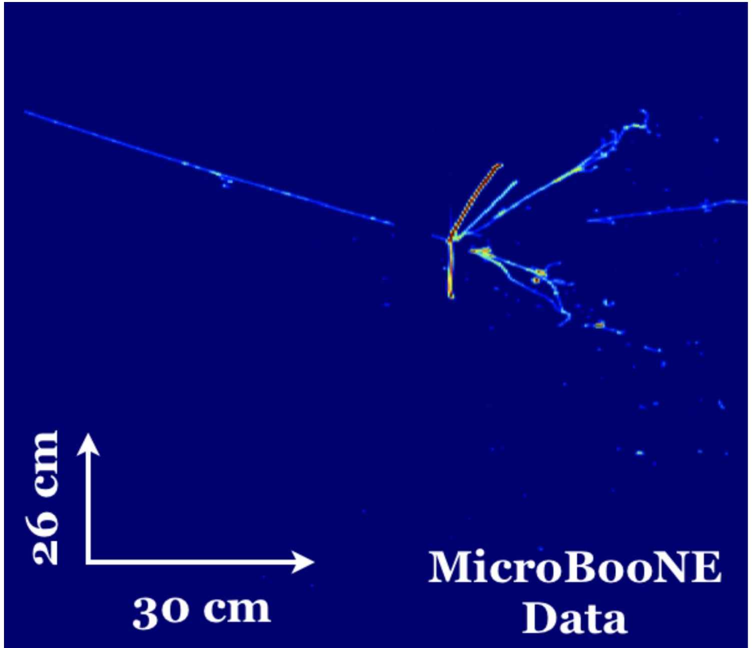
Cons on Composite Machine Learning Models

Challenges in extending ML for all reconstruction tasks + combining them

- **Factorization** is useful (e.g. application of domain knowledge, interpretable intermediate outputs) **but may be a bottleneck for learning capability.**

Where is the vertex?

Human brains are capable to inspect multiple scenario simultaneously / recursively. (i.e. "look twice")



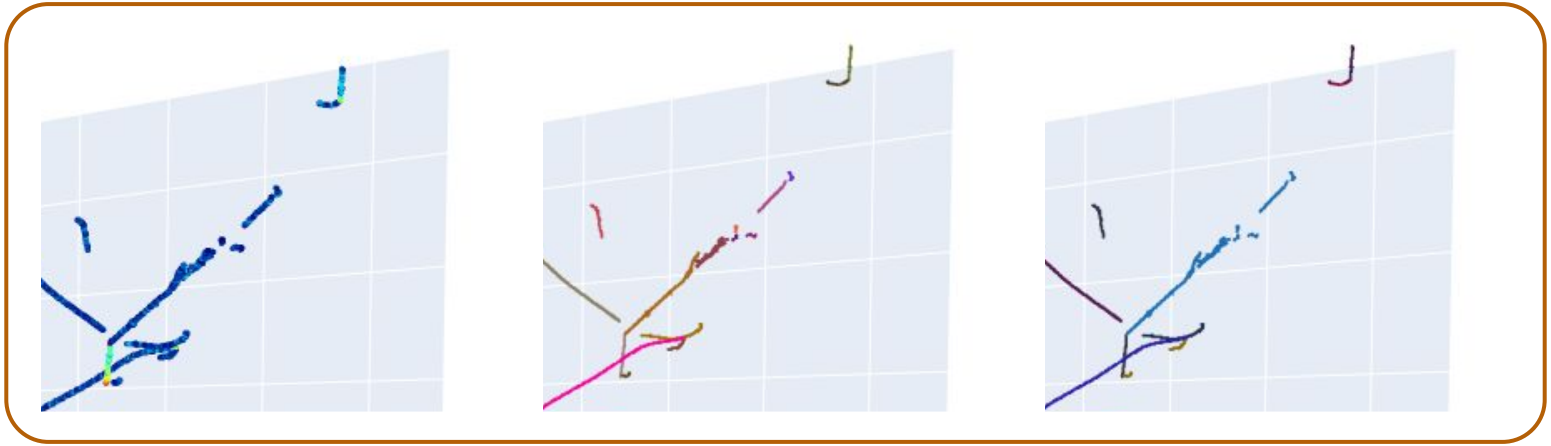
Scalable, Extensible, General AI for HEP

Cons on Composite Machine Learning Models

SLAC

Challenges in extending ML for all reconstruction tasks + combining them

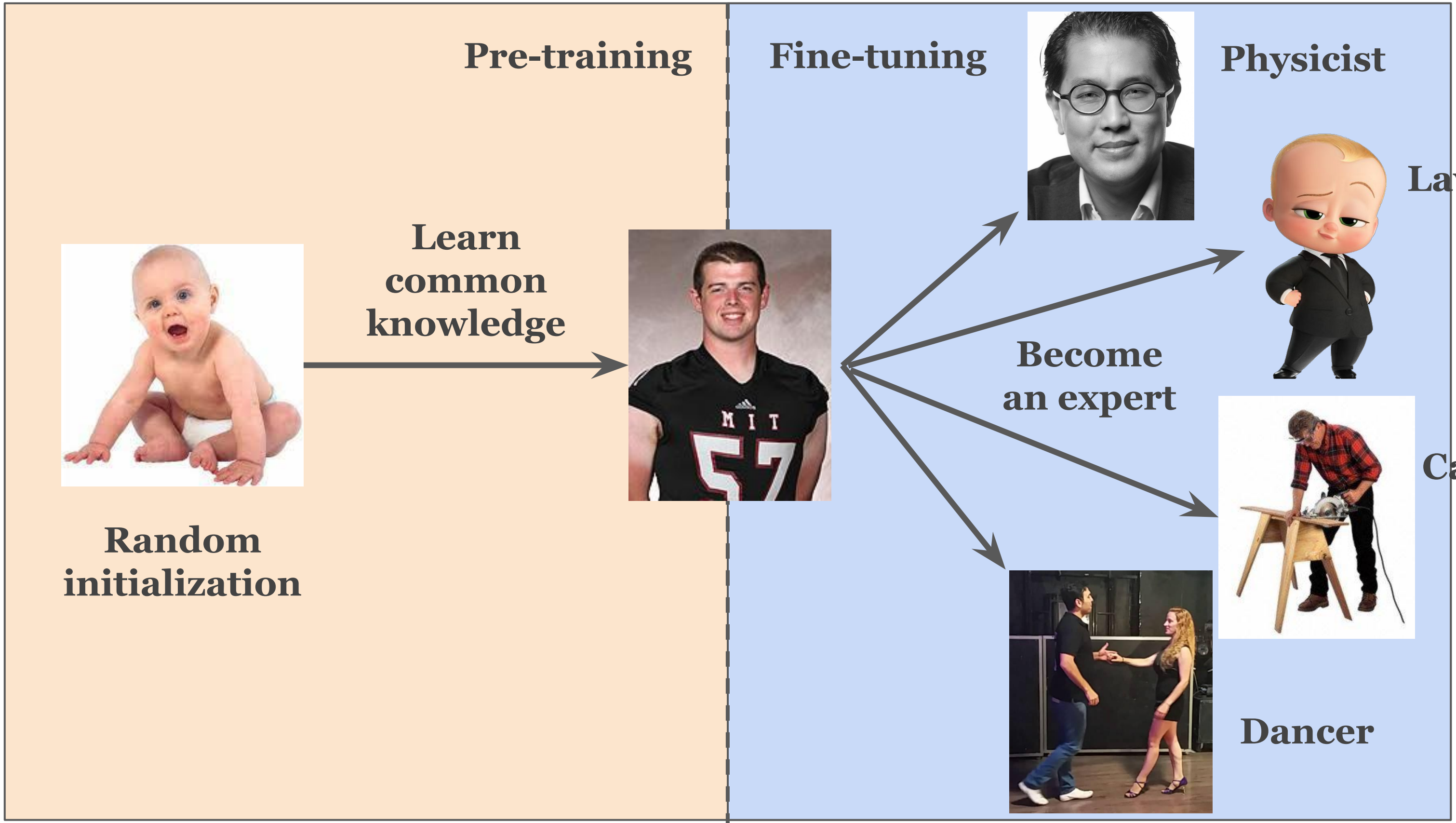
- **Factorization** is useful (e.g. application of domain knowledge, interpretable intermediate outputs) but may be a bottleneck for learning capability.
- **Multiple task-specific models ~ duplicated modeling** = energy inefficiency



Concept of a particle instance and trajectory is learned multiple times

Scalable, Extensible, General AI for HEP

General AI: how do we "train" a human?



General AI: self-supervised representation learning

Pre-training

Learn common knowledge

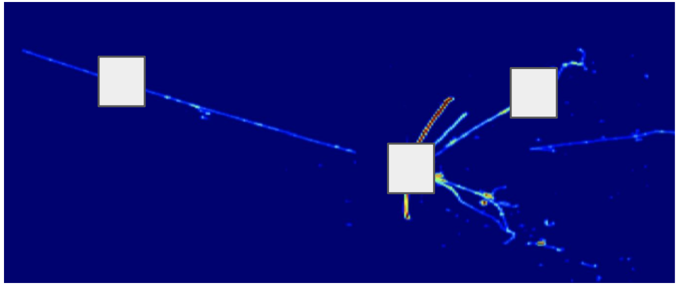
Random initialization

Self-Supervised Learning

- "Mask" portions of input data, task the model to predict what is under the mask.

Physicists love free ______.

I need coffee in the ______.



- No labels needed
- Task-agnostic: engineers general features ("representation")

Scalable, Extensible, General AI for HEP
Foundation Model: Task-agnostic, Representation Learning

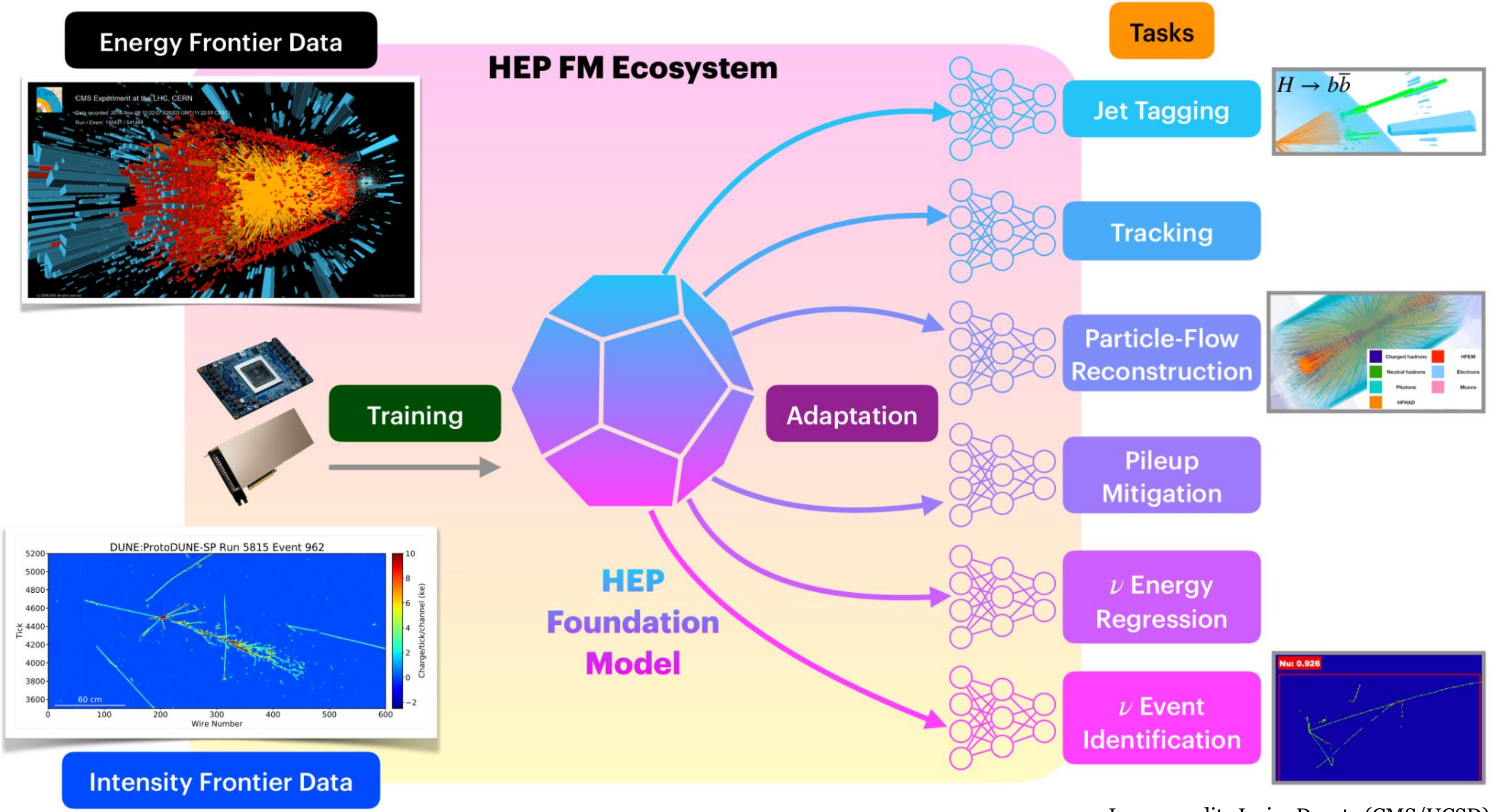


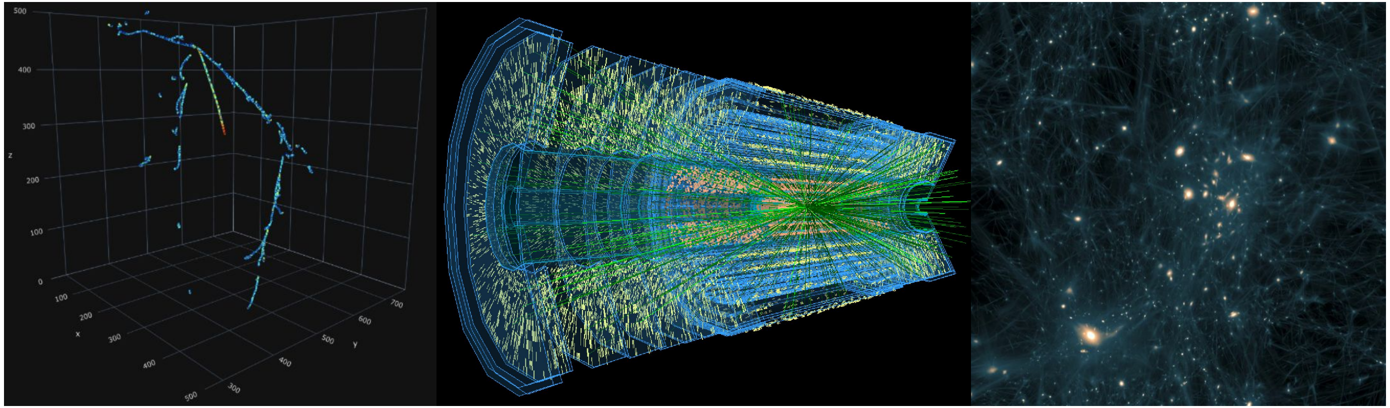
Image credit: Javier Duarte (CMS/UCSD)

HEP AI Ecosystem

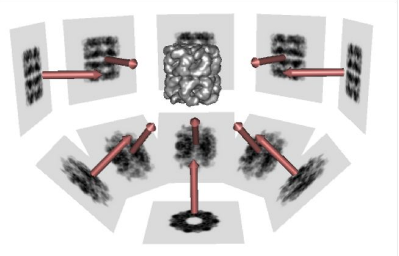
Foundation Model: Task-agnostic, Representation Learning

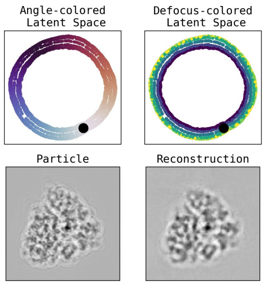
ML is a "solution pattern" v.s. a domain-specific "hard-coded" solution.
It's **naturally reusable across domains including software tools**
supported by a large community of researchers.

e.g.) physics inference on data from imaging detectors



Intensity Frontier | **Energy Frontier** | **Cosmic Frontier**

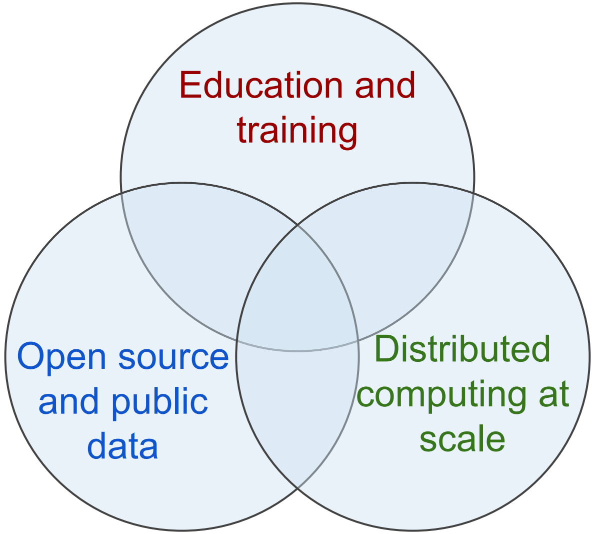




e.g.) Cryo-EM

HEP AI Ecosystem
Foundation Model: Task-agnostic, Representation Learning

ML is a "solution pattern" v.s. a domain-specific "hard-coded" solution. It's **naturally reusable across domains including software tools** supported by a large community of researchers.



HEP Ecosystem for AI research

- Accessible **education and training** at all levels
- **Reusable software tools** to unlock modern compute accelerators and networking (distributed ML)
- **Public datasets** with documentation and performance metrics for transparent, reproducible science
- Artificial Intelligence and Technology Office (AITO)
  - Federated, equitable, responsible, trustworthy AI
  - **AI is an accelerator**. It is coming. Don't avoid. **Participate to make sure the use is good.**

… wrapping up …

Wrapping-Up

SLAC

Take-aways

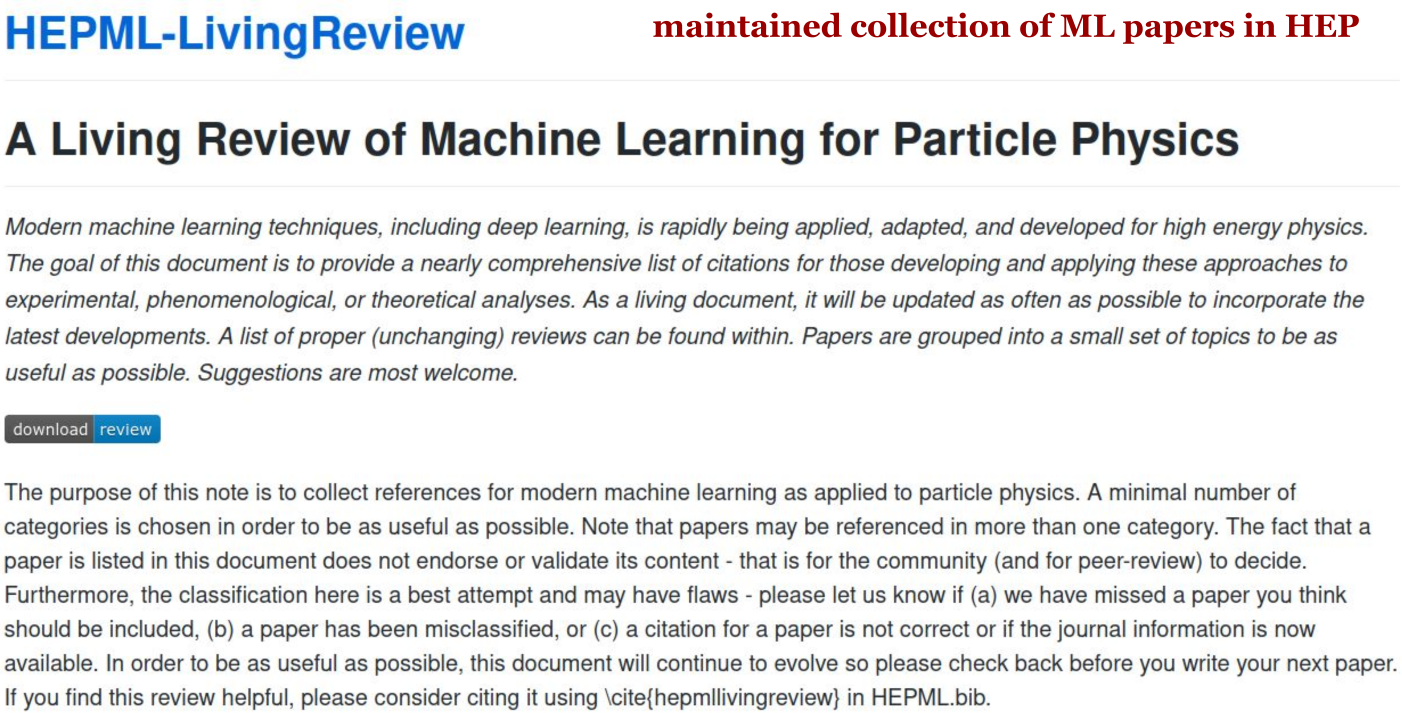
- Deep learning for an “end-to-end” physics object reconstruction
  - Now widely developed + exposed challenges (e.g. domain shift)
- Likelihood-free inference
  - Wide applications in physics parameter inference. Example in this talk = differentiable NN surrogate for modeling optical visibility
- Representation learning by Foundation Models
  - Foundation Models: strong overlap with general AI research and science

Topics not covered (not exclusive list)

- Uncertainty quantification for ML methods (example paper)
- Physics-informed Neural Networks … include physics constraints in optimization
- More examples on likelihood-free / simulation-based inference

Some references

maintained collection of ML papers in HEP


HEPML-LivingReview

A Living Review of Machine Learning for Particle Physics

Modern machine learning techniques, including deep learning, is rapidly being applied, adapted, and developed for high energy physics. The goal of this document is to provide a nearly comprehensive list of citations for those developing and applying these approaches to experimental, phenomenological, or theoretical analyses. As a living document, it will be updated as often as possible to incorporate the latest developments. A list of proper (unchanging) reviews can be found within. Papers are grouped into a small set of topics to be as useful as possible. Suggestions are most welcome.

download | review

The purpose of this note is to collect references for modern machine learning as applied to particle physics. A minimal number of categories is chosen in order to be as useful as possible. Note that papers may be referenced in more than one category. The fact that a paper is listed in this document does not endorse or validate its content - that is for the community (and for peer-review) to decide. Furthermore, the classification here is a best attempt and may have flaws - please let us know if (a) we have missed a paper you think should be included, (b) a paper has been misclassified, or (c) a citation for a paper is not correct or if the journal information is now available. In order to be as useful as possible, this document will continue to evolve so please check back before you write your next paper. If you find this review helpful, please consider citing it using \cite{hepmllivingreview} in HEPML.bib.

Data Reconstruction in Experimental Particle Physics

Wrapping-Up

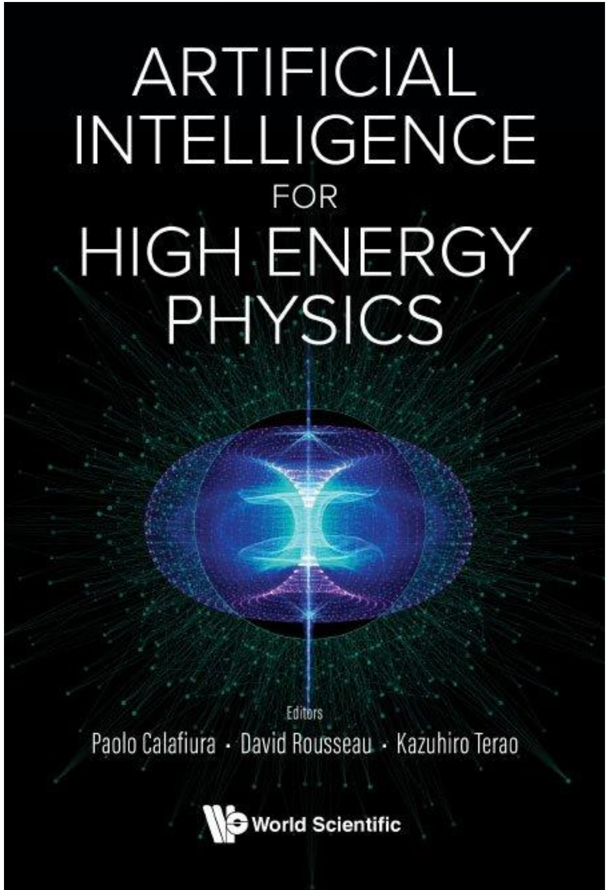
SLAC

Some references

Recently published review book covering ML for HEP

ML review chapter in PDG book

particle data group

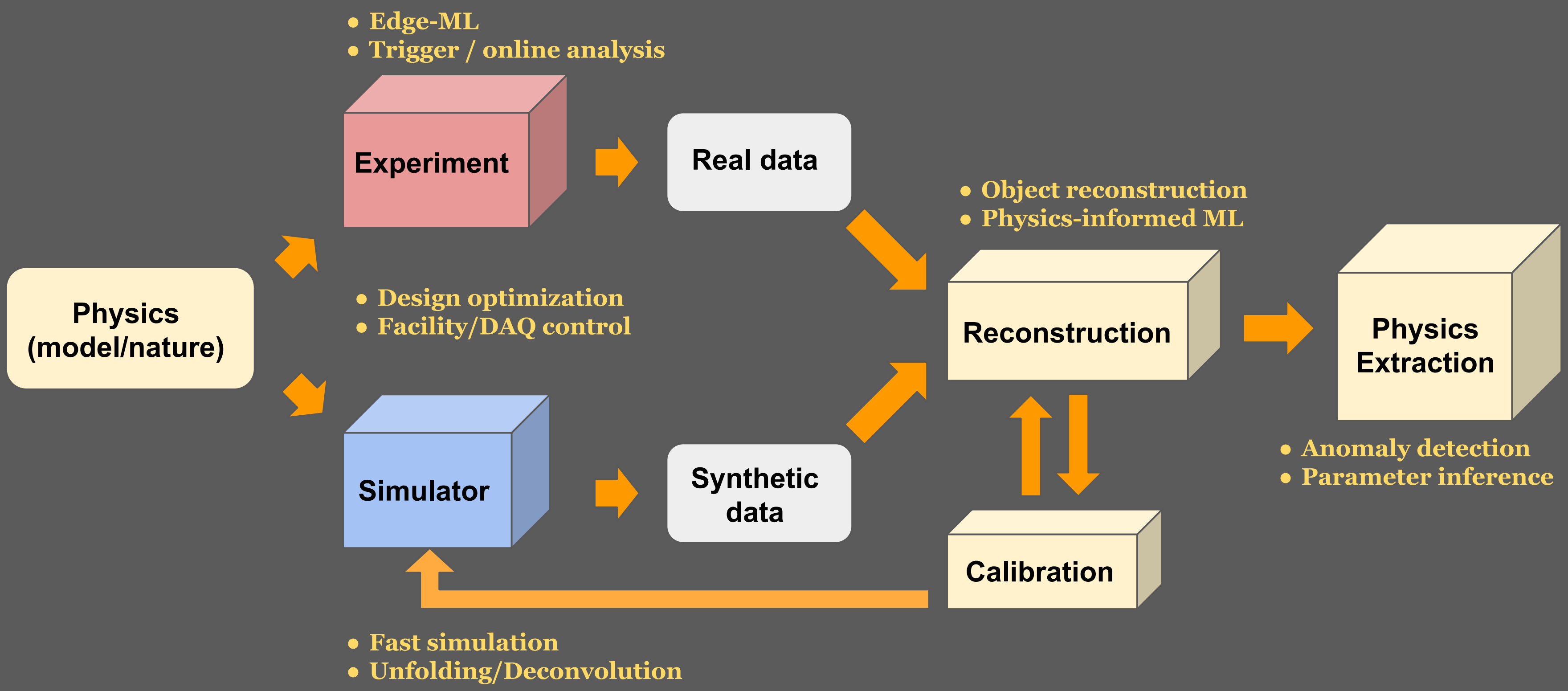


41. Machine Learning

Written November 2021 by K. Cranmer (NYU), U. Seljak (UC Berkeley; LBNL) and K. Terao (SLAC; Stanford U.).

