Data Reconstruction Using Deep Neural Networks for Particle Imaging Neutrino Detectors

François Drielsma

on behalf of the SLAC ML group

drielsma@slac.stanford.edu

April 22, 2020

U.S. DEPARTMENT OF ENERGY
Office of Science

SLAC NATIONAL ACCELERATOR LABORATORY

SLAC Group

At SLAC, research supported by DoE ML grants (**K. Terao**):

- **Deep-learning-based data reconstruction chain for liquid argon time-projection chambers**
- μBOONE, pDUNE, ICARUS, ArgonCube 2x2, DUNE

Group consists of three **scientists**, three **postdocs**, three **grad students**

T. Usher
ICARUS

F. Drielsma
ICARUS

Q. Lin
ICARUS

L. Domine
ICARUS

P. Tsang
pDUNE

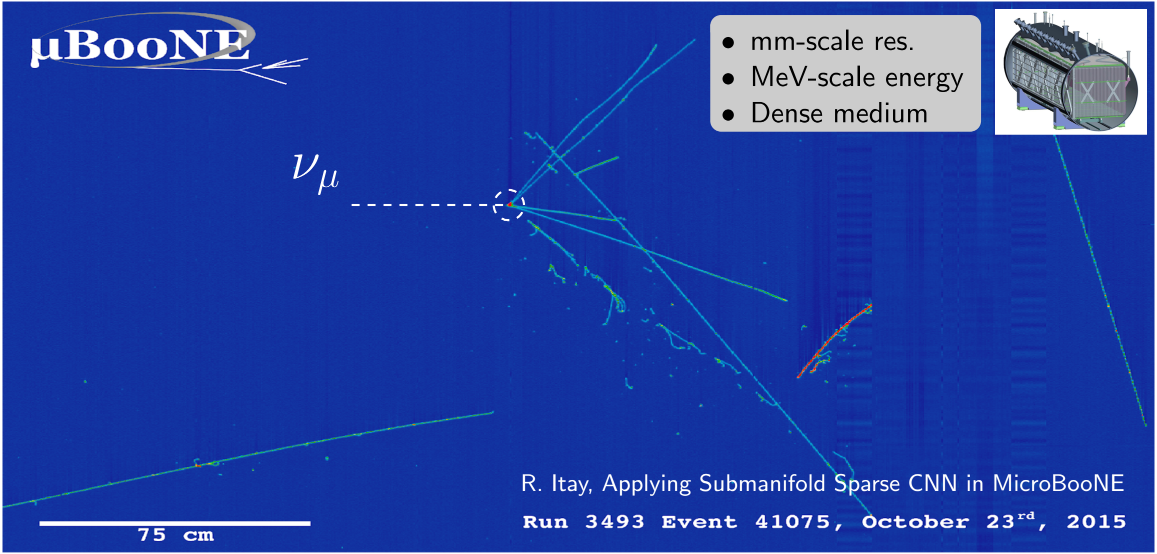
R. Itay
μBOONE

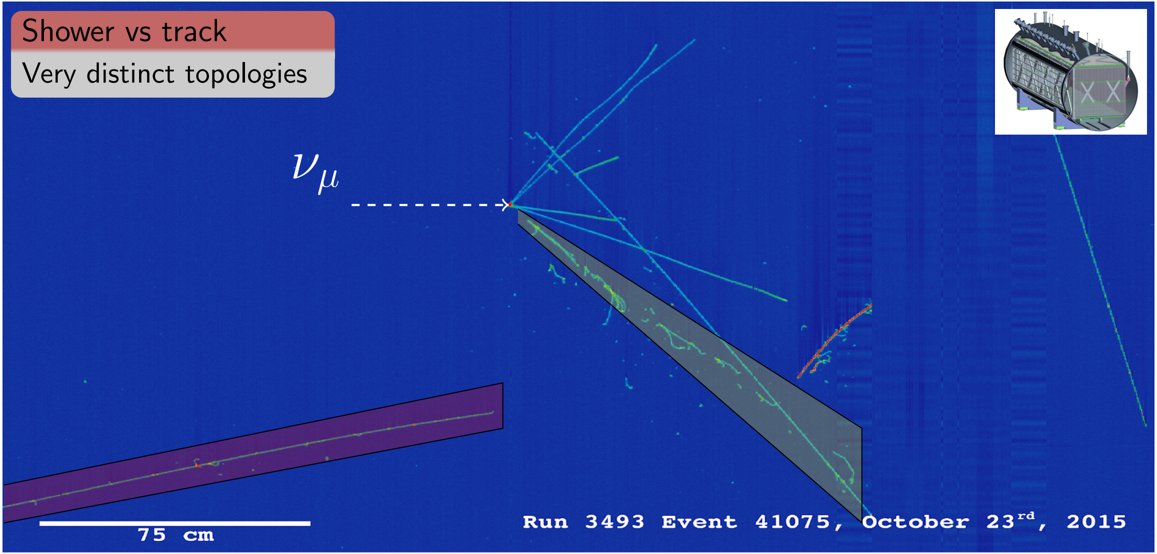
D.H. Koh
ICARUS

P. Cotes de Soux

Example 2D image

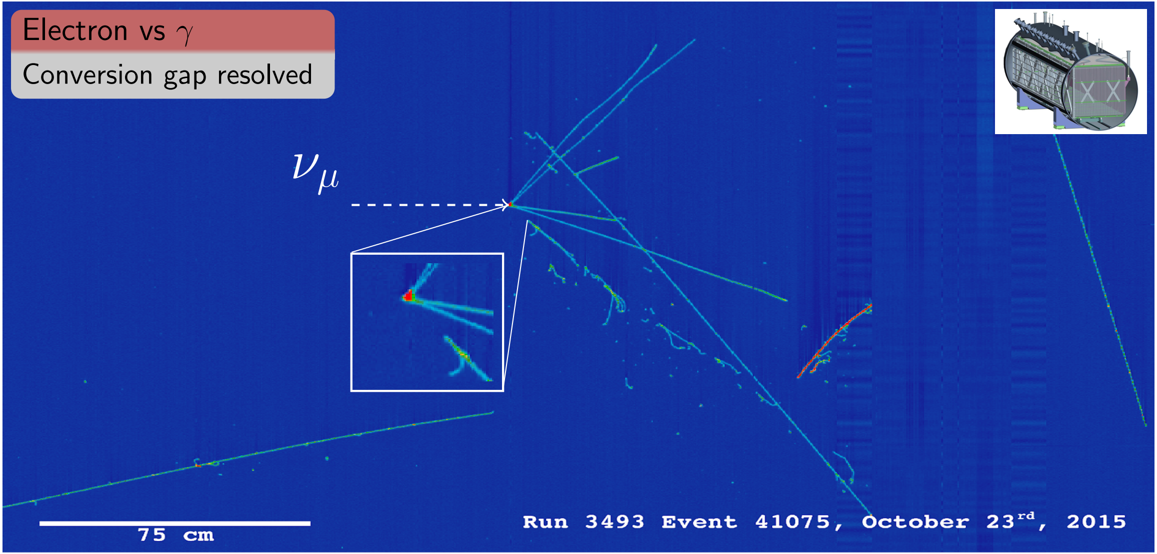
- mm-scale res.
- MeV-scale energy
- Dense medium



