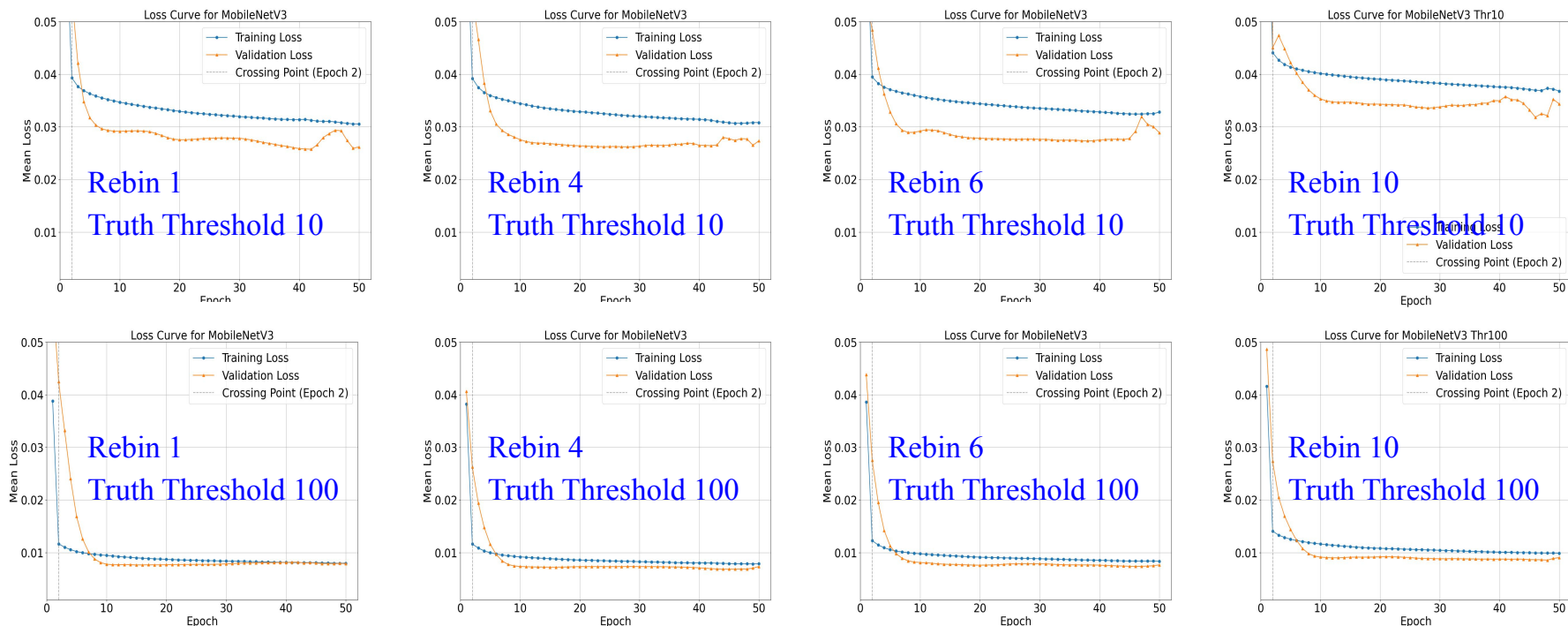




Status report on **DNNROI sigproc**

Hokyeong Nam
Chung-Ang University

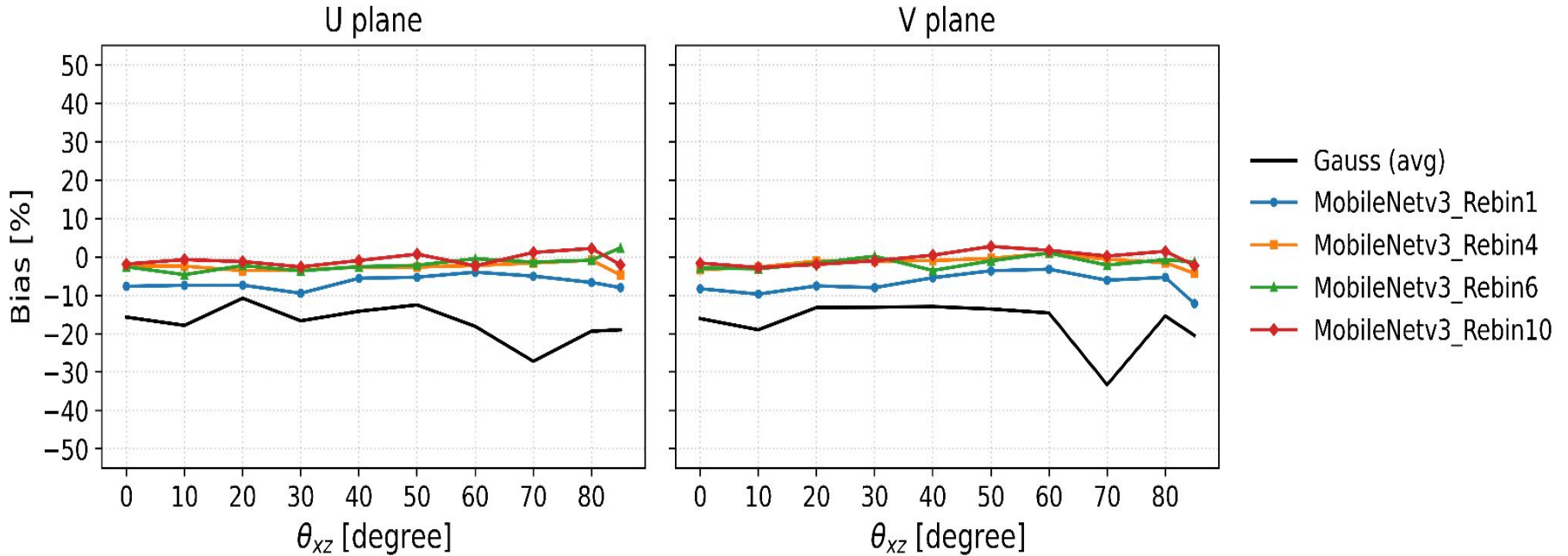
Train. vs. Val. Loss Curve



- MobileUNet V3
- Dataset: cosmic-ray 590 events
- High-resolution images leads lower final training loss
- The threshold 100 showed better performance across rebin factors

