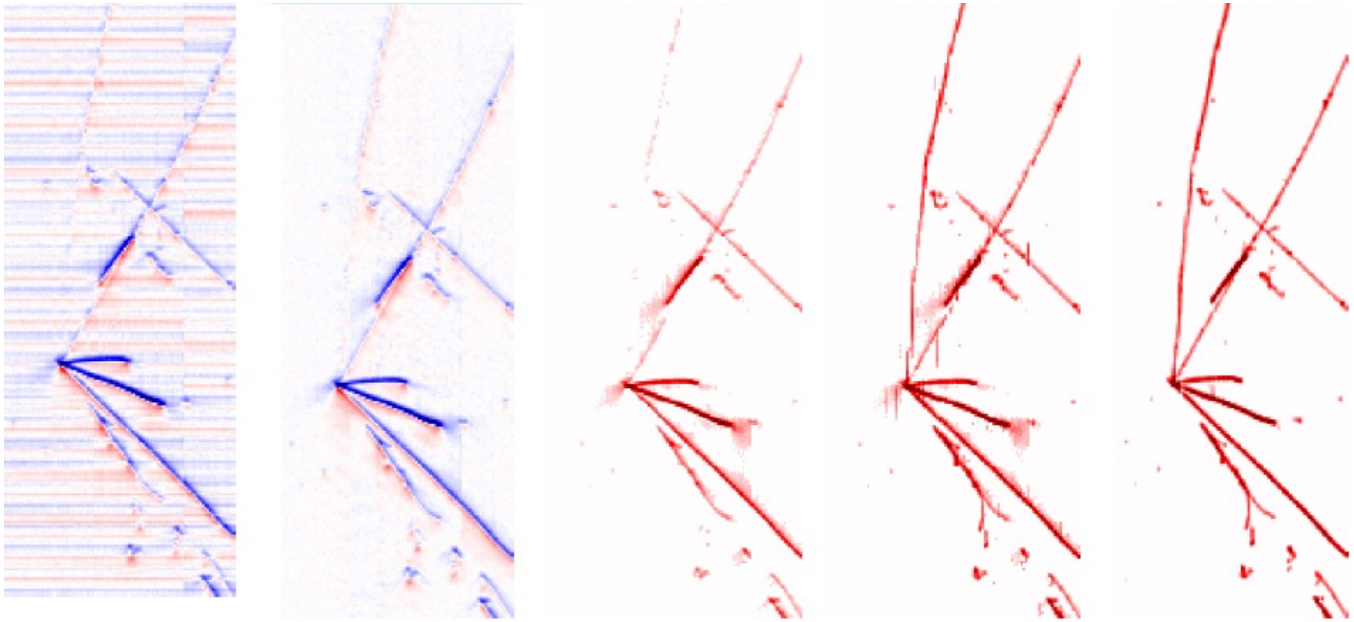
LArTPC Simulation & Signal Processing in Wire-Cell

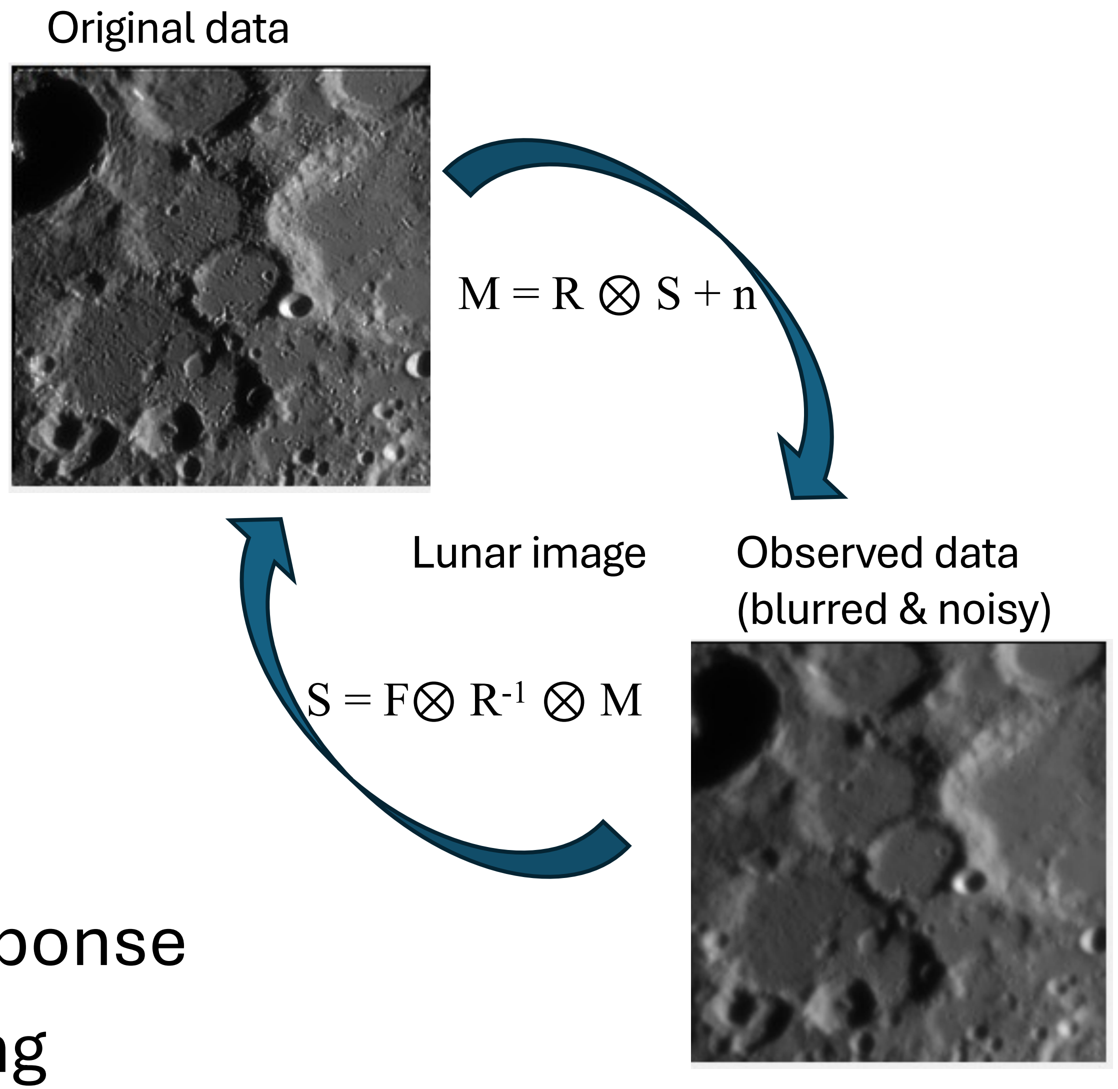


Wenqiang Gu

Brookhaven National Laboratory

Outline

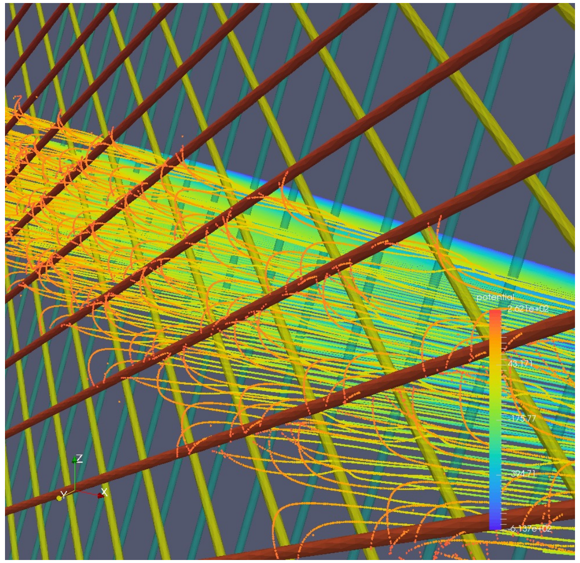
- Digital signal processing widely used in image measurements and analyses such as medical imaging, astronomy imaging, etc
- For high-energy physics application, a realistic Monte Carlo simulation (e.g. detector response) is crucial



Today’s talk will cover

- Modeling of LArTPC ionization response
- Basic principle of signal processing
  - Noise filtering, deconvolution, signal region-of-interest (ROI)

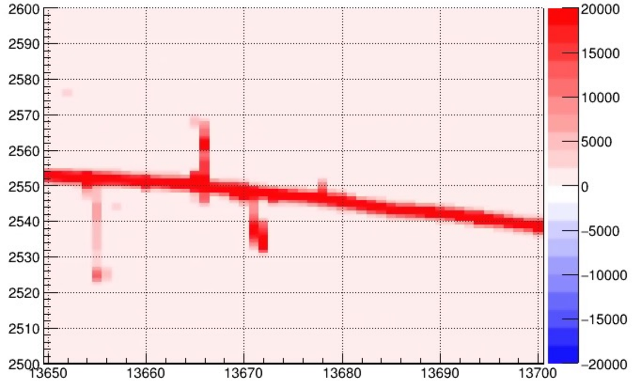
2D-Convolution based LArTPC Simulation



- LArTPC wire-readout measures:
  ionized charge ⊗ response

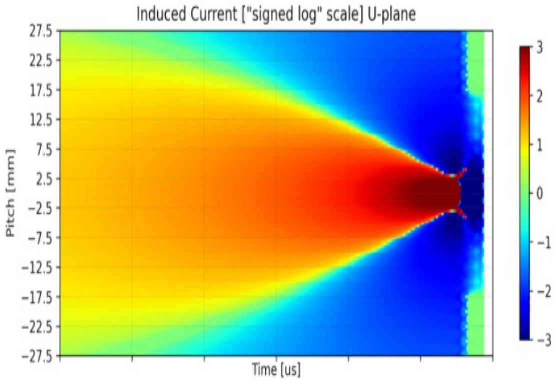
$$M(t',x') = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} R(t,t',x,x')\cdot S(t,x)dtdx + N(t',x')$$

