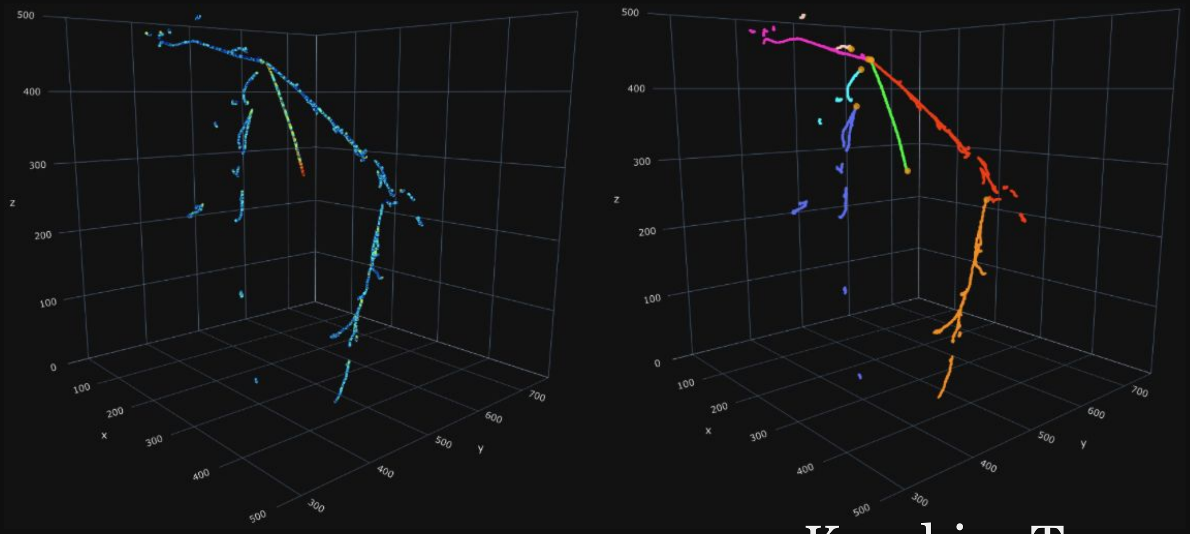
AI/ML for Automation of Scientific Workflows

For my AI/ML research focused on neutrinos, go look at [this youtube]!

Original image credit: xkcd

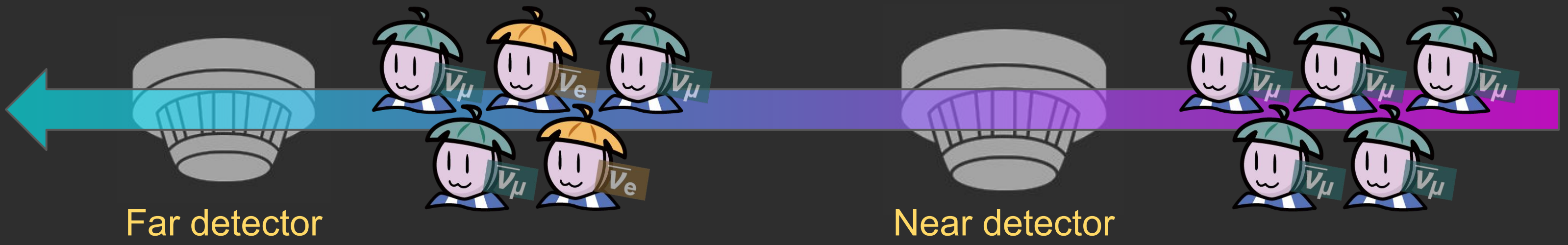


Kazuhiro Terao

ML@IPA Workshop
Mar. 21st 2023 @ ETH Zurich

Challenges @ the Neutrino Frontiers

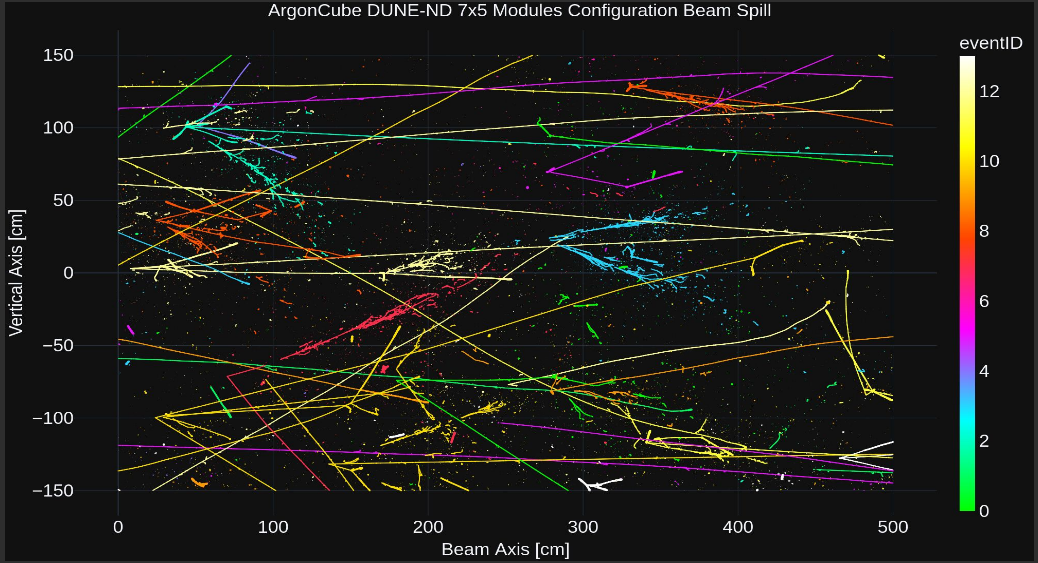
Accelerator-based Neutrino Experiments



accelerator

Analysis steps

- Identify individual neutrinos
- Infer neutrino properties
- Compare observables between two detectors
- Infer the physics

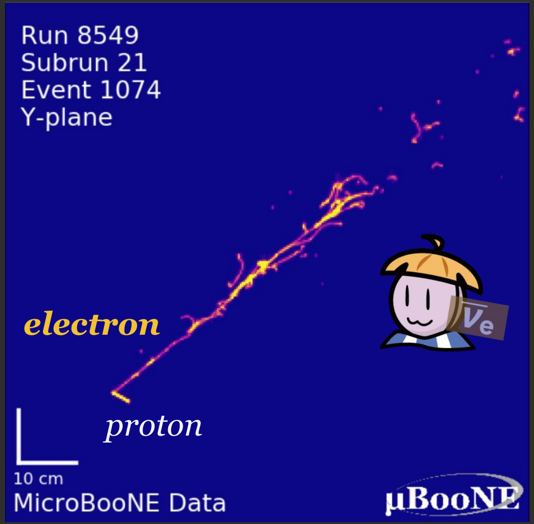


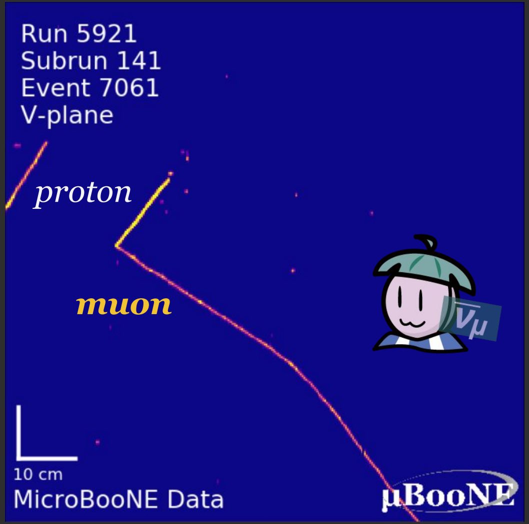
Accelerator-based Neutrino Experiments

accelerator

Analysis steps

- Identify individual neutrinos
- Infer neutrino properties
- Compare observables between two detectors
- Infer the physics

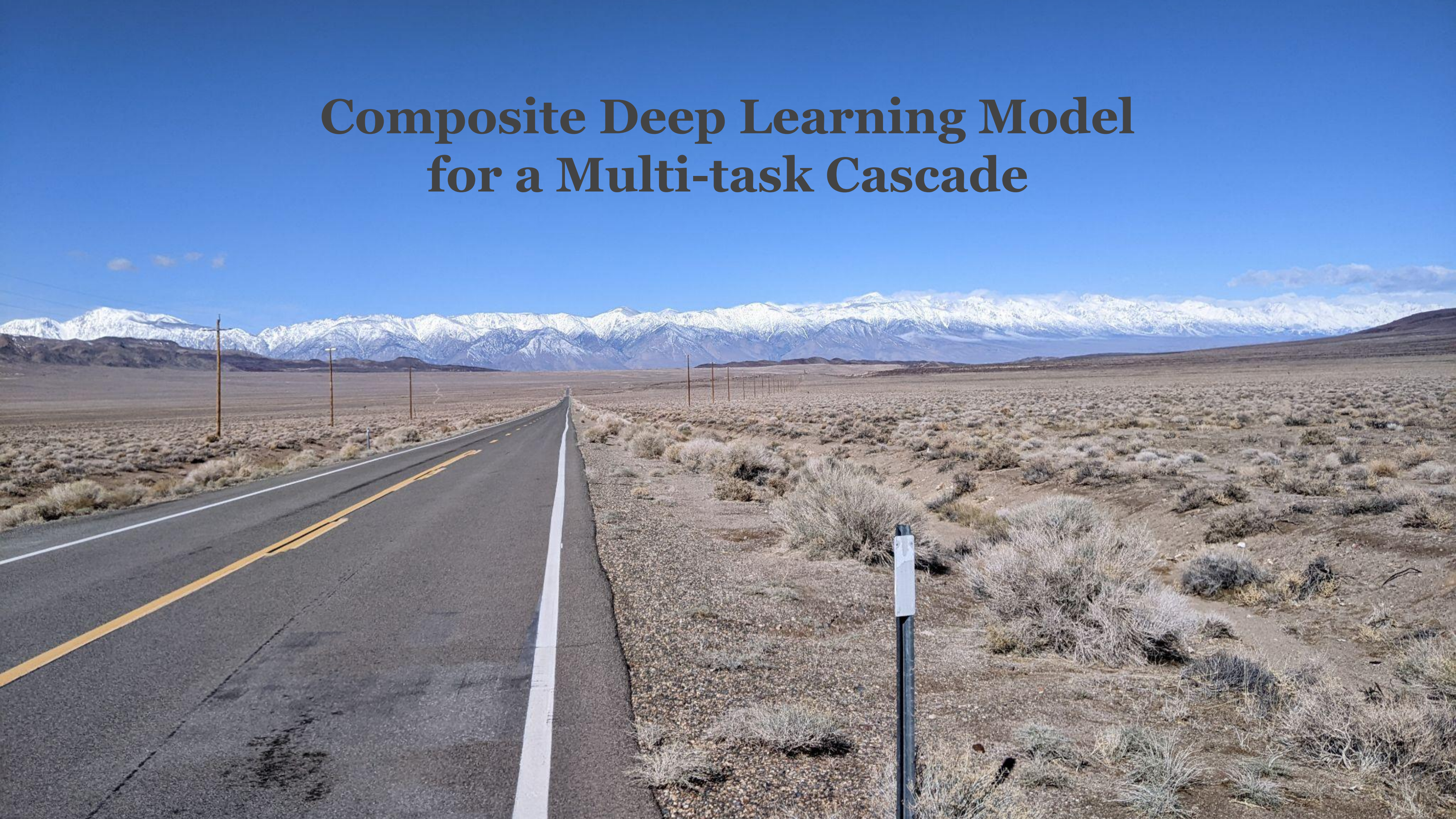




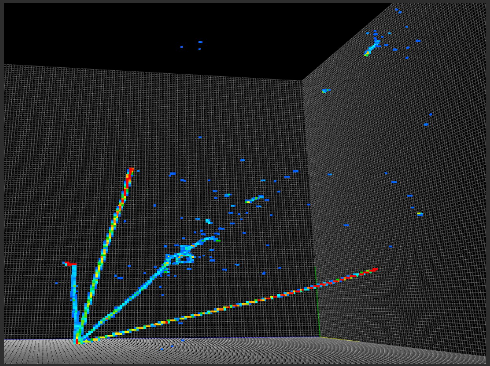
Future Directions of the AI/ML Development

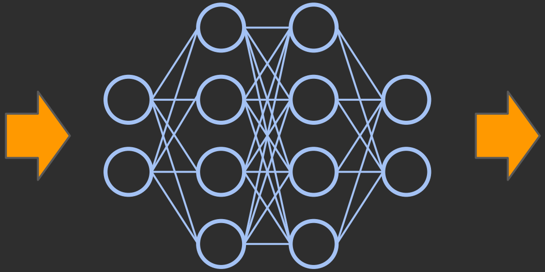
- ~~Challenges in accelerator-based neutrino experiments~~
- A multi-task workflow optimization with a composite model
- Differentiable physics models and inference applications
- Summary

Composite Deep Learning Model for a Multi-task Cascade



AI/ML is Impactful: can we make it better?





0 = "electron neutrino"

1 = "muon neutrino"

Today: going beyond a "simple" end-to-end AI/ML

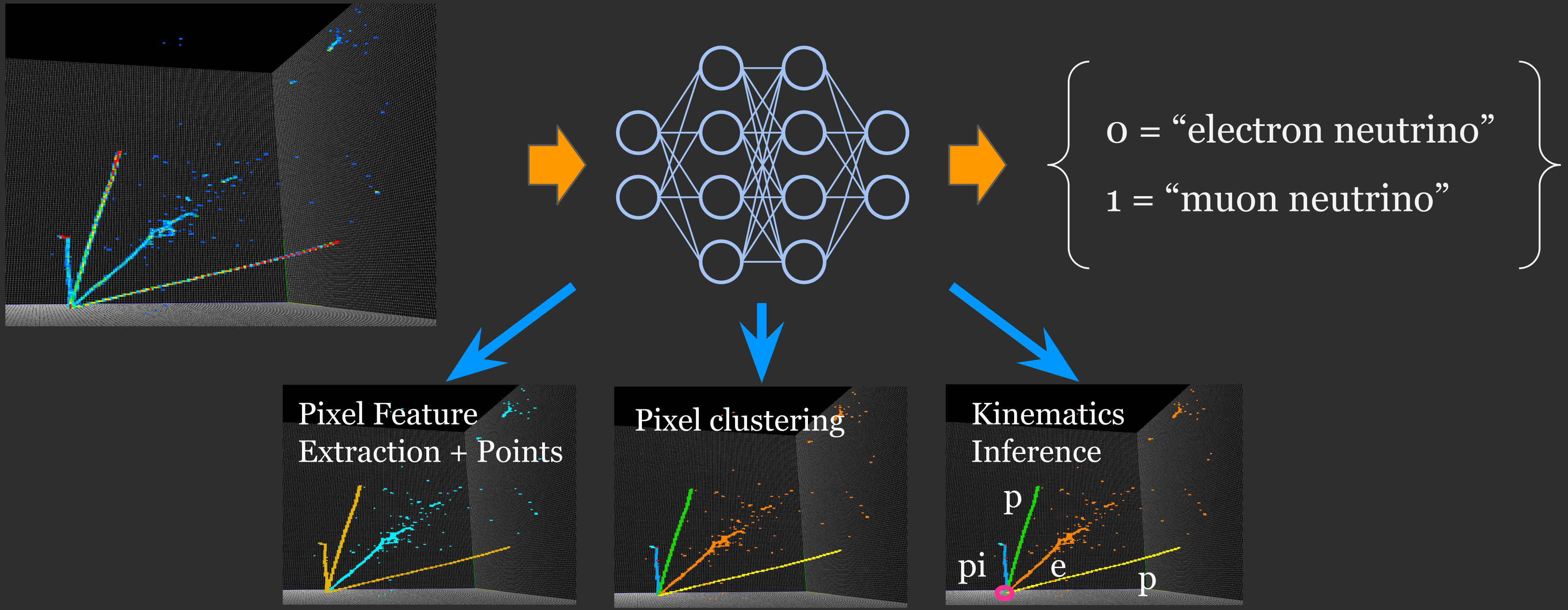
Robustness

Interpretability / Explainability

Reusability

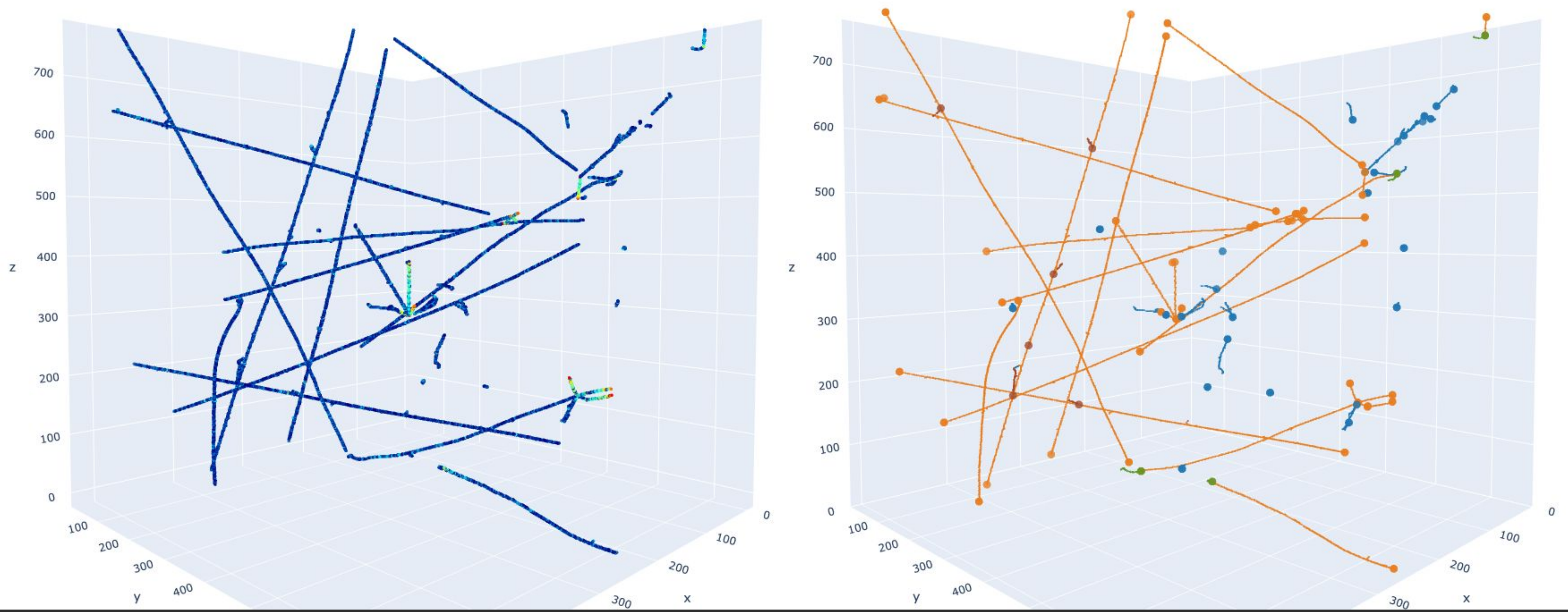
… without losing goodies like powerful optimization methods, automation, etc.

Inductive Bias: Algorithm Structure



Reconstruction: produce intermediate physical observables
with sensible hierarchical correlations

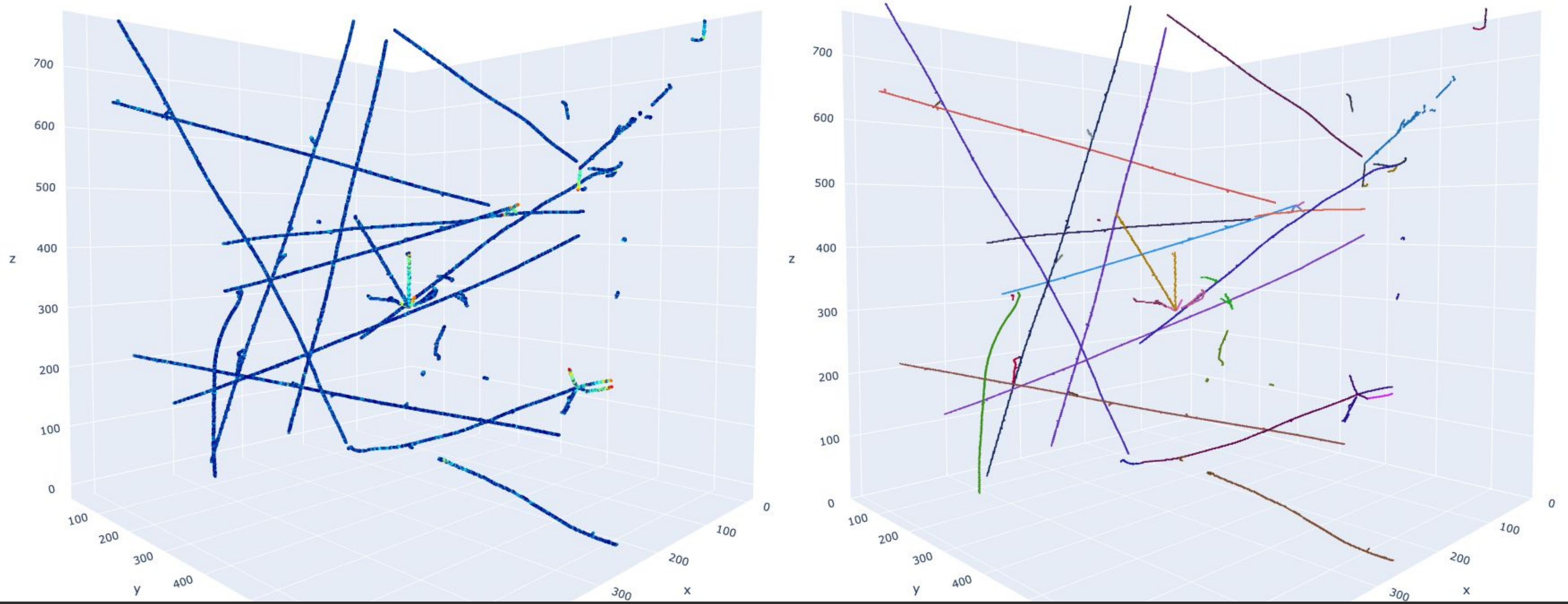
Full Data Reconstruction via Deep Learning



Step 1: Identifying pixel-level key features
(Sparse-CNN for globally-sparse, locally-dense images)

Full Data Reconstruction via Deep Learning

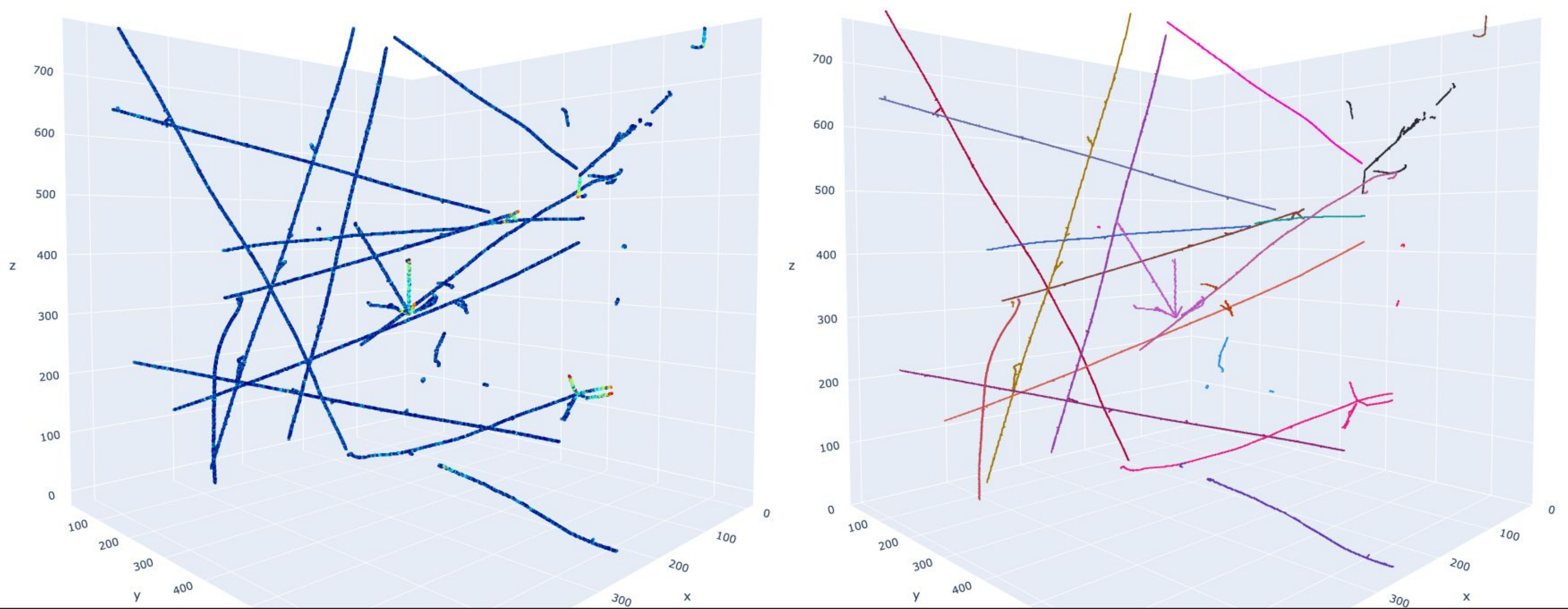
Full chain (NeurIPS WS)
Public dataset
1, 2, 3, 4



Step 2: Identifying individual particles
(CNN for dense-pixel clustering + GNN for scattered cluster aggregation)