Performance evaluation

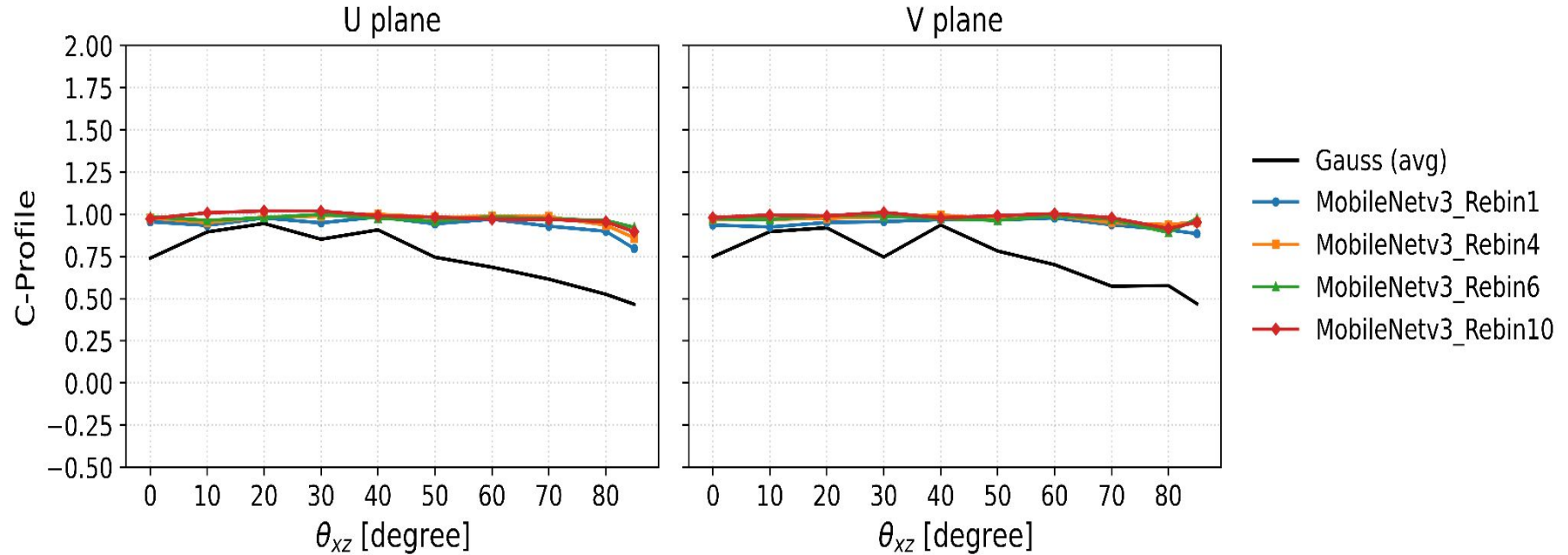
Bias vs θ_{xz} | 1GeV | DNN vs Gauss(avg)



- MobileUNet V3, Truth Threshold = 100
- Dataset: cosmic-ray 590 events

Performance evaluation

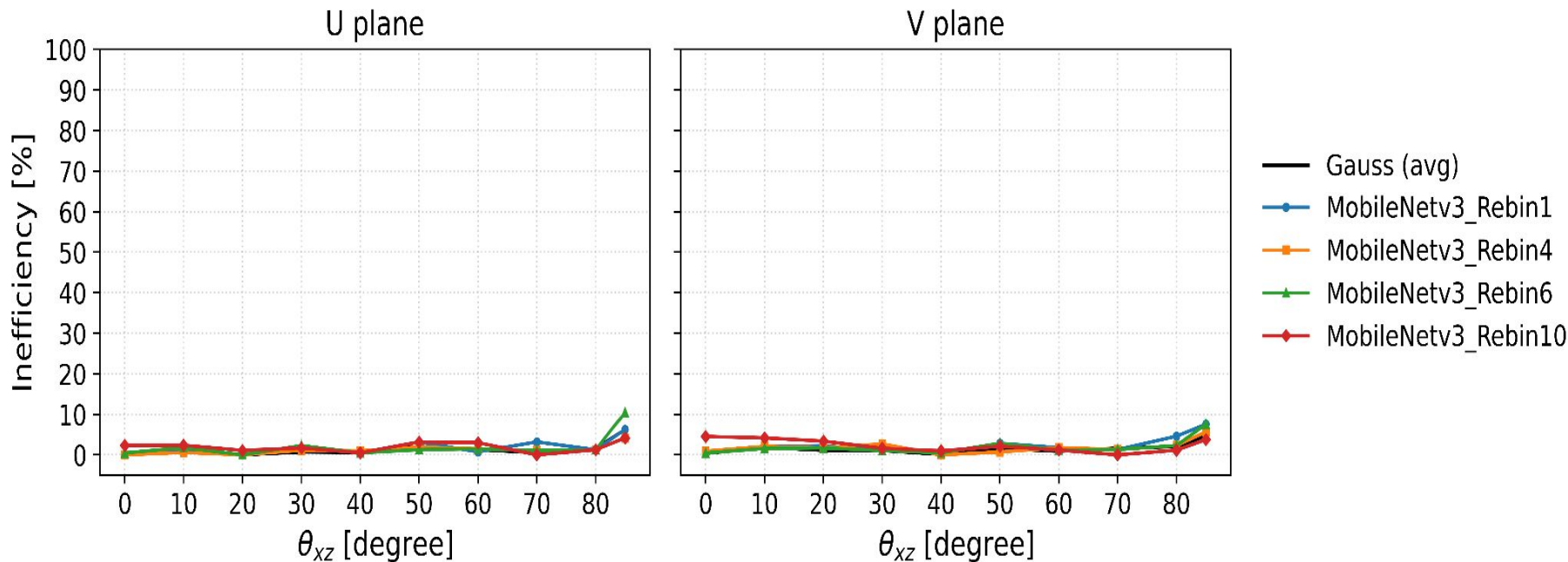
Cprofile vs θ_{xz} | 1GeV | DNN vs Gauss(avg)



