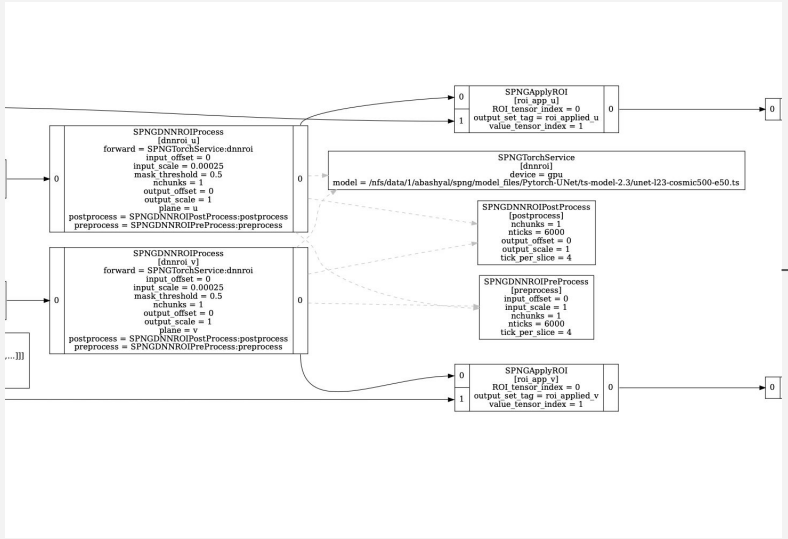
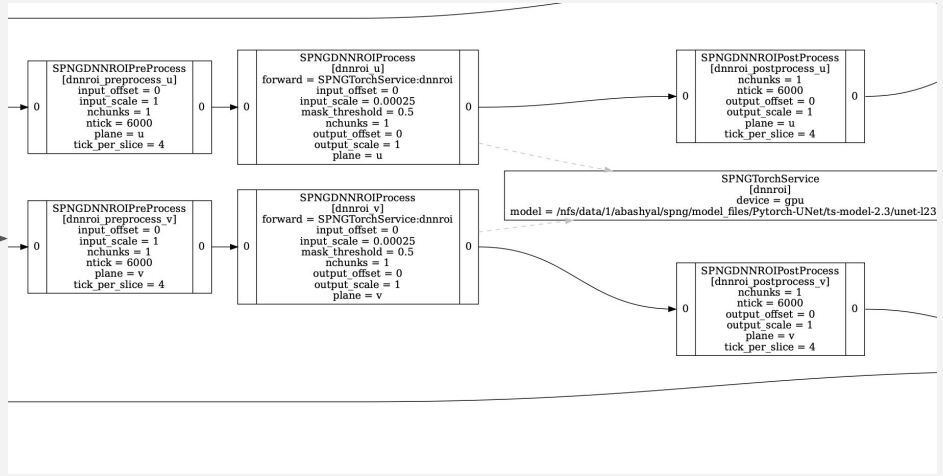
DUNE SPNG Updates