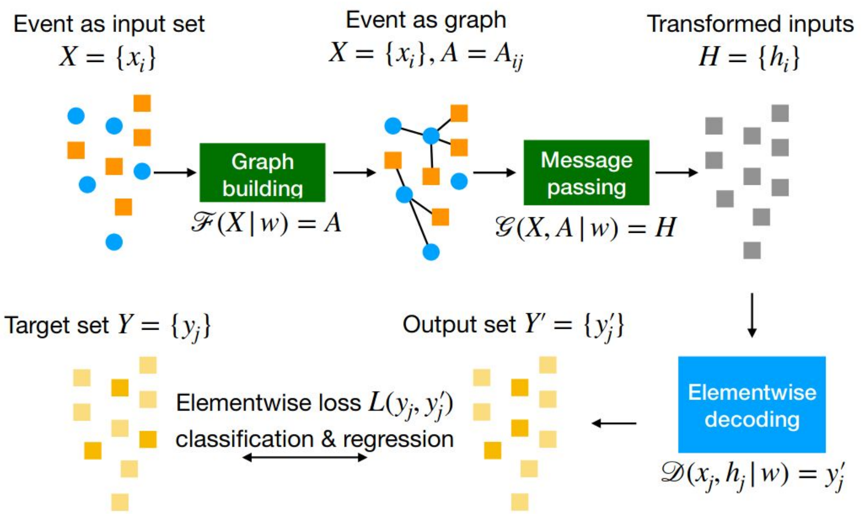
Back-up slides

Machine Learning in Particle Physics
Experiment Pipeline



Data Reconstruction in Experimental Particle Physics

ML Particle Flow @ Collider (Reco for Multi-modal Data)



$x_i = [\text{elem.type}, p_T, E_{ECAL}, E_{HCAL}, \eta, \phi, \eta_{outer}, \phi_{outer}, q, \ldots]$

$y_j = [\text{PID}, p_T, E, \eta, \phi, q], \quad \text{PID} \in \{\text{none, charged hadron, neutral hadron}, \gamma, e^{\pm}, \mu^{\pm}, \ldots\}$

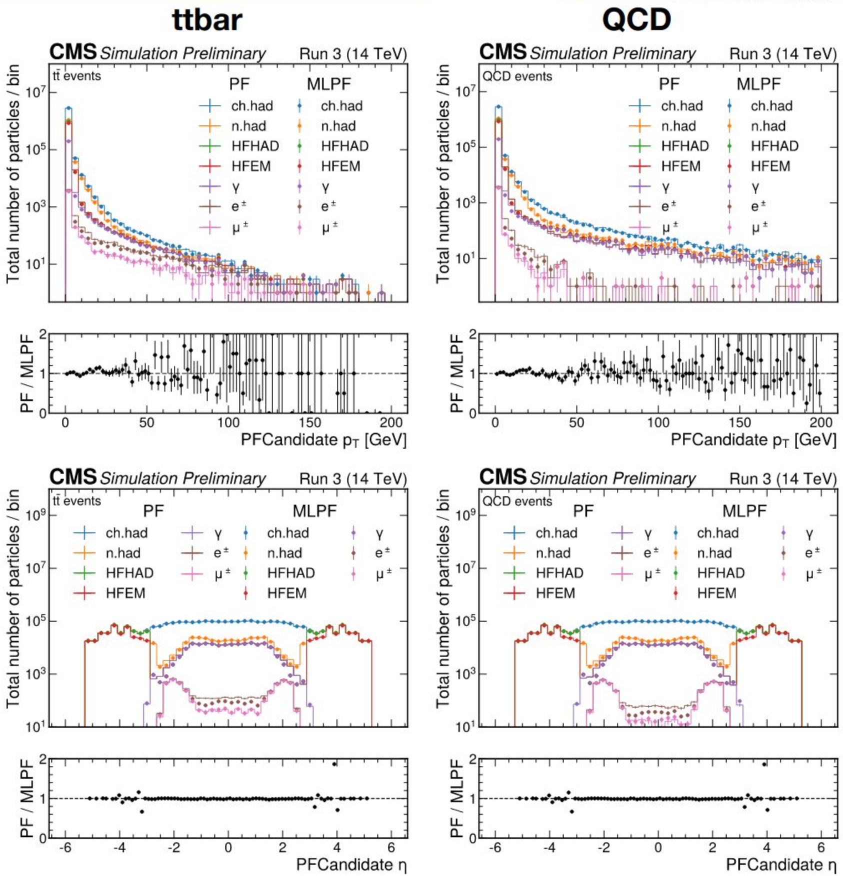
$h_i \in \mathbb{R}^{N_{hidden}}$

Trainable neural networks: $\mathscr{F}, \mathscr{G}, \mathscr{D}$

● - track, ■ - calorimeter cluster, ■ - encoded element

■ - target (predicted) particle, ■ - no target (predicted) particle

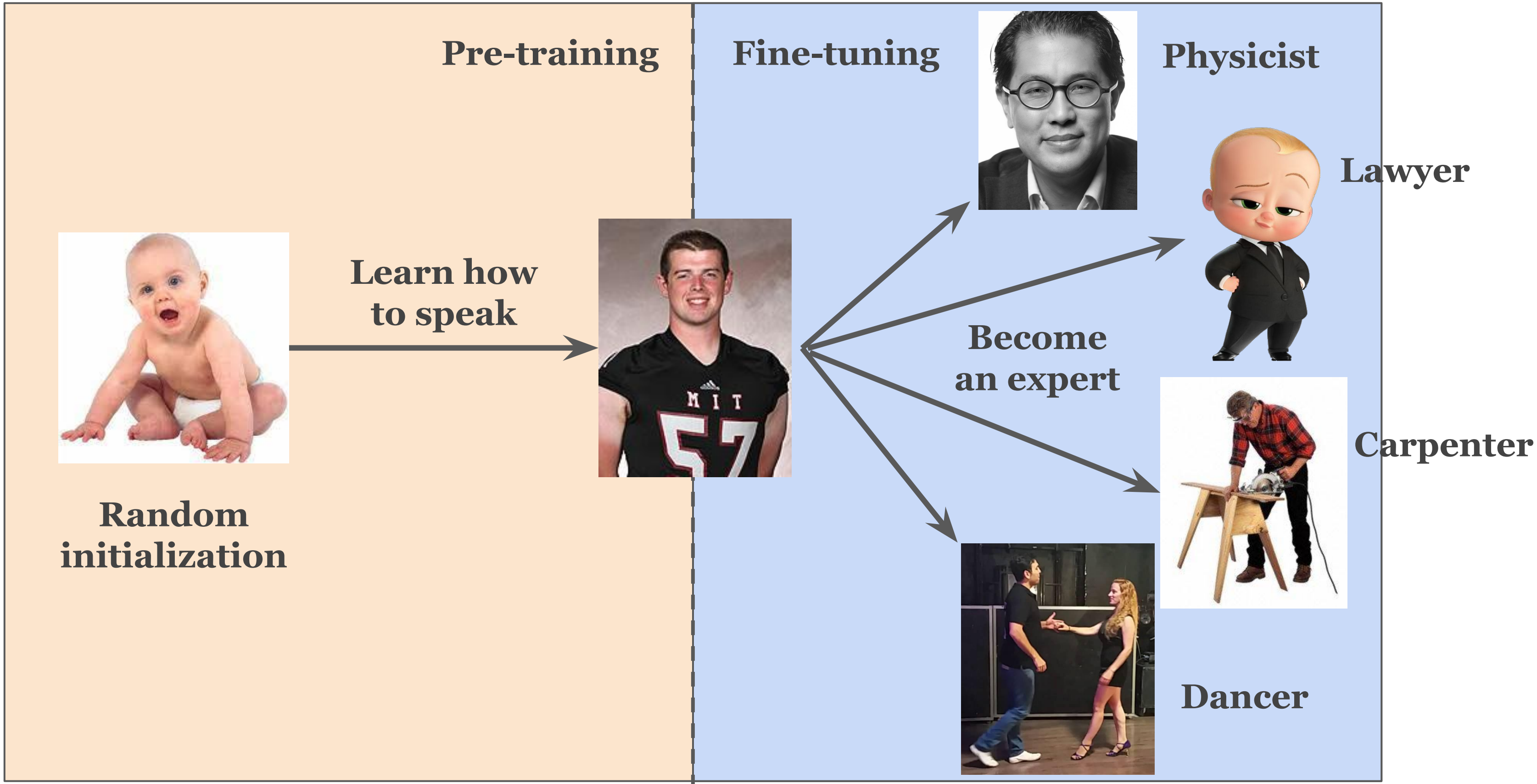
Diagram/figures from Joseph P. (NICPB/CMS) @ ACAT2021



Transformer to GPT

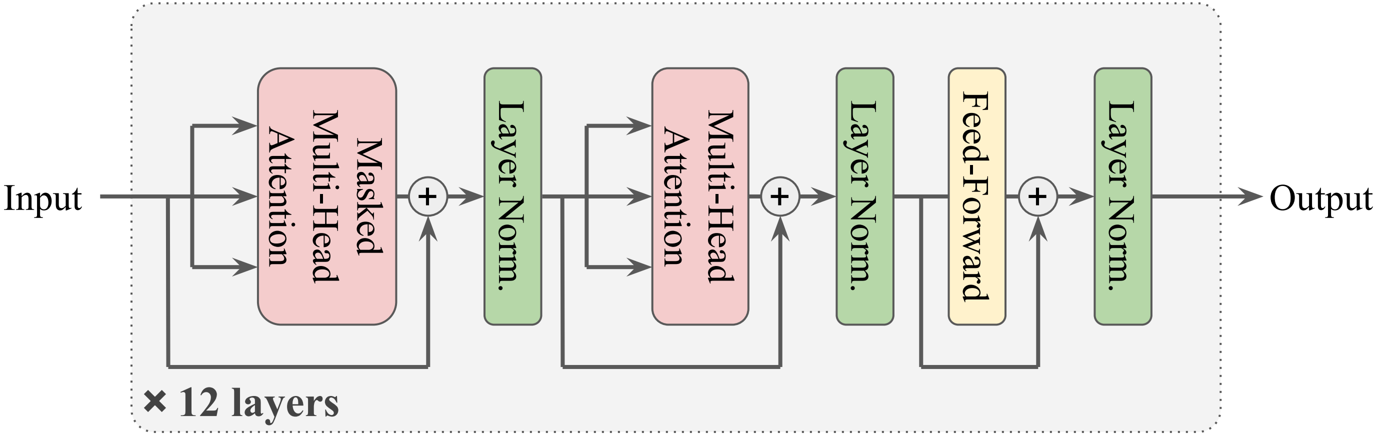
General Pre-trained Transformer (GPT)

Idea: Pre-train on big data, then fine-tune w/ small data on a specialized task



General Pre-trained Transformer 1 (GPT-1)

Idea: Pre-train on big data, then fine-tune w/ small data on a specialized task

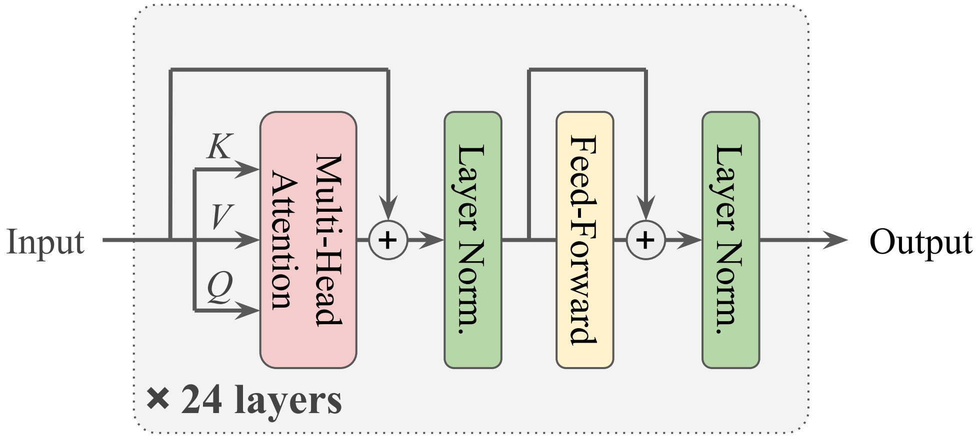


Pre-training: "**next word prediction**" using the decoder of a transformer (above) + linear layer + softmax. No need to generate labels = massive amount of dataset (all digitally available literature) can be used to train. This task allows the model to learn language.

Fine-tuning: a specialized task with small amount of labeled data. Change the final linear layer + softmax depending on the task, but re-use the same model before these layers.

Bidirectional Encoder Representations from Transformers (BERT)

Idea: use the whole sequence + no architecture change at fine-tuning



Implemented in Google Translation (end of 2019)

Pre-training: "**masked language prediction**" using the encoder of a transformer (above). The model is tasked to fill the masked word in the input sequence. "Next sentence prediction" is a classification task whether two sentences are in the right sequence or not. Both dataset can be generated from digital literature easily.

Fine-tuning: a specialized task with small amount of labeled data. No change in model architecture and successfully fine-tuned on multiple tasks

GPT-2 and GPT-3

Idea: can we skip even fine-tuning?

Same (almost) as GPT-1 in terms of an architecture, but make the model and dataset larger. Can it learn all language tasks from unlabeled pre-training dataset?

- **One-shot learning**: give a single example as a fine-tuning.
  - Possible if the model already learned the task during a pre-training, and a single example is used to map the task onto the learned knowledge space.

- **Zero-shot learning**: test a model on tasks that is never trained for.
  - Possible only if the model learned the task and solution space during pre-training.

What is the color of your ふく?

GPT-2 and GPT-3

Idea: can we skip even fine-tuning?

Same (almost) as GPT-1 in terms of an architecture, but make the model and dataset larger. Can it learn all language tasks from unlabeled pre-training dataset?

- **One-shot learning**: give a single example as a fine-tuning.
  - Possible if the model already learned the task during a pre-training, and a single example is used to map the task onto the learned knowledge space.

- **Zero-shot learning**: test a model on tasks that is never trained for.
  - Possible only if the model learned the task and solution space during pre-training.

What is the color of your ふく?

ふく= 衣服

GPT-2 and GPT-3

Idea: can we skip even fine-tuning?

Same (almost) as GPT-1 in terms of an architecture, but make the model and dataset larger. Can it learn all language tasks from unlabeled pre-training dataset?

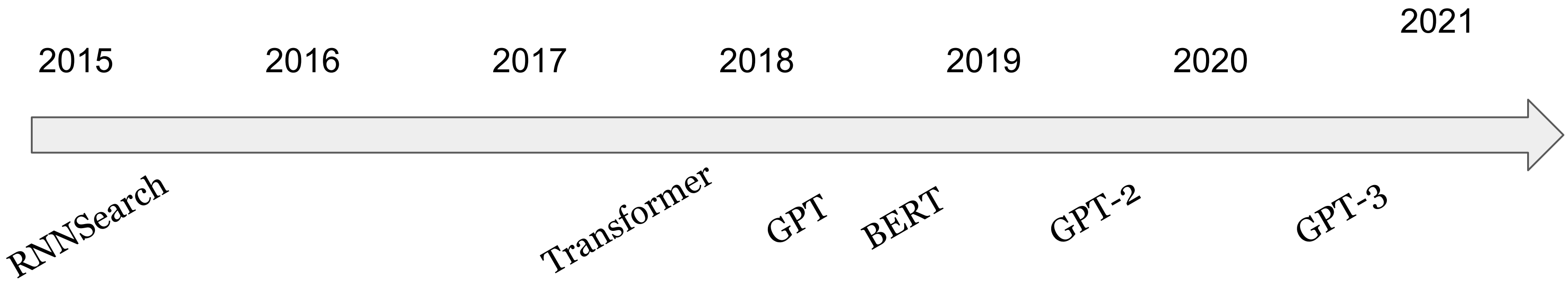
- **One-shot learning**: give a single example as a fine-tuning.
  - Possible if the model already learned the task during a pre-training, and a single example is used to map the task onto the learned knowledge space.

- **Zero-shot learning**: test a model on tasks that is never trained for.
  - Possible only if the model learned the task and solution space during pre-training.

What is the color of your ふく?

ふく= 衣服 = clothing

… attention mechanism is expanding …



Applications/Relation to image analysis

DALL-E, ViT, Perceiver, …

Reconstruction chain

End-to-End
ML Reco Chain for
Neutrino Detectors

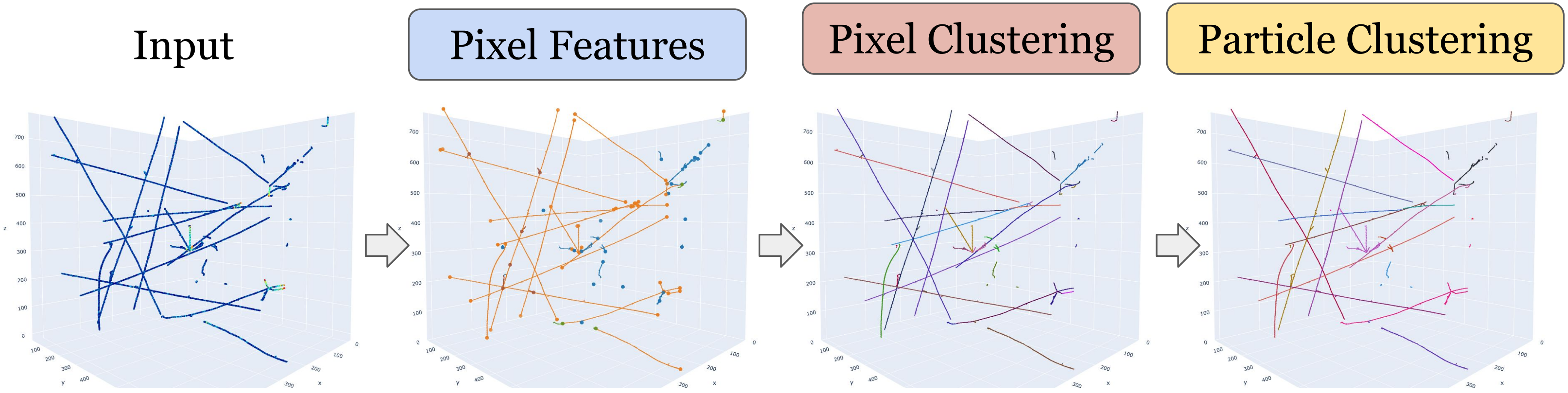
ML for Analyzing Big Image Data in Neutrino Experiments

End-to-end data reconstruction using ML

SLAC

Machine Learning for Neutrino Image Data Analysis

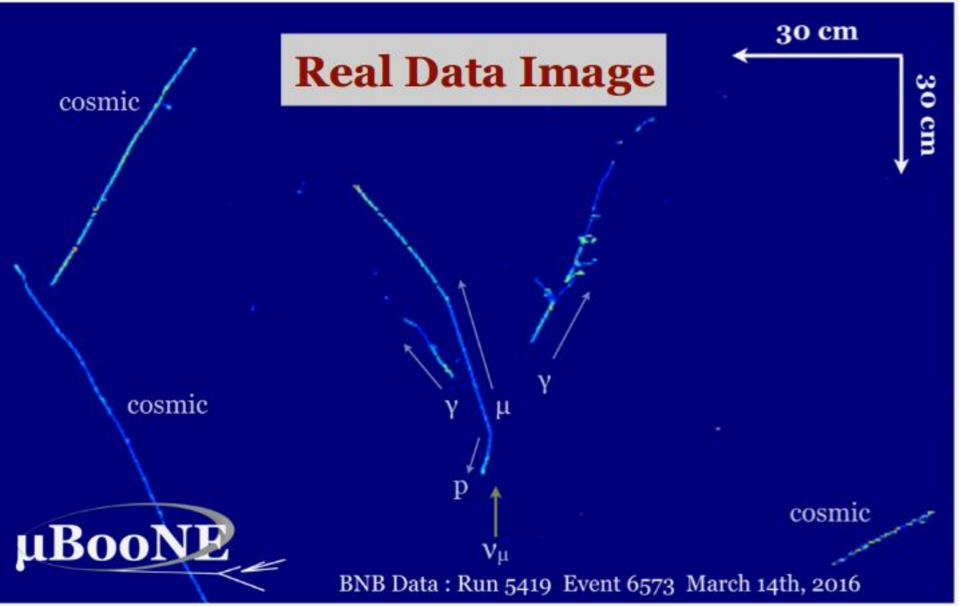
- **Goal**: particle-level type and energy reconstruction
- **How**: extract physically meaningful, hierarchical features (evidences) by chaining multiple ML models designed for each task



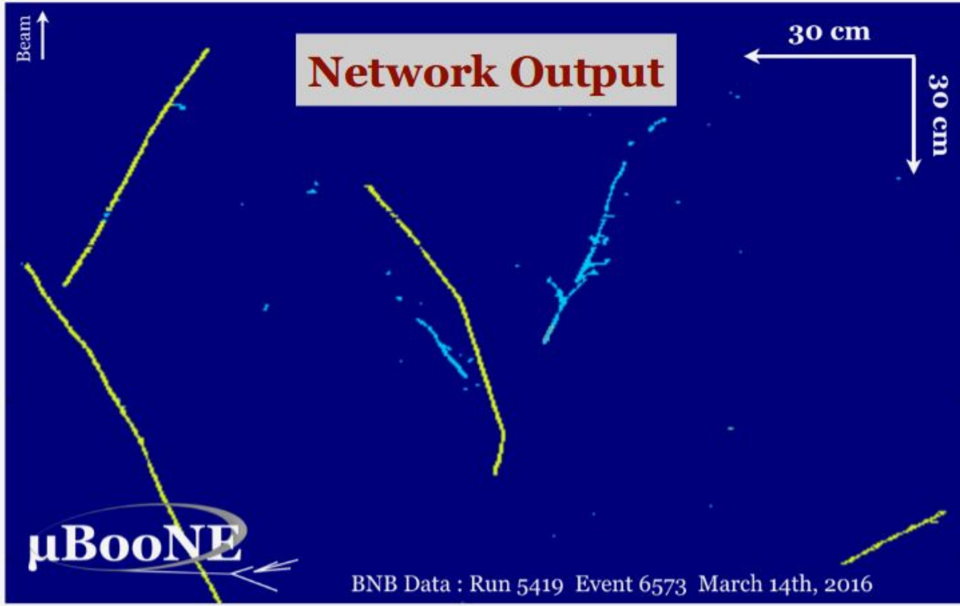
Three major stages of reconstruction

ML for Analyzing Big Image Data in Neutrino Experiments
Stage 1: pixel-level feature extraction

Distinguish 2 distinct topologies: **showers v.s. tracks** (for the next stage = clustering)
Identify trajectory **edge points** (track start/end, shower start)



Network Input

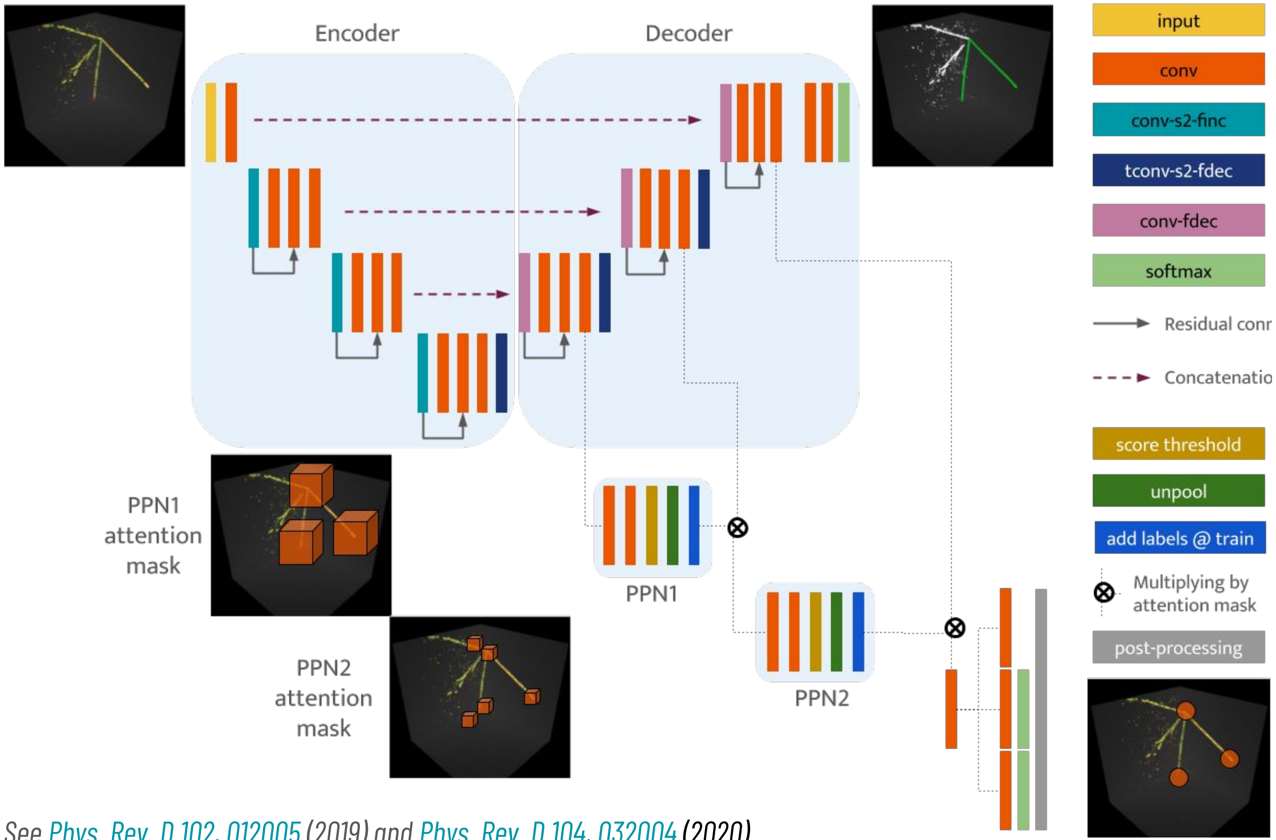


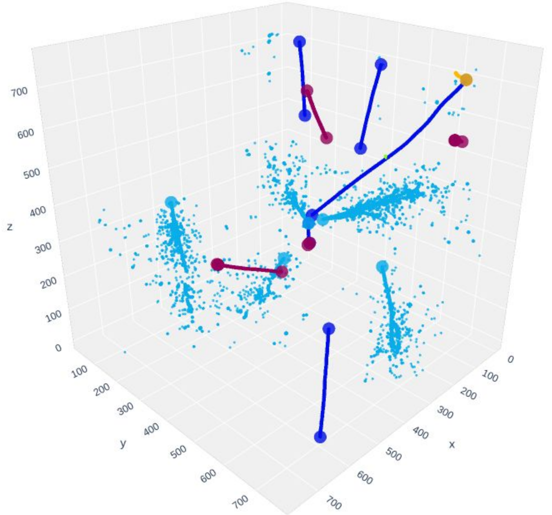
Network Output

PRD 99 092001 (2018)

ML for Analyzing Big Image Data in Neutrino Experiments

Stage 1: pixel-level feature extraction





Semantic segmentation
(U-Net + residual conn.)

