ToReROI

Tomographically Regularized Regions-of-Interest Finder

Brett Viren

June 30, 2025

Topics

- Introductory context and “rhetorical questions”.
- Approximate “big picture” for today’s SP=decon+ROI+DNNROI.
- Identification of problems with SP for SPNG to solve for PSP.
- Introduce a new (DNN) ROI scheme: ToReROI.
- Some “rhetorical questions” to explain some ToReROI details.
- Steps, issues and validations to realize ToReROI.

Context: DUNE Prompt Signal Processing (PSP) R&D

PSP kernel: “Signal Processing, Next Generation” (SPNG)

- Efficiency/purity must be at least as good as `OmnibusSigProc`.
  - But faster on less computing resources.
- Takes a **GPU-first** strategy based on `libtorch` tensors.

Status:

- Developing in Wire-Cell Toolkit’s `spng/` sub-package.
- 1st SP “decon” stage is rewritten and validated.

2nd ROI stage is trickier

- Strategy: replace the ROI heuristics with AI/ML.
- Rely only on DNNROI to do full ROI finding?
- Something new? (hint: yes, probably)

Why ROIs?

The deconconvolution (decon) stage amplifies noise

- Especially LF noise on induction planes

ROIs are a form of **time-domain adaptive filter** to suppress noise.

- Identify small regions around likely signal.
- Calculate new, local baseline and subtract.
- Zero everything outside of any ROIs.

Why DNNROI?

DNNROI increases efficiency for difficult tracks

- Prolonged tracks where bipolar responses leads to a cancellation in ADC waveforms.

DNNROI uses more information than conventional ROI finding:

- DNN in general can utilize marginal data that might otherwise lost due to discrete “cuts” in conventional algorithms.
- DNNROI in particular makes use of “multi-plane” information.

Why not something else?

We don't really care about ROIs, per se. We want to get rid of noise.

- Could apply DNN to denoise at ADC level.
  - Need care to retain the faint prolonged signals.
- Could apply DNN to denoise at decon level.
  - DNN can learn a better **adaptive filter**

However, this approach will alter the signal content in ways that are hard to understand (compared to simple spectra filters, HF, Gauss, etc).

- ROIs (DNN-made or otherwise) still alter signal but in ways that are more obvious, and thus easier to evaluate sim/real bias.

SP with DNNROI

DNNROI uses **depo flux splat** for “true ROIs”

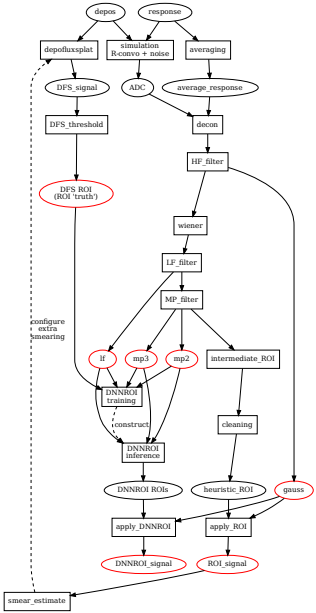
- Depos + “extra” Gausian smearing measured from conventional SP output.
- Leads to a “cycle” in the information flow.

DNNROI uses “multi-plane ROI protection/generation”

- mp2/mp3 images require RayGrid algorithms.
- RayGrid on libtorch possible, but performance is still unknown.

If DNNROI-only (no conventional ROI finding) works, we can consider using it in SPNG.

Caveat: this flow graph is conceptual and may not be 100% accurate.



Why mp2/mp3 is input to DNNROI?

Decon signal images with per-plane information leads to inefficiencies.

- Wiener (inspired) filter does maximize signal/noise.
- But, highly prolonged tracks still inefficient.

Project info from other planes to where it might be in a target plane given wire geometry.

- In principle, a DNN can learn this multi-plane correlation if given enough training data and if it sees all planes together.
- Feeding it mp2/mp3 lets it more easily take into account multi-plane info.
- Also helps to keep DNNROI per-plane so smaller images.

Can we do “better”?

Break the info cycle

The smearing of post-drift ionization electron distribution in conventional SP output is due to:

- Fine-response convolution in sim (and real) detector.
- Average response and HF + Gaussian filters in deconvolution.
- Some (small?) effect due to noise-related errors in ROI-based re-baseline.

