Status report on

DNN SigProc in PD-HD

Hokyeong Nam
Chung-Ang University

Outline

- Comparison of DNN Models
- DNN-ROI Performance Evaluation
    - Single Track Events
    - Single Shower Events
- Time/Memory Usage Measurement
- Summary

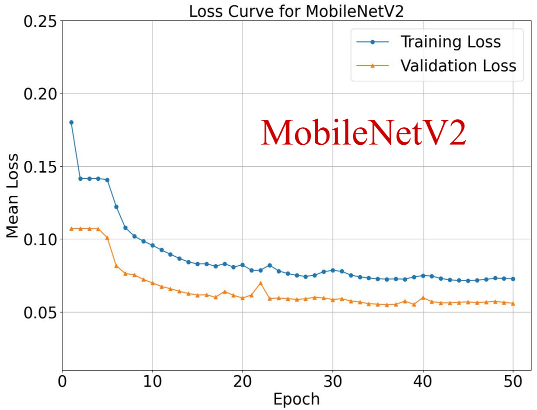
Model Comparison - Network Architectures

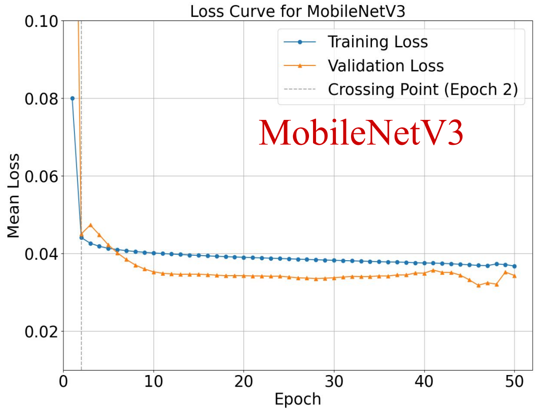
Model	Encoder	Decoder	Skip Connection	Downsampling Depth	Activations	
					Encoder	Decoder
UNet	Convs	4 Convs	Yes	1/16	ReLU	ReLU
MobileNetV2-UNet	Depthwise separable	2 Convs	None	1/32	ReLU6	ReLU
MobileNetV3-UNet	Depthwise separable + SE	4 Convs	Yes	1/32	h-swish + ReLU	ReLU
Transformer-UNet	Convs + Transformer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

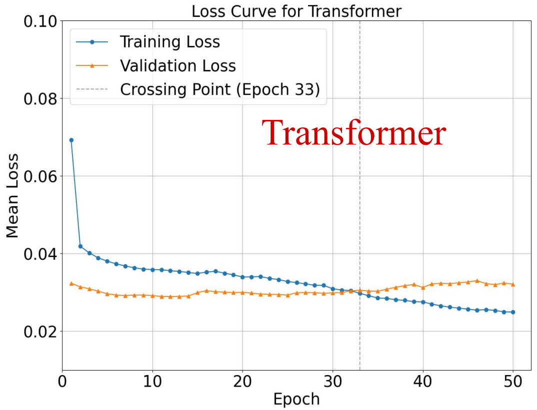
- Training dataset: 590 cosmic-ray events
- Optimizer: SGD (Stochastic Gradient Descent)
- Learning rate: 0.1
- Early stopping: Enabled
- Train/Val split: 0.9/0.1
- Loss: BCELoss (Binary Cross-Entropy Loss)
- Number of epochs : 50
- Output activation function: Sigmoid

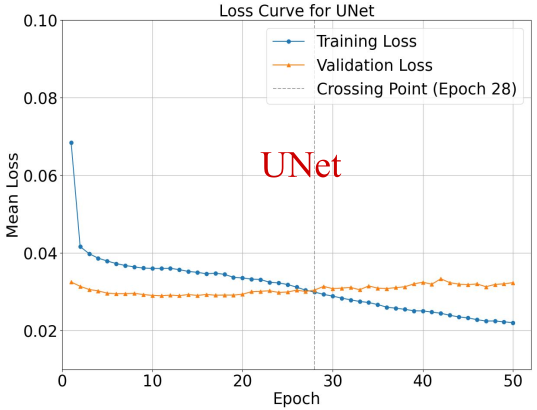
Training was carried out on the WC Cluster using an NVIDIA GeForce RTX 4090 GPU (24 GB)

Model Comparison - Train vs Val loss









- Rebin factor was set to 10 during training
- Training losses at the selected checkpoints:
  - UNet: 0.029 (Epoch 28)
  - MobileNetV2: 0.072 (Epoch 50)
  - MobileNetV3: 0.044 (Epoch 50)
  - Transformer: 0.029 (Epoch 32)
- Among the models, UNet and Transformer reached the most stable convergence with the lowest final losses

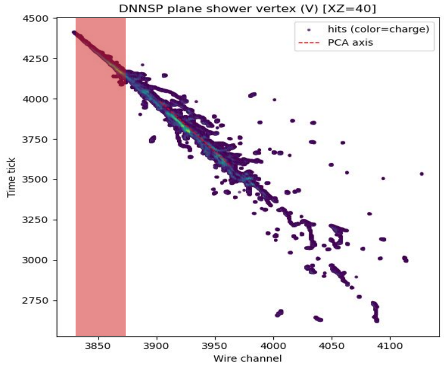
DNN-ROI Performance Evaluation

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

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

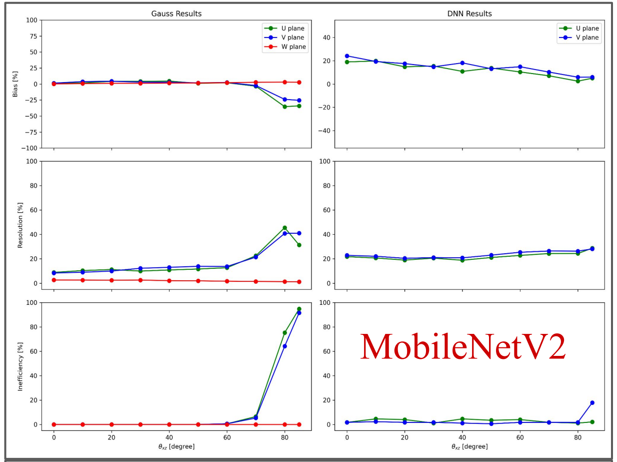
- For shower events, a charge profile based on vertex information was added as the fourth metric:
  - Sum the charge along the shower direction up to 42 wire channels (≈ 1–2 radiation lengths)
  - Compare the reconstructed-to-truth ratio charge ratio

