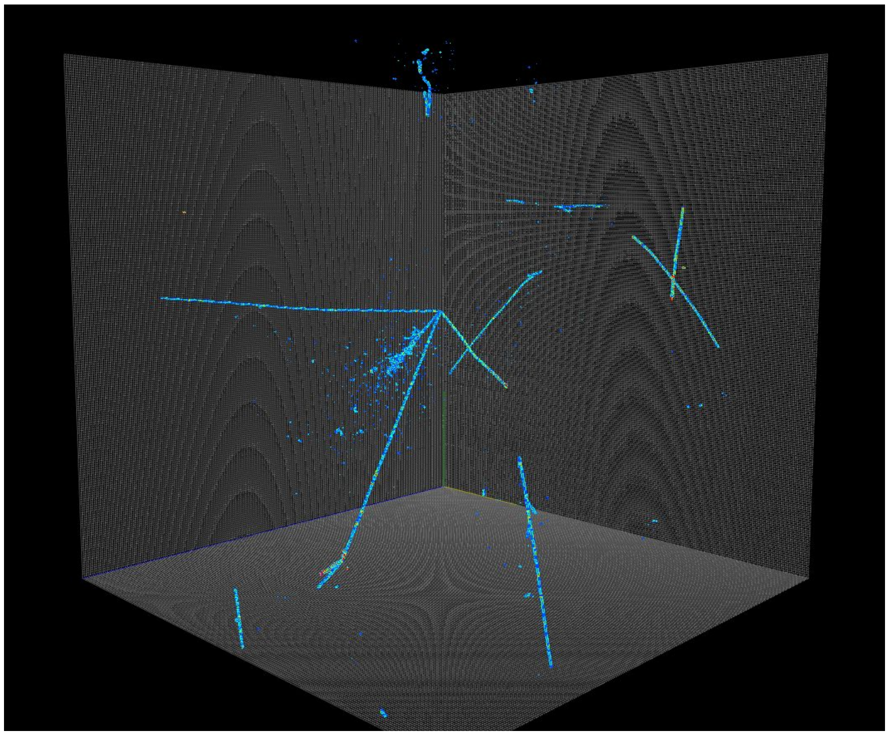
Pi-Zero Reconstruction Approach

1. Prepare data in 3D image (Nx,Ny,Nz,Nc)
2. U-ResNet for pixel segmentation
3. Point Proposal Network for start finding
4. PCA for direction estimate
5. Shower clustering
6. Pi-Zero mass reconstruction

