THE UNIVERSITY OF CHICAGO

DNN-Augmented ROI Finding for SBN Signal Processing

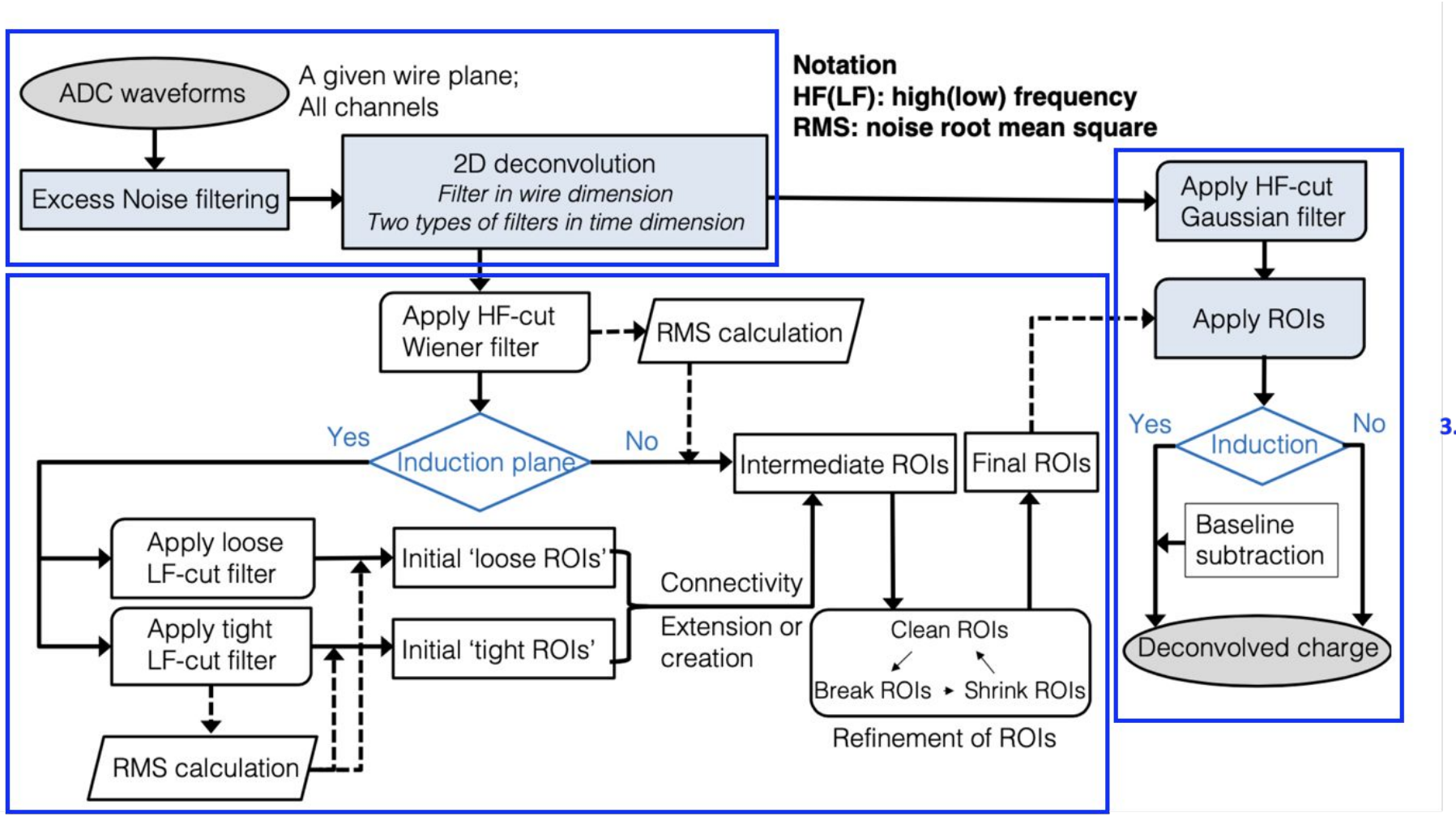
Mun Jung Jung

WireCell Signal Processing

1. 2D Deconvolution

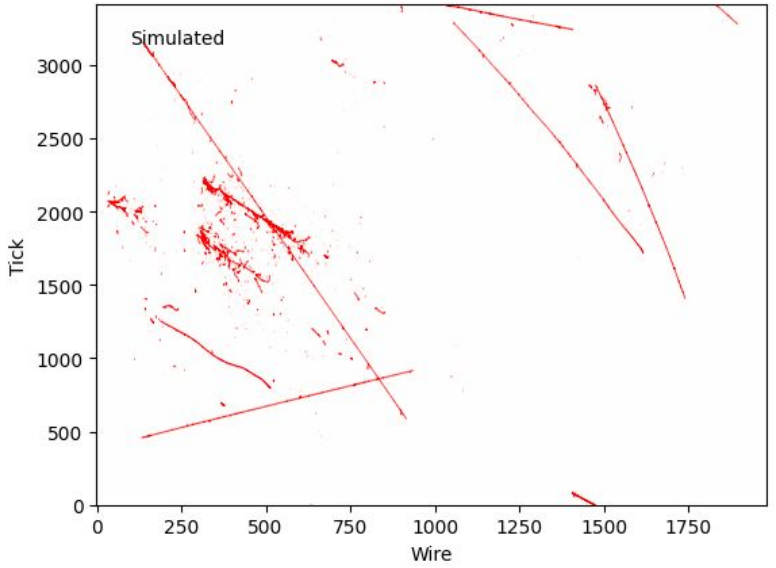
2. ROI Finding

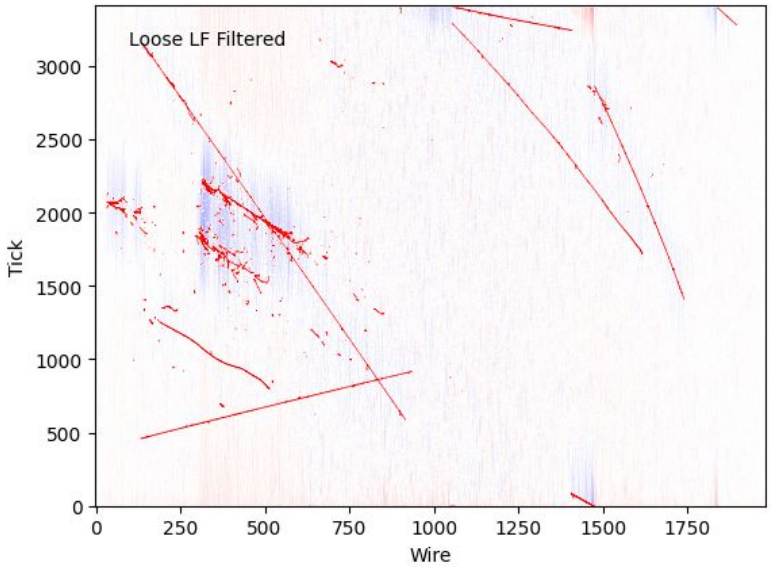
3. Charge Extraction

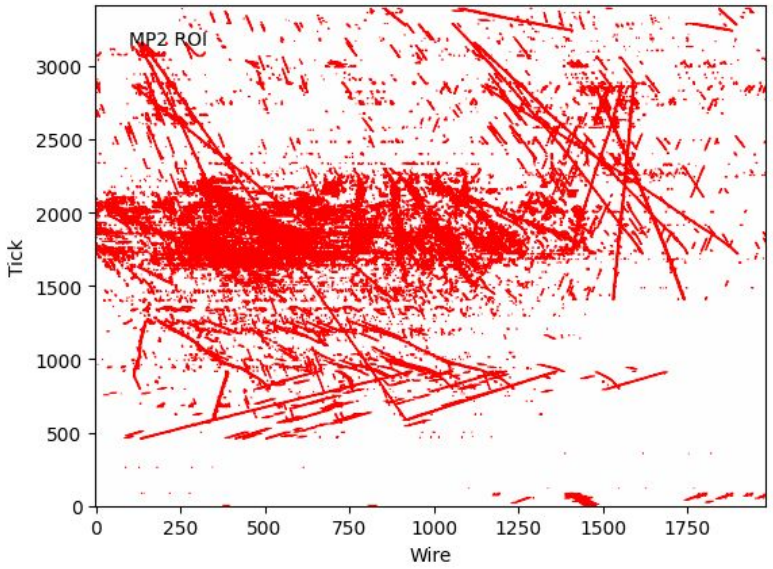


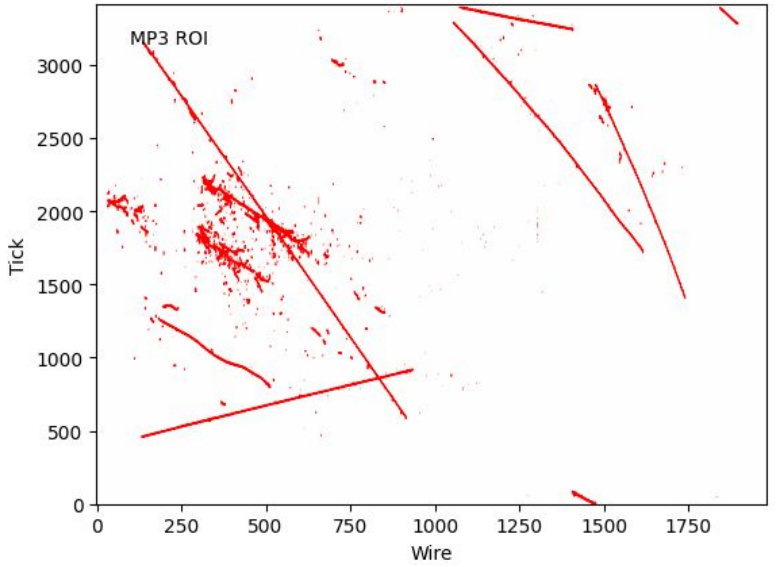
DNN-augmented ROI Finding

1. multi-plane matching: information across plane
2. 2D info as input images





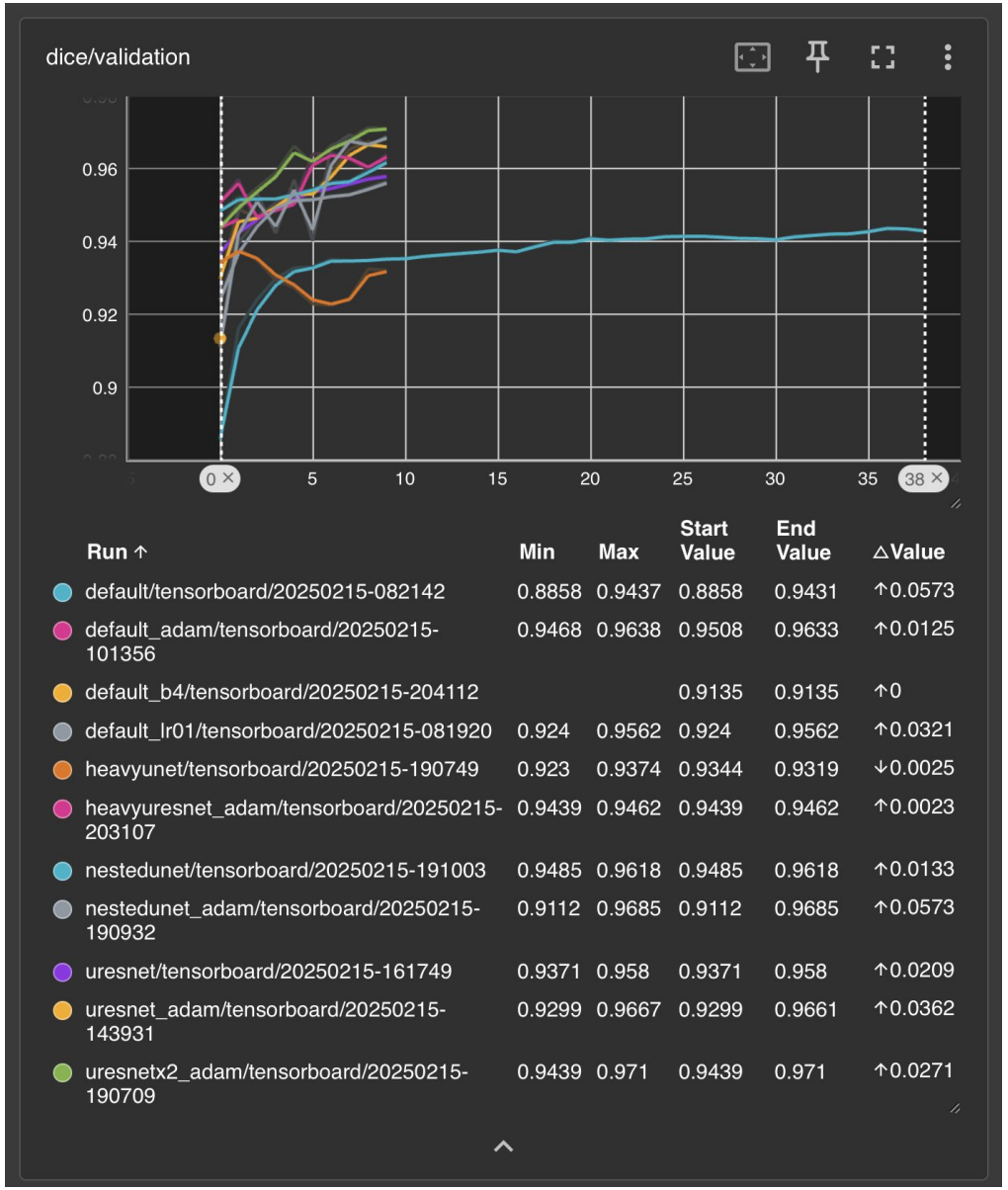




ICARUS Training

ICARUS

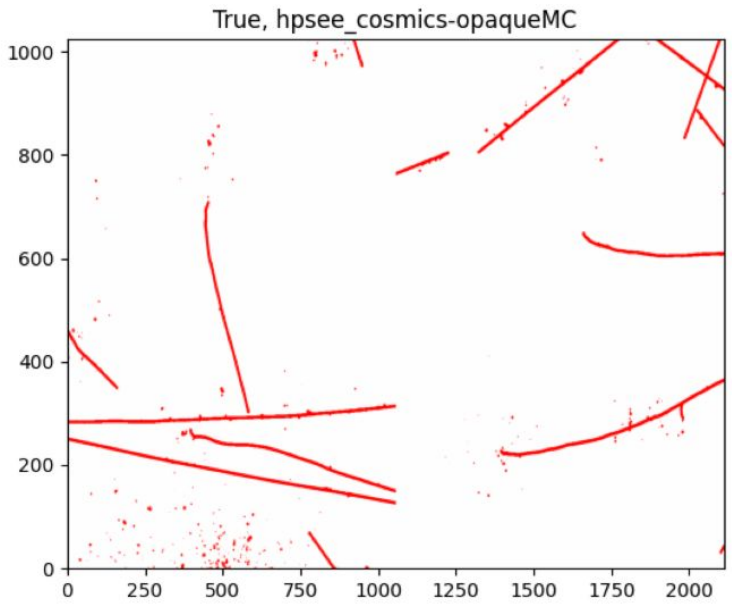
- these variations resulted ~same performance for SBND, but we see clear differences for ICARUS!
- summary:
  - UResNet > UNet
  - larger network -> better

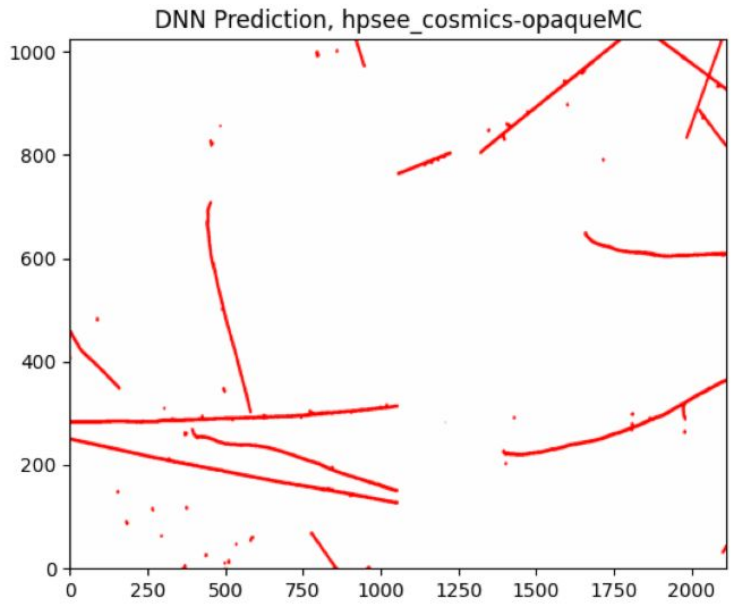


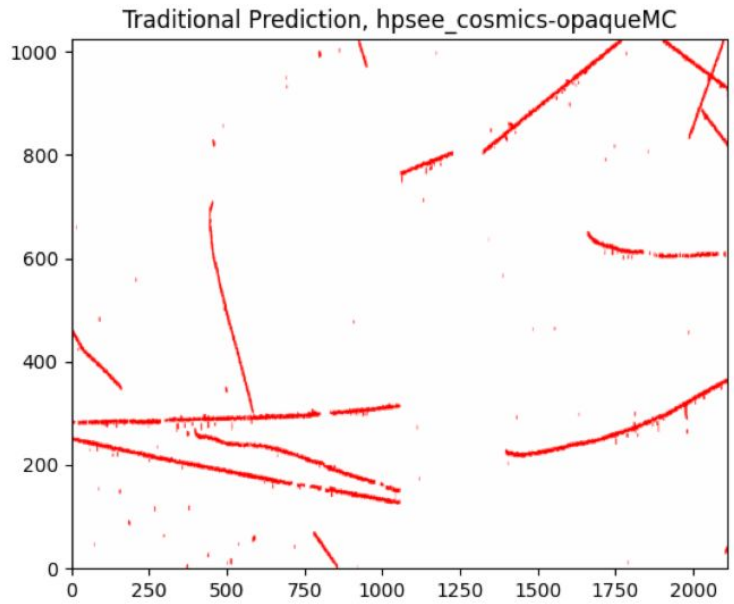
Run ↑	Min	Max	Start Value	End Value	△Value
default/tensorboard/20250215-082142	0.8858	0.9437	0.8858	0.9431	↑0.0573