Full Data Reconstruction via Deep Learning

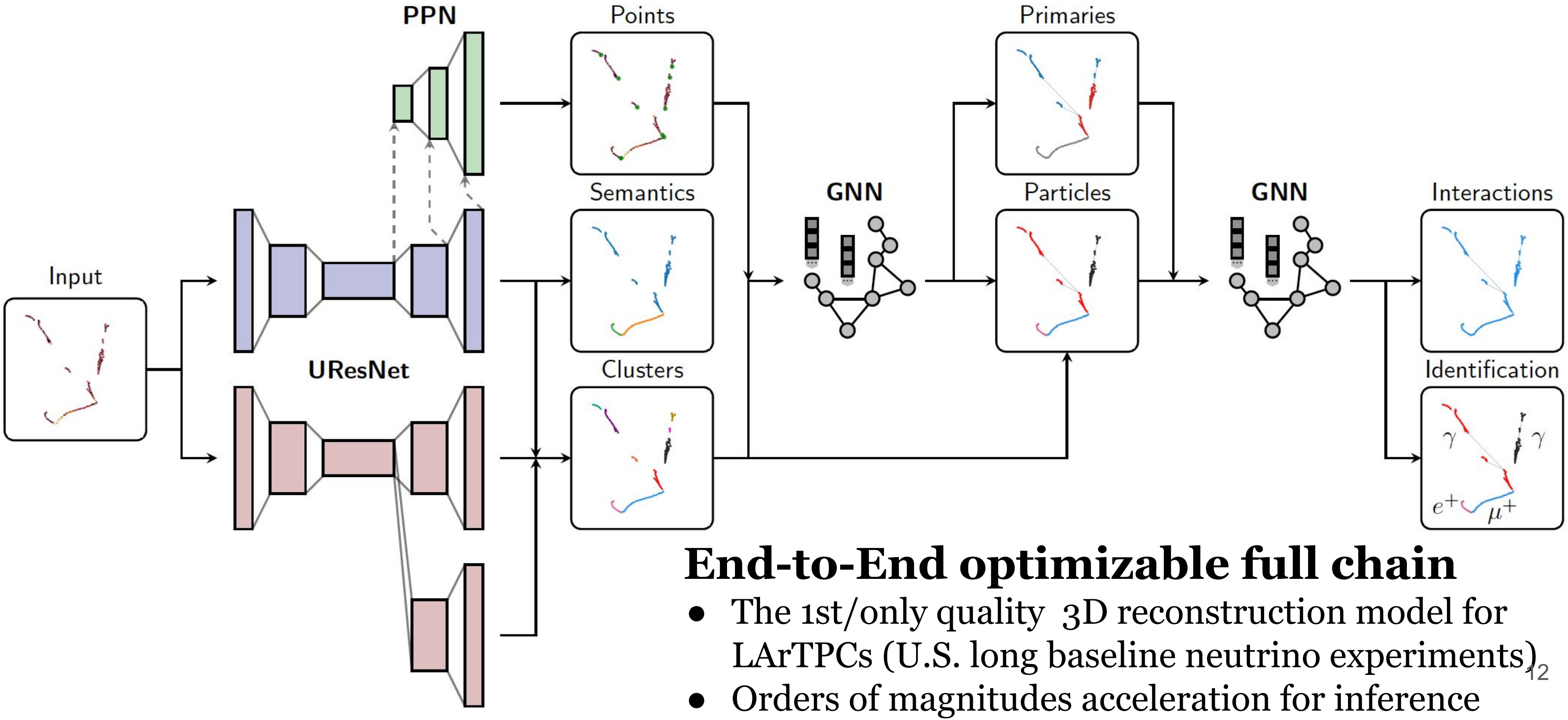
[Full chain (NeurIPS WS)](#)
[Public dataset](#)
[1](#), [2](#), [3](#), [4](#)



Step 3: Identifying particle-to-particle correlations
(GNN with directed graph and node/edge aggregation layers)

Full Data Reconstruction via Deep Learning

Full chain (NeurIPS WS)
Public dataset
1, 2, 3, 4

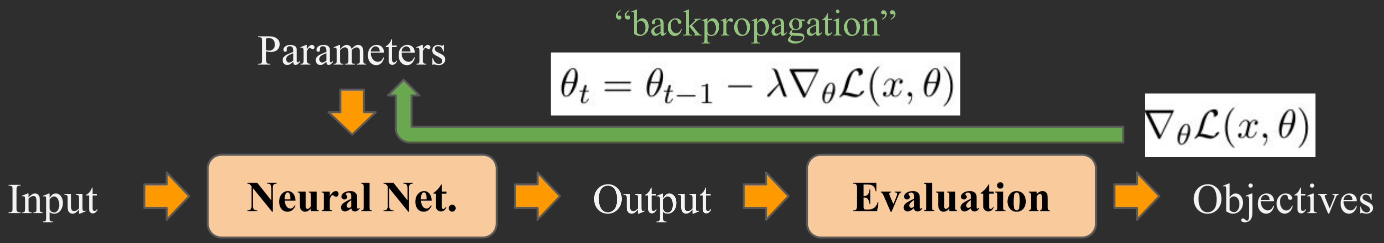


End-to-End optimizable full chain

- The 1st/only quality 3D reconstruction model for LArTPCs (U.S. long baseline neutrino experiments)
- Orders of magnitudes acceleration for inference

SciML: Applying AI/ML Hiking Skills For Physics Inference

Gradient-based Optimization



Differentiable Physics Models

"backpropagation"

$$\theta_t = \theta_{t-1} - \lambda \nabla_\theta \mathcal{L}(x, \theta)$$

Parameters

$\nabla_\theta \mathcal{L}(x, \theta)$

Input → **Physics Sim.** → Output → **Evaluation** → Objectives

Cla∂ TensorFlow

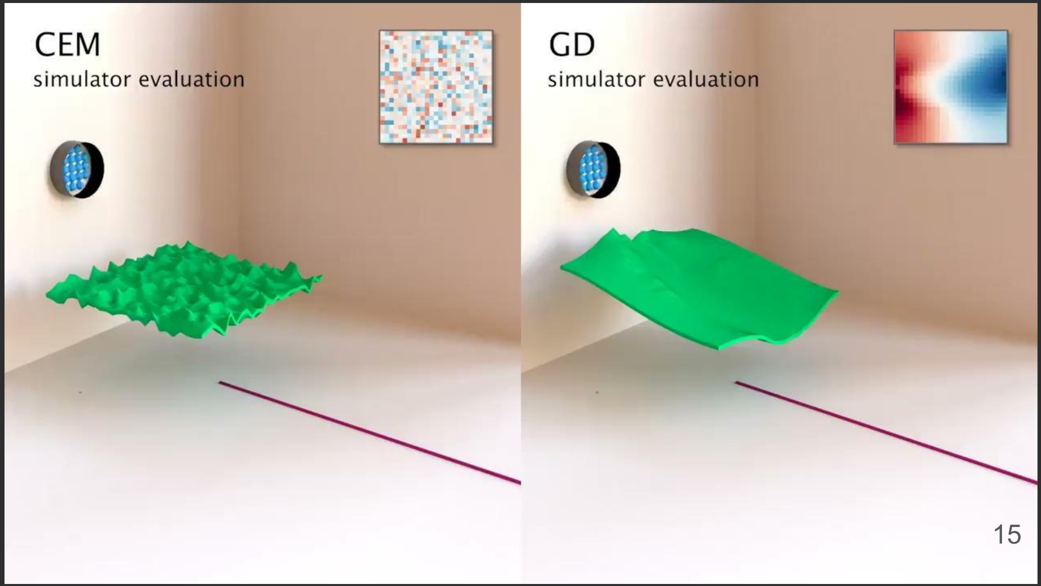
JAX PyTorch Zygote

x θ

Program

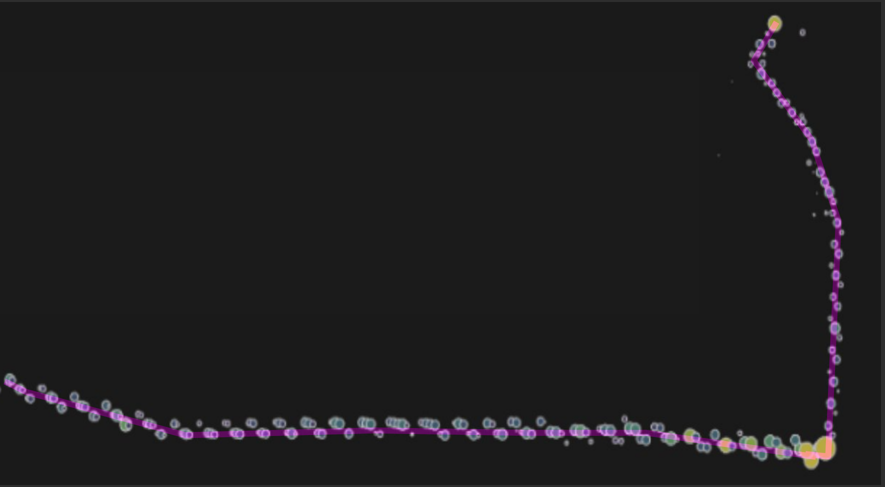
$f_\theta(x)$ $\nabla_\theta f_\theta(x)$

Physical Design using Differentiable Learned Simulators (DeepMind [2202.00728](https://arxiv.org/abs/2202.00728))



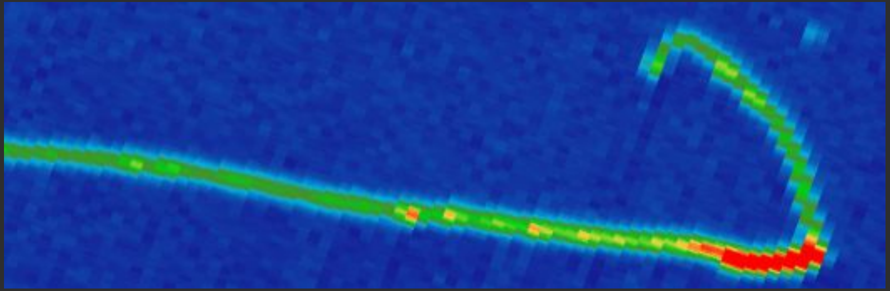
Differentiable Physics Models

Modeling Detector Physics

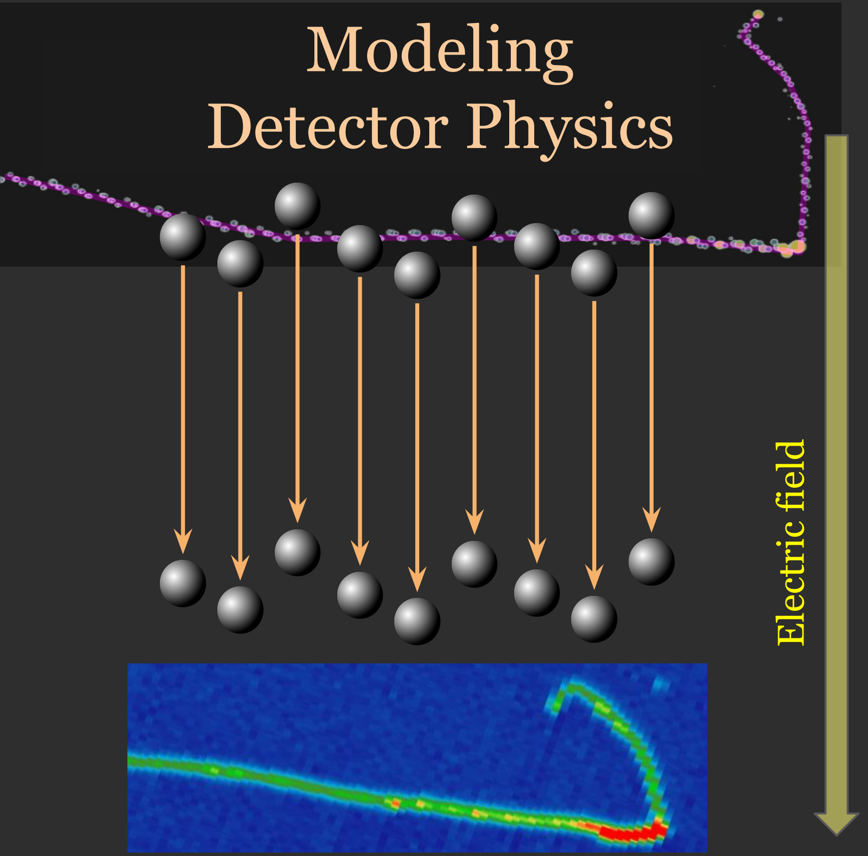


Example: Liquid Argon TPC

Objective: given a calibration dataset (i.e. images of particle trajectories with approximated dE/dX values), "fit" the detector physics parameters



Differentiable Physics Models



Example: Liquid Argon TPC

- Charged particle ionize electrons
- Electrons drifts under E-field
- Signal diffuse and attenuated

Detector Simulation

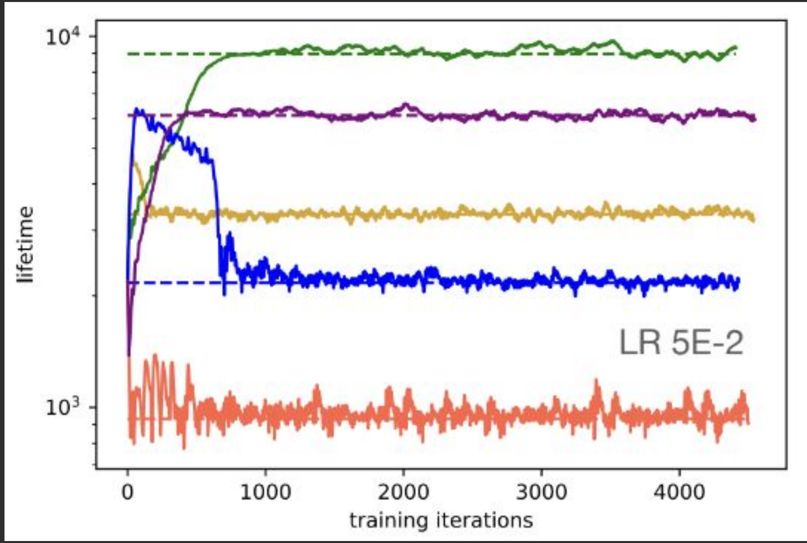
$$\mathscr{R}_{\text{ICARUS}} = \frac{A_B}{1 + k_B \cdot (dE/dx)/\mathscr{E}}$$

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

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

...

Differentiable Physics Models



Optimizing the “lifetime” physics parameter directly from calibration dataset

Example: Liquid Argon TPC

- Charged particle ionize electrons
- Electrons drifts under E-field
- Signal diffuse and attenuated

Detector Simulation

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

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

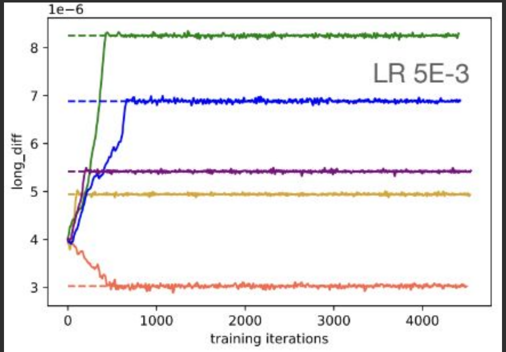
$$\sigma_t^2(t) \simeq \sigma_t^2(0) + \left(\frac{2D_L}{v_d^2}\right) t$$

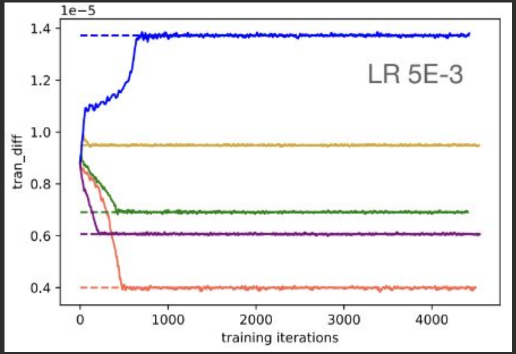
...

Differentiable Physics Models

Diffusion during the drift

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

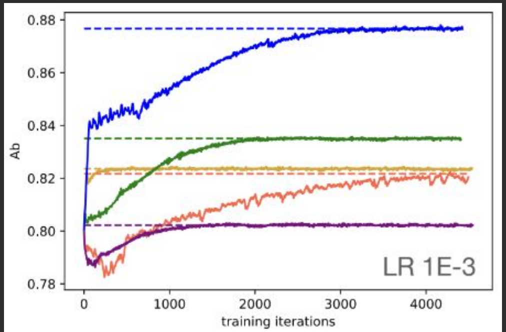


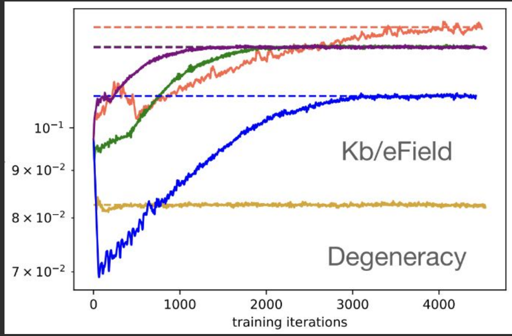


Work credit due (from left):
ML/Math: Youssef N., Sean G., Daniel R.
neutrino: Yifan C., Roberto S.

Ionization (signal) yield

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





Lots of applications

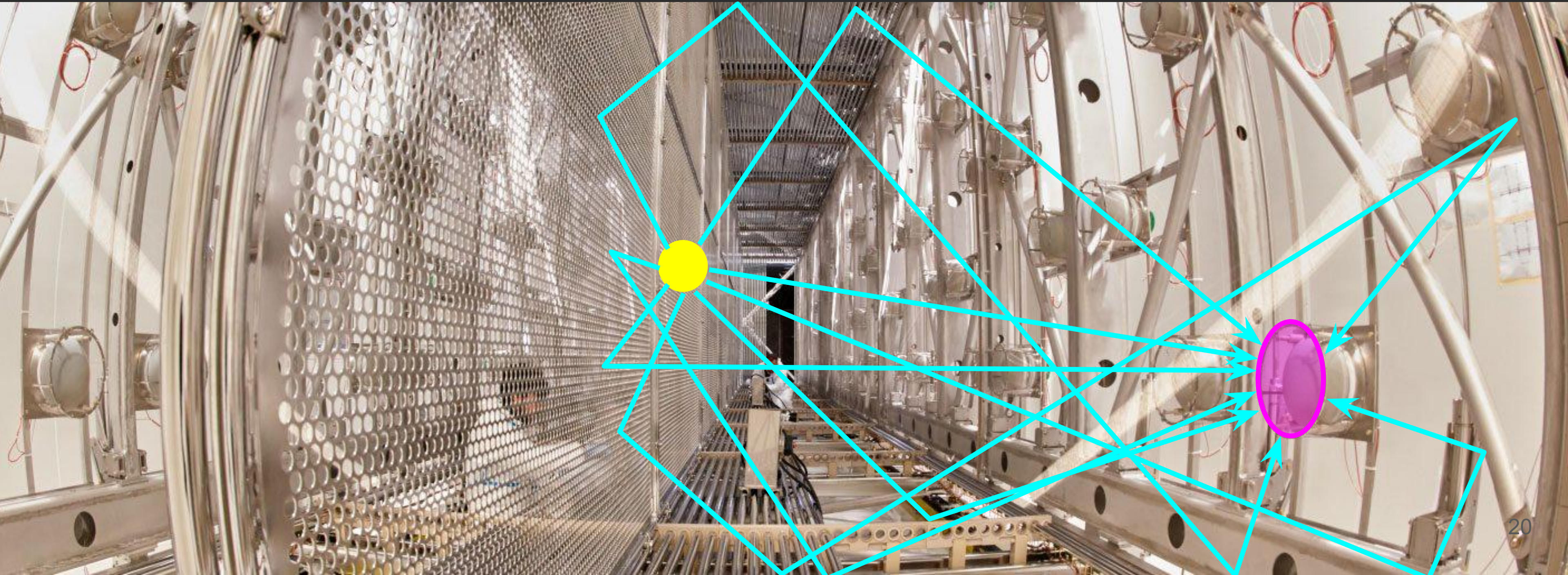
- Simultaneous multi-parameter fit
- Inter-parameter dependency study
- Automation of calibration workflow
- Inverse imaging (i.e. reconstruction)

Differentiable Surrogate as a Simulator

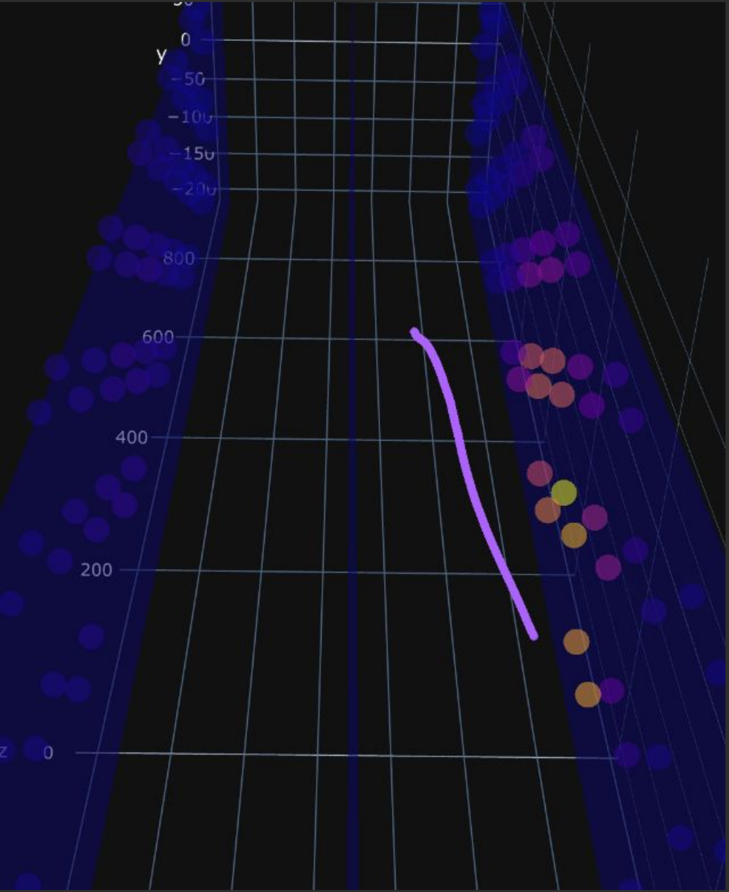
Photo-multiplier tubes (PMTs) detect scintillation photons produced isotropically from an Argon atom

1 meter muon produces ~ **5M photons**

Optical Photon Transport



Differentiable Surrogate as a Simulator



Control dataset: 3D TPC trajectory for which XYZ position of space-points are accurately measured

Predicted P.E.

Deposited charge at the point i

Quantum efficiency of the PMT j

$$P_j = \sum_{i}^{\text{all points}} C_i \times Y \times Q_j \times \Phi(\mathbf{r}_i)_j$$

light yield

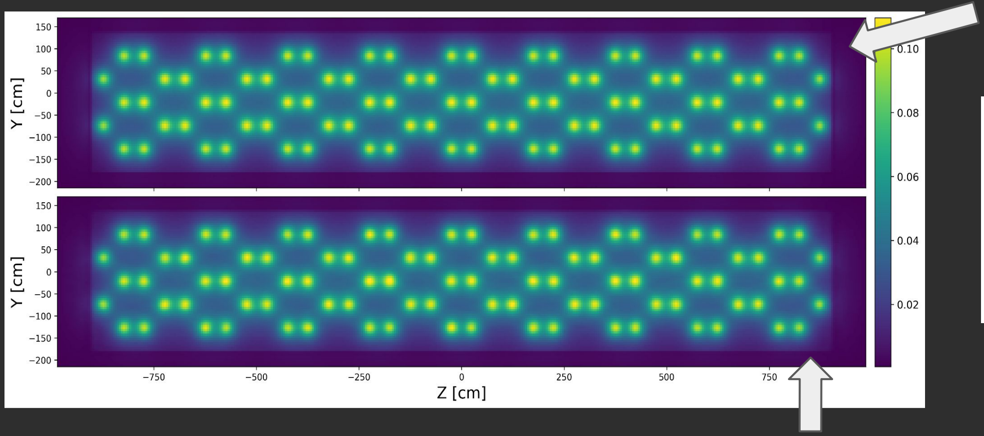
SIREN prediction for the point i at the PMT j

$$\text{Loss} = \sum_{j}^{\text{all PMTs}} \frac{(P_j - O_j)^2}{P_j^2 + \epsilon}$$

numerical stability, ~25 P.E.-squared

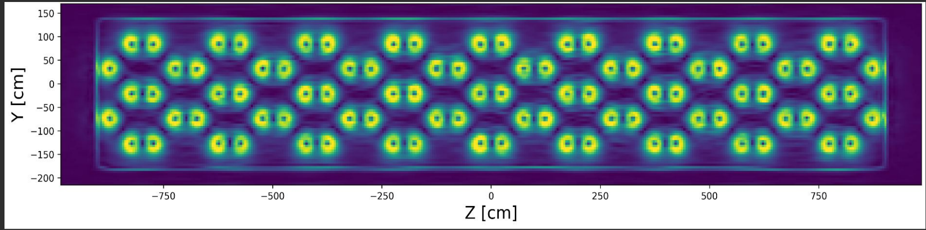
Differentiable Surrogate as a Simulator

Simulation



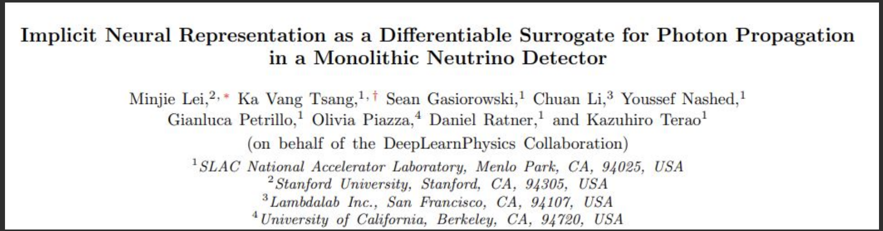
SIREN also learns the gradient map

SIREN



SIREN as a differentiable surrogate is used for data reconstruction in addition to simulations

Preprint [arXiv:2210.01505](https://arxiv.org/abs/2210.01505)


Implicit Neural Representation as a Differentiable Surrogate for Photon Propagation in a Monolithic Neutrino Detector

Minjie Lei,[2,*] Ka Vang Tsang,[1,†] Sean Gasiorowski,[1] Chuan Li,[3] Youssef Nashed,[1] Gianluca Petrillo,[1] Olivia Piazza,[4] Daniel Ratner,[1] and Kazuhiro Terao[1]
(on behalf of the DeepLearnPhysics Collaboration)
[1] *SLAC National Accelerator Laboratory, Menlo Park, CA, 94025, USA*
[2] *Stanford University, Stanford, CA, 94305, USA*
[3] *Lambdalab Inc., San Francisco, CA, 94107, USA*
[4] *University of California, Berkeley, CA, 94720, USA*

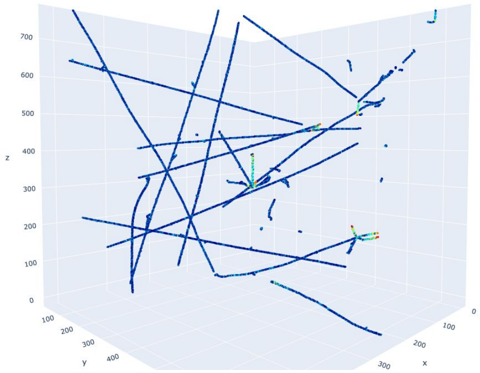
Work credit (from left): Olivia P. (UC Berkeley), Minjie L. (SLAC), Patrick T. (SLAC), , Gordon W. (Stanford CS), Chuan L. (Lambda Labs)

Collaborative work between SLAC, Stanford CS, and Lambda Labs (AI start up in SF)

Heading Where?
Present and Future R&D

Differentiable Physics Model: Many Applications

Beyond being a self-calibrating machine…
e.g.) solve the inverse problem (unfolding the detector effects)



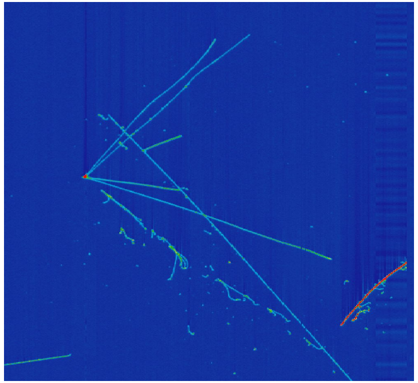
$\mathbf{X} \in \mathcal{D}_I$

Input domain of detector process (simulation-only)

Backpropagate to solve (optimize) for the input

$F\,(Y|X,\,\theta_F)$

Differentiable LArTPC Simulator



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

Output domain of detector process (inc. real data)

Differentiable Physics Model: Many Applications

Beyond being a self-calibrating machine…
… or use as a regularization + enable real-data training for a NN inverse solver

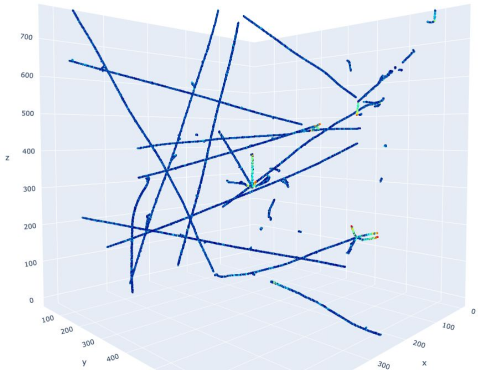
G (X|Y, θ_G)
Inverse Image Solver

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

and / or

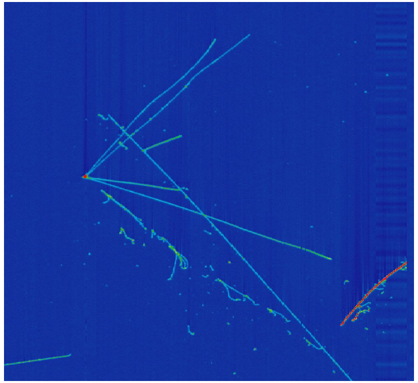
$$\mathcal{L}_{\text{cc}} = |F(G(\mathbf{Y})) - \mathbf{Y}|^2$$

F (Y|X, θ_F)
Differentiable LArTPC Simulator



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

Input domain of LArTPC simulator (inaccessible)



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

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

Closing Overlook

Future Directions of AI/ML and Science

SciML: science for AI/ML

- Multi-task, end-to-end optimizable, hierarchical object reconstruction chain
- Differentiable physics modeling (both explicit and neural representations)
- Things briefly glanced / skipped:
  - Diffusion model as a non-iterative inverse solver for detector physics
  - Foundation model for particle detector physics with multi-modal input
  - Generative graph for modeling a many-body system (nuclear FSI, cosmo.)
  - Calibrated model uncertainty quantification methods (e.g. paper)
- Common themes across the threads:
  - Explainability/interpretability, reusability, automation, acceleration
  - Enable new techniques or orders of magnitude improvement

Fun

CatGPT

What if ChatGPT were a cat?

