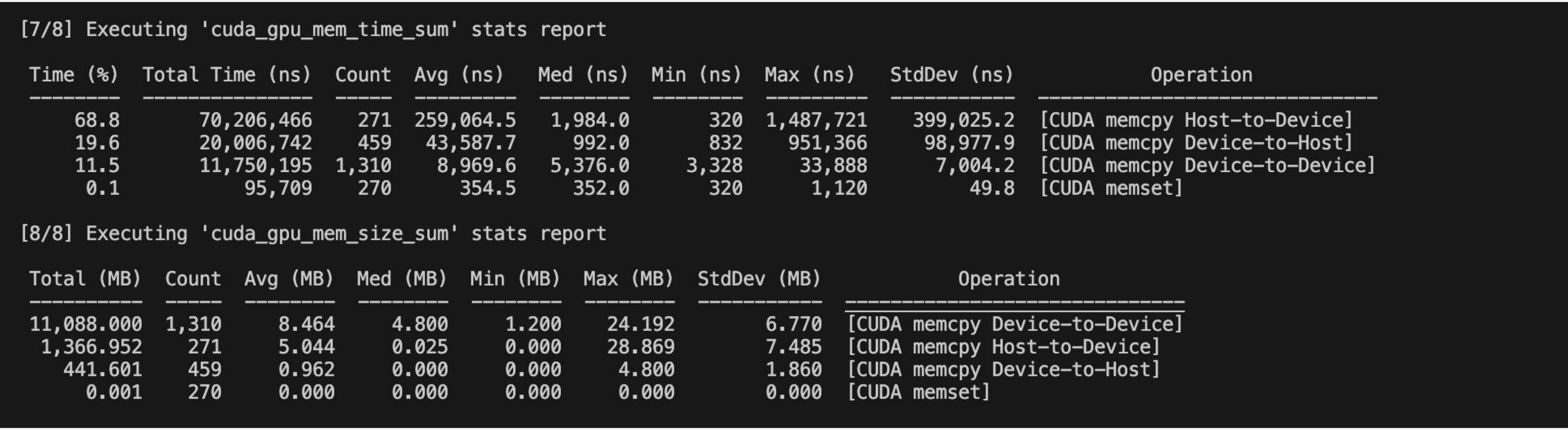
February 13

Running DNNROI Formation With GPU

[12:13:18.917] I [timer] Timer: Total 63.197 wall-sec, 83.694 core-sec

[12:57:13.014] I [timer] Timer: 44.883 wall-sec, 44.959 core-sec: (WireCell::Gen::DepoTransform) "tpc3"

[12:57:13.014] I [timer] Timer: 0.103 wall-sec, 0.261 core-sec: (WireCell::SPNG::CellViews) "tpc3_preface"



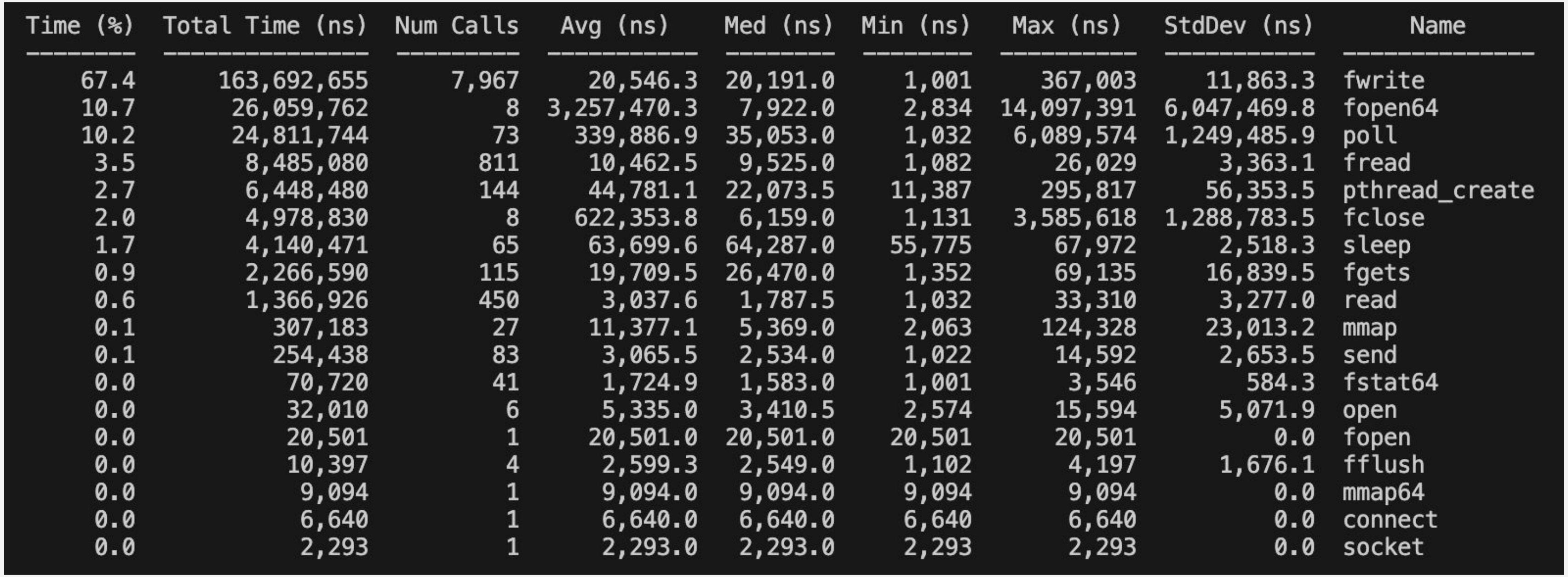
```
[7/8] Executing 'cuda_gpu_mem_time_sum' stats report

 Time (%)  Total Time (ns)  Count  Avg (ns)   Med (ns)  Min (ns)  Max (ns)   StdDev (ns)            Operation
 --------  ---------------  -----  ---------  --------  --------  ---------  -----------  ------------------------------
     68.8       70,206,466    271  259,064.5   1,984.0       320  1,487,721    399,025.2  [CUDA memcpy Host-to-Device]
     19.6       20,006,742    459   43,587.7     992.0       832    951,366     98,977.9  [CUDA memcpy Device-to-Host]
     11.5       11,750,195  1,310    8,969.6   5,376.0     3,328     33,888      7,004.2  [CUDA memcpy Device-to-Device]
      0.1           95,709    270      354.5     352.0       320      1,120         49.8  [CUDA memset]

[8/8] Executing 'cuda_gpu_mem_size_sum' stats report

 Total (MB)  Count  Avg (MB)  Med (MB)  Min (MB)  Max (MB)  StdDev (MB)            Operation
 ----------  -----  --------  --------  --------  --------  -----------  ------------------------------
 11,088.000  1,310     8.464     4.800     1.200    24.192        6.770  [CUDA memcpy Device-to-Device]
  1,366.952    271     5.044     0.025     0.000    28.869        7.485  [CUDA memcpy Host-to-Device]
    441.601    459     0.962     0.000     0.000     4.800        1.860  [CUDA memcpy Device-to-Host]
      0.001    270     0.000     0.000     0.000     0.000        0.000  [CUDA memset]
```