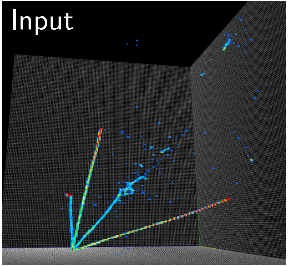

Shower vs track
Very distinct topologies
ν_μ
75 cm
Run 3493 Event 41075, October 23rd, 2015

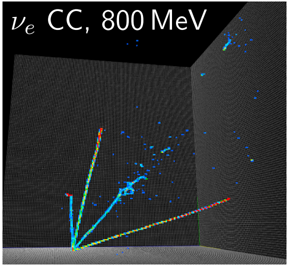
Electron vs γ

Conversion gap resolved

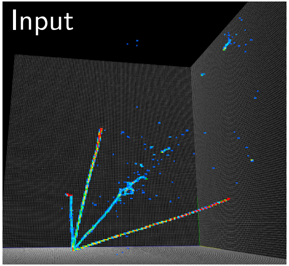


Hierarchical feature extraction | Big Picture

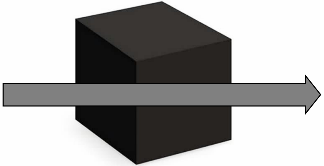


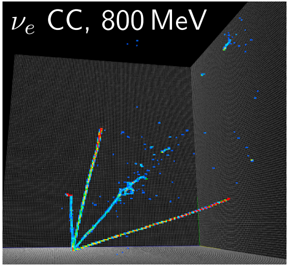


Hierarchical feature extraction | Big Picture



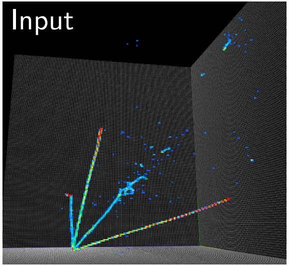
CNN ?





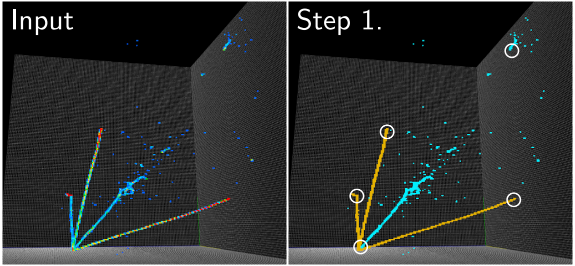
SLAC NATIONAL ACCELERATOR LABORATORY

Enforce extraction of **hierarchical physics features**



Enforce extraction of **hierarchical physics features**

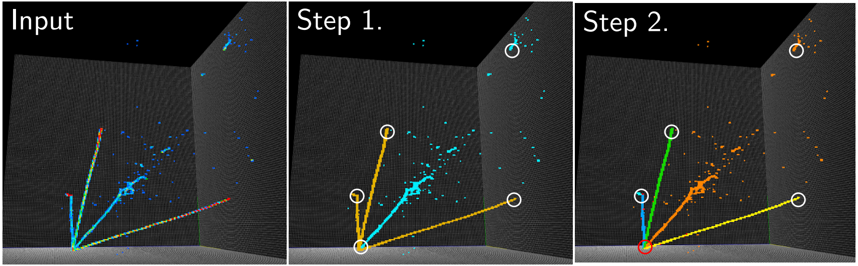
1. Pixel feature extraction + key points (particle start/end)



Hierarchical feature extraction | Big Picture

Enforce extraction of **hierarchical physics features**

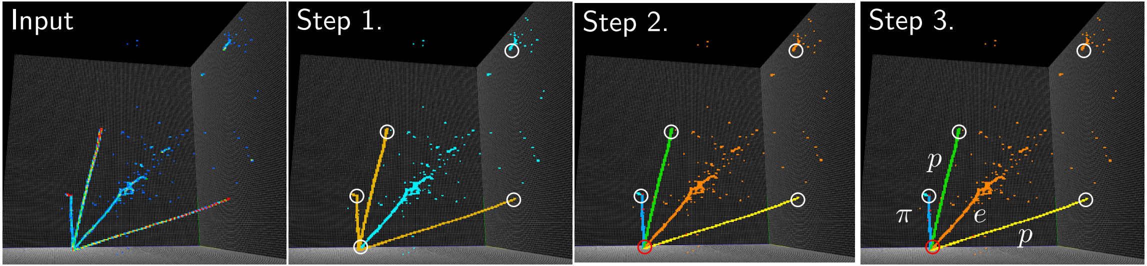
1. Pixel feature extraction + key points (particle start/end)
2. Vertex finding + particle clustering



Hierarchical feature extraction | Big Picture

Enforce extraction of **hierarchical physics features**

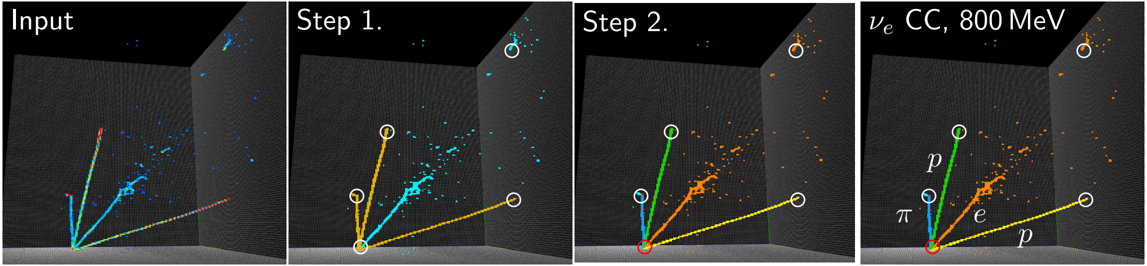
1. Pixel feature extraction + key points (particle start/end)
2. Vertex finding + particle clustering
3. Particle type + energy/momentum



Hierarchical feature extraction | Big Picture

Enforce extraction of **hierarchical physics features**

1. Pixel feature extraction + key points (particle start/end)
2. Vertex finding + particle clustering
3. Particle type + energy/momentum
4. Interaction (particle flow) reconstruction