Meow, meow meow meow, meow meow?

What are the goals of the ICARUS experiment?

Meow meow meow meow meow meow meow meow meow meow meow meow meow meow meow.

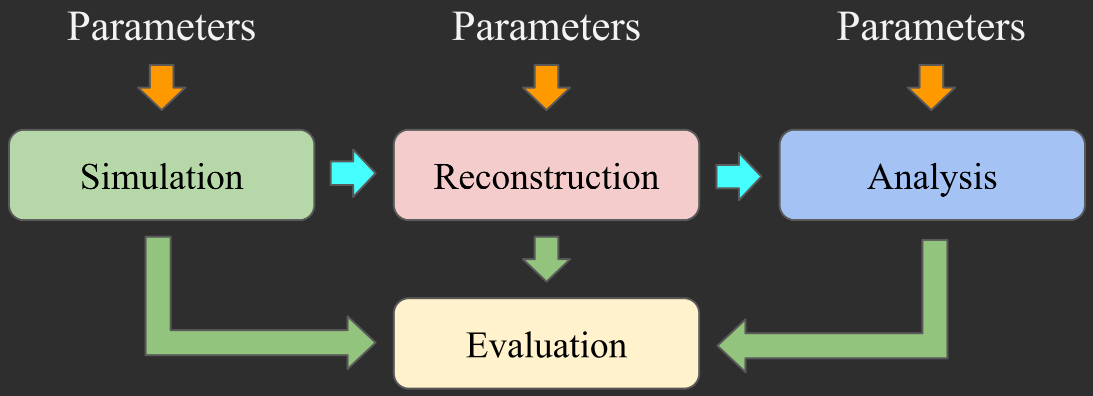
I learned something. Thank you.

Meow meow meow meow meow meow meow meow meow meow meow meow meow meow meow.

Back-up Slides

An Approach Opposite to SciML Foundation Models ~ Learning Physics w/ Representations from Real Data ~

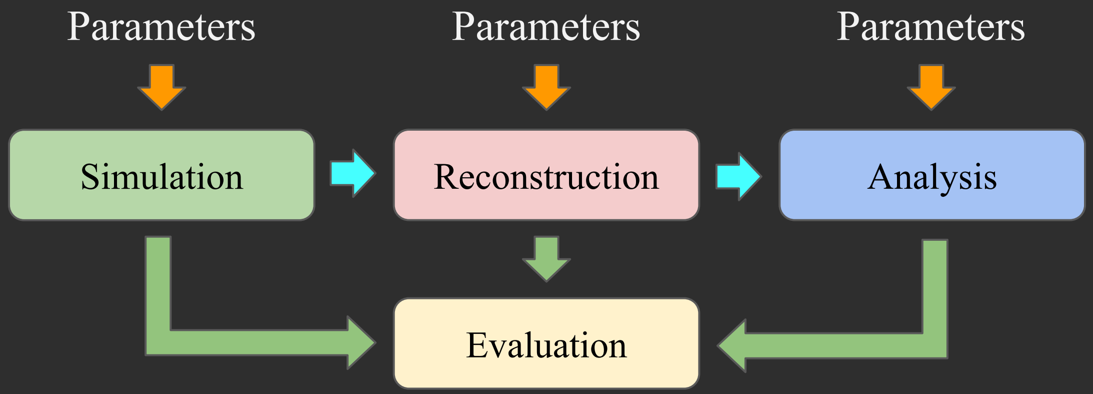
Toward Fully SciML-based Pipeline



Enables:

- Automated optimization of model parameters for the entire workflow
- Physics simulation and inference, design optimization, UQ study
- Explainable & interpretable by design where physics knowledge applied

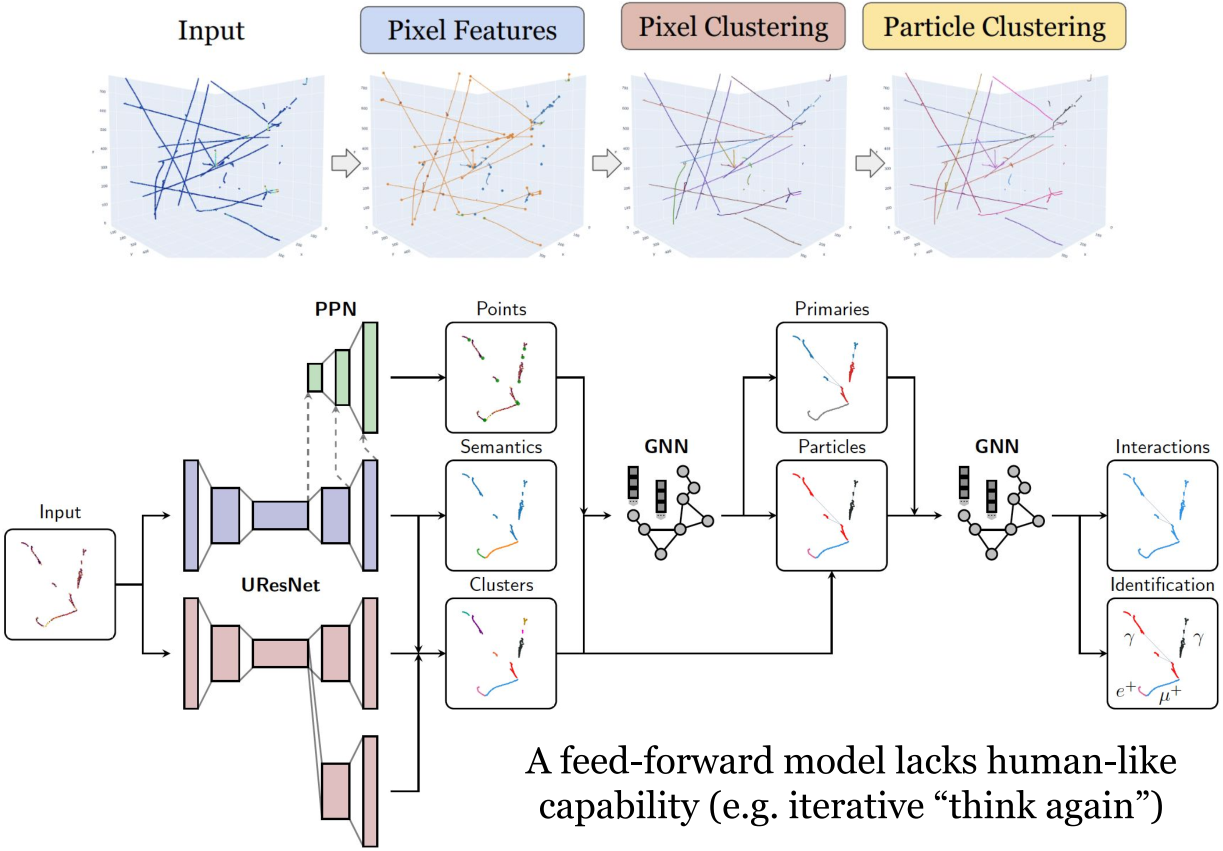
Toward Fully SciML-based Pipeline



… yet, lots of challenges!:

- Non-differentiable operations (e.g. stochastic and/or discrete processes)
- Might suffer from a “missing (new) physics model” in data
- Feed-forward: the model architecture is fixed = cannot “discover”

Toward Fully AI/ML-based Pipeline



A feed-forward model lacks human-like capability (e.g. iterative “think again”)

New AI/ML Tool: Foundation Models

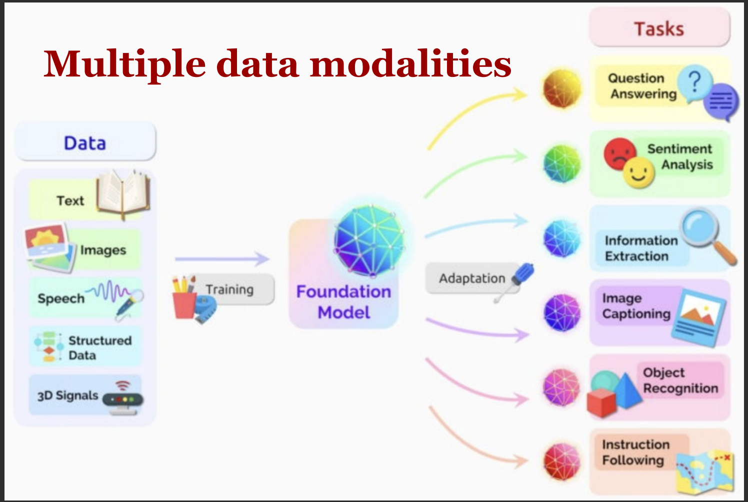
Self-supervision
Learn from co-occurrence patterns.
"The color of an apple is ▢."
Can utilize lots of unlabeled data!

"Representation Learning"
Must learn the universe represented by data

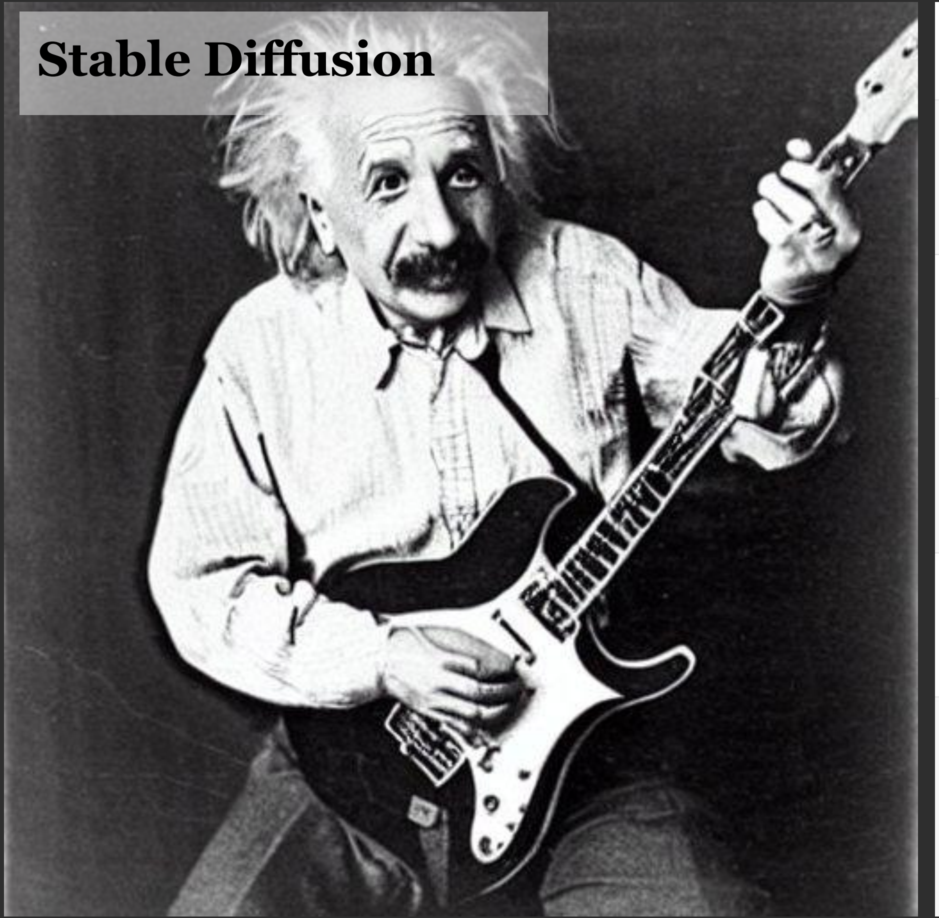
Transfer Learning
Conceptualizing the world, the model can be adopted to perform various tasks (task-agnostic)

Rep. Learning
(via self-supervision)

Transfer Learning
(fine-tuning w/ labels)



Foundation Models



(Chat-) GPT

Let me ask questions about logic and commonsense

Sure, I'll do my best to answer any questions you have about logic and common sense. Please go ahead and ask!

I give you a premise followed by a question.
Tell me your answer, either Yes or No, followed by your reasoning.

Sure, I'll do my best to answer your questions based on the given premise. Please go ahead and provide the premise and question.

Premise: I have not had food.
Question: Have I had friedrice?

Based on the premise that you have not had any food, my answer would be "No, you have not had fried rice."

My reasoning is that if you have not had any food, it would be impossible for you to have had fried rice, which is a type of food.

R&D: A shared “detector physics” model

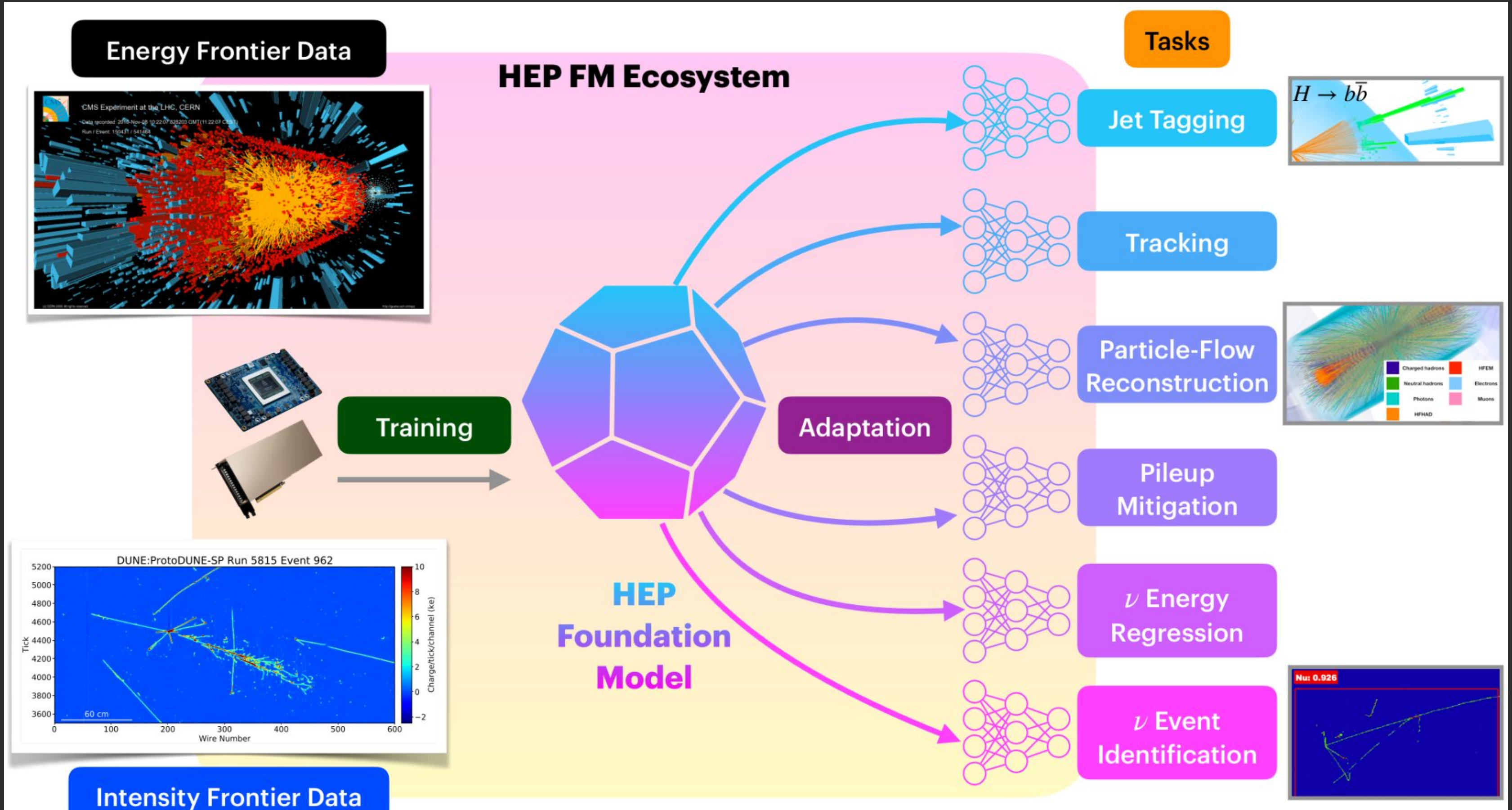
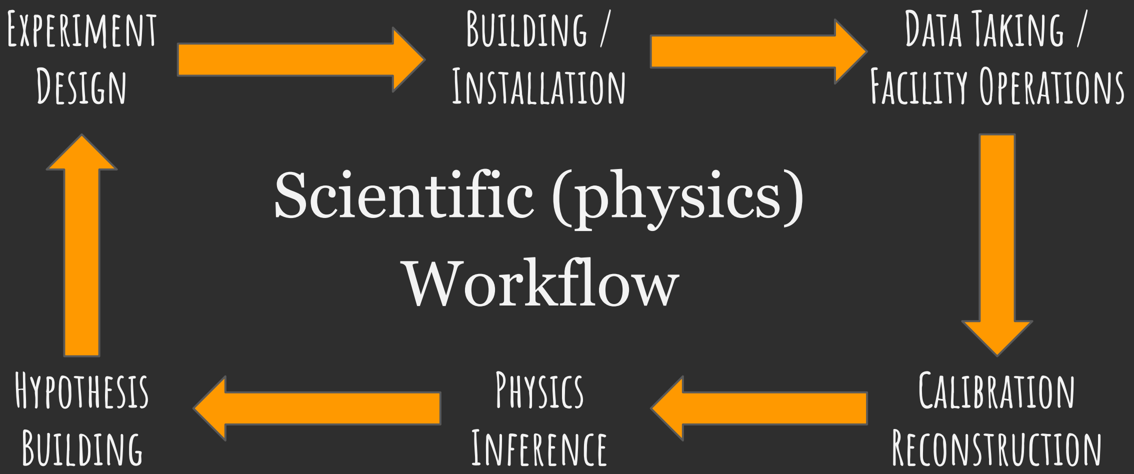


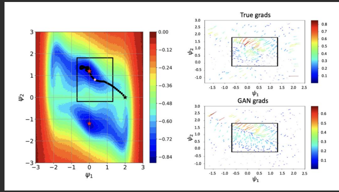
Image credit: Javier Duarte (CMS/UCSD)

What’s in the “Experimental Physics Pipeline”?

Scientific (physics) Workflow



Aparatus design optimization
(2002.04632)



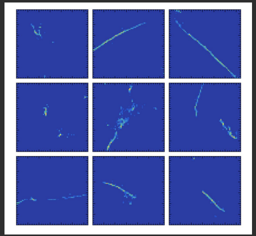
AR for support (NASA)

Fast-ML / Edge-ML
(HLS4ML)

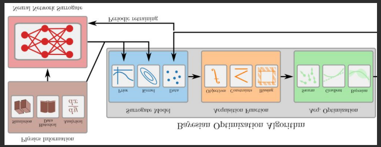
EXPERIMENT DESIGN → BUILDING / INSTALLATION → DATA TAKING / FACILITY OPERATIONS → CALIBRATION RECONSTRUCTION → PHYSICS INFERENCE → HYPOTHESIS BUILDING → EXPERIMENT DESIGN

AI/ML is Everywhere

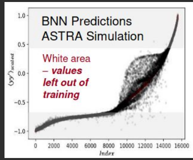
Fast Simulation
(2204.02496)



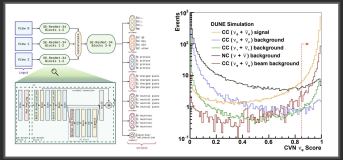
Bayesian Optimization
Reinforcement Learning
(2010.09824, 2202.07747)



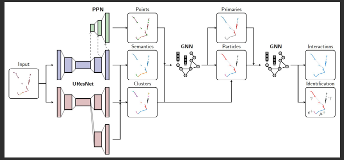
Uncertainty Quantification
(PRAB 24.114601)



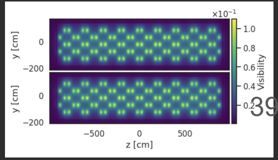
Physics Inference
(2006.15052)



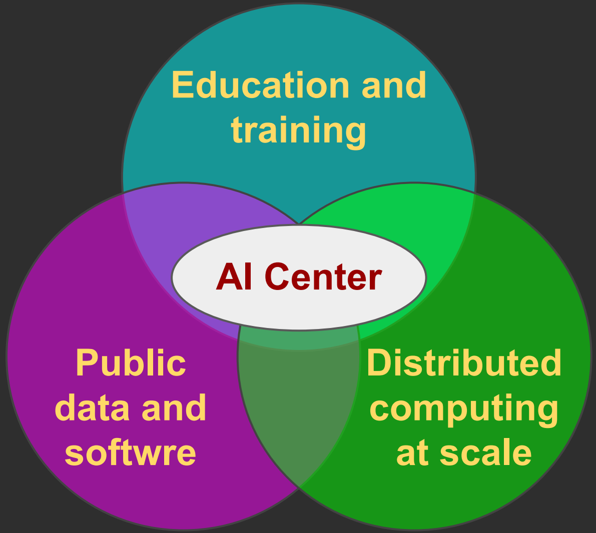
Object Reconstruction
(2102.01033)



Detector Response
Calibration
(2102.01033)



Closing Note: Ecosystem for AI/ML Research



HEP Ecosystem for AI research

- Accessible **education and training** at all levels
- **Interdisciplinary research environment** (e.g. workshops, hackathons, visiting scholars program)
- **Shared computing resources** available always
- **Open and reusable datasets and software** with documentation and performance metrics

- **Ethics of AI**: how to ensure diversity, equity, and inclusion which is already terrible in STEM? How to ensure small and large projects both benefits from AI?

AI is an accelerator. It is coming. Don't avoid.
Participate to make sure the use is good.

Examples of Scientific ML (SciML)

Scientific AI/ML by Physics

Parameters

$\downarrow$

Input $\rightarrow$ **Algorithm** $\rightarrow$ Output $\rightarrow$ **Evaluation** $\rightarrow$ Objectives

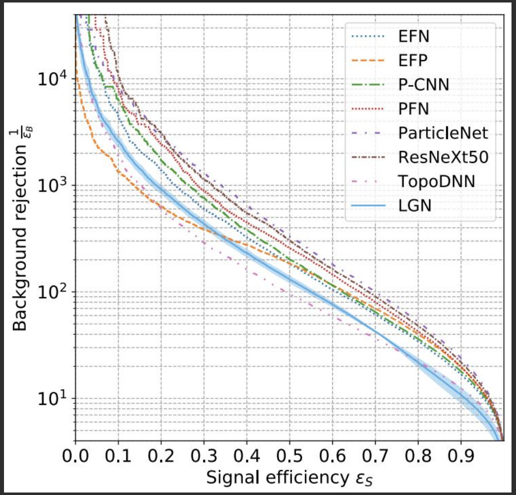
Inject inductive bias (physics knowledge)

1. Given a neural network, inject physics
2. Given a physics model, equip with the AI/ML tools

Inductive Bias: Injecting Physics Knowledge

Parameters

Input → **Algorithm** → Output → **Evaluation** → Objectives

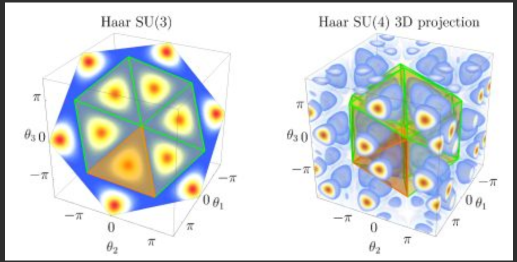


Lorentz Group Equivariant Neural Networks ([A. Bogatskiy et al.](#))

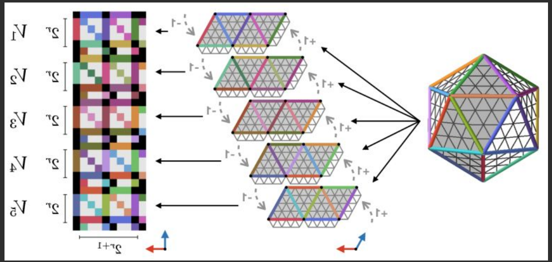
Analog to CNNs: Add constraints to the math operations in the algorithm to preserve the invariance under certain transformations

e.g.) “Lorentz invariant” neural network

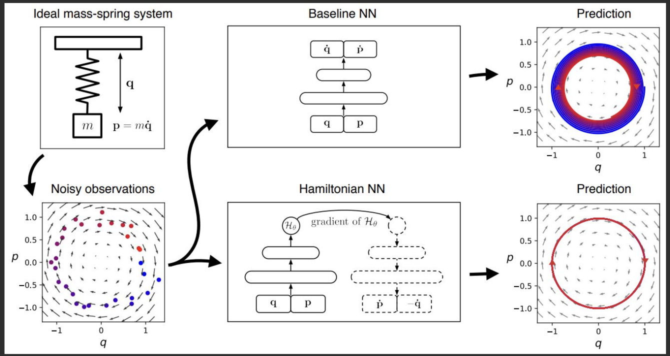
Inductive Bias: Injecting Physics Knowledge



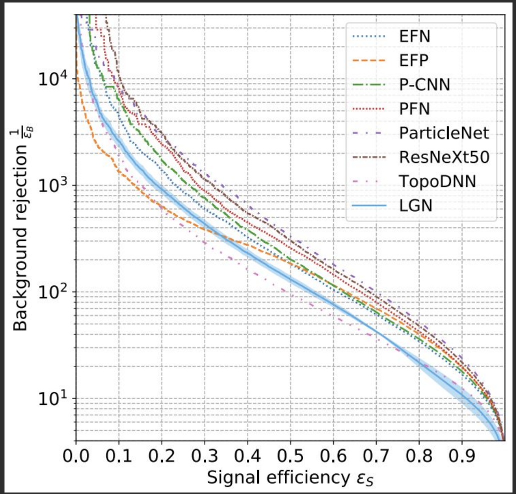
Sampling using SU(N) gauge equivalent flows ([D. Boyda et al.])