Running DNNROI-Formation with CPU

[12:37:25.542] I [timer] Timer: Total 75.719 wall-sec, 400.407 core-sec

[12:37:25.542] I [timer] Timer: 44.176 wall-sec, 44.180 core-sec: (WireCell::Gen::DepoTransform) "tpc3"

[12:37:25.542] I [timer] Timer: 15.094 wall-sec, 281.177 core-sec: (WireCell::SPNG::CellViews) "tpc3_preface"



```
 Time (%)  Total Time (ns)  Num Calls    Avg (ns)     Med (ns)   Min (ns)    Max (ns)     StdDev (ns)        Name
 --------  ---------------  ---------  -----------  ---------  ---------  -----------  ------------  --------------
     67.4      163,692,655      7,967     20,546.3   20,191.0      1,001      367,003      11,863.3  fwrite
     10.7       26,059,762          8  3,257,470.3    7,922.0      2,834   14,097,391   6,047,469.8  fopen64
     10.2       24,811,744         73    339,886.9   35,053.0      1,032    6,089,574   1,249,485.9  poll
      3.5        8,485,080        811     10,462.5    9,525.0      1,082       26,029       3,363.1  fread
      2.7        6,448,480        144     44,781.1   22,073.5     11,387      295,817      56,353.5  pthread_create
      2.0        4,978,830          8    622,353.8    6,159.0      1,131    3,585,618   1,288,783.5  fclose
      1.7        4,140,471         65     63,699.6   64,287.0     55,775       67,972       2,518.3  sleep
      0.9        2,266,590        115     19,709.5   26,470.0      1,352       69,135      16,839.5  fgets
      0.6        1,366,926        450      3,037.6    1,787.5      1,032       33,310       3,277.0  read
      0.1          307,183         27     11,377.1    5,369.0      2,063      124,328      23,013.2  mmap
      0.1          254,438         83      3,065.5    2,534.0      1,022       14,592       2,653.5  send
      0.0           70,720         41      1,724.9    1,583.0      1,001        3,546         584.3  fstat64
      0.0           32,010          6      5,335.0    3,410.5      2,574       15,594       5,071.9  open
      0.0           20,501          1     20,501.0   20,501.0     20,501       20,501           0.0  fopen
      0.0           10,397          4      2,599.3    2,549.0      1,102        4,197       1,676.1  fflush
      0.0            9,094          1      9,094.0    9,094.0      9,094        9,094           0.0  mmap64
      0.0            6,640          1      6,640.0    6,640.0      6,640        6,640           0.0  connect
      0.0            2,293          1      2,293.0    2,293.0      2,293        2,293           0.0  socket
```

GPU Usage

- GPU Usage = (Time spent by GPU Running Kernels)/ (Total Time Profiling Tool Ran)
  - Calculated from nvtx profiling
- MP Formation
  - GPU Usage during the Cell View → ~3 %
- Inference
  - GPU Usage → ~20%