default_adam/tensorboard/20250215-101356	0.9468	0.9638	0.9508	0.9633	↑0.0125
default_b4/tensorboard/20250215-204112			0.9135	0.9135	↑0
default_lr01/tensorboard/20250215-081920	0.924	0.9562	0.924	0.9562	↑0.0321
heavyunet/tensorboard/20250215-190749	0.923	0.9374	0.9344	0.9319	↓0.0025
heavyuresnet_adam/tensorboard/20250215-203107	0.9439	0.9462	0.9439	0.9462	↑0.0023
nestedunet/tensorboard/20250215-191003	0.9485	0.9618	0.9485	0.9618	↑0.0133
nestedunet_adam/tensorboard/20250215-190932	0.9112	0.9685	0.9112	0.9685	↑0.0573
uresnet/tensorboard/20250215-161749	0.9371	0.958	0.9371	0.958	↑0.0209
uresnet_adam/tensorboard/20250215-143931	0.9299	0.9667	0.9299	0.9661	↑0.0362
uresnetx2_adam/tensorboard/20250215-190709	0.9439	0.971	0.9439	0.971	↑0.0271

ICARUS examples

ICARUS induction planes are unhealthy – improvement from DNN ROI finding is much more significant

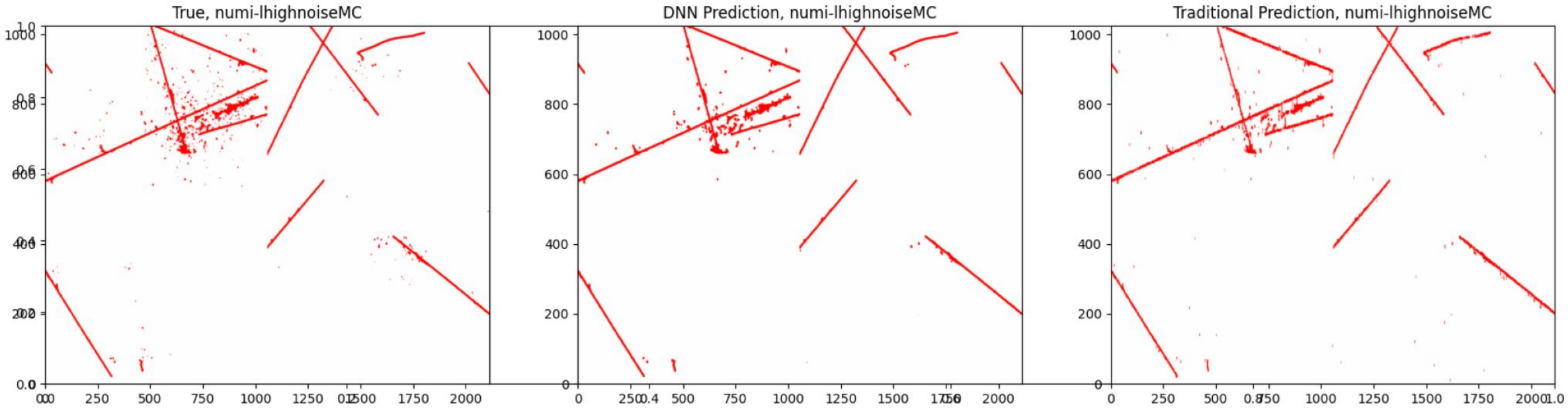






ICARUS examples

ICARUS induction planes are unhealthy – improvement from DNN ROI finding is much more significant