November 14, 2025

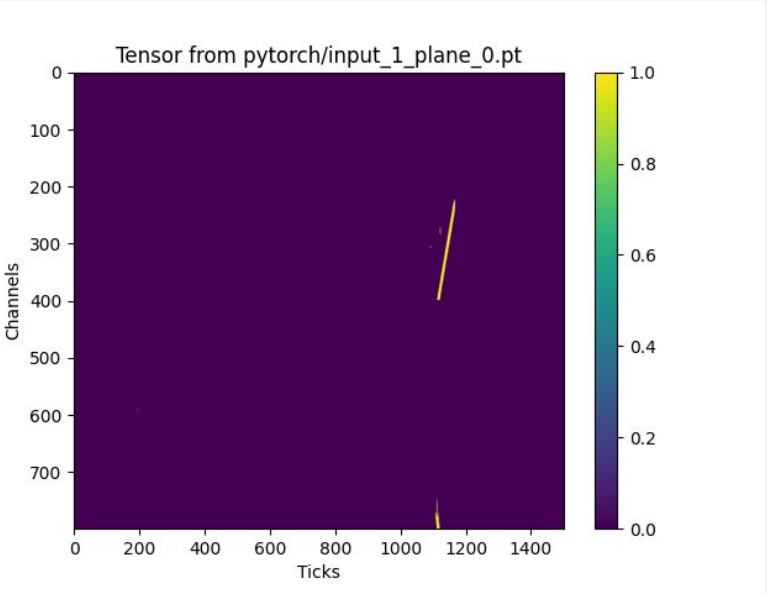
From Last Week

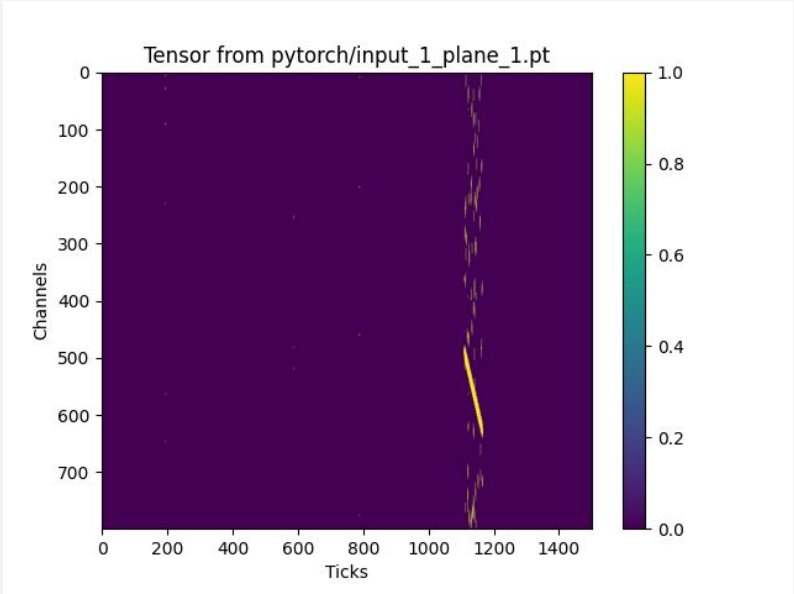




Pre and Post Processing of DNN-ROI changed from Service to Nodes

From Last Week



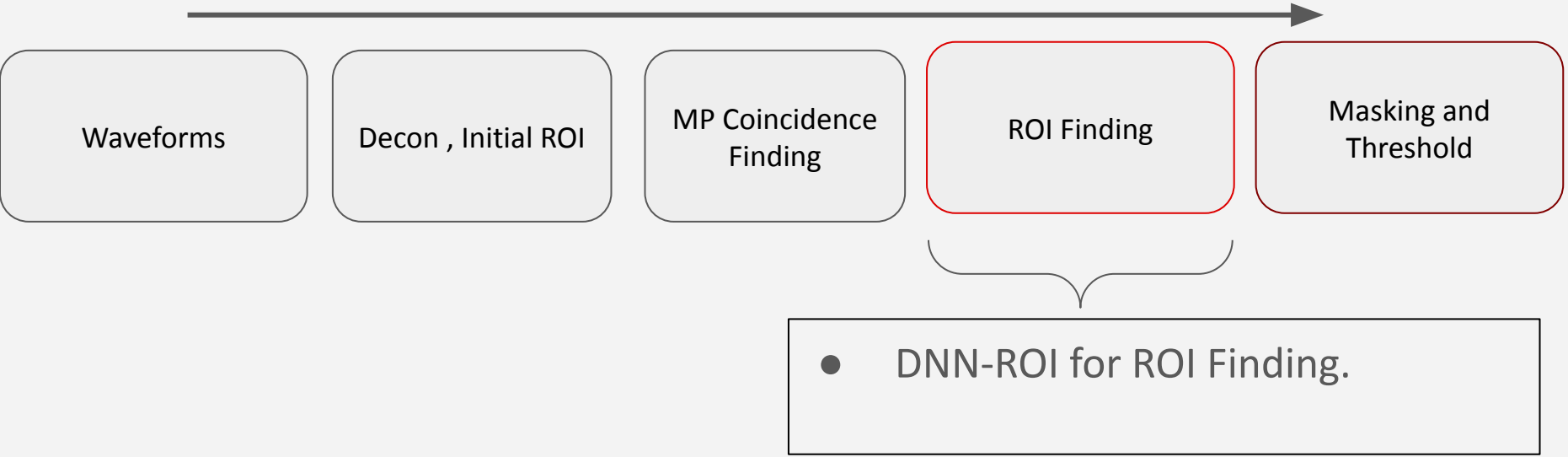


MP2 plot for the V plane (right) from OSP. Seems like this artifact is really there (?)

Validating the DNNROI implementation in the SPNG

- Validate the implementation of DNNROI by benchmarking against the OSP (Original Signal Processing) Steps

Signal Processing Step in Rough for illustration

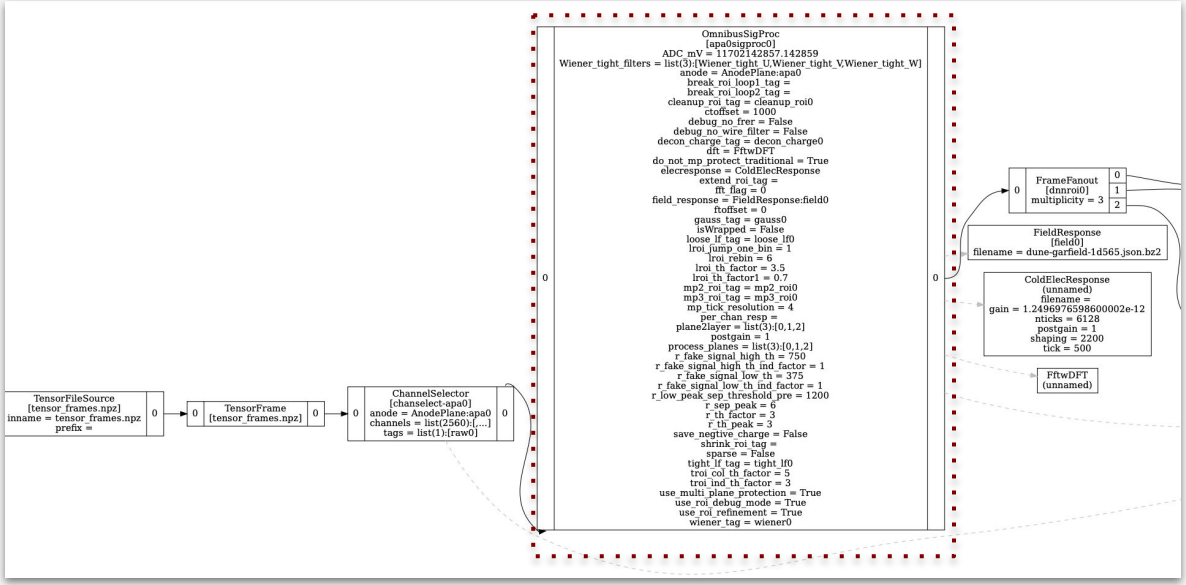


Signal Processing Step OSP

Waveforms

Decon , Initial ROI

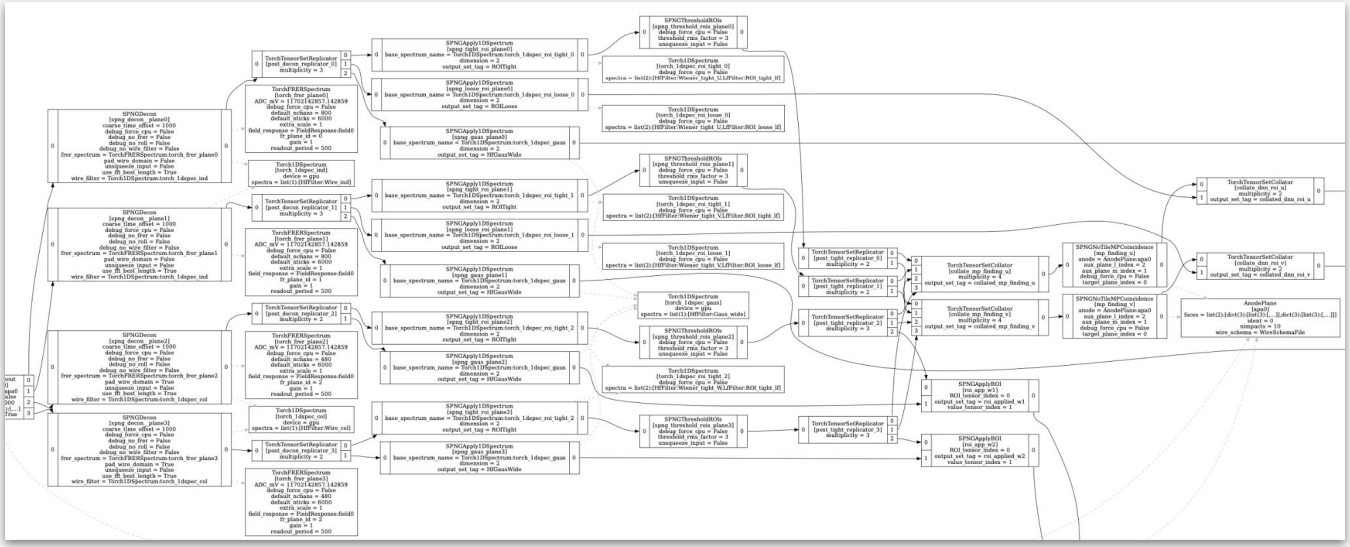
MP Coincidence Finding



Most of the heavy lifting done by OmnibusSigProc.