Edge point detection
(Faster R-CNN)

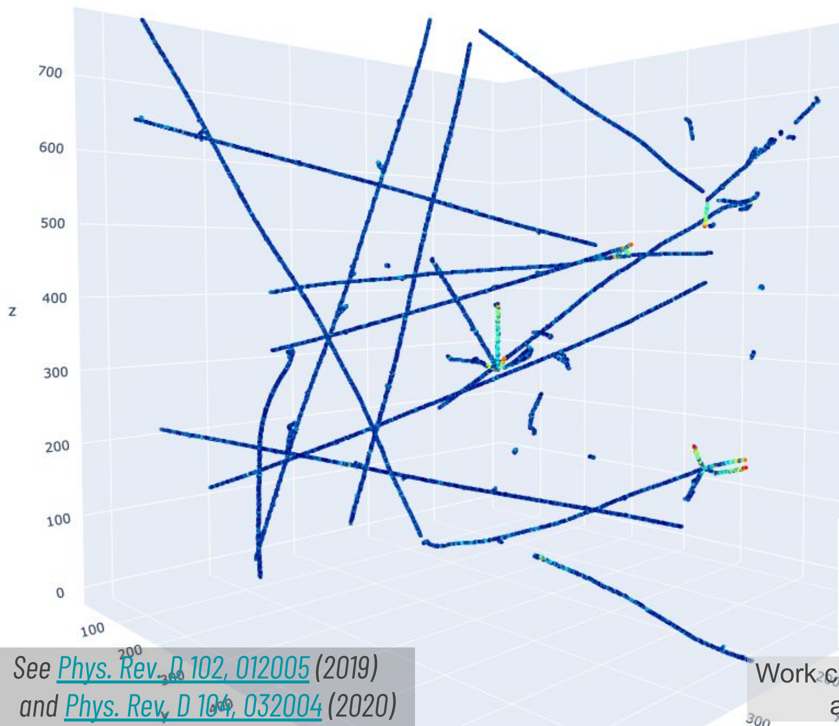
Sparse tensor operation
(Minkowski Engine)

See Phys. Rev. D 102, 012005 (2019) and Phys. Rev. D 104, 032004 (2020)

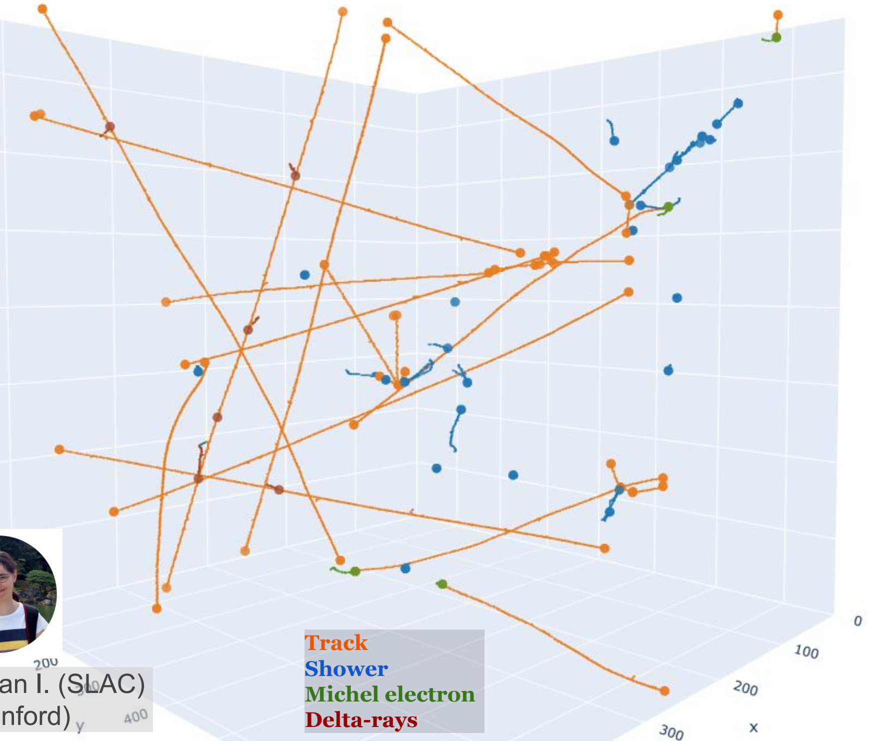
ML for Analyzing Big Image Data in Neutrino Experiments
Stage 1: input & output

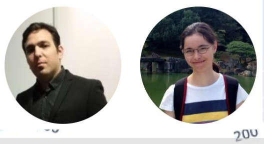
SLAC

Stage 1 Input



Stage 1 Output





See Phys. Rev. D 102, 012005 (2019)
and Phys. Rev. D 104, 032004 (2020)

Work credit (from left) Ran I. (SLAC)
and Laura D. (Stanford)

Stage 2-a: dense pixel clustering

Clustering in the embedding space

- Use CNN to learn a transformation function from the 3D voxels to the embedding space where clustering can be performed in a simple manner

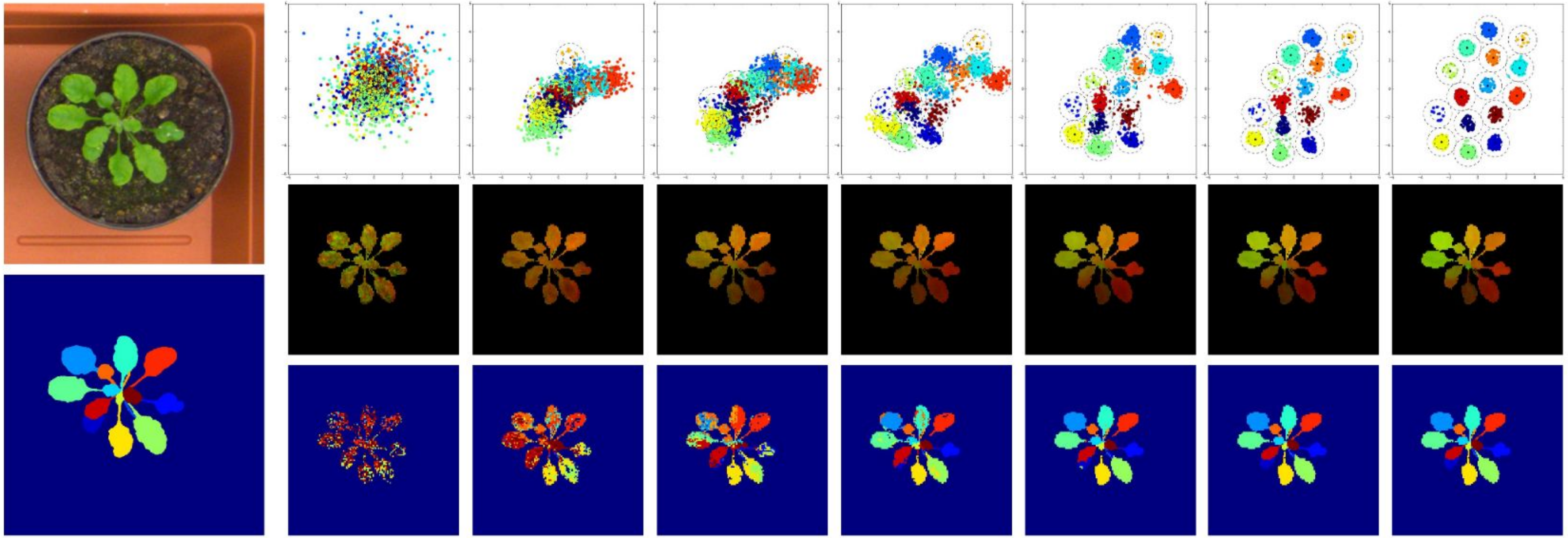
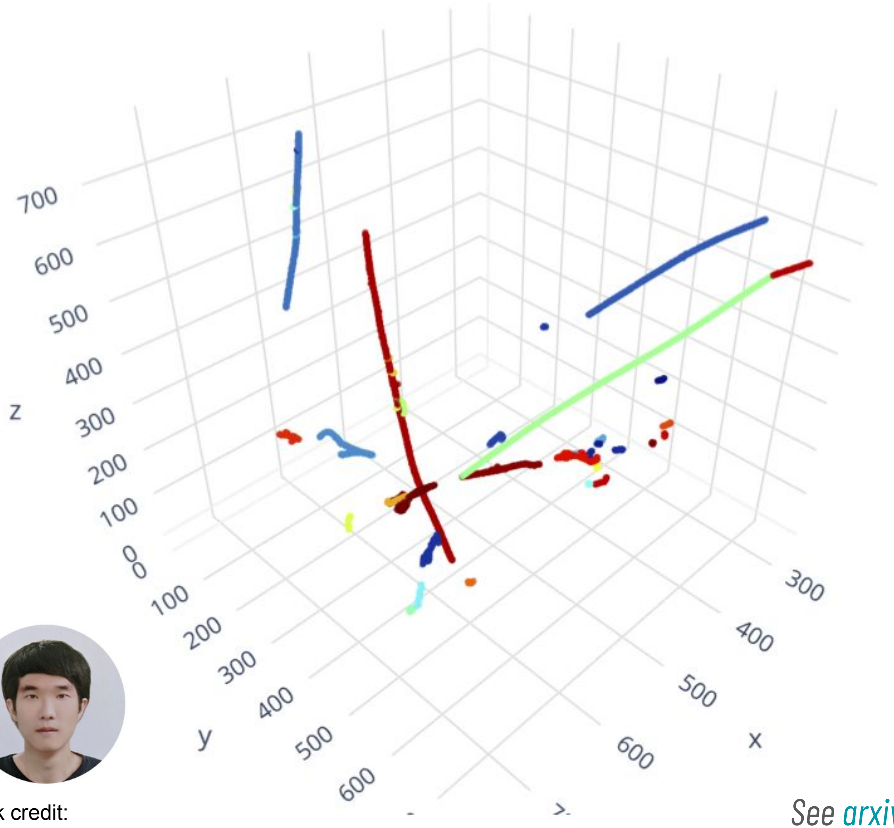
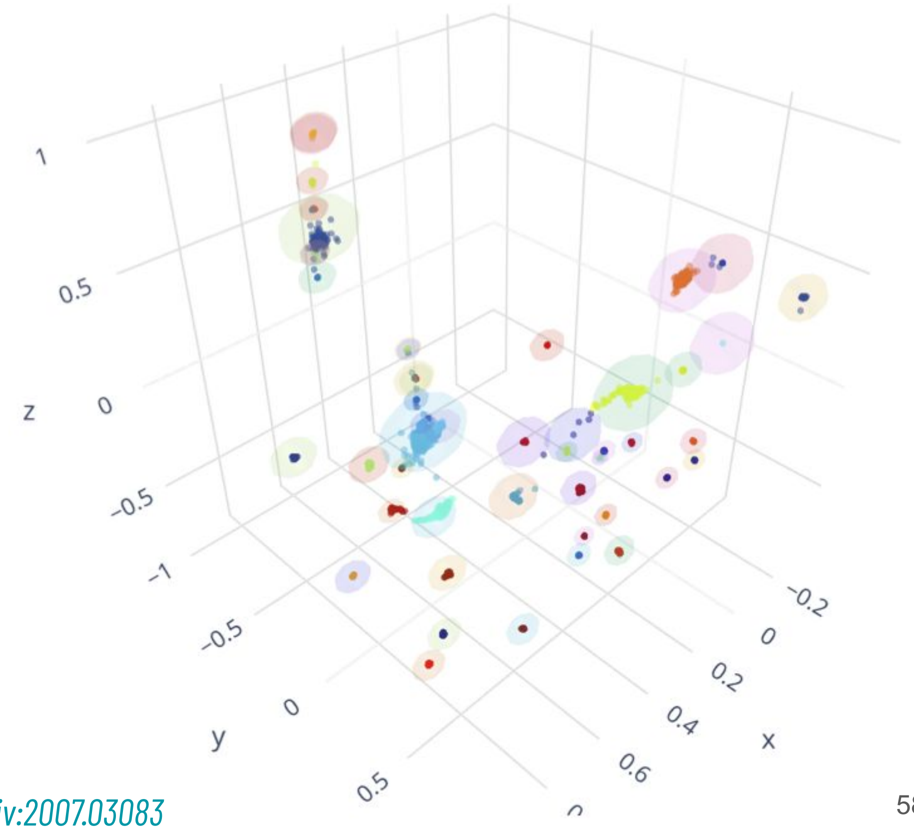


Image credit: arXiv 1708.02551

ML for Analyzing Big Image Data in Neutrino Experiments

Stage 2-a: dense pixel clustering



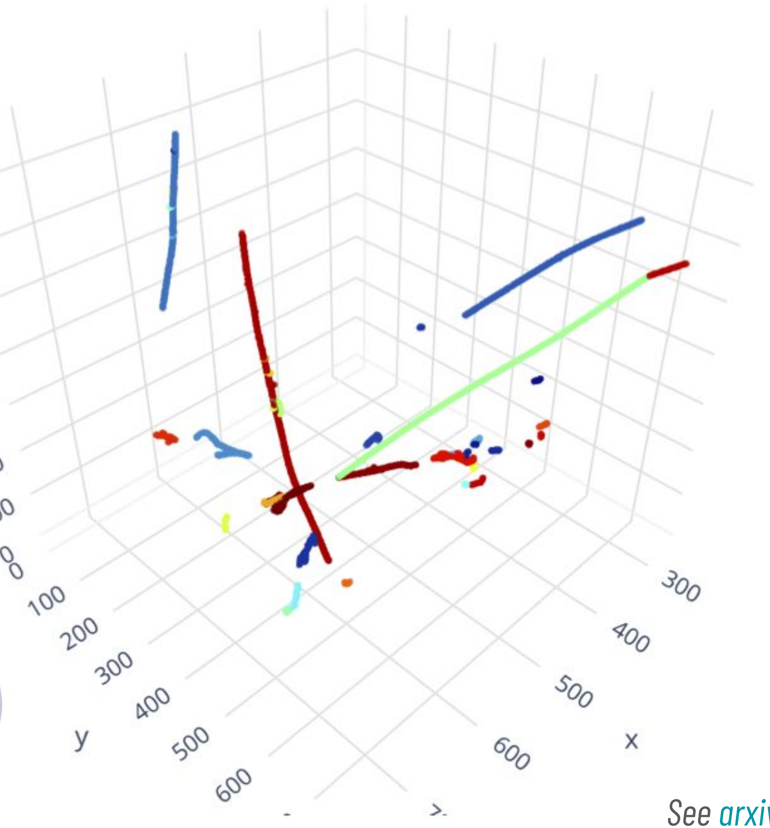


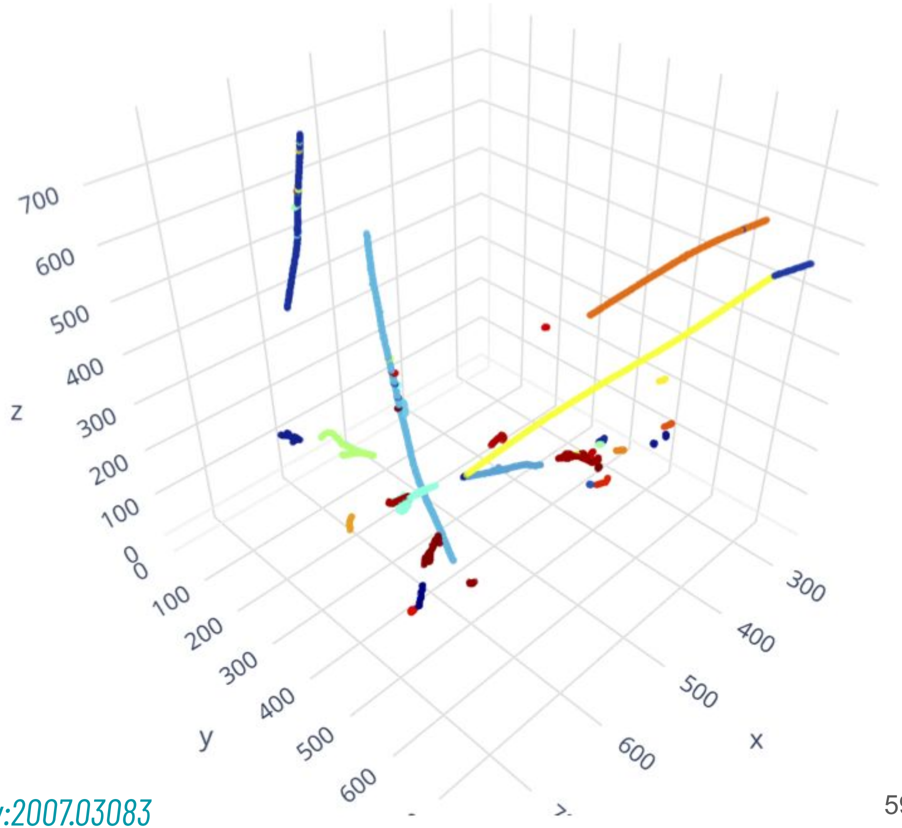
Work credit:
Dae Heun Koh (Stanford)

See [arxiv:2007.03083](https://arxiv.org/abs/2007.03083)

ML for Analyzing Big Image Data in Neutrino Experiments

Stage 2-a: dense pixel clustering





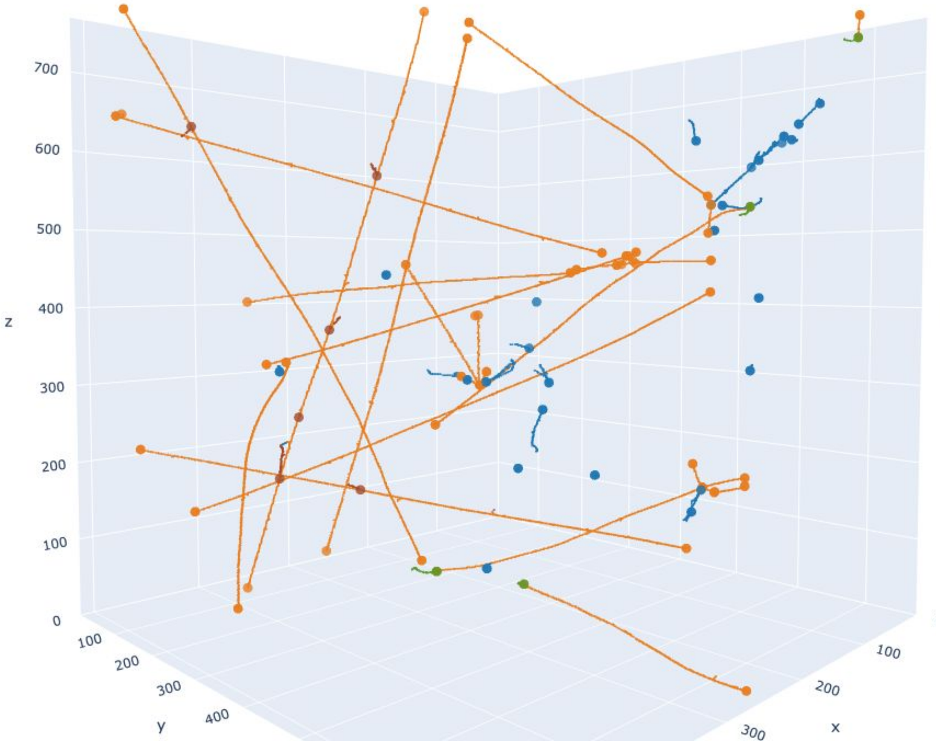
Work credit:
Dae Heun Koh (Stanford)

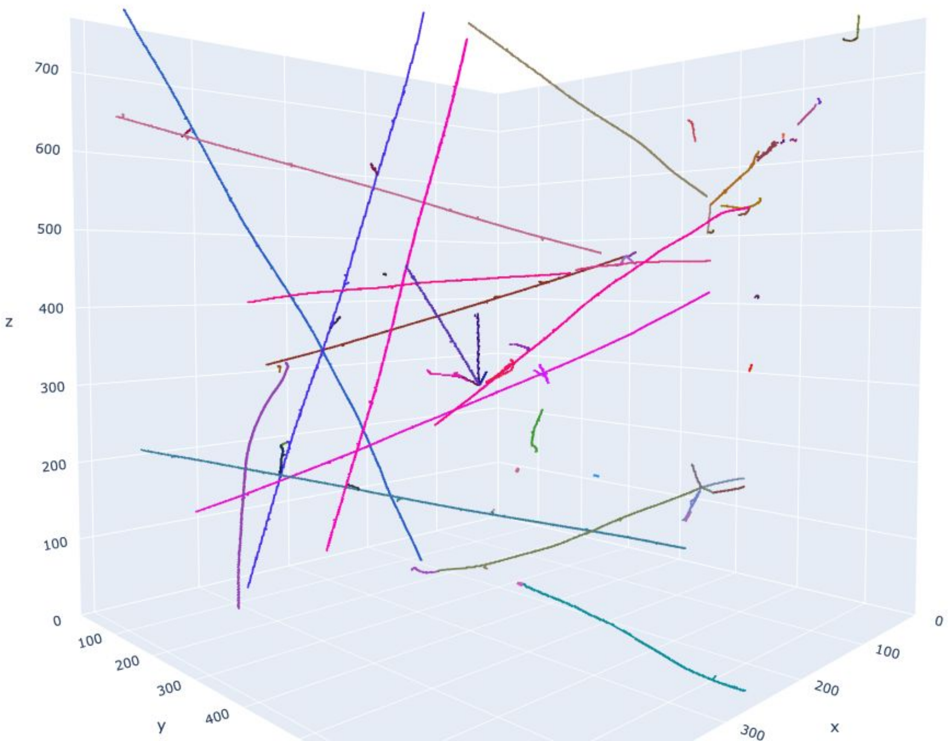
See [arxiv:2007.03083]

ML for Analyzing Big Image Data in Neutrino Experiments

Stage 2-a: input & output

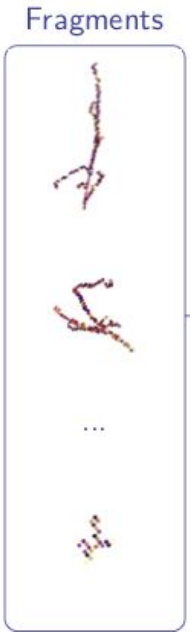
Stage 2-a Input	Stage 2-a Output

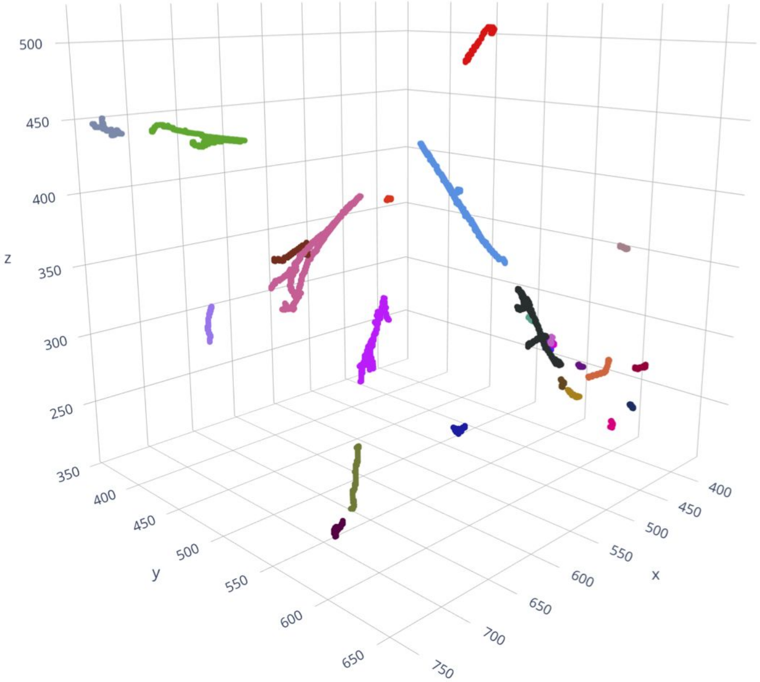




Stage 2-b: sparse fragment clustering

Identifying 1 shower … which consists of **many fragments**

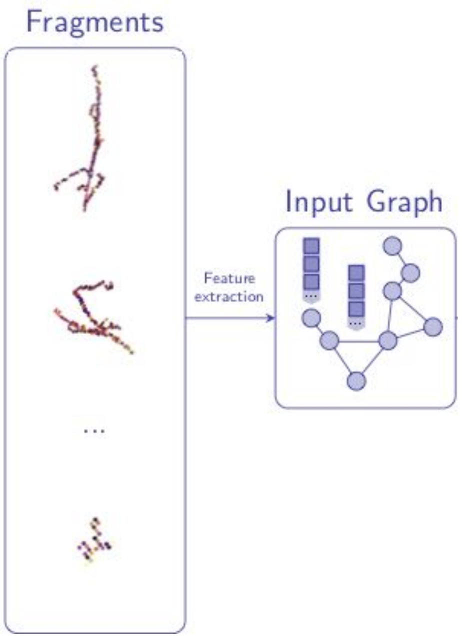


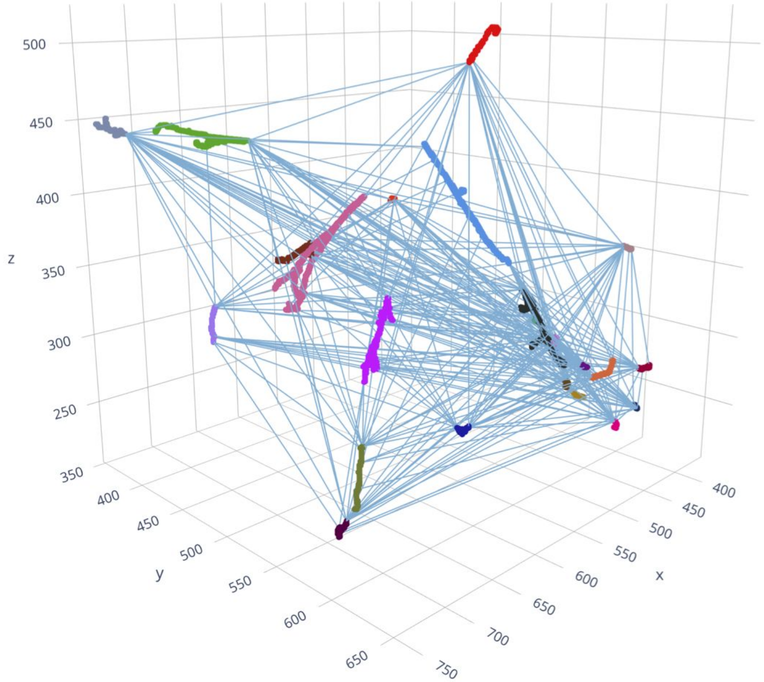


Stage 2-b: sparse fragment clustering

Identifying 1 shower ... which consists of **many fragments**

- Interpret each fragment as a graph node + edges connect nodes in the same cluster