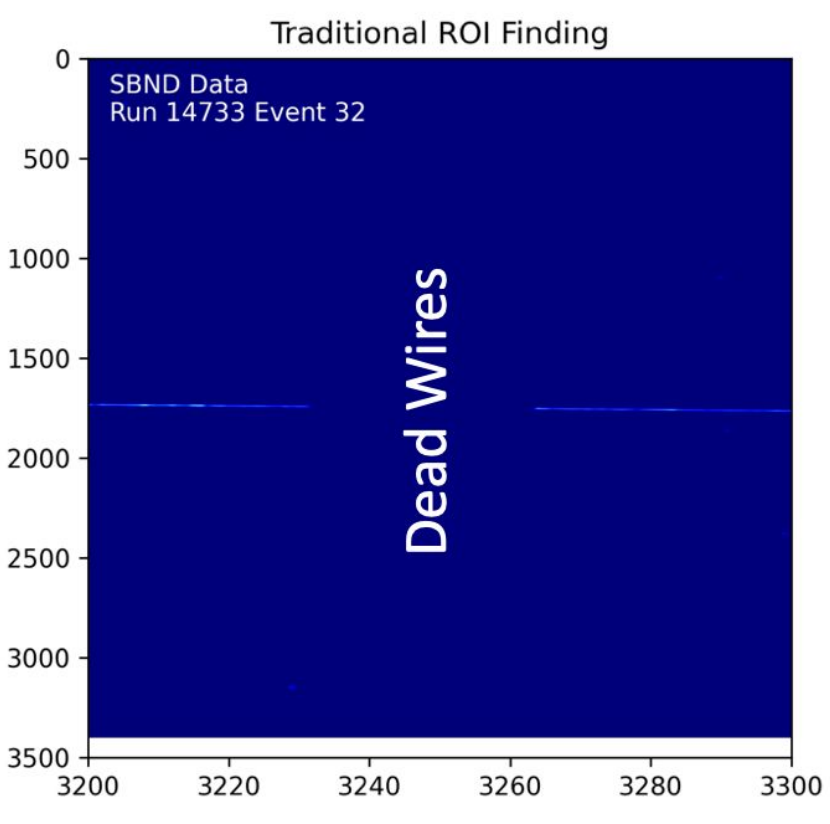
Resilience to Detector Variations

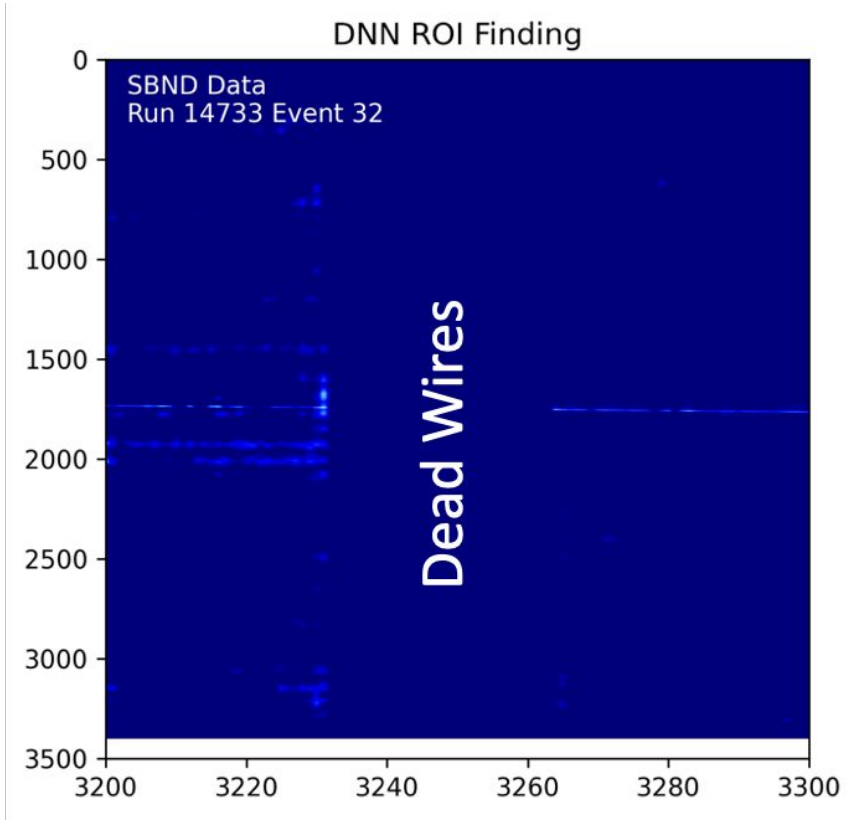
SBND: Input Sample Variations

- vary input samples to emulate detector effects
  - assuming there aren't variations that break/significantly impact the cross-plane matching
  - randomly vary half of input samples to loose LF input to emulate detector variations
    - mask out channels: unexpected dead channels
    - weight pixel values with masks: gain variations
    - randomly smearing in tick direction (sigma = 4ticks)
    - vary truth mask threshold: noise