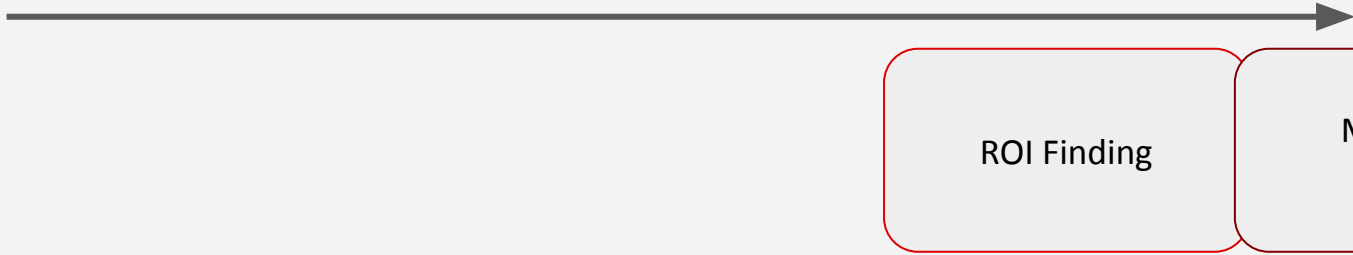
Signal Processing Step SPNG

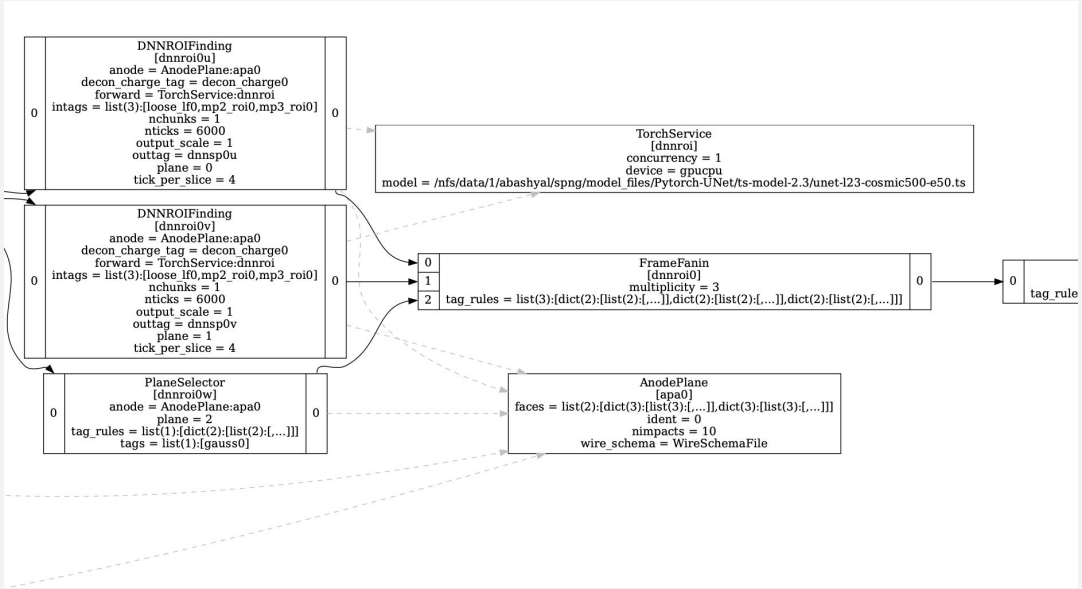
Waveforms → Decon → Initial ROI → MP Coincidence Finding



In SPNG, split into Decon, filters, MP2/MP3 co-incidence finding.

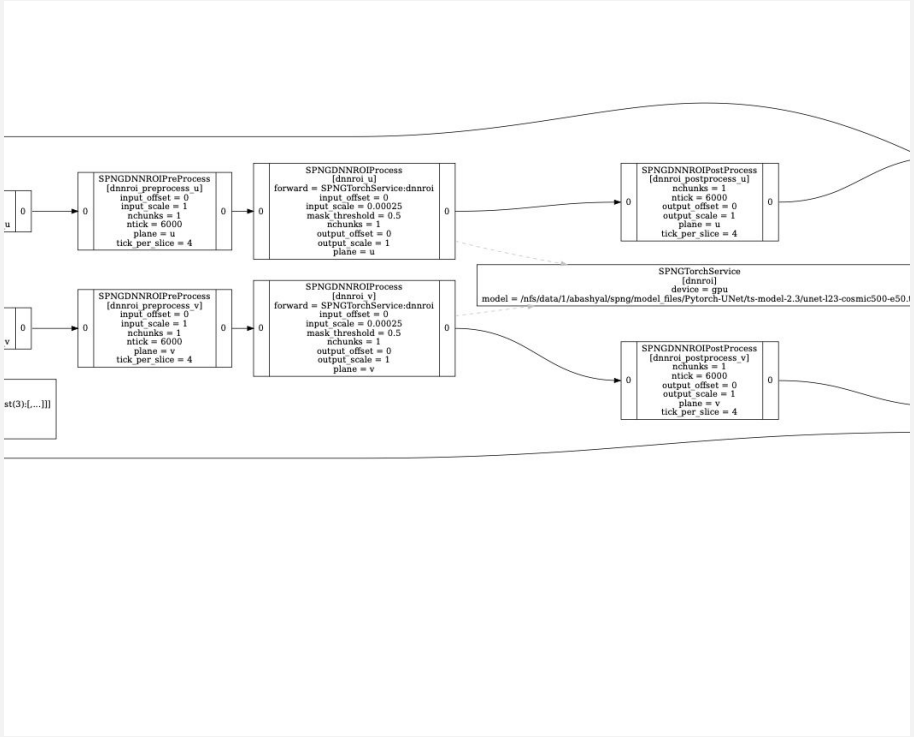
(DNN) ROI Finding (OSP)





- DNNROIFinding
  - Preprocessing, inference (via TorchService) and Postprocessing)
  - Masking and Threshold

(DNN) ROI Finding (SPNG)



ROI Finding

Masking and Threshold

- Pre-Processing [DNNROIPreProcess]
- Inference [DNNROIProcess]
- Post-Processing [DNNROIPostProcess]

Pre-Processing

- Pre-Process Target Tensor [u,v], MP2 and MP3 coincidence tensors
  - Scaling and offsetting (User defined scale and offset factor)
  - Rebin Ticks [6000 ticks → user defined [1500 ticks] (**Downsampling**)
  - Shape consistency → [Batch, Channel, Time]
  - Stack pre-processed tensors (target, mp2,mp3)
    - Consistency in stacking

Processing

- Forward stacked Tensors for inference using TorchService
- Retrieve Back the post inference Tensor (Target tensor only)

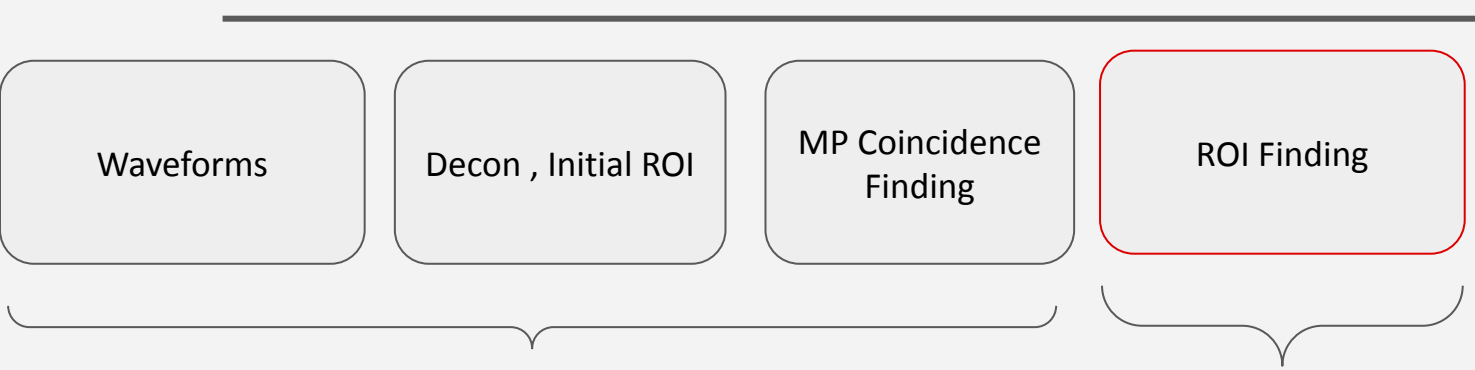
Post Processing

- Upsample [1500 ticks → 6000 ticks] post inference tensor
- Dimension consistency → 3 D tensor
- OSP
  - Masking and Thresholding
- SPNG
  - Marking and Threshold in another stage