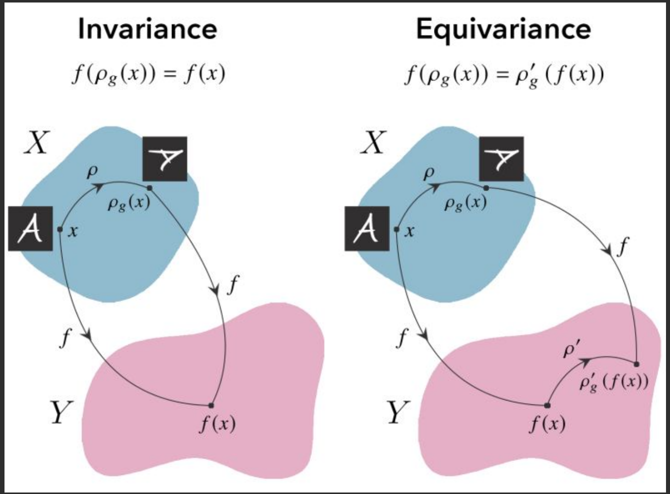
Gauge Equivariant CNN ([T.S. Cohen et al.])



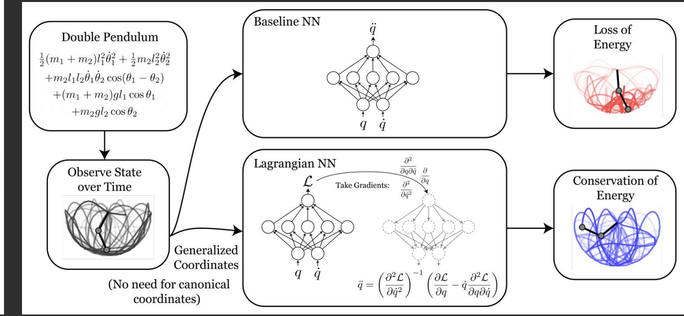
Hamiltonian Neural Networks ([S. Greydanus et al.])



Lorentz Group Equivariant Neural Networks ([A. Bogatskiy et al.])



Symmetry Group Equivariant Architecture ([A. Bogatskiy et al.])



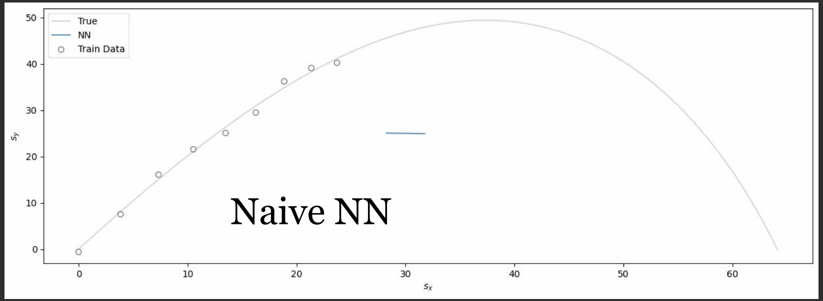
Lagrangian Neural Networks ([M. Cranmer et al.])

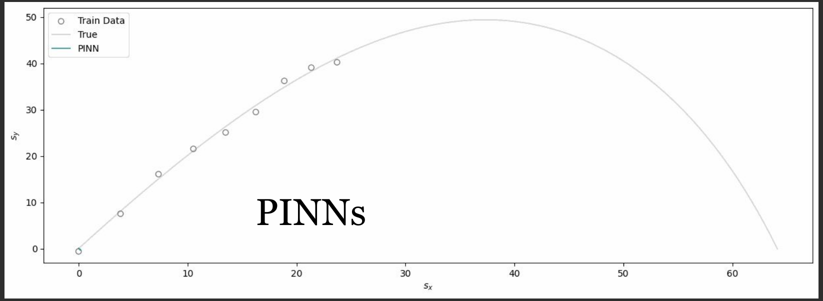
Inductive Bias: Injecting Physics Knowledge

Parameters

↓

Input → **Algorithm** → Output → **Evaluation** → Objectives





Introduce physics concepts in the loss as regularizations (e.g. boundary conditions, partial differential equations)

e.g.) Physics-Informed Neural Networks

([M. Raissi et al.](#))

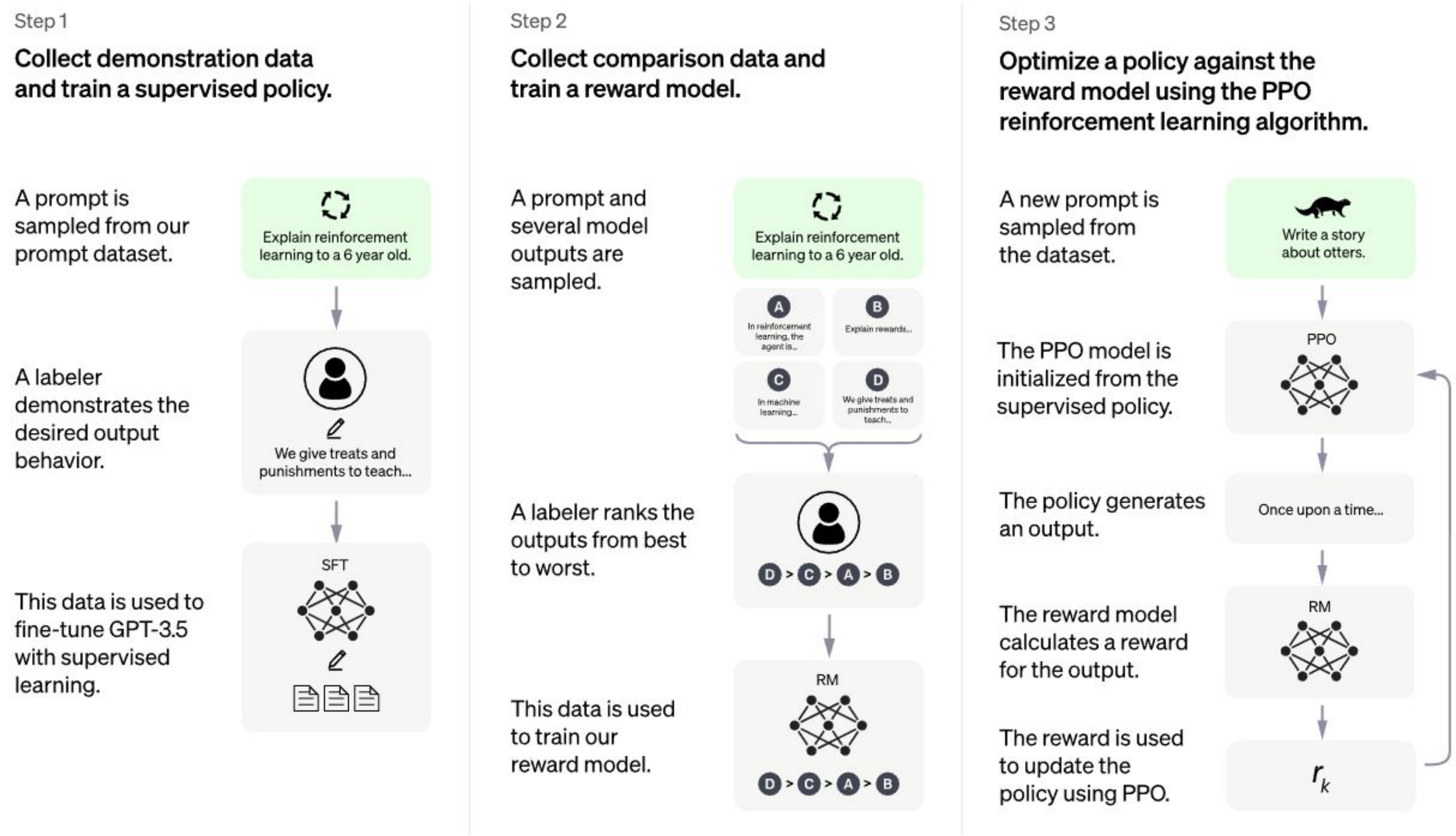
$$L = \|x(0) - x_0\| + \left\|\frac{dx(0)}{dt} - v_0\right\| + \sum_i \left\| m\frac{d^2x(t^i)}{dt^2} - mg \right\|$$

Free-fall physics model regularizations for the loss

Human-in-the-Loop Optimization of Chat-GPT

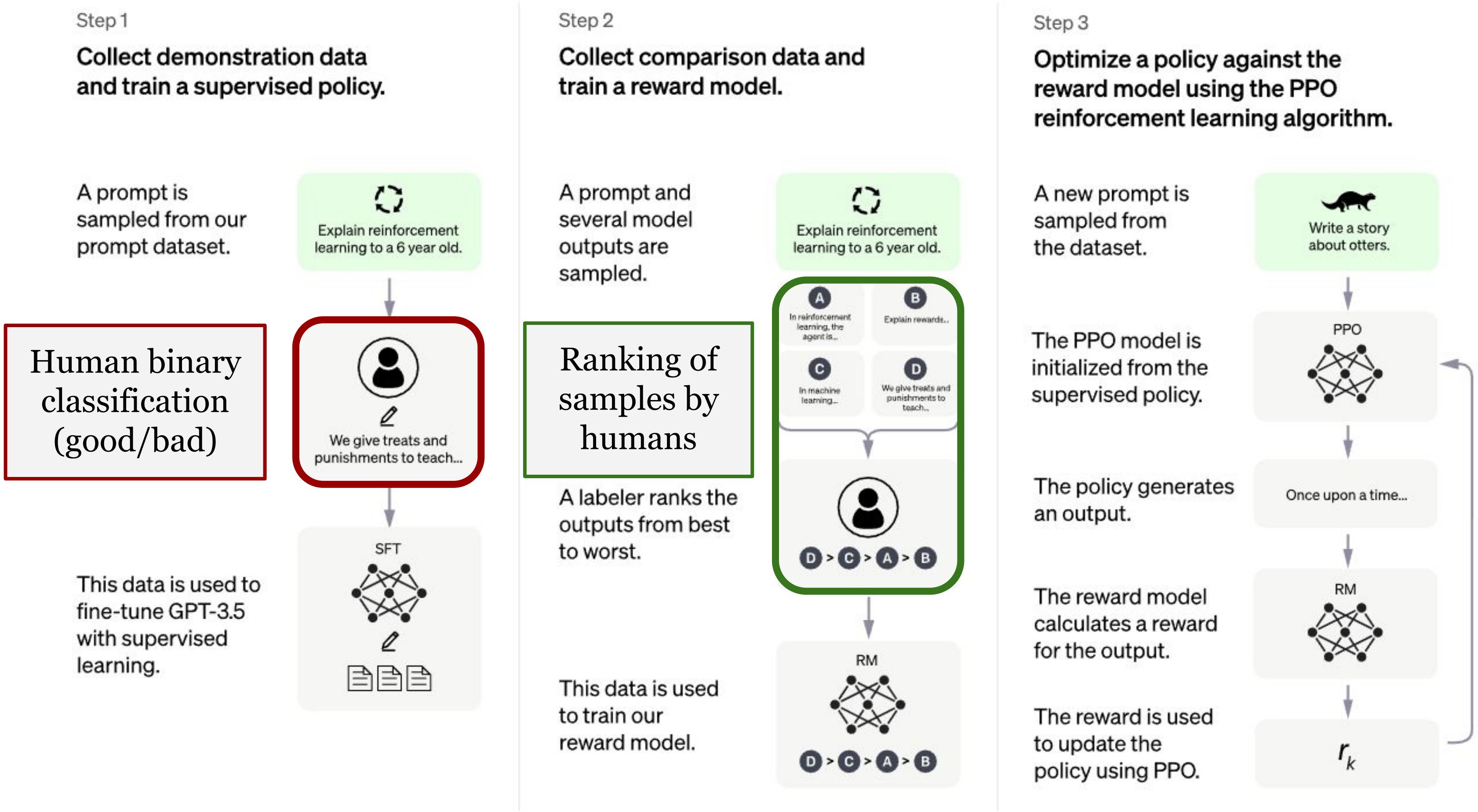
Human-in-the-Loop (RLHF)

[OpenAI ChatGPT blog post]



Human-in-the-Loop (RLHF)

[OpenAI ChatGPT blog post]



Additional Details
Neural Scene Representation for Optical Transport Modeling

Differentiable surrogate for optical photon transport

Optical Detector
Simulation

ML for Detector Physics Modeling
LAr scintillator light detection

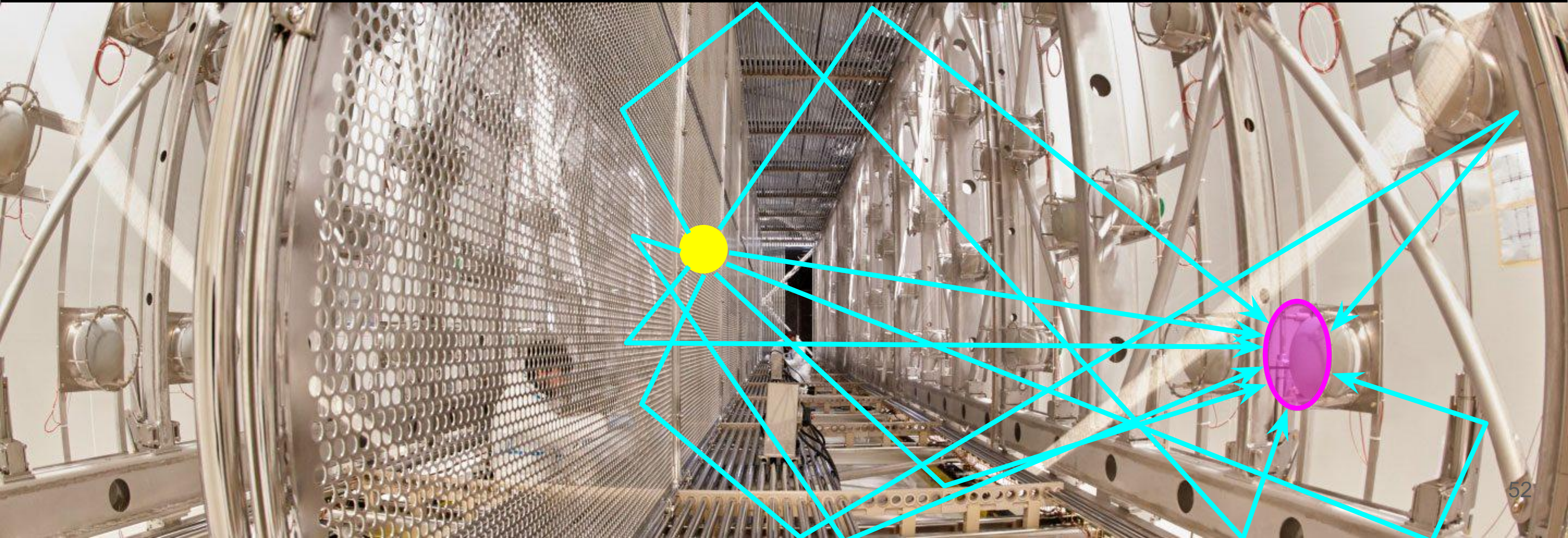
Photo-multiplier tubes (PMTs) detect scintillation photons

Optical Photon Transport

ML for Detector Physics Modeling
LAr scintillator light detection

Photo-multiplier tubes (PMTs) detect scintillation photons produced isotropically from an Argon atom
1 meter muon produces ~ **5M photons**

Optical Photon Transport



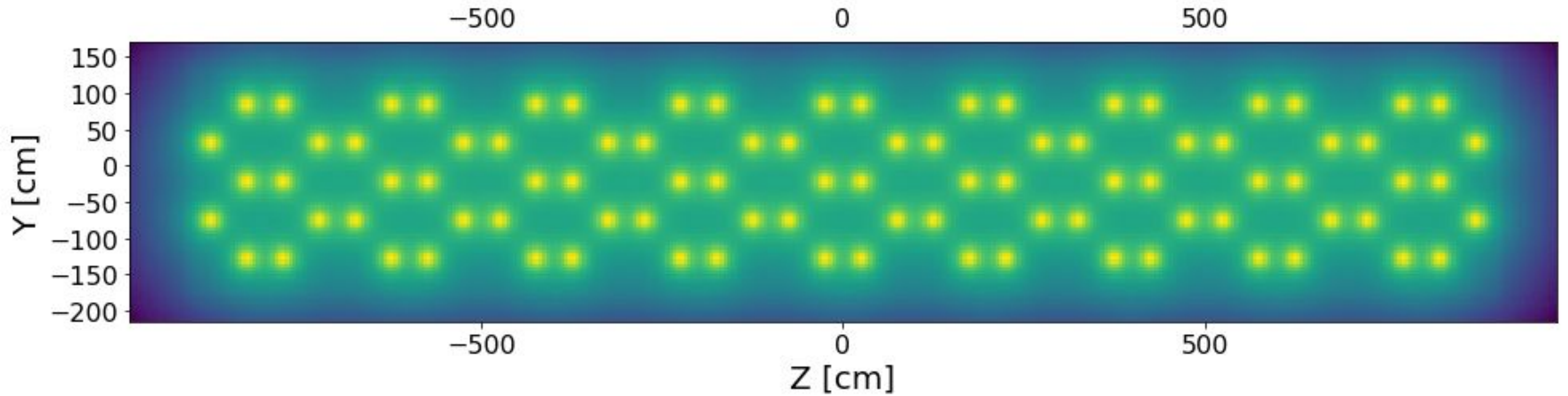
ML for Detector Physics Modeling
LAr scintillator light simulation

SLAC

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

Issue: static and **not scalable**

Optical Photon Transport



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

ML for Detector Physics Modeling
LAr scintillator light simulation

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

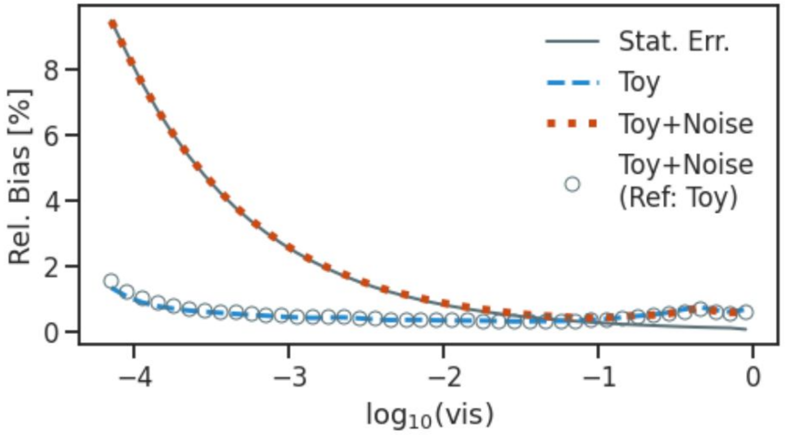
Issue: static and **not scalable**

- Implicitly optimized based on simulation update (~2 weeks to produce each time)
- Limited scalability … ~1E9 voxels for ICARUS
  - Coarse voxel size (~5cm cubic)
  - Large statistical error (~30k photons/vox.)

Difficult to scale full DUNE

Optical Photon Transport

$$\text{Relative Bias} = \frac{\left|\langle \text{P.E.}\rangle_{\text{true}} - \langle \text{P.E.}\rangle_{\text{pred.}}\right|}{\langle \text{P.E.}\rangle_{\text{true}} + \langle \text{P.E.}\rangle_{\text{pred.}}}$$

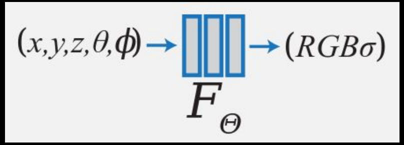


Example: ICARUS detector, 2D slice of a 3D map

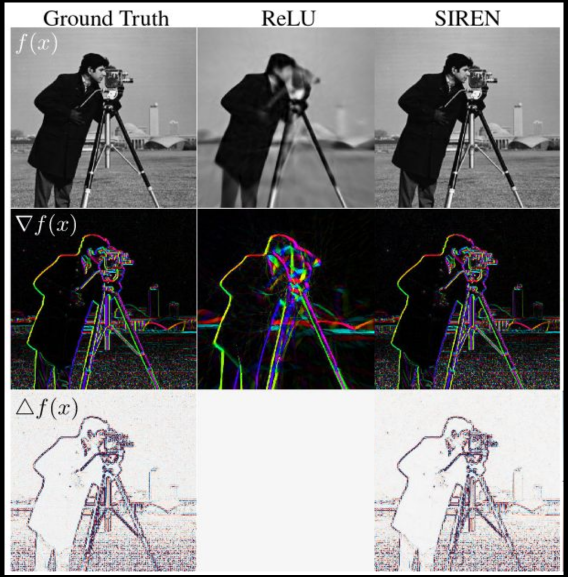
ML for Detector Physics Modeling

SIREN as a differentiable surrogate for optical detectors

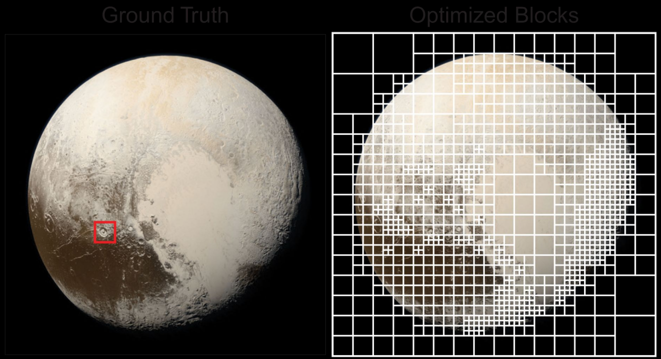
Differentiable Neural Scene Representation



NeRF breakthrough on high resolution image representation by a very simple nerual network



SIREN success of learning the 1st and 2nd order derivatives



ACORN scalable version of SIREN by adding spatial feature compression (essentially a learnable kd-tree)

… only a few examples

ML for Detector Physics Modeling

SIREN as a differentiable surrogate for optical detectors

SLAC

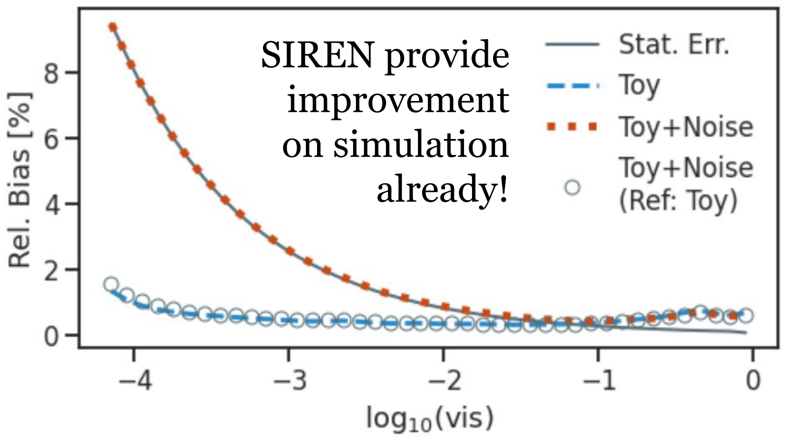
Differentiable Neural Scene Representation

SIREN for LArTPC detectors

- Designed as an implicit representation of a continuous function in space (suited to “visibility”, “E-field”, etc.)
  - Can be seen as a trade-off between an analytical function and a table
- “Differentiable” implies we can directly optimize against “data v.s. simulation discrepancy” given control samples

SIREN trained on “Toy + Noise” successfully learned the underlying analytical function shape (simulation)

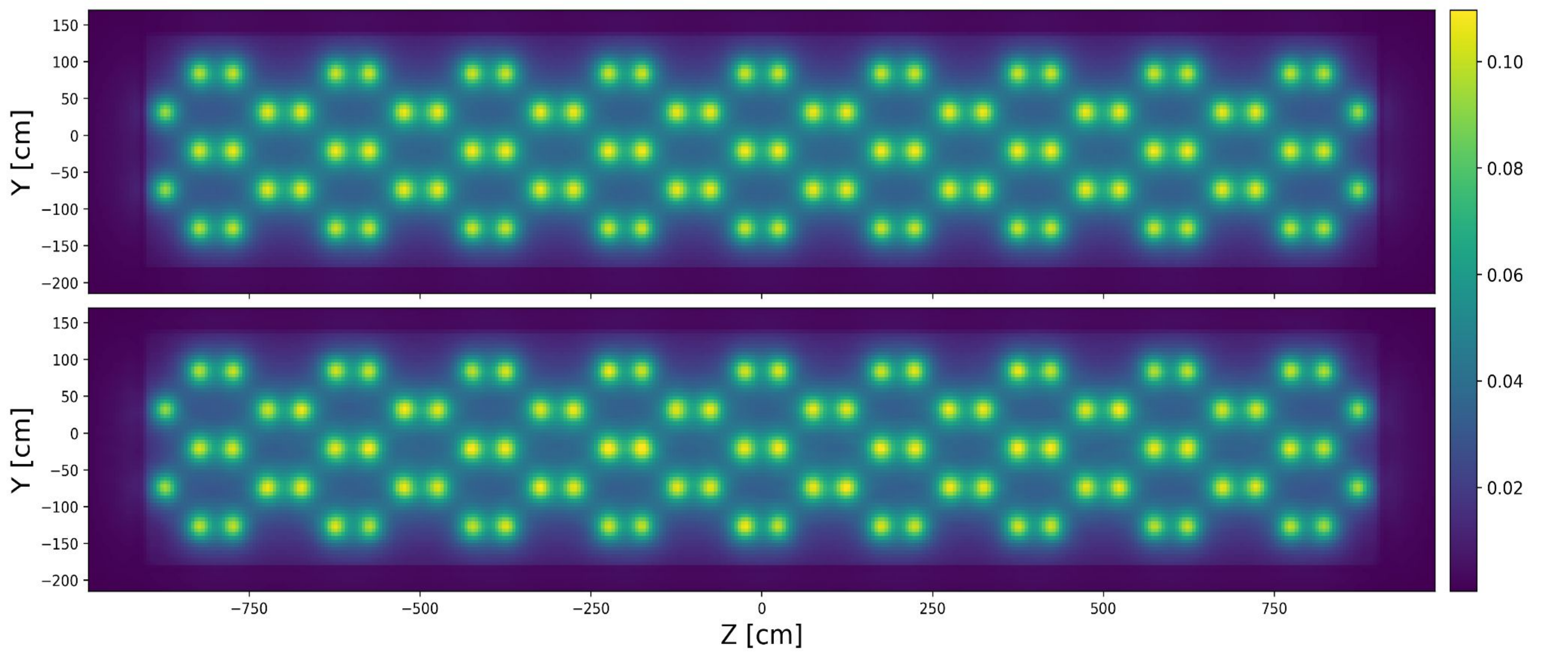
$$\text{Relative Bias} = \frac{\left|\langle \text{P.E.} \rangle_{\text{true}} - \langle \text{P.E.} \rangle_{\text{pred.}}\right|}{\langle \text{P.E.} \rangle_{\text{true}} + \langle \text{P.E.} \rangle_{\text{pred.}}}$$