ex) Dead Wires

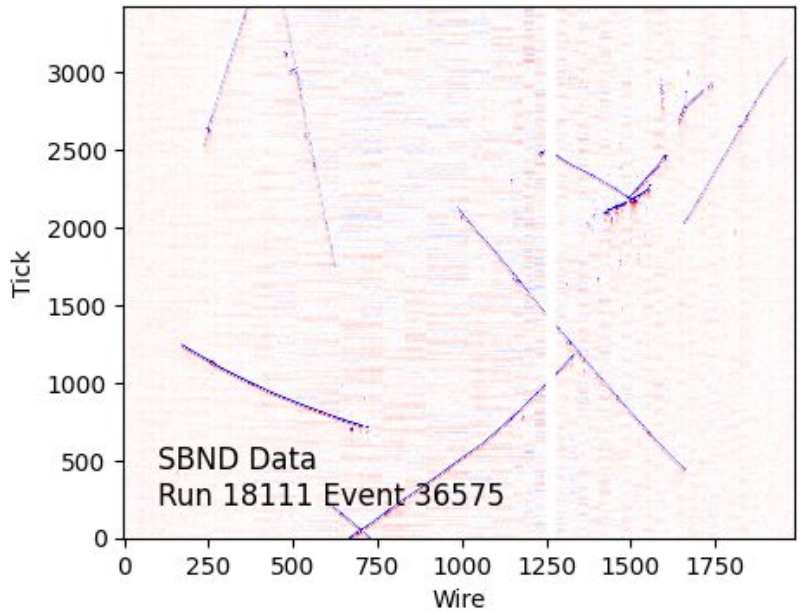
without sample variation

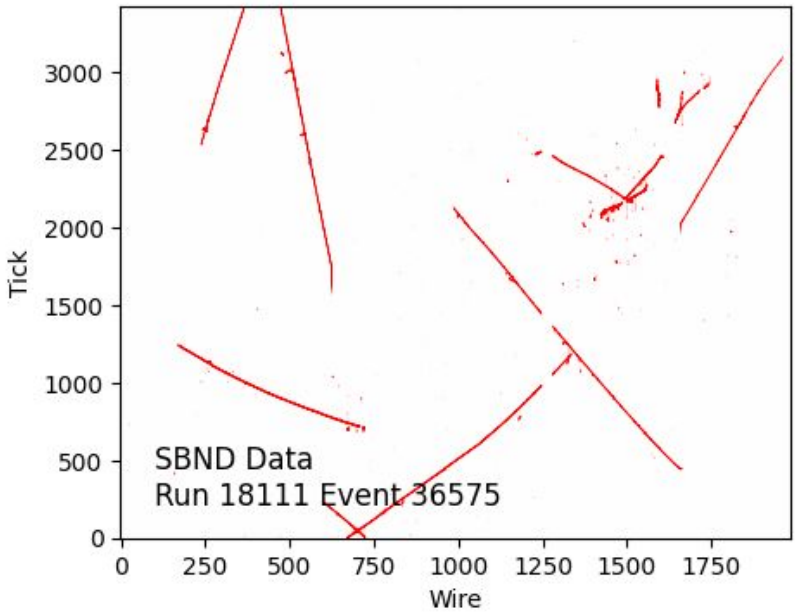


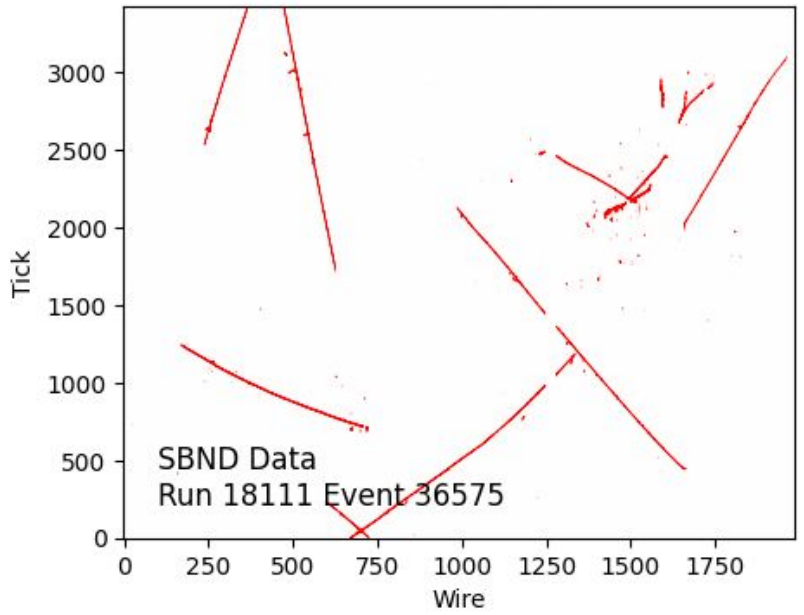


ex) Dead Wires

with sample variation

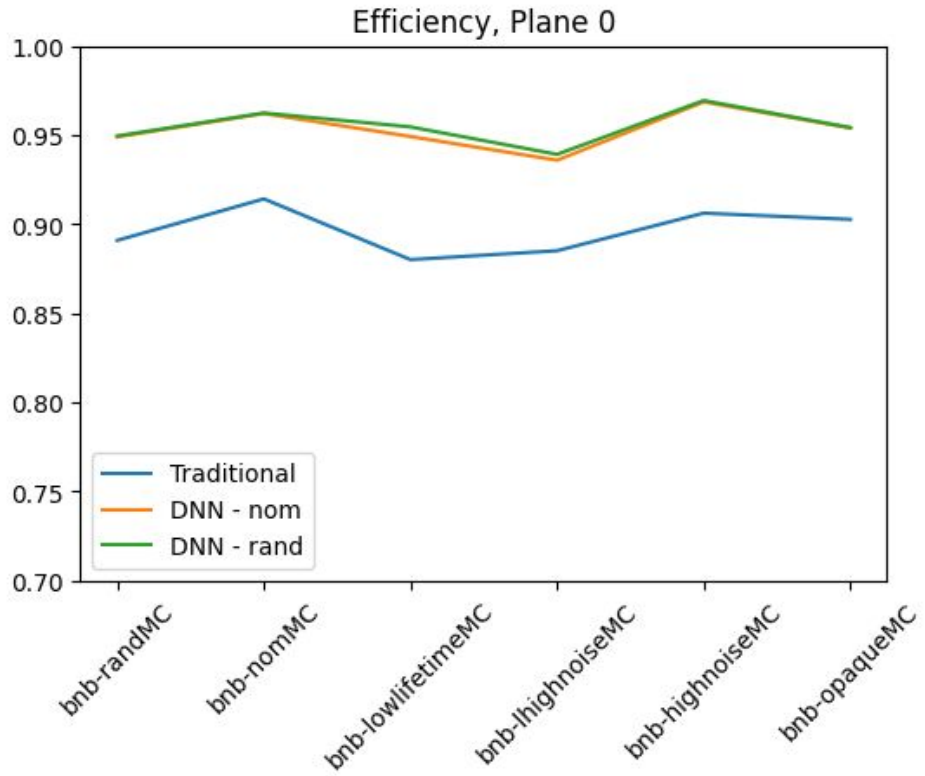


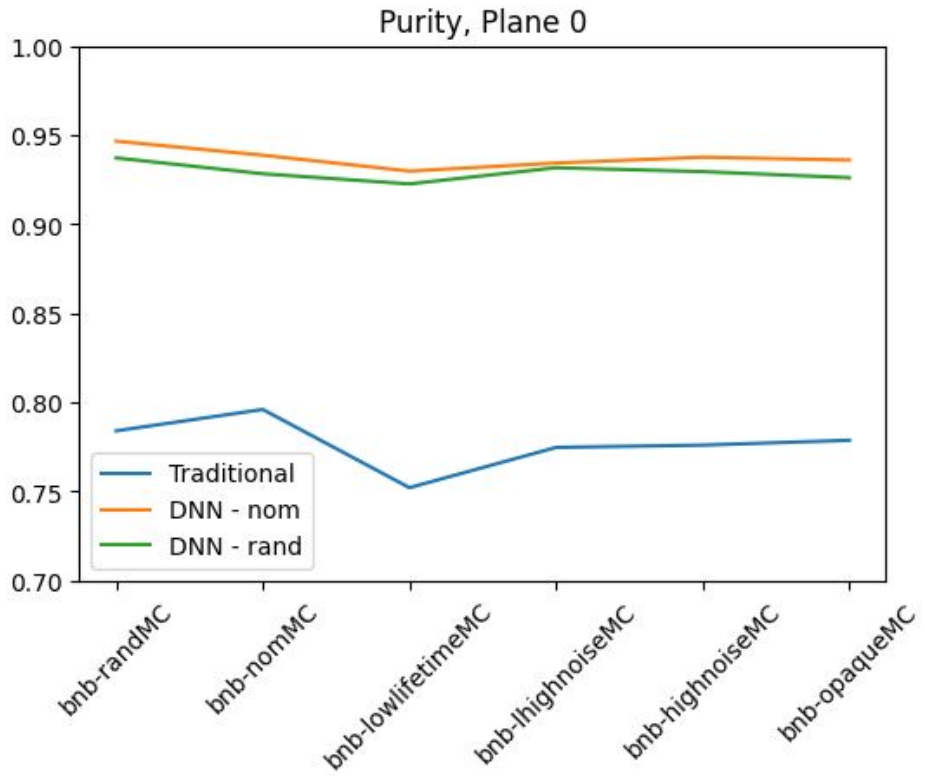




ICARUS: generate samples from larsoft detvar

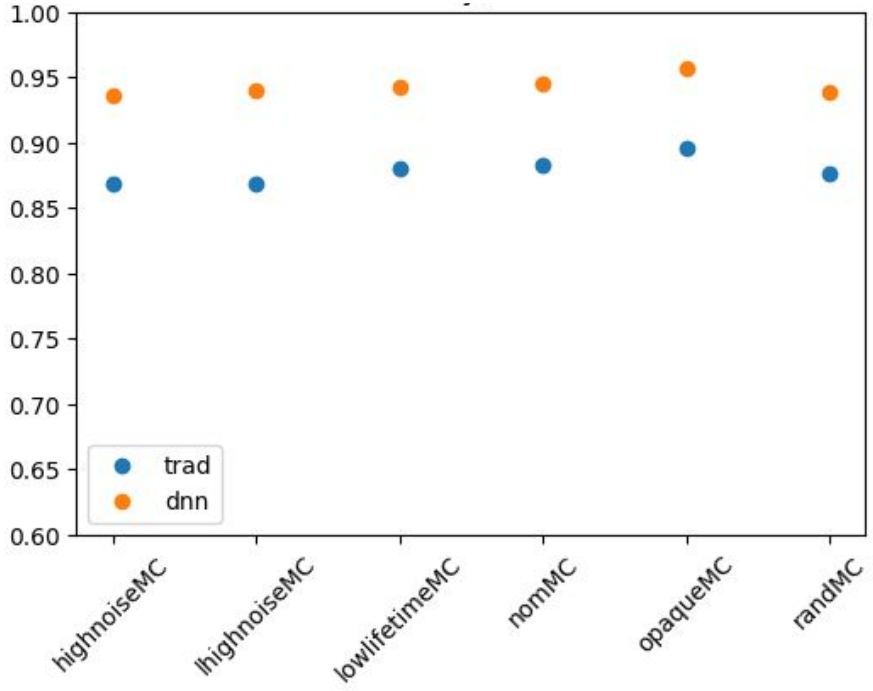
- variations: noise (2 levels), electron lifetime, wire transparency, random combinations of above
- no meaningful difference in trained network



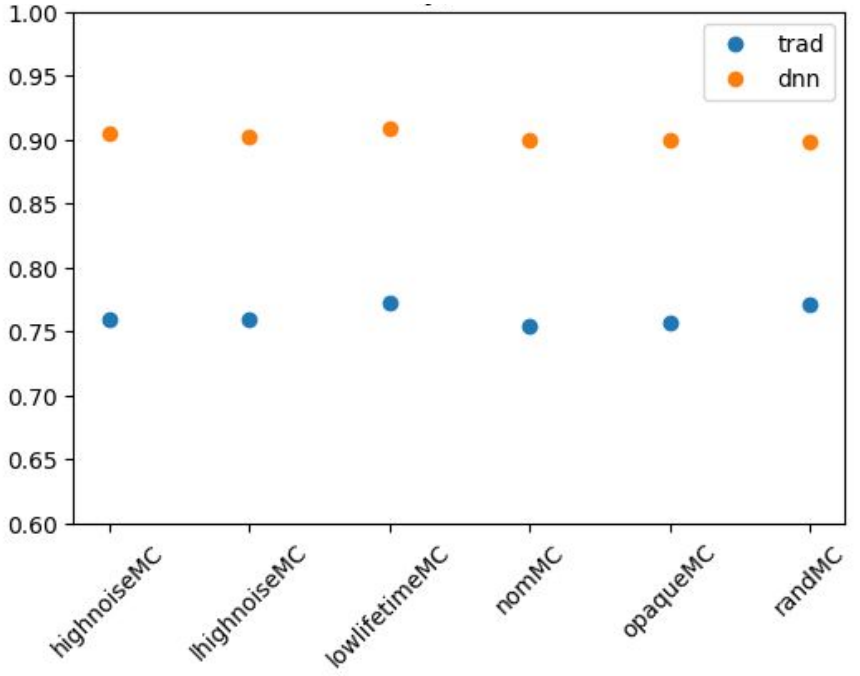


ICARUS evaluation on detvar samples

eff



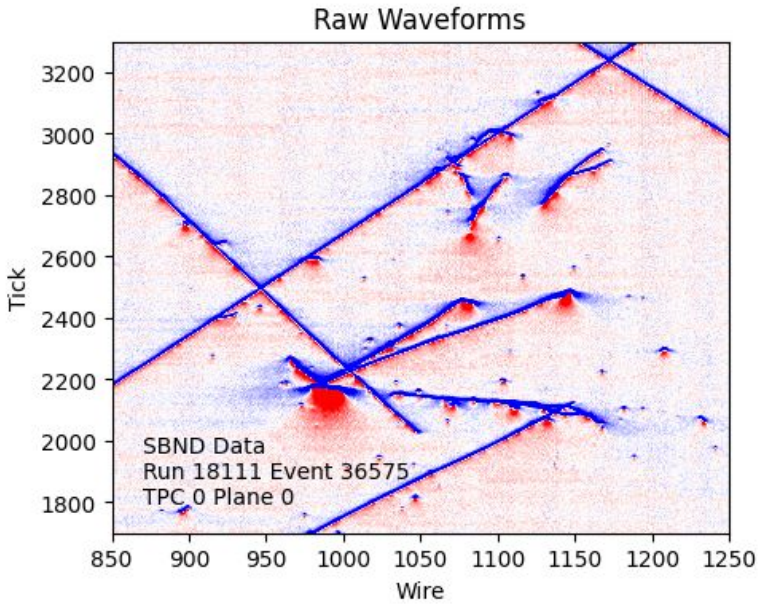
purity

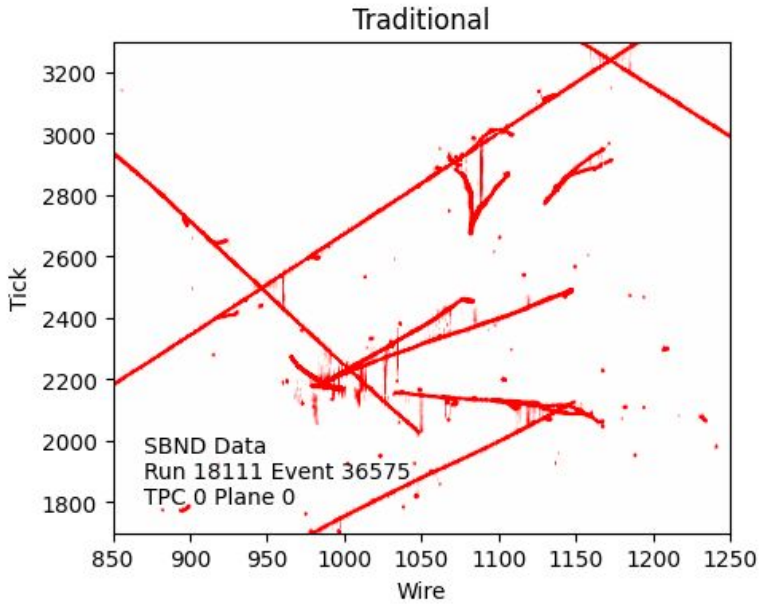


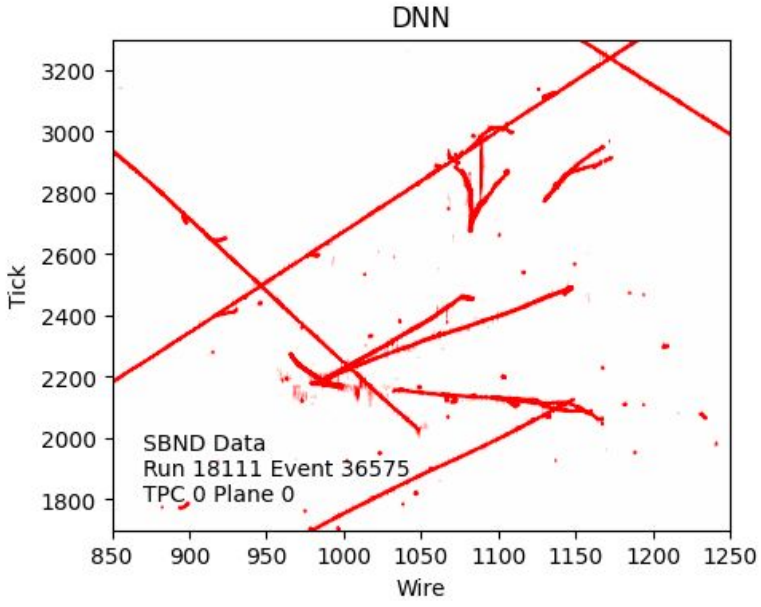
Validation on Data

Performance on SBND Data

DNN performs well on data out of the box!
*note that traditional SP was re-optimized after this