Benchmarking SPNG DNNROI against OSP

- Downsample
  - OSP: Use of WireCell::Array libray library on Eigen Arrays
  - SPNG: torch libraries (slicing, average-pool)
- Inference
- Upsampling
  - OSP: Use of WireCell::Array libray library on Eigen Arrays
  - SPNG: Use of torch libraries

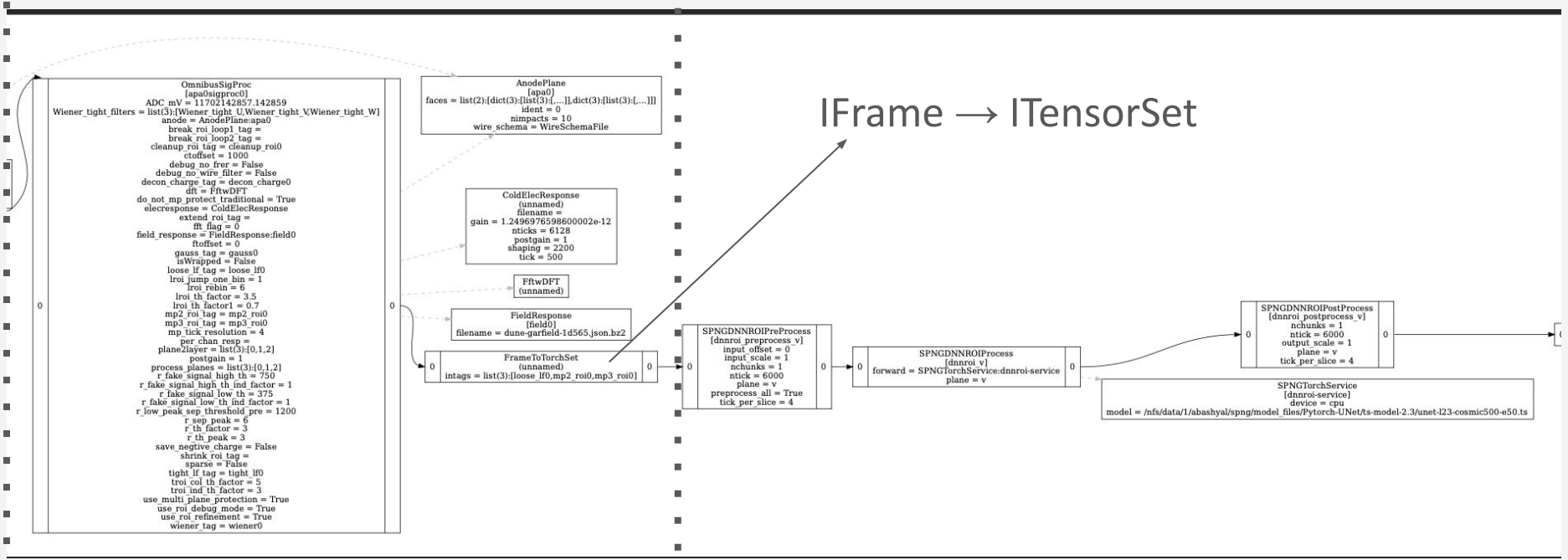
Benchmarking



Keep this identical

We Want to Bench Mark This Step
Run with OSP DNNROI
Run with SPNG DNNROI

Hybrid Workflow

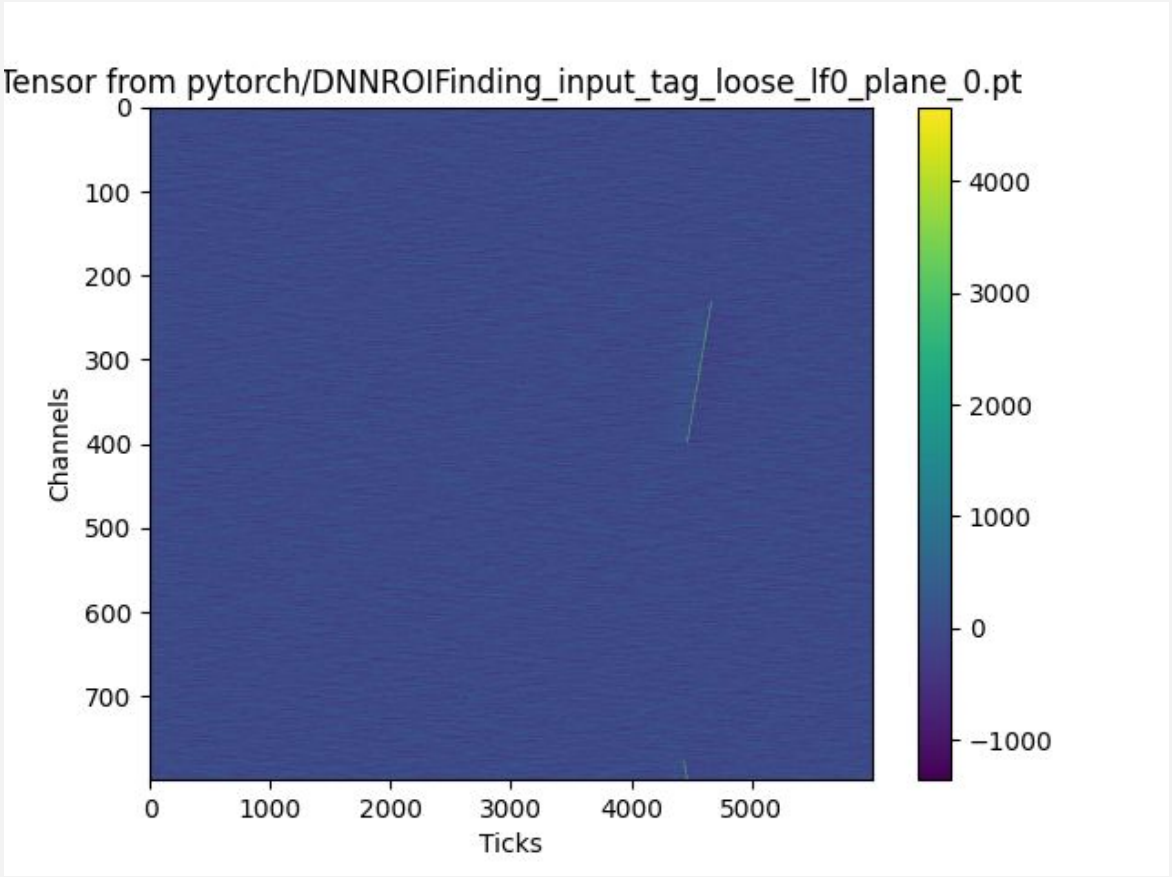


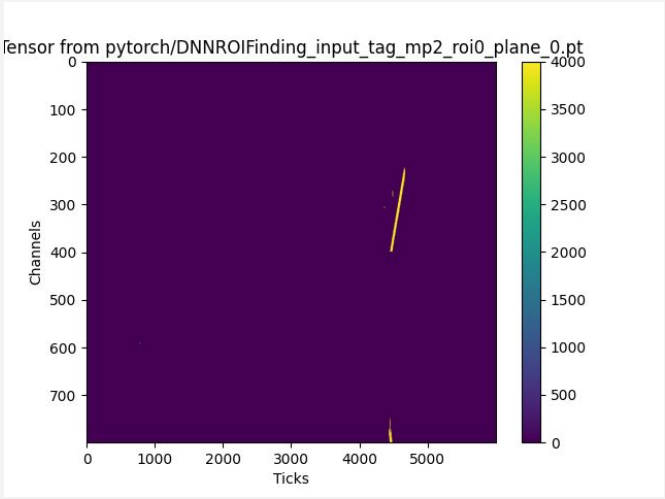
Upstream stages from OSP to ensure both OSP and SPNG DNNROI get identical input

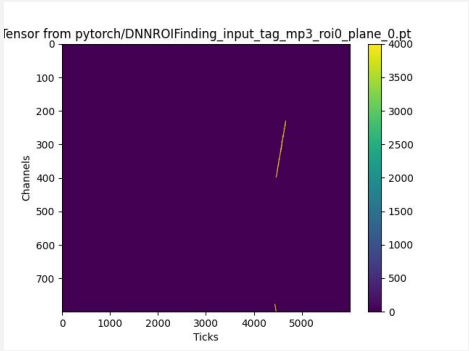
Input Files

- Only showing the U planes
- Except for one slide, comparison shows processing done in the CPU

Input Files


Tensor from pytorch/DNNROIFinding_input_tag_loose_lf0_plane_0.pt
Channels
Ticks
0
100
200
300
400
500
600
700
0
1000
2000
3000
4000
5000
4000
3000
2000
1000
0
−1000