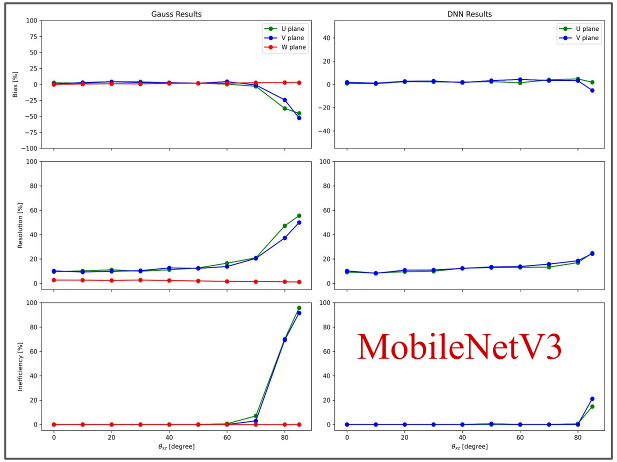
$$Q_{method} = \sum_{w \in W} Q_{method}(w) \qquad R_{cprofile} = \frac{Q_{reco}}{Q_{truth}}$$

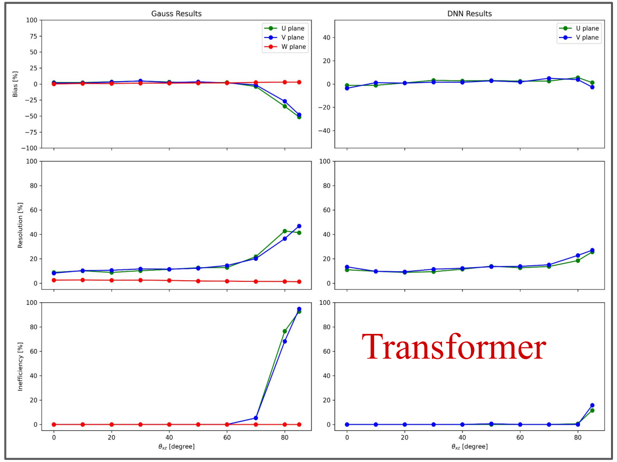


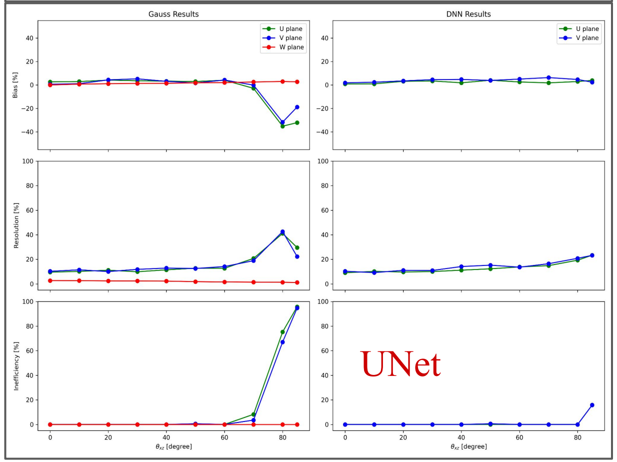
- Samples were generated with
  - Detector configuration: ProtoDUNE - Horizontal Drift (PD-HD)
  - XZ angle: 0°, 10°, 20°, 30°, 40°, 50°, 60°, 70°, 80°, 85°
  - Shower energies: 100 MeV, 500 MeV, 1 GeV, 2 GeV, 3 GeV, 5 GeV
  - Software: WCT standalone (Tracks), LAr-WCT (Showers)