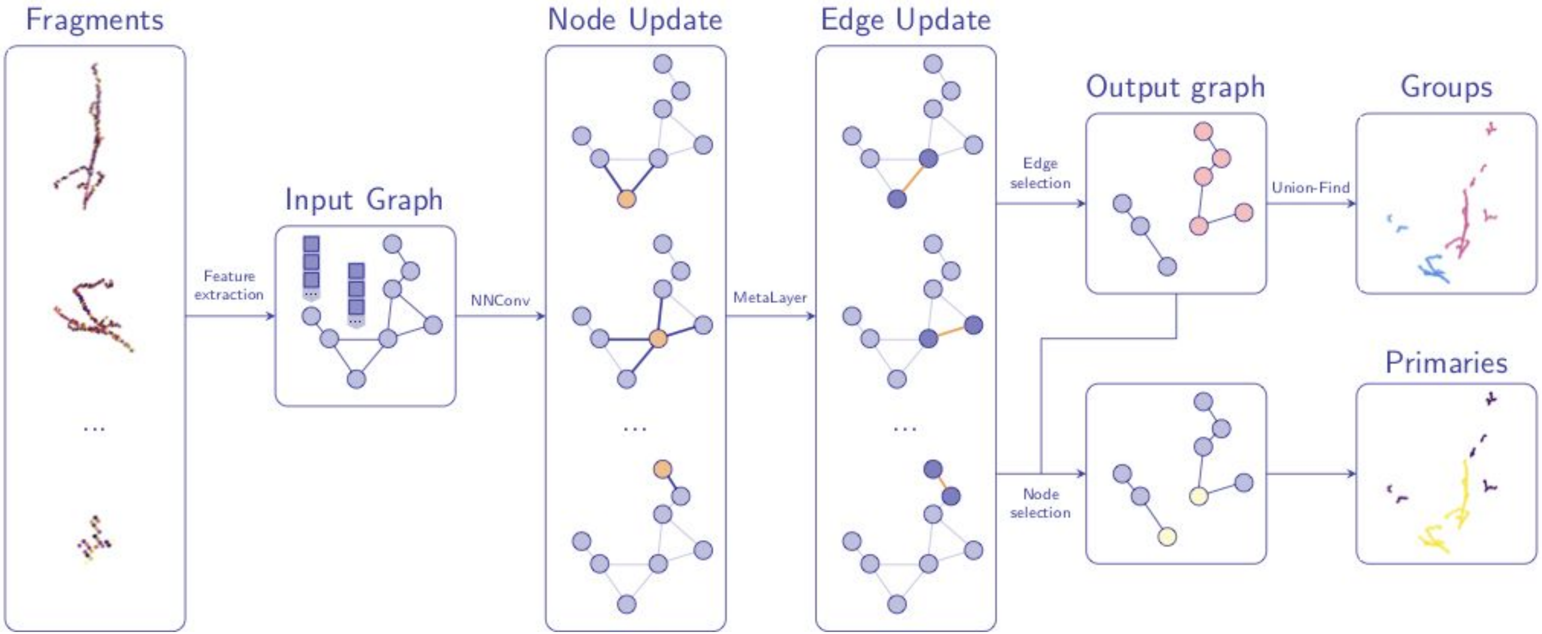




Stage 2-b: sparse fragment clustering

Identifying 1 shower … which consists of **many fragments**

- Interpret each fragment as a graph node + edges connect nodes in the same cluster
- Cast the problem to a classification of node (e.g. particle type) and edge (clustering)

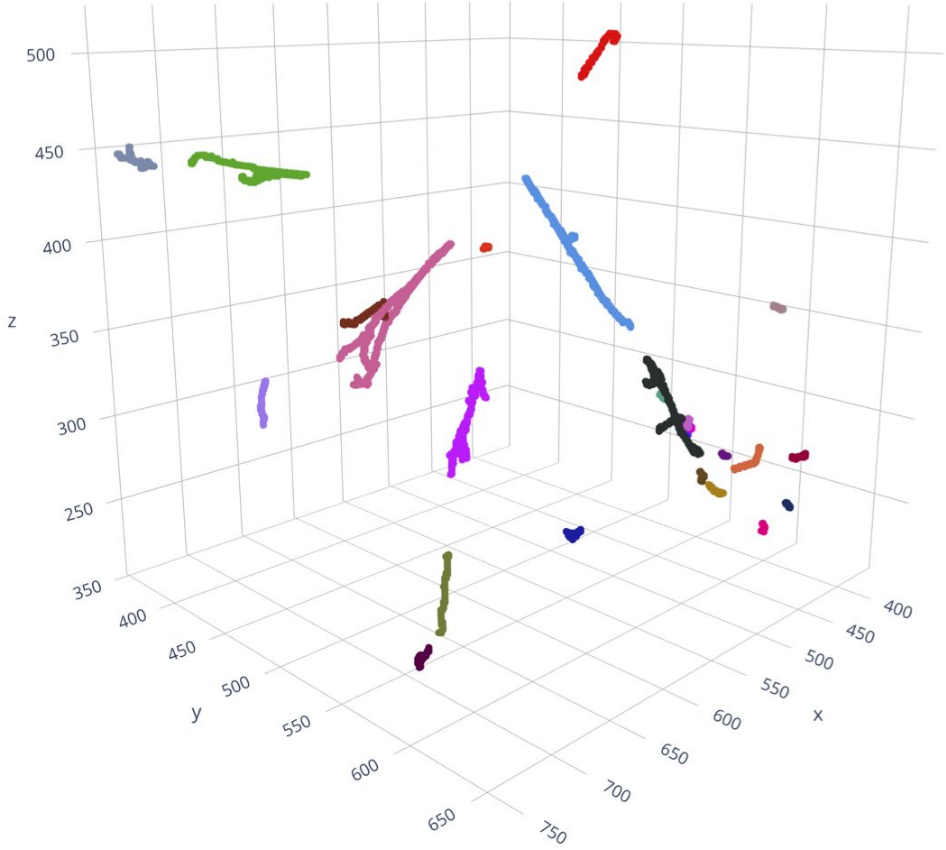


Stage 2-b: sparse fragment clustering

Graph-NN for Particle Aggregation (GrapPA)

Input:

- Fragmented EM showers



See Phys. Rev. D 104, 072004

Stage 2-b: sparse fragment clustering

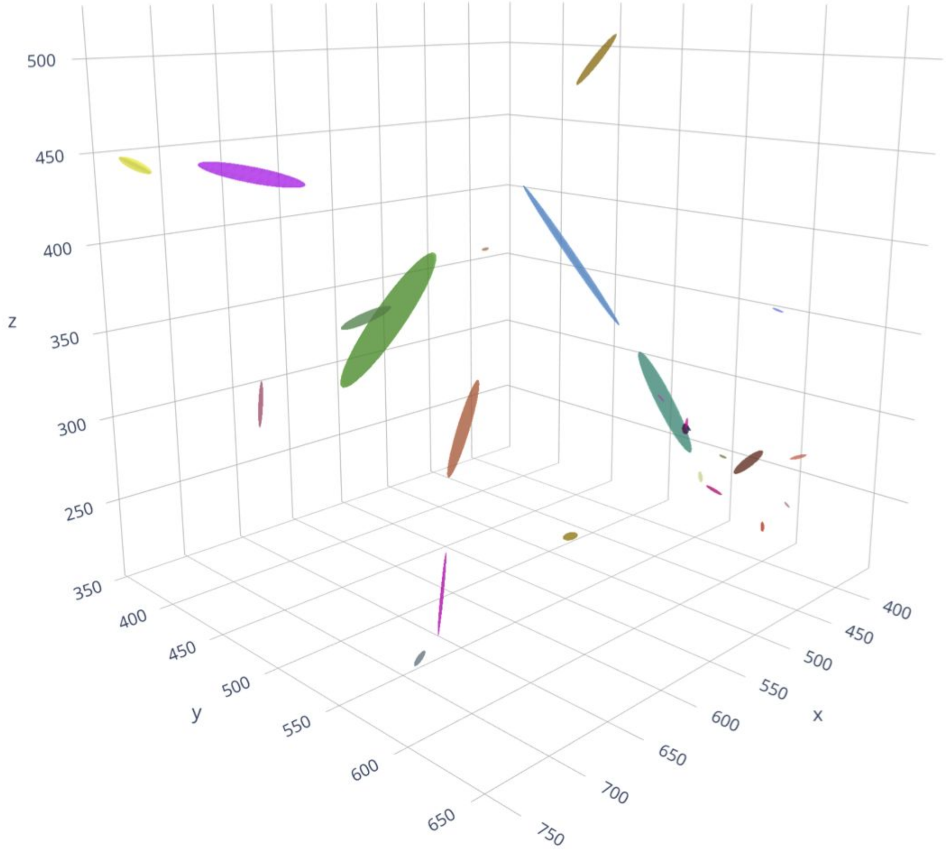
Graph-NN for Particle Aggregation (GrapPA)

Input:

- Fragmented EM showers

Node features:

- Centroid, Covariance matrix, PCA
- Start point, direction (PPN)



See Phys. Rev. D 104, 072004

ML for Analyzing Big Image Data in Neutrino Experiments
Stage 2-b: sparse fragment clustering

SLAC

Graph-NN for Particle Aggregation (GrapPA)

Input:

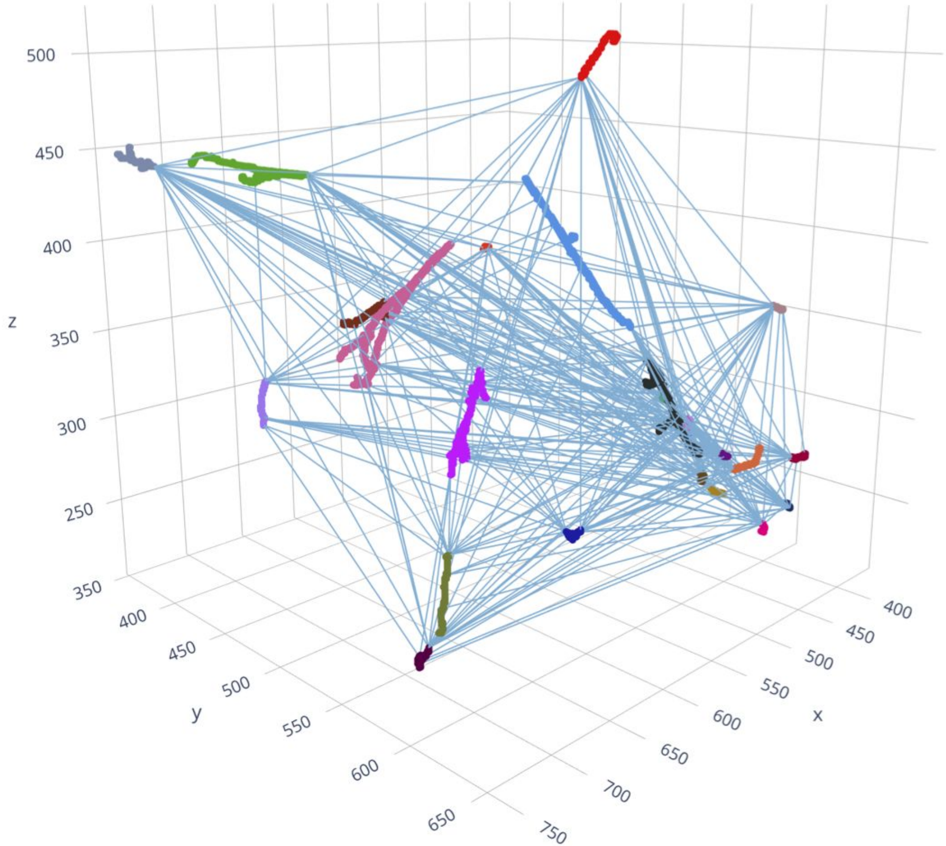
- Fragmented EM showers

Node features:

- Centroid, Covariance matrix, PCA
- Start point, direction (PPN)

Input graph:

- Connect every node with every other node (complete graph)



See Phys. Rev. D 104, 072004

ML for Analyzing Big Image Data in Neutrino Experiments
Stage 2-b: sparse fragment clustering

SLAC

Graph-NN for Particle Aggregation (GrapPA)

Input:

- Fragmented EM showers

Node features:

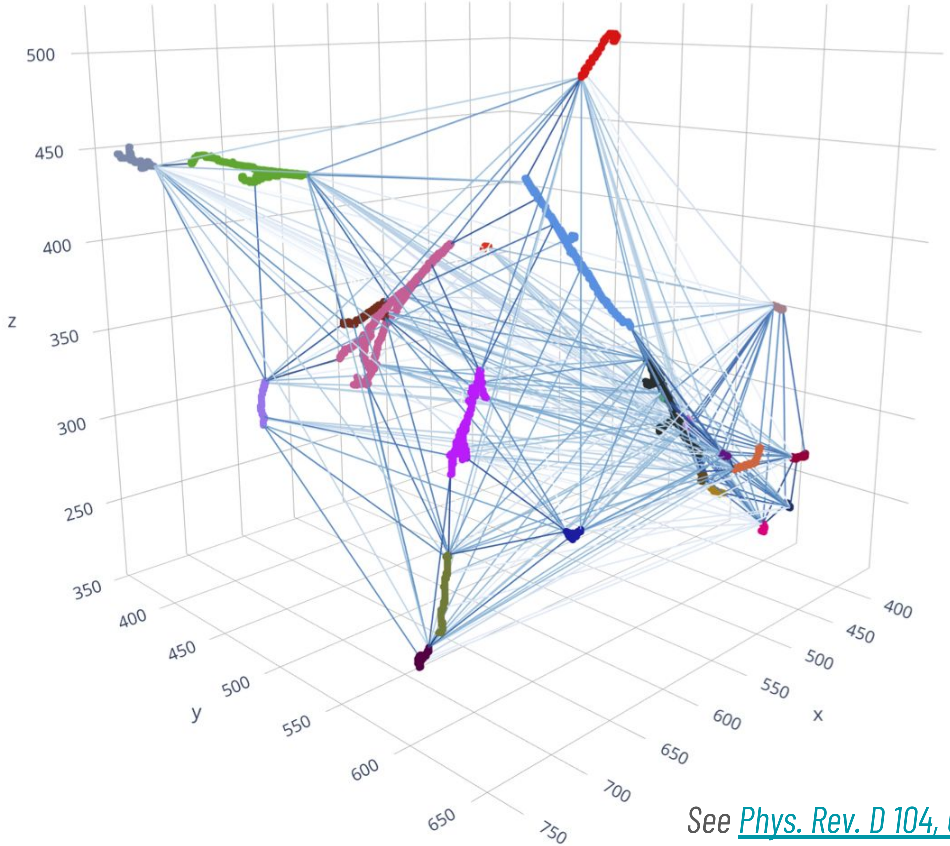
- Centroid, Covariance matrix, PCA
- Start point, direction (PPN)

Input graph:

- Connect every node with every other node (complete graph)

Edge features:

- Displacement vector
- Closest points of approach

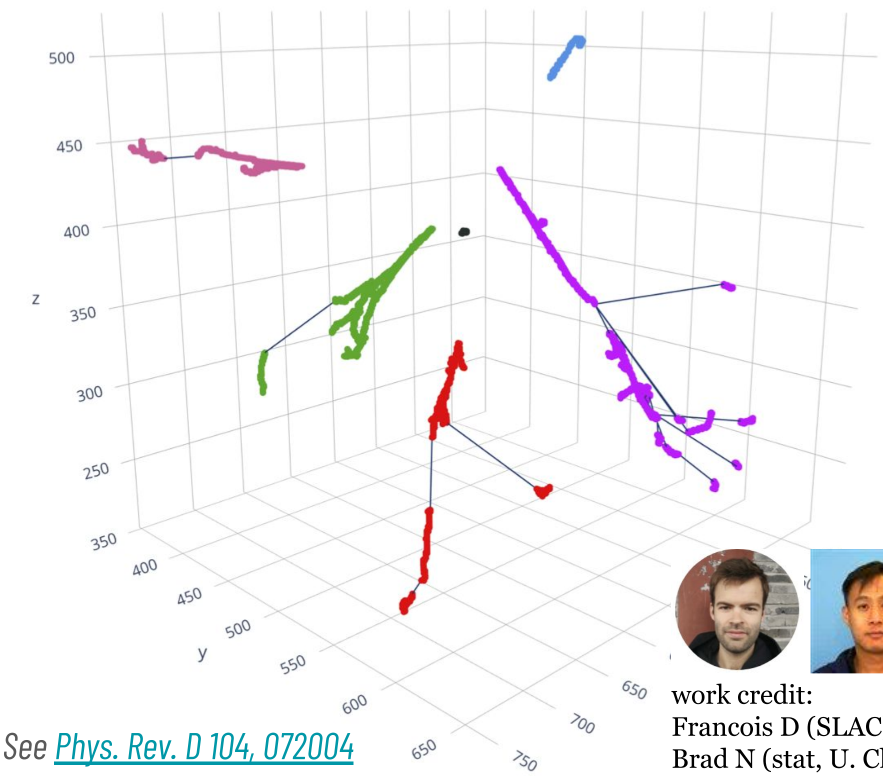


See Phys. Rev. D 104, 072004

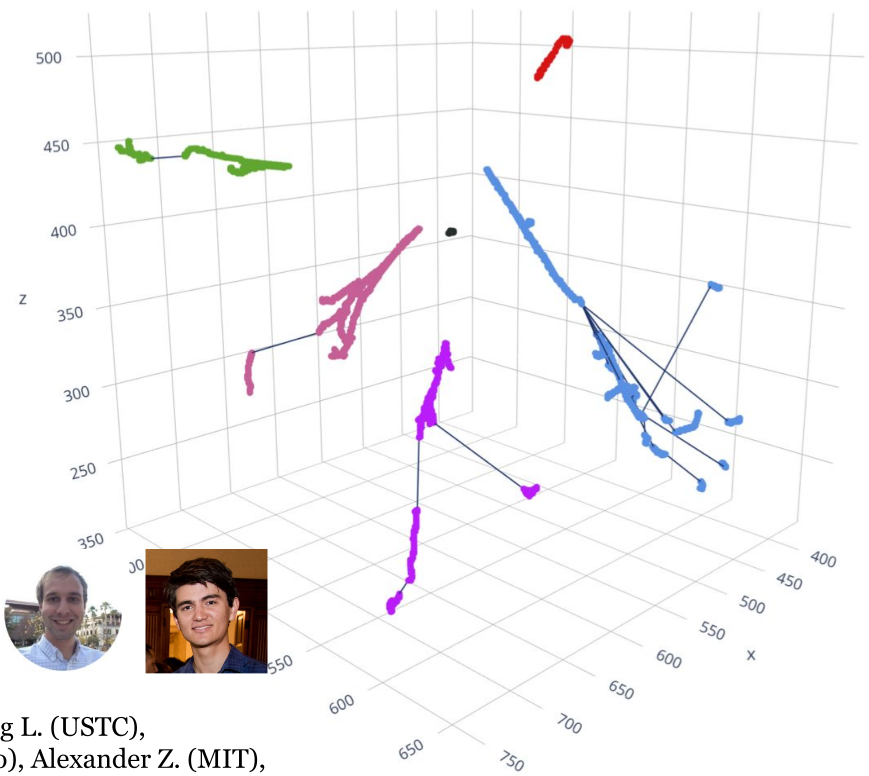
ML for Analyzing Big Image Data in Neutrino Experiments

Stage 2-b: sparse fragment clustering

Target



Prediction



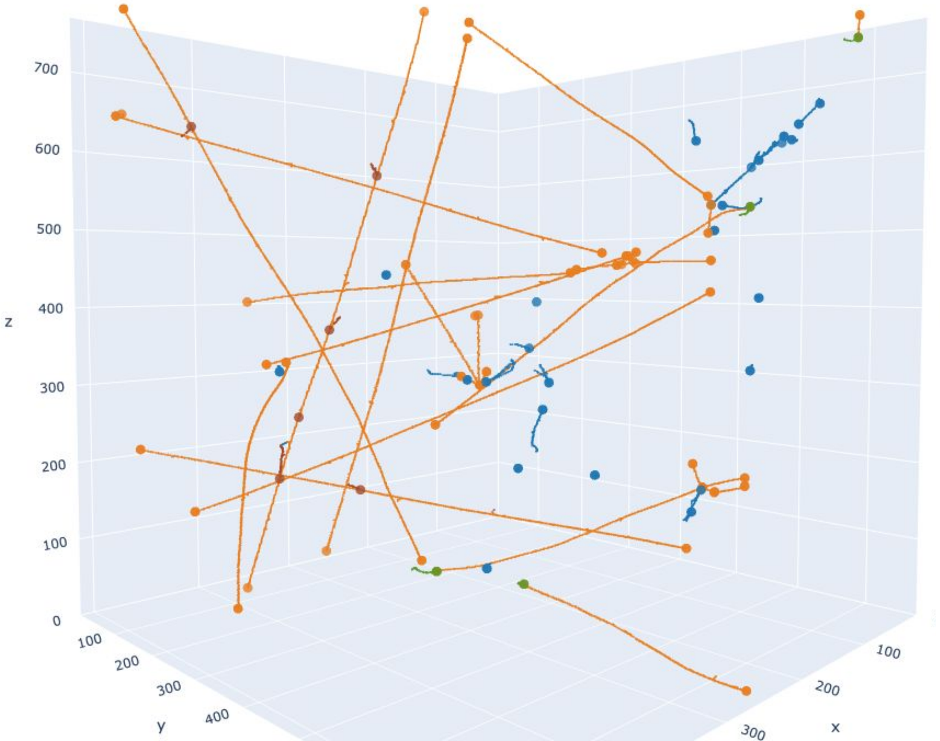
work credit:
Francois D (SLAC), Qing L. (USTC),
Brad N (stat, U. Chicago), Alexander Z. (MIT),

See Phys. Rev. D 104, 072004

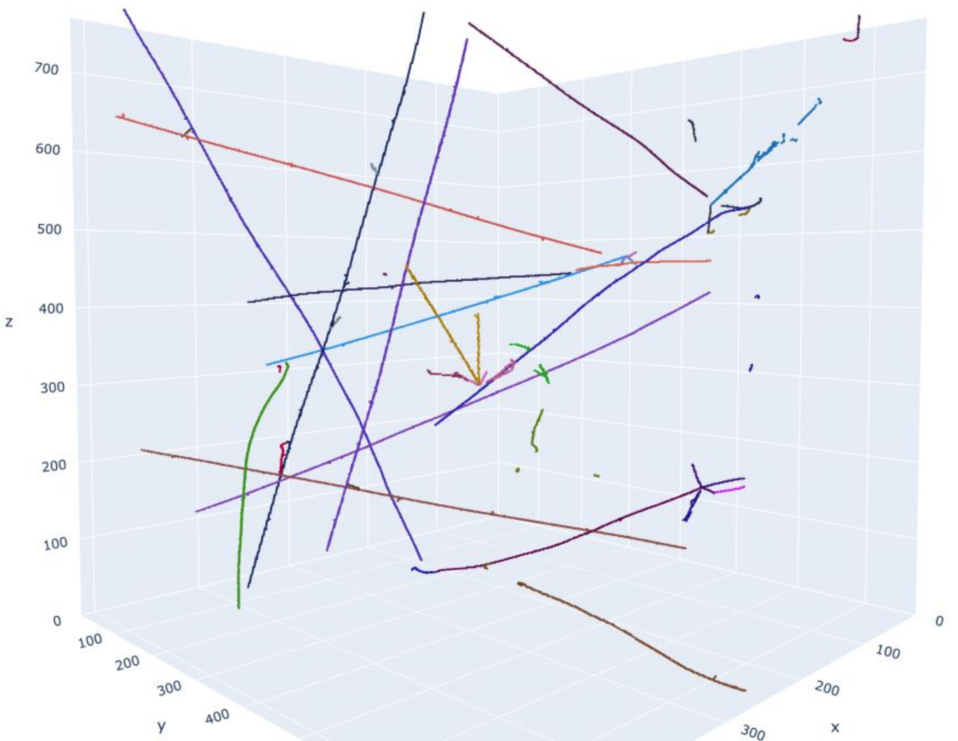
ML for Analyzing Big Image Data in Neutrino Experiments