Energy depo + diffusion + rasterization



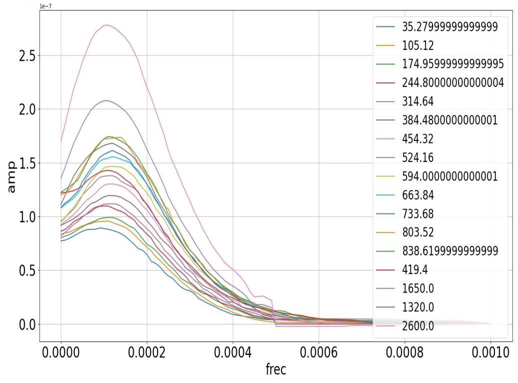
⊗

Long-range and position-dependent field response



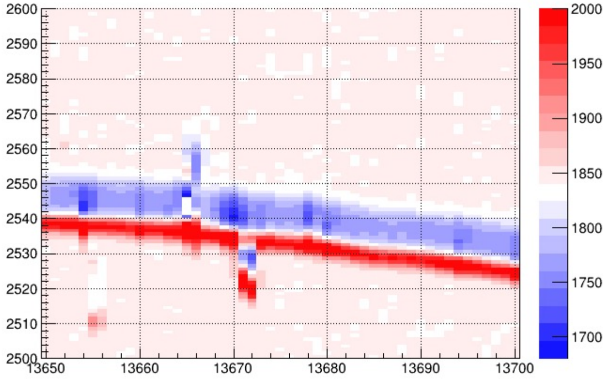
⊕

Noise Spectrum

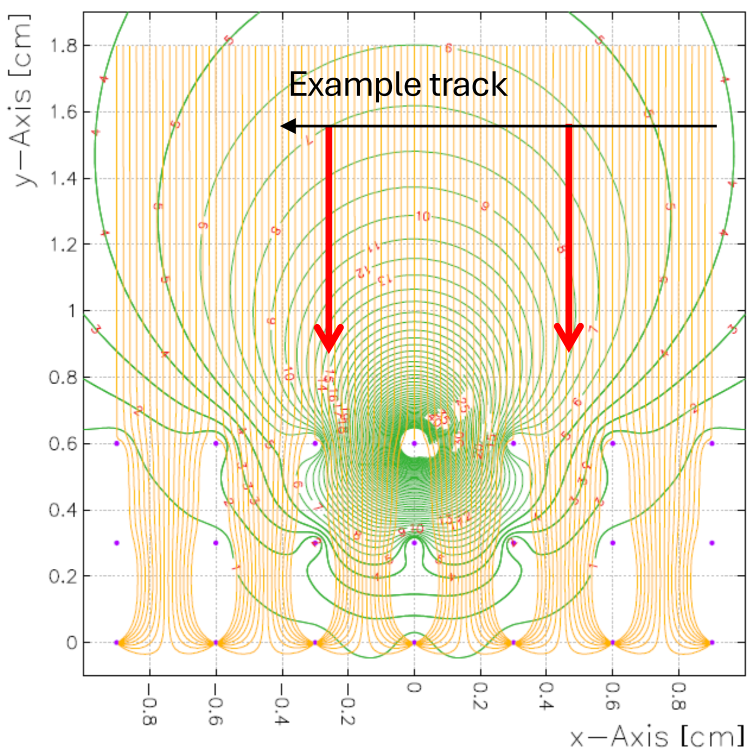


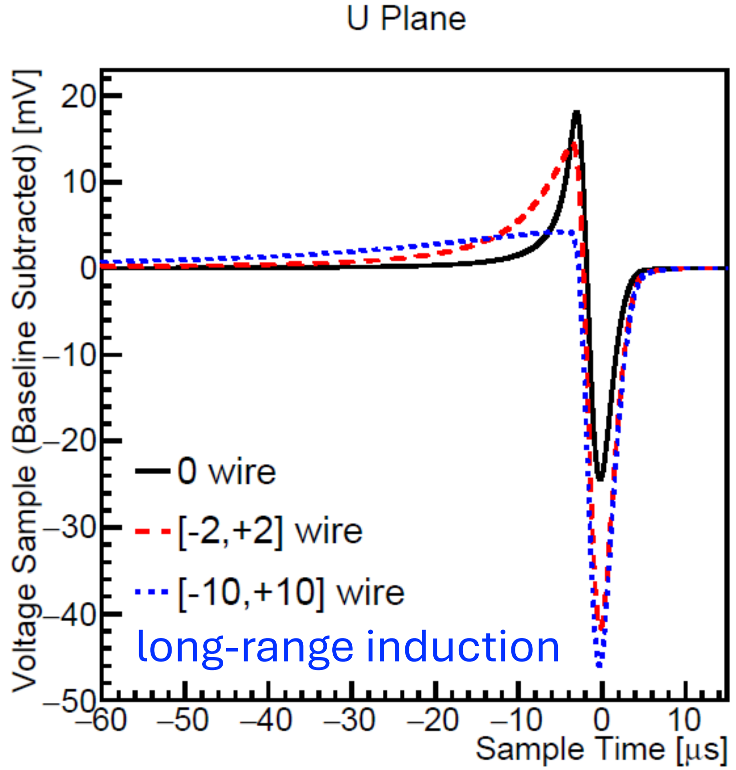
⟶

Final Signal



Single-Phase TPC Signal Formation





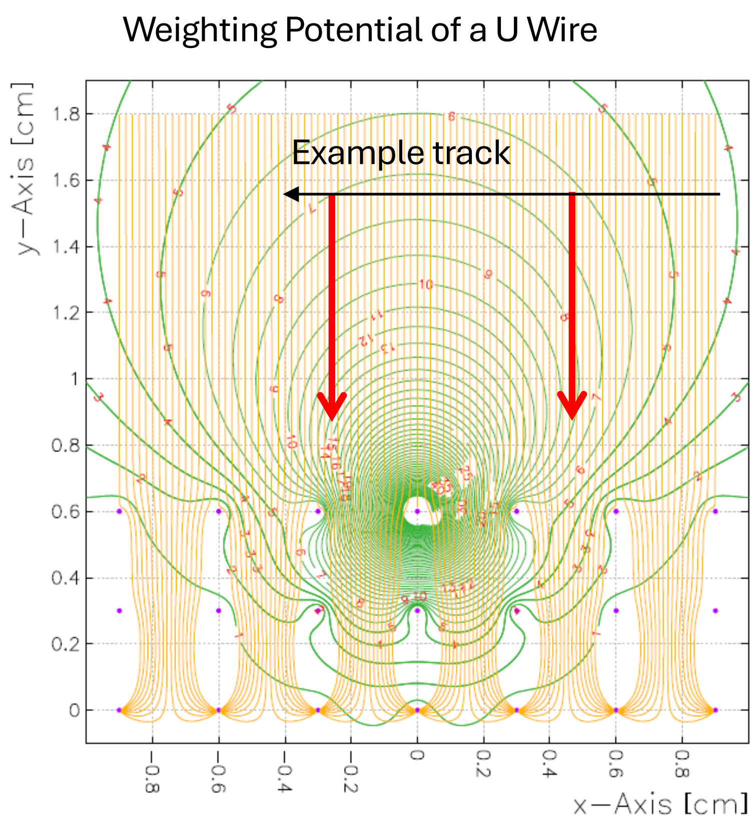
Ramo theorem

$$i = -q \cdot \vec{E}_w \cdot \vec{v}_q$$

v_q: velocity
E_w: weighting field
q: charge

- Induction plane signal strongly depends on the local charge distribution, collection plane signal is much simpler

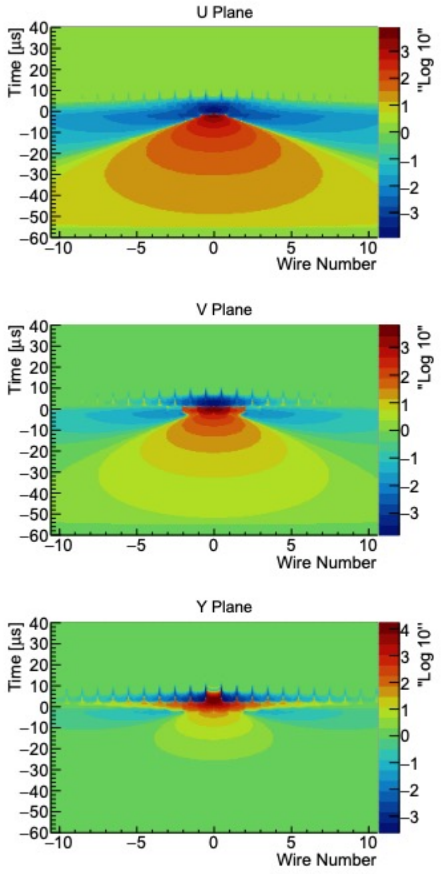
Field Response Model: 1D vs. 2D



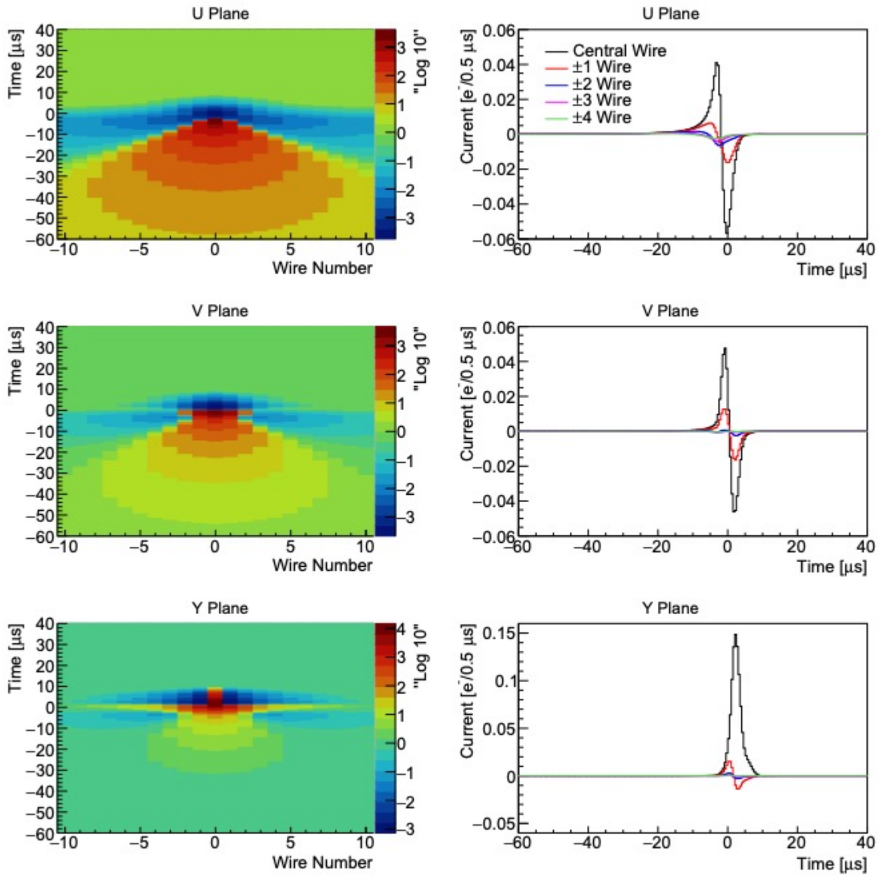
- 1D response model
  - Depends on only 1D coordinate (drift direction)
    - Sim and SigProc assume current only in the wire nearest to drifting electrons
  - **Pros:** computationally fast and algorithmically easy
  - **Cons:** long-range induction effect cannot be ignored
- 2D response model
  - Depends on 2D coordinates (drift + pitch directions)
  - **Pros:** works well on some **non-2D** geometries (e.g., wires)
  - **Cons:**
    - Calculation more difficult than 1D, but reasonable (GARFIELD)
    - Sim & sigproc algorithm more complex, slower than 1D
    - Imperfect for more complicated **3D geometry** (e.g., strips + holes)