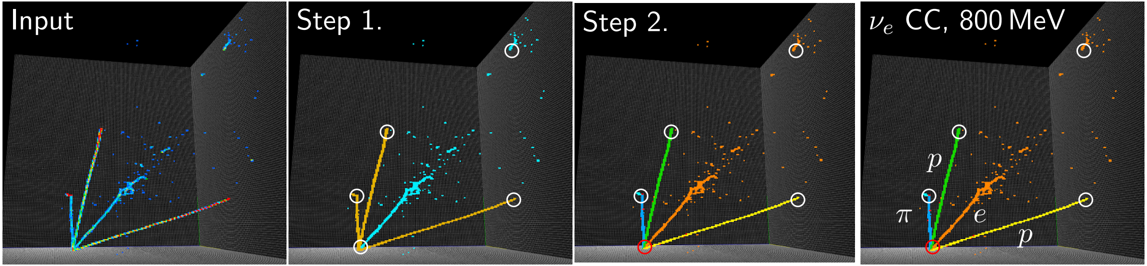


Hierarchical feature extraction | Big Picture

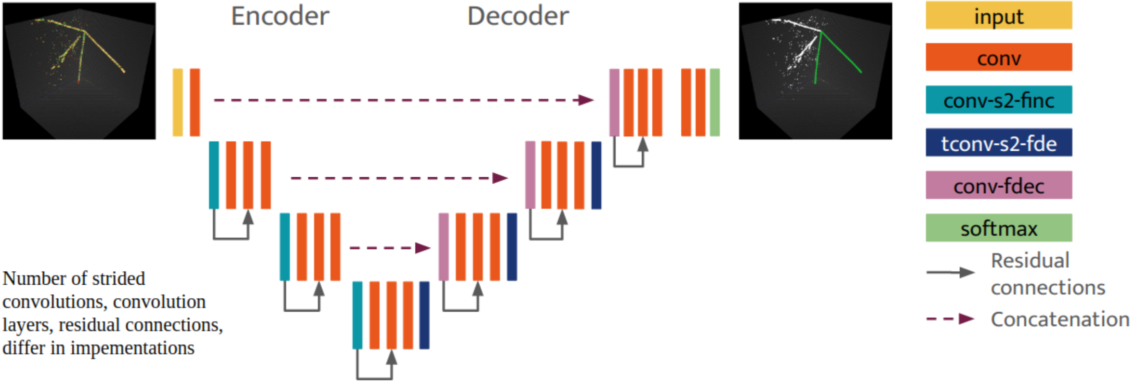
Enforce extraction of **hierarchical physics features**

1. Pixel feature extraction + key points (particle start/end)
2. Vertex finding + particle clustering
3. Particle type + energy/momentum
4. Interaction (particle flow) reconstruction

Make it for 2D/3D data + whole chain trainable



UNet + Residual connections + Sparse convolution → **Sparse UResNet**



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

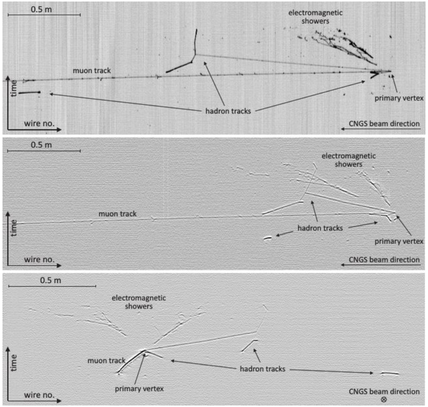
arXiv:1903.05663

L. Domine

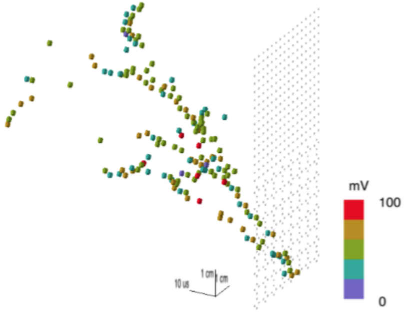
Pixel feature extraction | Input

Wire Data
Pixel Data

Image → UNet → Types / Points / Fragments → GNN → Particles → GNN → Interactions → Physics

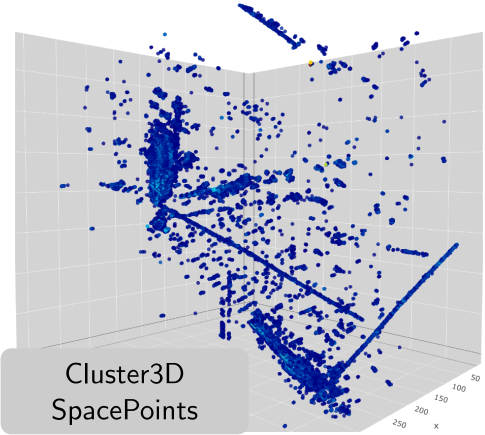


ICARUS, arXiv:1210.5089



LArPix, arXiv:1808.02969

Pixel feature extraction | Space points



Algorithms to reconstruct 3D images from 2D projections (tomography) is hard with only three projections.

Use an algorithm designed for high efficiency, relies on downstream space point solver

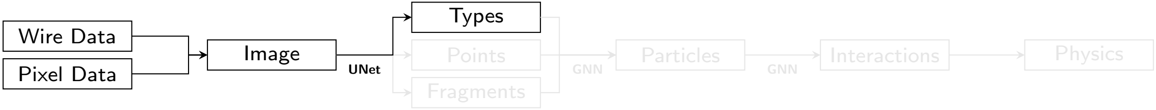
- Traditional likelihood-based
- **Semantic segmentation to discriminate against "ghost" points**

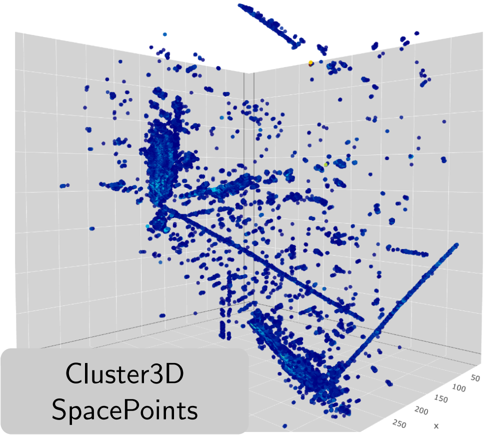
← **ICARUS simulation** on $2.3^3\ \mathrm{m}^3$ region