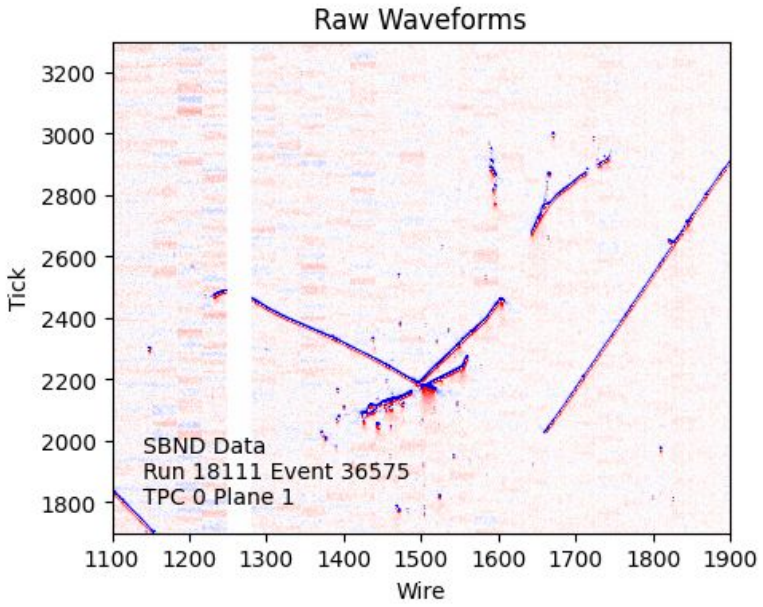


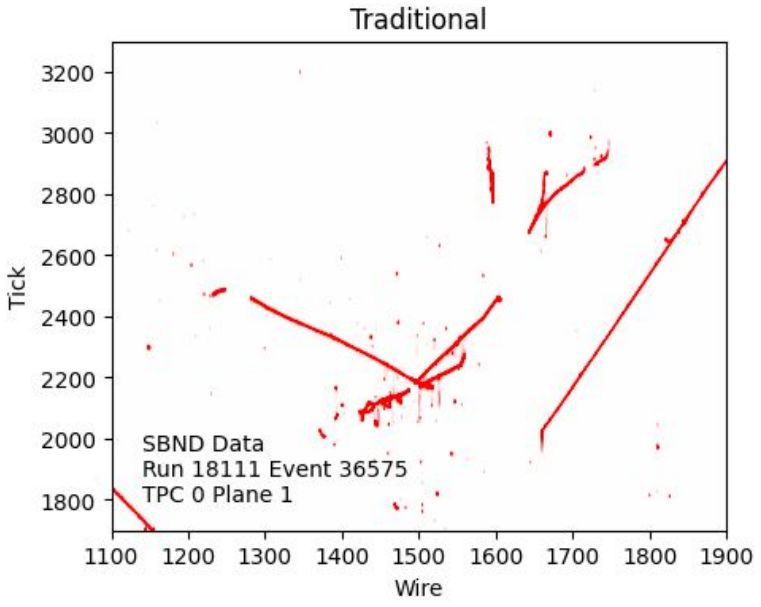


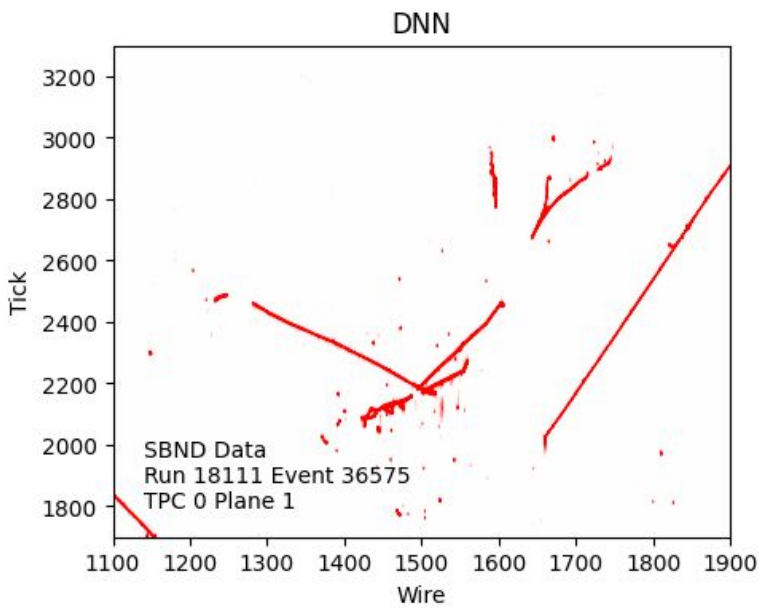


Performance on SBND Data

DNN performs well on data out of the box!
*note that traditional SP was re-optimized after this

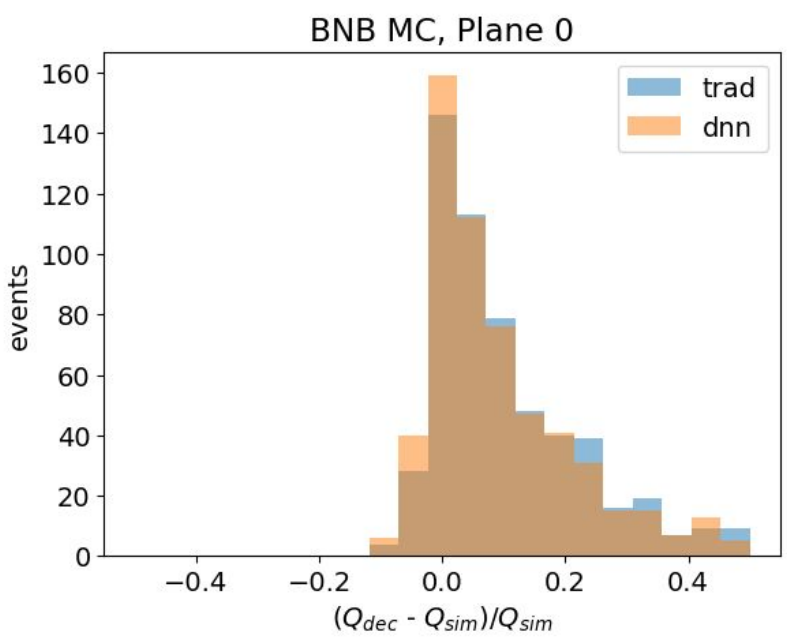


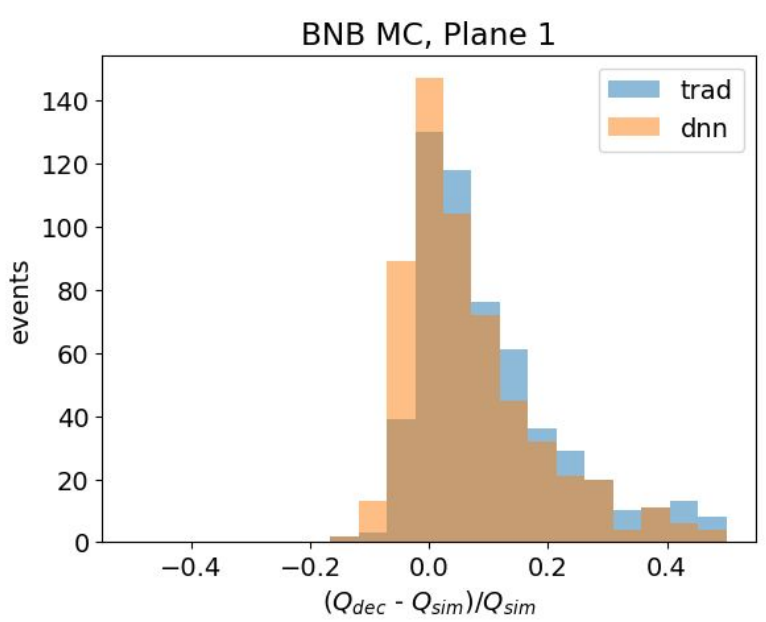


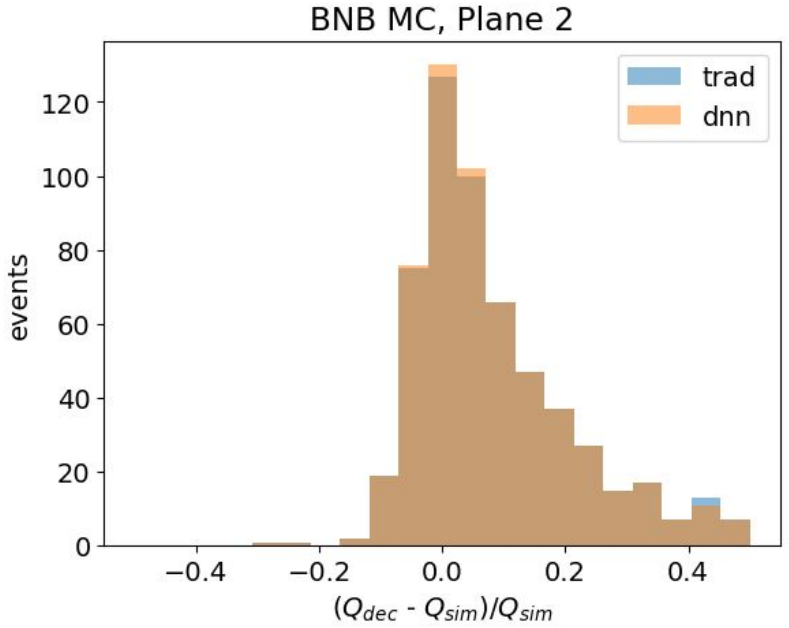


Charge reconstruction on plane

sum across all channels in one tpc for an event
plane 2 slightly different because of MP