Stage 2: input & output

Stage 2 Input

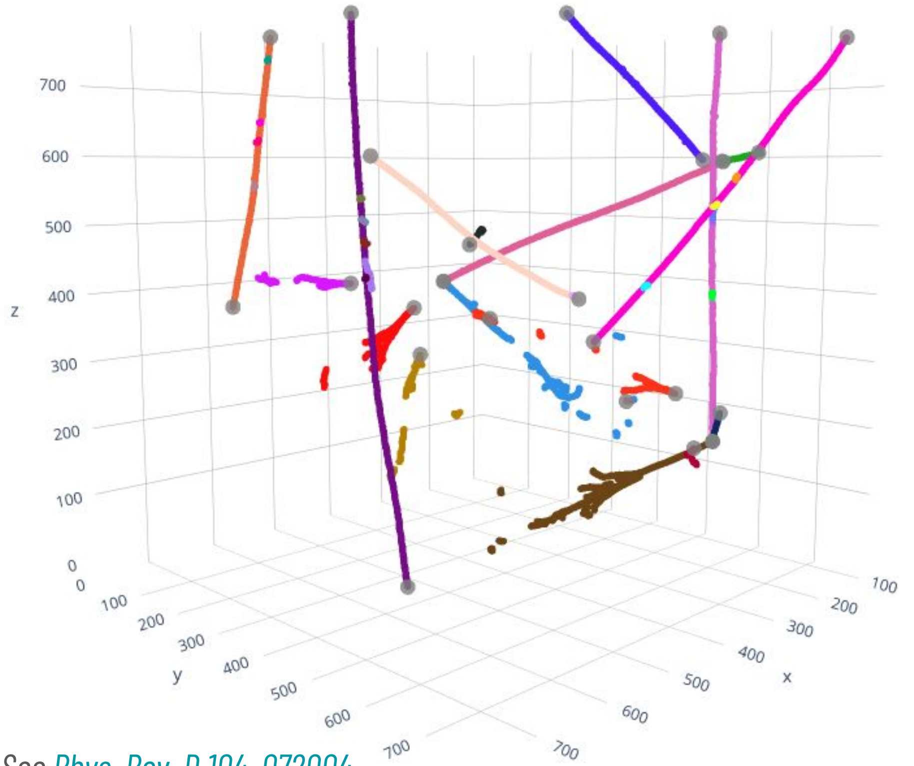


Stage 2 Output



ML for Analyzing Big Image Data in Neutrino Experiments

Stage 3: clustering of particles into an event



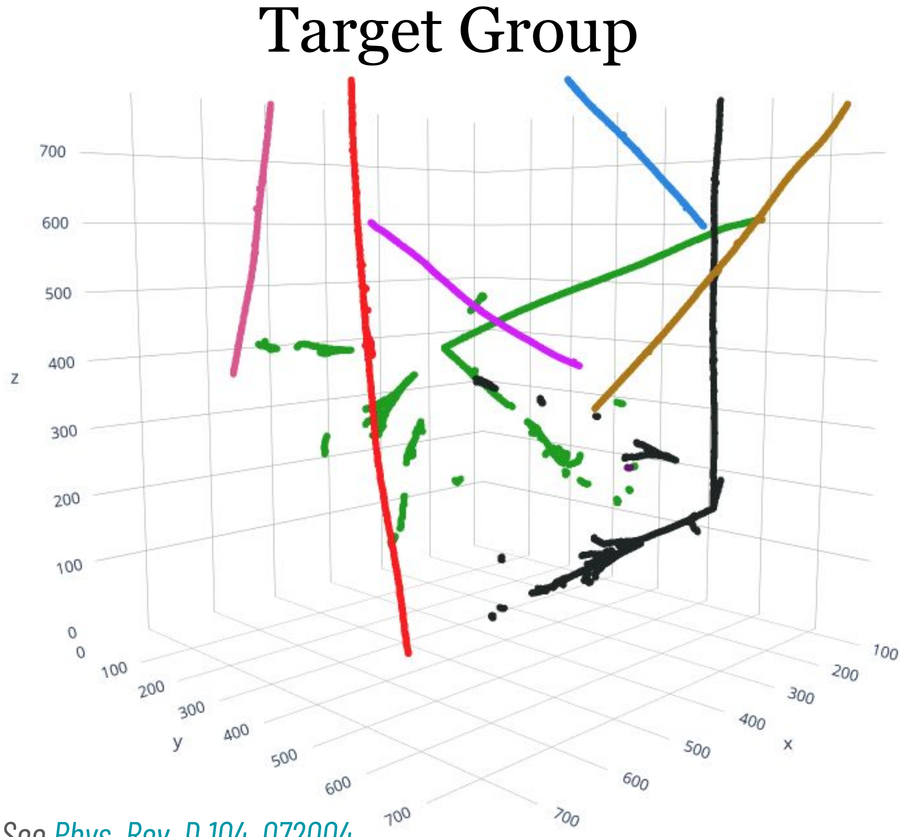
See [Phys. Rev. D 104, 072004]

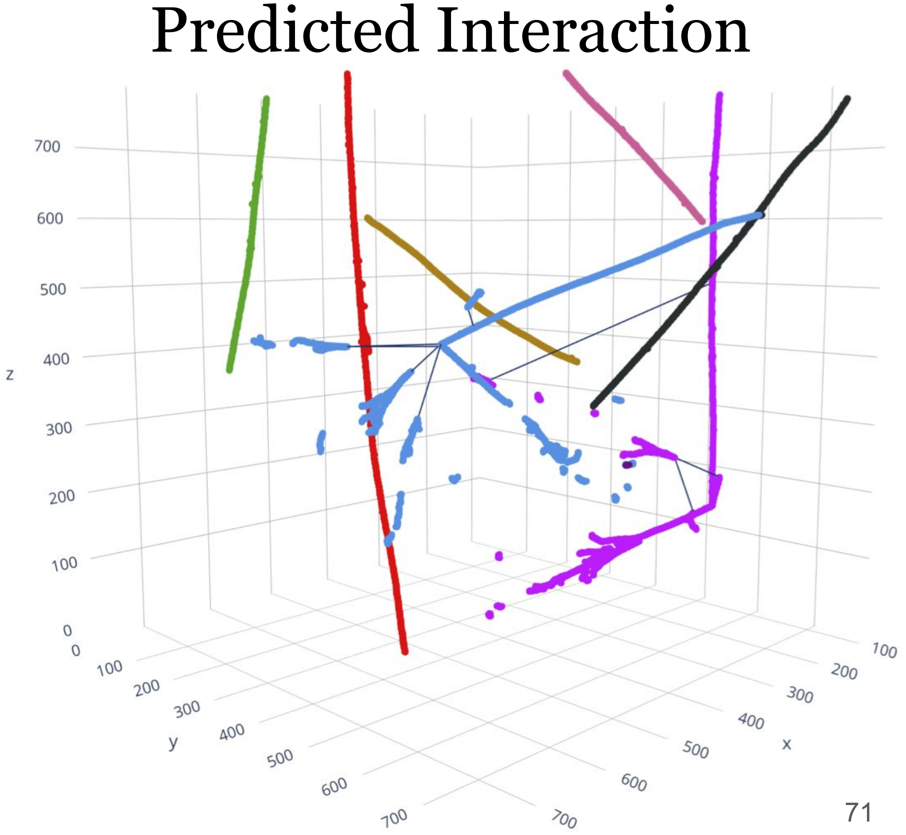
Identifying Each Interaction?

Grouping task = re-use GrapPA!

- Interaction = a group of particles that shared the same origin (i.e. neutrino interaction)
- Edge classification to identify an interaction
- Node classification for particle type ID

ML for Analyzing Big Image Data in Neutrino Experiments
Stage 3: clustering of particles into an event

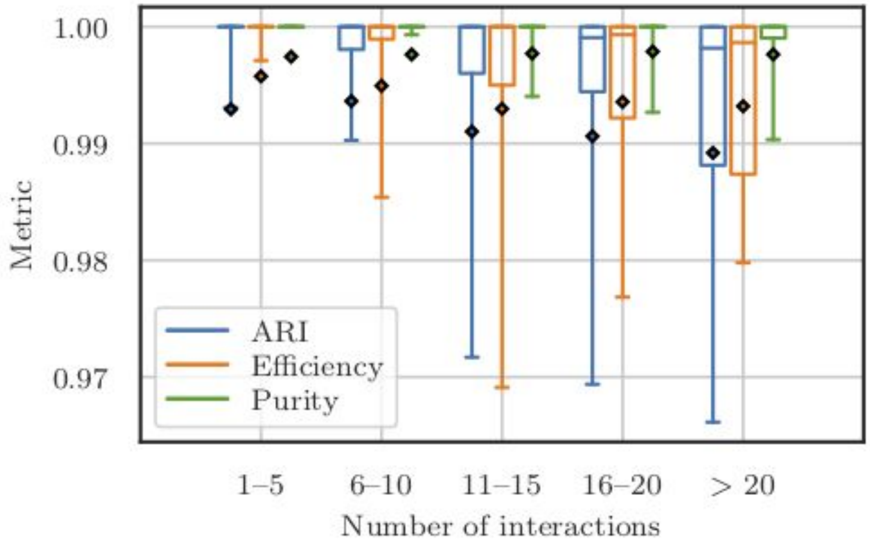




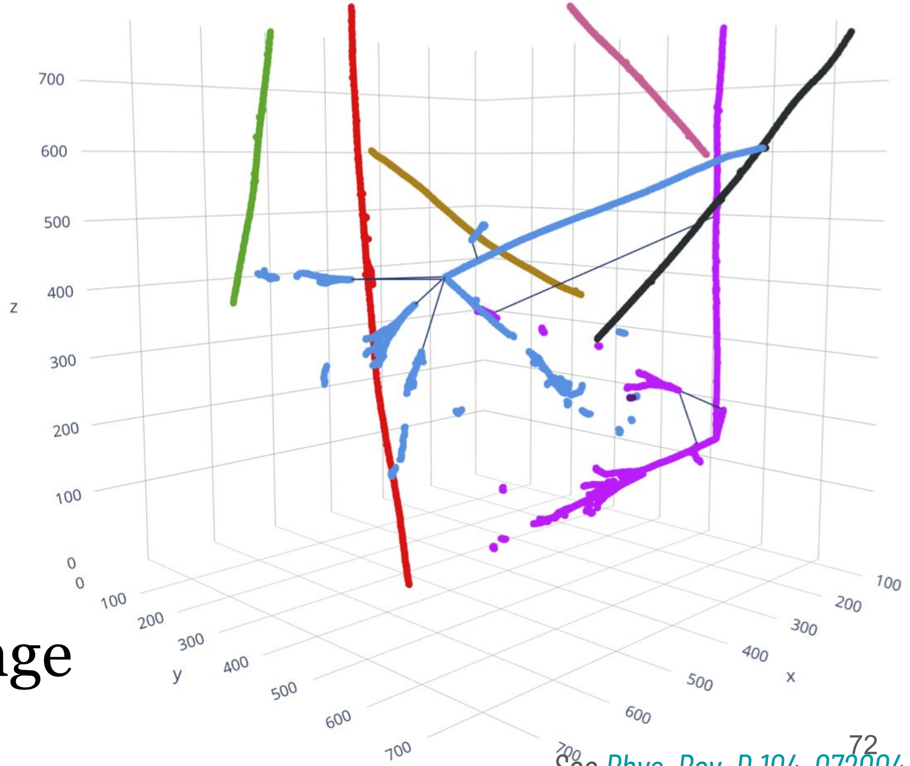
See Phys. Rev. D 104, 072004

ML for Analyzing Big Image Data in Neutrino Experiments

Stage 3: clustering of particles into an event



Predicted Interaction



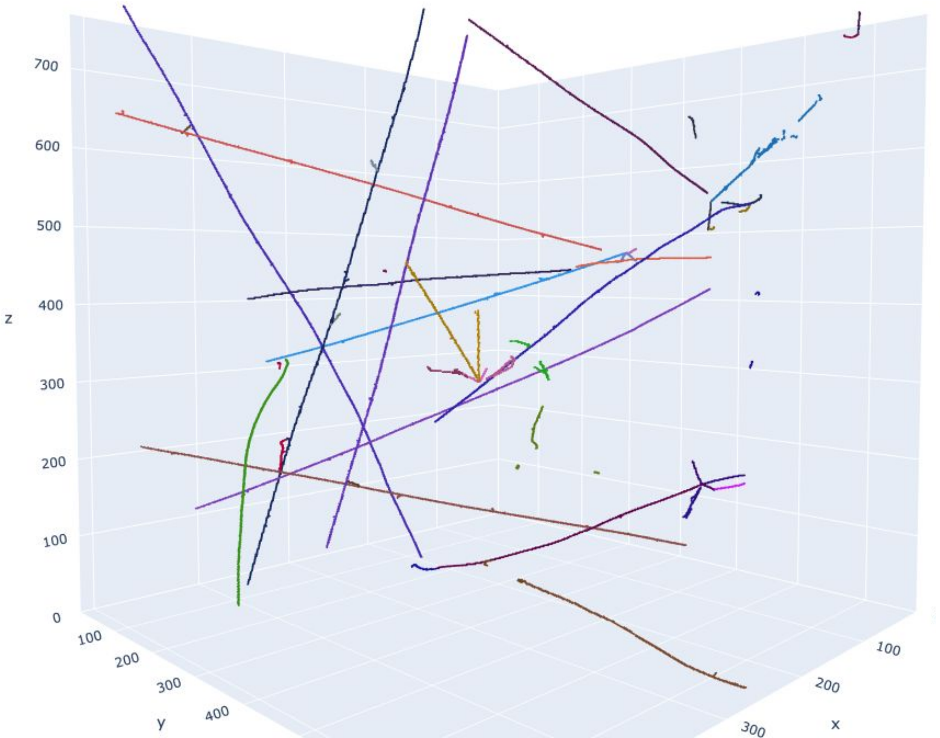
Promising result to address DUNE-ND reconstruction challenge (~20 neutrino pile-up)

See Phys. Rev. D 104, 072004

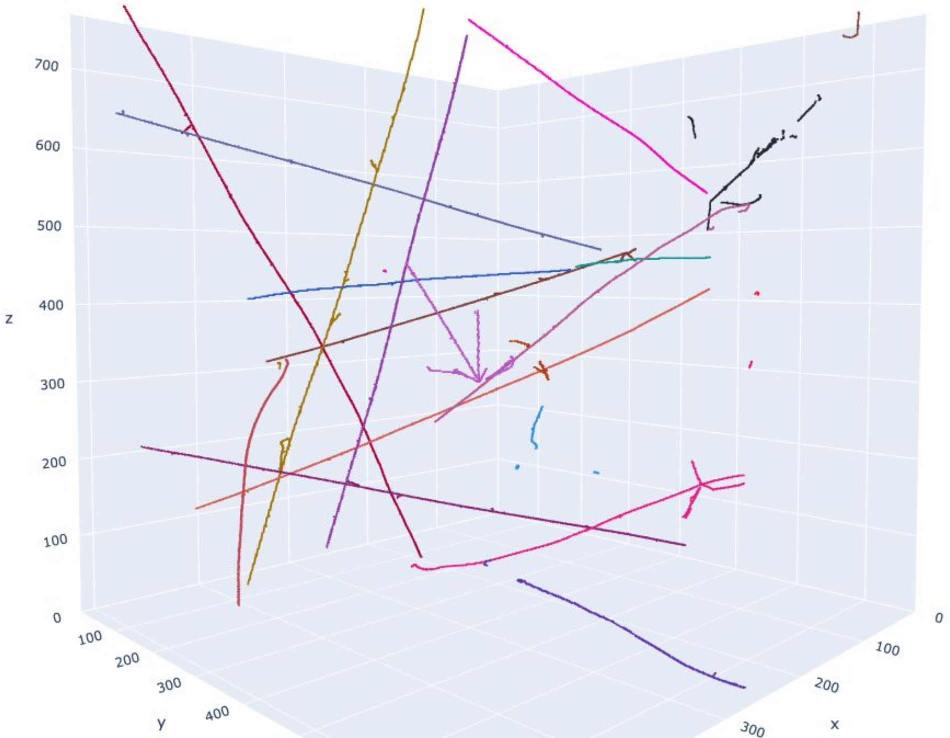
ML for Analyzing Big Image Data in Neutrino Experiments
Stage 3: input & output

SLAC

Stage 3 Input



Stage 3 Output



Example Application
for
Modeling Detector Physics

ML for Analyzing Big Image Data in Neutrino Experiments

Differentiable detector simulator

SLAC

Photo-multiplier tubes (PMTs) detect scintillation photons

Optical Photon Transport

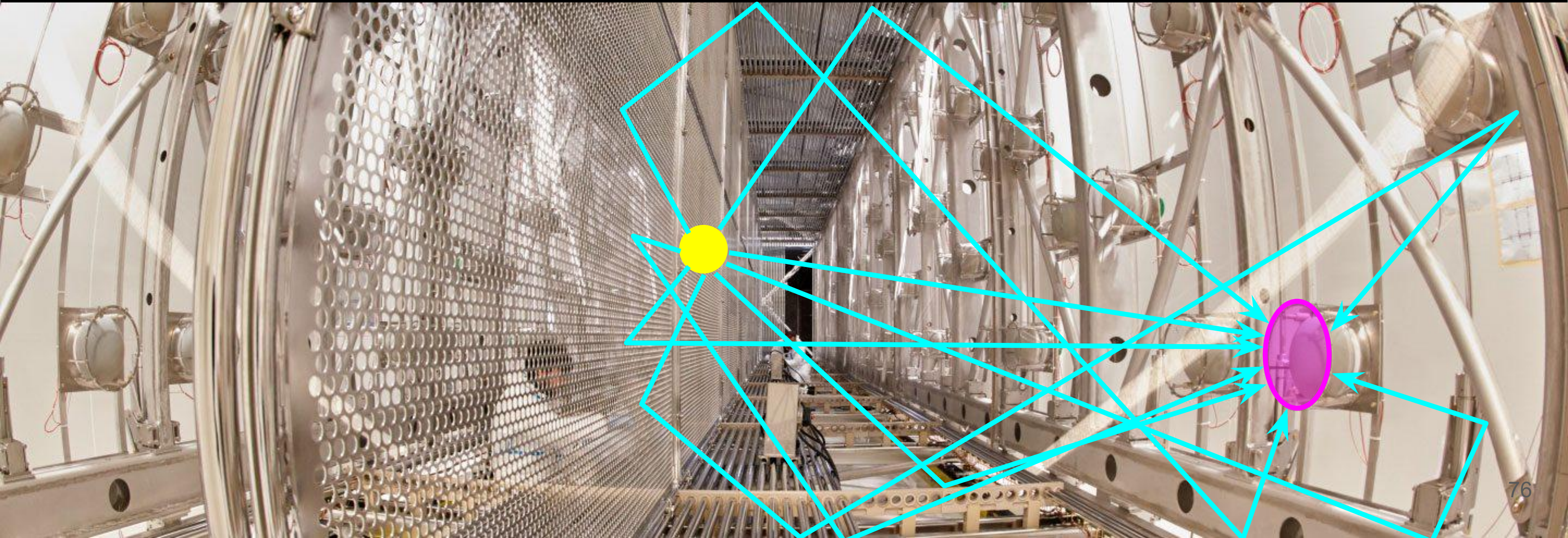
ML for Analyzing Big Image Data in Neutrino Experiments

Differentiable detector simulator

Photo-multiplier tubes (PMTs) detect scintillation photons produced isotropically from an Argon atom

1 meter muon produces **> 4M photons**

Optical Photon Transport



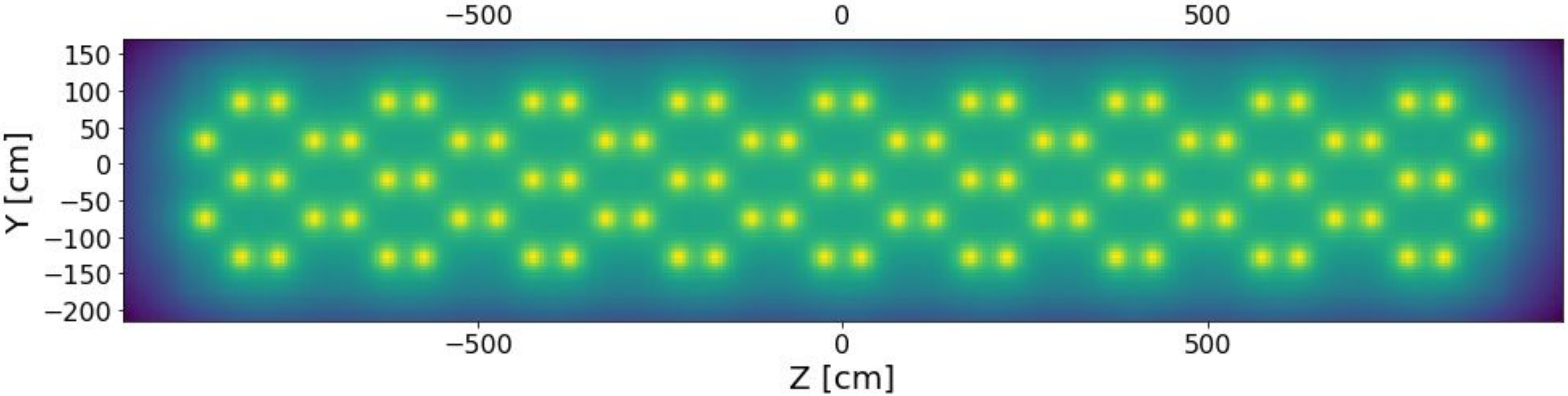
Differentiable detector simulator

SLAC

A marginalized "**Visibility Map**" for 3D voxelized volume used to estimate photon count at each PMT

Issue: static, not scalable

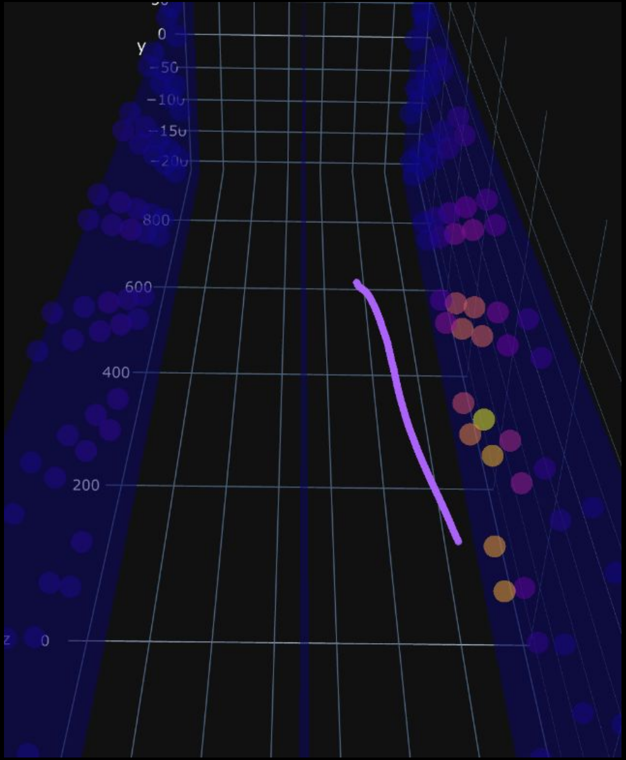
Optical Photon Transport



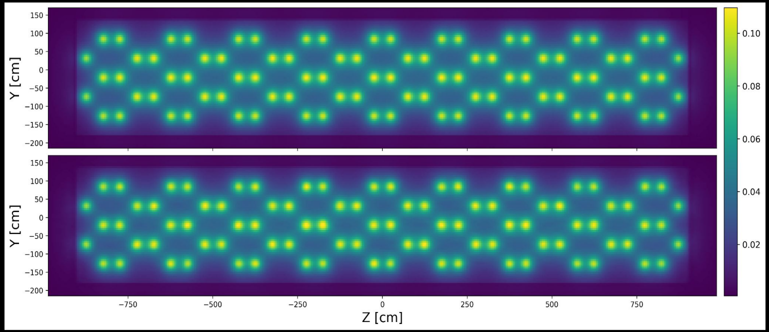
Example: ICARUS detector, 2D slice of a 3D map

ML for Analyzing Big Image Data in Neutrino Experiments

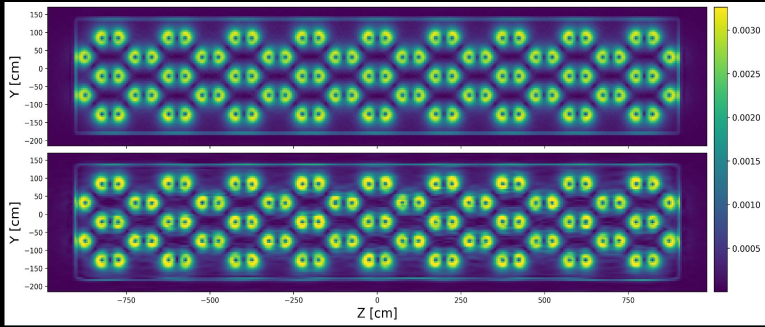
Differentiable detector simulator



Static map (top) v.s. SIREN



Gradient map (top, sobel filter) v.s. SIREN

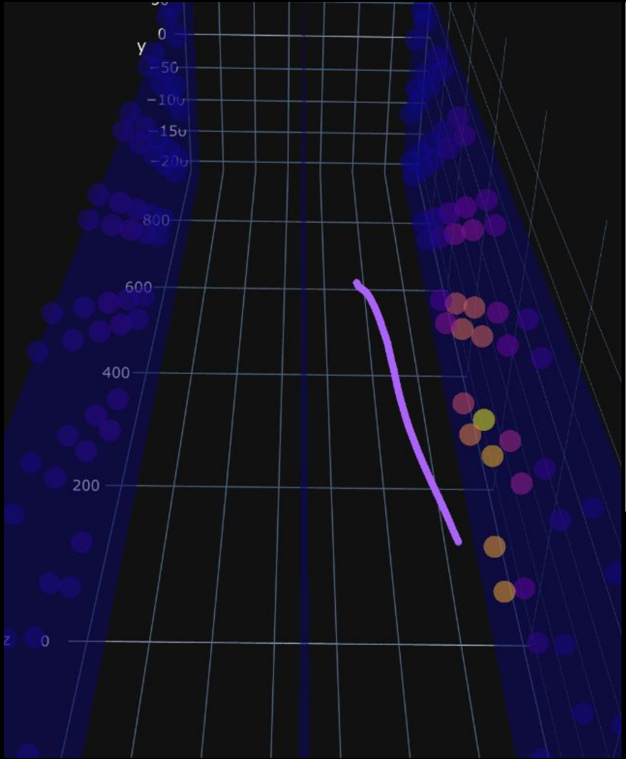


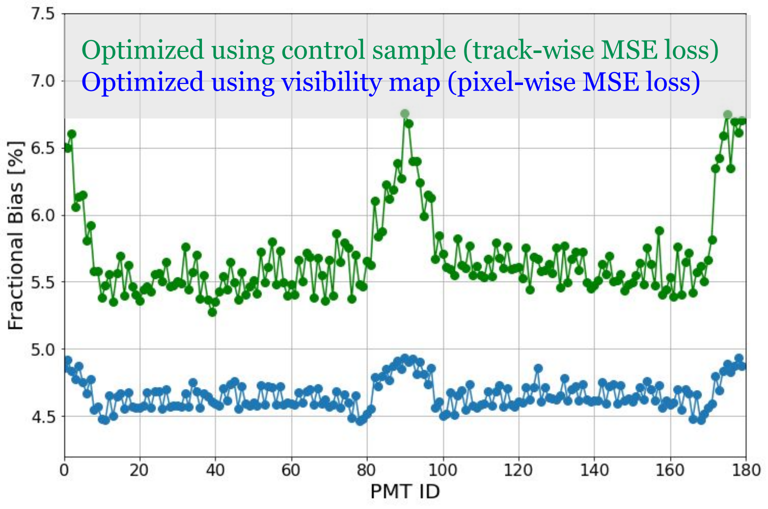
Optical Photon Transport using **Differentiable Surrogate (SIREN)**