- MobileUNet V3, Truth Threshold = 100
- Dataset: cosmic-ray 590 events

Performance evaluation

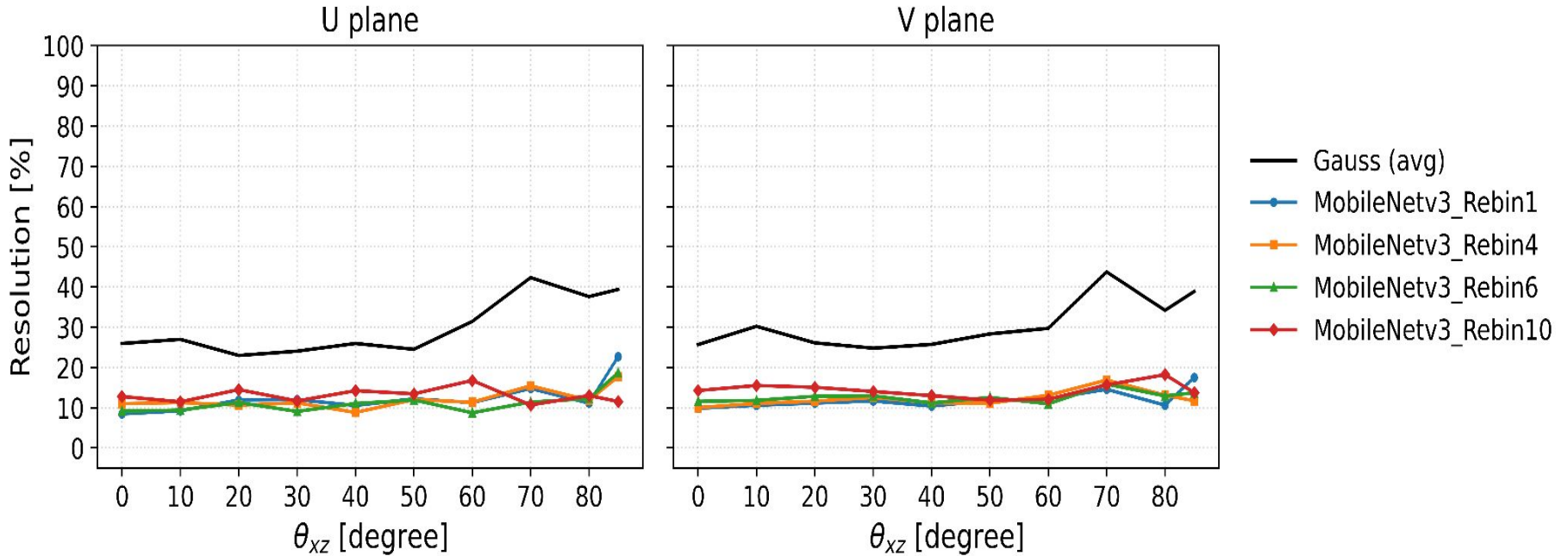
Inefficiency vs θ_{xz} | 1GeV | DNN vs Gauss(avg)



- MobileUNet V3, Truth Threshold = 100
- Dataset: cosmic-ray 590 events

Performance evaluation

Resolution vs θ_{xz} | 1GeV | DNN vs Gauss(avg)



- MobileUNet V3, Truth Threshold = 100
- Dataset: cosmic-ray 590 events

Time/Memory Measurement (Chunking + Rebinning)

	MobileUNet V3 (MB)				UNet (MB)
nchunks	Rebin 1 (full)	Rebin 4	Rebin 6	Rebin 10	
1	8608	4662	4210	3931	6468
2	6000	4032	3861	3723	4962
3	5242	3860	3736	3616	4410
4	4760	3777	3687	3622	4151
5	4539	3721	3641	3658	4021
10	3977	3657	3648	3660	3792

- PD-HD cosmic-ray data: 27673-1

DNN-ROI Memory Consumption Study

```
// decon charge frame to eigen -----  
// (4) Convert Decon charge  
Array::array_xxf decon_charge_eigen;  
{  
    auto traces = Aux::tagged_traces(inframe, m_cfg.decon_charge_tag);  
    mu(fmt::format("4-1 decon tagged_traces (n={})", traces.size()));  
  
    decon_charge_eigen = traces_to_eigen(traces);  
    mu(fmt::format("4-2 decon eigen filled ({}x{})", m_nrows, m_ncols));  
  
    if (decon_charge_eigen.sum() == 0.0) {  
        log->warn("call={} no traces for input tag {}, using zeros", m_sa  
    }  
    else {  
        log->debug("call={} tag={} ntraces={}", m_save_count, m_cfg.decon  
    }  
}
```

- 6-0 to 6-1b: Setup, loading TorchScript
- 0-0: Baseline
- 1-x: Frame → Eigen conversion
- 2-x: Eigen → Tensor conversion
- 3-x: Chunking & Inference
- 6-2 to 6-7:
TorchService input conversion & output conversion
- 4-x: Deconvolution charge → Eigen conversion
- 5-x: ROI application & frame reconstruction

- Include WireCellUtil/MemUsage.h to “DNNROIFinding.cxx” and “TorchService.cxx”
- Printed the RSS memory at each step