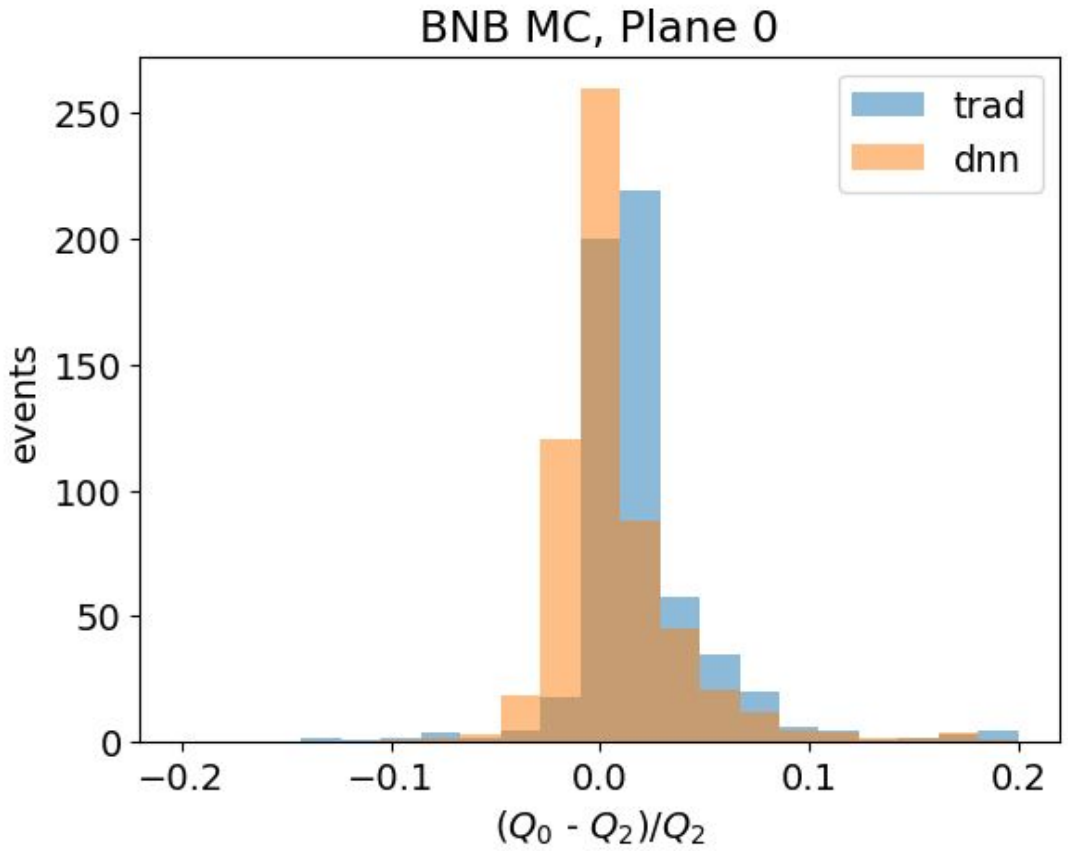


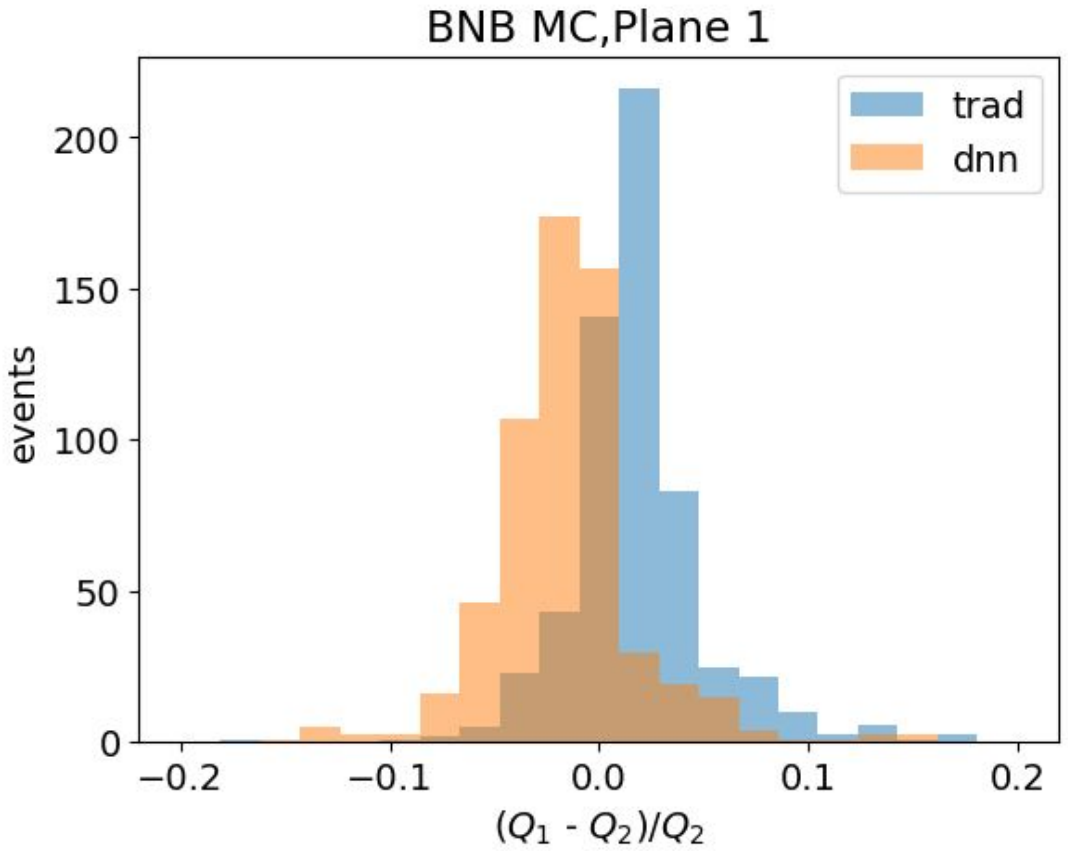




Charge consistency across planes

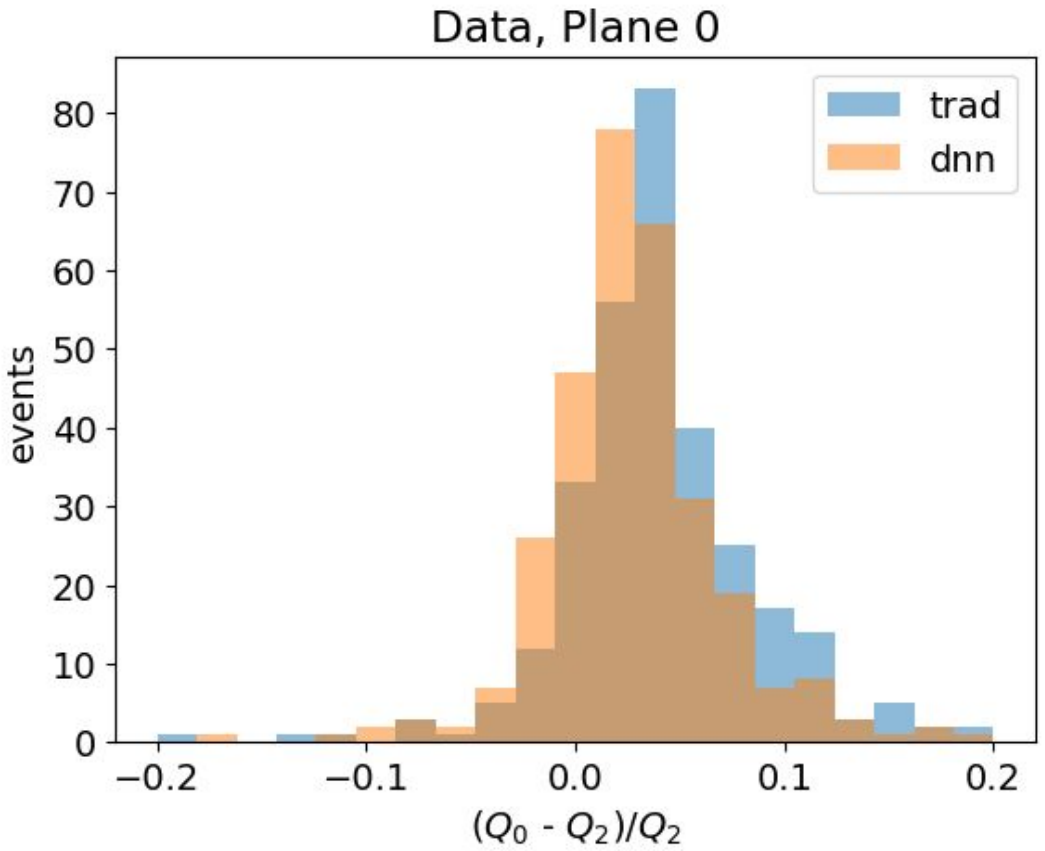
note that ROI finding is ~identical for collection plane