Neural scene representation (alternative: NeRF inc. differentiable rendering)

ML for Analyzing Big Image Data in Neutrino Experiments

Differentiable detector simulator





Work credit (from left): Olivia P. (UC Berkeley), Minjie L. (SLAC), Patrick T. (SLAC), , Gordon W. (Stanford CS), Chuan L. (Lambda Labs)

Optical Photon Transport using **Differentiable Surrogate (SIREN)**

Neural scene representation (alternative: NeRF inc. differentiable rendering)

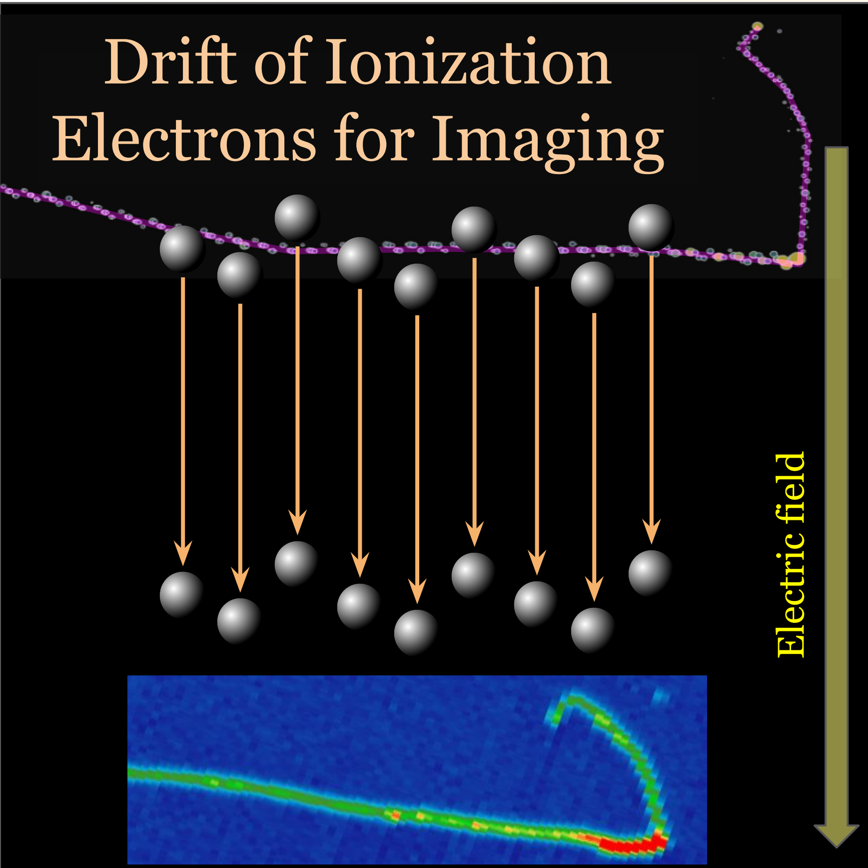
Drift of Ionization Electrons for Imaging

Differentiable detector simulator

Drift of Ionization Electrons for Imaging

1. Particle ionize Argon

Differentiable detector simulator



1. Particle ionize Argon

2. Ionization electron drift in E-field at a constant velocity, some charge lost due to capture

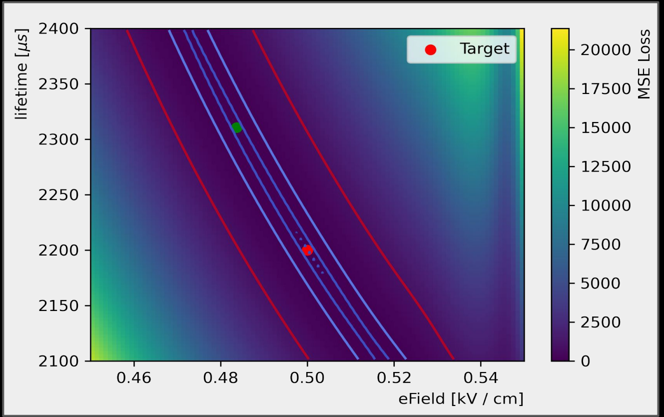
3. Imaging by charge-sensitive plane (detectors) at the anode

Tuning simulation = extract physics model parameter values from data

ML for Analyzing Big Image Data in Neutrino Experiments

Differentiable detector simulator

SLAC

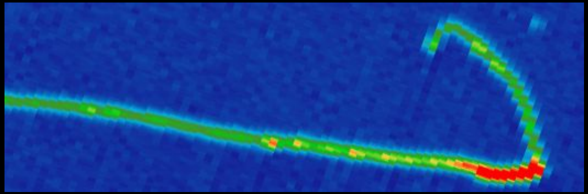


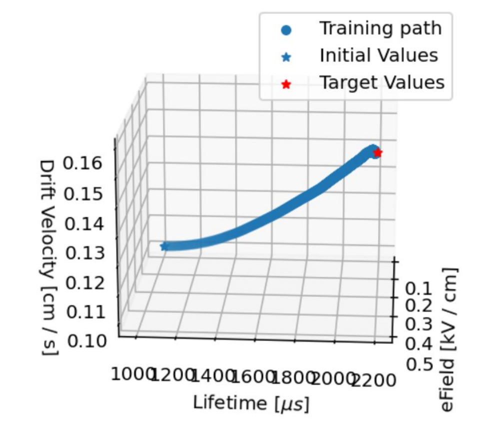
Drift of Ionization Electrons for Imaging

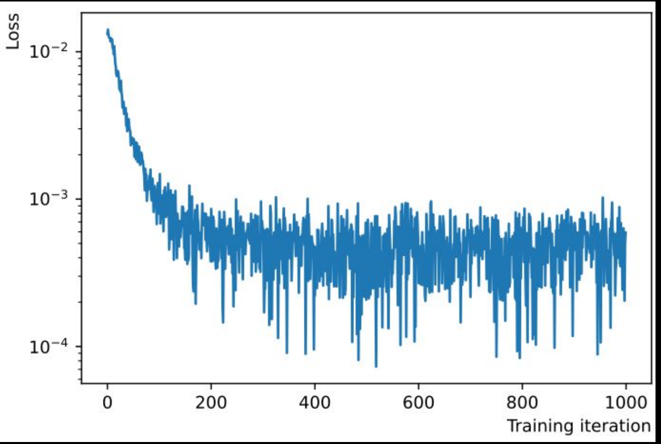
Differentiable Simulator

using explicit gradient calculation using AD-enabled tools (JAX/Pytorch)

Work credit due (from left):
SLAC-ML: Youssef N., Sean G., Daniel R.
SLAC-neutrino: Yifan C.
LBNL-neutrino: Roberto S.

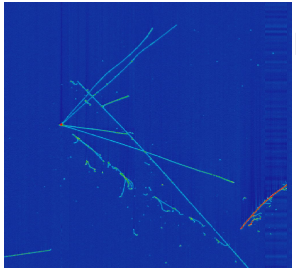






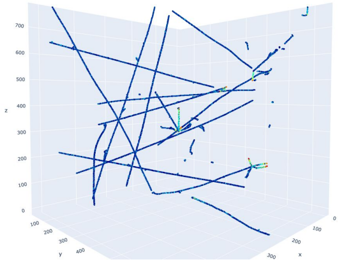
Simulation v.s. Reconstruction/Calibration

“**Reconstruction**” is a process of inferring
a high(er) level physics quantities from raw data.



Detector Output
(ADC)

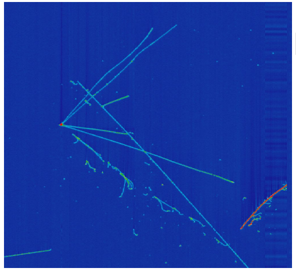
Reconstruction



Reconstructed

Simulation v.s. Reconstruction/Calibration

"**Calibration**" infers (part of) nuisance parameters to infer target physics analysis, often using (part of) reconstructed information

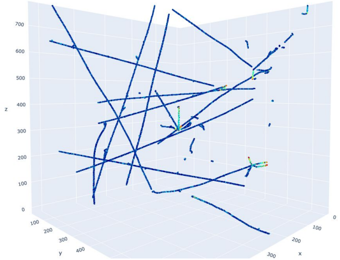


Detector Output (ADC)

Reconstruction

Detector physics model parameters

Calibration



Reconstructed (Calibrated dE/dX)

Physics knowledge (models)

$$\mathscr{R}_{\mathrm{ICARUS}} = \frac{A_B}{1 + k_B \cdot (dE/dx)/\mathscr{E}}$$

$$Q = Q_0 \exp(-v_{\mathrm{drift}} t/\tau)$$

$$\sigma_t^2(t) \simeq \sigma_t^2(0) + \left(\frac{2D_L}{v_d^2}\right) t$$

...

Simulation v.s. Reconstruction/Calibration

Detector Simulation

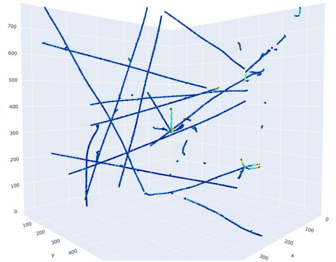
$$\mathscr{R}_{\text{ICARUS}} = \frac{A_B}{1 + k_B \cdot (dE/dx)/\mathscr{E}}$$

$$Q = Q_0 \exp(-v_{\text{drift}} t/\tau)$$

$$\sigma_t^2(t) \simeq \sigma_t^2(0) + \left(\frac{2D_L}{v_d^2}\right) t$$

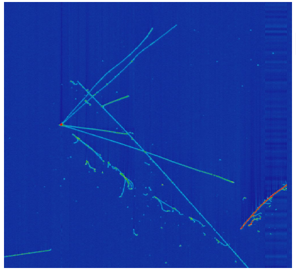
...

Simulation Input
(true dE/dX)



"**Simulation**" takes in target physics information and convolve effects associated with the measurements (i.e. nuisance)

Physics knowledge
(models)



Detector Output
(ADC)

Reconstruction

Detector physics model parameters

Calibration

Reconstructed
(dE/dX)

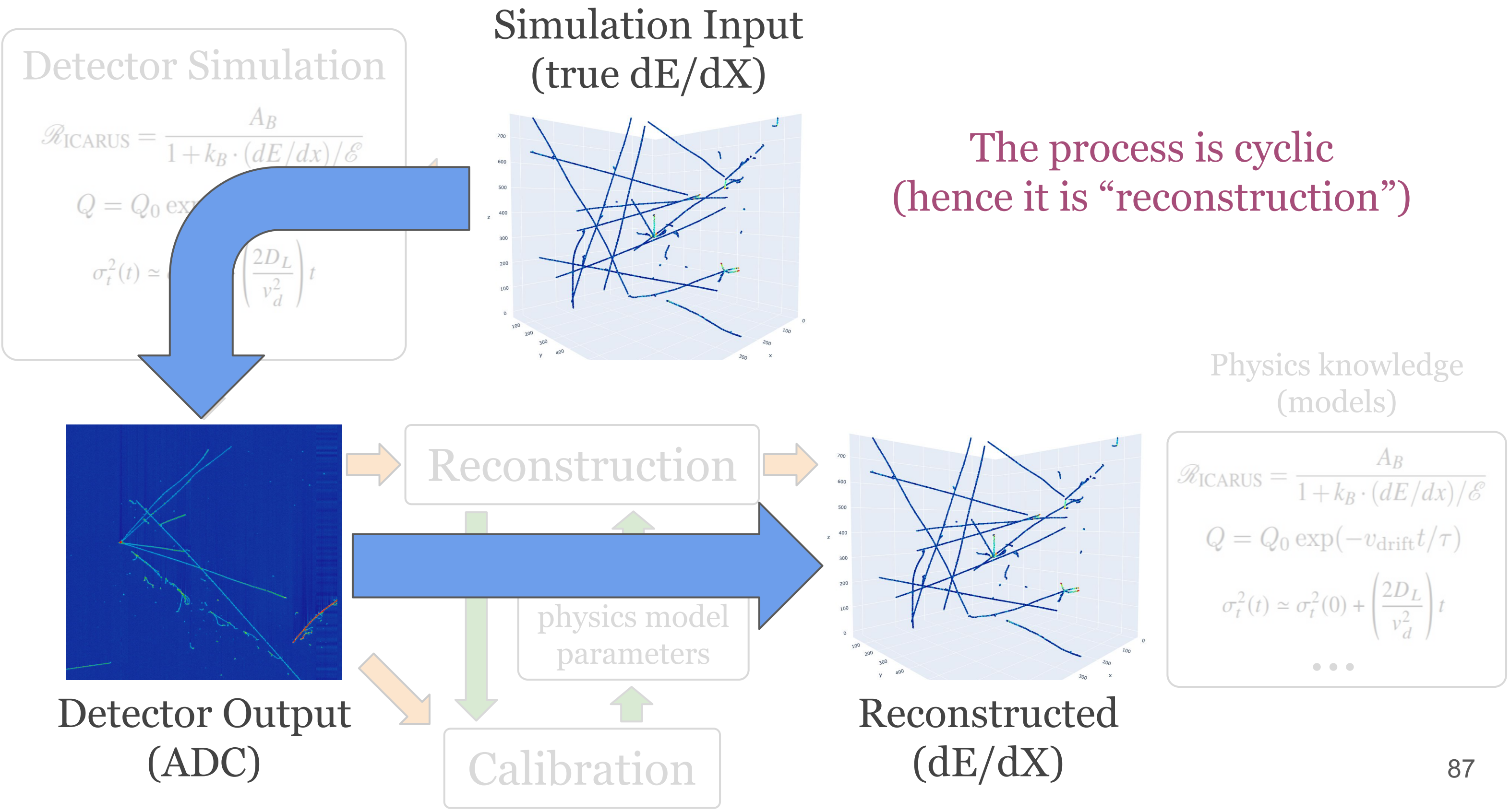
$$\mathscr{R}_{\text{ICARUS}} = \frac{A_B}{1 + k_B \cdot (dE/dx)/\mathscr{E}}$$

$$Q = Q_0 \exp(-v_{\text{drift}} t/\tau)$$

$$\sigma_t^2(t) \simeq \sigma_t^2(0) + \left(\frac{2D_L}{v_d^2}\right) t$$

...

Simulation v.s. Reconstruction/Calibration



Simulation v.s. Reconstruction/Calibration

Detector Simulation

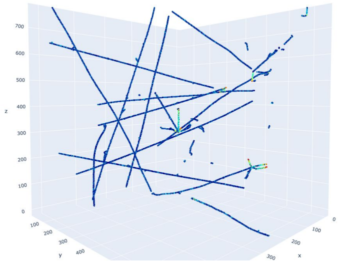
$$\mathscr{R}_{\mathrm{ICARUS}} = \frac{A_B}{1 + k_B \cdot (dE/dx)/\mathscr{E}}$$

$$Q = Q_0 \exp(-v_{\mathrm{drift}} t/\tau)$$

$$\sigma_t^2(t) \simeq \sigma_t^2(0) + \left(\frac{2D_L}{v_d^2}\right) t$$

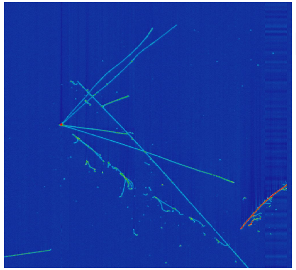
...

Simulation Input
(true dE/dX)



The same (nuisance) models applied at two separate steps.

Extraction and application of model parameter values are done manually.



Detector Output
(ADC)

Reconstruction

Detector physics model parameters

Calibration

Reconstructed
(dE/dX)

Physics knowledge
(models)

$$\mathscr{R}_{\mathrm{ICARUS}} = \frac{A_B}{1 + k_B \cdot (dE/dx)/\mathscr{E}}$$

$$Q = Q_0 \exp(-v_{\mathrm{drift}} t/\tau)$$

$$\sigma_t^2(t) \simeq \sigma_t^2(0) + \left(\frac{2D_L}{v_d^2}\right) t$$

...

Simulation v.s. Reconstruction/Calibration

Detector Simulation

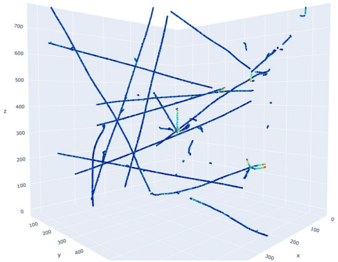
$$\mathscr{R}_{\text{ICARUS}} = \frac{A_B}{1 + k_B \cdot (dE/dx)/\mathscr{E}}$$

$$Q = Q_0 \exp(-v_{\text{drift}} t/\tau)$$

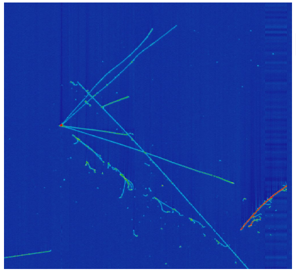
$$\sigma_t^2(t) \simeq \sigma_t^2(0) + \left(\frac{2D_L}{v_d^2}\right) t$$

...

Simulation Input (true dE/dX)



Proposal: automatize the process of detector physics modeling and inference of the input (reconstruction) through an **innovative simulator**



Detector Output (ADC)

Reconstruction

Detector physics model parameters

Calibration

Reconstructed (dE/dX)

Physics knowledge (models)

$$\mathscr{R}_{\text{ICARUS}} = \frac{A_B}{1 + k_B \cdot (dE/dx)/\mathscr{E}}$$

$$Q = Q_0 \exp(-v_{\text{drift}} t/\tau)$$

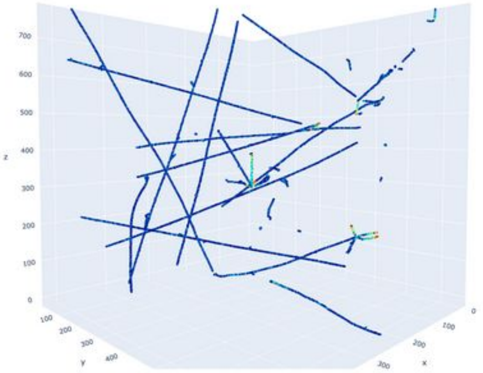
$$\sigma_t^2(t) \simeq \sigma_t^2(0) + \left(\frac{2D_L}{v_d^2}\right) t$$

...

Solving the inverse
… or a direct solver G

Note: G can be trained using only the latter loss as well. Then it's **unsupervised** (purely data-driven)

G $(X|Y, \theta_G)$
Inverse Image Solver



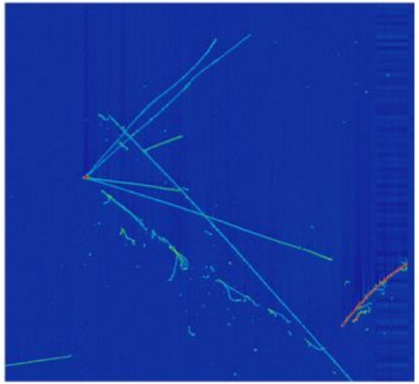
$\mathbf{X} \in \mathcal{D}_I$

Input domain of LArTPC simulator (inaccessible)

$$\boxed{\mathcal{L}_{\text{inv}} = |G(\mathbf{Y}) - \mathbf{X}|^2}$$

and / or

$$\boxed{\mathcal{L}_{\text{cc}} = |F(G(\mathbf{Y})) - \mathbf{Y}|^2}$$



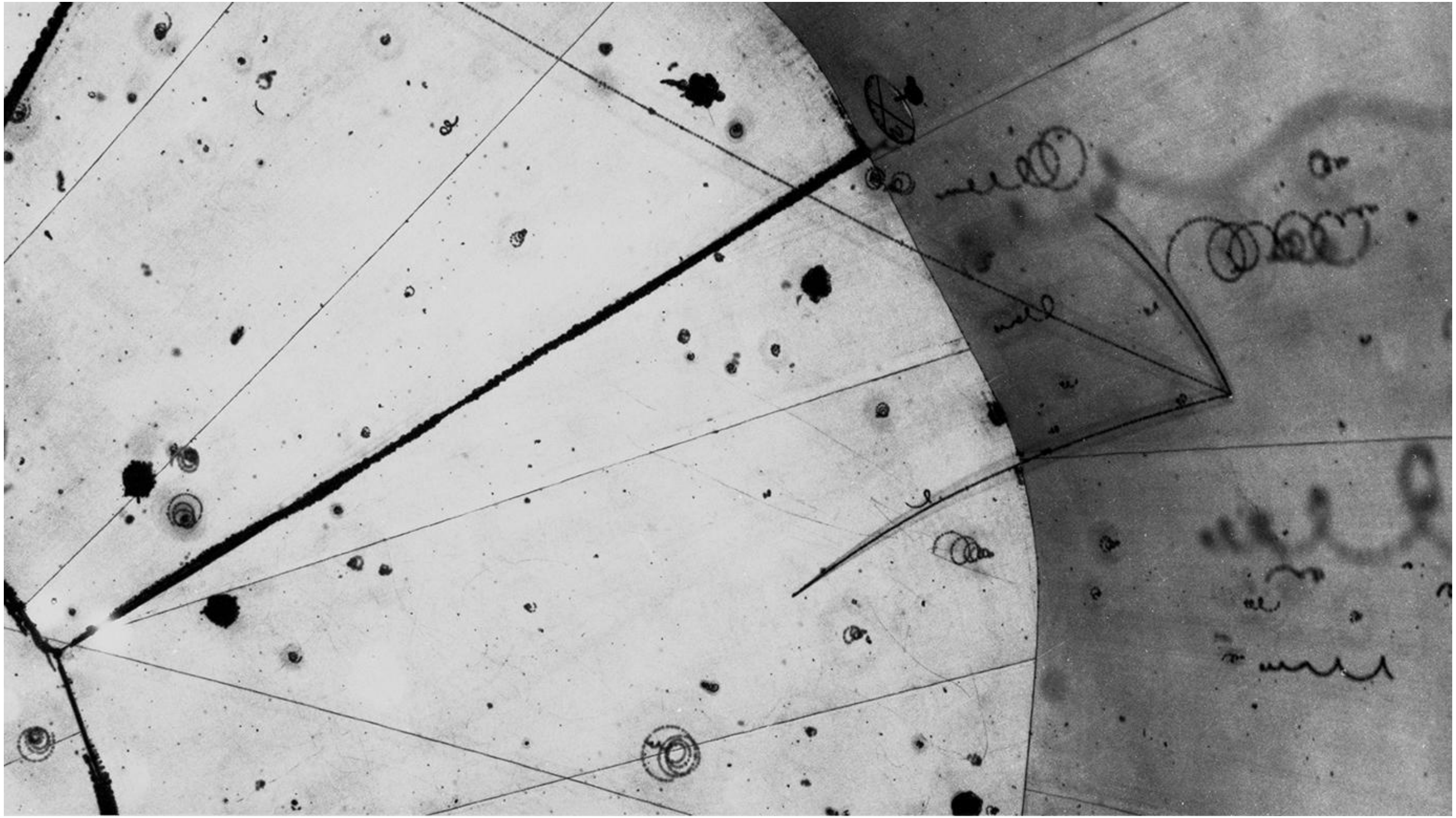
$\mathbf{Y} \in \mathcal{D}_O$

Output domain of LArTPC simulator (e.g. real data)

F $(Y|X, \theta_F)$
Differentiable LArTPC Simulator

ML for Analyzing Big Image Data in Neutrino Experiments

How can we find a neutrino?



How can we find a ~~neutrino~~ cat?

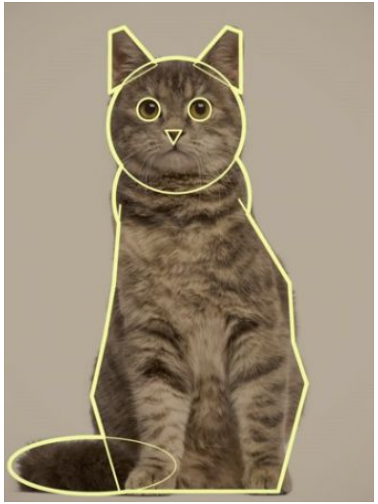
How to write an algorithm to identify a cat?

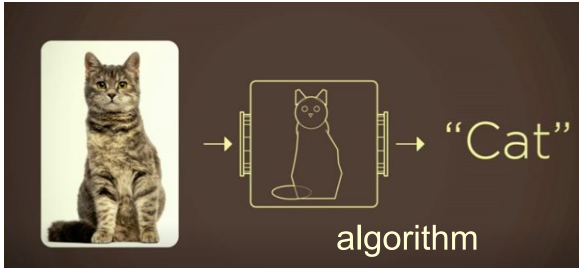
... very hard task ...

16	08	67	15	83	09
37	52	77	23	22	74
35	42	48	72	85	27
68	36	43	54	21	33
79	60	10	25	54	71
18	55	38	73	50	47

Development Workflow for non-ML reconstruction

1. Write an algorithm based on physics principles





A cat = collection of certain shapes
(or, a neutrino)

Images courtesy of Fei Fei Li's TED talk

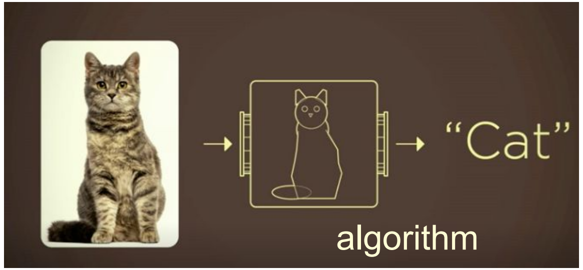
Development Workflow for non-ML reconstruction

1. Write an algorithm based on physics principles
2. Run on simulation and data samples
3. Observe failure cases, implement fixes/heuristics

Partial cat
(escaping the detector)

Images courtesy of Fei Fei Li's TED talk

Stretching cat (Nuclear Physics)



A cat = collection of certain shapes
(or, a neutrino)

How can we find a ~~neutrino~~ cat?

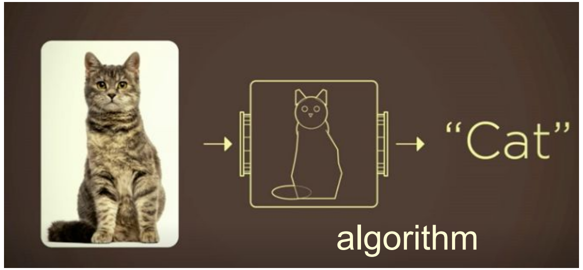
Development Workflow for non-ML reconstruction

1. Write an algorithm based on physics principles
2. Run on simulation and data samples
3. Observe failure cases, implement fixes/heuristics
4. Iterate over 2 & 3 till a satisfactory level is achieved
5. Chain multiple algorithms as one algorithm, repeat 2, 3, and 4.