DNN-ROI Memory Consumption Study

```
[09:07:29.420] I [ torch ] <TorchService:dnno> MemUsage summary (forward)
MEM: total: size=5.07427e+06K, res=2.42924e+06K increment: size=0K, res=0K TorchService:forward begin (dev=cpu)
MEM: total: size=5.07427e+06K, res=2.42924e+06K increment: size=0K, res=0K 6-2a before from_tensor
MEM: total: size=5.07427e+06K, res=2.42924e+06K increment: size=0K, res=0K 6-2b after from_tensor
MEM: total: size=5.07427e+06K, res=2.42924e+06K increment: size=0K, res=0K 6-3 before module.forward
MEM: total: size=5.07427e+06K, res=2.42924e+06K increment: size=0K, res=0K 6-4 after module.forward
MEM: total: size=5.07427e+06K, res=2.42924e+06K increment: size=0K, res=0K 6-5 before to_tensor
MEM: total: size=5.07427e+06K, res=2.42924e+06K increment: size=0K, res=0K 6-6 after to_tensor

[09:07:29.663] I [ torch ] <TorchService:dnno> MemUsage summary (forward)
MEM: total: size=5.07427e+06K, res=2.42924e+06K increment: size=0K, res=0K TorchService:forward begin (dev=cpu)
MEM: total: size=5.07427e+06K, res=2.42924e+06K increment: size=0K, res=0K 6-2a before from_tensor
MEM: total: size=5.07427e+06K, res=2.42924e+06K increment: size=0K, res=0K 6-2b after from_tensor
MEM: total: size=5.07427e+06K, res=2.42924e+06K increment: size=0K, res=0K 6-3 before module.forward
MEM: total: size=5.07427e+06K, res=2.42924e+06K increment: size=0K, res=0K 6-4 after module.forward
MEM: total: size=5.07427e+06K, res=2.42924e+06K increment: size=0K, res=0K 6-5 before to_tensor
MEM: total: size=5.07427e+06K, res=2.42924e+06K increment: size=0K, res=0K 6-6 after to_tensor

[09:07:29.731] I [ torch ] <DNNROI:findingidnnroi2v> MemUsage summary for call=0
MEM: total: size=5.04237e+06K, res=2.39739e+06K increment: size=0K, res=0K call=0 begin
MEM: total: size=5.04237e+06K, res=2.39739e+06K increment: size=0K, res=0K 0-0 frame baseline
MEM: total: size=5.04237e+06K, res=2.39739e+06K increment: size=0K, res=0K 1-0 [loose_lf2] start
MEM: total: size=5.04237e+06K, res=2.39739e+06K increment: size=0K, res=0K 1-1 [loose_lf2] traces collected (n=0)
MEM: total: size=5.04237e+06K, res=2.39739e+06K increment: size=0K, res=0K 1-2 [loose_lf2] eigen allocated+filled (800x6000)
MEM: total: size=5.04237e+06K, res=2.39739e+06K increment: size=0K, res=0K 1-3 [loose_lf2] input scale/offset applied
MEM: total: size=5.04237e+06K, res=2.39739e+06K increment: size=0K, res=0K 1-4 [loose_lf2] downsampled by 10
MEM: total: size=5.04237e+06K, res=2.39739e+06K increment: size=0K, res=0K 1-5 [loose_lf2] pushed into ch_eigen (size=1)
MEM: total: size=5.04237e+06K, res=2.39739e+06K increment: size=0K, res=0K 1-0 [mp2_roi2] start
MEM: total: size=5.04237e+06K, res=2.39739e+06K increment: size=0K, res=0K 1-1 [mp2_roi2] traces collected (n=2560)
MEM: total: size=5.04237e+06K, res=2.39739e+06K increment: size=0K, res=0K 1-2 [mp2_roi2] eigen allocated+filled (800x6000)
MEM: total: size=5.04237e+06K, res=2.39739e+06K increment: size=0K, res=0K 1-3 [mp2_roi2] input scale/offset applied
MEM: total: size=5.04237e+06K, res=2.39739e+06K increment: size=0K, res=0K 1-4 [mp2_roi2] downsampled by 10
MEM: total: size=5.04237e+06K, res=2.39739e+06K increment: size=0K, res=0K 1-5 [mp2_roi2] pushed into ch_eigen (size=2)
MEM: total: size=5.04237e+06K, res=2.39739e+06K increment: size=0K, res=0K 1-0 [mp3_roi2] start
MEM: total: size=5.04237e+06K, res=2.39739e+06K increment: size=0K, res=0K 1-1 [mp3_roi2] traces collected (n=2560)
MEM: total: size=5.04237e+06K, res=2.39739e+06K increment: size=0K, res=0K 1-2 [mp3_roi2] eigen allocated+filled (800x6000)
MEM: total: size=5.04237e+06K, res=2.39739e+06K increment: size=0K, res=0K 1-3 [mp3_roi2] input scale/offset applied
MEM: total: size=5.04237e+06K, res=2.39739e+06K increment: size=0K, res=0K 1-4 [mp3_roi2] downsampled by 10
MEM: total: size=5.04237e+06K, res=2.39739e+06K increment: size=0K, res=0K 1-5 [mp3_roi2] pushed into ch_eigen (size=3)