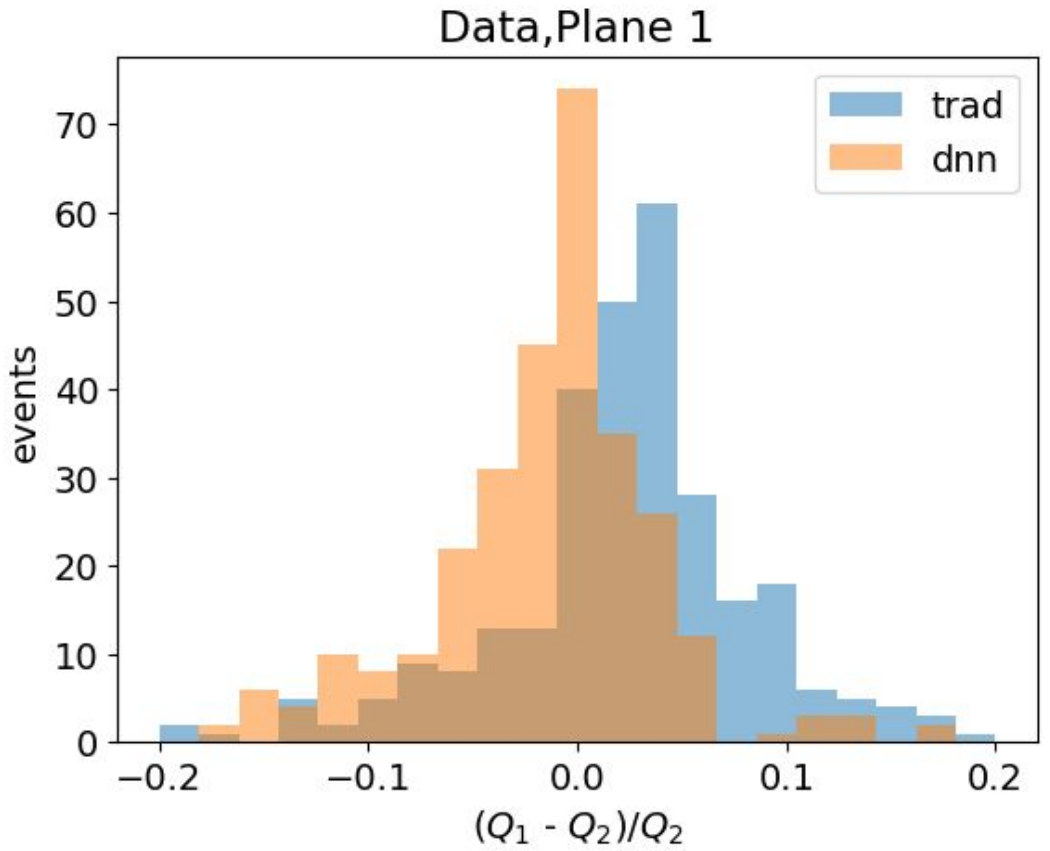




Charge consistency across planes

note that ROI finding is ~identical for collection plane

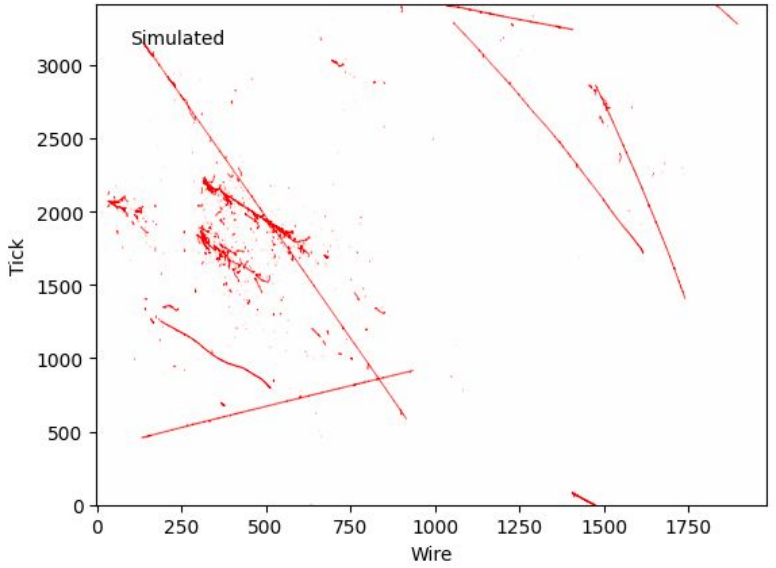


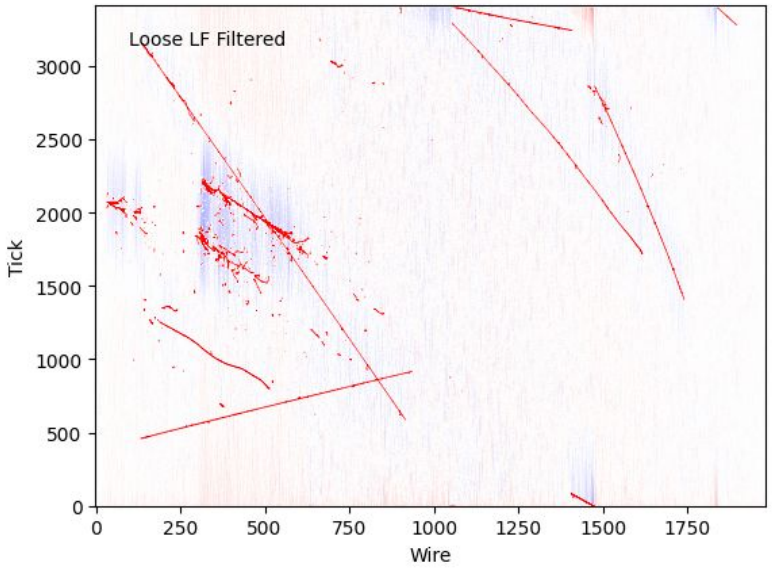


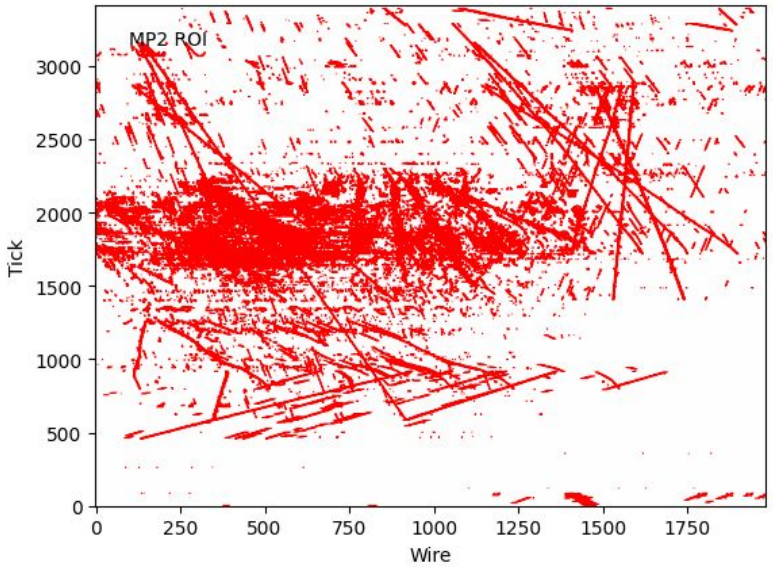
Further Improvements?

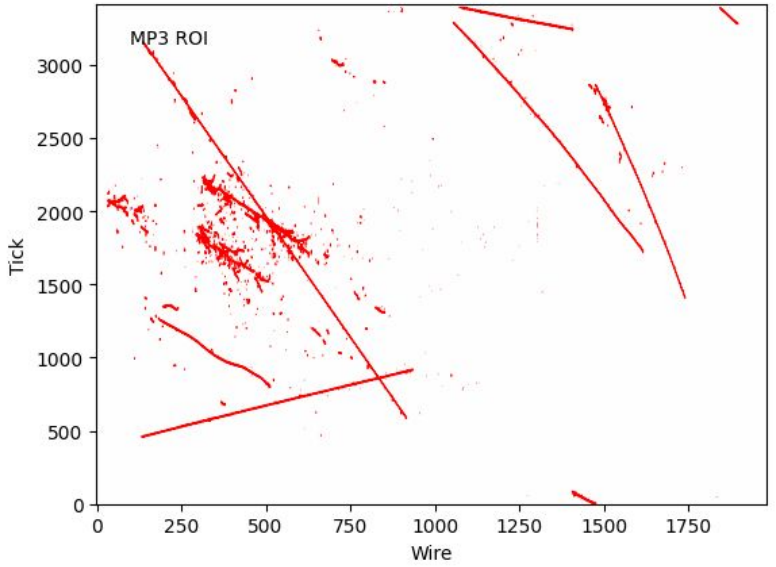
DNN-augmented ROI Finding

1. multi-plane matching: information across plane
2. 2D info as input images



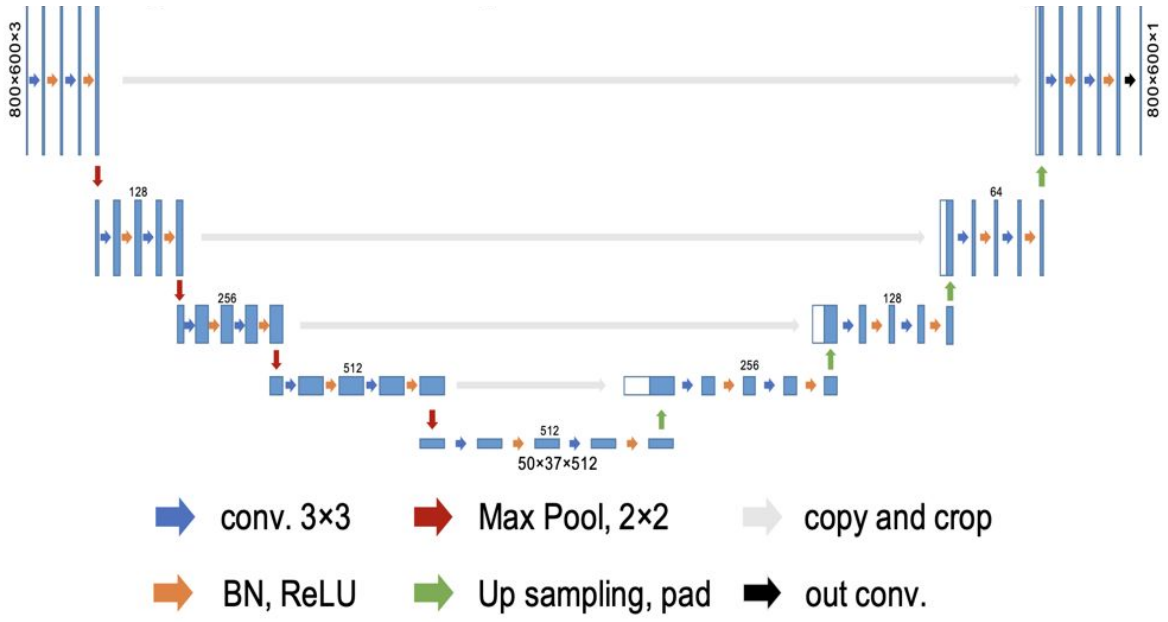






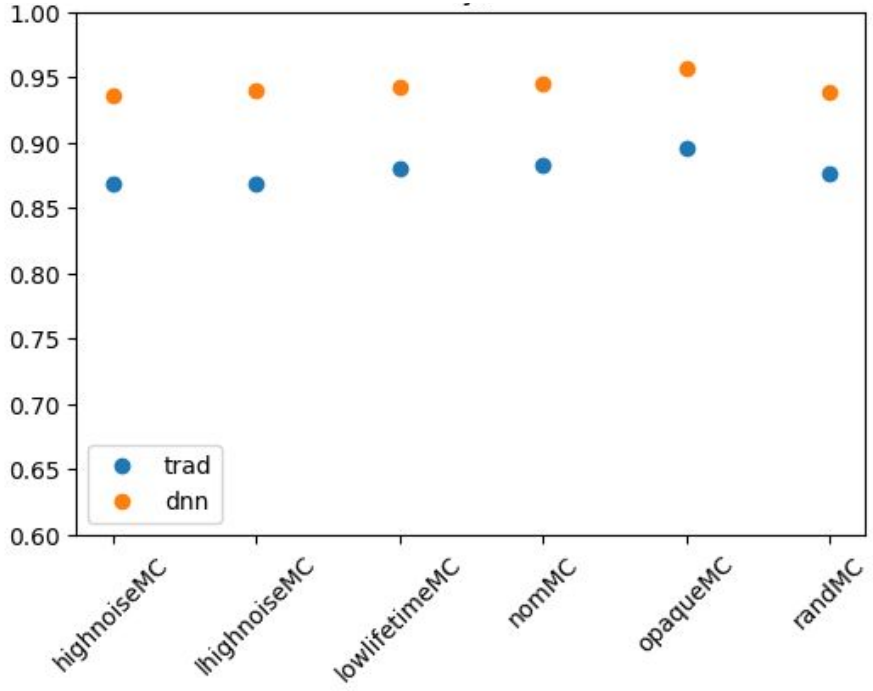
DNN-augmented ROI Finding

- network: UResNet
- output: ROI scores between 0 and 1, threshold for positive-ROI adjustable during inference
- apply weights when calculating loss function
  - ROI : non-ROI = 9:1