T. Usher, P. Tsang, L. Domine   

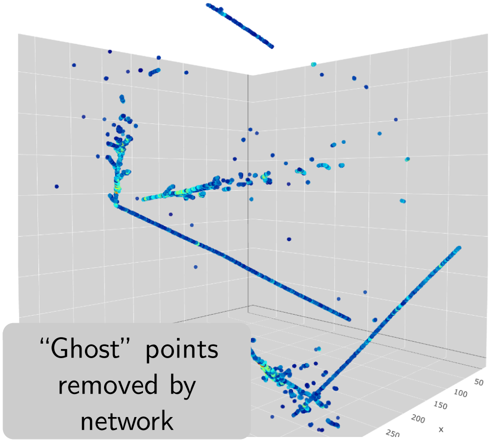
Pixel feature extraction | Deghosting

SLAC NATIONAL ACCELERATOR LABORATORY



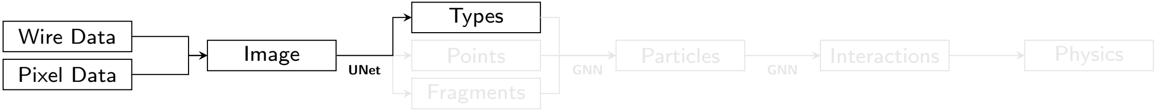


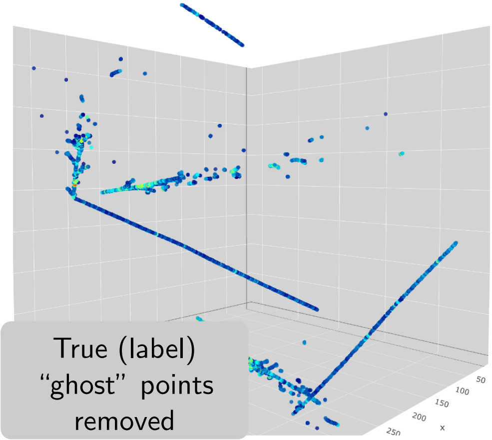
Cluster3D SpacePoints



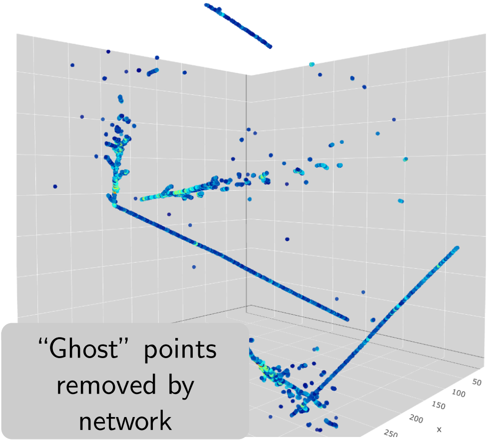
“Ghost” points removed by network

Pixel feature extraction | Deghosting



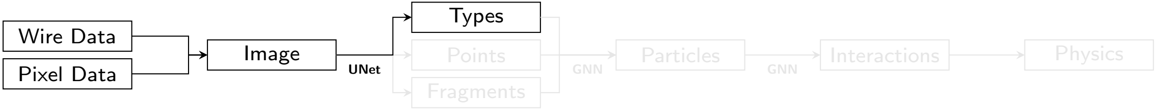


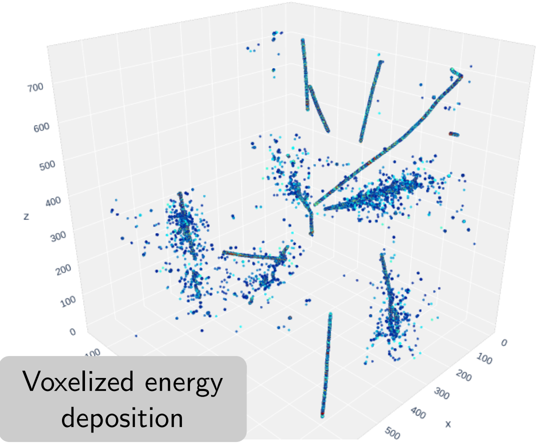
True (label) "ghost" points removed



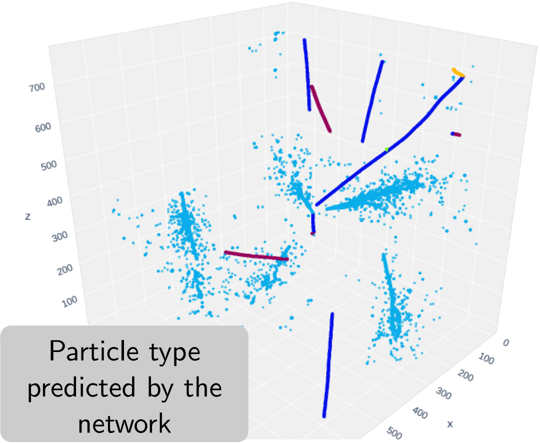
"Ghost" points removed by network

Pixel feature extraction | Particle type

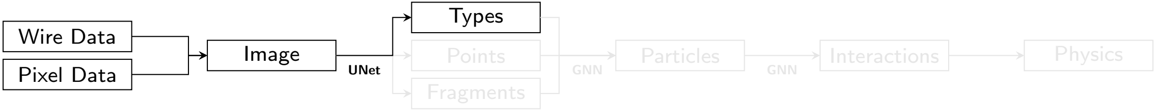




Voxelized energy deposition



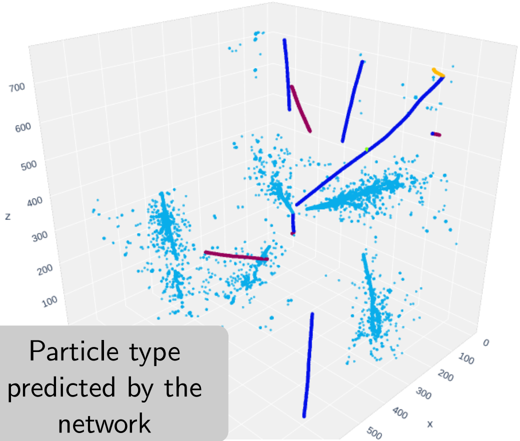
Particle type predicted by the network



Particle type identification accuracy:

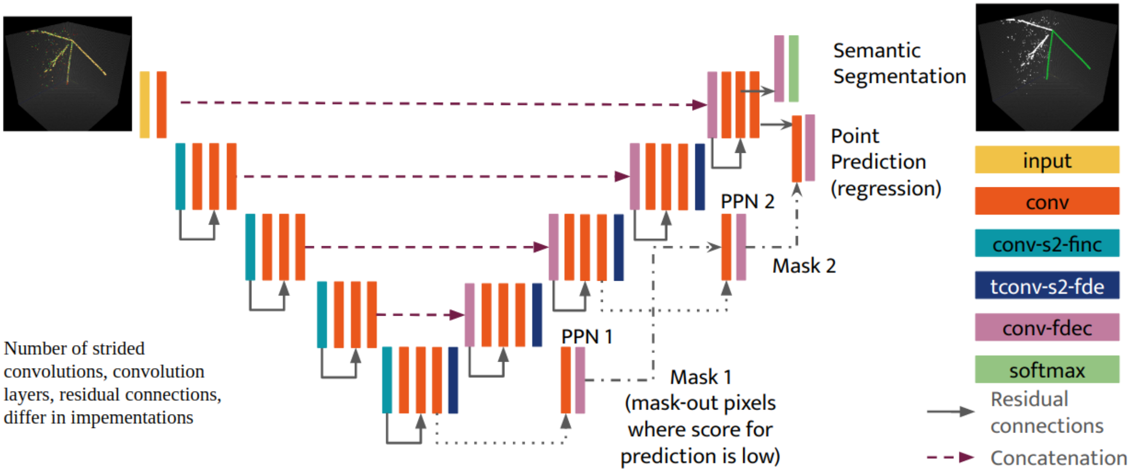
Particle type	Voxel fraction	Accuracy
HIP	17 %	98.2 %
MIP	34 %	99.4 %
Showers	47 %	99.2 %
Delta rays	1 %	96 %
Michel	1 %	94.7 %
Total		**99 %**