Pi-Zero Reconstruction Approach

1. Prepare data in 3D image (Nx,Ny,Nz,Nc)

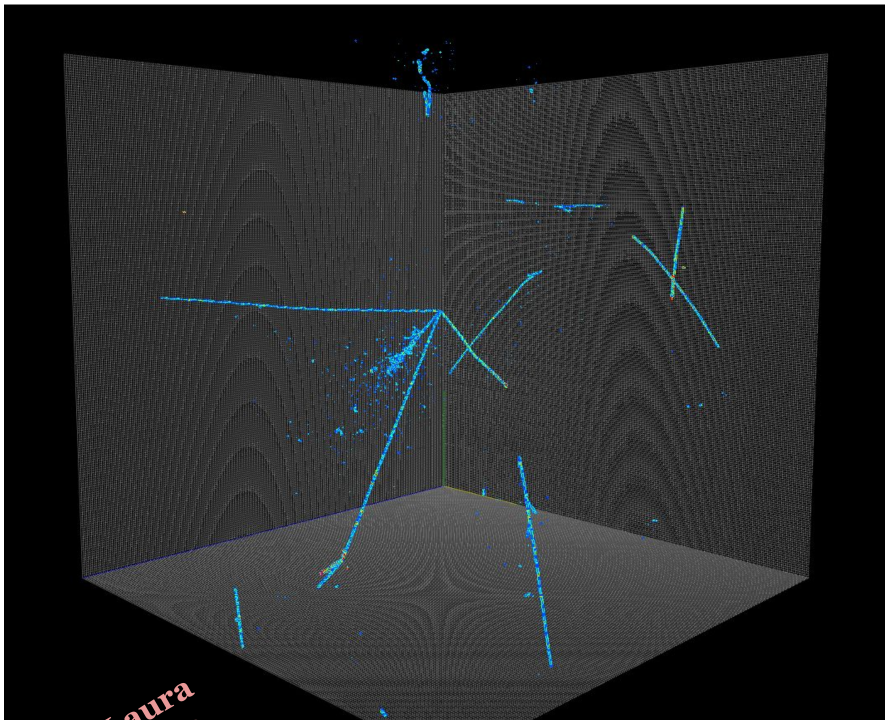


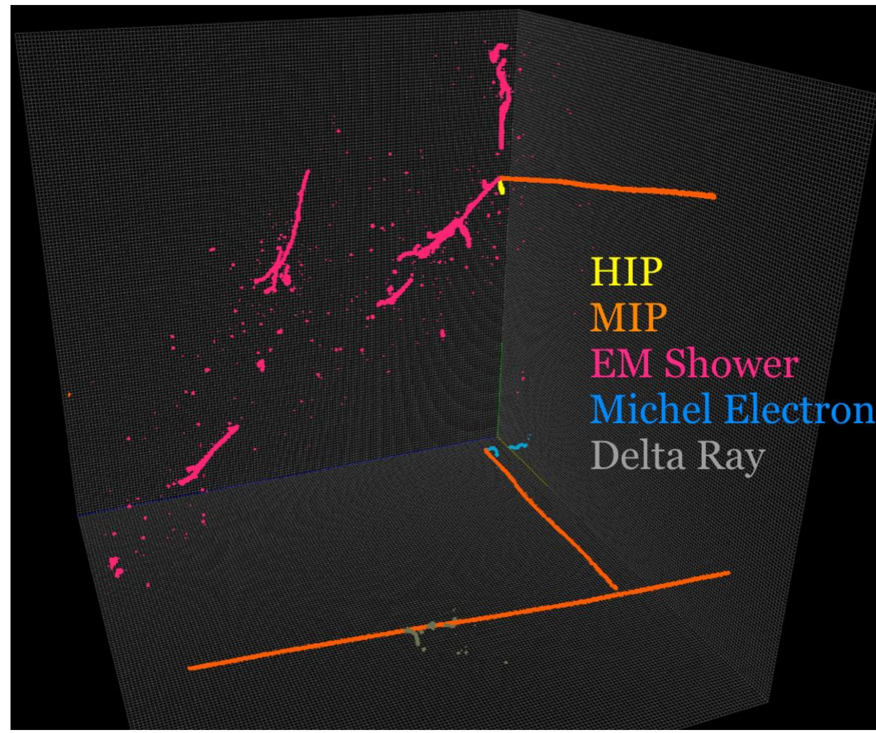
Current approach

- LArCV format
  - Sparse tensor, C++ with numpy wrapper
- Cluster3D (else?)
- SimEnergyDeposit
  - For "label"
  - Alt: SimChannel