MEM: total: size=5.04237e+06K, res=2.39739e+06K increment: size=0K, res=0K 2-1 from_blob tag#0 (600x800)
MEM: total: size=5.04237e+06K, res=2.39739e+06K increment: size=0K, res=0K 2-1 from_blob tag#1 (600x800)
MEM: total: size=5.04237e+06K, res=2.39739e+06K increment: size=0K, res=0K 2-1 from_blob tag#2 (600x800)
MEM: total: size=5.04237e+06K, res=2.39739e+06K increment: size=0K, res=0K 2-2 stack(ch,0)
MEM: total: size=5.04237e+06K, res=2.39739e+06K increment: size=0K, res=0K 2-3 transpose(1,2)
MEM: total: size=5.04237e+06K, res=2.39739e+06K increment: size=0K, res=0K 2-4 batch stack -> {1,ntags,nchannels,nticks}
MEM: total: size=5.04237e+06K, res=2.39739e+06K increment: size=0K, res=0K 3-1 chunk view created (N=5)
MEM: total: size=5.04237e+06K, res=2.39739e+06K increment: size=0K, res=0K 3-2 chunk0 cloned(if needed)+wrapped
MEM: total: size=5.04237e+06K, res=2.39739e+06K increment: size=0K, res=0K 3-3 chunk0 to_tensor
MEM: total: size=5.04237e+06K, res=2.39739e+06K increment: size=0K, res=0K 3-4 chunk0 forward done
MEM: total: size=5.04237e+06K, res=2.39739e+06K increment: size=0K, res=0K 3-5 chunk0 from_tensor (to cpu tensor)
MEM: total: size=5.04237e+06K, res=2.39739e+06K increment: size=0K, res=0K 3-6 chunk0 pushed(clone) into outputs (size=1)
MEM: total: size=5.04237e+06K, res=2.39739e+06K increment: size=0K, res=0K 3-2 chunk1 cloned(if needed)+wrapped
MEM: total: size=5.04237e+06K, res=2.39739e+06K increment: size=0K, res=0K 3-3 chunk1 to_tensor
MEM: total: size=5.04237e+06K, res=2.39739e+06K increment: size=0K, res=0K 3-4 chunk1 forward done
MEM: total: size=5.04237e+06K, res=2.39739e+06K increment: size=0K, res=0K 3-5 chunk1 from_tensor (to cpu tensor)
MEM: total: size=5.04237e+06K, res=2.39739e+06K increment: size=0K, res=0K 3-6 chunk1 pushed(clone) into outputs (size=2)
MEM: total: size=5.04237e+06K, res=2.39739e+06K increment: size=0K, res=0K 3-2 chunk2 cloned(if needed)+wrapped
MEM: total: size=5.04237e+06K, res=2.39739e+06K increment: size=0K, res=0K 3-3 chunk2 to_tensor
MEM: total: size=5.04237e+06K, res=2.39739e+06K increment: size=0K, res=0K 3-4 chunk2 forward done
MEM: total: size=5.04237e+06K, res=2.39739e+06K increment: size=0K, res=0K 3-5 chunk2 from_tensor (to cpu tensor)
MEM: total: size=5.04237e+06K, res=2.39739e+06K increment: size=0K, res=0K 3-6 chunk2 pushed(clone) into outputs (size=3)
MEM: total: size=5.04237e+06K, res=2.39739e+06K increment: size=0K, res=0K 3-2 chunk3 cloned(if needed)+wrapped
MEM: total: size=5.04237e+06K, res=2.39739e+06K increment: size=0K, res=0K 3-3 chunk3 to_tensor
MEM: total: size=5.04237e+06K, res=2.39739e+06K increment: size=0K, res=0K 3-4 chunk3 forward done
MEM: total: size=5.04237e+06K, res=2.39739e+06K increment: size=0K, res=0K 3-5 chunk3 from_tensor (to cpu tensor)
MEM: total: size=5.04237e+06K, res=2.39739e+06K increment: size=0K, res=0K 3-6 chunk3 pushed(clone) into outputs (size=4)
MEM: total: size=5.04237e+06K, res=2.39739e+06K increment: size=0K, res=0K 3-2 chunk4 cloned(if needed)+wrapped
MEM: total: size=5.04237e+06K, res=2.39739e+06K increment: size=0K, res=0K 3-3 chunk4 to_tensor
MEM: total: size=5.04237e+06K, res=2.39739e+06K increment: size=0K, res=0K 3-4 chunk4 forward done
MEM: total: size=5.04237e+06K, res=2.39739e+06K increment: size=0K, res=0K 3-5 chunk4 from_tensor (to cpu tensor)
MEM: total: size=5.04237e+06K, res=2.39739e+06K increment: size=0K, res=0K 3-6 chunk4 pushed(clone) into outputs (size=5)

MEM: total: size=5.08177e+06K, res=2.43682e+06K increment: size=0K, res=384K 4-2 decon eigen filled (800x6000)
MEM: total: size=5.13802e+06K, res=2.49298e+06K increment: size=56252K, res=56152K 5-2 mask applied (with transpose)

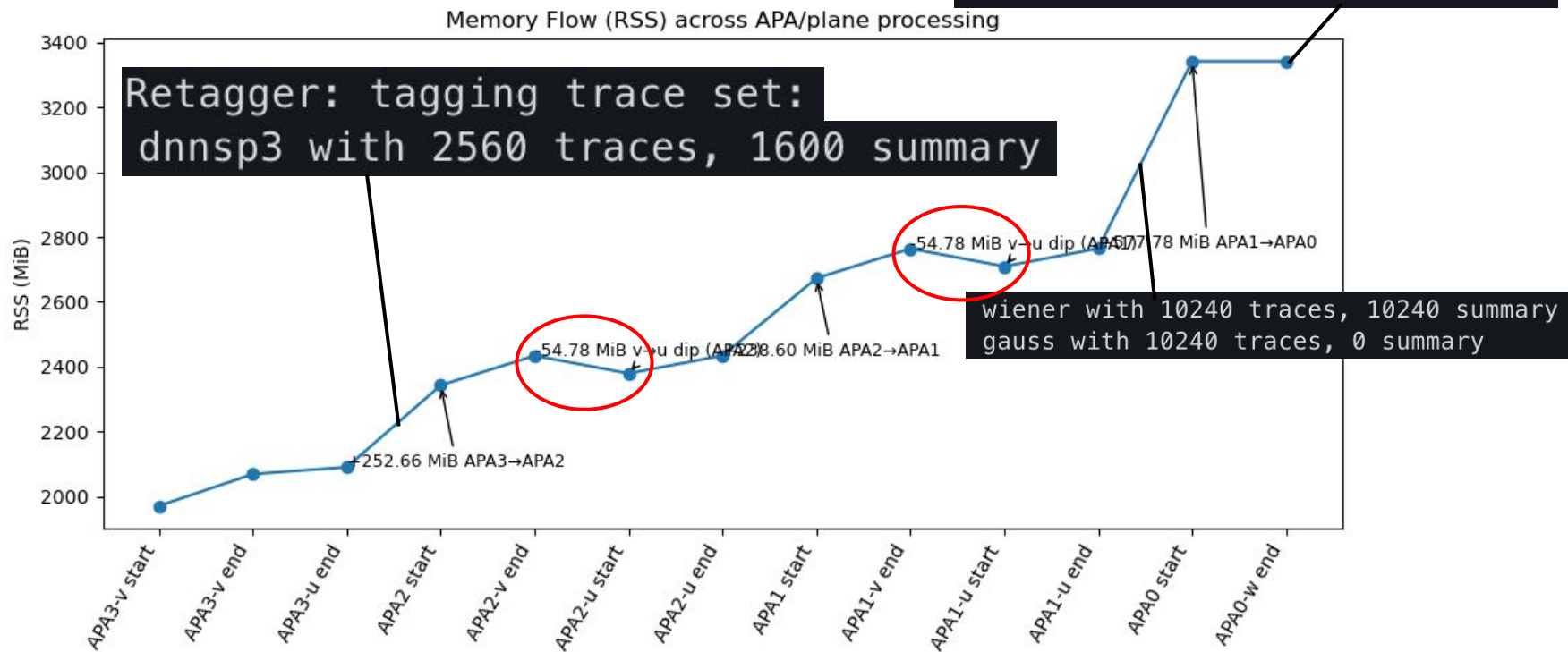
MEM: total: size=5.13802e+06K, res=2.49254e+06K increment: size=0K, res=0K 5-5 eigen to_traces (output traces built)
MEM: total: size=5.13802e+06K, res=2.49254e+06K increment: size=0K, res=0K 5-6 frame built & tagged
```