Wire-Cell 2D Response for MicroBooNE

- Response model for simulation: drift vs impact position

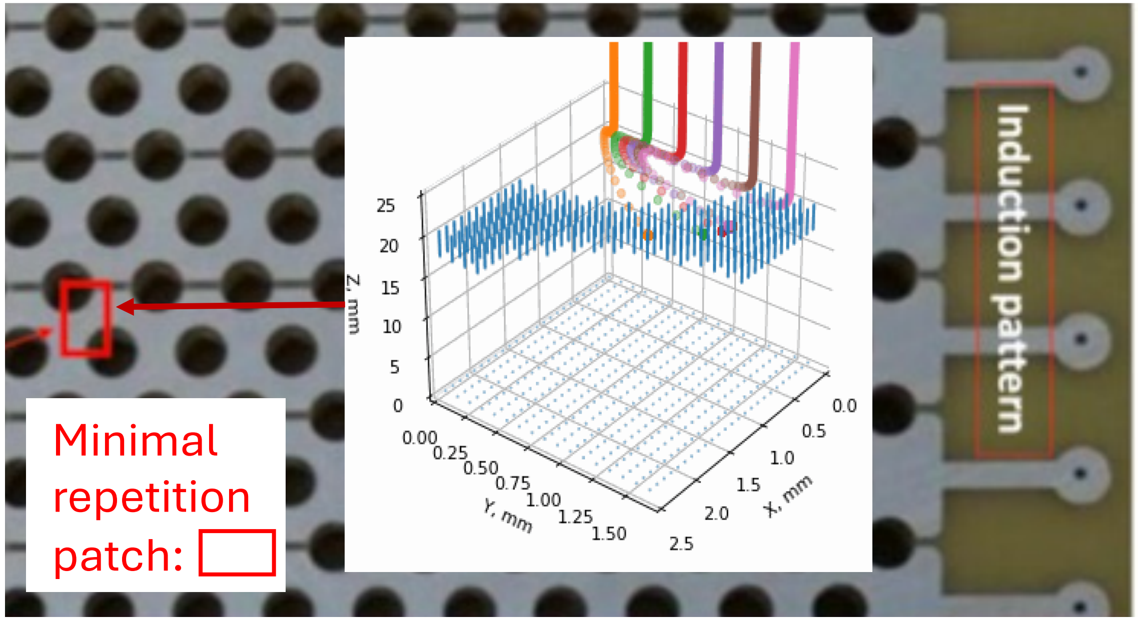


- Response model for sigproc: drift vs wire position & central wire



"2.5-D" Response Model

- Recent LArTPC designs utilize electrodes formed on printed circuit board (PCB) in the shape of **strips with through holes**
- The holes break the approximate **translational symmetry** as in wire-based LArTPCs



- Full 3D model is computationally expensive, instead 3D (near electrodes) + 2D (far field) is faster and precise
- SigProc assumes translational symmetry, i.e. averaged 2D response model
  - 3D Sigproc is conceivable, but it would be an iterative way

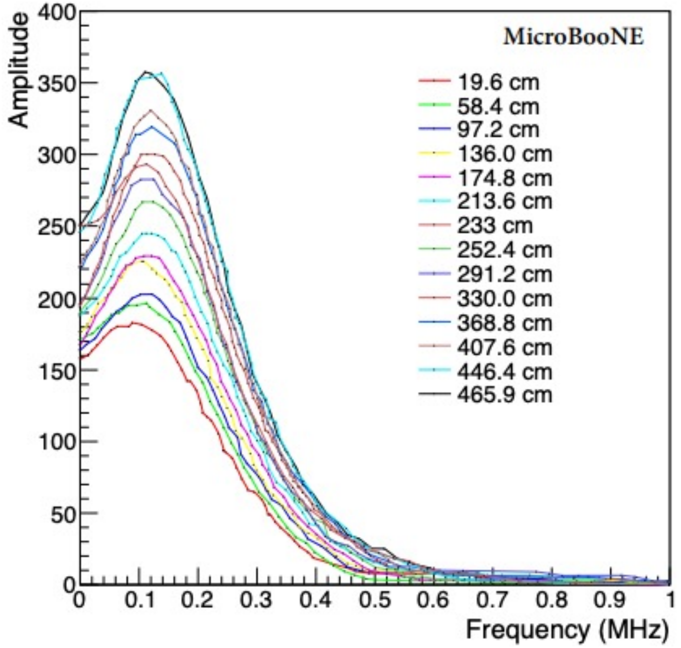
Noise Model

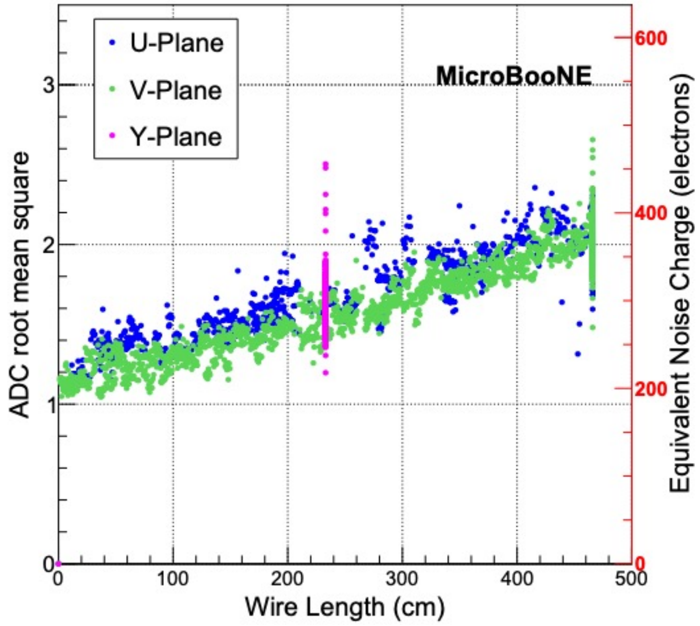
- The stochastic behavior of noise is analytically simulated in the frequency space
- A 2-D random walk process in amplitude and phase: Rayleigh distribution

$$F(\omega) = r(\omega) \cdot e^{-i\alpha_\omega},$$

$r(\omega)$: $R(r;\sigma) = \frac{r}{\sigma^2} e^{-\frac{r^2}{2\sigma^2}},$

$\alpha(\omega)$: $[0, 2\pi)$



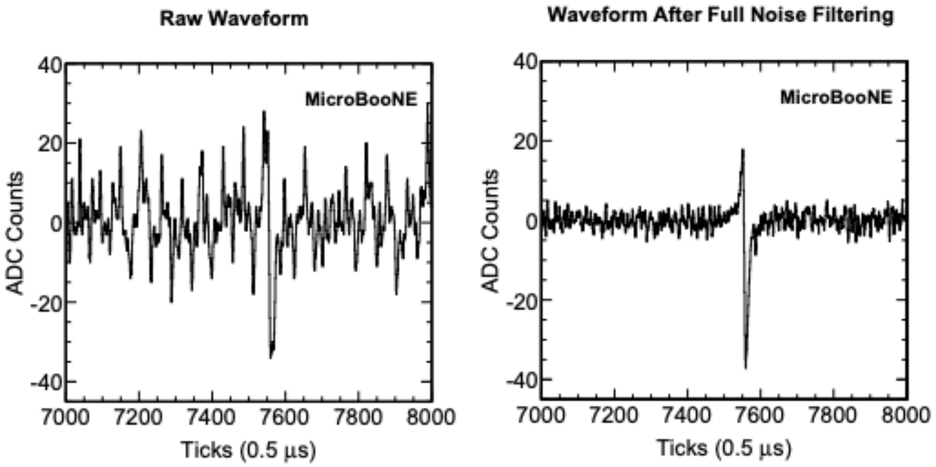


Signal Processing

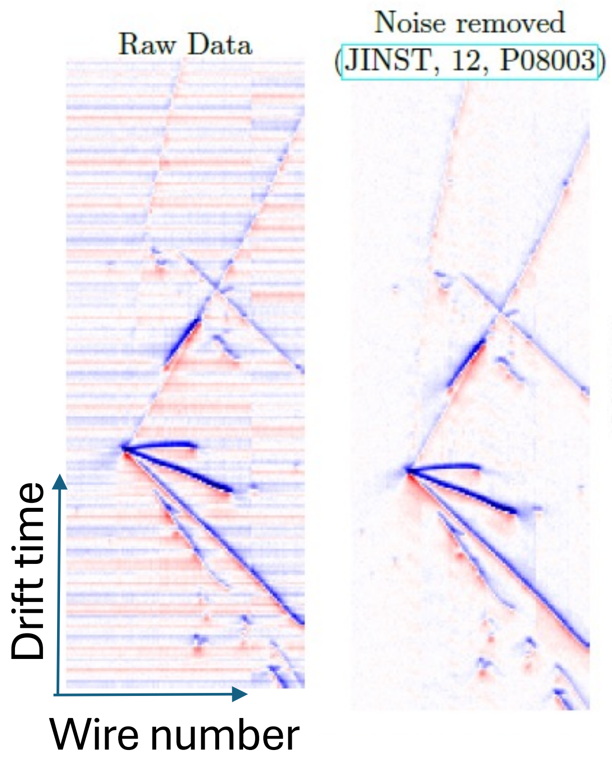
JINST **12** P08003

Noise Filter

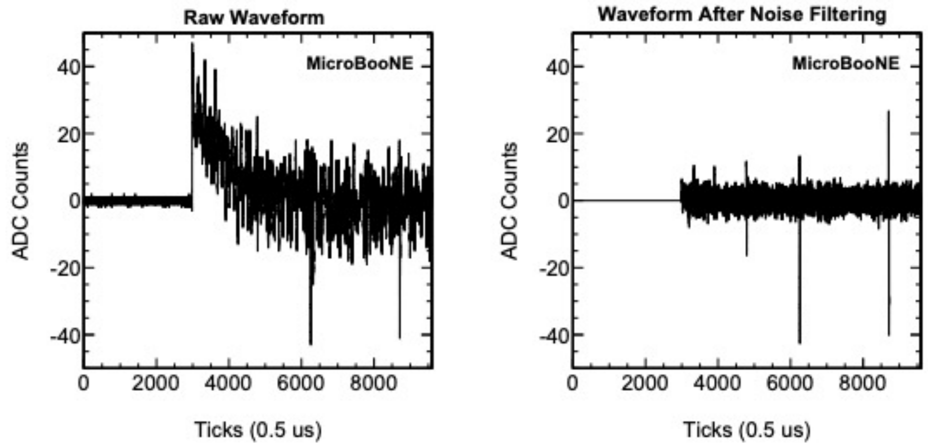
- Noise excess/hardware malfunction can be filtered/fixed before charge deconvolution, e.g.,