Will look at it in more detail later

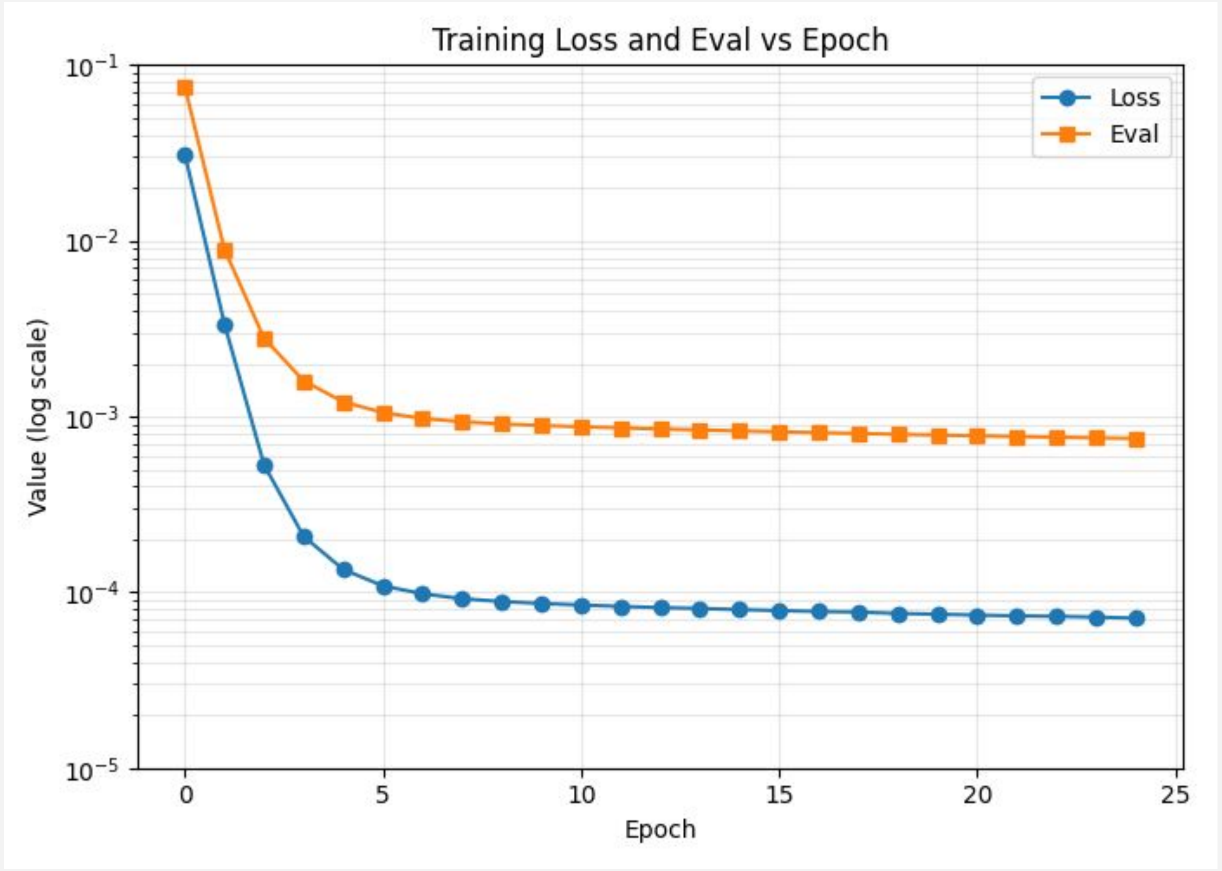
Playing with Data Augmentation

Events 100
Learning Rate: 0.05
Momentum: 0.9
Weight Decay: 0.0005

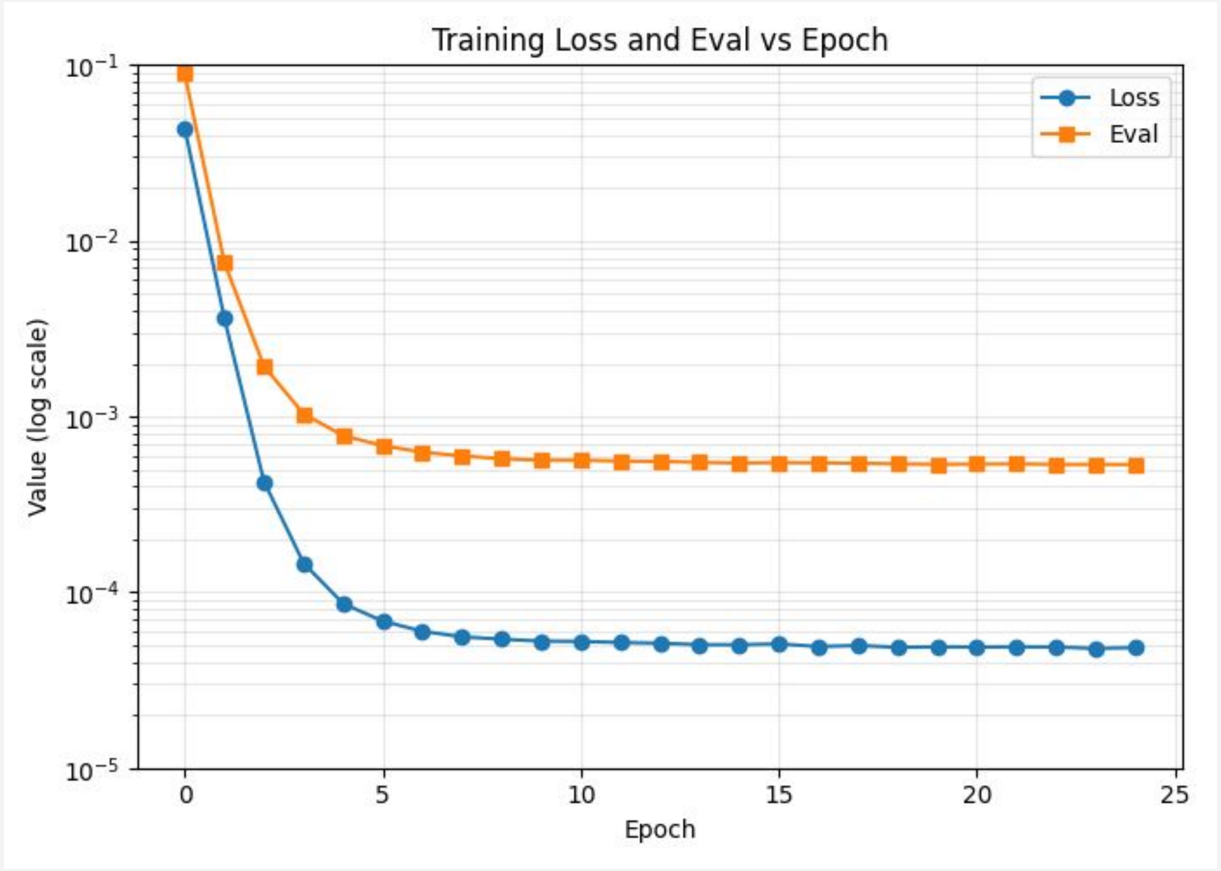
Augmentation And Comparative models

- For each unit sample (features [dense, mp2,mp3] and labels [truth])
- Apply horizontal or vertical flipping [Keeps the tensor shape same between each unit sample]
  - Probability of augmentation at 0.5