Pi-Zero Reconstruction Approach

1. Prepare data in 3D image (Nx,Ny,Nz,Nc)
2. U-ResNet for pixel segmentation





Type	HIP	MIP	Shower	Delta rays	Michel
Accuracy*	0.993	0.981	0.992	0.972	0.957

*Accuracy = correctly labeled pixel fraction

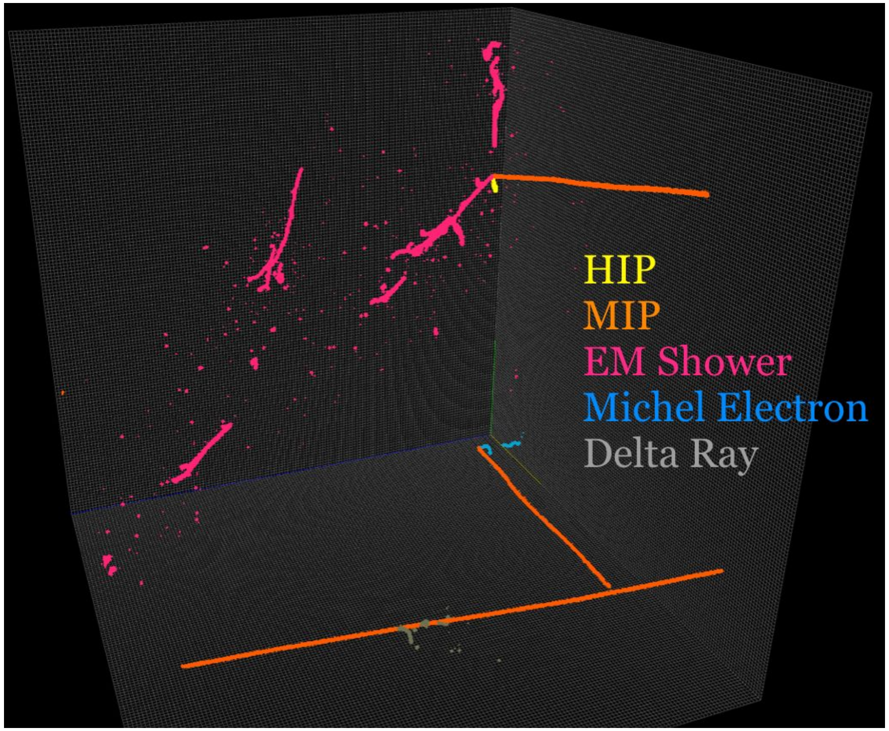
Note: using public simulation set

Pi-Zero Reconstruction Approach

1. Prepare data in 3D image (Nx,Ny,Nz,Nc)
2. U-ResNet for pixel segmentation

Segmentation for Reco
(e.g. Michel electron)

- Mask all pixels but Michel electrons, run spatial DBScan
- Same for MIP pixels
- Keep only Michel clusters which edge touches with MIP edge



By Laura Domine

Type	HIP	MIP	Shower	Delta rays	Michel
Accuracy*	0.993	0.981	0.992	0.972	0.957

*Accuracy = correctly labeled pixel fraction

Note: using public simulation set

Pi-Zero Reconstruction Approach

1. Prepare data in 3D image (Nx,Ny,Nz,Nc)
2. U-ResNet for pixel segmentation

Segmentation for Reco
(e.g. Michel electron)

- Mask all pixels but Michel electrons, run spatial DBScan
- Same for MIP pixels
- Keep only Michel clusters which edge touches with MIP edge

Type	HIP	MIP	Shower	Delta rays	Michel
Accuracy*	0.993	0.981	0.992	0.972	0.957

*Accuracy = correctly labeled pixel fraction

Note: using public simulation set

Pi-Zero Reconstruction Approach

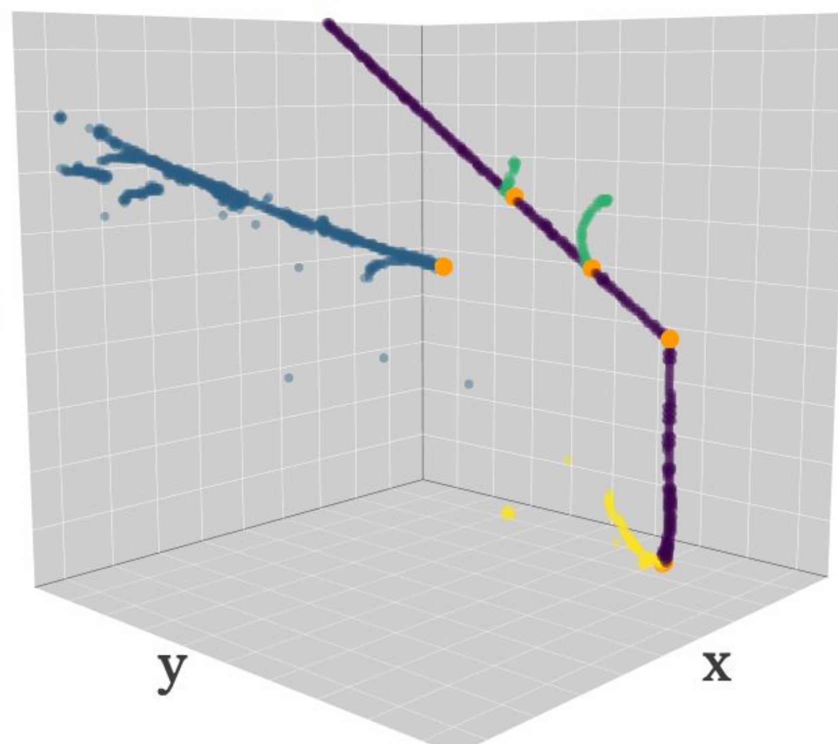
1. Prepare data in 3D image (Nx,Ny,Nz,Nc)
2. U-ResNet for pixel segmentation
3. Point Proposal Network for start finding
4. PCA for direction estimate

PPN: Point Proposal Network

- Look for start/end of MIP/HIP as well as start of showers
- Work-in-progress for merging with sparse U-ResNet (Laura)