Partial cat
(escaping the detector)

Images courtesy of Fei Fei Li's TED talk

Stretching cat (Nuclear Physics)



A cat = collection of certain shapes
(or, a neutrino)

Development Workflow for non-ML reconstruction

1. Write an algorithm based on physics principles
2. Run on simulation and data samples
3. Observe failure cases, implement fixes/heuristics
4. Iterate over 2 & 3 till a satisfactory level is achieved
5. Chain multiple algorithms as one algorithm, repeat 2, 3, and 4

"Machine learning (ML)"

- Design a solution pattern (instead of an explicit algorithm)
- Automation of optimization (steps 2-4)
- Multi-task optimization possible (step 5)

Image Classifications: a lot of applications

Especially great for: **“a rare event in a quiet detector”**

- **Quiet** = can assume “almost always neutrino”
  - e.g.) no cosmic-ray background
- **Rare** = “only 1 neutrino”

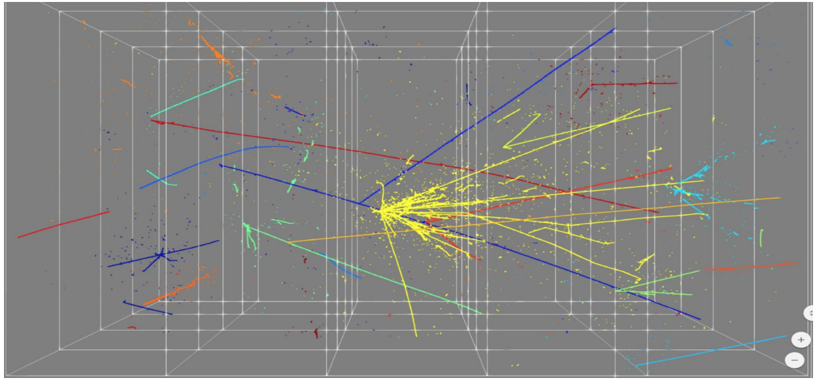
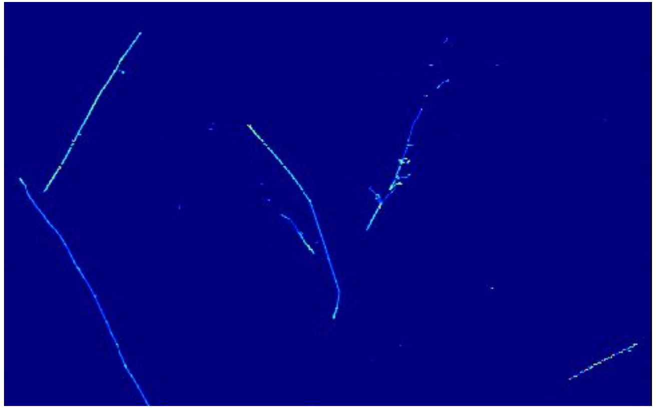
Image Classifications: a lot of applications

Especially great for: **"a rare event in a quiet detector"**

- **Quiet** = can assume "almost always neutrino"
  - e.g.) no cosmic-ray background
- **Rare** = "only 1 neutrino"
  - the same "image classification architecture" can be applied for…
    - neutrino flavor (topology) classification
    - energy regression (image to one FP32 value)
    - vertex regression (image to three FP32 value)
    - etc. …

Machine Learning & Computer Vision in Neutrino Physics
Image Classifications: a lot of applications

Especially great for: **“a rare event in a quiet detector”**

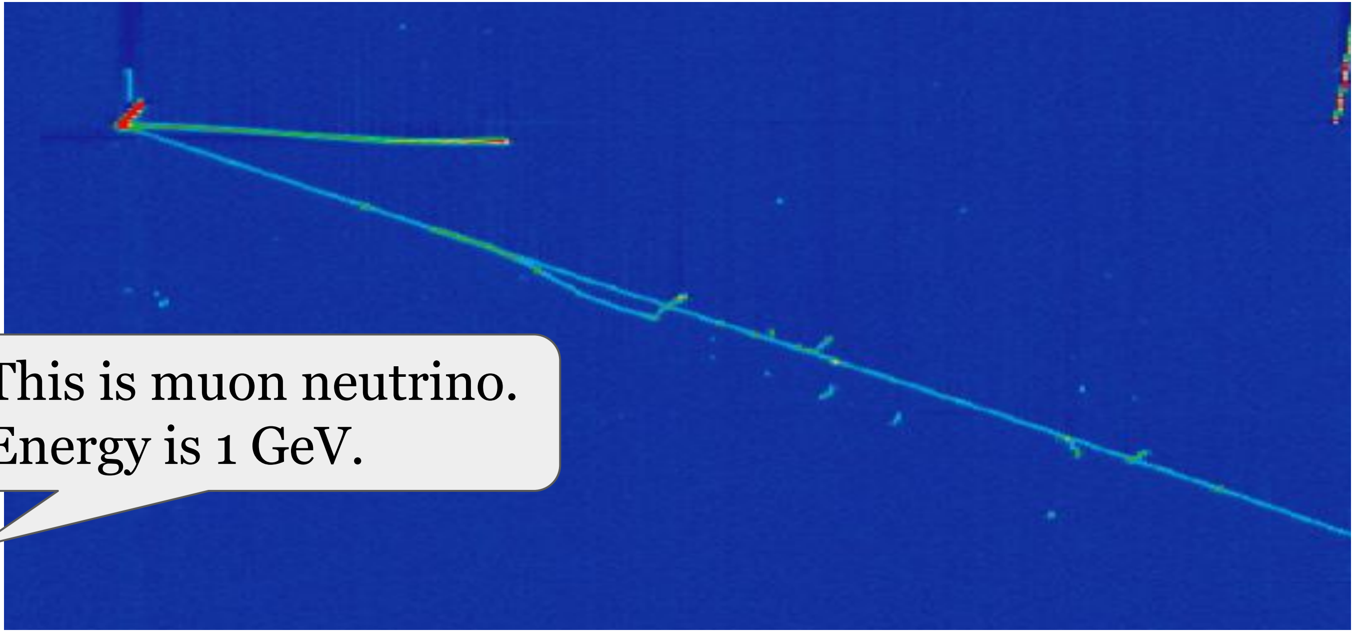



… **but most of LArTPC detectors are not** …

- MicroBooNE, ICARUS, SBND, ProtoDUNE … physics in next 5 years
  - Busy: typically dozens of cosmic rays in each event
- DUNE-ND
  - Not rare (busy): a dozen of neutrino interaction pile-up in each event

Why Data Reconstruction

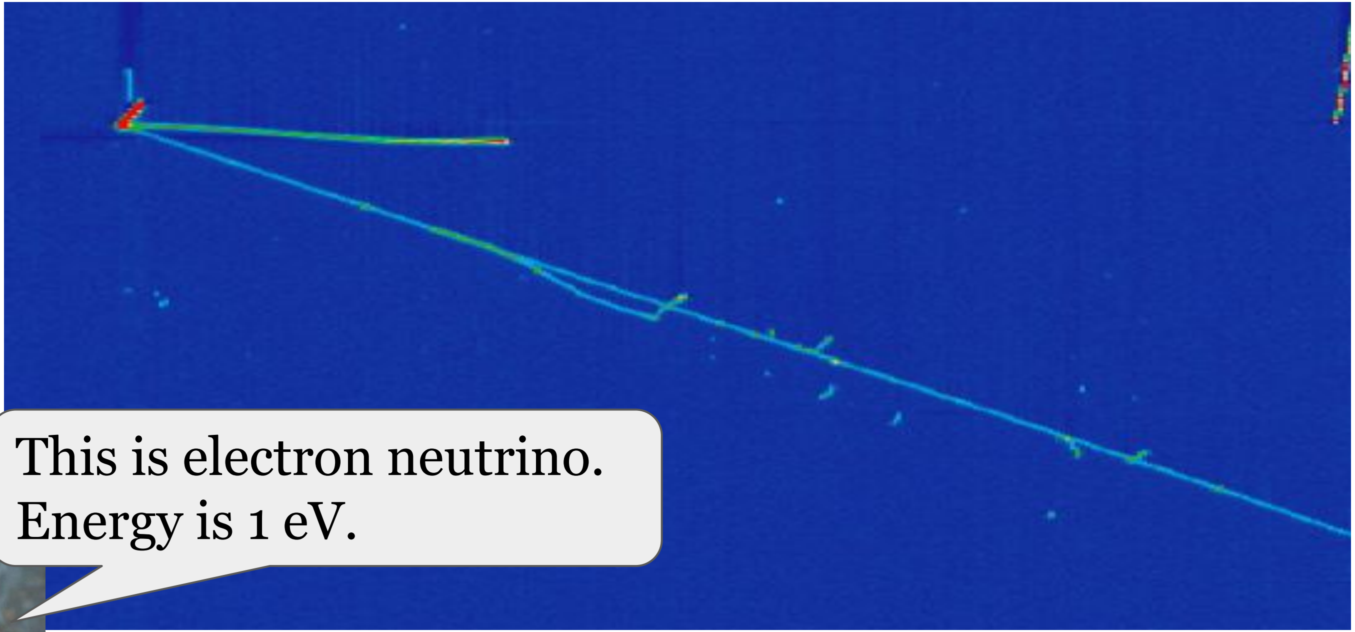
Image classification/regression: straight to "**flavour & energy**"



This is muon neutrino.
Energy is 1 GeV.

Why Data Reconstruction

... but also challenging: a huge single-step of information reduction



This is electron neutrino.
Energy is 1 eV.

... **would be nice to know why you thought so** ...

Reconstruction Details

Stage 1-a: Pixel Feature Extraction + Scalablility

"Applying CNN" is simple, but **is it scalable for us?**

CNN applies **dense matrix operations**

In photographs, **all pixels are meaningful**

grey pixels = dolphins,
blue pixels = water, etc...

Stage 1-a: Pixel Feature Extraction + Scalablility

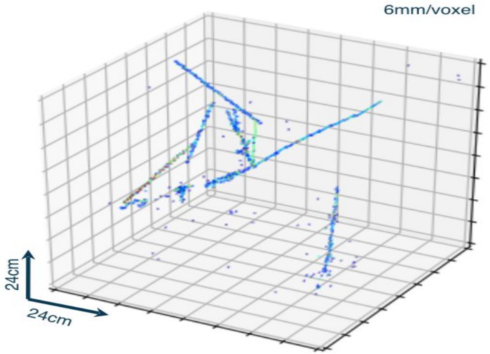
"Applying CNN" is simple, but **is it scalable for us?**

LArTPC data is generally sparse, but locally dense

CNN applies **dense matrix operations**

In photographs, **all pixels are meaningful**

grey pixels = dolphins,
blue pixels = water, etc...



Empty pixels = no energy

<1% of pixels are non-zero in LArTPC data

Zero pixels are meaningless!

Figures/Texts: courtesy of Laura Domine @ Stanford

Figure credit: Laura Domine @ Stanford

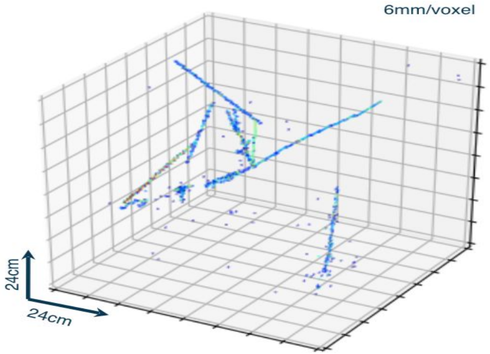
ML-based Neutrino Data Reconstruction Chain
Stage 1-a: Pixel Feature Extraction + Scalablility

"Applying CNN" is simple, but **is it scalable for us?**

LArTPC data is generally sparse, but locally dense

CNN applies **dense matrix operations**

In photographs, **all pixels are meaningful**



<1% of pixels are non-zero in LArTPC data

Zero pixels are meaningless!

Figures/Texts: courtesy of Laura Domine @ Stanford

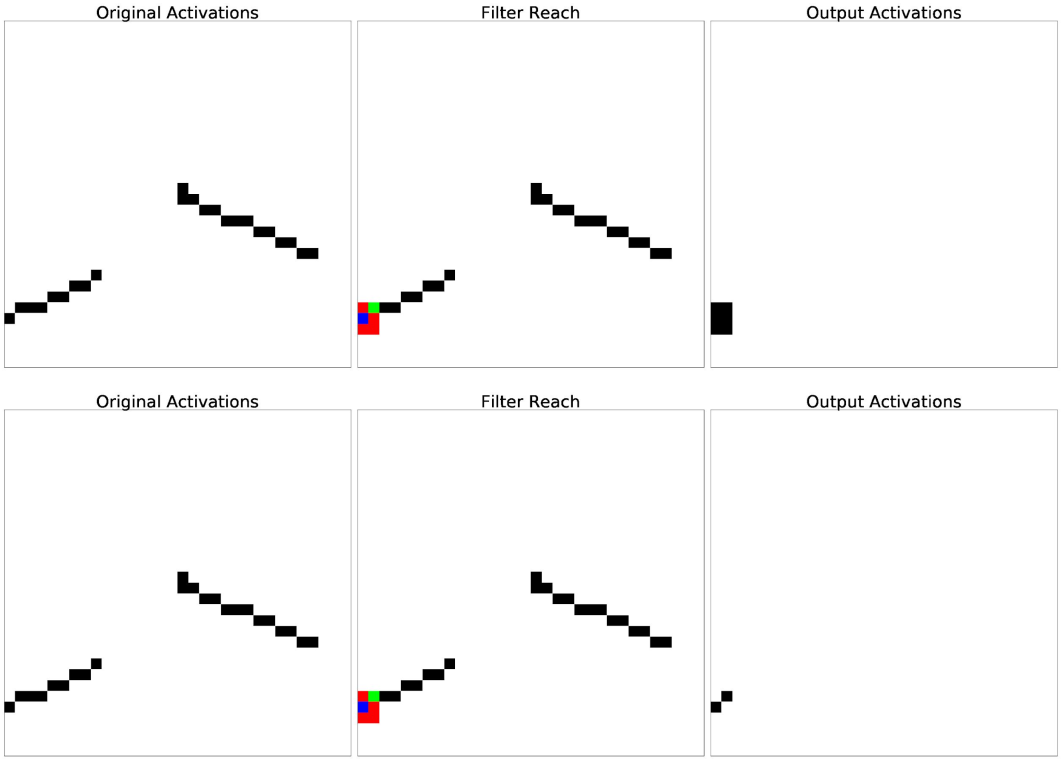
- **Scalability for larger detectors**
  - Computation cost increases linearly with the volume
  - But the number of non-zero pixels does not

Figure credit: Laura Domine @ Stanford

Sparse Submanifold Convolutions

Only acts on an active input pixels + can limit output activations for only the same pixels.

- 1st implementation by FAIR
- 2nd implementation by Stanford VL
  - ... also supported in NVIDIA now

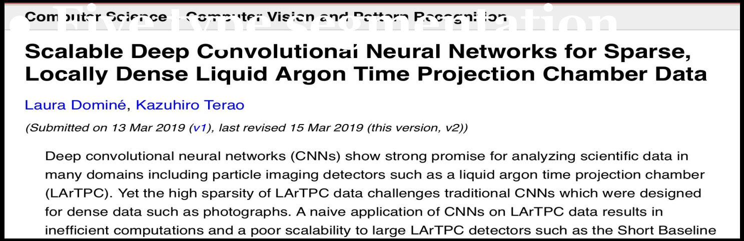


ML-based Neutrino Data Reconstruction Chain
Stage 1-a: Pixel Feature Extraction + Scalablility

SLAC

CNN on sparse tensors (MinkowskiEngine)

- **Public LArTPC simulation**
  - Particle tracking (Geant4) + diffusion, no noise, true energy



Computer Science > Computer Vision and Pattern Recognition

Scalable Deep Convolutional Neural Networks for Sparse, Locally Dense Liquid Argon Time Projection Chamber Data

Laura Dominé, Kazuhiro Terao

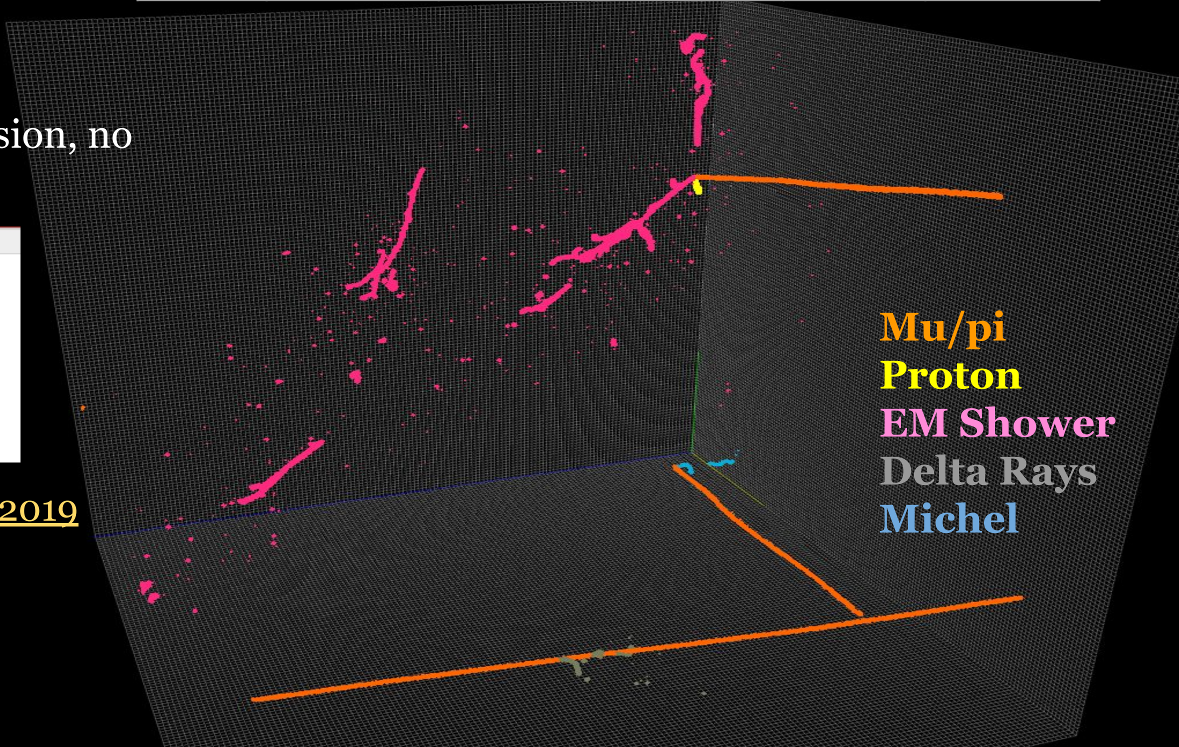
(Submitted on 13 Mar 2019 (v1), last revised 15 Mar 2019 (this version, v2))

Deep convolutional neural networks (CNNs) show strong promise for analyzing scientific data in many domains including particle imaging detectors such as a liquid argon time projection chamber (LArTPC). Yet the high sparsity of LArTPC data challenges traditional CNNs which were designed for dense data such as photographs. A naive application of CNNs on LArTPC data results in inefficient computations and a poor scalability to large LArTPC detectors such as the Short Baseline

PhysRevD.102.012005 presented @ ACAT 2019

- Memory reduction ~ 1/360
- Compute time ~ 1/30
- Handles large future detectors

Type	Proton	Mu/Pi	Shower	Delta	Michel
Acc.	0.99	0.98	0.99	0.97	0.96

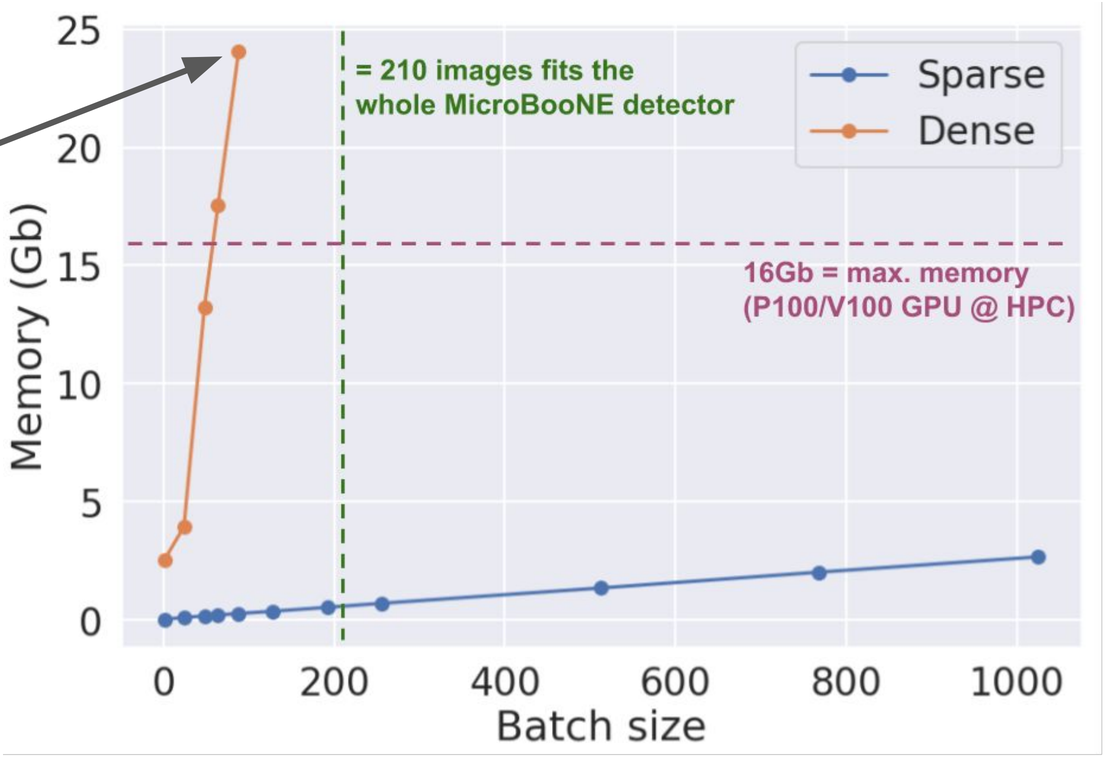


Stage 1-a: Pixel Feature Extraction + Scalablility

SLAC

Sparse U-ResNet fits more data in GPU + good scalability

@batch size 88
sparse uses
93x less memory
than dense and
computation is
3x faster



Can handle easily the whole ICARUS detector which is x6 larger than MicroBooNE.

DUNE-FD is piece of cake (larger volume but less non-zero pixels)

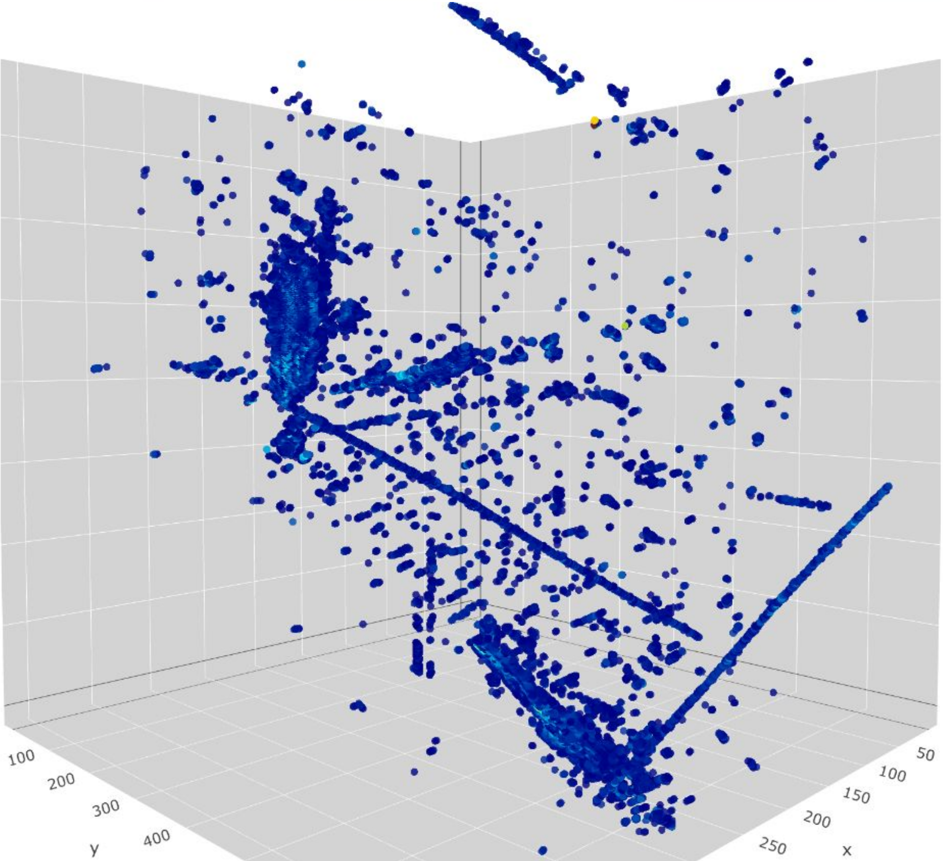
Work credit: Laura Domine (Stanford) and Ran Itay (SLAC)