Single Track Evaluation



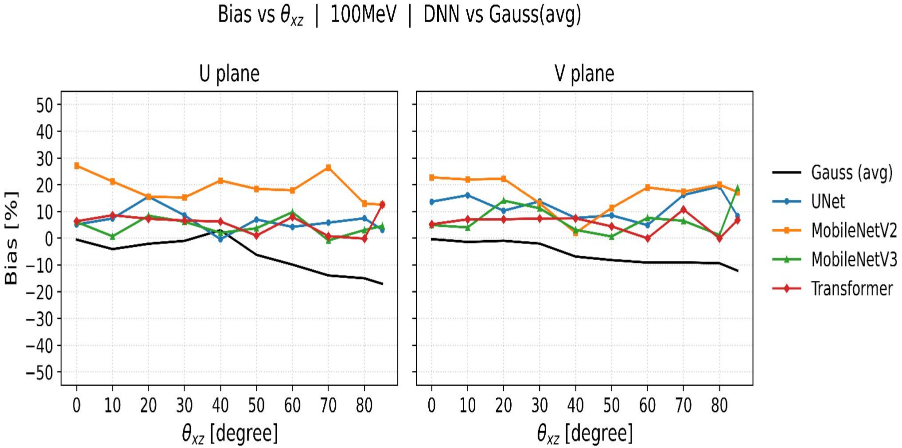


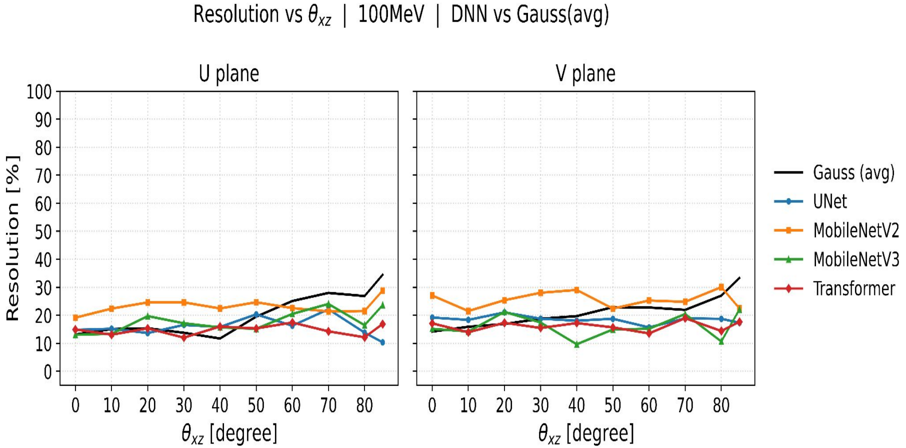


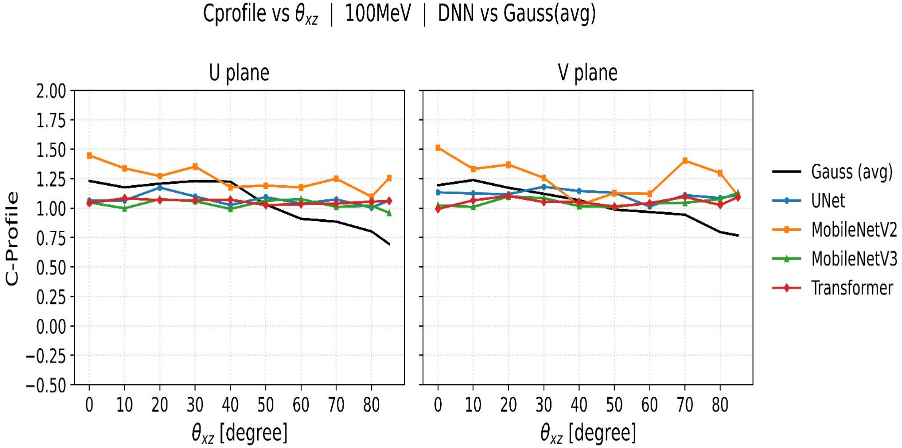


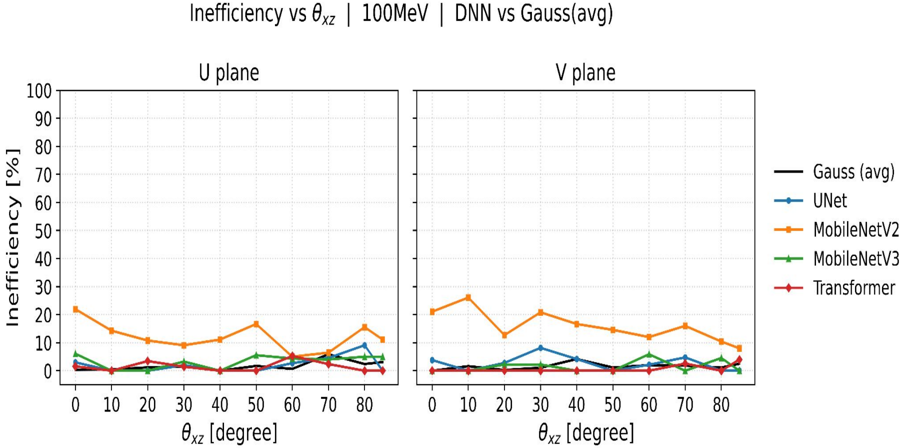
- All simulations were performed using the normal APA (2nd)
- MobileNetV2 exhibits ~20% bias at low angles
- The lower performance of MobileNetV2 is likely due to:
  - Fewer convolution blocks in the decoder
  - Absence of skip connections
- Other models demonstrate comparable performance

Single Shower Evaluation (100 MeV)