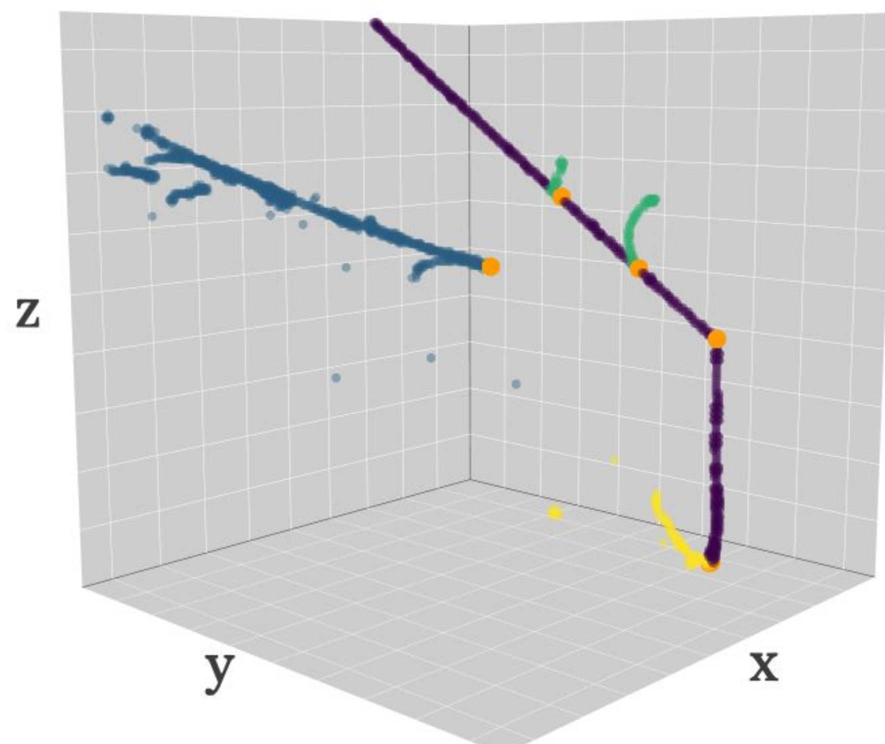
PCA: Principle Comp. Analysis

- Estimate shower direction using short fragment of shower pixels near the start.



Pi-Zero Reconstruction Approach

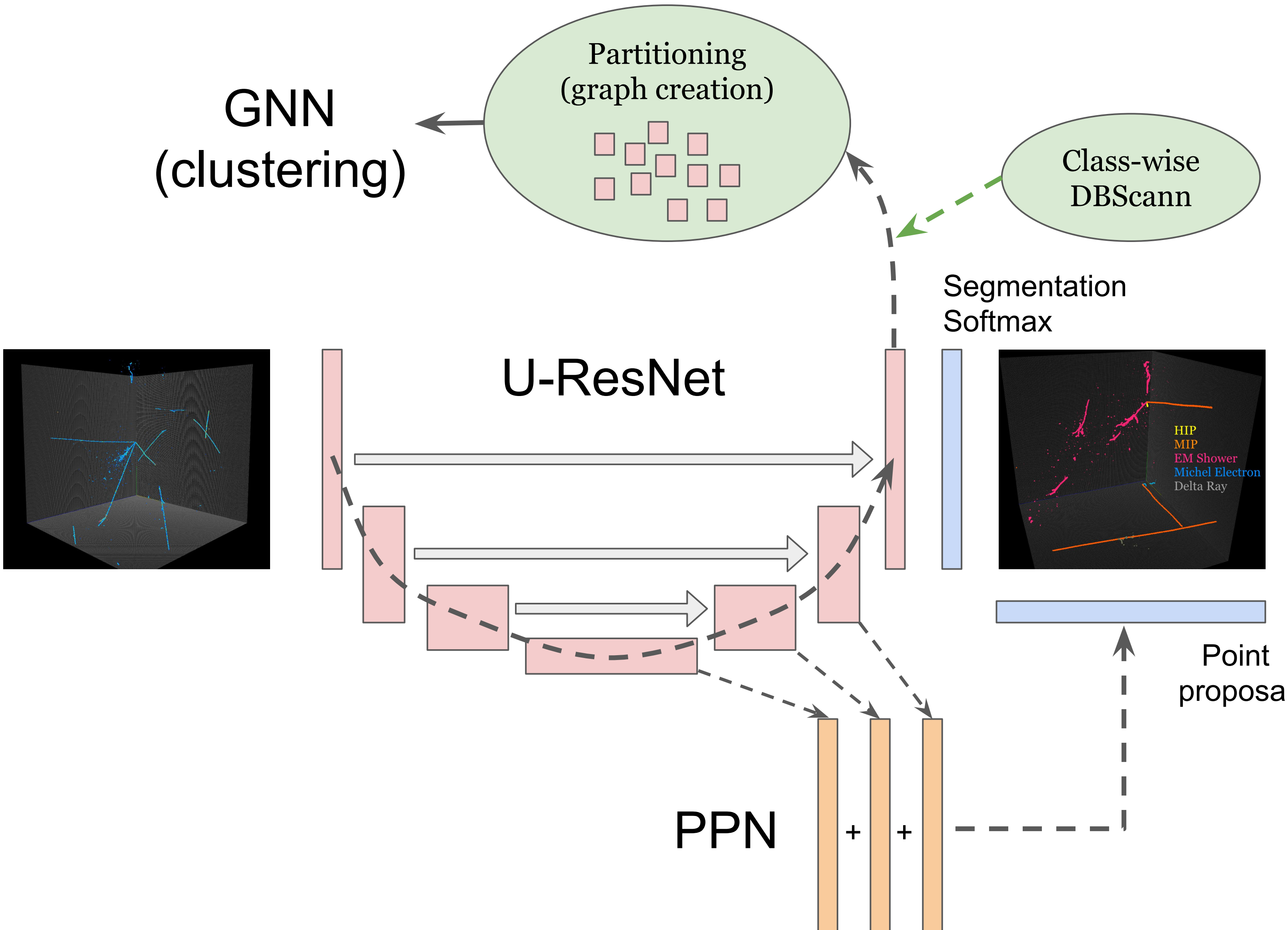
1. Prepare data in 3D image (Nx,Ny,Nz,Nc)
2. U-ResNet for pixel segmentation
3. Point Proposal Network for start finding
4. PCA for direction estimate
5. Shower clustering