- Fraction of logs for
 - MobileUNet V3, rebin = 10, nchunks = 5
 - PD-HD cosmic-ray data: 27673-1
- This study revealed decon to eigen conversion and apply ROI mask require ~60 MB of additional memory
- If so, where the other memory increment comes from?

DNN-ROI Memory Consumption Study

Segment	Forward Start RSS (MiB)	End of Call RSS (MiB)	Change (same APA, MiB)	Change (previous plane, MiB)
APA4-V	1970.68	2068.87	+98.19	—
APA4-U	2068.87	2090.29	+21.42	—
APA4 → APA3	2342.95	—	—	+252.66
APA3-V	2342.95	2434.12	91.17	—
V → U transition	2379.34	—	—	−54.78
APA3-U	2379.34	2434.55	+55.21	—
APA3 → APA2	2673.15	—	—	+238.60
APA2-V	2673.15	2764.39	+91.24	—
V → U transition	2709.61	—	—	−54.78
APA2-U	2709.61	2764.82	+55.21	—
APA2 → APA1	3342.60	—	—	+577.78
APA1-W	3342.60	3342.60	—	—

DNN-ROI Memory Consumption Study



- MobileUNet V3, rebin = 10, nchunks = 5
- PD-HD cosmic-ray data: 27673-1

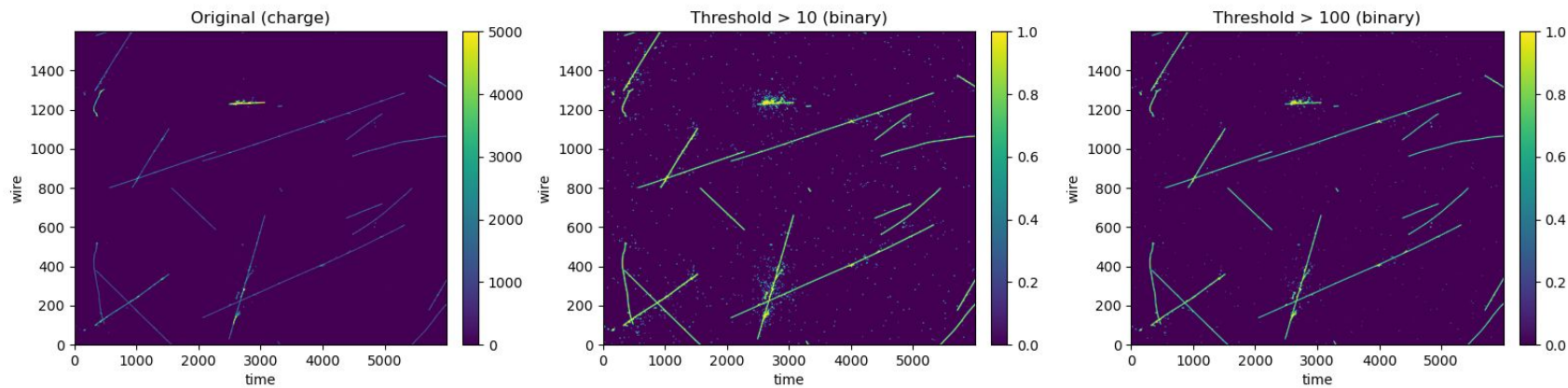
- ~ 1.2 GiB of memory comes from Retagger
- Study for different model & rebin & chunking?