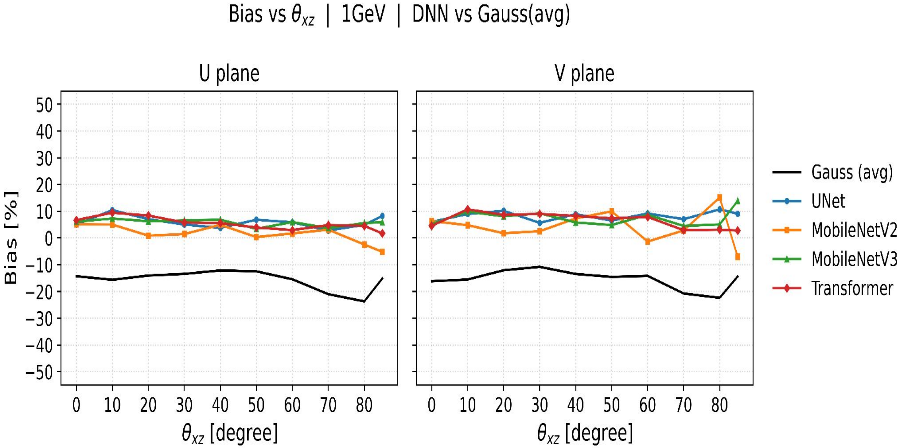


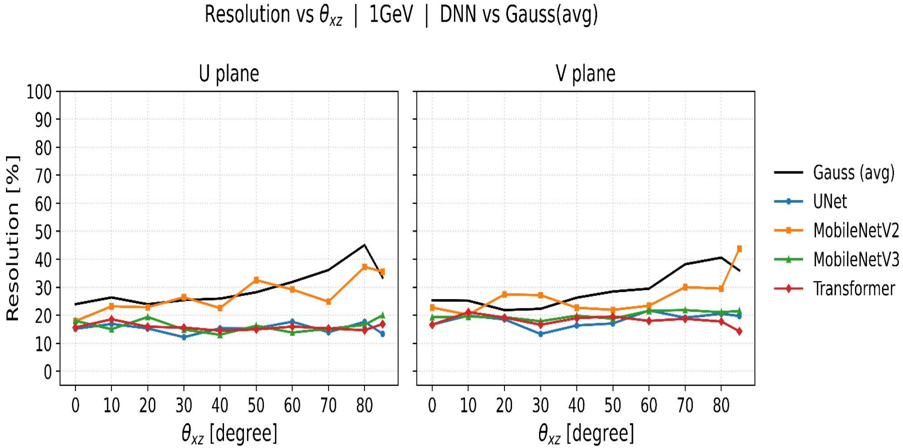


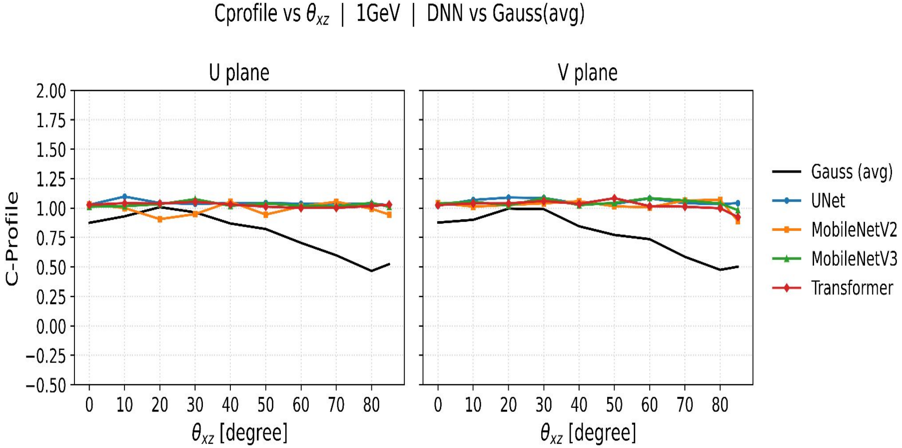


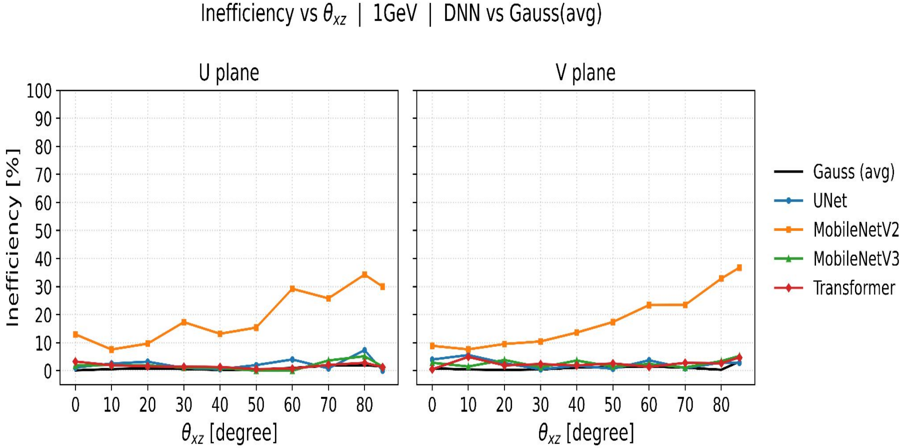


Single Shower Evaluation (1 GeV)