GCN: Graph Neural Network

- Put together the DBScan-ed shower fragments
- Treat each fragment as a graph node, connection to other nodes as edges. Analyze edges to find correct connections = clustering (work-in-progress by Brad)

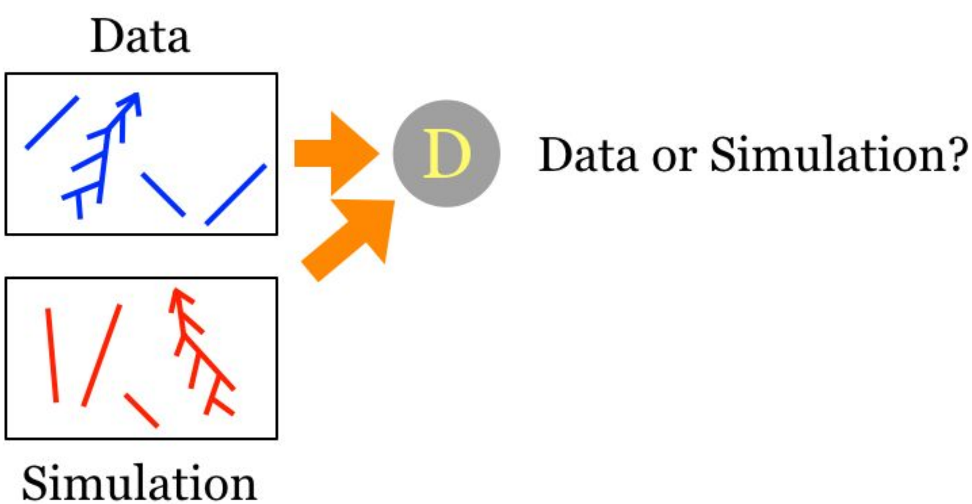
Whole Chain


Partitioning
(graph creation)
GNN
(clustering)
Class-wise
DBScann
Segmentation
Softmax
U-ResNet
HIP
MIP
EM Shower
Michel Electron
Delta Ray
Point
proposal
PPN
+
+

Data/Simulation Discrepancy Mitigation

What can we do about imperfect simulation?