Tensor from pytorch/DNNROIFinding_input_tag_mp2_roi0_plane_0.pt
Channels
Ticks
0
100
200
300
400
500
600
700
0
1000
2000
3000
4000
5000
4000
3500
3000
2500
2000
1500
1000
500
0

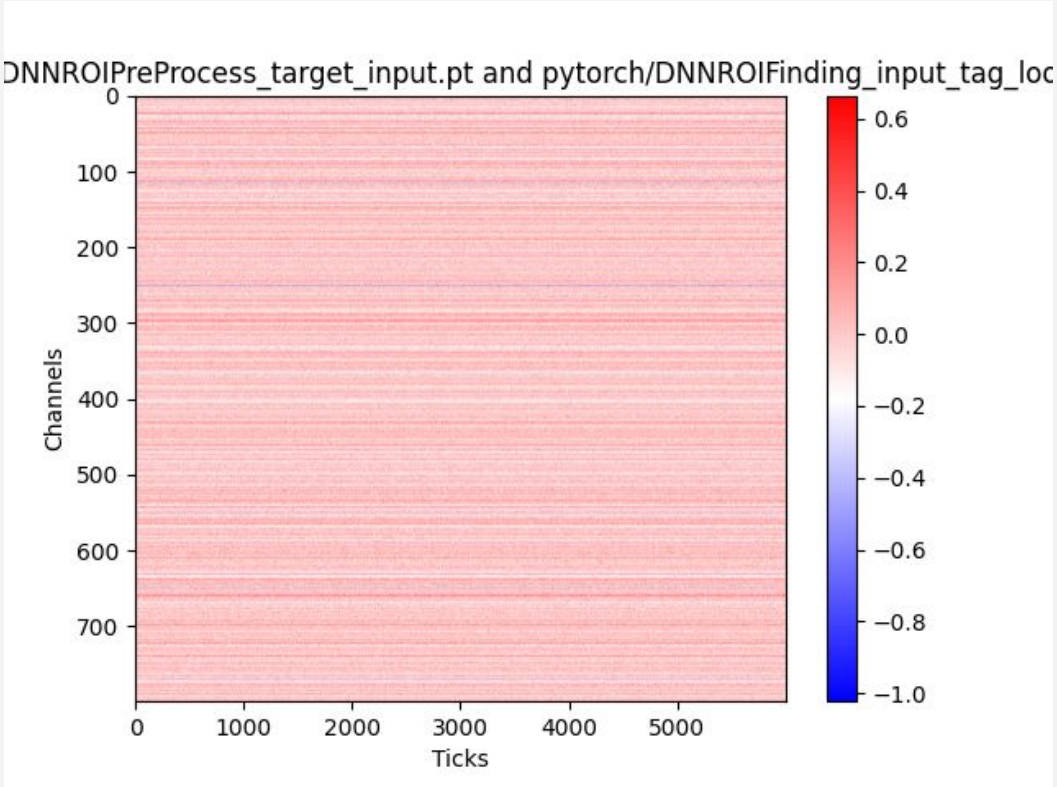

Tensor from pytorch/DNNROIFinding_input_tag_mp3_roi0_plane_0.pt
Channels
Ticks
0
100
200
300
400
500
600
700
0
1000
2000
3000
4000
5000
4000
3500
3000
2500
2000
1500
1000
500
0

Overall Summary

Stage	OSP	SPNG	MSE
Input File			
Input Target	DNNROIFinding_ir	DNNROIPreP	0
Input MP2	DNNROIFinding_ir	DNNROIPreP	0
Input MP3	DNNROIFinding_ir	DNNROIPreP	0
Downsampled			
Downsampled Target	DNNROIFinding_ir	DNNROIPreP	0
Downsampled MP2	DNNROIFinding_ir	DNNROIPreP	0
Downsampled MP3	DNNROIFinding_ir	DNNROIPreP	0
After Inference			
Target	DNNROIFinding_cl	DNNROIProc	4.57E-11
After Upsampling			
Target	DNNROIFinding_u	DNNROIPost	0
Masking		NA	
Threshold		NA	

Mean Squared Error as a metric to quantify the difference between the OSP and SPNG tensors