Harmonic Noise (36kHz and 108kHz)

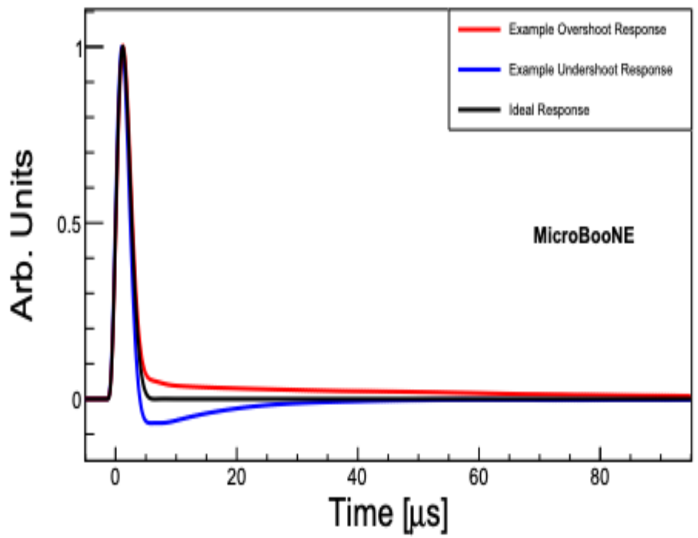


Coherent Noise



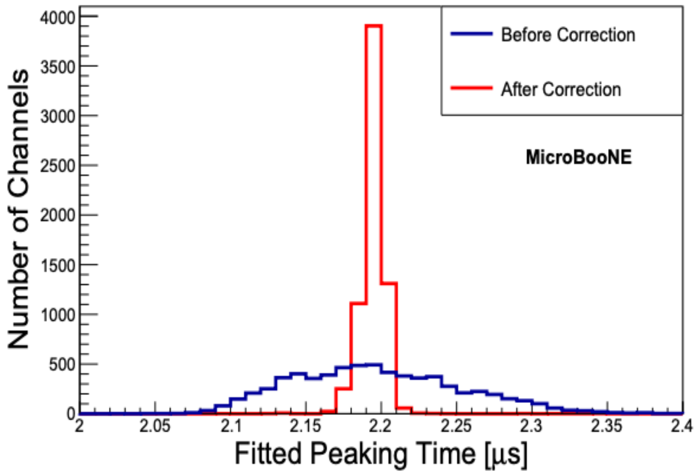
ASIC Saturation

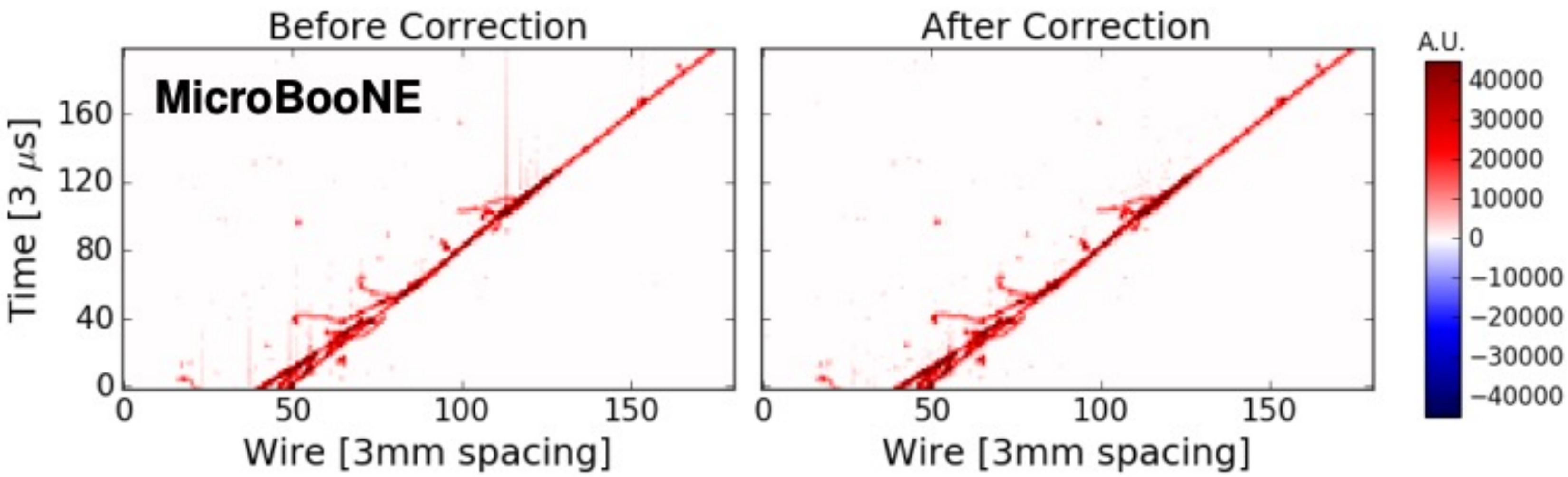
Calibration of Electronics Response Function



$$M_i^{corr}(\omega) = M_i(\omega) \cdot \frac{R_{ideal}(\omega)}{R_i(\omega)},$$

- Non-ideal electronics response function can be corrected channel-by-channel in the frequency space





Basics of (1D) Signal Processing

$$M(t_0) = \int_t R(t - t_0) \cdot S(t) \cdot dt$$

↓ **Fourier transform**

$$M(\omega) = R(\omega) \cdot S(\omega)$$

↓ **Deconvolution + Filter**

$$S(\omega) = \frac{M(\omega)}{R(\omega)} \cdot F(\omega)$$

↓ **Inverse Fourier transform**

$$S(t)$$

- Principal method to extract wire charge S(t) is deconvolution
- By given a response function R(t), signal S(t) can be easily derived via Fourier transform
- A filter function $F(\omega)$ introduced to suppress fluctuation after deconvolution
  - $O(N^3)$ matrix inversion achieved through a O(N logN) fast Fourier transformation: top 10 algorithms in 20^{th} century

Basics of (1D) Signal Processing

$$M(t_0) = \int_t R(t - t_0) \cdot S(t) \cdot dt$$

Fourier transform

$$M(\omega) = R(\omega) \cdot S(\omega)$$

Deconvolution + Filter

$$S(\omega) = \frac{M(\omega)}{R(\omega)} \cdot F(\omega)$$

Inverse Fourier transform

$$S(t)$$

- Without a filter function, the deconvolution process is equivalent to the matrix inverse problem

$$S = R^{-1} \cdot M$$

Basics of (1D) Signal Processing

$$\mathrm{M}(t_0) = \int_t R(t - t_0) \cdot S(t) \cdot dt$$

Fourier transform

$$M(\omega) = \mathrm{R}(\omega) \cdot \mathrm{S}(\omega)$$

Deconvolution + Filter

$$\mathrm{S}(\omega) = \frac{\mathrm{M}(\omega)}{\mathrm{R}(\omega)} \cdot F(\omega)$$

Inverse Fourier transform

$$\mathrm{S(t)}$$

- A Filter function is equivalent to a regularization in a χ^2 minimization problem

$$\chi^2 = \sum_i \left(M_i - \sum_j R_{ij} \cdot S_j \right)^2 + \sum_i \left(\sum_j F_{ij} \cdot S_j \right)^2$$

2nd derivative penalty

$$\frac{\partial \chi^2}{\partial S_k} = 0 \rightarrow \sum_i -2 \cdot \left(M_i - \sum_j R_{ij} \cdot S_j \right) \cdot R_{ik} + 2 \cdot \left(\sum_j F_{ij} \cdot S_j \right) \cdot F_{ik} = 0$$

$$\mathrm{R} \cdot \mathrm{M} = \left(R^2 + F^2 \right) \cdot S$$

$$\mathrm{S} = (1 + \frac{F^2}{R^2})^{-1} R^{-1} M \qquad \mathrm{A} = (1 + \frac{F^2}{R^2})^{-1}$$

Filter function

Basics of (1D) Signal Processing

$$M(t_0) = \int_t R(t - t_0) \cdot S(t) \cdot dt$$

↓ **Fourier transform**

$$M(\omega) = R(\omega) \cdot S(\omega)$$

↓ **Deconvolution + Filter**

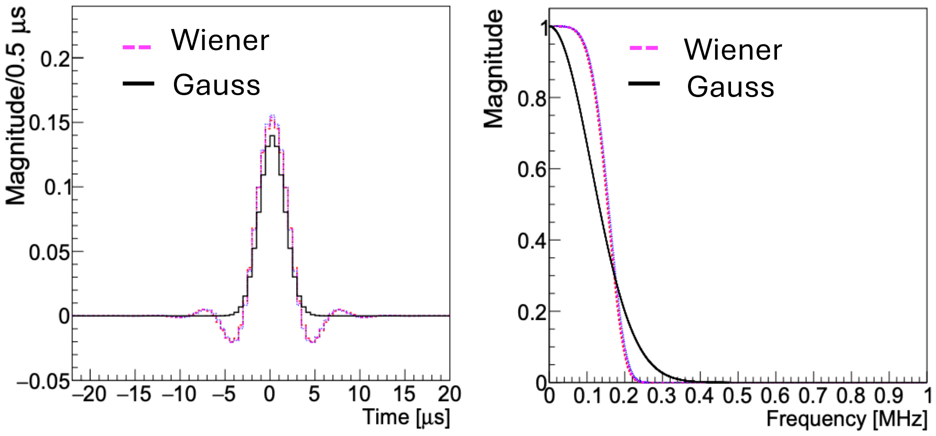
$$S(\omega) = \frac{M(\omega)}{R(\omega)} \cdot F(\omega)$$