ML for Detector Physics Modeling

SIREN as a differentiable surrogate for optical detectors

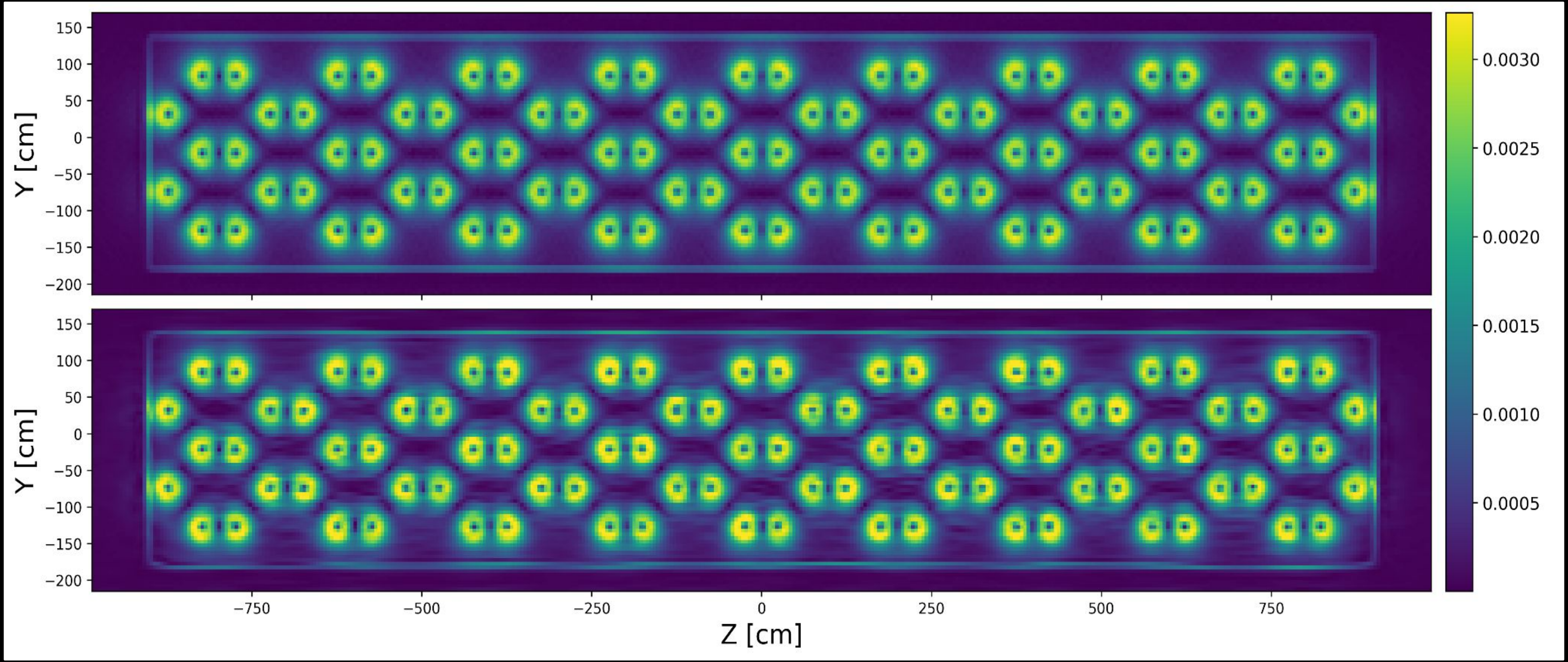
ICARUS: 2D slice, map (top) v.s. SIREN (bottom)



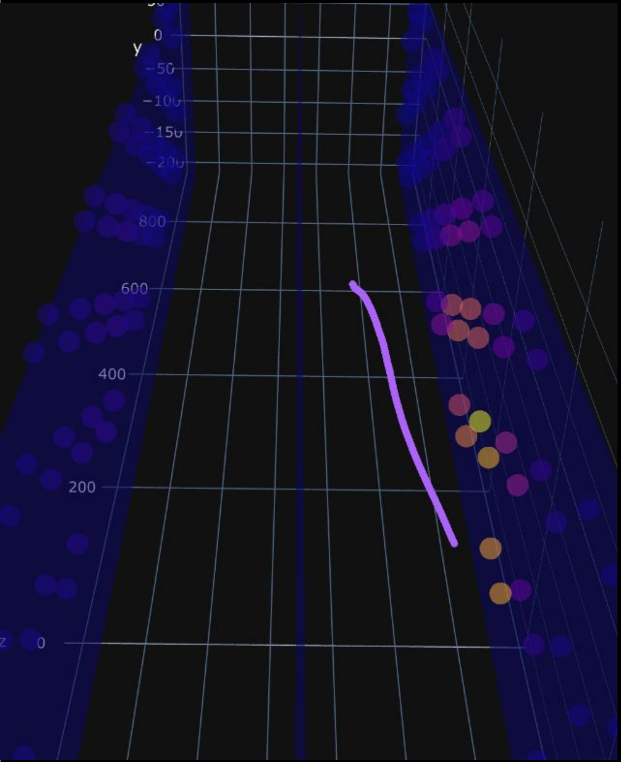
ML for Detector Physics Modeling

SIREN as a differentiable surrogate for optical detectors

ICARUS: 2D slice, map (top) v.s. SIREN (bottom)



Training SIREN on real data



Control dataset: 3D TPC trajectory for which XYZ position of space-points are accurately measured

Predicted P.E.

Deposited charge at the point i

Quantum efficiency of the PMT j

$$P_j = \sum_{i}^{\text{all points}} C_i \times Y \times Q_j \times \Phi(\mathbf{r}_i)_j$$

light yield

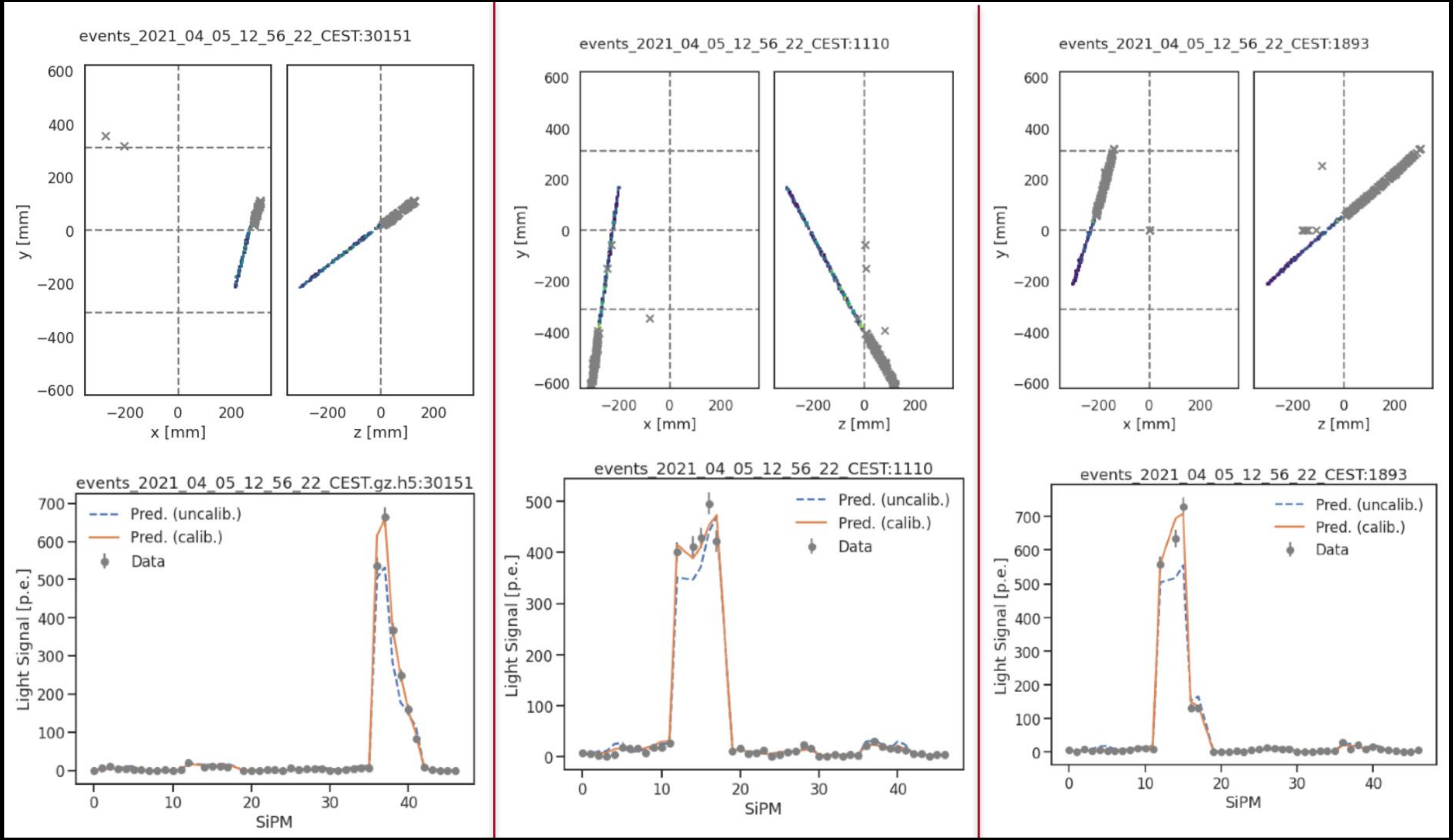
SIREN prediction for the point i at the PMT j

$$\text{Loss} = \sum_{j}^{\text{all PMTs}} \frac{(P_j - O_j)^2}{P_j^2 + \epsilon}$$

numerical stability, ~25 P.E.-squared

ML for Detector Physics Modeling

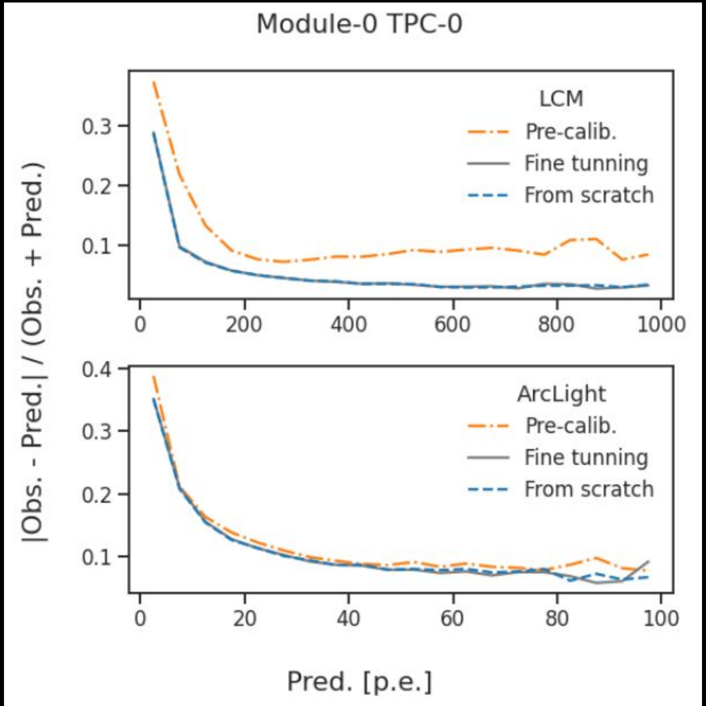
SIREN as a differentiable surrogate for optical detectors

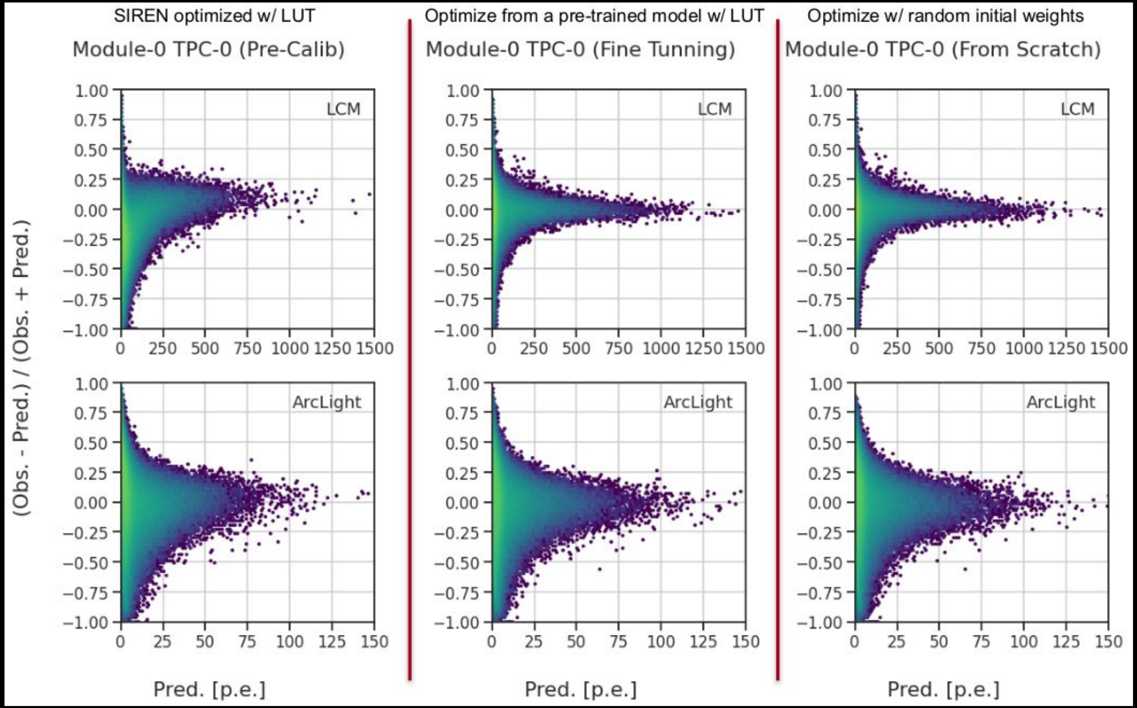


ML for Detector Physics Modeling

SIREN as a differentiable surrogate for optical detectors

Training SIREN on real data





ML for Detector Physics Modeling

SIREN as a differentiable surrogate for optical detectors

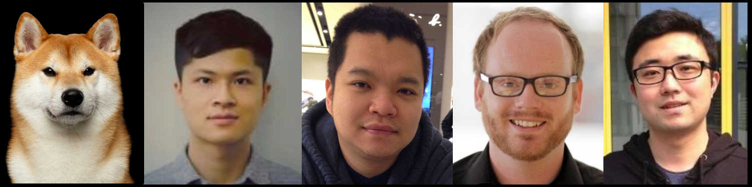
Implicit Neural Representation as a Differentiable Surrogate for Photon Propagation in a Monolithic Neutrino Detector

Minjie Lei,[2,*] Ka Vang Tsang,[1,†] Sean Gasiorowski,[1] Chuan Li,[3] Youssef Nashed,[1]
Gianluca Petrillo,[1] Olivia Piazza,[4] Daniel Ratner,[1] and Kazuhiro Terao[1]
(on behalf of the DeepLearnPhysics Collaboration)

[1] *SLAC National Accelerator Laboratory, Menlo Park, CA, 94025, USA*
[2] *Stanford University, Stanford, CA, 94305, USA*
[3] *Lambdalab Inc., San Francisco, CA, 94107, USA*
[4] *University of California, Berkeley, CA, 94720, USA*

Optical photons are used as signal in a wide variety of particle detectors. Modern neutrino experiments employ hundreds to tens of thousands of photon detectors to observe signal from millions to billions of scintillation photons produced from energy deposition of charged particles. These neutrino detectors are typically large, containing $\mathcal{O}(10^2 - 10^5)$ tons of target volume, and may consist of many materials with different optical properties. As a result, modeling individual photon propagation requires prohibitive computational resources. As an alternative to tracking individual photons, the experimental community has traditionally used a *look-up table*, which contains a mean probability of observing a photon per photon detector at each grid location in a uniformly voxelized detector volume. However, since the size of a table increases with detector volume for a fixed resolution, this method scales poorly for future larger detectors. Alternative approaches such as fitting a polynomial to the model could address the memory issue, but results in poorer performance. Furthermore, both look-up table and fitting approaches are prone to discrepancies between the detector simulation and the real-world detector response. We propose a new approach using SIREN, a implicit neural representation with periodic activation functions. In our approach, SIREN is used to model the look-up table as a "3D scene" and reproduces the acceptance map with high accuracy. The number of parameters in our SIREN model is orders of magnitude smaller than the number of voxels in the look-up table. As it models an underlying functional shape, SIREN is scalable to a larger detector. Furthermore, SIREN can successfully learn the spatial gradients of the photon library, providing additional information for downstream applications. Finally, as SIREN is a neural network representation, it is differentiable with respect to its parameters, and therefore tunable via gradient descent. We demonstrate the potential of optimizing SIREN directly on real data, which mitigates the concern of data vs. simulation discrepancies. We further present an application for data reconstruction where SIREN is used to form a likelihood function for photon statistics.

Preprint arXiv:2210.01505



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

Challenges for Differentiable Simulators

Differentiable Physics Models

Challenges: physics models involve stochastic, discrete operations that are not differentiable as they are.

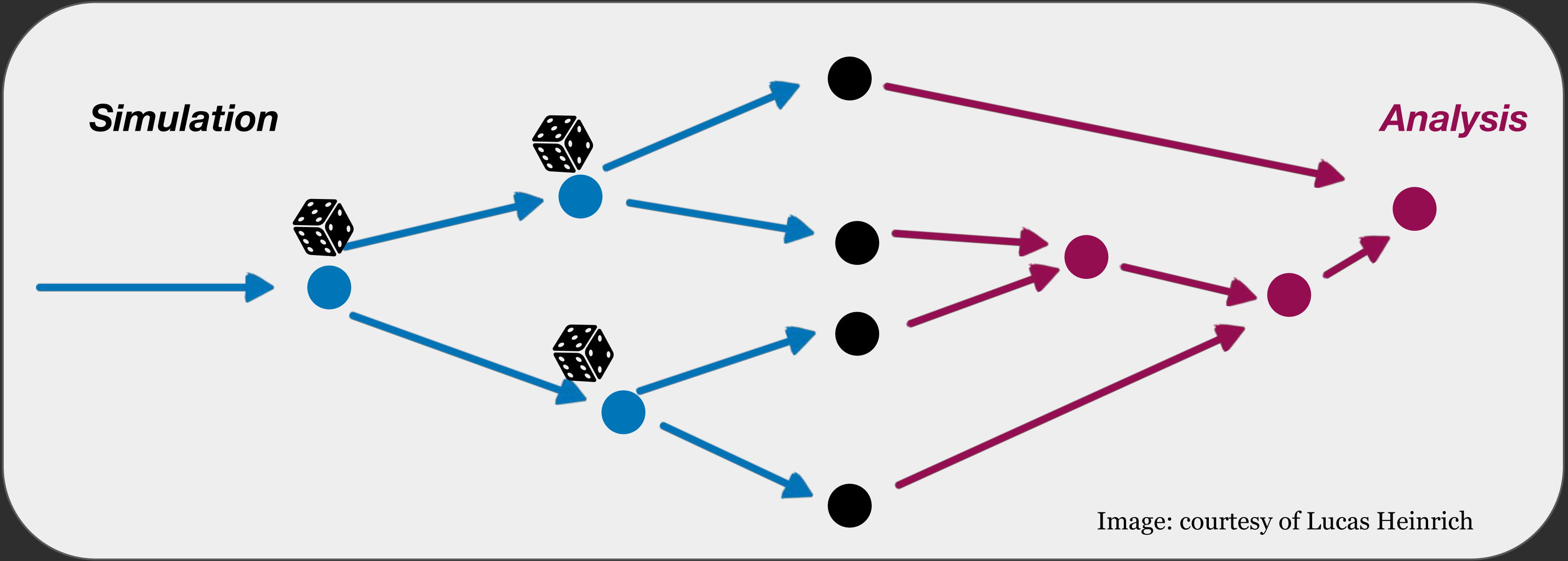


Image: courtesy of Lucas Heinrich

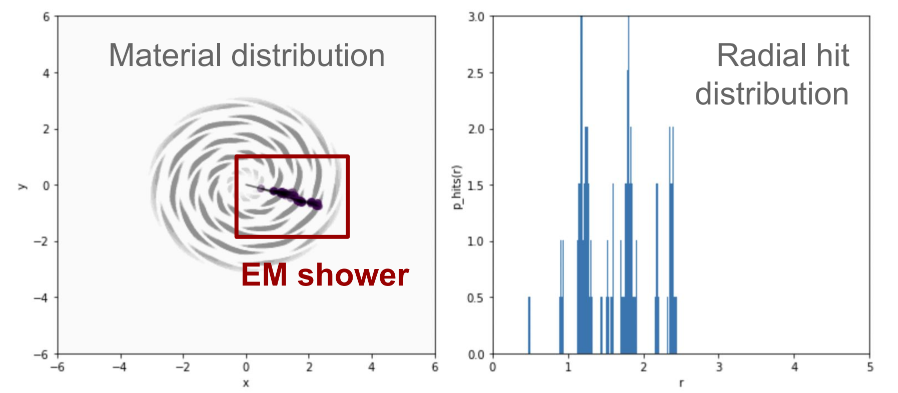
Differentiable Physics Models

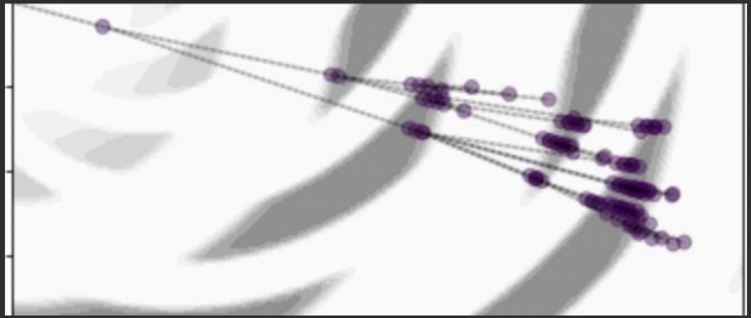
Challenges: physics models involve stochastic, discrete operations that are not differentiable as they are. But expectation values over statistics are usually smooth and differentiable (e.g. AI playing a game)

Differentiable Physics Models

Challenges: physics models involve stochastic, discrete operations that are not differentiable as they are. But expectation values over statistics are usually smooth and differentiable (e.g. AI playing a game)

Simple experiment: optimize the calorimeter radius to contain a shower

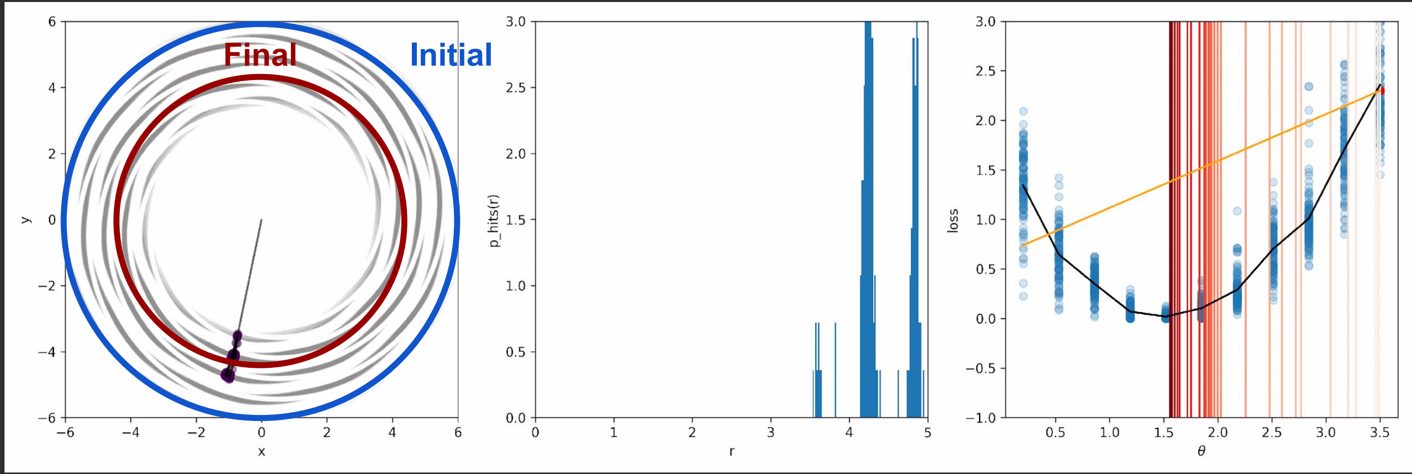




Differentiable Physics Models

Challenges: physics models involve stochastic, discrete operations that are not differentiable as they are. But expectation values over statistics are usually smooth and differentiable (e.g. AI playing a game)

Simple experiment: optimize the calorimeter radius to contain a shower



"Noisy gradient"
But it works to find the optimal radius correctly.