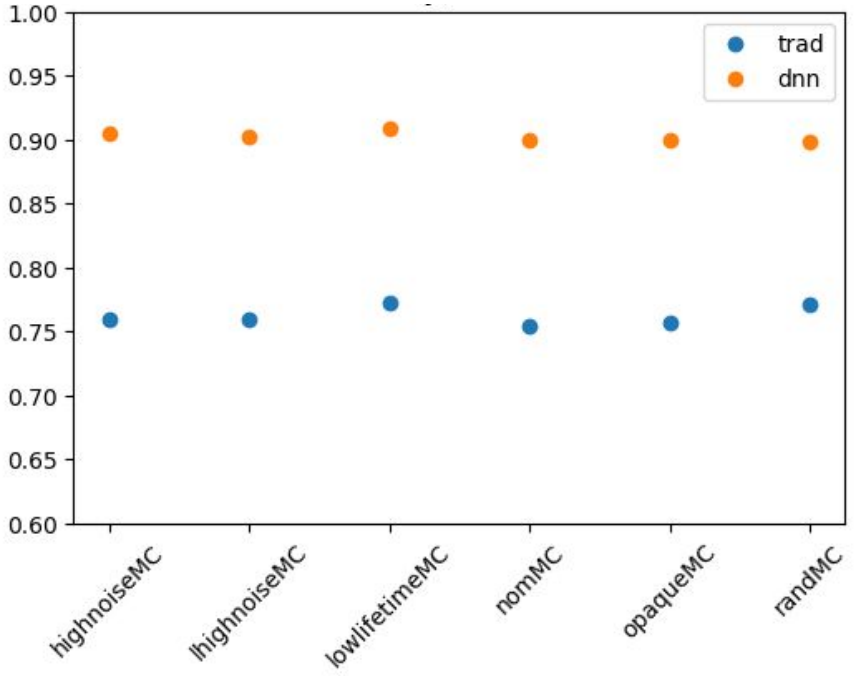


Efficiency vs. Purity - no pixel weight

eff

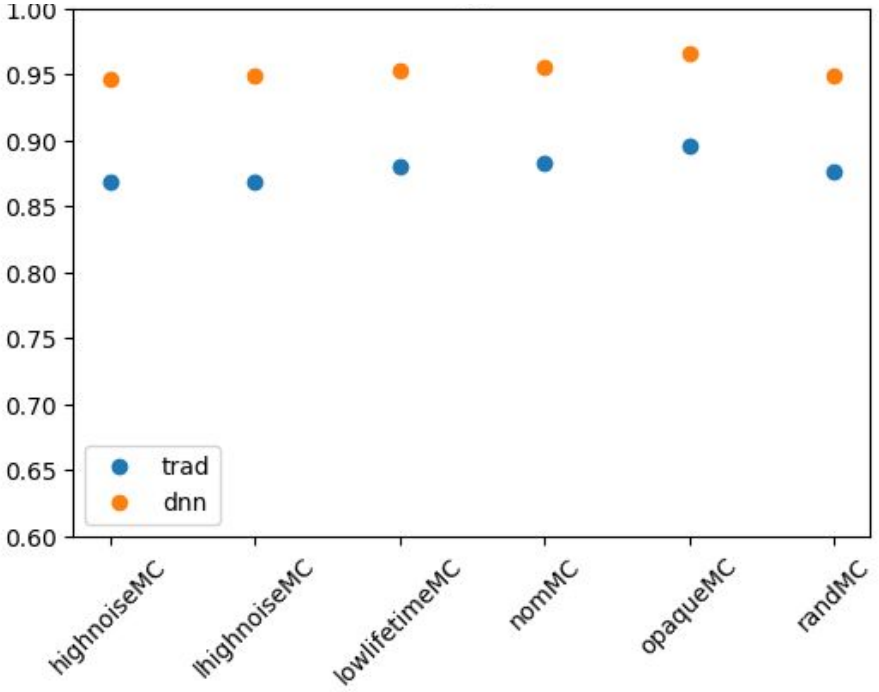


purity

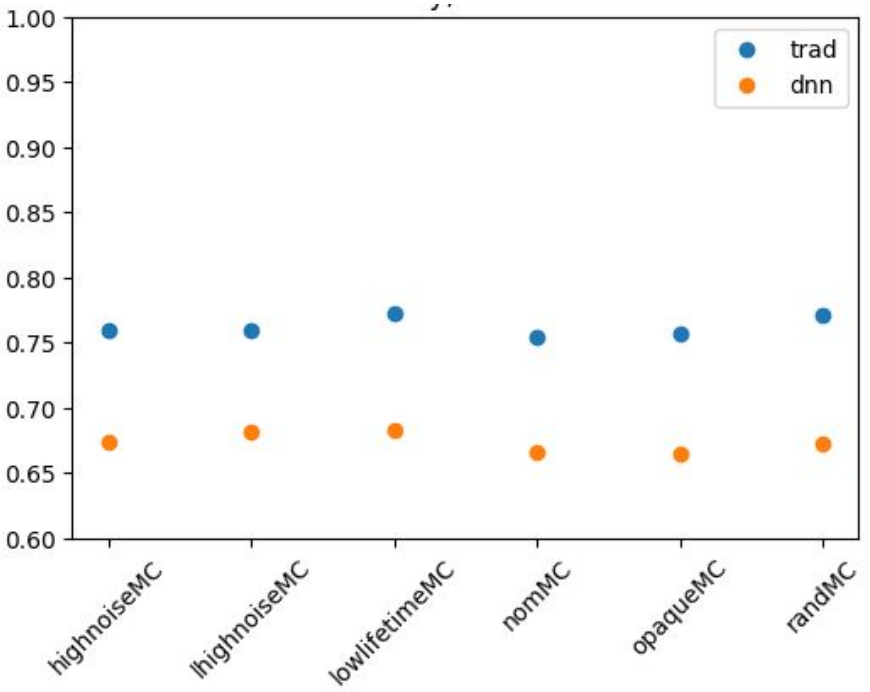


Efficiency vs. Purity - pixel weight 99:1

eff

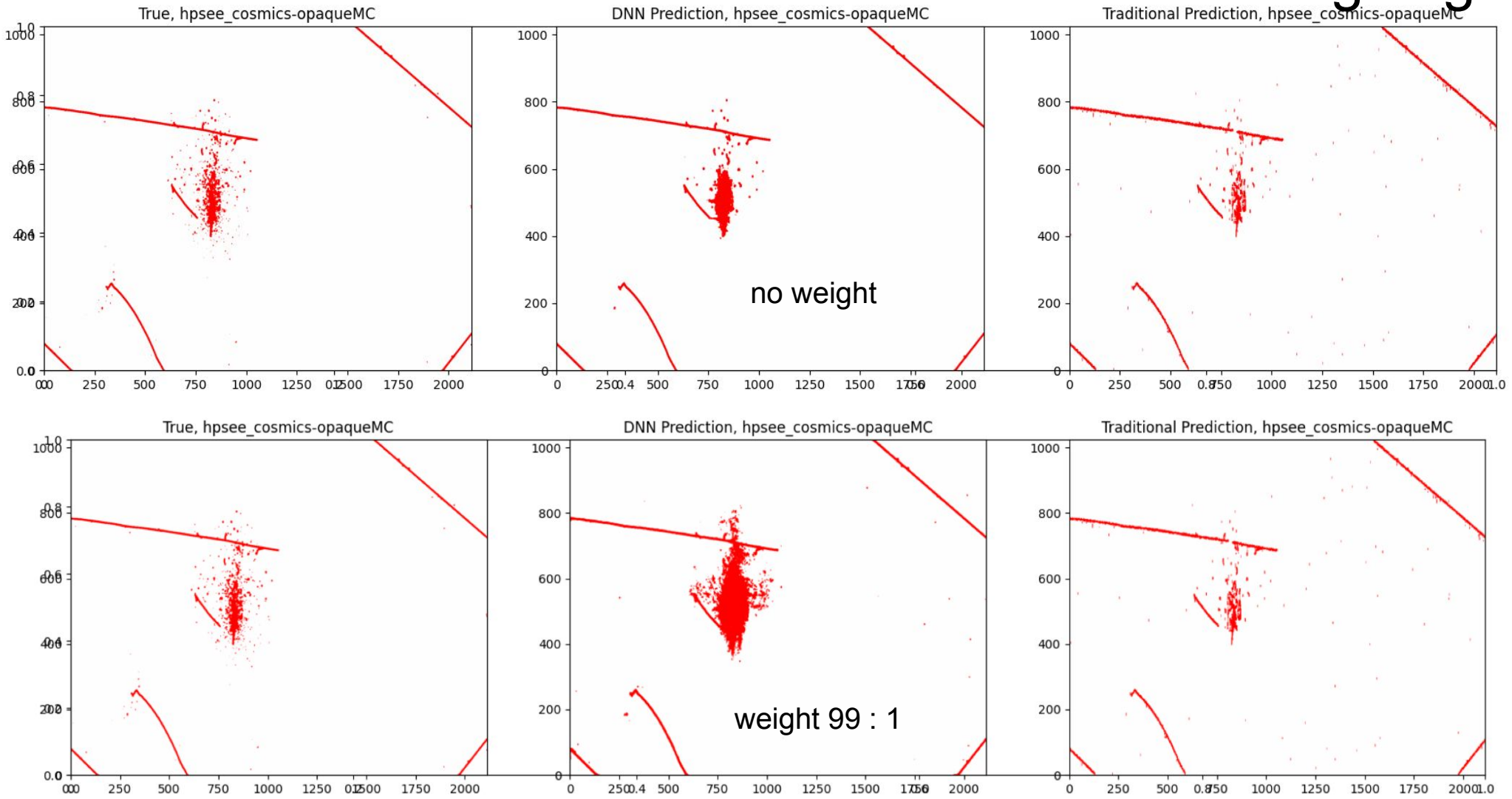


purity



Efficiency vs. Purity

what should we prefer? charge extraction studies ongoing…



Chunking

- This is effectively the same as using the network for smaller image

- no crazy edge effects observed
  - in retrospect this seems obvious – we don't train the NN to go crazy at the edge wires/ticks
- but need to retrain with input images with sizes of the chunks…