↓ **Inverse Fourier transform**

$$S(t)$$

- Typical filters in signal processing are low-pass filters
  - Gaussian filter: smoothness
  - Winer filter: minimal mean square error

"Data Unfolding with Wiener-SVD Method", W. Tang et al. [JINST 12, P10002]

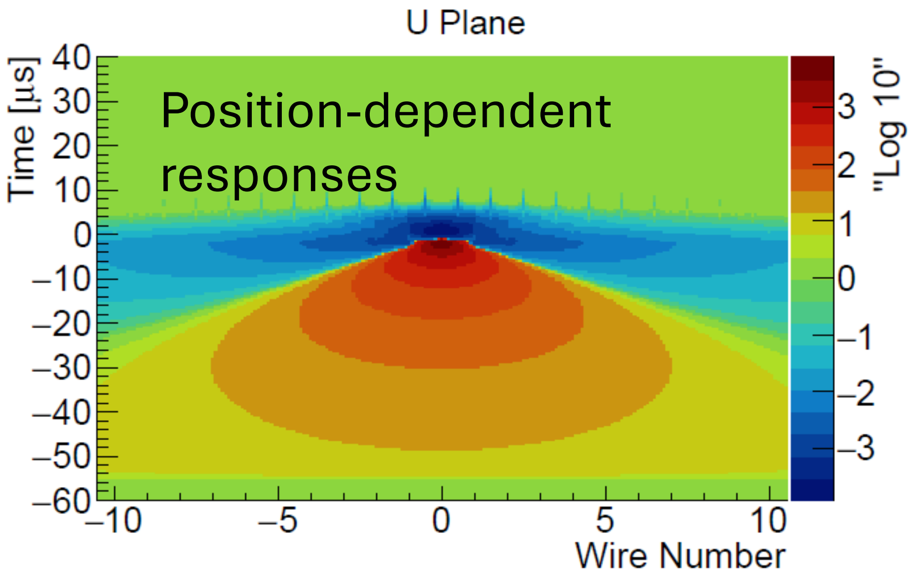


(a) Filters in the time domain. (b) Filters in the frequency domain.

2-D Deconvolution

$$M_i(t_0) = \int_t \left(R_0(t-t_0)\cdot S_i(t) + R_1(t-t_0)\cdot S_{i+1}(t) + \ldots\right) dt$$

$$M_i(\omega) = R_0(\omega)\cdot S_i(\omega) + R_1(\omega)\cdot S_{i+1}(\omega) + \ldots$$



$$\begin{pmatrix} M_1(\omega) \\ M_2(\omega) \\ \ldots \\ M_{n-1}(\omega) \\ M_n(\omega) \end{pmatrix} = \begin{pmatrix} R_0(\omega) & R_1(\omega) & \ldots & R_{n-2}(\omega) & R_{n-1}(\omega) \\ R_1(\omega) & R_0(\omega) & \ldots & R_{n-3}(\omega) & R_{n-2}(\omega) \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ R_{n-2}(\omega) & R_{n-3}(\omega) & \ldots & R_0(\omega) & R_1(\omega) \\ R_{n-1}(\omega) & R_{n-2}(\omega) & \ldots & R_1(\omega) & R_0(\omega) \end{pmatrix} \cdot \begin{pmatrix} S_1(\omega) \\ S_2(\omega) \\ \ldots \\ S_{n-1}(\omega) \\ S_n(\omega) \end{pmatrix}$$

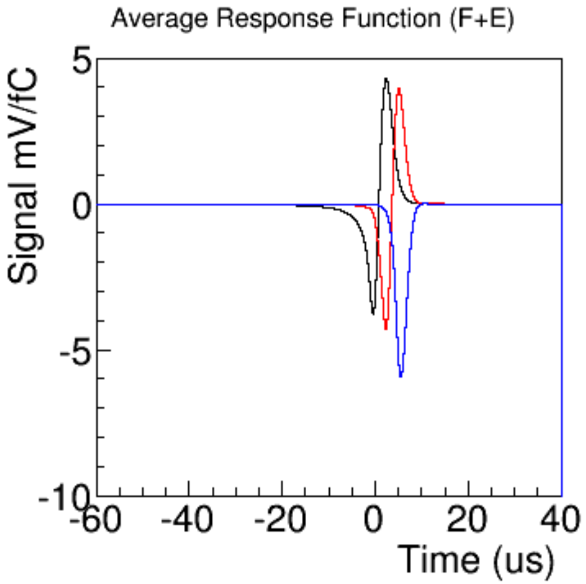
- With induced signals, the signal is still linear summation
  - R_1 represents the induced signal from i+1th wire signal to i-th wire
  - S_i and S_{i+1} are not directly related

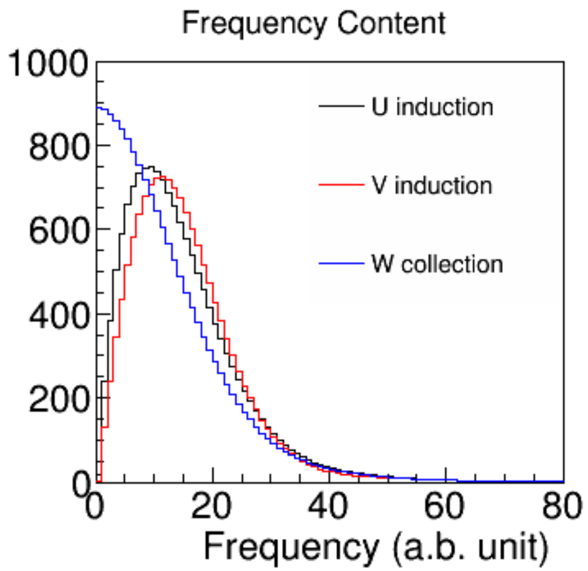
The inversion of matrix R can again be done with deconvolution through 2-D Fast Fourier Transformation

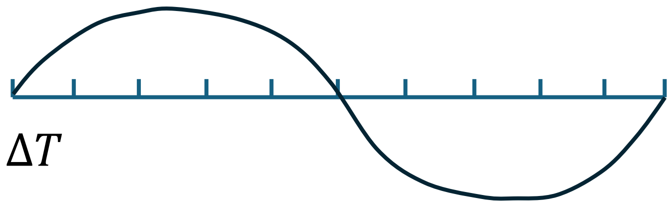
Just 2D deconvolution will not be enough

$$S(\omega) = \frac{M(\omega)}{R(\omega)} \cdot F(\omega)$$

- The bi-polar nature of induction signal amplifies low-frequency (LF) noise during deconvolution
- One can suppress the LF noise with a shorter length of signal region-of-interest (ROI)







$$T_{\max} = N \cdot \Delta T$$

$$f_{\min} = \frac{1}{N} \cdot f_{\mathrm{smp}}$$

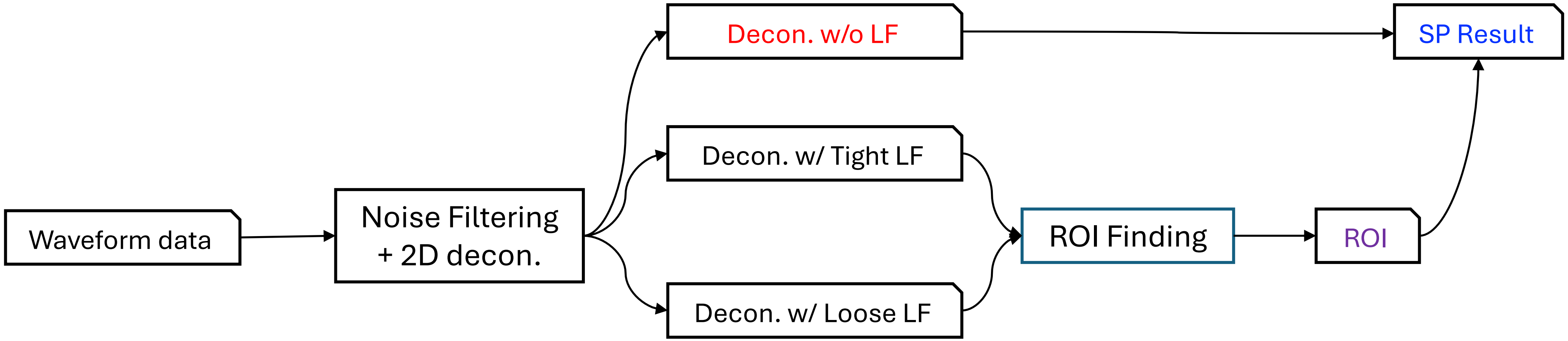
e.g., f_{smp} = 2 MHz (sampling rate)

N= 100 ticks (ROI length)

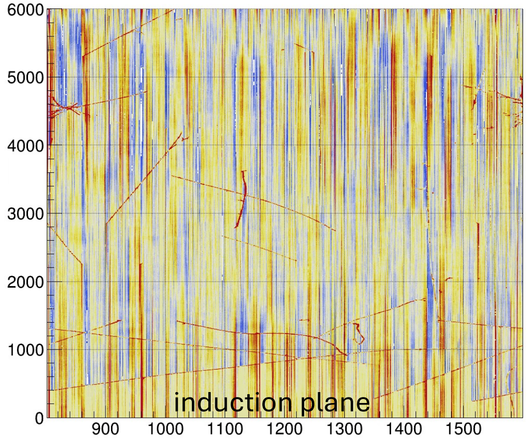
$\Rightarrow$ $f_{\min}$ = 1/100 * 2 MHz = 20 kHz