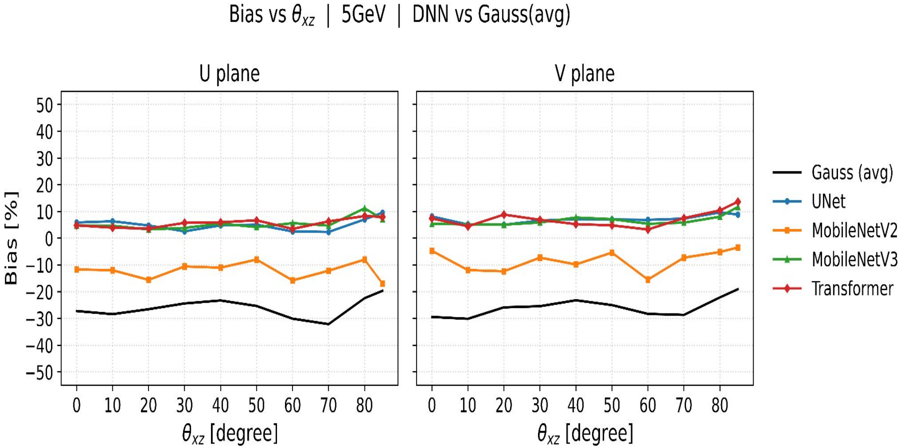


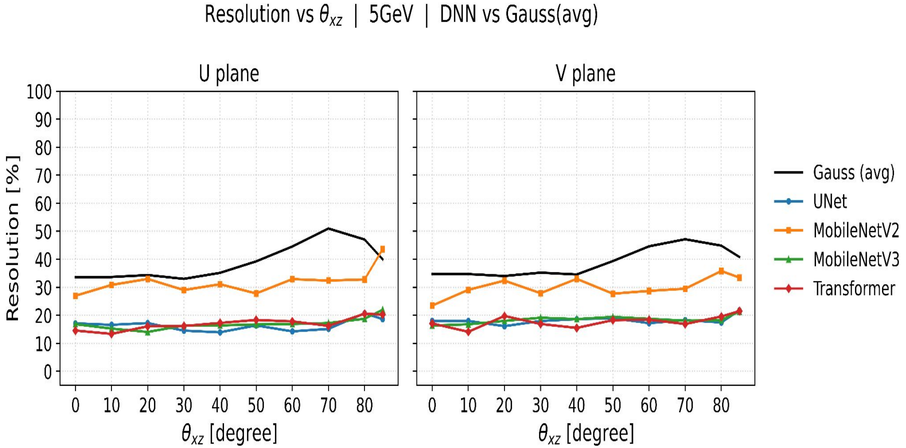


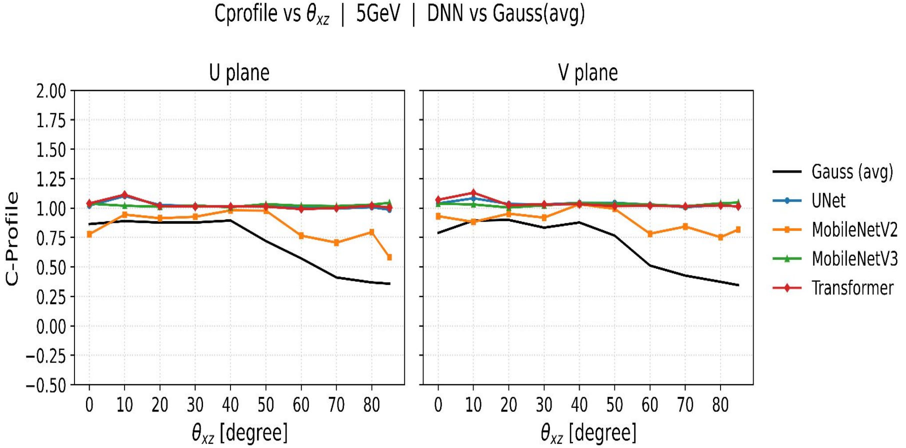


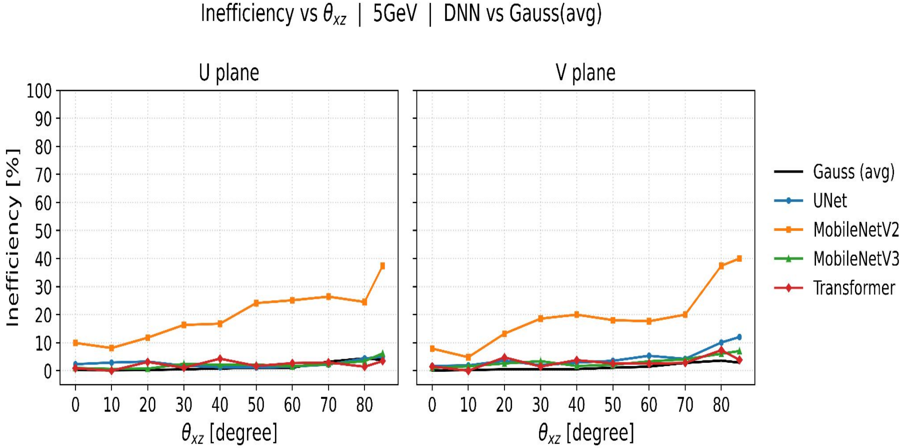


Single Shower Evaluation (5 GeV)









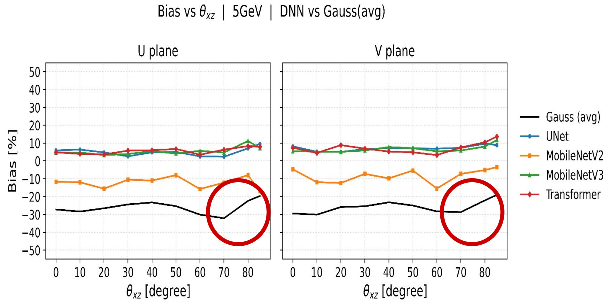
Single Shower Evaluation

Evaluation plots for 500 MeV, 2 GeV and 3 GeV can be found on slides 19-21

As the electron energy increases:

- Traditional ROI
  - Bias ↓, Charge Profile ↓, Resolution ↓, Inefficiency neary constant

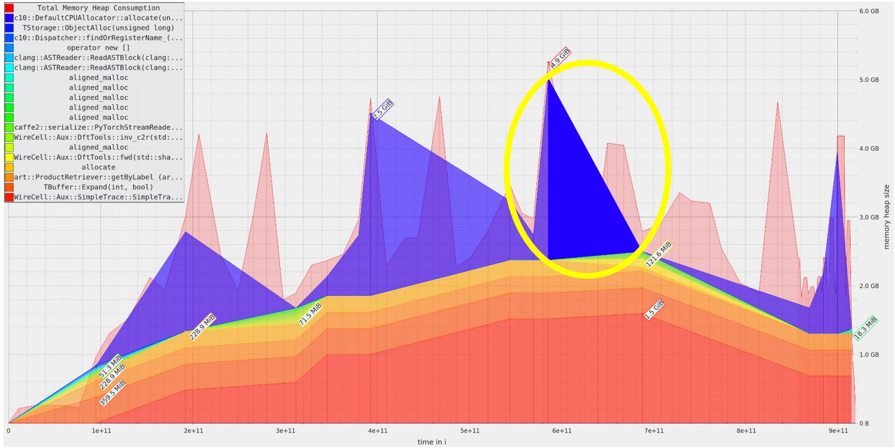
- DNN-ROI (except MobileNetV2)
  - Bias ↑, Charge Profile ↑
  - Resolution ↑ (≤ 1 GeV), then flattens
  - Inefficiency ↑ (≤ 500 MeV), then flattens

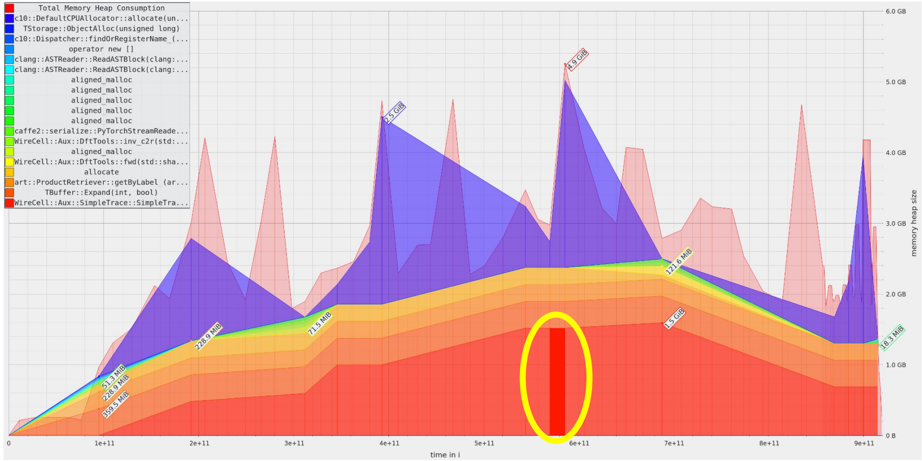


Remaining Questions

- Why does the Traditional ROI appear to improve at high angles as the energy increases?
- Why does the inefficiency of the Traditional ROI remain close to zero across all energies and angles?