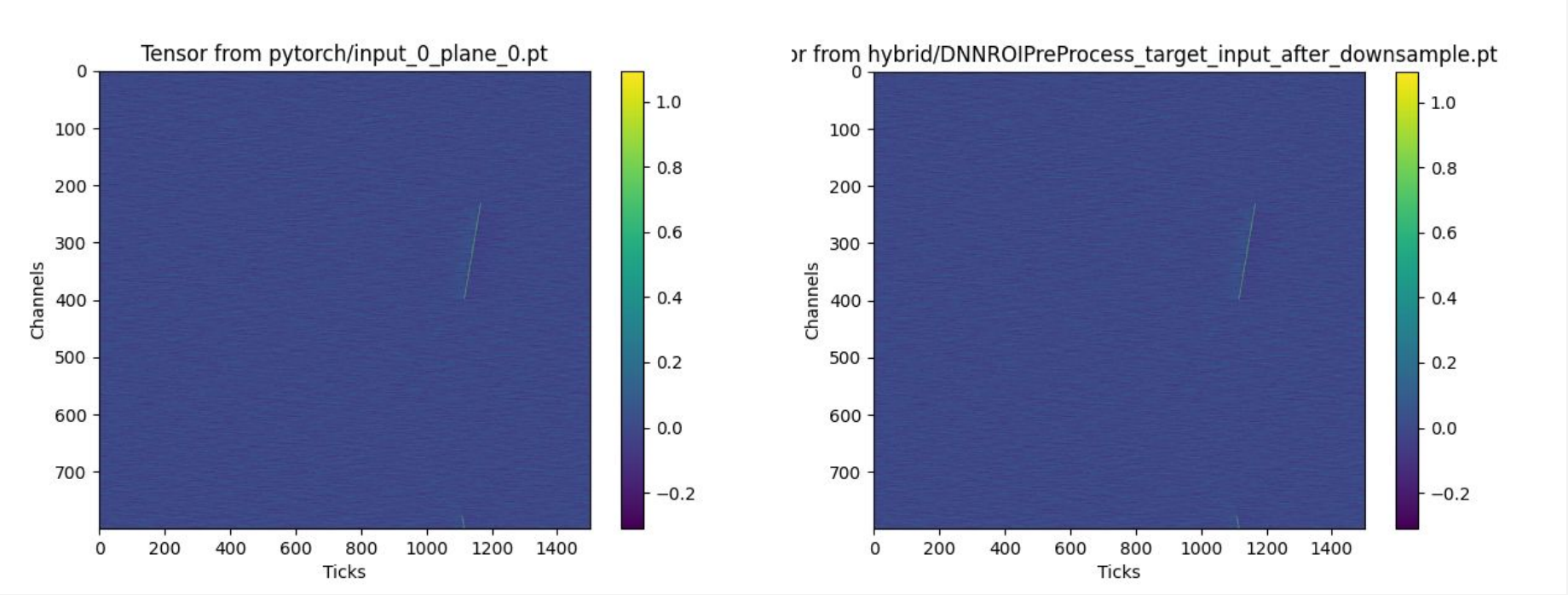
Difference of the input files (CPU- GPU)



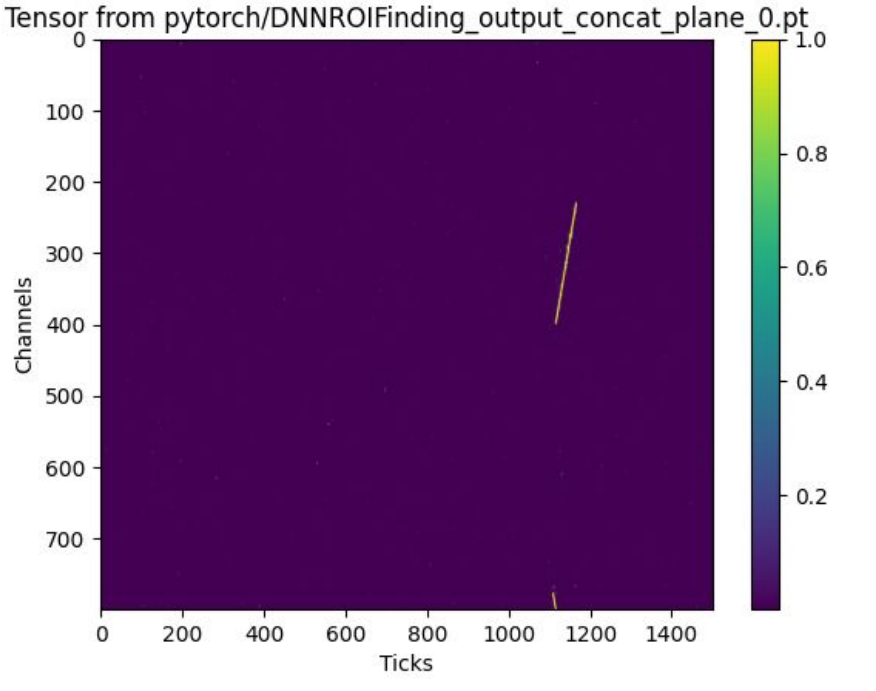
Mean squared error (MSE): ~0.014

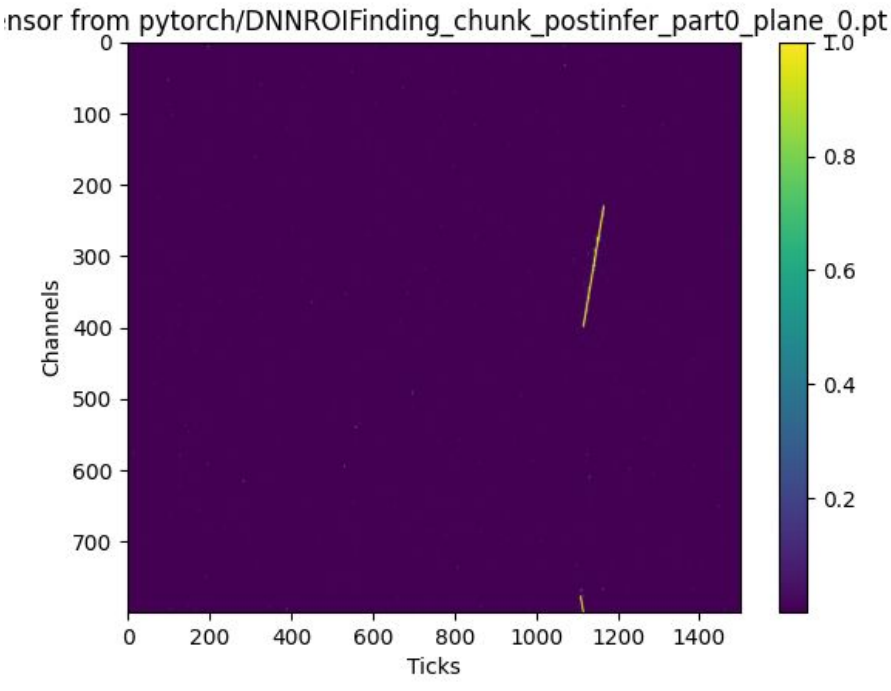
Difference between the output of Post Signal Processing in GPU and CPU.