No Augmentation



Augmentation

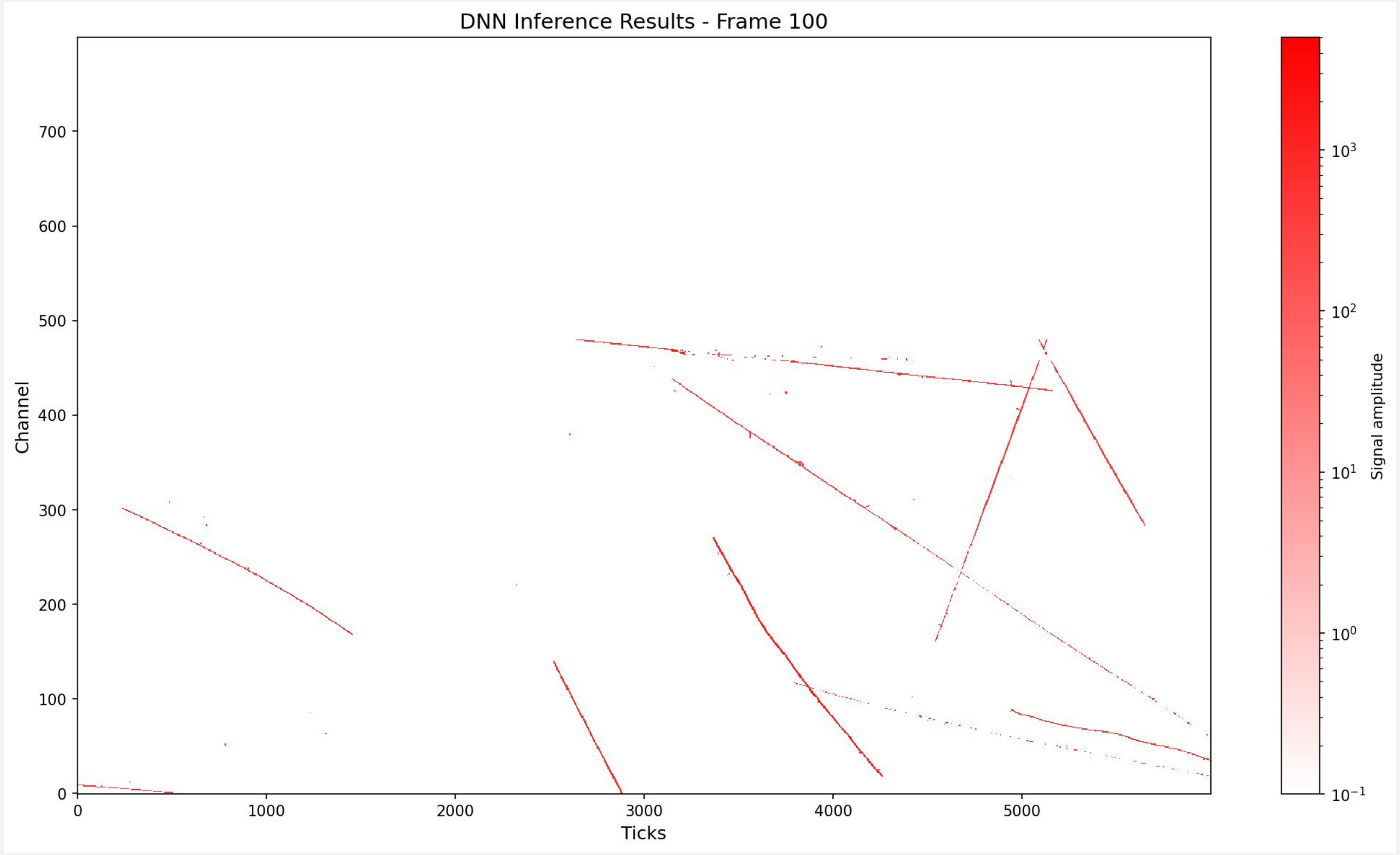


Comparing with existing model being used in SPNG

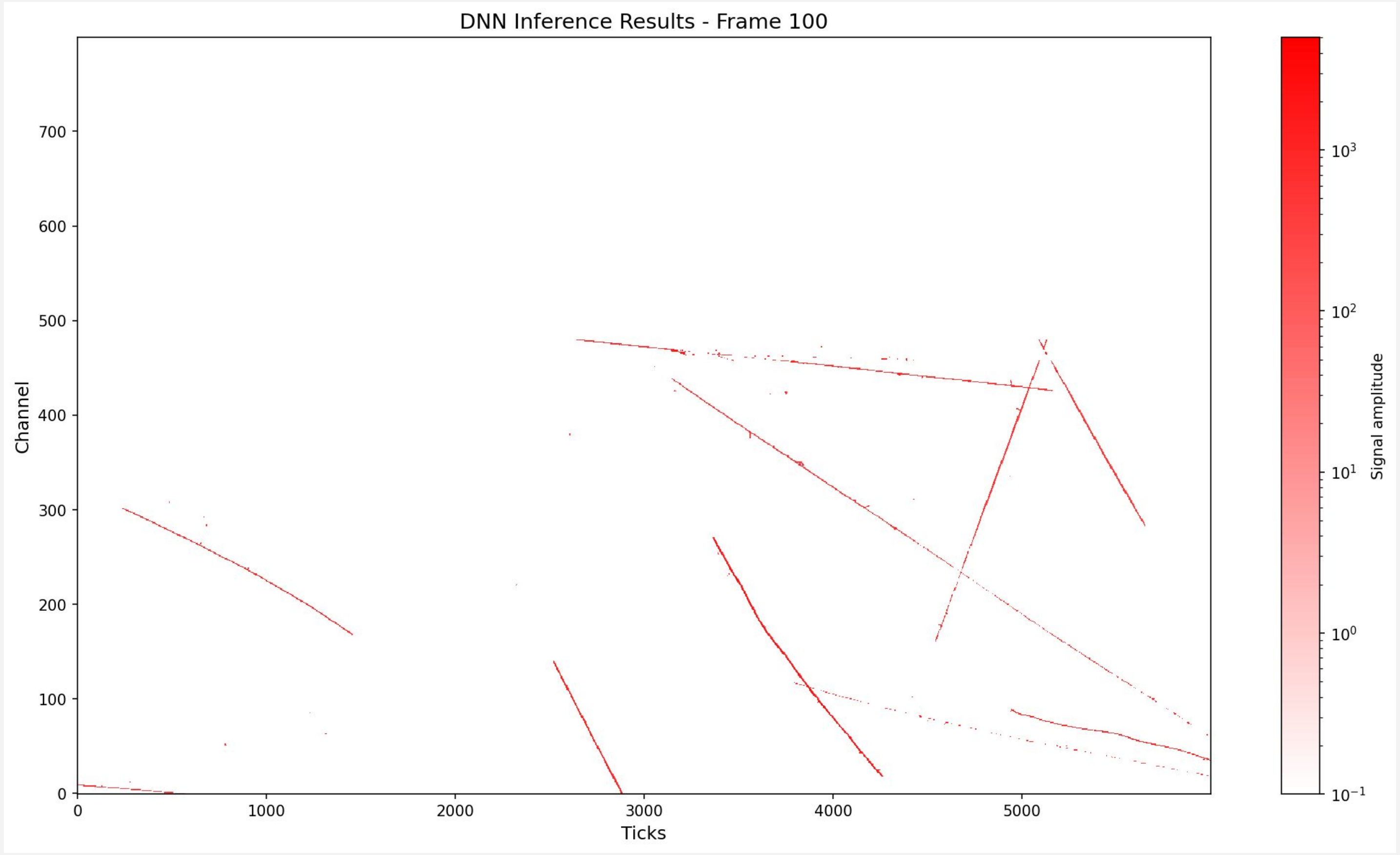
- For each unit sample (features [dense, mp2,mp3] and labels [truth])
- Apply horizontal or vertical flipping [Keeps the tensor shape same between each unit sample]
    - Probability of augmentation at 0.5

- For Comparison I am using the model created by Haiwang [that is the default model used in SPNG currently]
    - For the sake of convenience, I will call it **Haiwang-unet** and **SPNG-unet**
    - unet-l23-cosmic500-e50.ts
- Both models are unet models.
- Both are trained with PDHD simulation (but different events)