Figures courtesy of
Lucas Heinrich

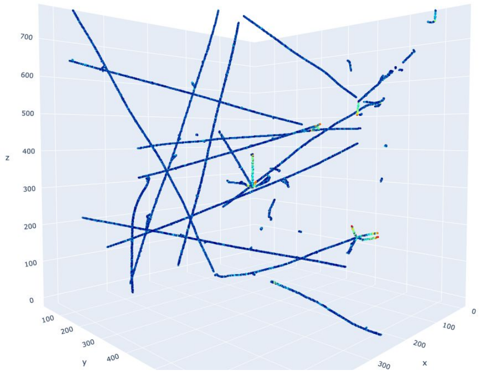
Extents of Detector Inverse Solvers

Inverse imaging using a differentiable simulator

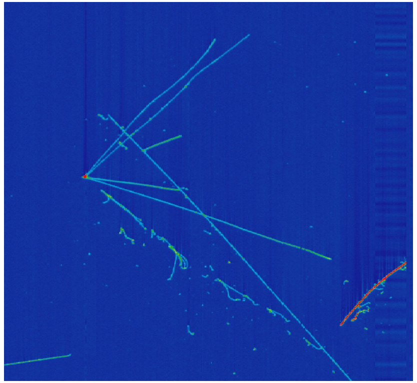
E.g. use for optimizing an image inverse solver

G (X|Y, θ_G)

Inverse Image Solver



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



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

Input domain of LArTPC simulator (inaccessible)

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

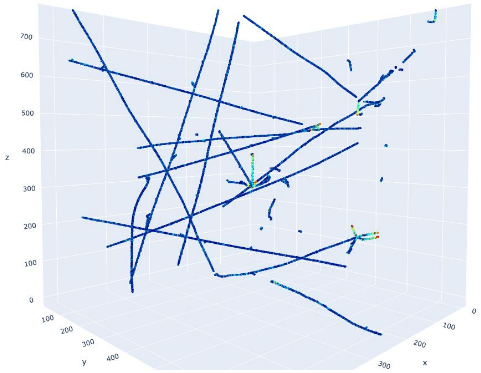
Output domain of LArTPC simulator (e.g. real data)

Inverse imaging using a differentiable simulator

E.g. use for optimizing an image inverse solver

$G\,(X|Y,\,\theta_G)$

Inverse Image Solver



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

Input domain of LArTPC simulator (inaccessible)

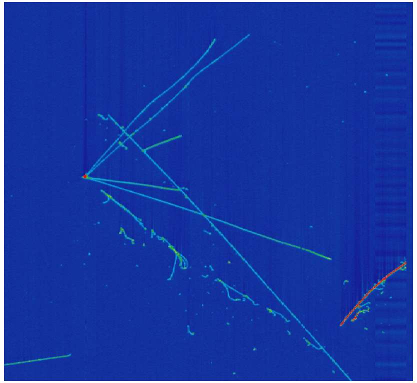
$$\mathcal{L}_{\text{inv}} = |G(\mathbf{Y}) - \mathbf{X}|^2$$

and / or

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

$F\,(Y|X,\,\theta_F)$

Differentiable LArTPC Simulator

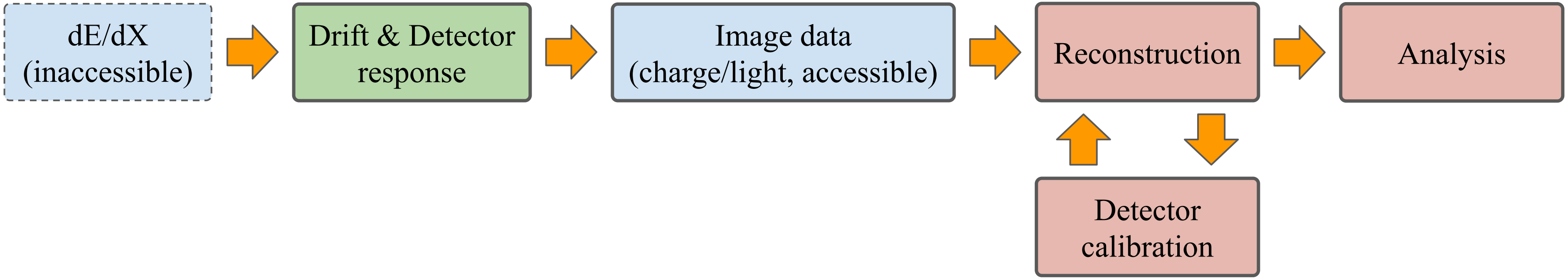


$\mathbf{Y} \in \mathcal{D}_O$

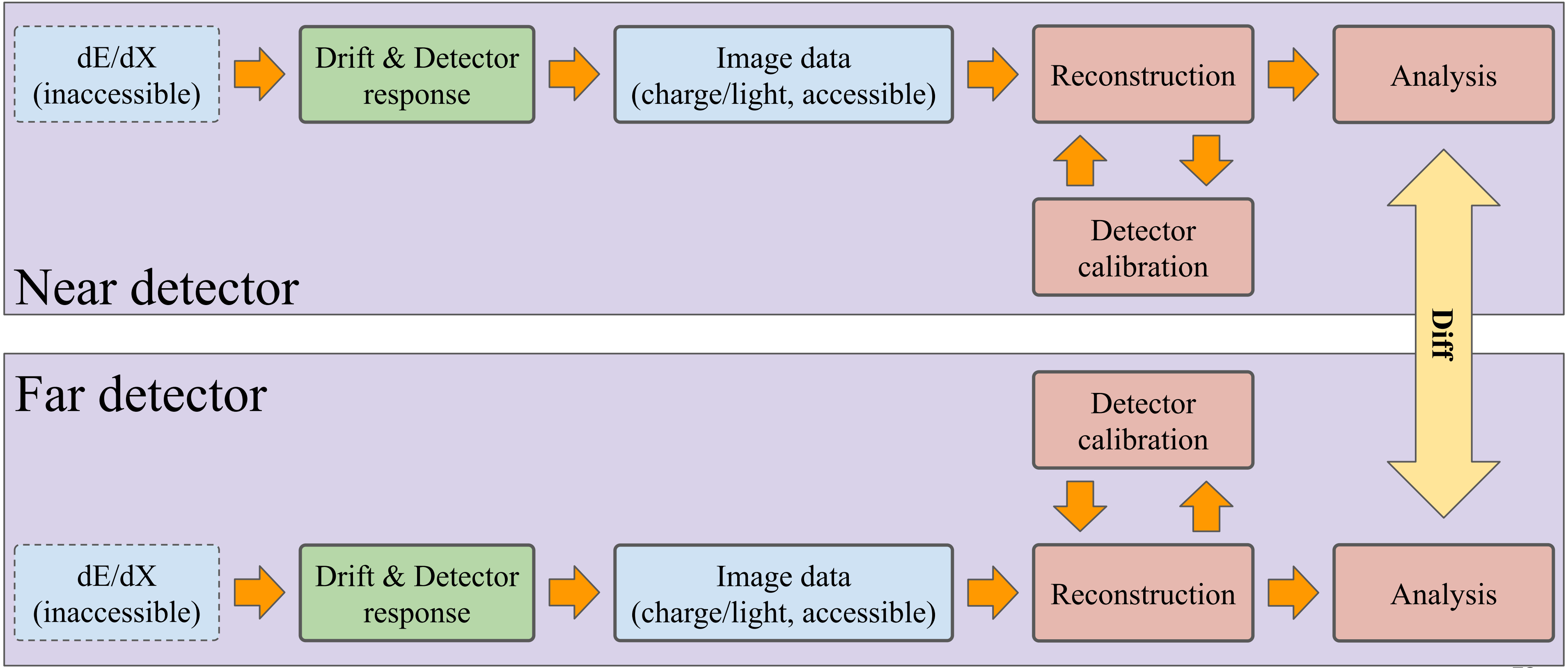
Output domain of LArTPC simulator (e.g. real data)

ML for Analyzing Big Image Data in Neutrino Experiments

Inverse imaging using a differentiable simulator

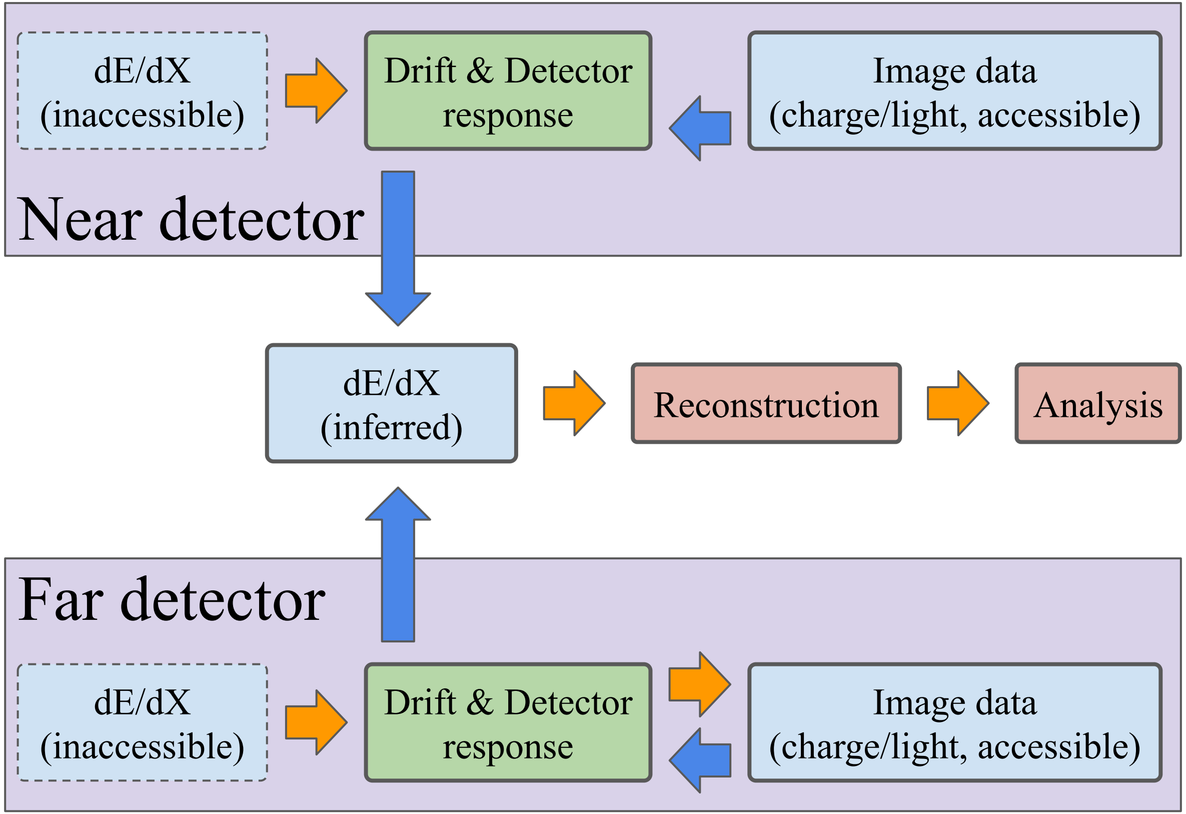


ML for Analyzing Big Image Data in Neutrino Experiments
Inverse imaging using a differentiable simulator



ML for Analyzing Big Image Data in Neutrino Experiments

Inverse imaging using a differentiable simulator

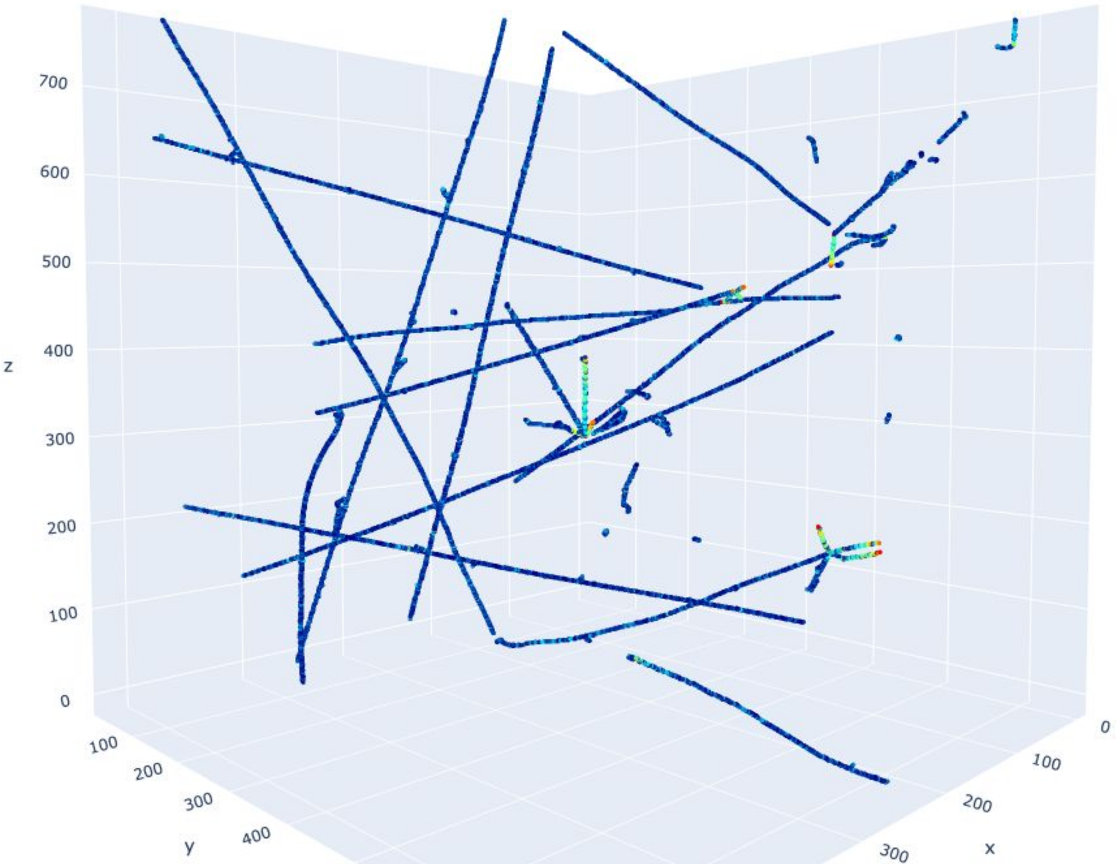


Detector calibration can be automated

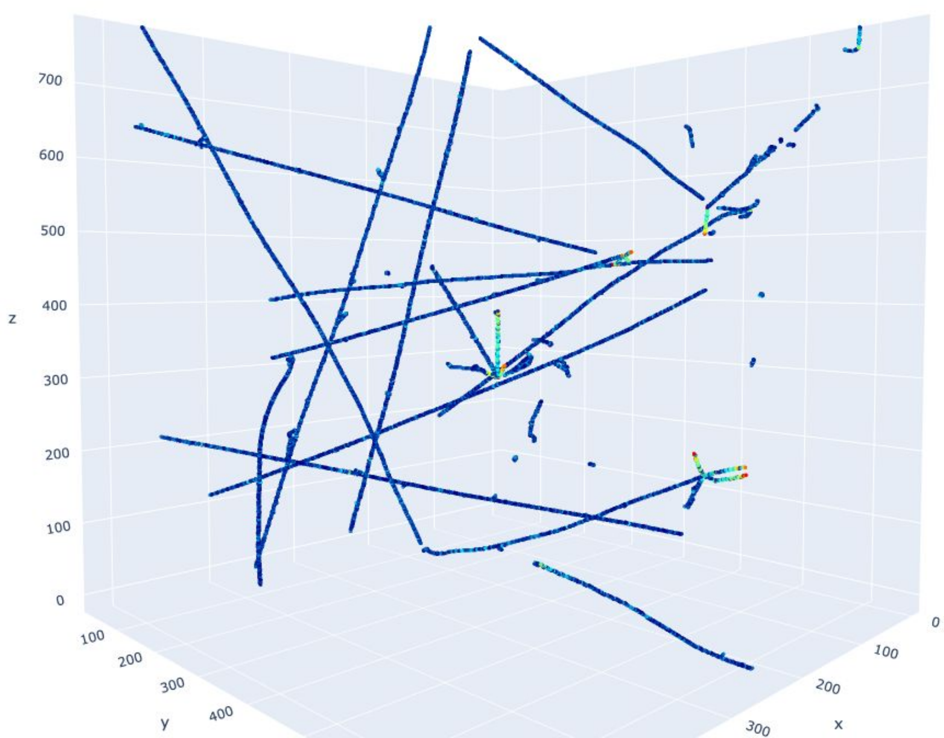
Reconstruction can be shared across detectors.

Application Details
Reconstruction Chain

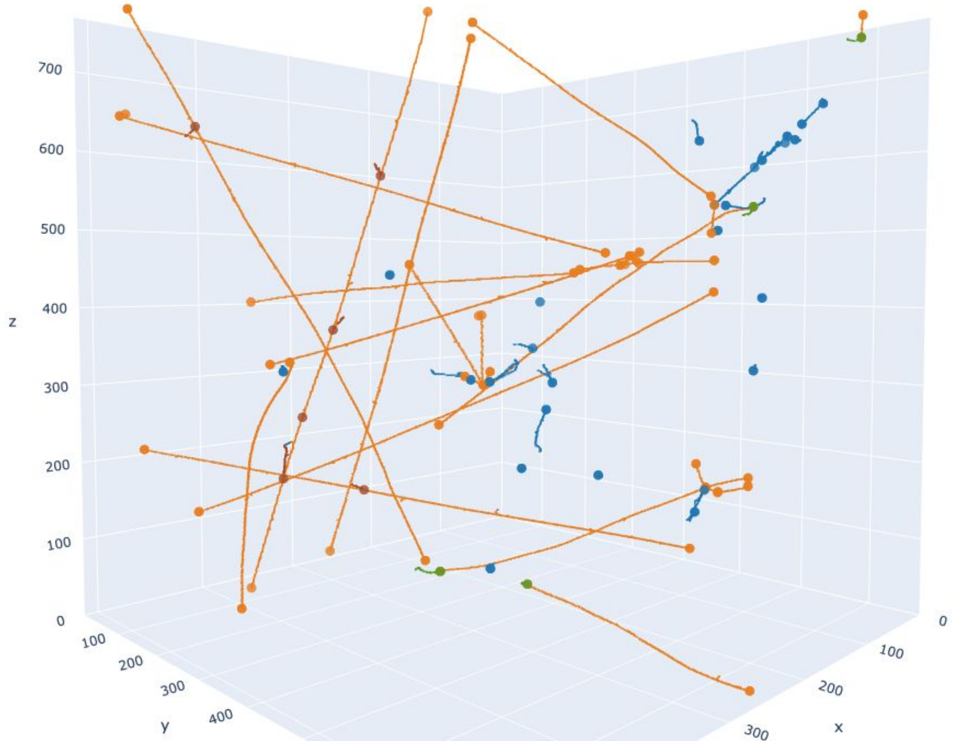
Machine Learning in Neutrino Physics & HEP
Deep Neural Network for Data Reconstruction



Stage 1

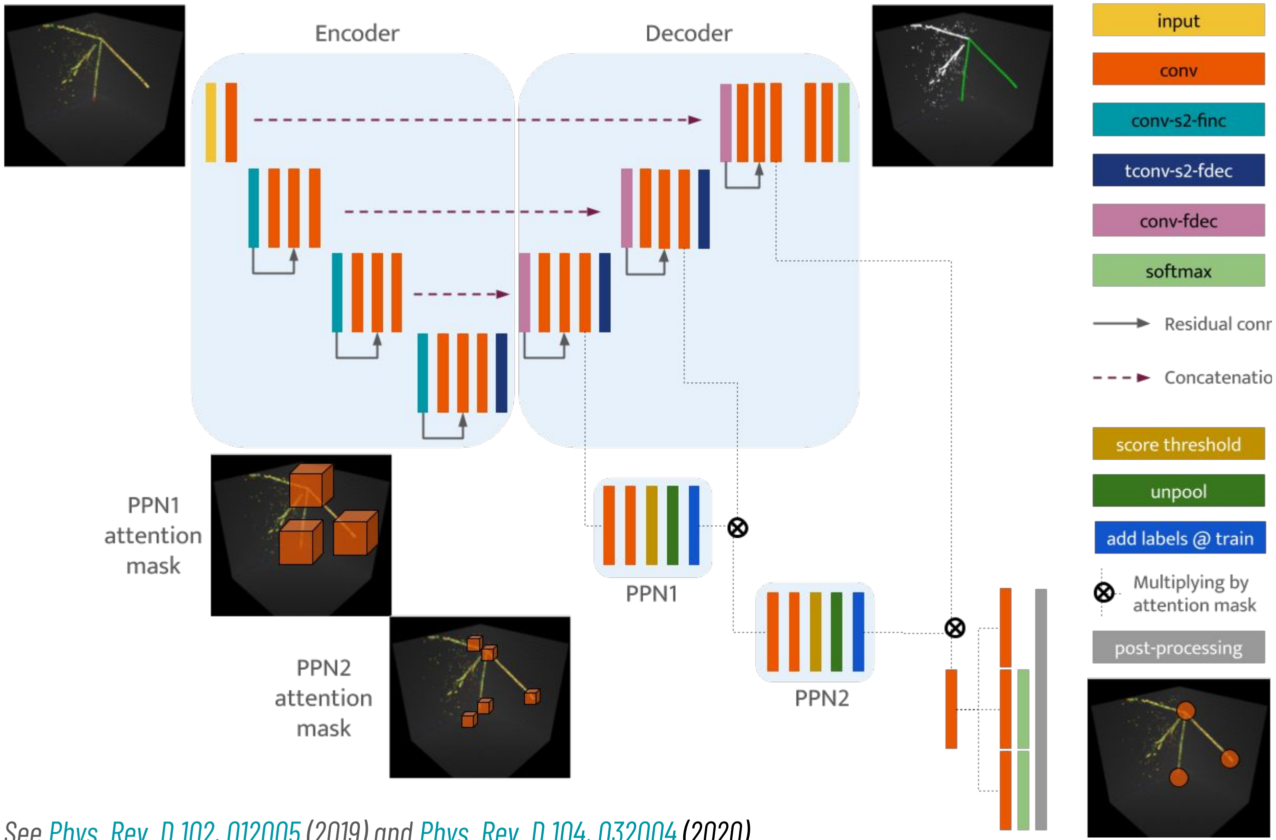


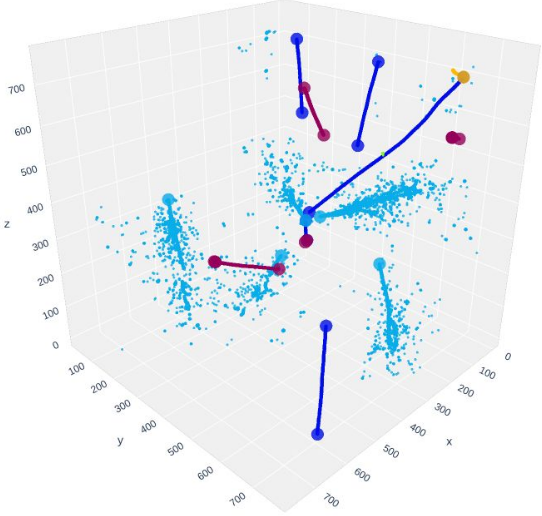
700
600
500
400
z
300
200
100
0
100
200
300
400
y
0
100
200
300
x



700
600
500
400
z
300
200
100
0
100
200
300
400
y
0
100
200
300
x

ML for Analyzing Big Image Data in Neutrino Experiments
Stage 1-b: Particle Edge-point Prediction





Semantic segmentation
(U-Net + residual conn.)

Edge point detection
(Faster R-CNN)

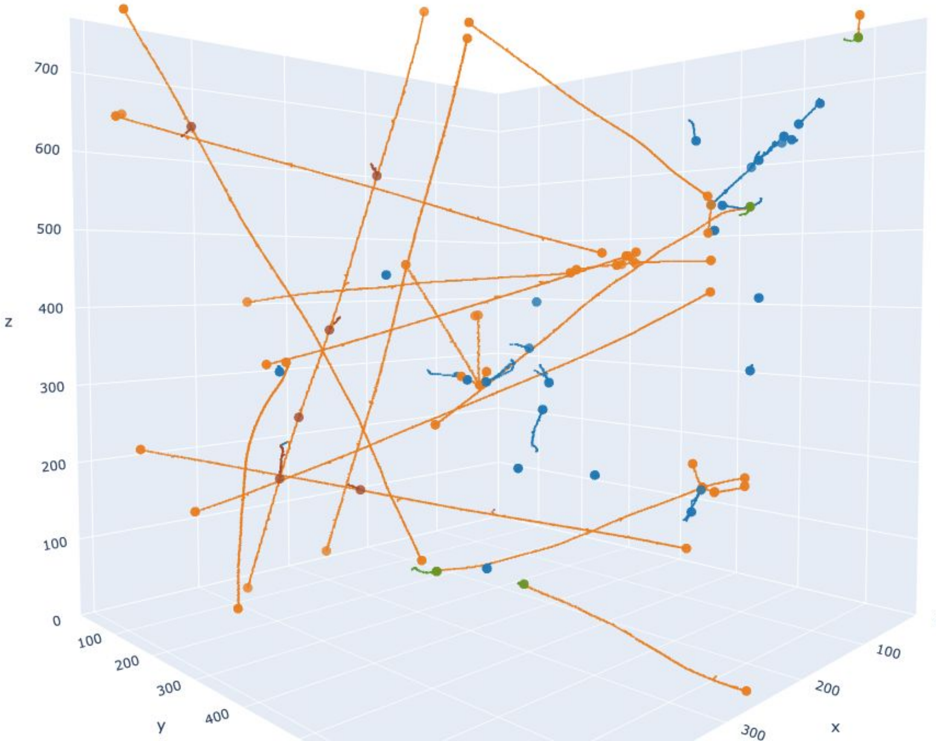
Sparse tensor operation
(Minkowski Engine)

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

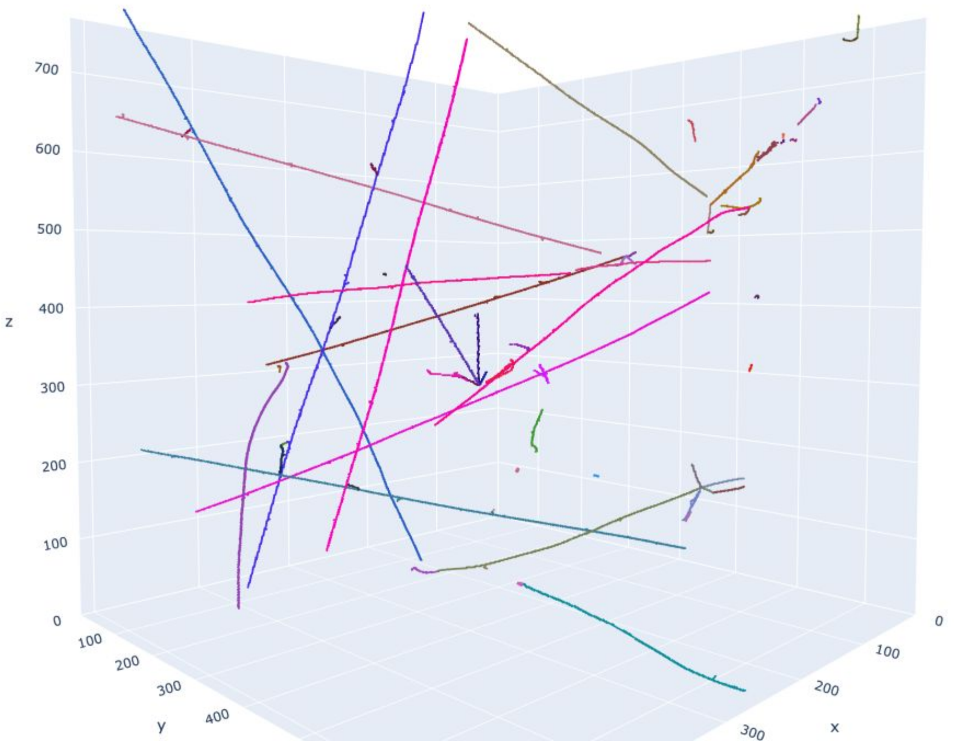
ML for Analyzing Big Image Data in Neutrino Experiments