Not sensitive to LF noise < 20 kHz,

Procedure of Wire-Cell Signal Processing

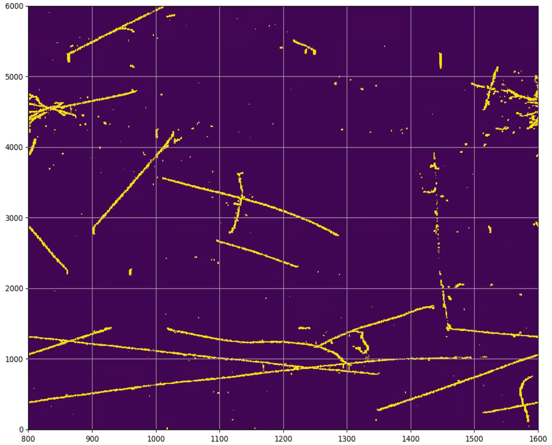


Decon. w/o low-freq. (LF) filter
Waveform ⟶ charge, dense



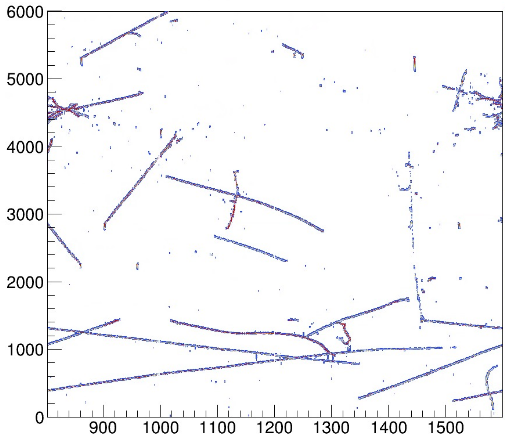
+

ROI:
Hit finding, sparsify

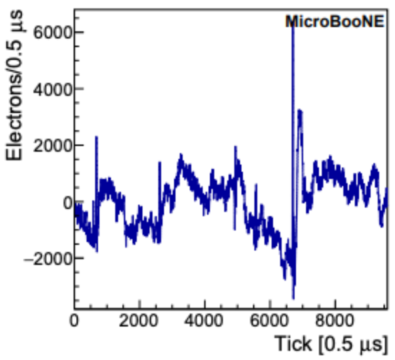


➡

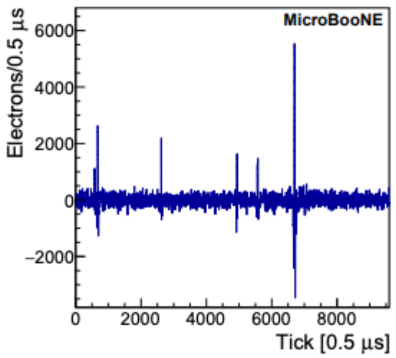
SP result:
Sparse, charge



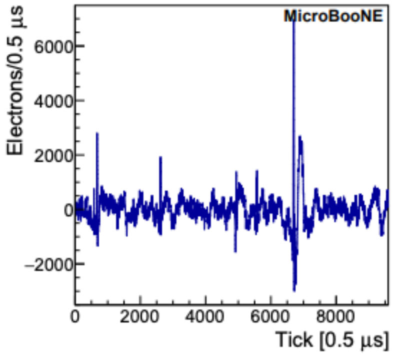
Identification of Signal Region-of-interest (ROI)



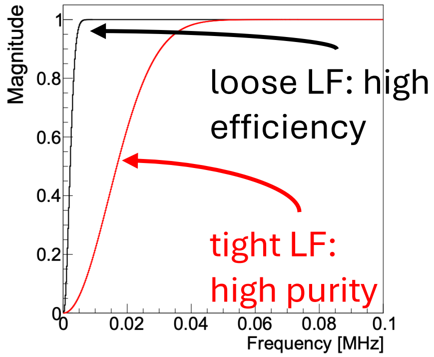
(a) Without low-frequency filter.

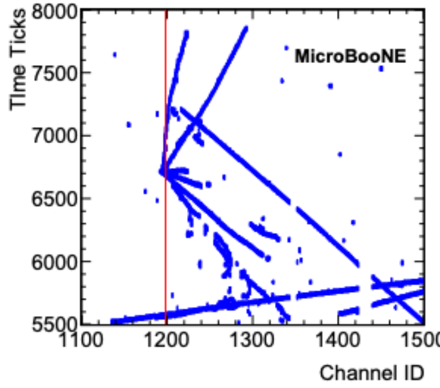


(b) With tight low-frequency filter.

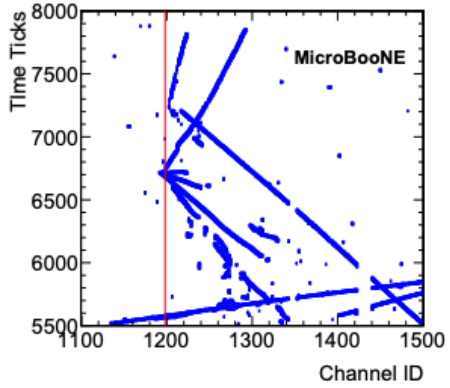


(c) With loose low-frequency filter.

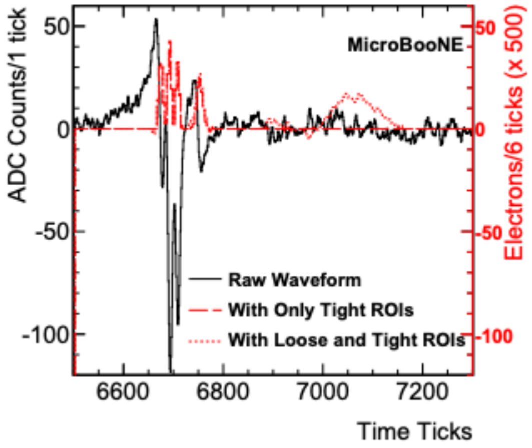




(a) Deconvolved signal with "loose and tight ROIs".

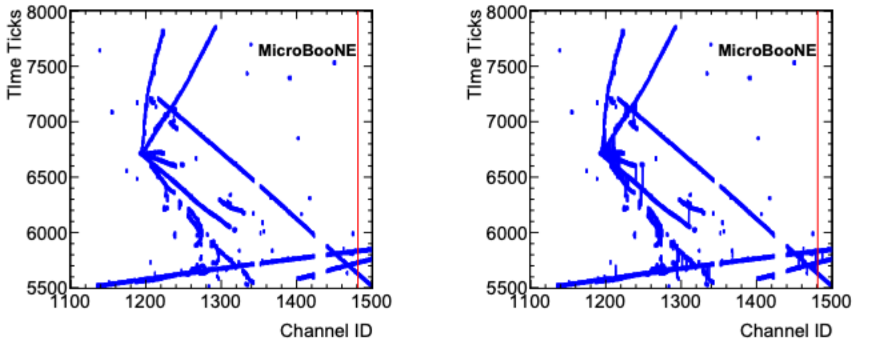


(b) Deconvolved signal with "tight ROIs".

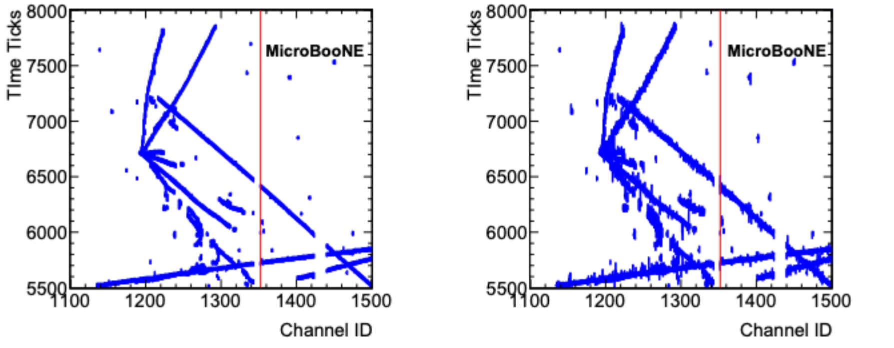


- Deconvolution amplifies low-freq. (LF) noise in induction wires
- LF filters are applied to search for ROIs
- Expect high efficiency but low purity from initial ROI search

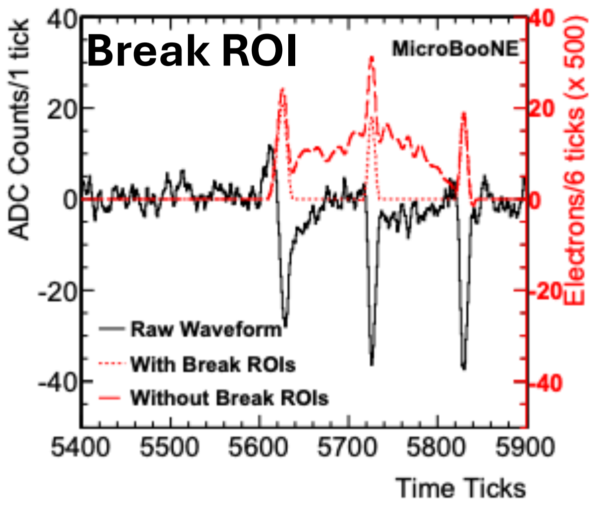
Rule-based ROI Refinement



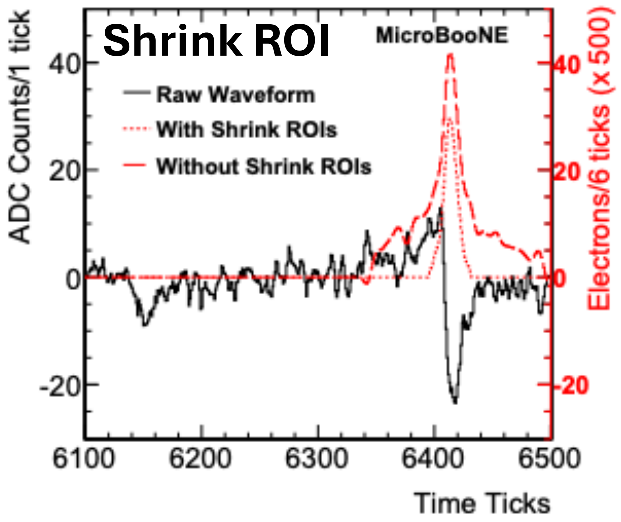
(a) Deconvolved signal with "break ROIs". (b) Deconvolved signal without "break ROIs".



(a) Deconvolved signal with "shrink ROIs". (b) Deconvolved signal without "shrink ROIs".