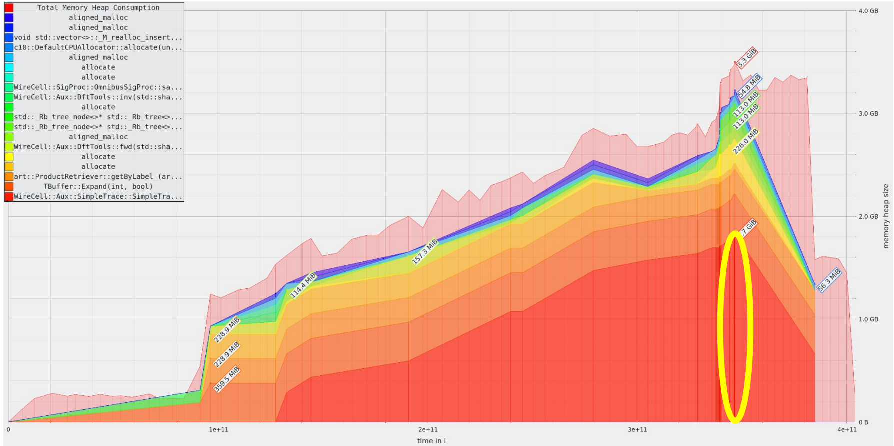
Memory/Time Consumption Measurement

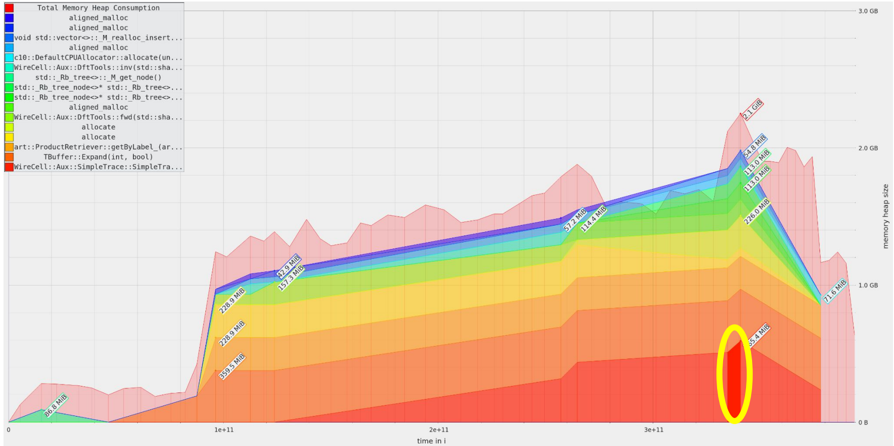




- Valgrind profiling was performed on the WC Cluster
  (using an AMD Ryzen Threadripper 7970X, 32 cores, 64 threads)
- DNN SigProc with **UNet** shows a peak memory usage of 4.9 GiB (≈ 5.26 GB)
- Two main contributors to memory consumption (highlighted in the yellow circle):
  - Blue region (2.5 GiB): *libtorch_cpu.so* (2.3GiB)
  - Red region (1.4 GiB): *save_data* (858 MiB) and *save_mproi* (343 MiB) in *OmnibusSigProc*

Memory/Time Consumption Measurement

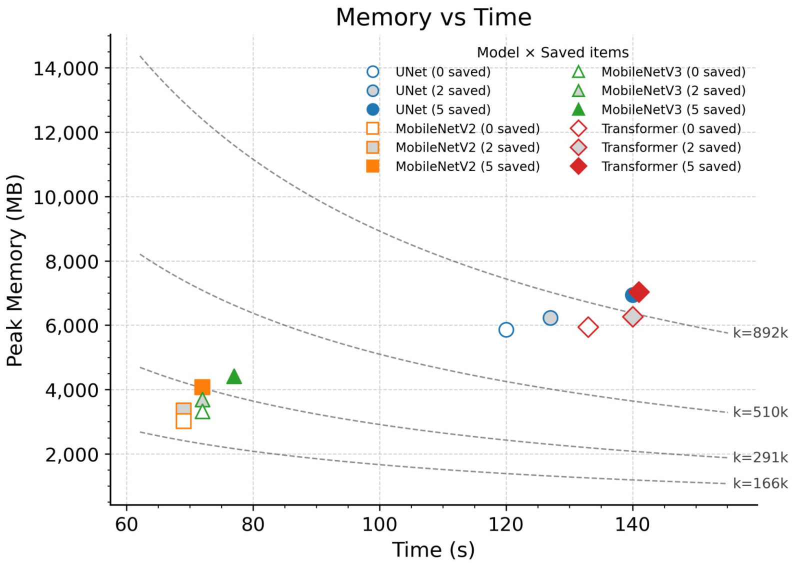




- Valgrind profiling was performed on the WC Cluster
  (using an AMD Ryzen Threadripper 7970X, 32 cores, 64 threads)
- With MobileNet, *libtorch_cpu.so* consumes **less than 50 MiB**
- The main remaining factor of memory consumption (highlighted in the yellow circle):
  - Red region (1.7 GiB): *save_data* (972 MiB) and *save_mproi* (440 MiB) in *OmnibusSigProc*
- When *save_data* is deactivated for unnecessary operations (right plot):
  - Red region (565 MiB): *save_mproi* (440 MiB) in *OmnibusSigProc*

Memory/Time Consumption Measurement

save_data	Model	Peak Memory (MB)	Time (s)
tight_lf, loose_lf, cleanup_roi, wiener, gauss	UNet	6942	140
	Transformer	7032	141
	MobileNetV2	4071	72
	MobileNetV3	4410	77
wiener, gauss	UNet	6225	127
	Transformer	6267	140
	MobileNetV2	***3343***	69
	MobileNetV3	***3687***	72
none	UNet	5863	120
	Transformer	5945	133
	MobileNetV2	***3012***	69
	MobileNetV3	***3324***	72



- Benchmark was performed on dunegpvm03 (AMD EPYC Processor, 4 vCPUs)
- The *lar* process utilized only a single vCPU
- DNN-ROI with MobileNetV3 consumes 1.5 GB more memory than Traditional ROI (see slide 24)

Data: np04hd_raw_run027673_0000_dataflow0_datawriter_0_20240704T050545.hdf5

Summary

DNN-ROI Performance Evaluation

- UNet, MobileNetV3, and Transformer exhibited similar performance, all outperforming the Traditional ROI
- The charge profile indicated that the Traditional ROI degrades as the angle increases
- The underperformance of MobileNetV2 is due to weaker design of its decoder architectures