Plan

- Padding to the truth ROI → check the 1D waveform and evaluate the results
 - Apply truth threshold = 100 or more →
reduce noise-like signals and narrow the ROI window in time axis
 - Loop over wire channels to identify continuous signal along time axis
 - Depending on the padding factor (e.g. $1 \rightarrow \pm 1$ time tick, $7 \rightarrow \pm 7$ time tick), include additional time ticks to recover the subtracted ROI window
- Snakemake workflow and metric vs. ideal MIP track direction (SPDIR plots)
- Replace WC standalone simulation with LArWC for single track simulation
- Revise the script for 2D waveform evolution plot to include
 - Charge cut applied to DNN SP results for imaging
 - Raw, after noise filter results

Back Up

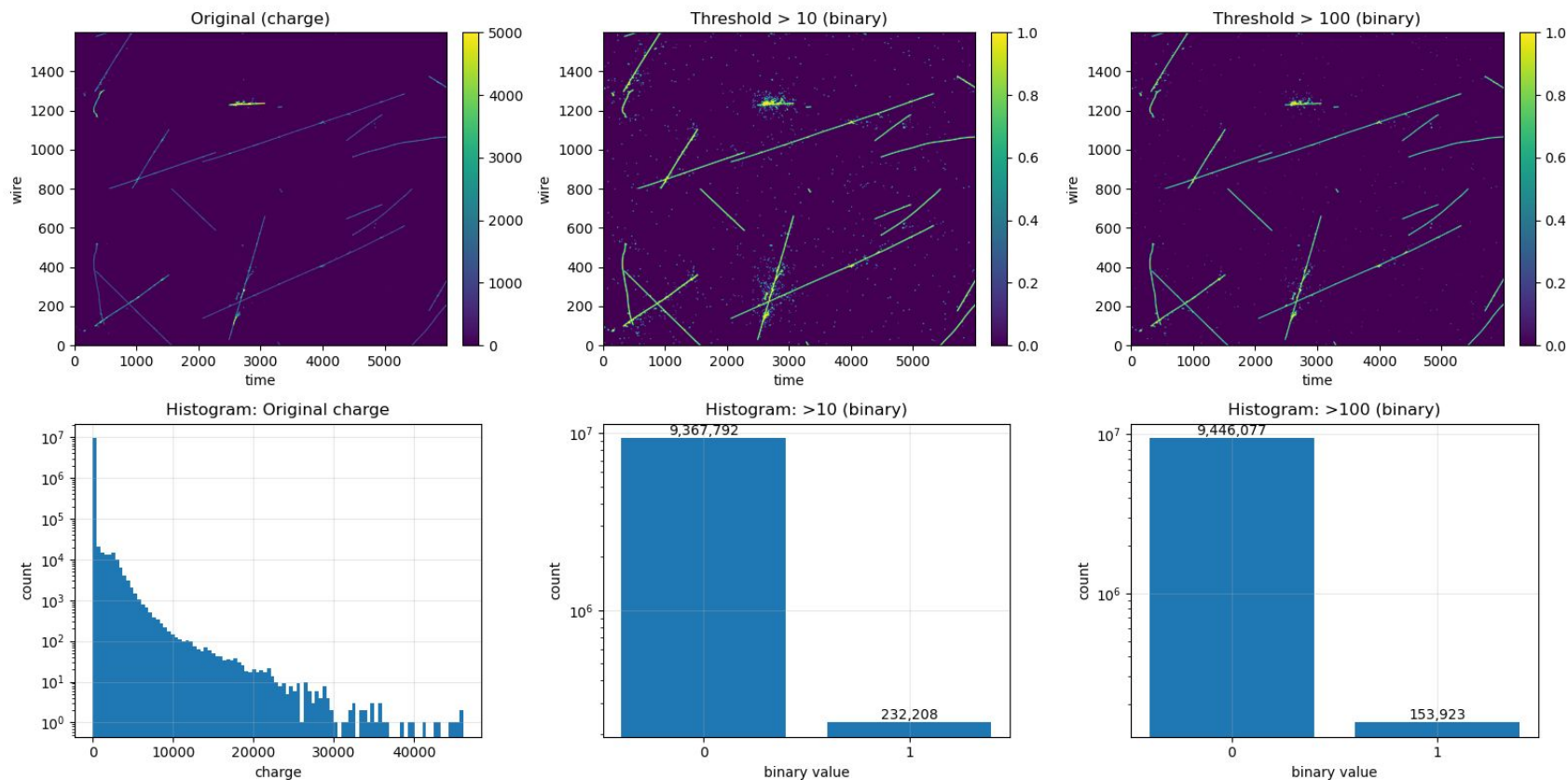
True ROI definition in training script



True ROI: Convert continuous charge map into binary mask for training (for BCE loss)

- Raw Truth Data: each pixel = deposited charge at (wire, time)
- Rebin & Crop: downsample in time using rebin factor, and select region for training
- Thresholding over bin content (truth_th in code):
 - If charge > threshold \rightarrow mask as signal (1)
 - If charge \leq threshold \rightarrow background (0)
- Low threshold \rightarrow more pixels labeled as signal (recall \uparrow , precision \downarrow)
- High threshold \rightarrow fewer pixels labeled as signal (recall \downarrow , precision \uparrow)

True ROI definition in training script



Model Comparison - Network Architectures

Model	Encoder	Decoder	Skip Connection	Downsampling Depth	Activations	
					Encoder	Decoder
UNet	Convs	4 Convs	Yes	1/16	ReLU	ReLU
MobileNetV2-UNet	Depthwise separable	2 Convs	None	1/32	ReLU6	ReLU
MobileNetV3-UNet	Depthwise separable + SE	4 Convs	Yes	1/32	h-swish + ReLU	ReLU
Transformer-UNet	Convs + Transformer bottleneck	4 Convs	Yes	1/16	ReLU	ReLU
					GELU (transformer)	

- Training dataset: 590 cosmic-ray events
- Optimizer: SGD (Stochastic Gradient Descent)
- Learning rate: 0.1
- Early stopping: Enabled
- Train/Val split: 0.9/0.1
- Loss: BCELoss (Binary Cross-Entropy Loss)
- Number of epochs : 50
- Output activation function: Sigmoid

Training was carried out on the WC Cluster using an NVIDIA GeForce RTX 4090 GPU (24 GB)