Network adapted to very sparse data,
see paper for details: arXiv:1903.05663



Particle type predicted by the network

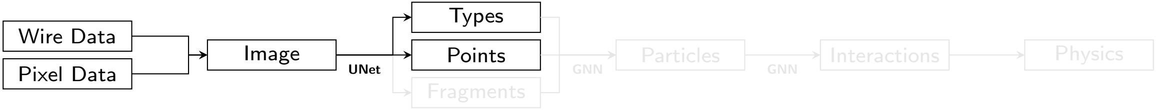
Pixel feature extraction | Point proposal

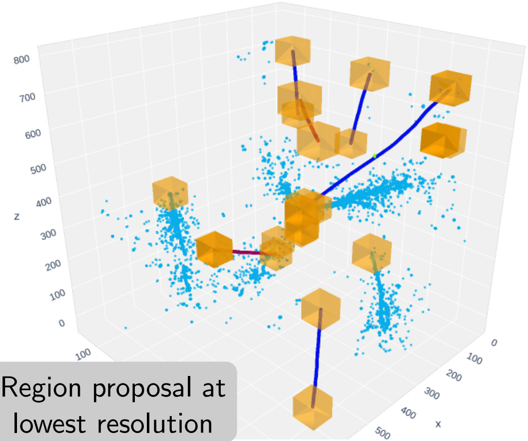


Number of strided convolutions, convolution layers, residual connections, differ in impementations

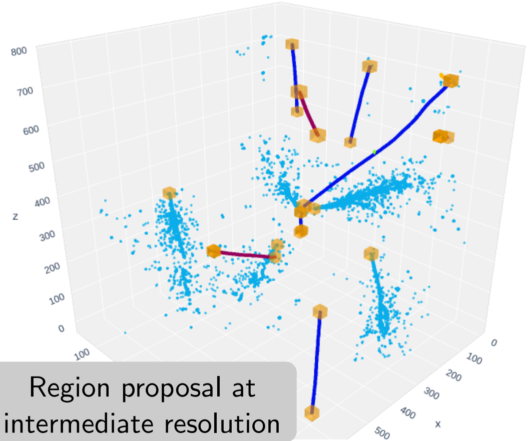
doi.org/10.5281/zenodo.1300713

L. Domine



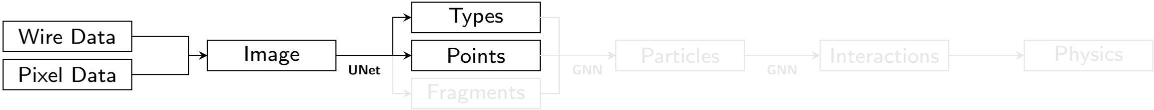


Region proposal at lowest resolution

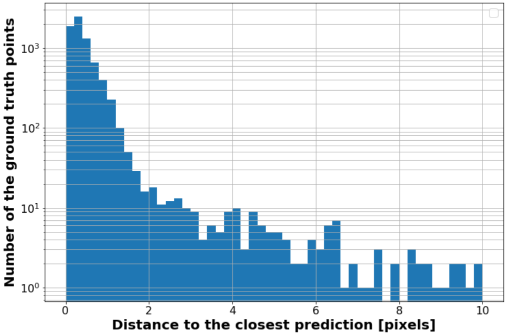


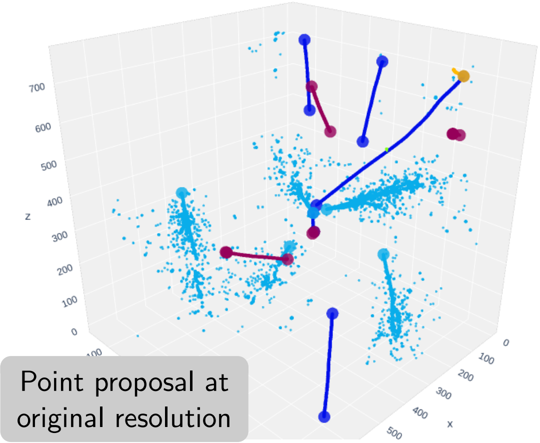
Region proposal at intermediate resolution

Pixel feature extraction | Point proposal



Point proposal efficiency (97 % < 10 px):

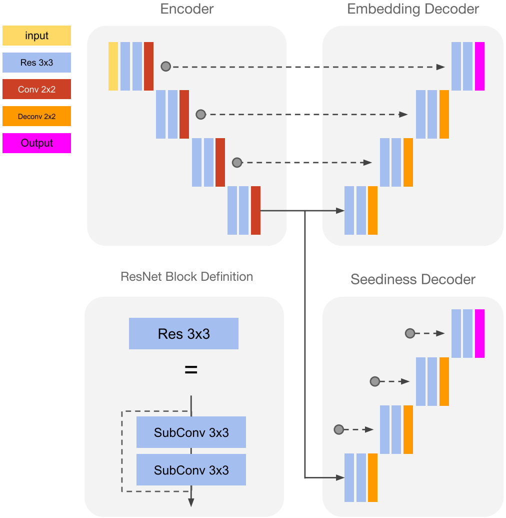


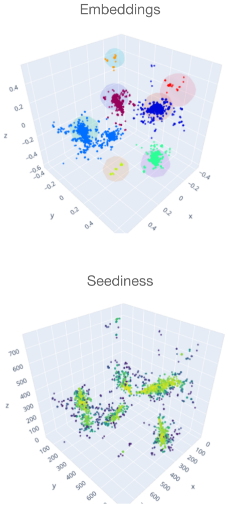


doi.org/10.5281/zenodo.1300713

Pixel feature extraction | Fragment clustering

NATIONAL ACCELERATOR LABORATORY





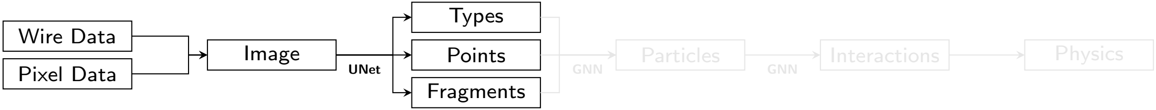
Network predicts 3 things:

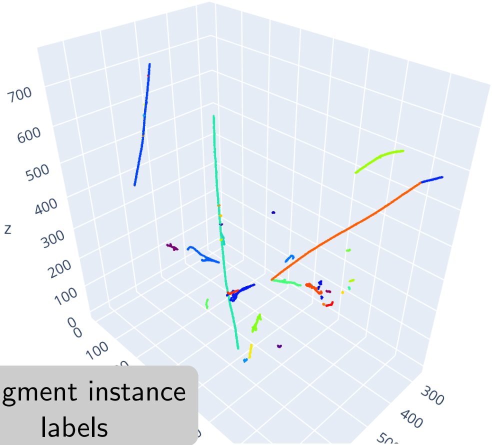
- **Embedding**: space in which fragments are spatially separated
- **Seediness**: likelihood that a voxel is a cluster centroid in embedding space
- **Margin**: Size of the cluster in embedding space