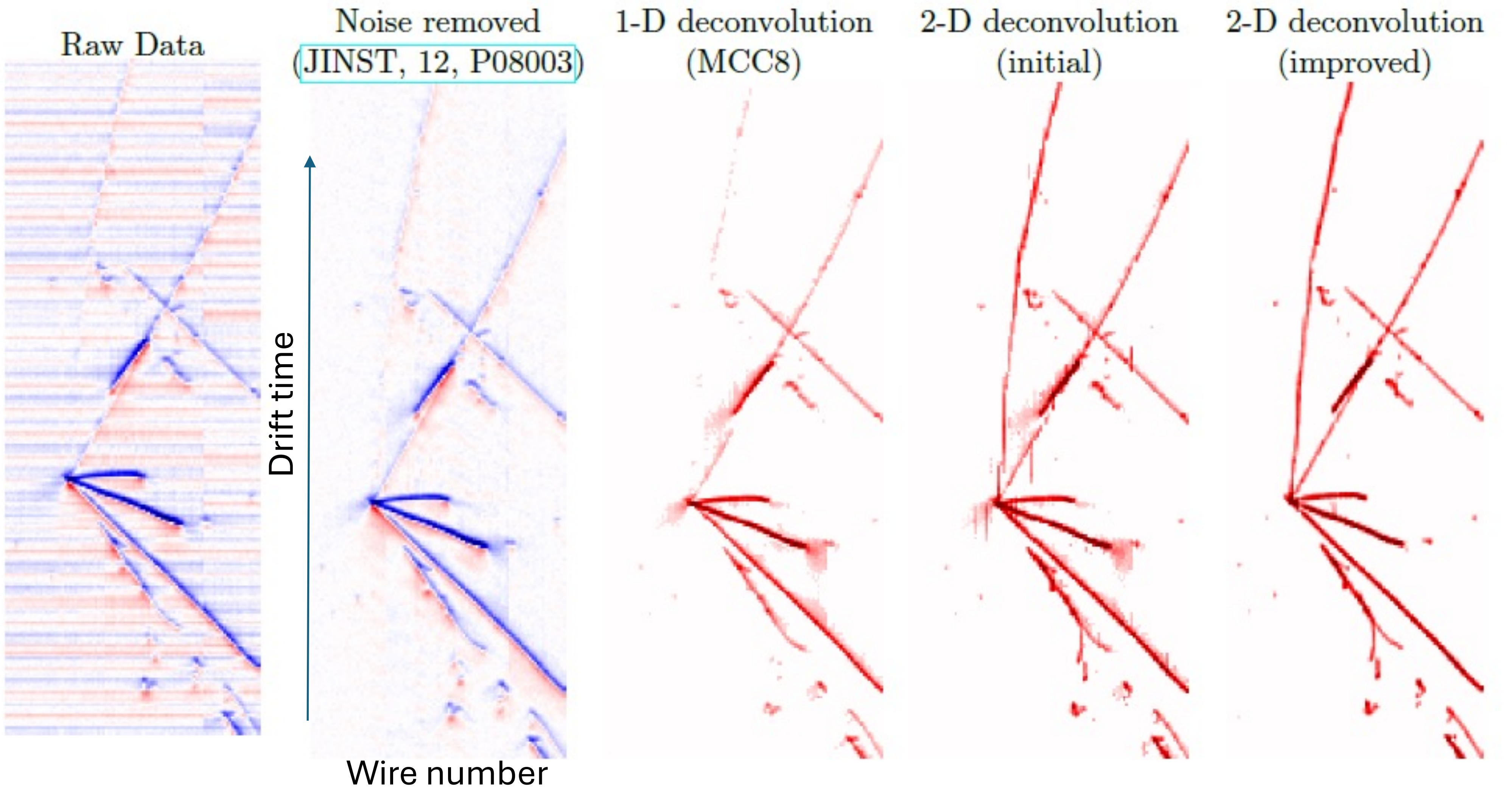


- ROIs close to each other are often identified with one ROI due to LF noise
- Important for ROIs close to particle interaction vertices



- Reduced ROI length based on "tight ROI" and its connectivity with "loose ROI" on adjacent channels

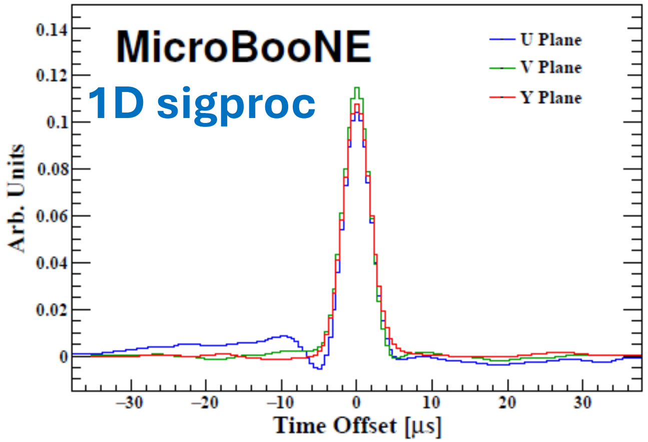
Wire-Cell NF/SP Performance on MicroBooNE



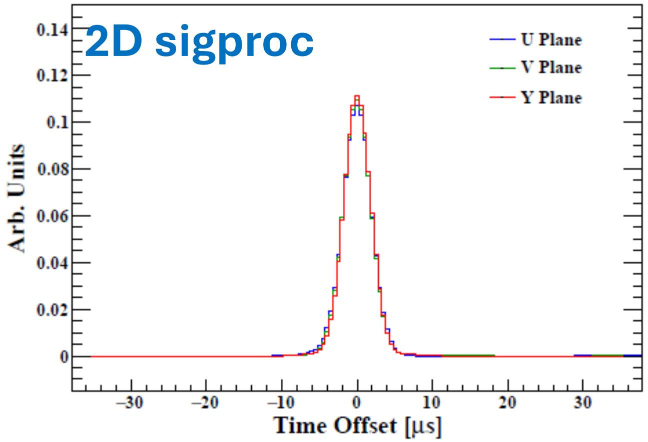
Two years efforts summarized in JINST 13 P07006 and JINST 13 P07007

JINST, 13, P07007

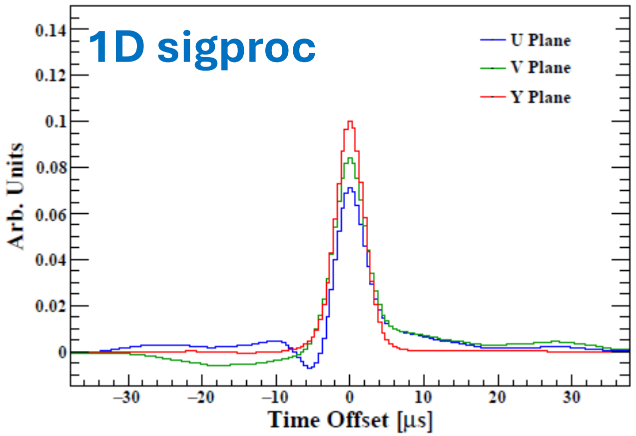
Signal Processing: 1D *vs* WireCell 2D



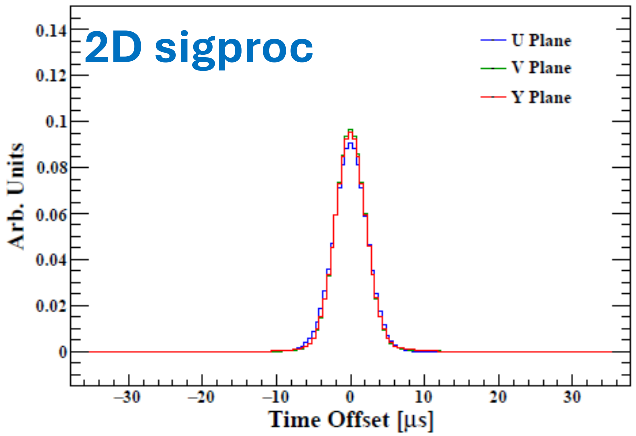
(a) 1D deconvolution, $5° < \theta_{xz} < 15°$.



(b) 2D deconvolution, $5° < \theta_{xz} < 15°$.



(e) 1D deconvolution, $30° < \theta_{xz} < 50°$.



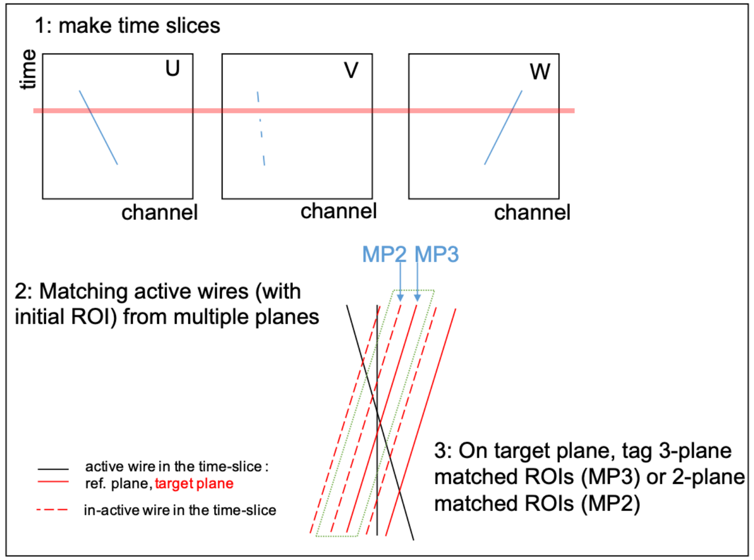
(f) 2D deconvolution, $30° < \theta_{xz} < 50°$.