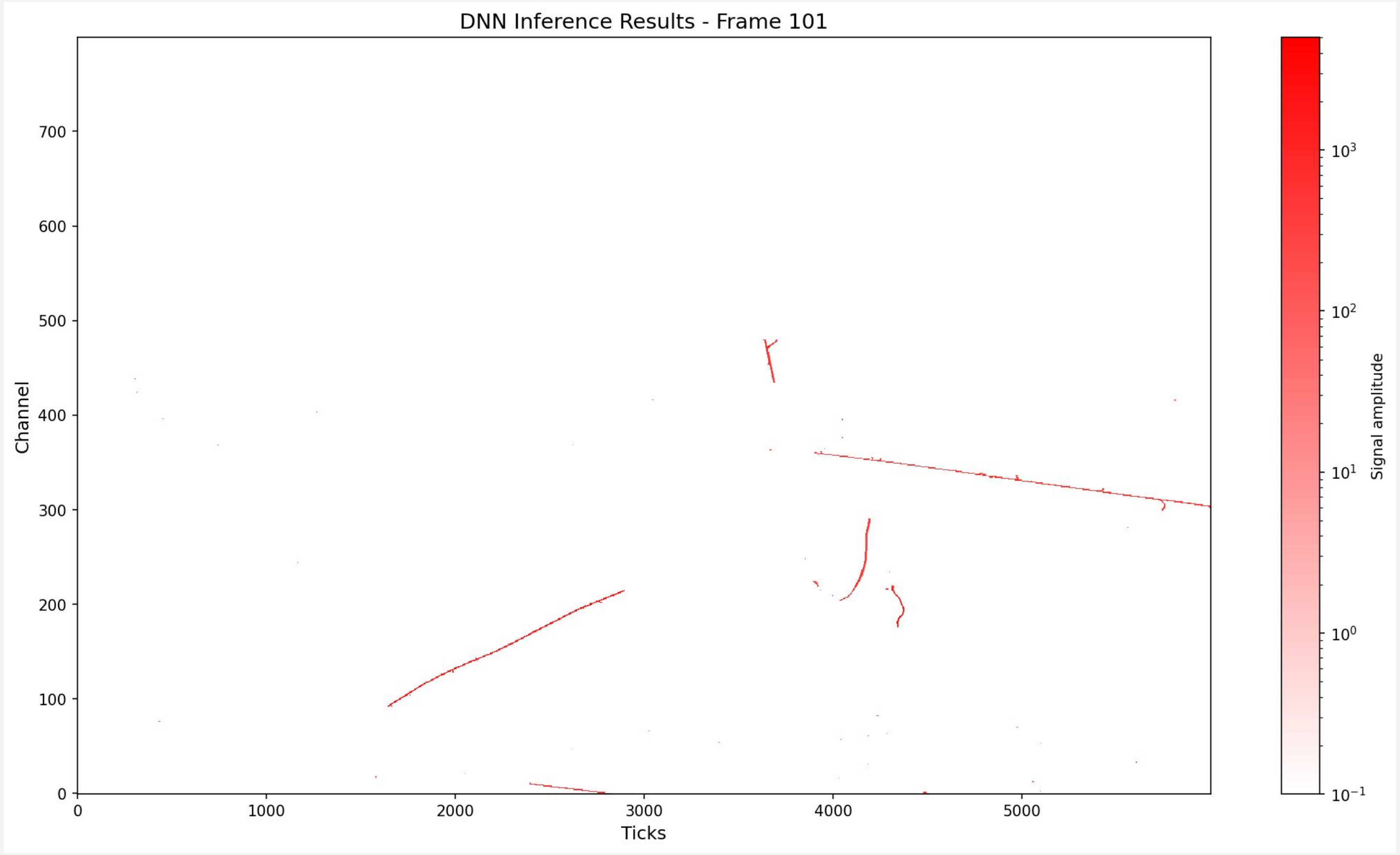
Inference with SPNG Unet (Event 1)



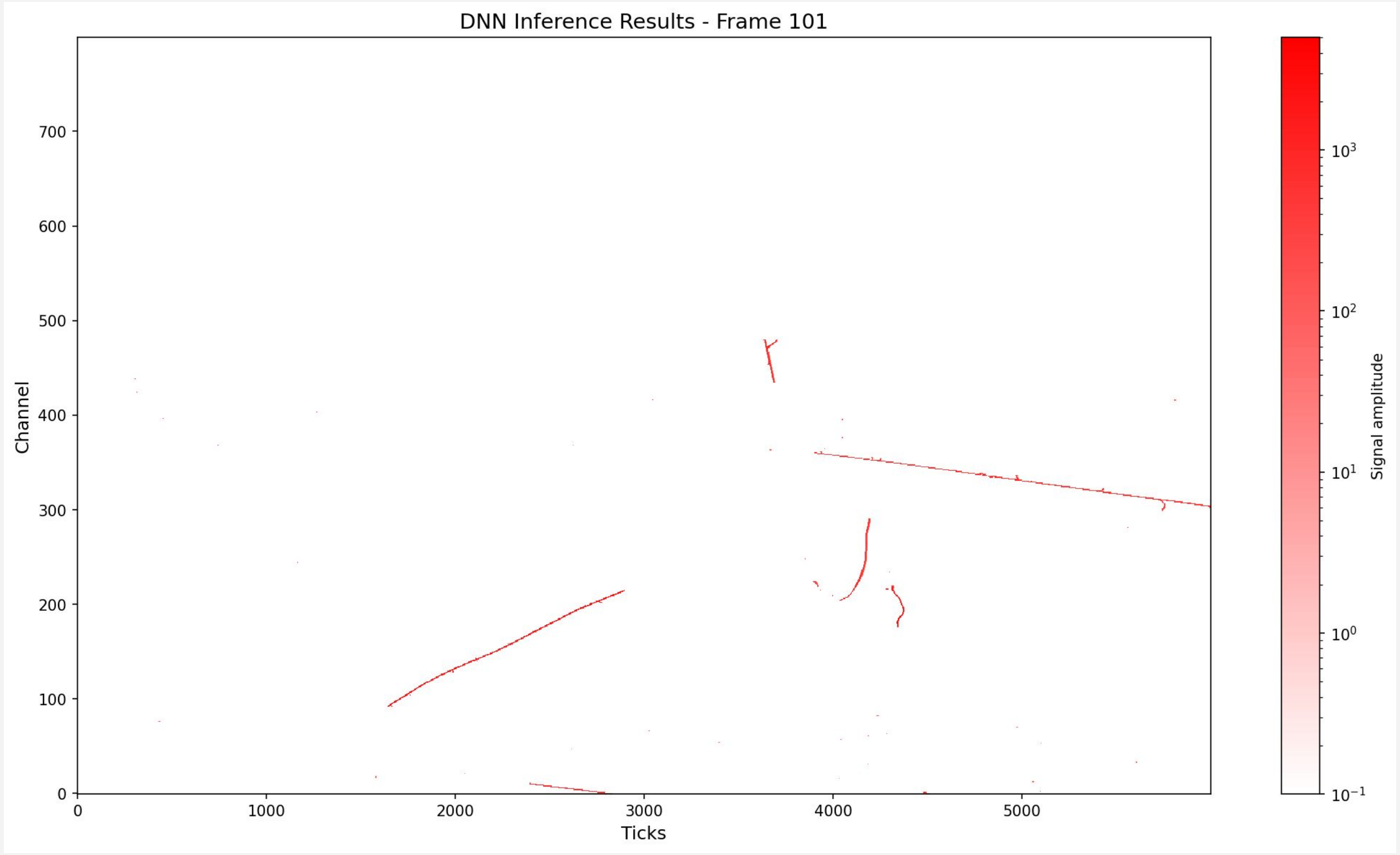
Inference with Haiwang Unet (Event 1)