OSP vs SPNG (Downsample)

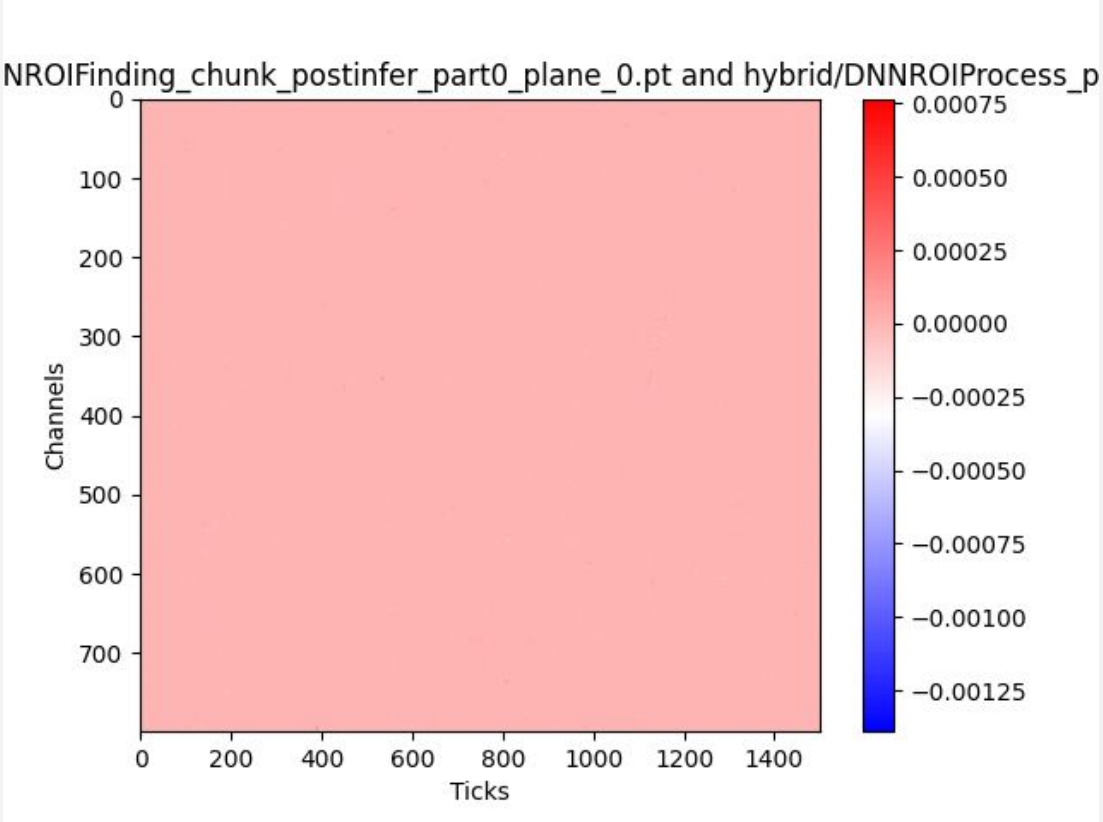


OSP vs. SPNG (Post Inference)