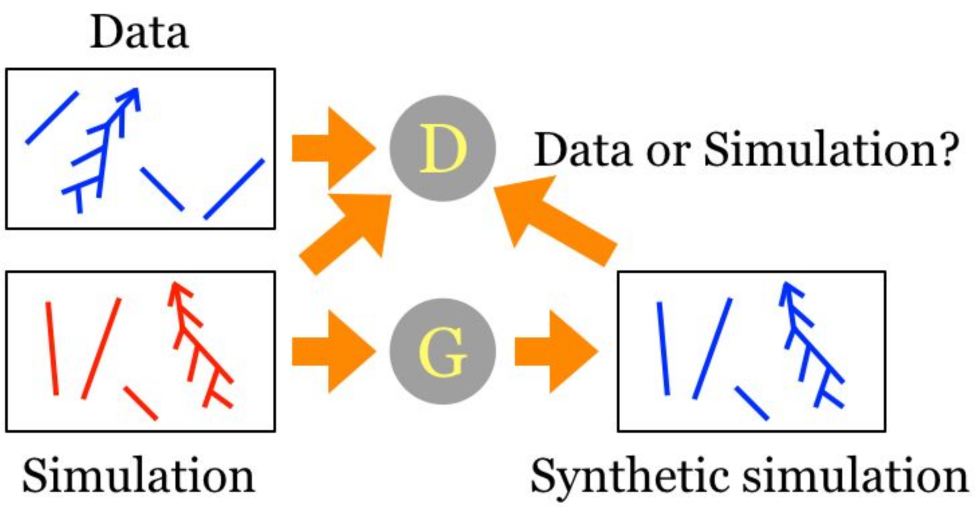
- Problematic: the "signal distribution" learnt by the algorithm may be different in two domains!
- Mitigation techniques in ML domain?
    - Can try DNN to "locate" where it is



Data/Simulation Discrepancy Mitigation

What can we do about imperfect simulation?

- Problematic: the “signal distribution” learnt by the algorithm may be different in two domains!
- Mitigation techniques in ML domain?
  - Can try DNN to “locate” where it is
  - Can try DNN to “fix” the discrepancy (hardest)

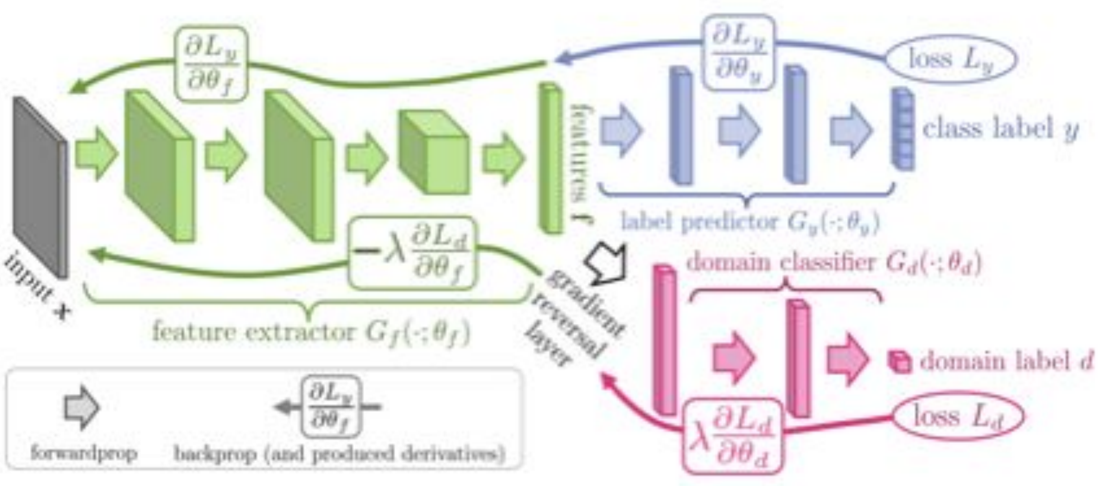


Generative Adversarial Network
Can learn the “mapping” between the data and simulation “distributions”. The generator network can be used as a synthetic image generator to train different neural networks

Data/Simulation Discrepancy Mitigation

What can we do about imperfect simulation?

- Problematic: the "signal distribution" learnt by the algorithm may be different in two domains!
- Mitigation techniques in ML domain?
  - Can try DNN to "locate" where it is
  - Can try DNN to "fix" the discrepancy (hardest)
  - Can try training technique to minimize the effect



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

Minerva Paper: arXiv:1808.08332

Domain-Adversarial Training of NN: J. Mach. Learn. Res. 17 92016)

Synergy

- ML-based 3D reconstruction chain development
  - MicroBooNE, ICARUS, protoDUNE, ArgonCUBE/DUNE-ND
  - Common denominator workers: Kazu/Laura/Brad
- Software + LArSoft
  - Currently Pytorch (+ dependencies) as default choice
  - Pytorch 1.0 integration w/ LArSoft on-going
  - Current model
    - Data generation using larcv+dunetpc build
    - Training/inference in Python + pytorch
    - Example to play with software/samples (not protoDUNE)?
      - Data, Container, U-ResNet repository
-