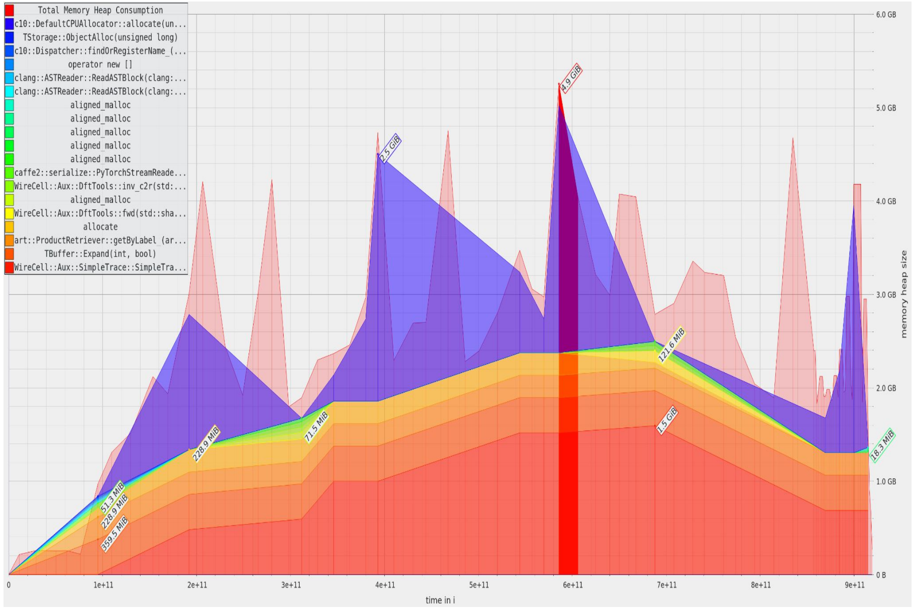
Time/Memory Consumption

- Deactivating *save_data* function can reduce memory usage by approximately 1 GB
- The UNet encoder architecture is the primary contributor (~3 GB) to memory usage

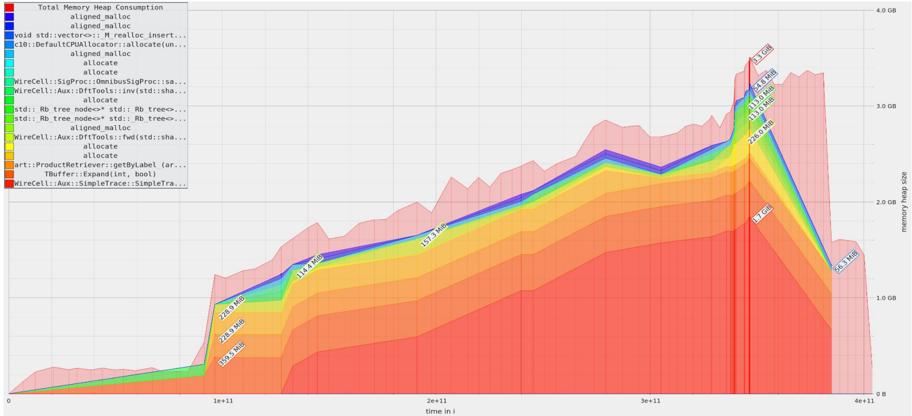
Considering both performance and computational constraints, **MobileNetV3** provided **the best overall balance**