Backup Slides

2D=>3D

Machine Learning & Computer Vision in Neutrino Physics

Bonus: isochronous ghost point removal



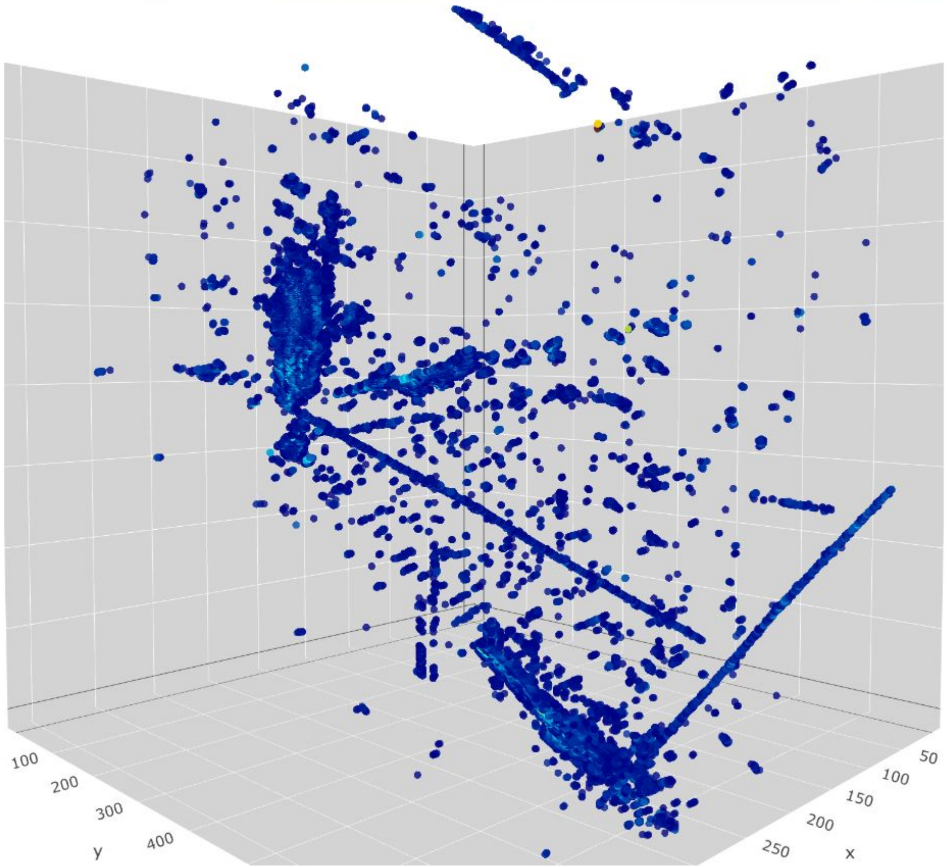
ICARUS Detector
Reconstructed 3D points

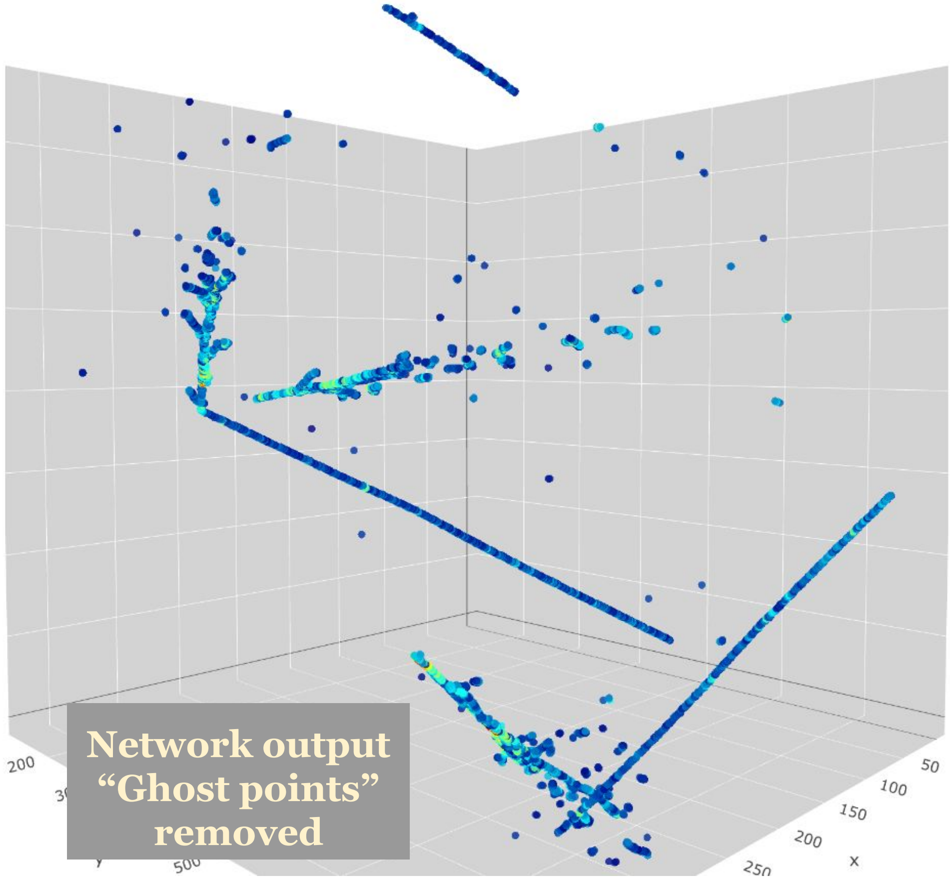
work credit:
Laura Domine
Patrick Tsang

Machine Learning & Computer Vision in Neutrino Physics

Bonus: isochronous ghost point removal

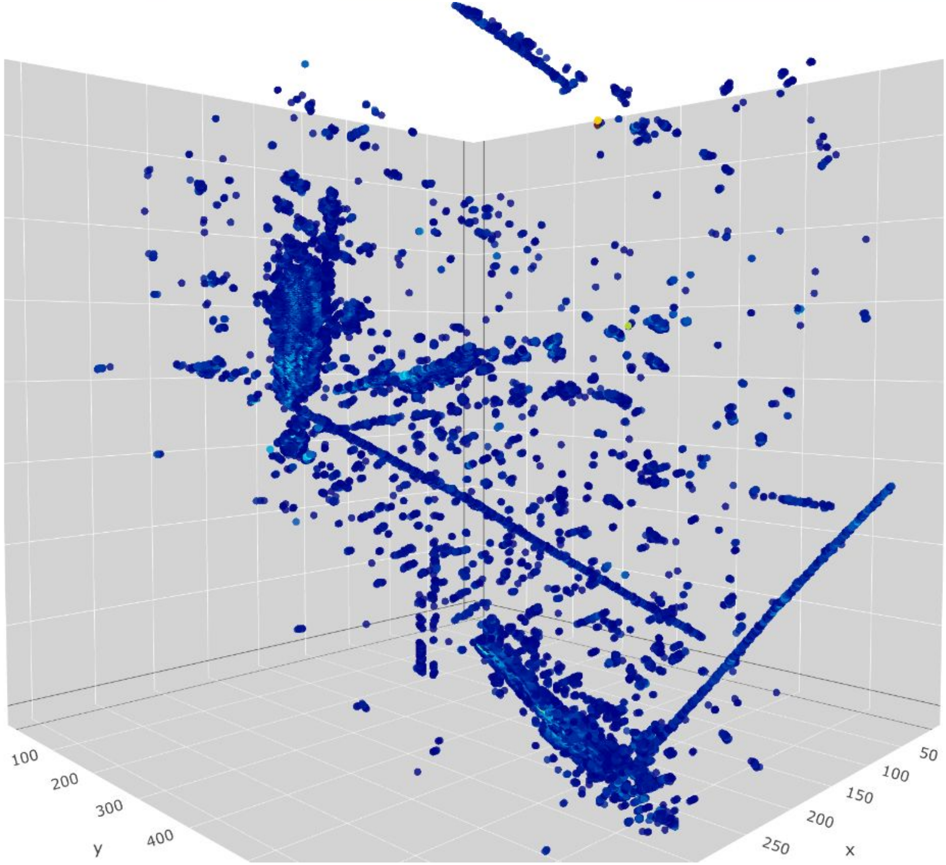
SLAC

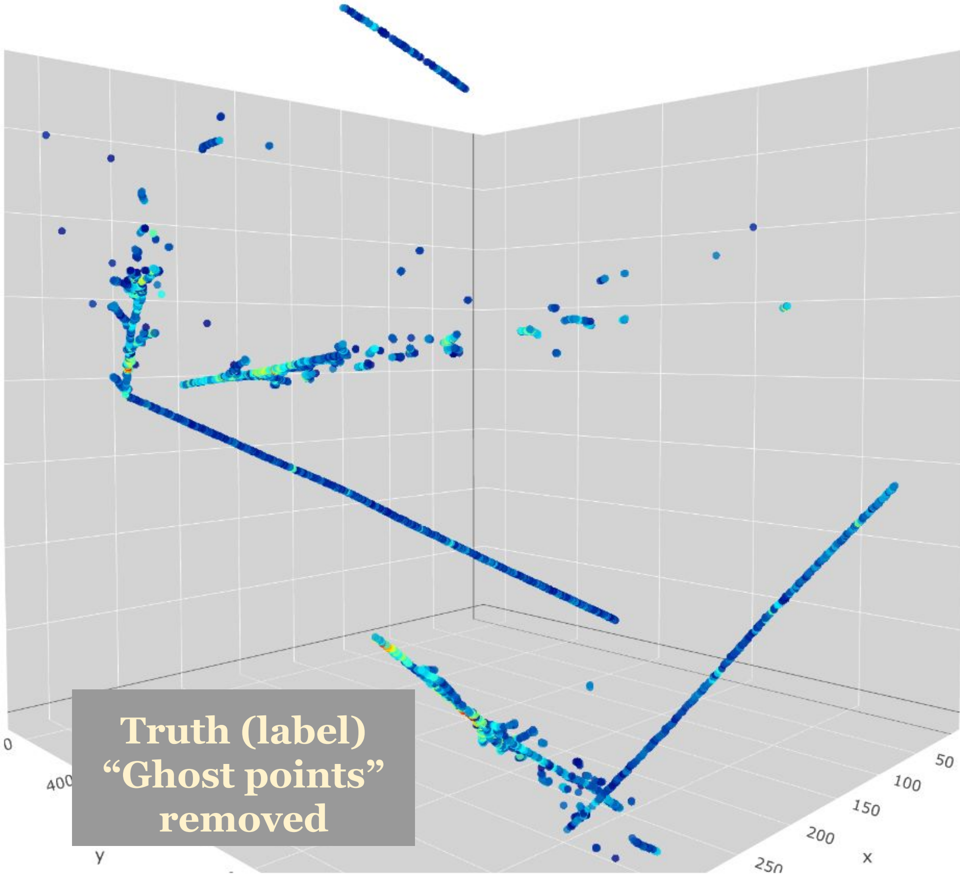




Machine Learning & Computer Vision in Neutrino Physics

Bonus: isochronous ghost point removal



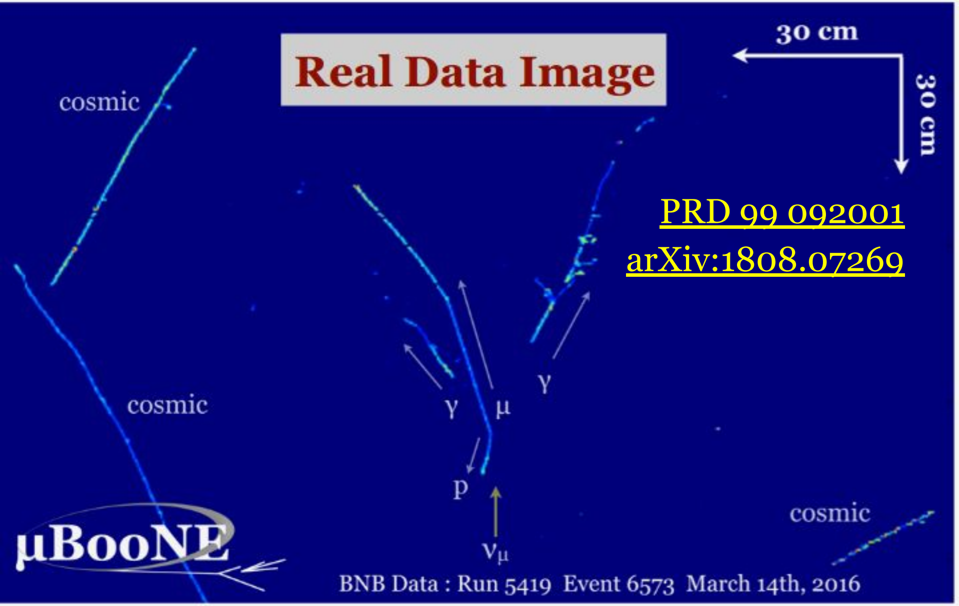


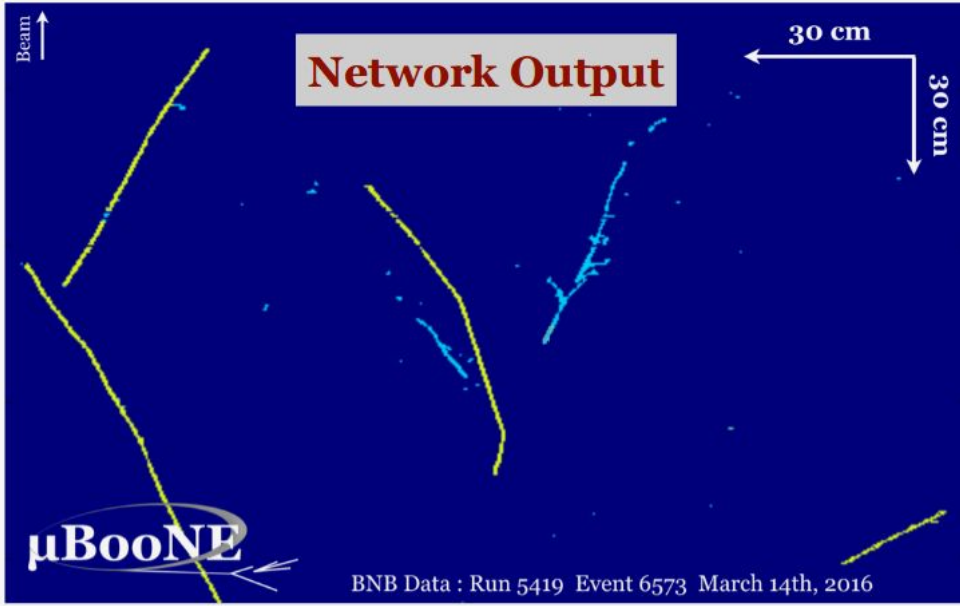
Segmentation Data

Machine Learning & Computer Vision in Neutrino Physics

Semantic Segmentation for Pixel-level Particle ID

Separate electron/positron energy depositions from other types at raw waveform level. Helps the downstream clustering algorithms (**data/sim comp. @ arxiv:1808.07269**)

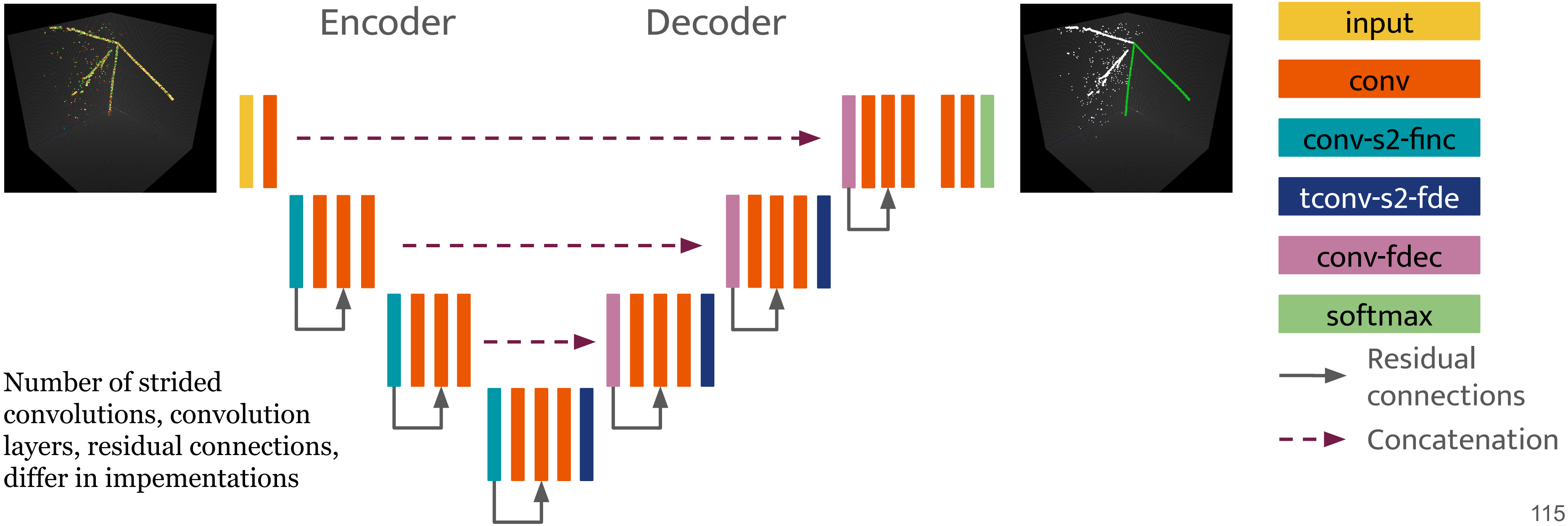




Network Input **Network Output**

Semantic Segmentation for Pixel-level Particle ID

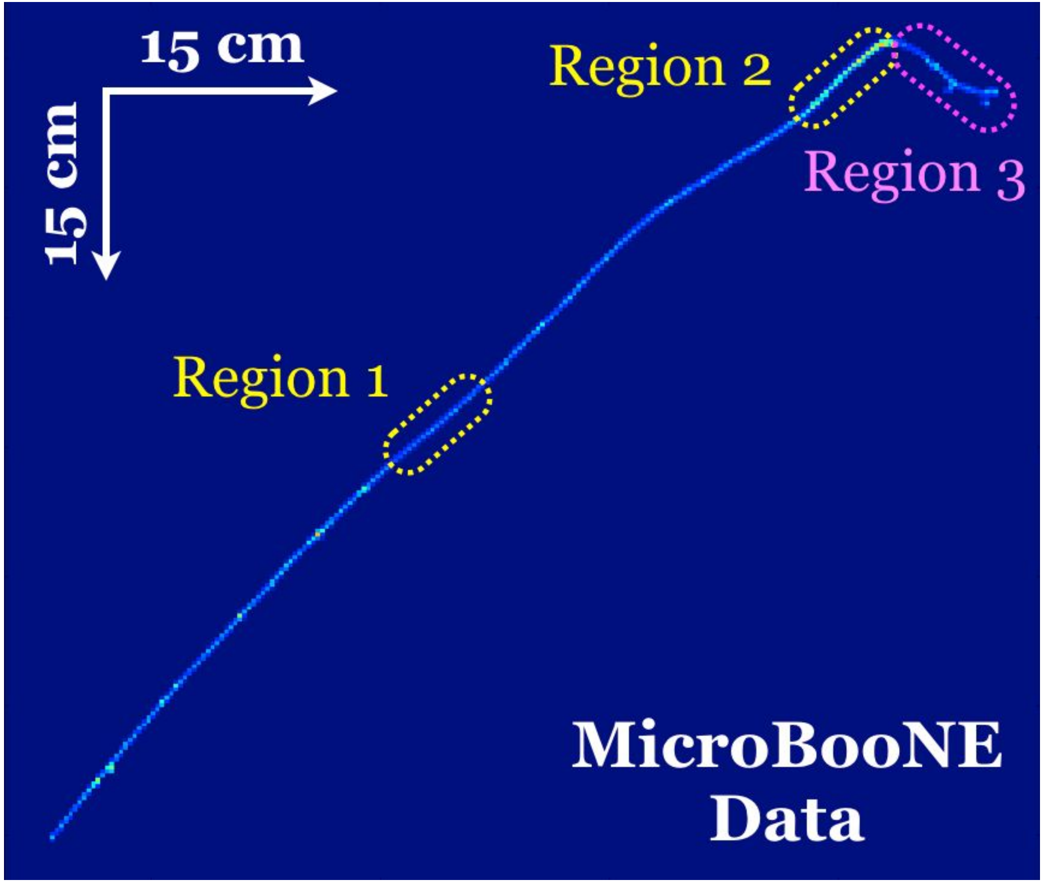
Architecture: **U-Net + Residual Connections**



Number of strided convolutions, convolution layers, residual connections, differ in impementations

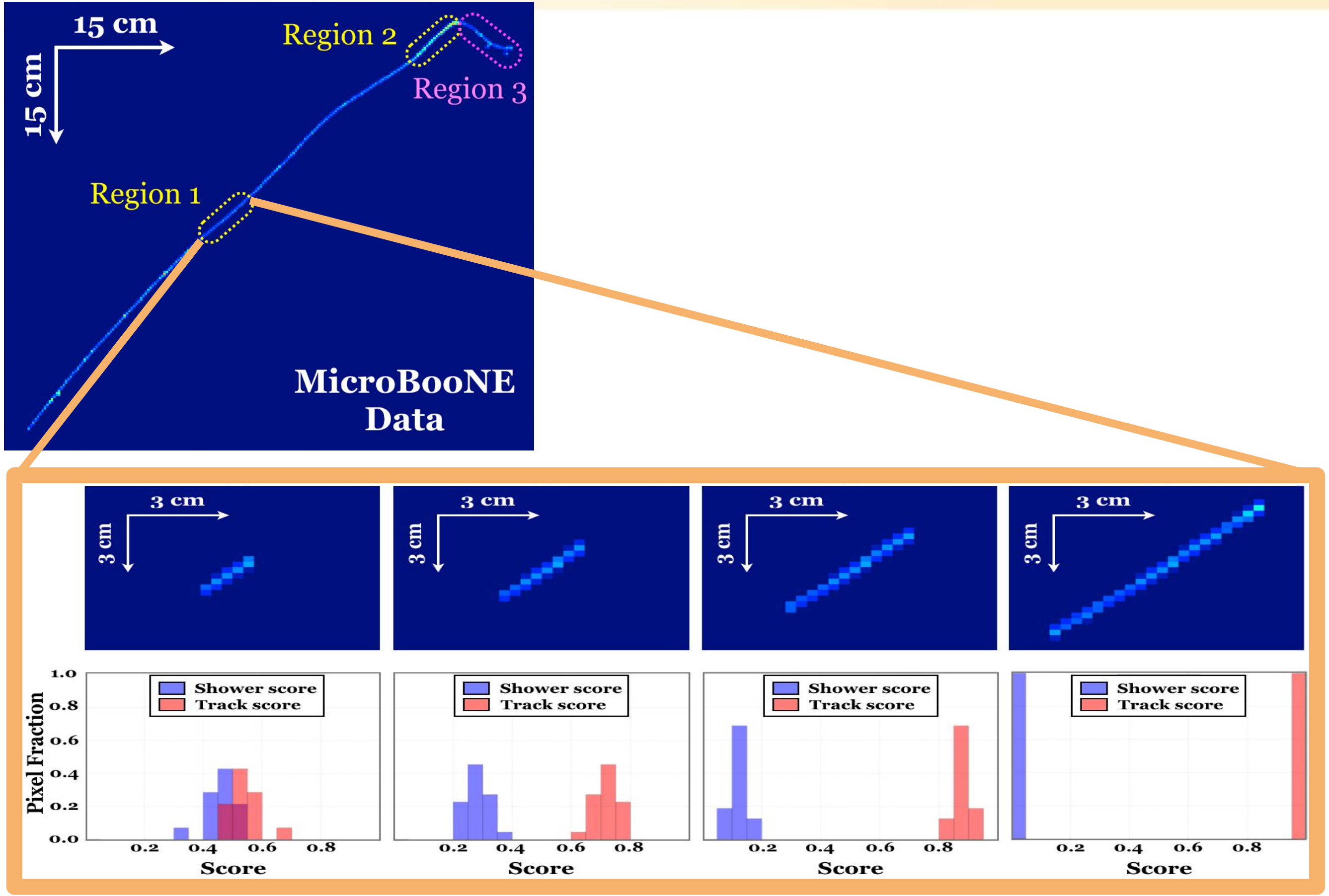
Image credit: Laura Domine @ Stanford

Fun Playing with Semantic Segmentation



Machine Learning & Computer Vision in Neutrino Physics

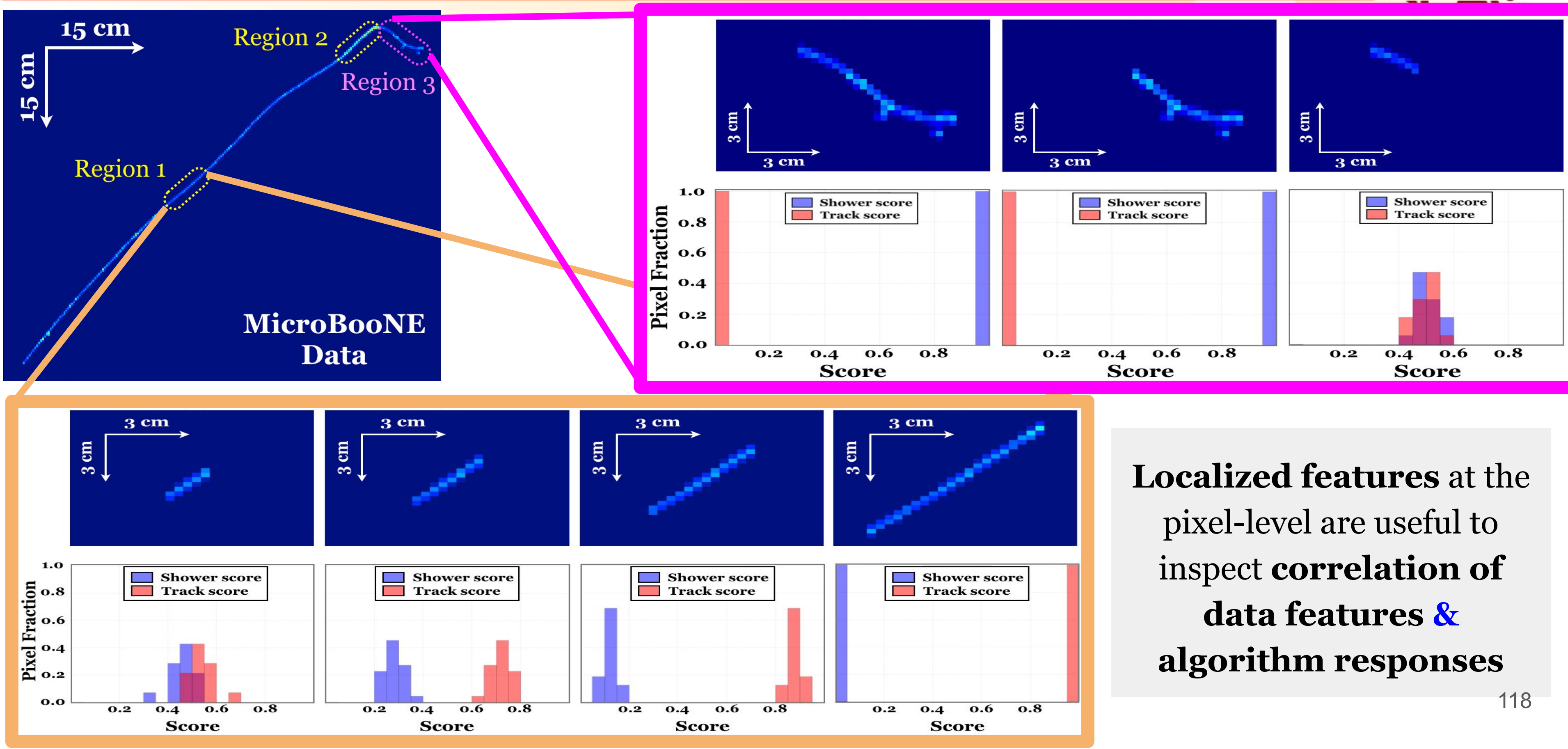
Fun Playing with Semantic Segmentation



Localized features at the pixel-level are useful to inspect **correlation of data features & algorithm responses**

Machine Learning & Computer Vision in Neutrino Physics

Fun Playing with Semantic Segmentation



Localized features at the pixel-level are useful to inspect **correlation of data features & algorithm responses**