- Average reconstructed ionization charge for cosmic tracks in different angle
- WireCell 2D signal processing can correctly recover identical charge from each wire plane
- Enables LArTPC tomographic reconstruction
- **More discussion in parallel session for Experimental Need**

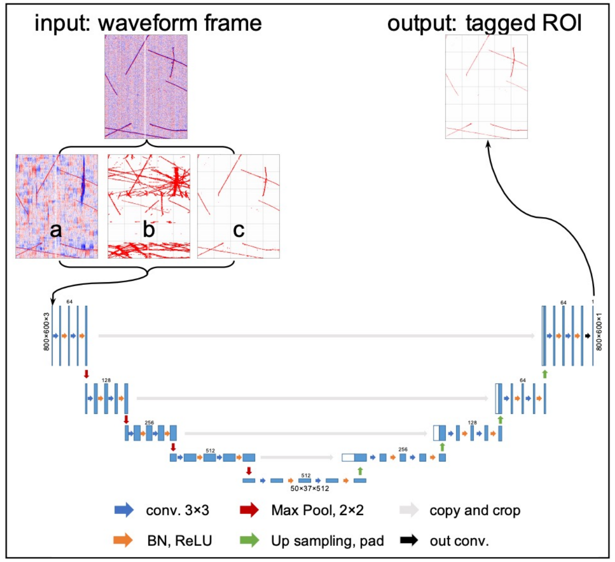
JINST 16 (2021) 01, P01036

DNN ROI with 3-plane Information

Multi-plane information in Signal Processing



DNN ROI finding with multiple input channel

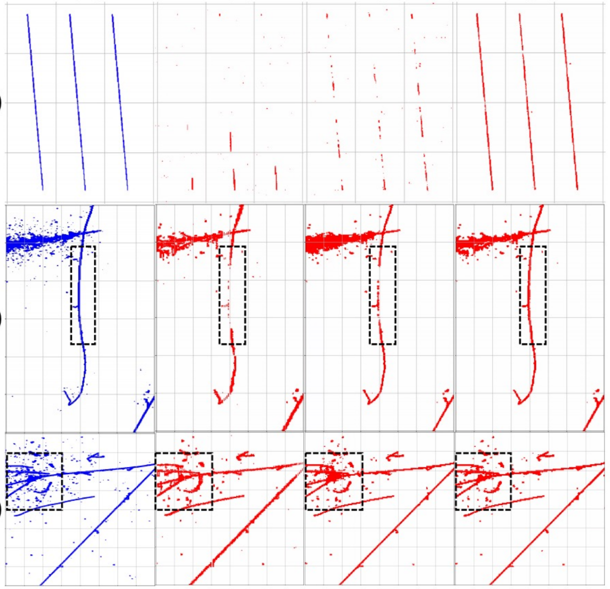


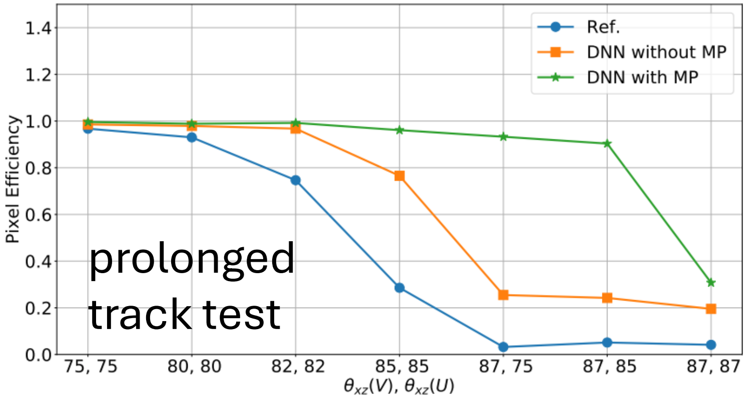
Truth (a) | Ref. (b) | DNN w/o MP (c) | DNN w/ MP (d)

(1)

(2)

(3)





- Information from other wire planes can be used to protect weak ROIs (e.g., low S/N in prolonged tracks)
- Deep learning technique can further improve the ROI refinement
- **Also see Lynn's talk on SBND signal processing**

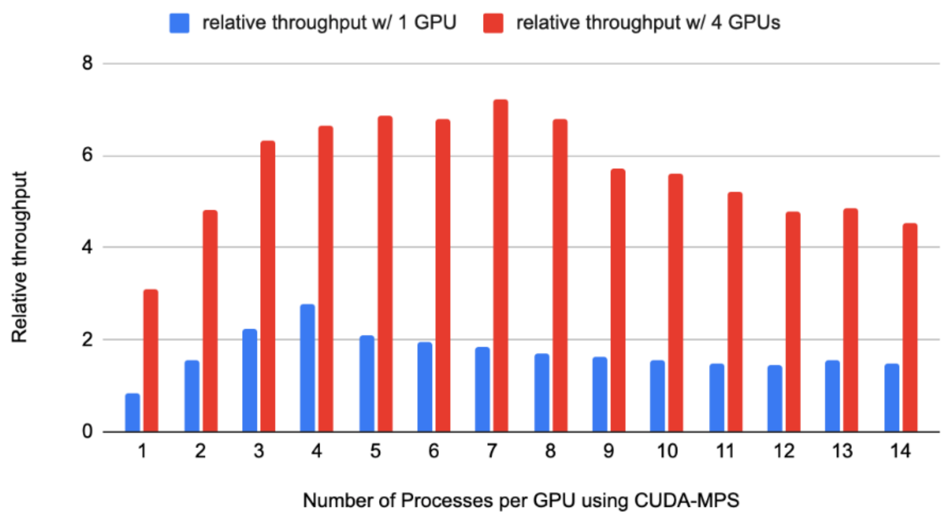
Portable parallelization for simulation

Developed by H. Yu and BNL CSI

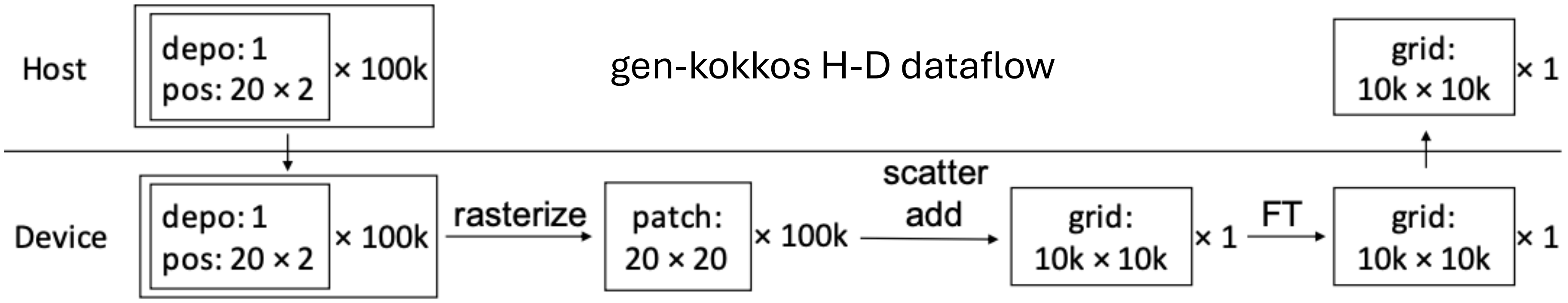
Kokkos as example

- ACAT21: https://arxiv.org/pdf/2203.02479.pdf
- Dedicated optimizations to batch data and to minimize host-device IO
- 33 times faster 1 A100 GPU vs. 1 CPU core
- 3 times throughput 1 A100 GPU vs. 64 CPU cores

gen-kokkos performance



Perlmutter A100 GPU, EPYC 7763 64-core CPU



IDFT: interface for multiple FFT implementation

- Developed by B. Viren and Brandon Feder
- Data copying between CPU/GPU needed
- Using IDFT alone seems not the optimal way to speed things up
  - useful add-on when having idling GPU cycles
- Still under development

FftwDFT

```
[21:15:51.634] I [ timer   ] Timer: WireCell::Gen::DepoTransform : 16.32 sec
[21:15:51.634] I [ timer   ] Timer: WireCell::Sio::FrameFileSink : 4.51 sec
[21:15:51.634] I [ timer   ] Timer: WireCell::Gen::DepoSetDrifter : 0.74 sec
[21:15:51.634] I [ timer   ] Timer: WireCell::Sio::DepoFileSource : 0.49 sec
[21:15:51.634] I [ timer   ] Timer: WireCell::Gen::Digitizer : 0.31 sec
[21:15:51.634] I [ timer   ] Timer: WireCell::Gen::AddNoise : 0.21 sec
[21:15:51.634] I [ timer   ] Timer: WireCell::Gen::Reframer : 0.03 sec
[21:15:51.634] I [ timer   ] Timer: WireCell::Gen::DepoSetFanout : 0 sec
[21:15:51.634] I [ timer   ] Timer: WireCell::Gen::FrameFanin : 0 sec
[21:15:51.634] I [ timer   ] Timer: WireCell::Gen::Retagger : 0 sec
[21:15:51.634] I [ timer   ] Timer: Total node execution : 22.609999937936664 sec
[21:15:51.635] D [   io    ] <FrameFileSink:> closing frames-pr173-cpu.tar.bz2 after 2 calls
```

cuFftDFT

```
[21:14:02.248] I [ timer   ] Timer: WireCell::Gen::DepoTransform : 13.09 sec
[21:14:02.248] I [ timer   ] Timer: WireCell::Sio::FrameFileSink : 6.01 sec
[21:14:02.248] I [ timer   ] Timer: WireCell::Gen::DepoSetDrifter : 0.89 sec
[21:14:02.248] I [ timer   ] Timer: WireCell::Gen::AddNoise : 0.67 sec
[21:14:02.248] I [ timer   ] Timer: WireCell::Sio::DepoFileSource : 0.54 sec
[21:14:02.248] I [ timer   ] Timer: WireCell::Gen::Digitizer : 0.35 sec
[21:14:02.248] I [ timer   ] Timer: WireCell::Gen::Reframer : 0.03 sec
[21:14:02.248] I [ timer   ] Timer: WireCell::Gen::FrameFanin : 0 sec
[21:14:02.248] I [ timer   ] Timer: WireCell::Gen::Retagger : 0 sec
[21:14:02.248] I [ timer   ] Timer: WireCell::Gen::DepoSetFanout : 0 sec
[21:14:02.248] I [ timer   ] Timer: Total node execution : 21.580000398680568 sec
```

Summary

- An accurate response model is crucial for both LArTPC simulation and signal processing
- A 2-D deconvolution technique is developed in Wire-Cell, providing better data-MC agreement with long-range induction effect considered
- The bi-polar nature of induction wires amplifies low-freq. noise in deconvolution, which can be suppressed with a proper determination of ROIs
- GPU accelerations are being investigated for Wire-Cell simulation and signal processing

References

Wire-Cell simulation and signal processing:

- *JINST 12 P08003*
- *JINST 13 P07006*
- *JINST 13 P07007*
- *M. Diwan. Basic mathematics of random noise part 1/2*
- "Wire-Cell TPC Responses, Simulation, Signal Processing and Implications for Vertical Drift Designs", Brett et al.

Wire-Cell DNN ROI finding:

- *JINST* 16 P01036

Wire-Cell Vertical Drift development:

- Noise Simulation
- Vertical Drift SimChannel and Horizontal Drift validation
- Wire-Cell TPC simulation for Vertical Drift
- Field Response Simulation
- Initial Shape Validation of the field simulation 50L 2view prototype