Stage 2-a: input & output

Stage 2-a Input



Stage 2-a Output



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

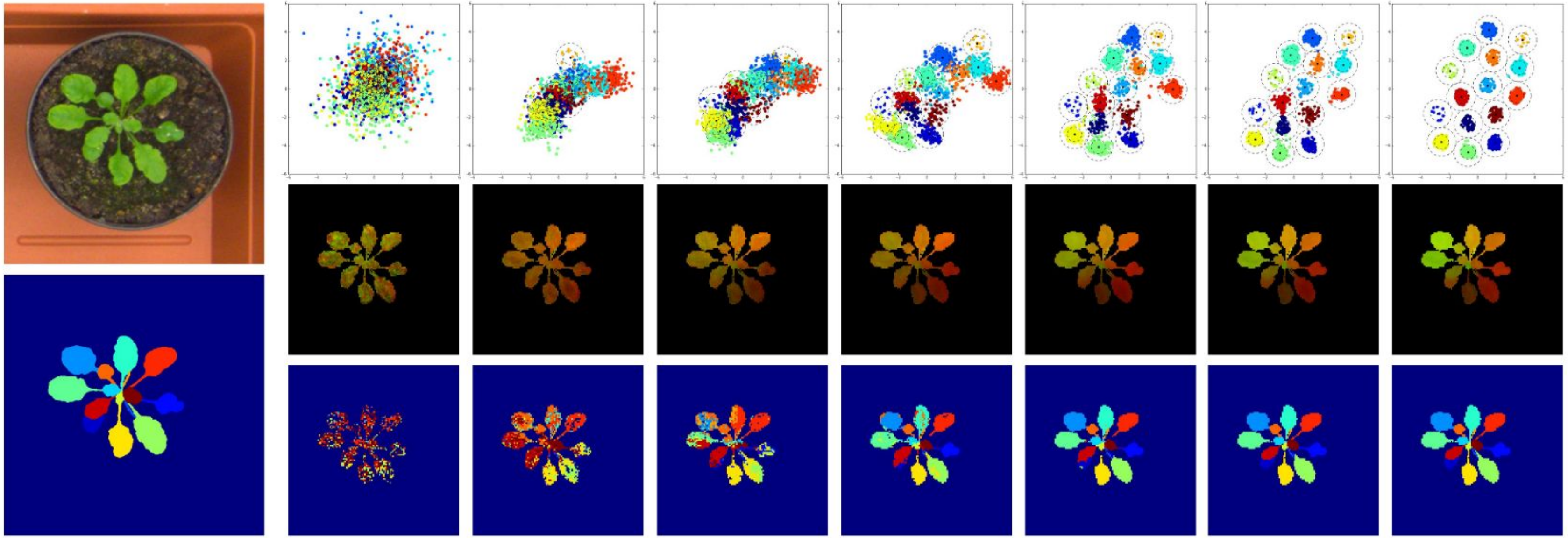
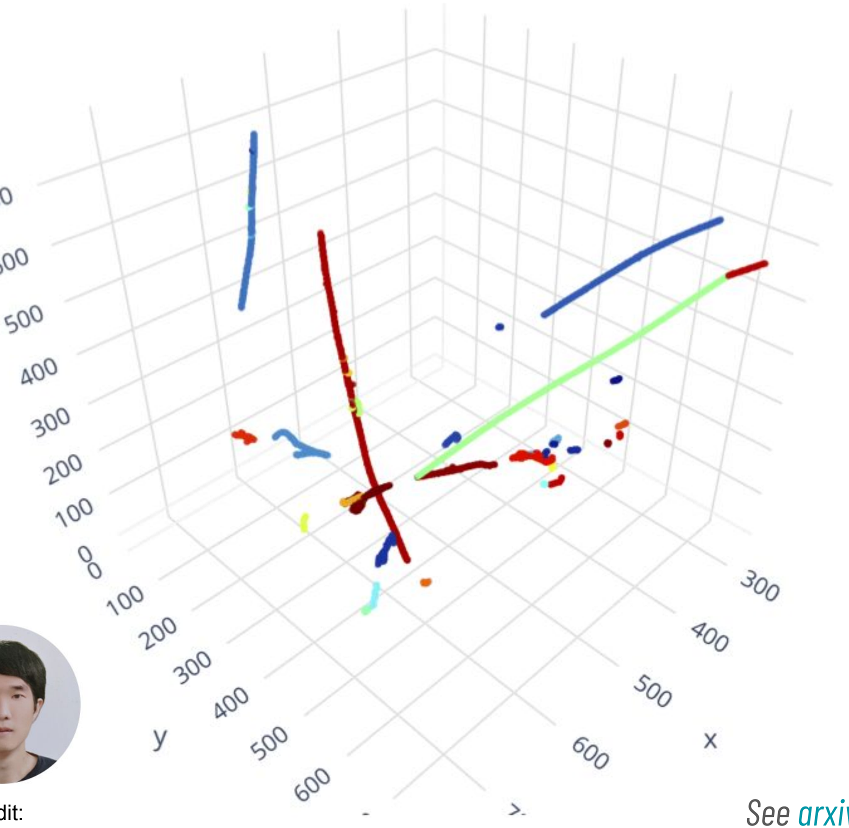
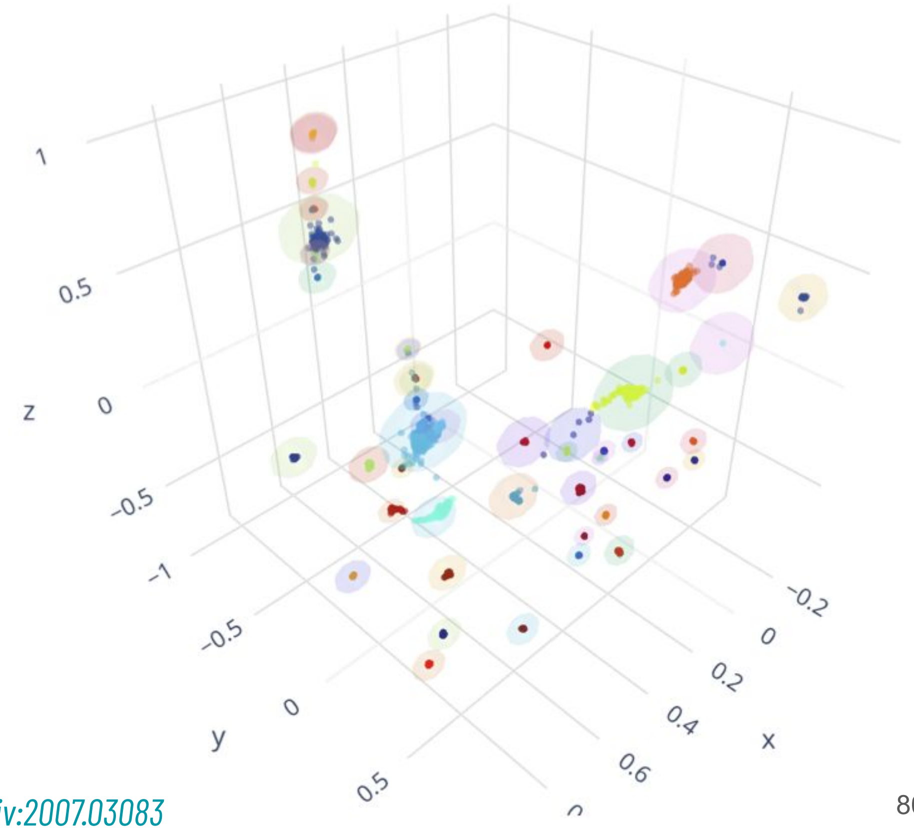


Image credit: [arXiv 1708.02551]

ML for Analyzing Big Image Data in Neutrino Experiments

Stage 2-a: Dense Pixel Clustering

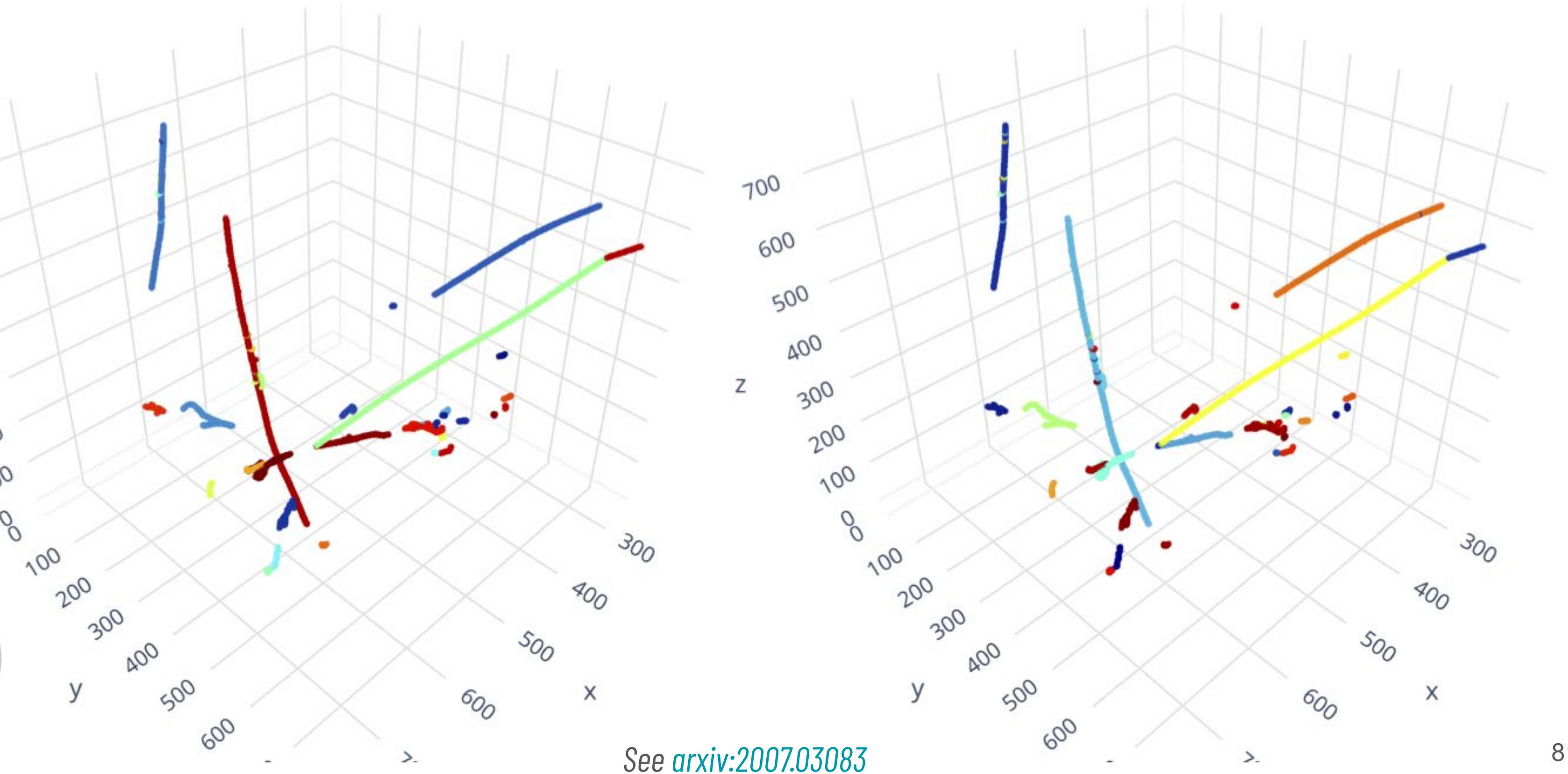




Work credit:
Dae Heun Koh (Stanford)

See arxiv:2007.03083

Stage 2-a: Dense Pixel Clustering



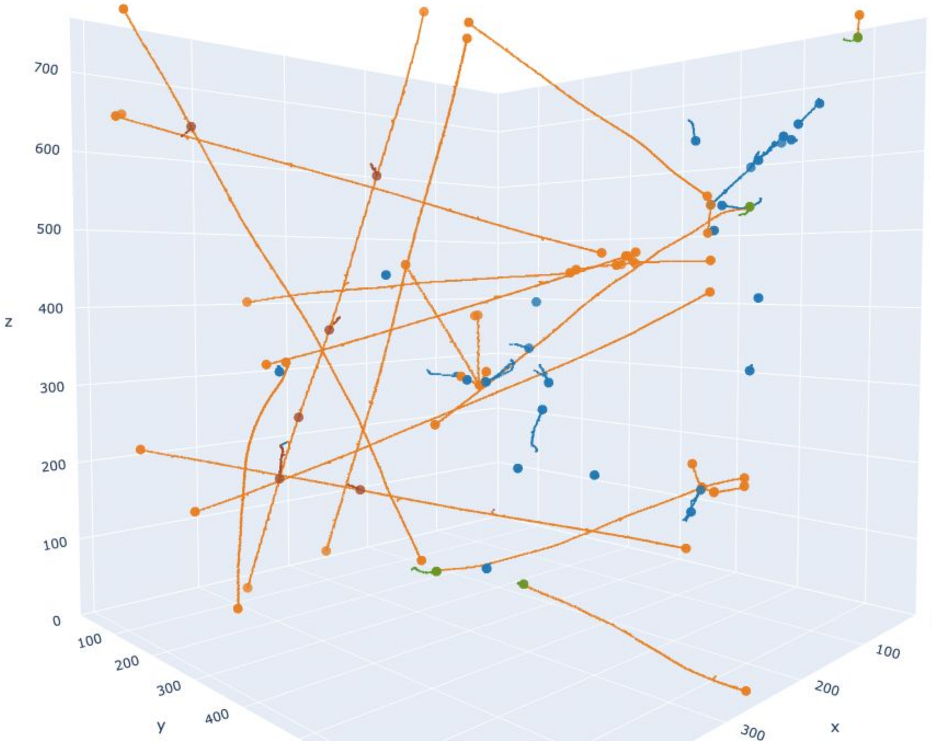
Work credit:
Dae Heun Koh (Stanford)

See arxiv:2007.03083

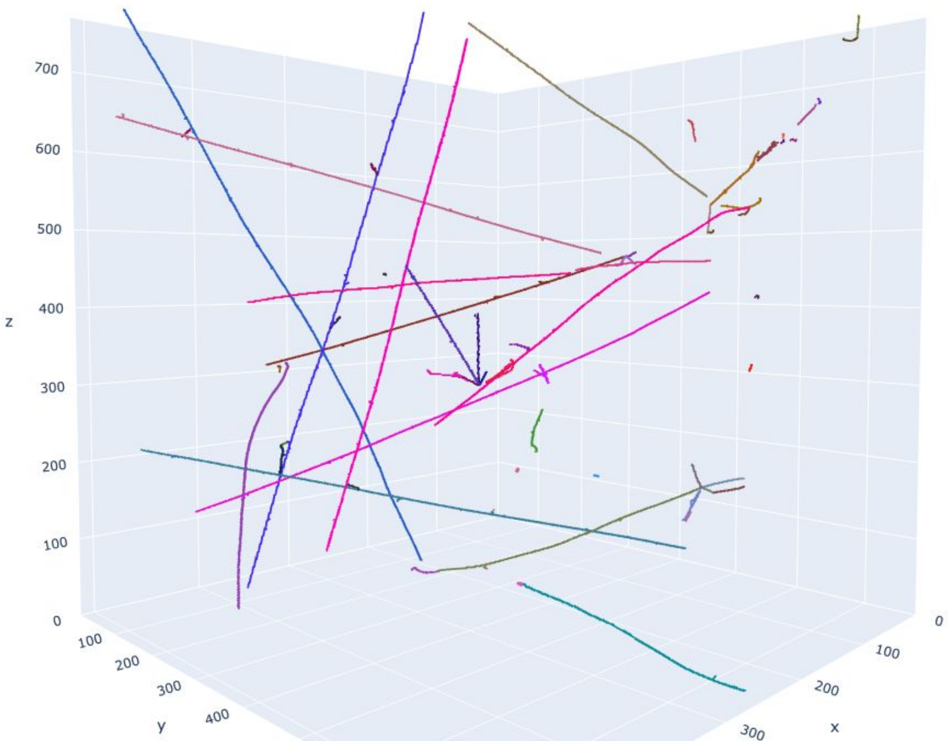
ML for Analyzing Big Image Data in Neutrino Experiments

Stage 2-a: input & output

Stage 2-a Input

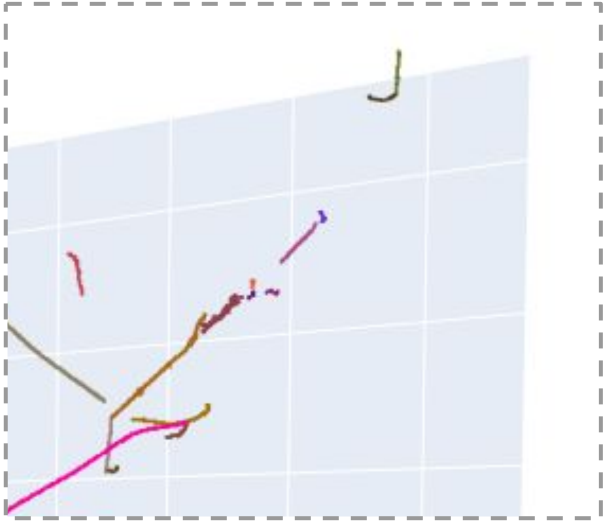


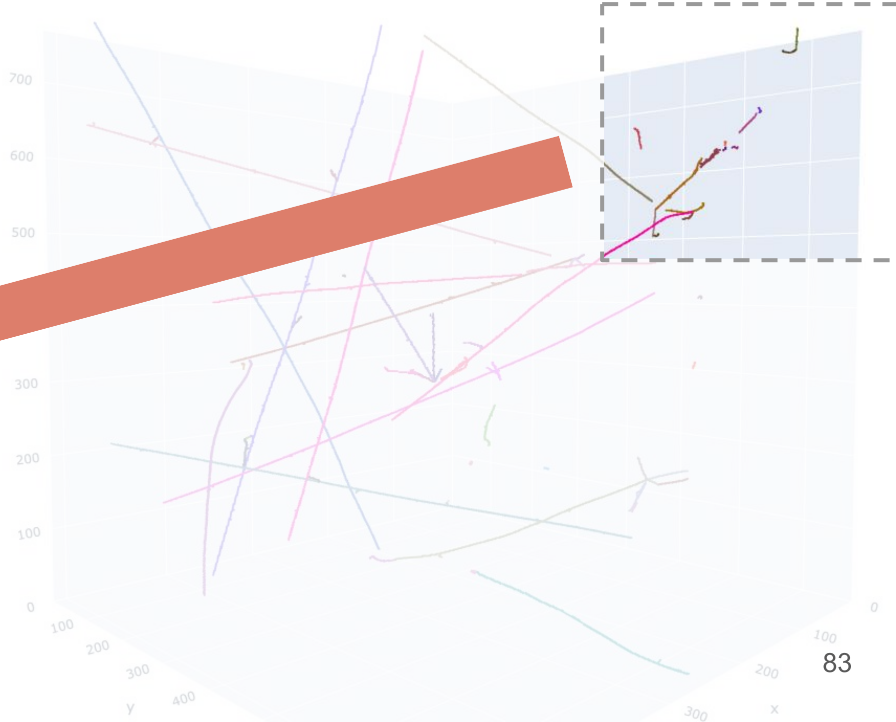
Stage 2-a Output



ML for Analyzing Big Image Data in Neutrino Experiments
Stage 2: grouping particles as a cluster

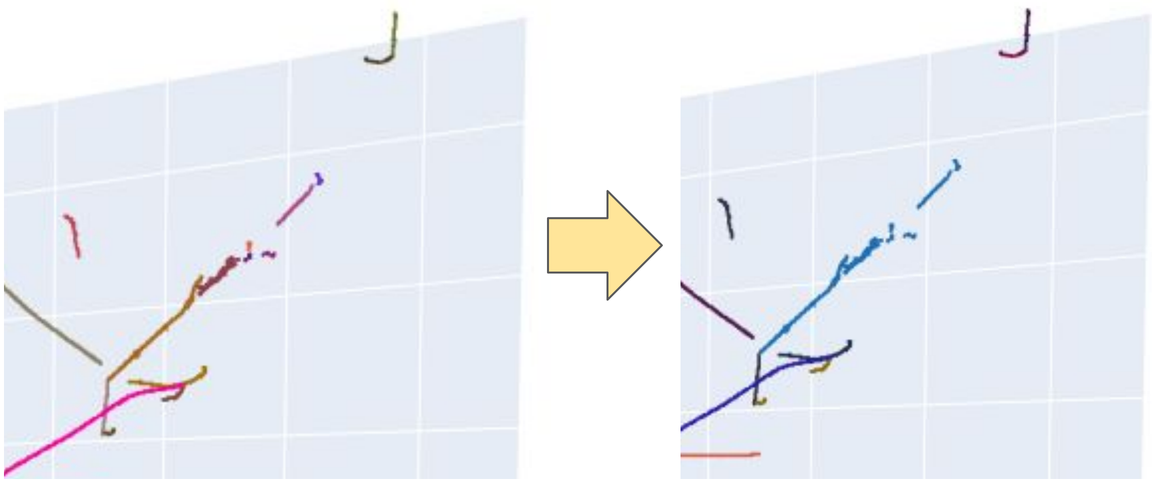
CNN for pixel-level regression dense clustering
(DeepLearnPhysics for DUNE)





ML for Analyzing Big Image Data in Neutrino Experiments
Stage 2: grouping particles as a cluster

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



Graph NN for analyzing correlations between entities which size and distance from other entities are arbitrary.

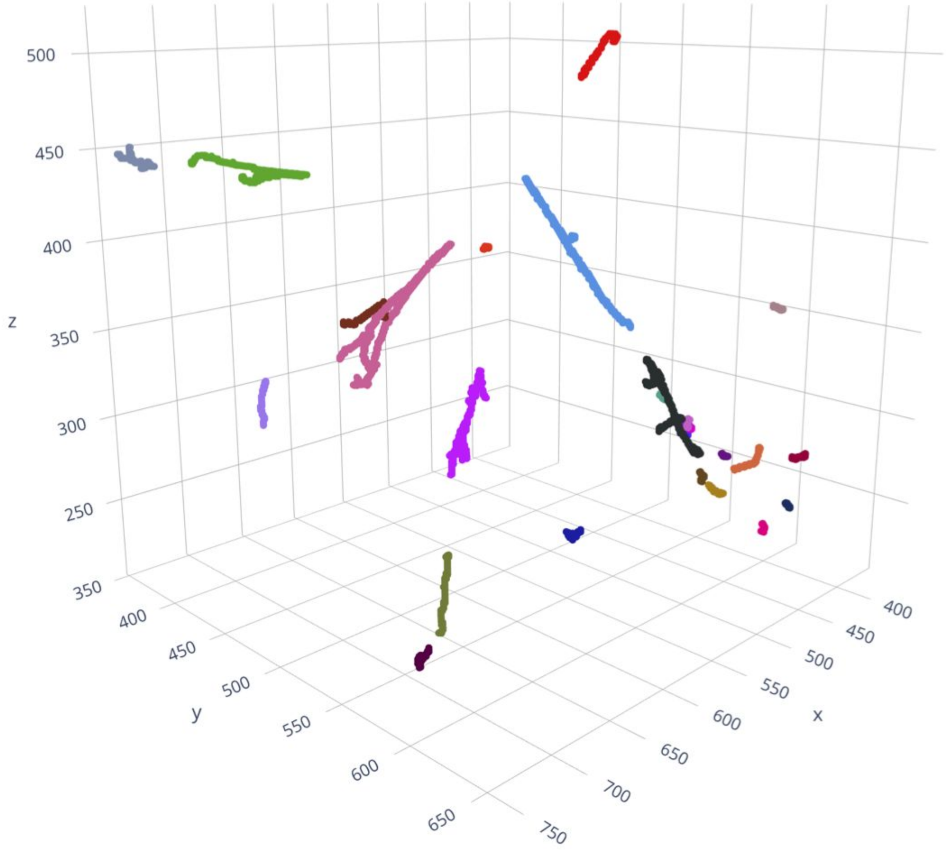
Stage 2-b: Sparse Fragment Clustering

SLAC

Graph-NN for Particle Aggregation (GrapPA)

Input:

- Fragmented EM showers



See Phys. Rev. D 104, 072004

Stage 2-b: Sparse Fragment Clustering

SLAC

Graph-NN for Particle Aggregation (GrapPA)

Input:

- Fragmented EM showers

Node features:

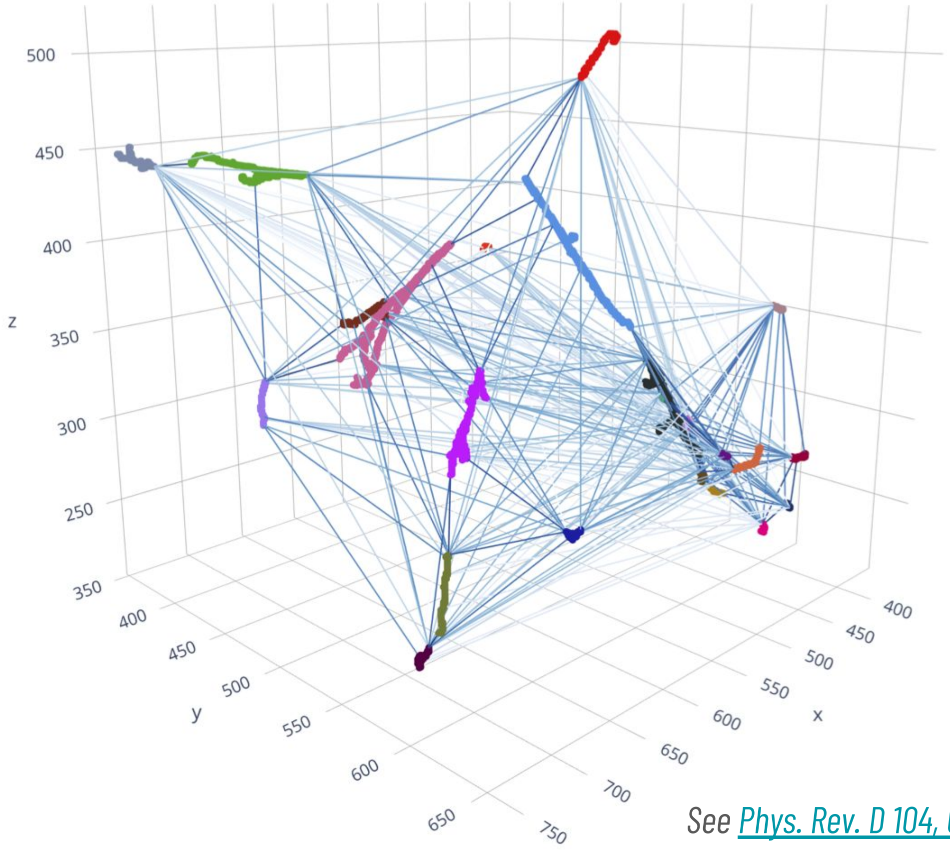
- Centroid, Covariance matrix, PCA
- Start point, direction (PPN)

Input graph:

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

Edge features:

- Displacement vector
- Closest points of approach

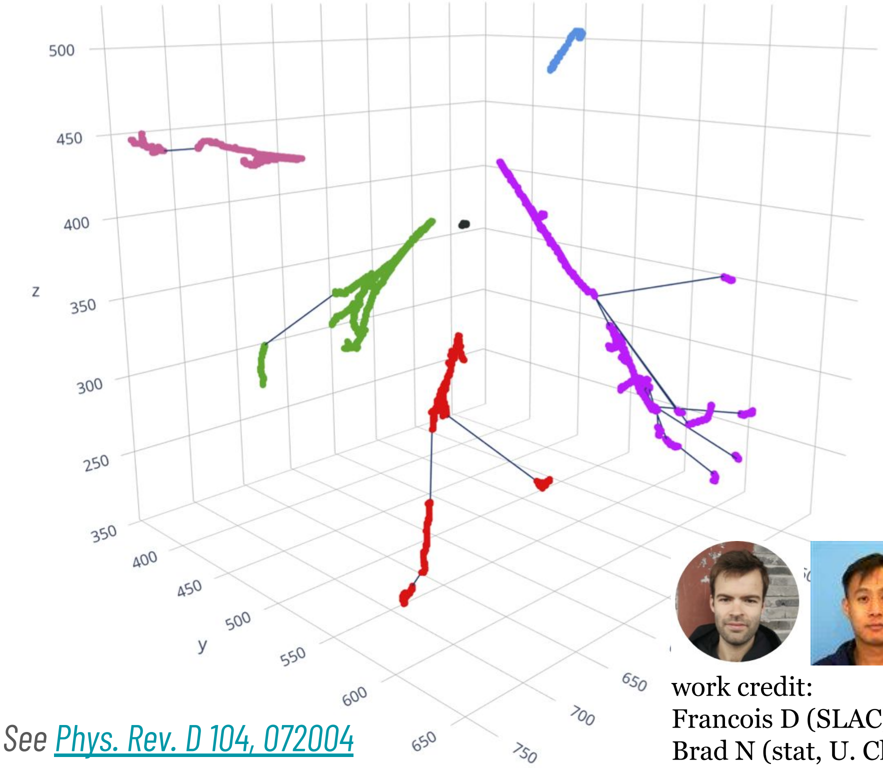


See Phys. Rev. D 104, 072004

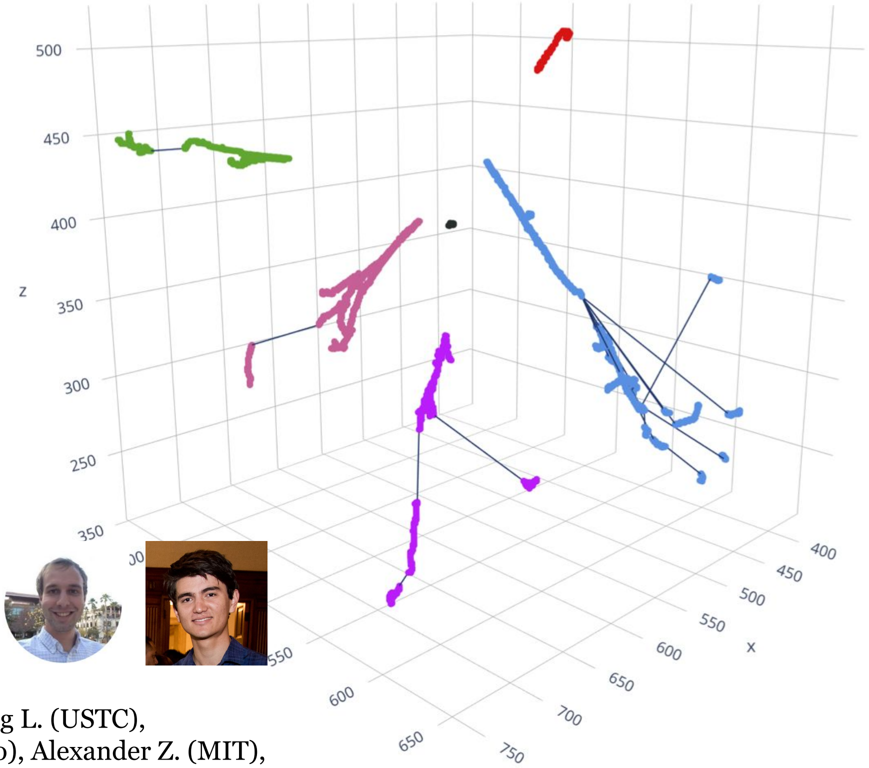
ML for Analyzing Big Image Data in Neutrino Experiments

Stage 2-b: Sparse Fragment Clustering

Target



Prediction

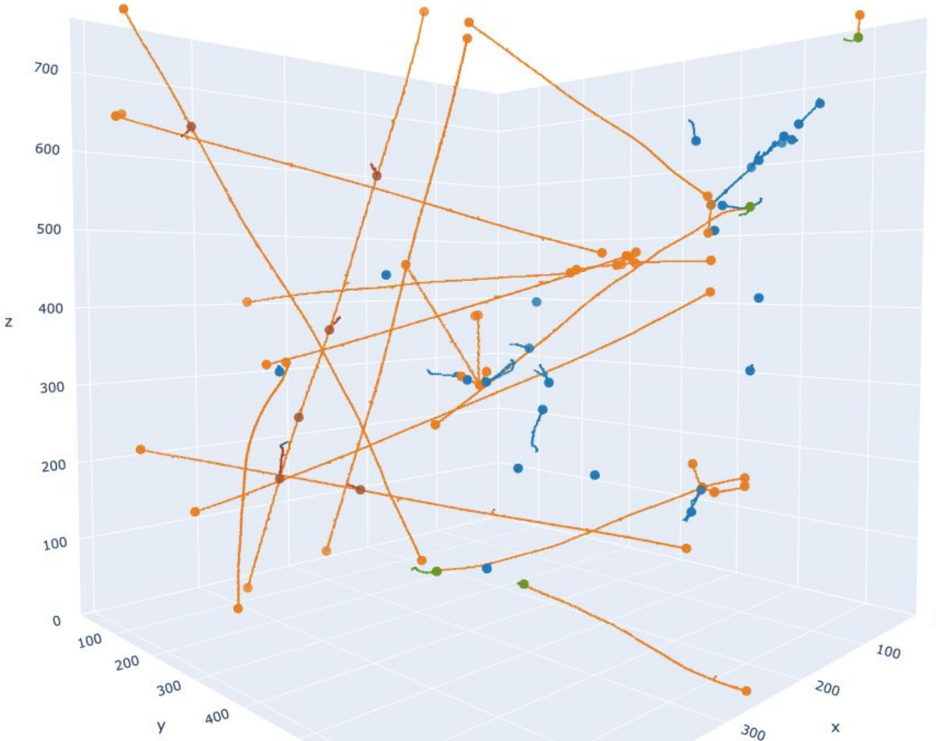


work credit:
Francois D (SLAC), Qing L. (USTC),
Brad N (stat, U. Chicago), Alexander Z. (MIT),

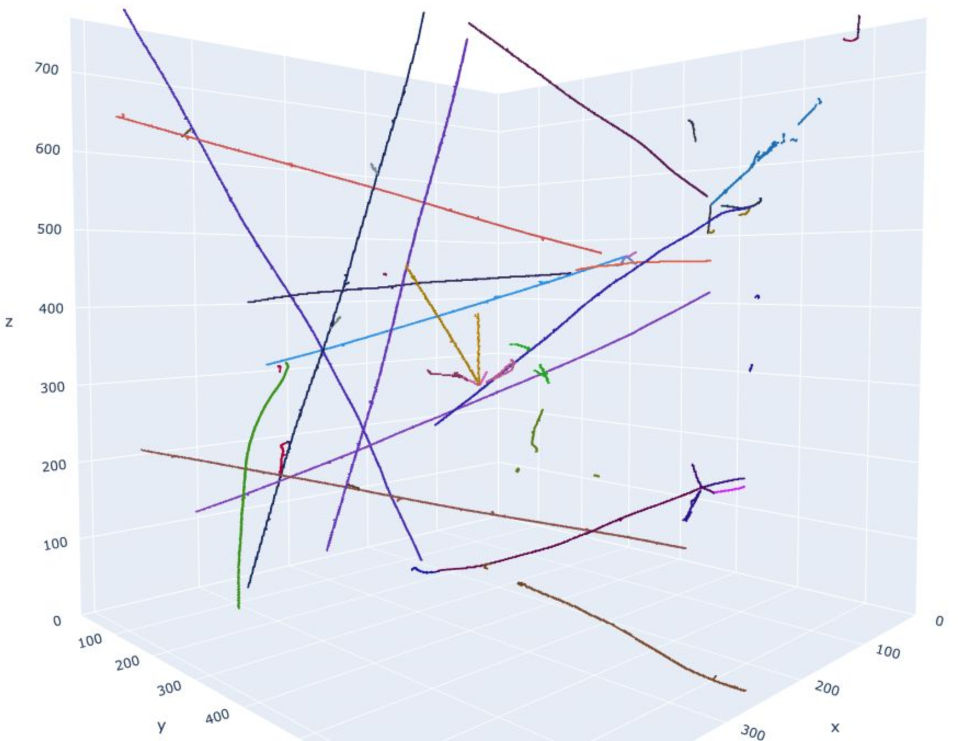
See [Phys. Rev. D 104, 072004]

ML for Analyzing Big Image Data in Neutrino Experiments
Stage 2: input & output

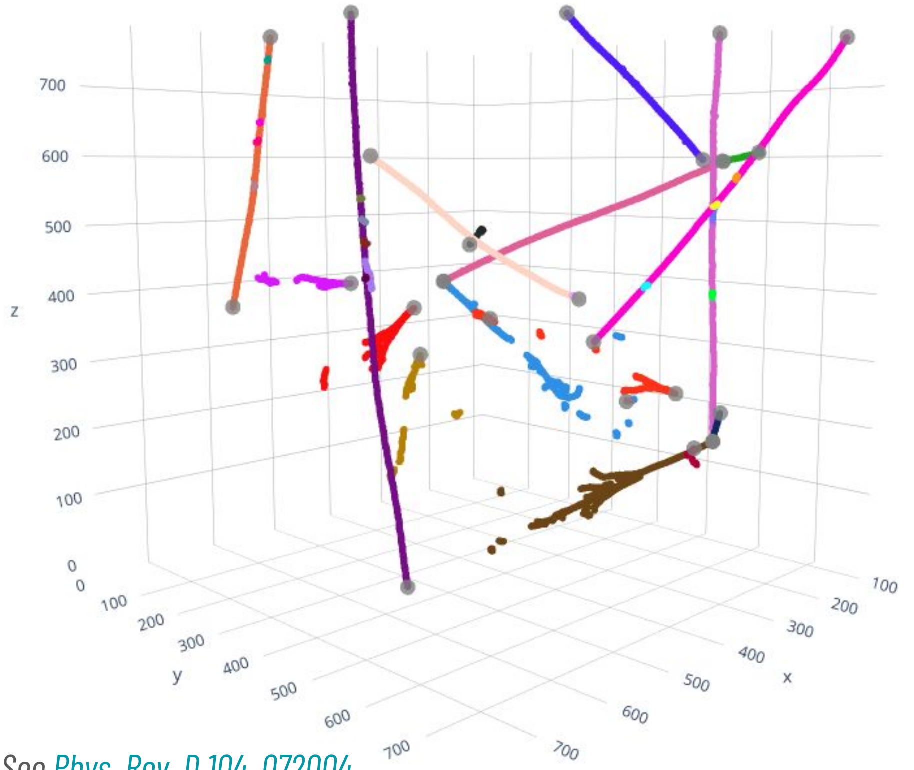
Stage 2 Input



Stage 2 Output



Stage 3: Interaction Clustering



Identifying Each Interaction?

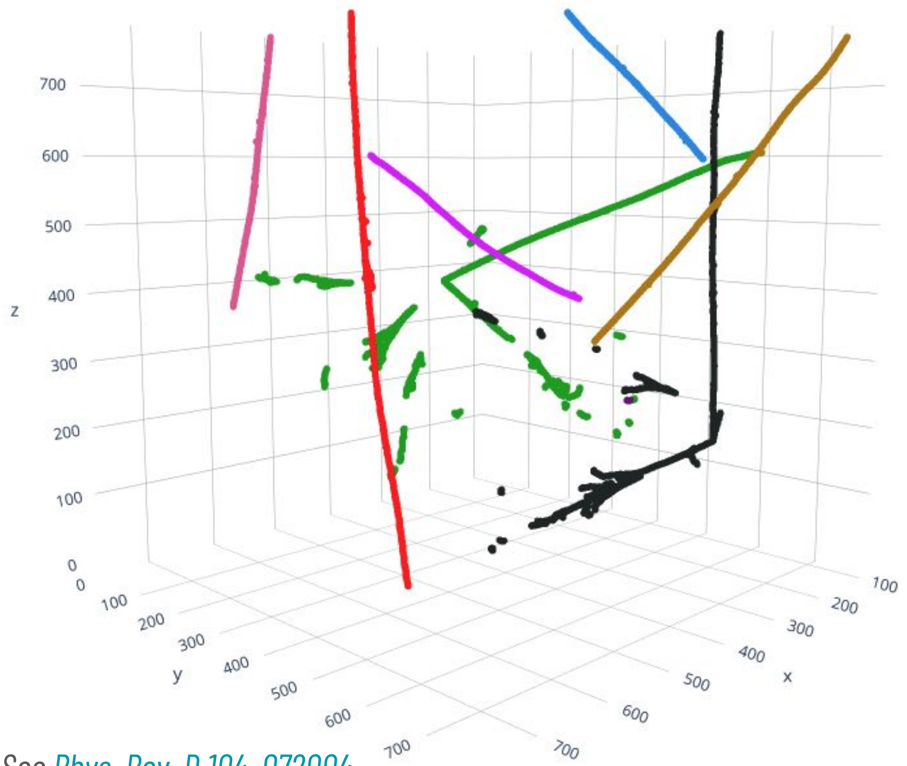
Grouping task = re-use GrapPA!

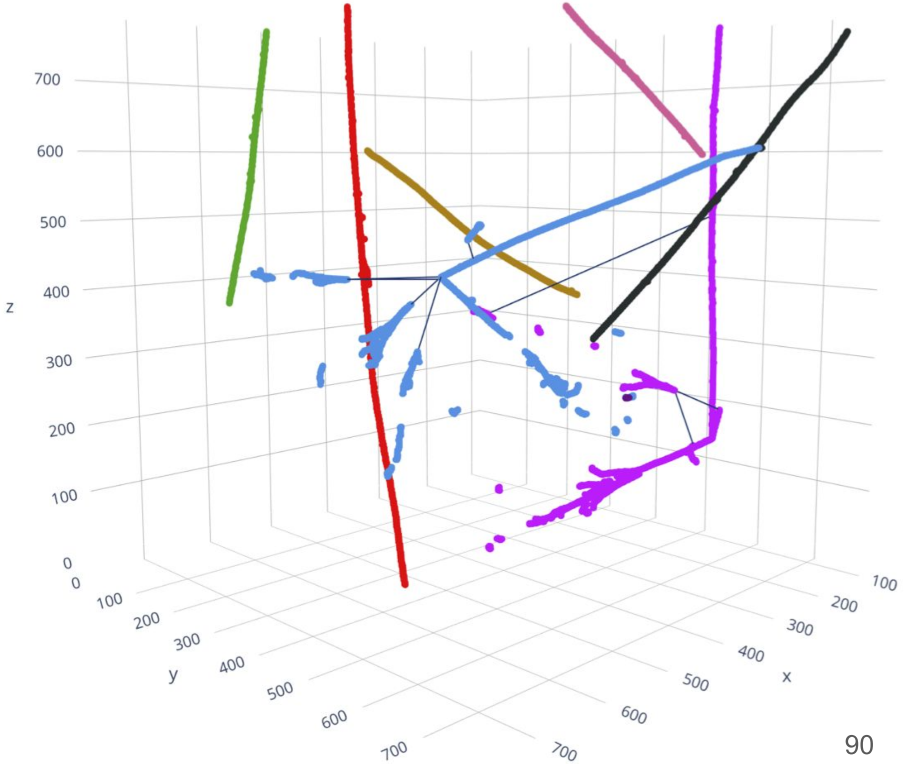
- Interaction = a group of particles that shared the same origin (i.e. neutrino interaction)
- Edge classification to identify an interaction
- Node classification for particle type ID

See Phys. Rev. D 104, 072004

ML for Analyzing Big Image Data in Neutrino Experiments

Stage 3: Interaction Clustering

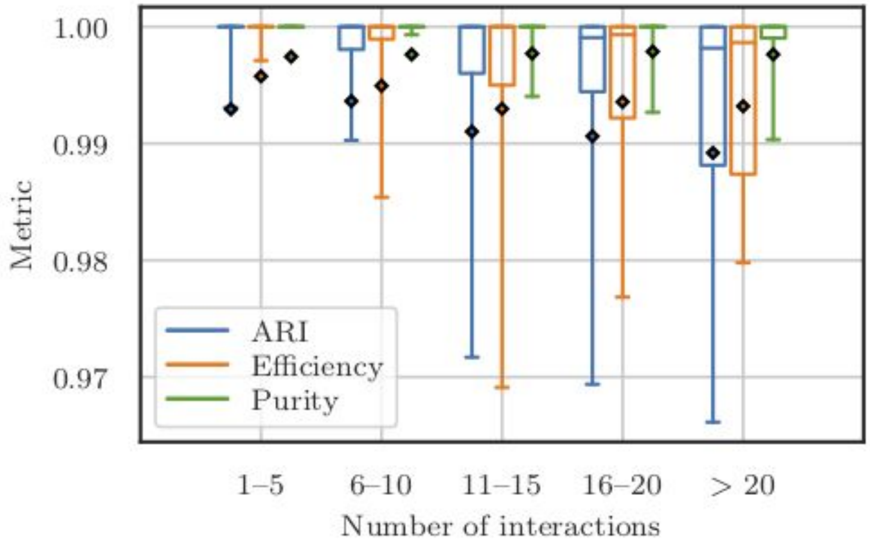




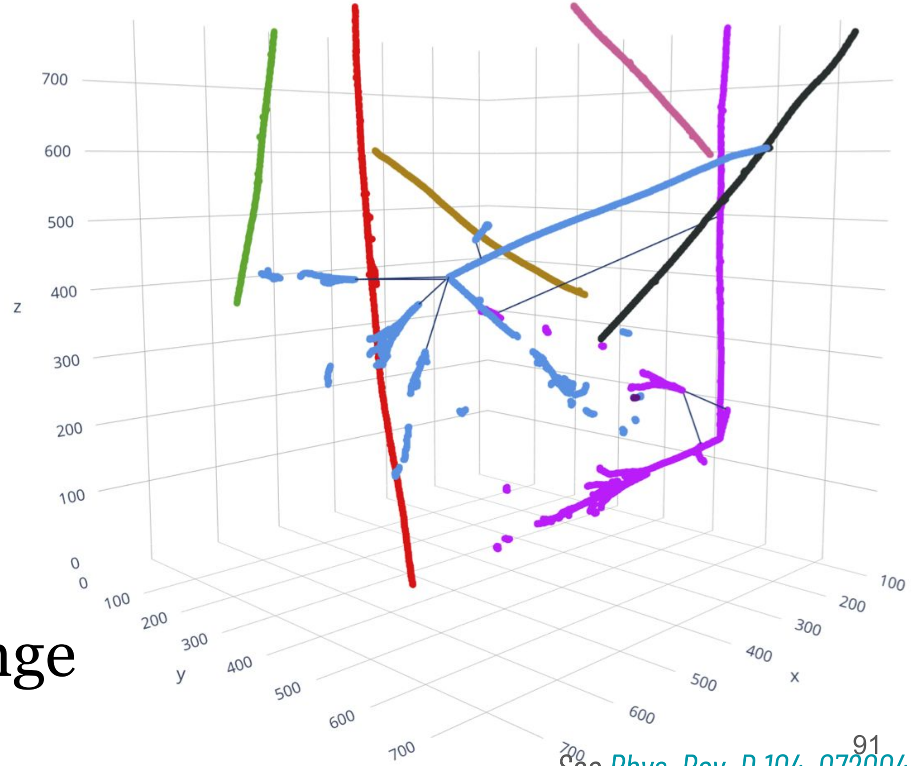
See Phys. Rev. D 104, 072004

ML for Analyzing Big Image Data in Neutrino Experiments

Stage 3: Interaction Clustering



Predicted Interaction



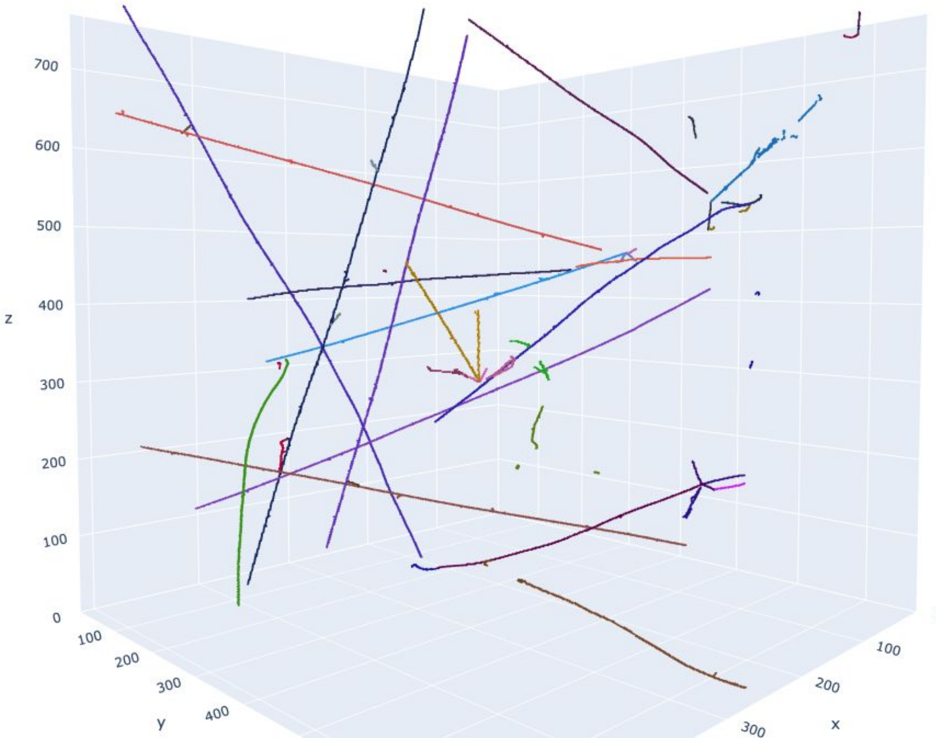
Promising result to address DUNE-ND reconstruction challenge (~20 neutrino pile-up)

See Phys. Rev. D 104, 072004

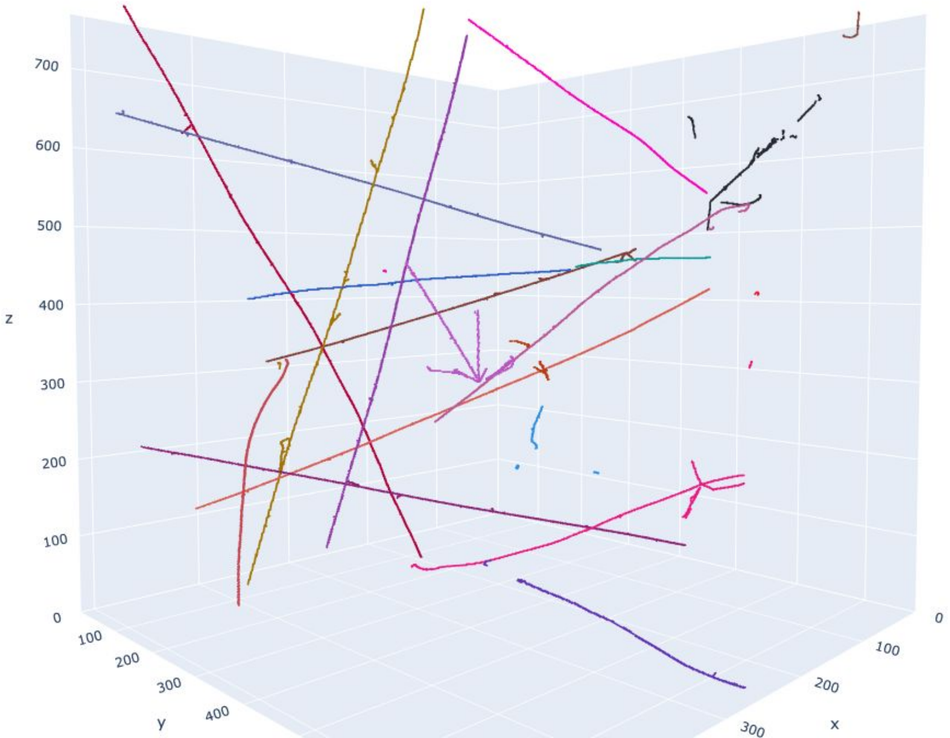
ML for Analyzing Big Image Data in Neutrino Experiments

Stage 3: input & output

Stage 3 Input

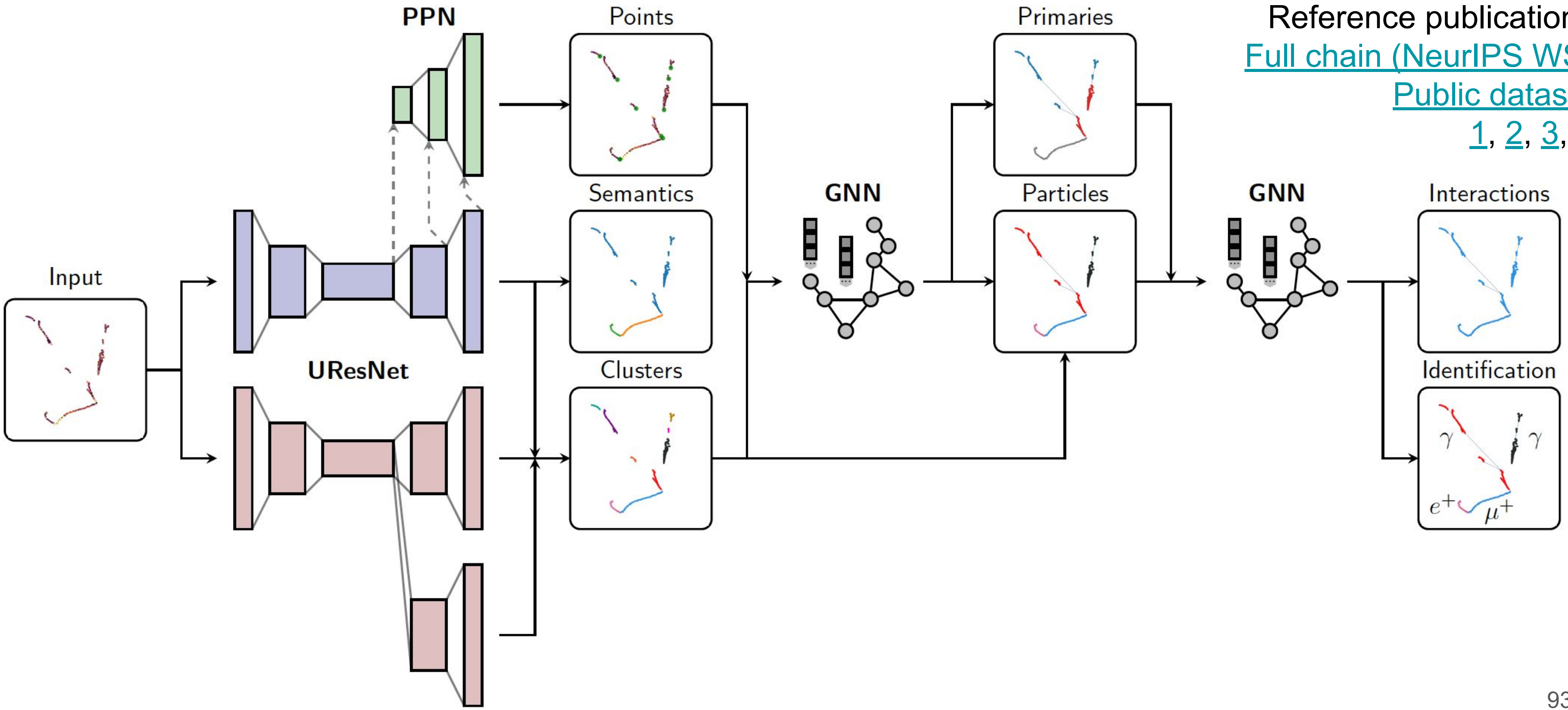


Stage 3 Output



Machine Learning in Neutrino Physics & HEP

Deep Neural Network for Data Reconstruction



Reference publications
[Full chain (NeurIPS WS)]
[Public dataset]
1, 2, 3, 4