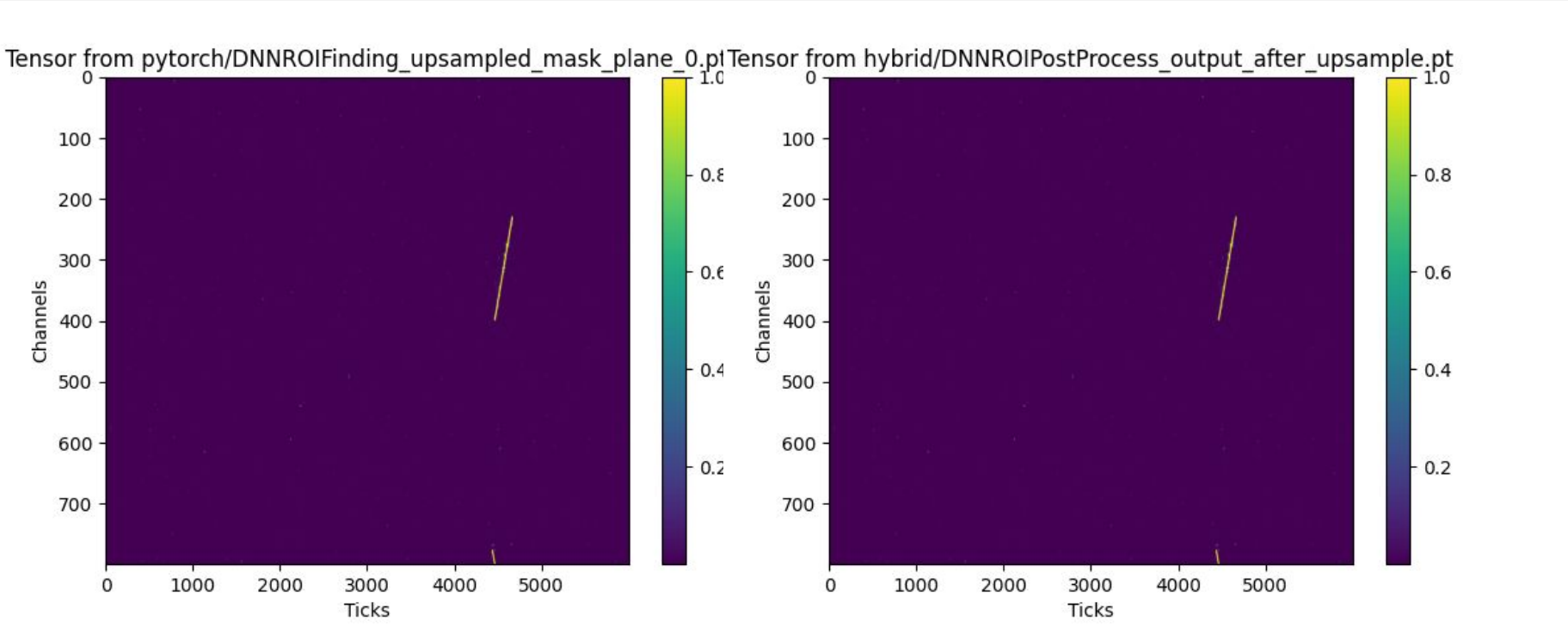




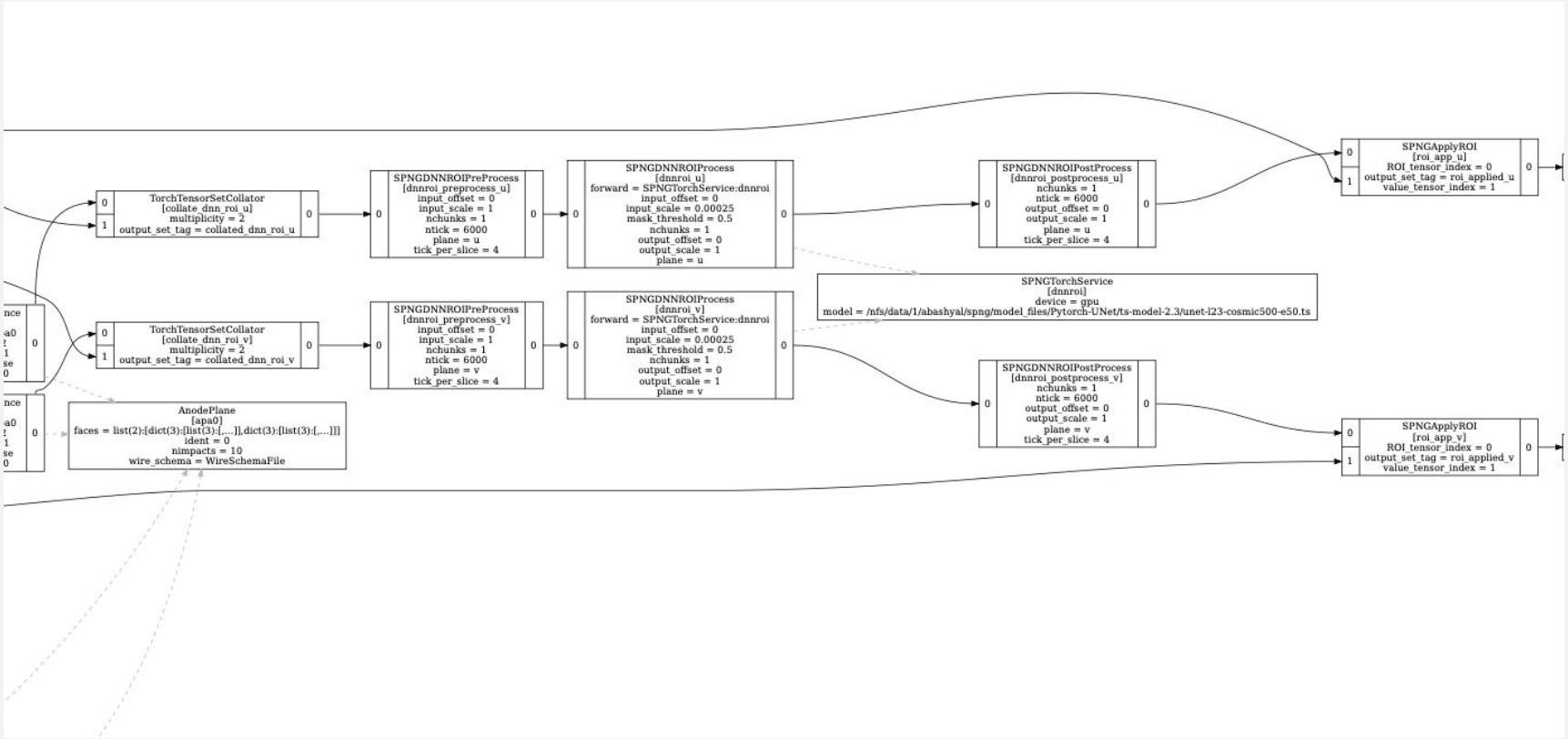
Difference Plot



OSP vs SPNG Upsampling

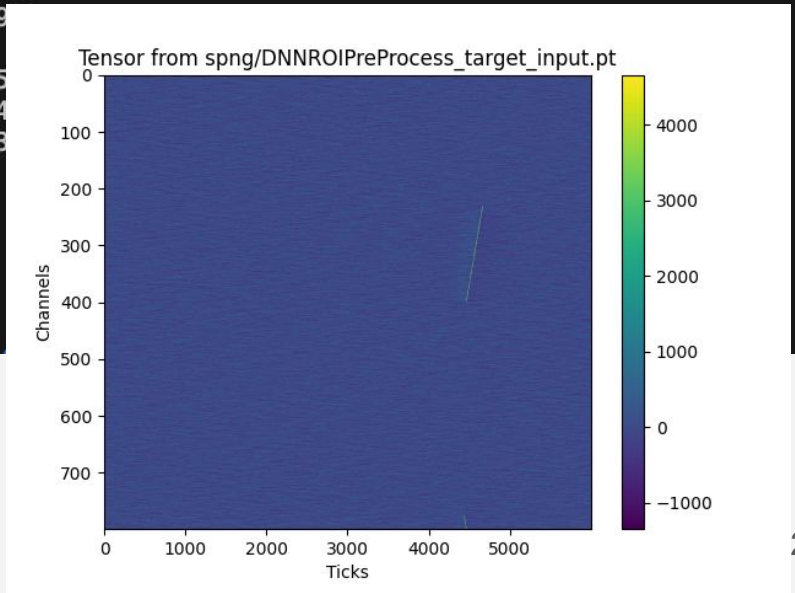


Changing Pre and Post Processing steps to Nodes from service



Input tensor comparison (spng/OSP) target Tensor

```
abashyal@wcgpu0:/nfs/data/1/abashyal/spng/spng_dev_09302025/spng_dev/wire-cell-toolkit/spng/test/pytools$ tpython compare_ten
IPreProcess_target_input.pt hybrid/DNNROIPreProcess_target_input.pt
Parameter from spng/DNNROIPreProcess_target_input.pt has shape torch.Size([1, 800, 6000]) and dtype torch.float64
Parameter from hybrid/DNNROIPreProcess_target_input.pt has shape torch.Size([1, 800, 6000]) and dtype torch.float32
Mean Squared Error between the two tensors: 82.09602838303353
Difference plot saved to ./comparison_DNNROIPreProcess_target_input.pt)_DNNROIPreProcess_target_input.pt.png
First 20 values of the first tensor:
[-282.031036   -384.61549069 -325.45189639 -128.70842339  122.32225584
  323.26104144  394.27375499  312.05754028  115.30507905 -116.27476158
 -298.08968777 -375.39473374 -341.9319162  -235.48779622 -113.86681631
  -23.87934457   20.60210069   40.0647697    70.00898544  132.7779269
First 20 values of the second tensor:
[-288.83716  -391.31946  -332.2404   -135.69762   115.13656   315.975
  387.0088    304.89624   108.27026  -123.21338  -304.99573  -382.344
 -348.9874   -242.6676   -121.12116   -31.09877    13.53353    33.193
   63.264835  125.99377 ]
First 20 values of the difference:
[6.8061222  6.70396732 6.78849057 6.98919318 7.18569731 7.2854555
 7.26496592 7.16130005 7.03481538 6.93861733 6.90603977 6.94935196
 7.05548004 7.17980632 7.2543461  7.21942557 7.06857046 6.87081141
 6.74415008 6.78416015]
```



We start with different tensors….

Post Inference we end up with different tensors

```
Mean Squared Error between the two tensors: 0.00032544127316214144
Difference plot saved to ./comparison_DNNROIProcess_post_forward_tensor.pt)_DNNROIProcess_post_forward_tensor.pt.png
First 20 values of the first tensor:
[0.1673405  0.23058009 0.10084029 0.0187975  0.00861831 0.02837664
 0.01102848 0.02291899 0.01905112 0.08326053 0.03201512 0.01032963
 0.1302534  0.05322308 0.00793222 0.05788688 0.0517725  0.02288031
 0.00573107 0.00473321]
First 20 values of the second tensor:
[0.09786492 0.12614973 0.07085852 0.0160213  0.00960159 0.01806434
 0.00729345 0.00891055 0.00560344 0.02399396 0.01196703 0.00594258
 0.0657875  0.03202351 0.00668285 0.03213002 0.01989424 0.00787695
 0.00487487 0.00441002]
First 20 values of the difference:
[ 0.06947558  0.10443036  0.02998176  0.00277621 -0.00098327  0.01031
  0.00373503  0.01400844  0.01344769  0.05926657  0.02004809  0.00438
  0.0644659   0.02119957  0.00124937  0.02575687  0.03187826  0.01500
  0.0008562   0.00032319]
```