Now, we measure the extra smearing. But, we know it, so we can simply apply it.

Avoid complexity in training and inference paths

DNNROI requires LF, mp2 and mp3 images.

- mp2/mp3 in particular requires accelerated RayGrid

But, the end goal is to exploit multi-plane info, and there are other ways.

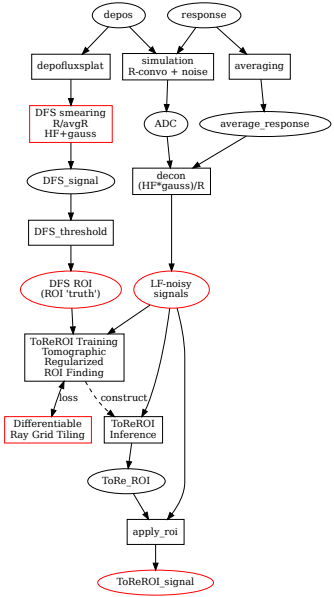
ToReROI workflow

Still use depo flux splat for “true ROIs”

- Smear using known kernels to break cycle

The DNN inputs images spanning all three planes.

- images are simply the decon output
- multi-plane info comes in through **regularization** loss terms.



Why Gauss for depo-flux-splat and decon?

Wiener (inspired) waveforms are more peaked than Gauss.

- Better for finding signal above noise.
- But we don't do conventional peak/ROI finding here.

Gauss preserves charge.

- And, produces peaks that are more broad than Wiener
- ROIs from Wiener would be too narrow.

True ROIs are calculated by **noise free** depo-flux-splat

- Better signal/noise is not relevant.
- DNN is given "all info" so will find peaks if they are there to be found.

Using Gauss filtered waveforms everywhere provides consistency.

What is ToReROI anyways?

Basic UNet segmentation task like DNNROI

- Label pixels as in/not-in ROIs given true ROIs.
- Plus a final ToRe layer

Tomographic regularization is fancy word for:

- Take output ROIs, run RayGrid Tiling (the “tomographic” part), determine value for regularization terms, add them to the loss.

The regularization “rewards” (or punishes) the network when it produces ROIs that are consistent (inconsistent) with the wire geometry and the input images.

- The network implicitly learns the wire geometry instead of having the implication of wire geometry (mp2/mp3) as input.
- RayGrid Tiling is not needed during inference.

Note, the network does not learn to do tiling. It learns to find ROIs that maximizes the ability for signal to be valid for later tiling.

Steps needed to realize ToReROI - smearing

Need “point spread function” for simulation and relevant parts of SP

- Can run sim + SP on point source and fit shape to 2D Gaussian
  - The “morse” test does this.
- But, should be able to construct correct filter a’priori given FR, ER, HF, Gauss, etc.
- Validate: DFS+smear = sim+SP?

Steps needed to realize ToReROI - training

- Test that the LLM transaction of RayGrid -> Python/PyTorch actually works.
- Convert/check that it is **differentiable**.
  - This is crucial for back-propagation to be used for proper training.
  - If not achievable there are some tricks we may try.
- Come up with loss function regularization terms based on “conserved” quantities.
  - Same total charge inside all ROIs compared to each of the three planes.
  - Total charge in ROIs compared to planes (but gently due to noise).
  - Strong metric with 3 plane consistency but allow for of only-2 as well.
- Implement training in `wirecell-dnn`.
- Make training data set and do the training.

Possible / likely issues

Image size

ToReROI inherently needs all 3 planes.

- DNNROI (on CPU) struggles with memory size, ToReROI will need more.
- But, we target GPU so maybe okay? And/or, follow DNNROI's path with "chunking".
- May consider different network architect from start.
- Note, we may take rebin-4 "for free" because of HF filter.

Regularization

Choosing regularization terms and weighting is half art.

- We must invent some quality metrics.
- And, be prepared to perform hyper-parameter optimization.

Validations

Sim, per-event

Make sample of sim and: decon+ROI, decon+ROI+DNNROI and decon+ToReROI.

- Check detailed results, make pixel level image "diffs"
- Check "a few" events with variety of track directions

Sim, statistical

Run the "spdir" job to get bias, efficiency, resolution vs angle.

Real data

Repeat "sim, per-event" validations but using real data.