D. H. Koh

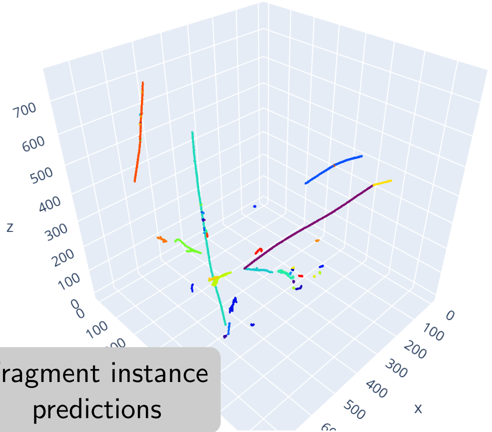
arXiv:1906.11109

SLAC NATIONAL ACCELERATOR LABORATORY

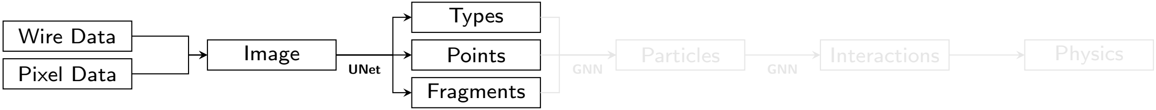




Fragment instance labels

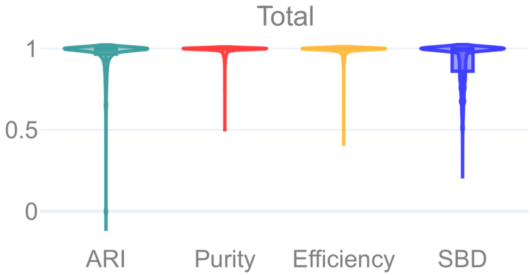


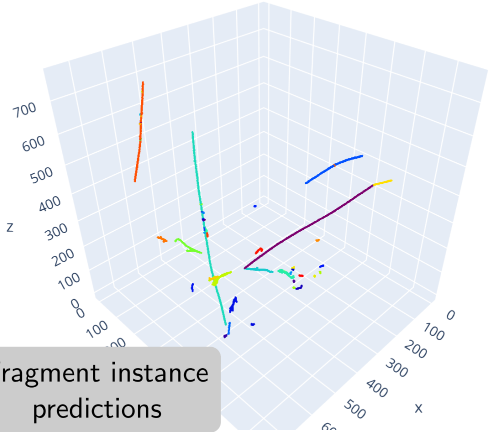
Fragment instance predictions



Fragment clustering accuracy:

- Mean ARI: **95.2 %**

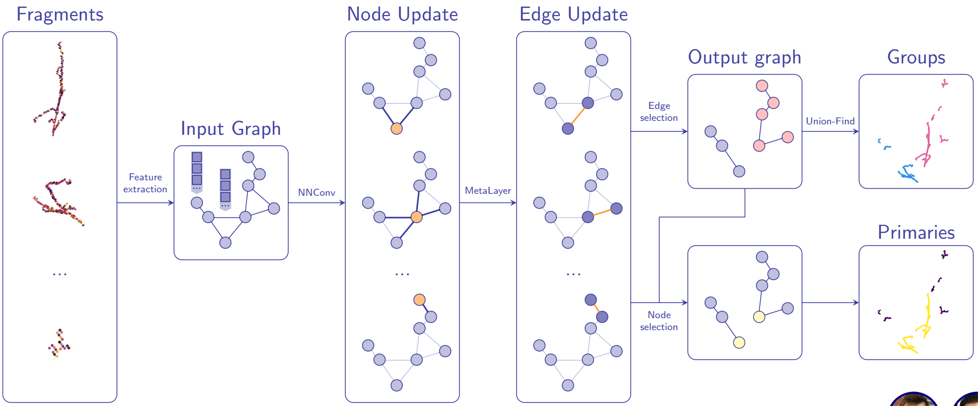




Fragment instance predictions

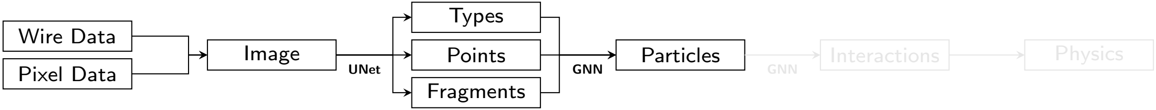
Graphical Neural Networks (GNN) are ideal to cluster spatially detached objects:

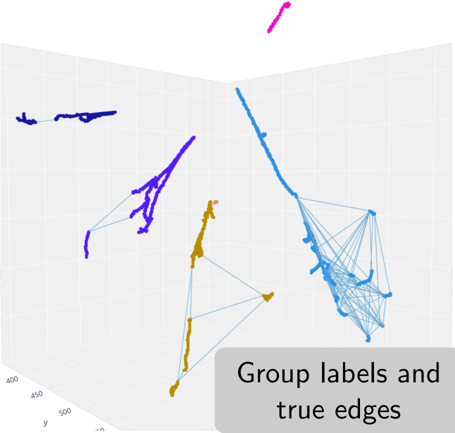
- Based on *nodes* and *edges*. Features propagate by *message passing* (MP)



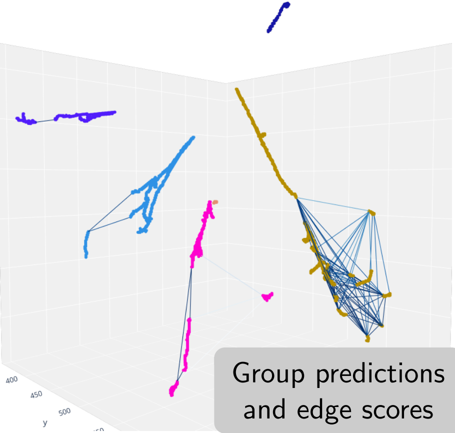
See my other CTD talk for more details !