DNN-ROI Performance Evaluation

- For track events, three metrics are used: Bias, Resolution, and Inefficiency

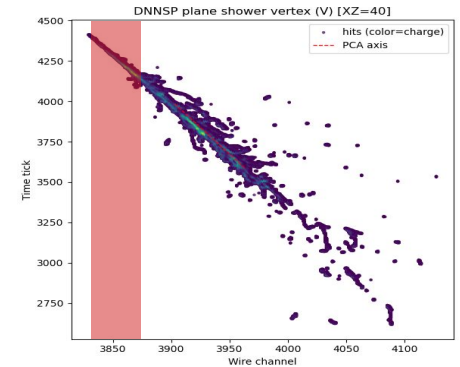
$$\text{Bias} = 100 \times \left(\left\langle \frac{Q_{reco}}{Q_{truth}} \right\rangle - 1 \right) \quad \text{Resolution} = 100 \times \frac{\text{RMS} \left(\frac{Q_{reco}}{Q_{truth}} \right)}{\left\langle \frac{Q_{reco}}{Q_{truth}} \right\rangle} \quad \text{Inefficiency} = 100 \times \frac{\text{Number of bad channels}}{\text{Number of valid truth channels}}$$

- For shower events, a charge profile based on vertex information was added as the fourth metric:
 - Sum the charge along the shower direction up to 42 wire channels ($\approx 1-2$ radiation lengths)
 - Compare the reconstructed-to-truth ratio charge ratio

$$Q_{method} = \sum_{w \in W} Q_{method}(w) \quad R_{cprofile} = \frac{Q_{reco}}{Q_{truth}}$$

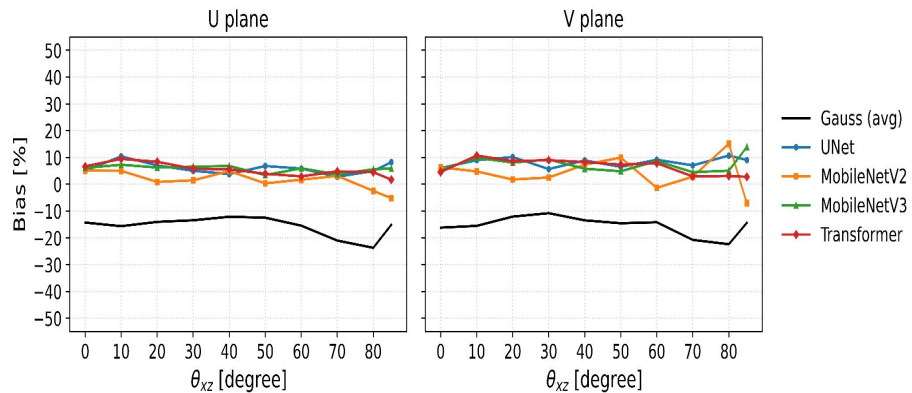
- Samples were generated with

- Detector configuration: ProtoDUNE - Horizontal Drift (PD-HD)
- XZ angle: $0^\circ, 10^\circ, 20^\circ, 30^\circ, 40^\circ, 50^\circ, 60^\circ, 70^\circ, 80^\circ, 85^\circ$
- Shower energies: 100 MeV, 500 MeV, 1 GeV, 2 GeV, 3 GeV, 5 GeV
- Software: WCT standalone (Tracks), LAr-WCT (Showers)

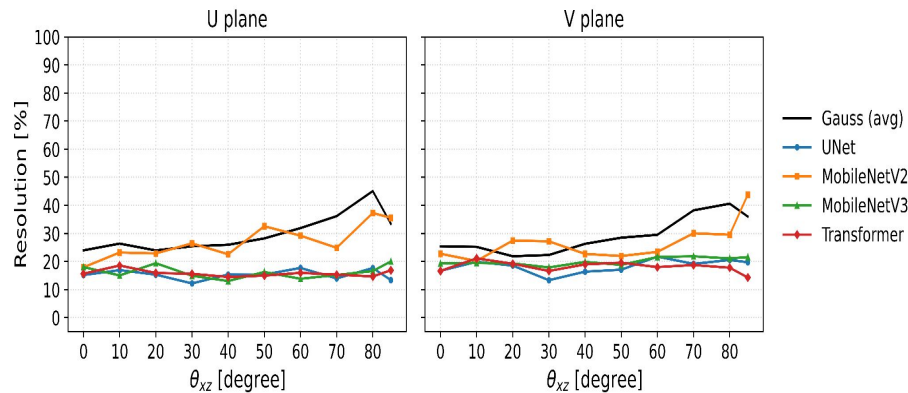


Single Shower Evaluation (Thr 100, 1 GeV)

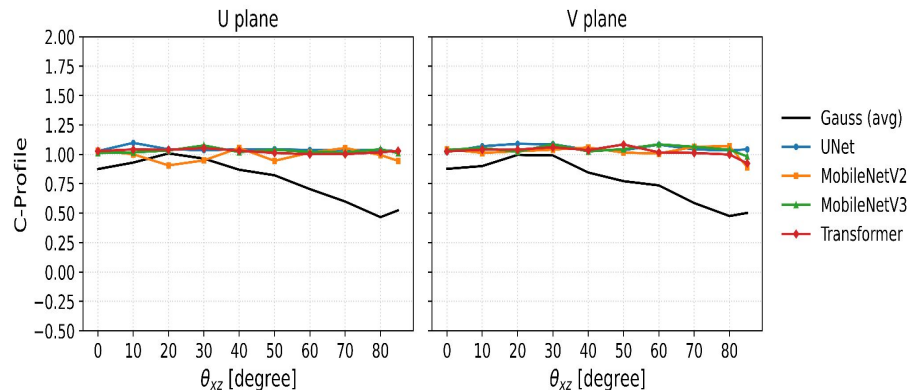
Bias vs θ_{xz} | 1GeV | DNN vs Gauss(avg)



Resolution vs θ_{xz} | 1GeV | DNN vs Gauss(avg)



Cprofile vs θ_{xz} | 1GeV | DNN vs Gauss(avg)



Inefficiency vs θ_{xz} | 1GeV | DNN vs Gauss(avg)