Inference with SPNG Unet (Event 2)



Inference with Haiwang Unet (Event 2)

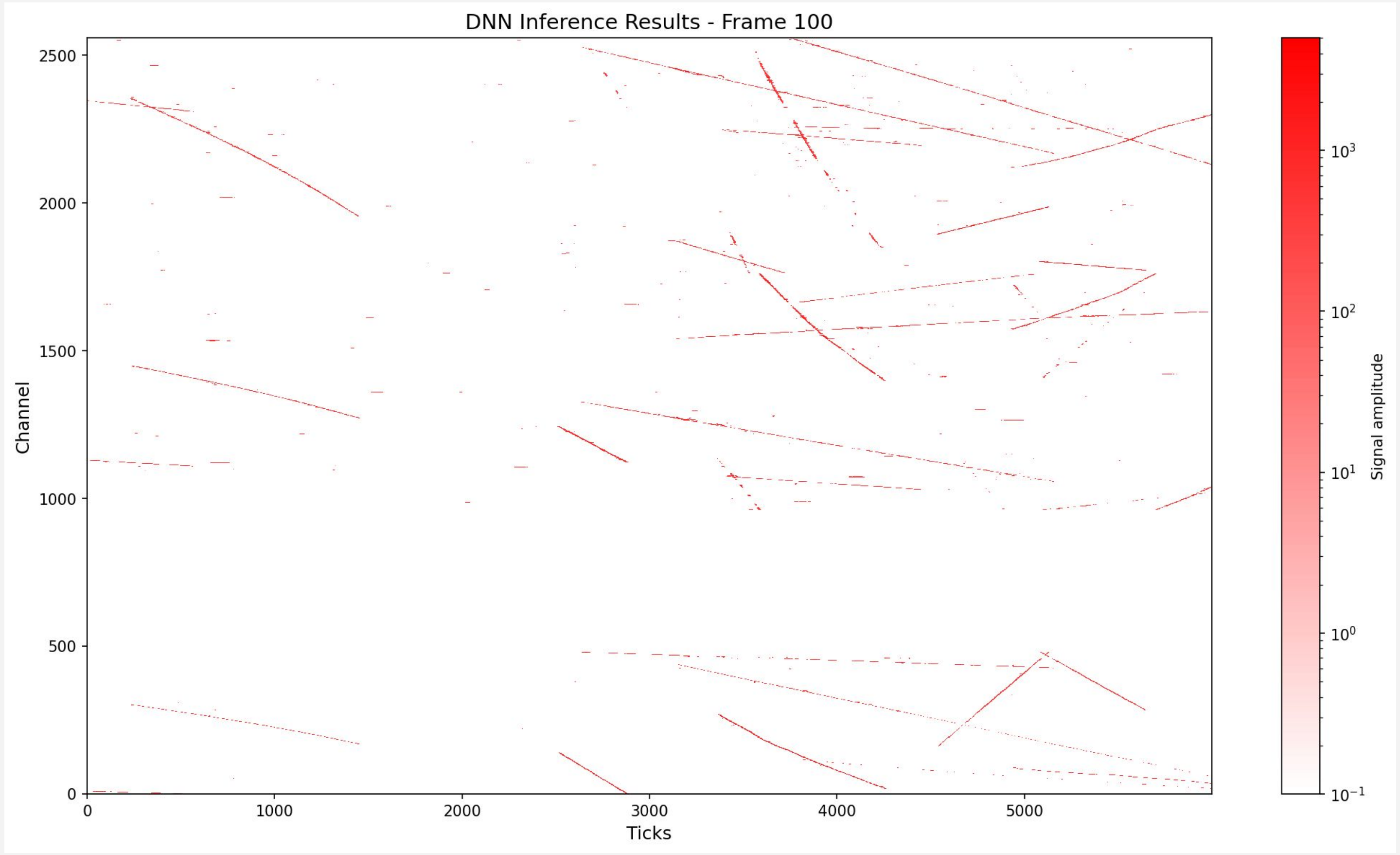


Model trainings (Haiwang's Unet and SPNG Unet)