F. Drielsma, Q. Lin, P. Cotes de Soux

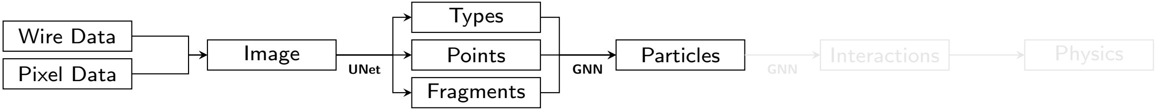




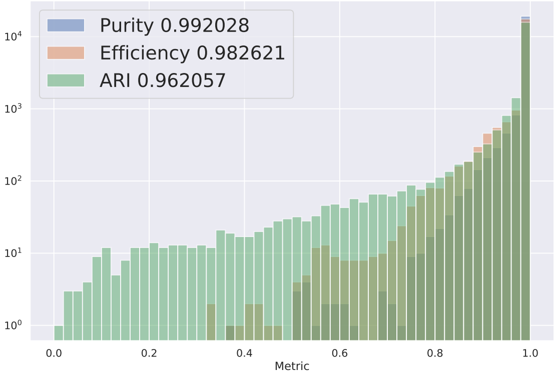
Group labels and true edges

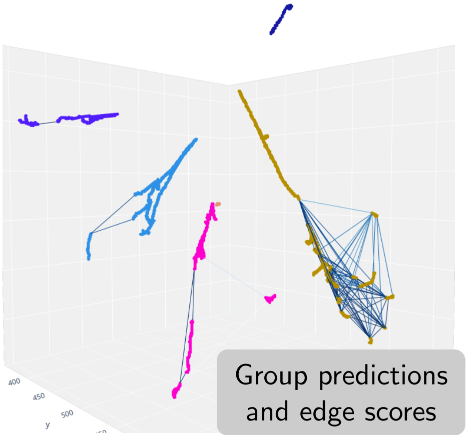


Group predictions and edge scores

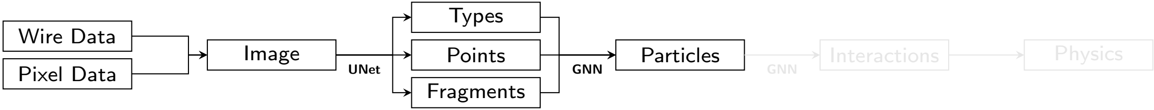


Shower clustering accuracy:

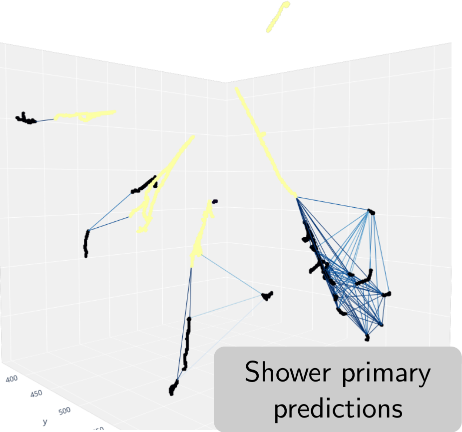




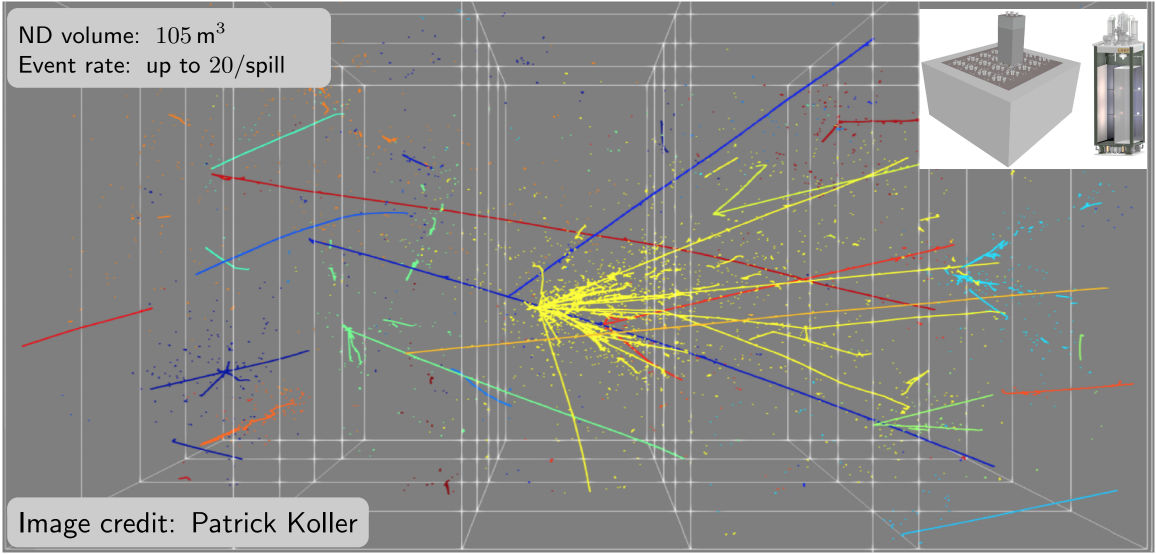
Shower Clustering | Start identification



Shower primary accuracy (**98.5 %**):



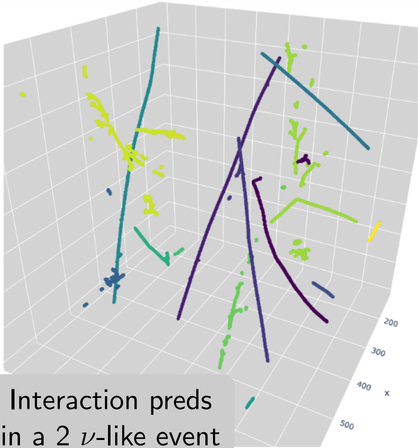
High rate LArTPCs | DUNE ND



Wire Data / Pixel Data → Image $\xrightarrow{\text{UNet}}$ Types / Points / Fragments $\xrightarrow{\text{GNN}}$ Particles $\xrightarrow{\text{GNN}}$ Interactions → Physics

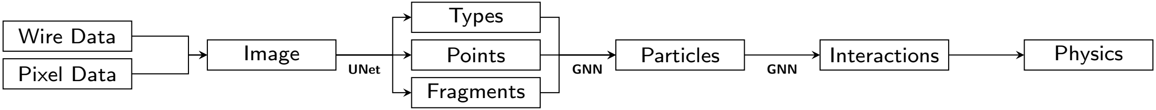
Interaction clustering performance:

Metric	# of ν-like	Mean Score
Efficiency	1	98.8 %
	2	**97.6 %**
	4	95.6 %
Purity	1	99.4 %
	2	**99.3 %**
	4	99.3 %
ARI	1	95.6 %
	2	**93.2 %**
	4	88.0%



Interaction preds in a 2 ν-like event

Summary



ML Reconstruction Chain for LArTPCs:

- Trend in neutrino detection: high-resolution particle imaging
- Resulting analysis trend: computer vision $\rightarrow$ **Machine Learning**
- LArTPC images too information rich to be reduced to simple variables in one pass
- **Hierarchical feature extraction** very successful so far

Areas we will work on not covered in this talk:

- Deghosting autoencoders
- Data vs simulation domain discrepancy

Please email me for more details or if you want to participate !

Back-up slides

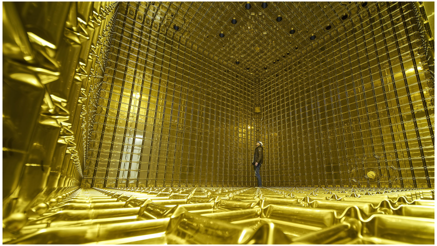
Sizes of current and future LArTPCs:

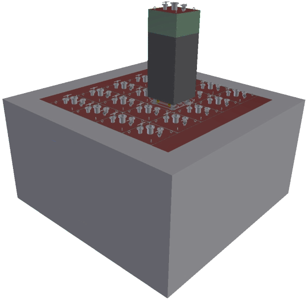
- μBOONE: 100 t (10x2.5x2.5 m^3)
- pDUNE: 200 t (6x6x6 m^3)
- ICARUS: 400 t (2x(20x3x3) m^3)
- ArgonCube 2x2: 10 t (4x(0.67x0.67x2) m^3)
- DUNE-ND: 150 t (35x(1x1x3) m^3)
- DUNE-FD: 40 kt (4x(12x12x60) m^3)

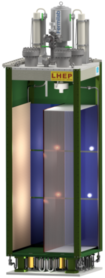
Some example numbers (for ICARUS):

- Wire pitch: 3 mm
- Angle between planes: 60°
- Drift field: 500 V/cm
- Drift velocity: ~ 0.15 cm/μs
- TPC time resolution: $0.4\,\mu$s (< 1 mm)
- PMT coverage: $\sim$2 %
- Scintillation light: 20 % prompt (6 ns), 80 % late (1.5 μs)
- Photon yield: 24000/MeV

Liquid Argon TPCs | Pictures

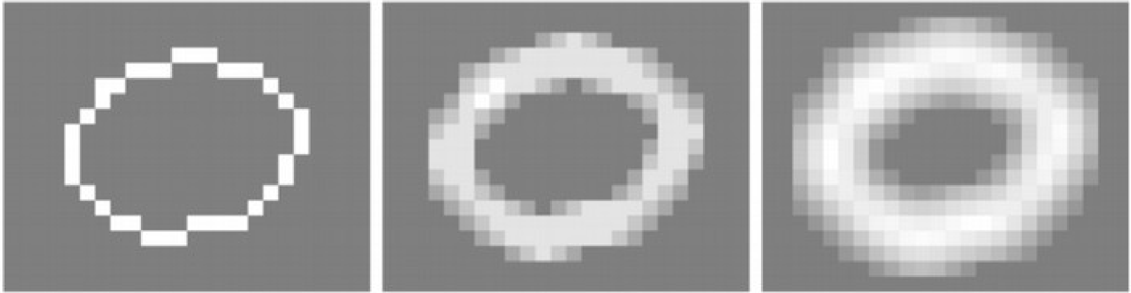






Submanifold Sparse Convolutions

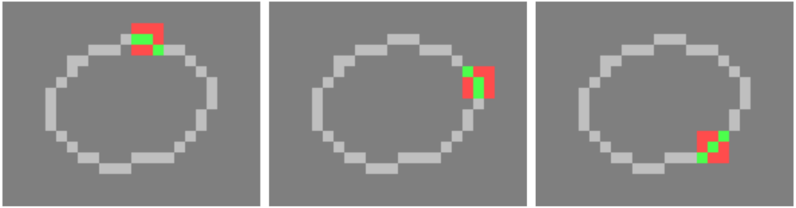
1. Resources waste of dense convolutions on sparse data
2. Dilation problem
   - One nonzero site leads to 3d nonzero sites after 1 convolution
   - How to keep the same level of sparsity throughout the network?



https://arxiv.org/pdf/1711.10275.pdf

In more details, two new operations:

- Sparse convolutions (SC)
  - Discards contribution of non-active input sites
  - Output site active if at least one input site is active
- Sparse submanifold convolutions (SSC)
  - Output size = Input size
  - Output site active iff center of receptive field active
  - Only compute features for active output sites



https://arxiv.org/pdf/1711.10275.pdf

PPN outputs **voxel location**, **position within voxel** and **point class**

Three components to the point proposal loss:

- Pixel classification loss at each of three depth (pixel contains point or not)

$$\mathcal{L}_{\text{class},i} = \frac{1}{N_i}\sum_{k=1}^{N_i} y_k \log(p_k) + (1-y_k)\log(1-p_k)$$

- L^1 distance from true point at highest resolution **on active voxels**

$$\mathcal{L}_{\text{dist}} = \frac{1}{N_3^*}\sum_{k=1}^{N_3^*} \min_j ||\vec{p}_i - \vec{q}_j||$$

- Particle type loss at highest resolution **on active voxels**

$$\mathcal{L} = \frac{1}{N_3^*}\sum_{k=1}^{N_3^*}\sum_c y_{k,c}\log(p_{k,c})$$

SLAC NATIONAL ACCELERATOR LABORATORY

Instance Segmentation:

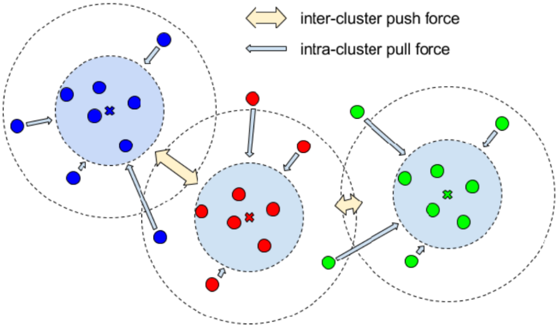
- Three component loss: pull together points that belong to the same cluster, keep distance between clusters, and regularization

$$L = \alpha L_{var} + \beta L_{dist} + \gamma L_{reg},$$

$$L_{var} = \frac{1}{C}\sum_{c=1}^{C}\frac{1}{N_c}\sum_{i=1}^{N_c}[\max(0, \|\mu_c - x_i\| - \delta_v)]^2$$

$$L_{dist} = \frac{1}{C(C-1)}\sum_{\substack{c_A, c_B = 1 \\ c_A \neq c_B}}^{C}[\max(0, 2\delta_d - \|\mu_{c_A} - \mu_{c_B}\|)]^2$$

$$L_{reg} = \frac{1}{C}\sum_{c=1}^{C}\|\mu_c\|$$

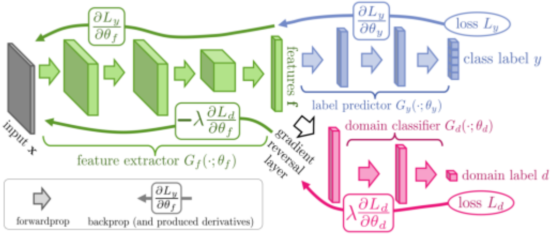


arXiv:1708.02551

Domain discrepancies

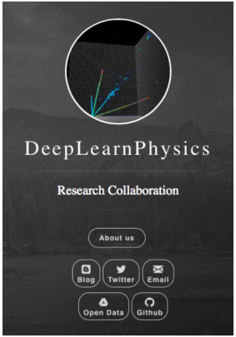
What can we do about imperfect simulation ?

- Issue: the signal distribution learned by the algorithm may be different in two domains!
- Mitigation techniques in ML domain ?
  - Can try CNN to locate where it is
  - Can try CNN to fix the discrepancy
  - Can try a training technique to minimize the effect



Maximize the loss to discriminate data vs. simulation, feature extractors are penalized to key on simulation specific information

Domain-Adversarial Training of Neural Networks: J. Mach. Learn. Res. 17 (2016)



DeepLearnPhysics: Collaboration for ML technique R&D

- Open simulation sample (used throughout this talk)
  - Open real data ? Soon ! (3D prototype R&D at SLAC)
- Open source container (Singularity)
- Open source code (GitHub)
  - All the code used to make this talk is available

→ **Reproducible results !**

- Readers have reproduced arXiv:1903.05663