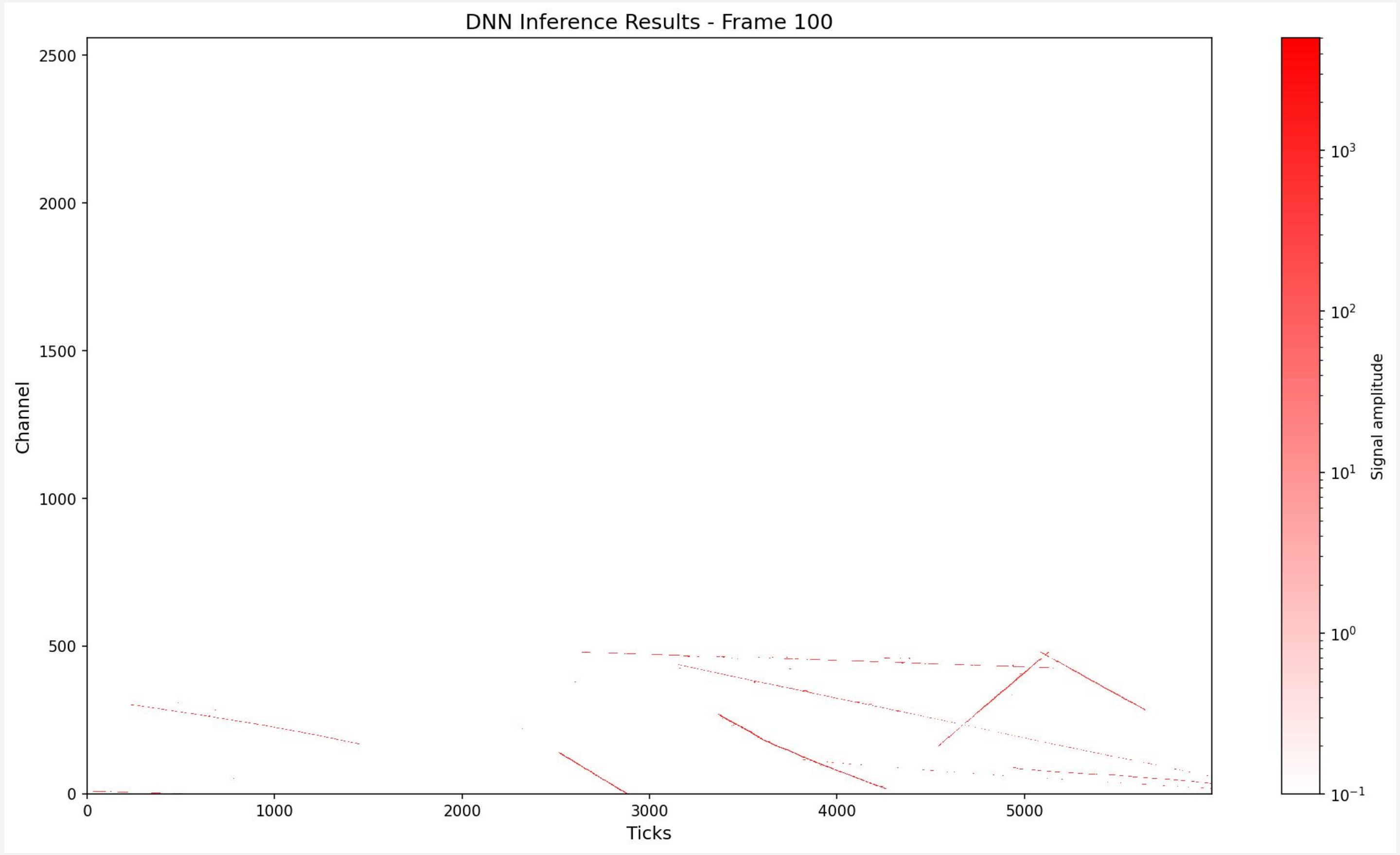
- Outputs after inference using both models are identical (to every single element)
    - Not sure if it is expected
- But the coverage of the models are different
- Probably it has to do with how input data are fed during training

- SPNG Unet
    - Trained with each view (U or V) along with corresponding MP2 and MP3 information
- Haiwang Unet
    - Input data is the whole frame (U,V W1 and W2) with corresponding MP2 and MP3 information
    - MP2/MP3 corresponding to W1 and W2 views being empty

Inference Plot (with Haiwang Unet)



Inference Plot (with SPNG Unet)



Few Other things

- Next: SPDIR Metrics for OSP, SPNG in SnakeMake(?)
- Train with Individual View
- Inference on Individual Views
  - Consistent with current SPNG setup as well
- Workflow for both training and inference in Jake’s Snakemake
  - Uses dnnroi-training.jsonnet for generating training sample in npz
  - Intermediate npz to HDF5 converter for the training/testing
  - Uses test-subgraphs.jsonnet for inference
- Perhaps containerize the whole workflow to be run in the BNL cluster(s)