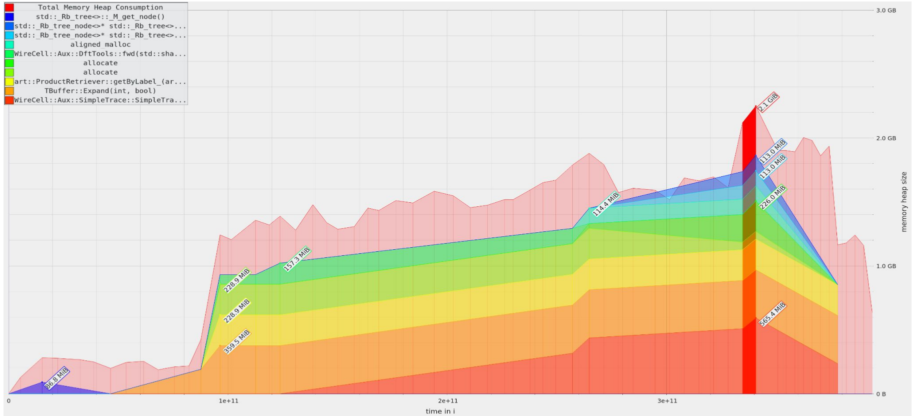
Back Up

Valgrind Profiling - UNet Peak Memory



Valgrind Profiling - MobileNetV2 Peak Memory



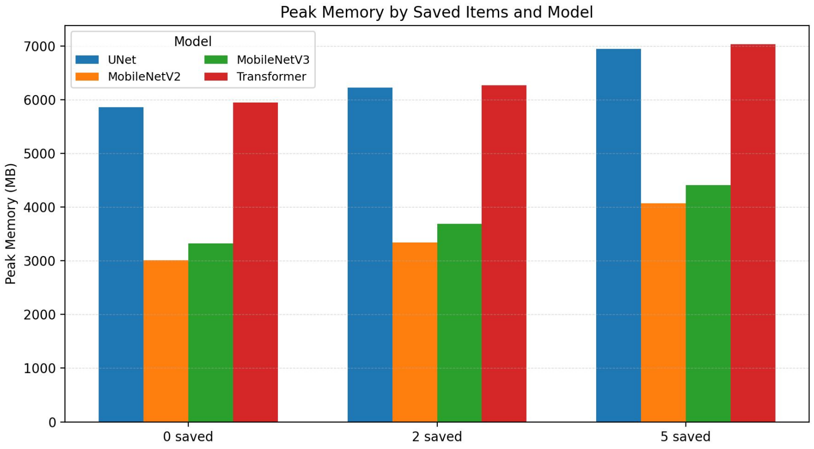


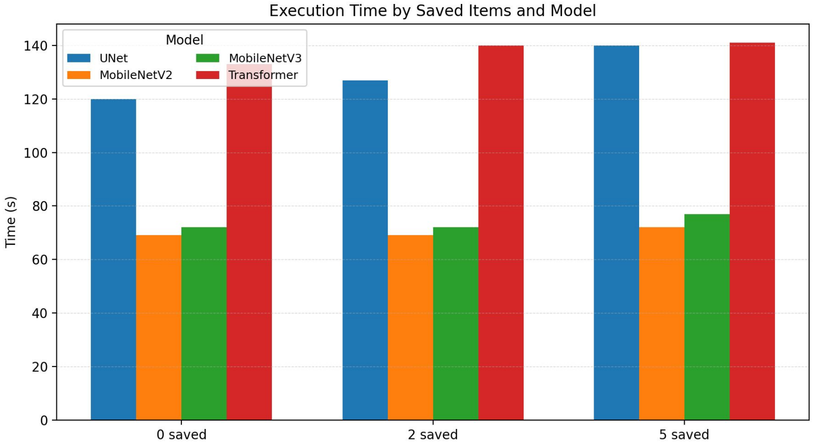
- 3.3 GiB: Snapshot #78 (peak)
  - 1.7 GiB: allocate (new_allocator.h:137)
    - 1.7 GiB: allocate (alloc_traits.h:464)
      - 1.7 GiB: _M_allocate (stl_vector.h:378)
        - 1.7 GiB: _M_create_storage (stl_vector.h:395)
          - 1.7 GiB: _Vector_base (stl_vector.h:332)
            - 1.7 GiB: vector (stl_vector.h:598)
              - 1.7 GiB: WireCell::Aux::SimpleTrace::SimpleTrace(int, int, std::vector<> const&) (SimpleTrace.cxx:8)
                - 972.7 MiB: WireCell::SigProc::OmnibusSigProc::save_data(std::vector<>&, std::vector<>&, int, std::vector<> const&, std::vector<>&, std::__cxx11::basic_string<> const&, bool) (OmnibusSigProc.cxx:512)
                - 440.7 MiB: WireCell::SigProc::OmnibusSigProc::save_mproi(std::vector<>&, std::vector<>&, int, std::multimap<> const&) (OmnibusSigProc.cxx:763)
                - 228.9 MiB: WireCell::SigProc::OmnibusSigProc::save_roi(std::vector<>&, std::vector<>&, int, std::vector<>&) (OmnibusSigProc.cxx:612)
                - 109.9 MiB: _Construct<> (stl_construct.h:119)

- 2.1 GiB: Snapshot #48 (peak)
  - 565.4 MiB: allocate (new_allocator.h:137)
    - 565.4 MiB: allocate (alloc_traits.h:464)
      - 565.4 MiB: _M_allocate (stl_vector.h:378)
        - 565.4 MiB: _M_create_storage (stl_vector.h:395)
          - 565.4 MiB: _Vector_base (stl_vector.h:332)
            - 565.4 MiB: vector (stl_vector.h:598)
              - 565.4 MiB: WireCell::Aux::SimpleTrace::SimpleTrace(int, int, std::vector<> const&) (SimpleTrace.cxx:8)
                - 455.5 MiB: WireCell::SigProc::OmnibusSigProc::save_mproi(std::vector<>&, std::vector<>&, int, std::multimap<> const&) (OmnibusSigProc.cxx:763)
                - 109.9 MiB: _Construct<> (stl_construct.h:119)

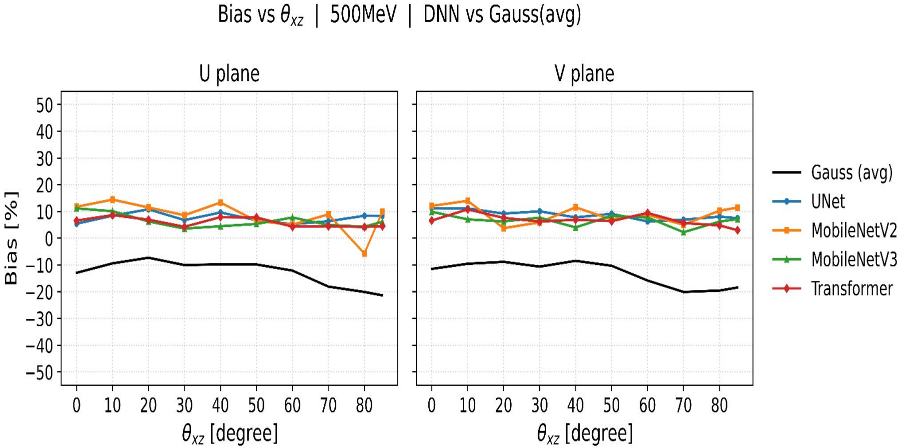
Memory/Time Consumption Measurement

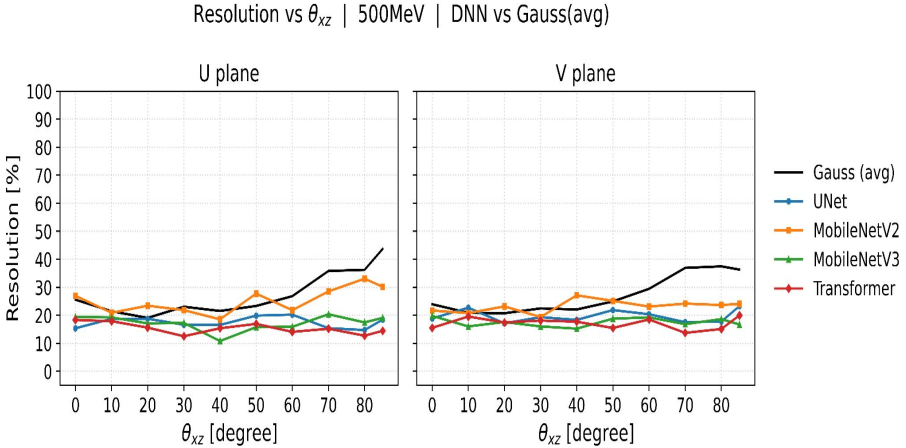
save_data	Model	Peak Memory (MB)	Time (s)
tight_lf, loose_lf, cleanup_roi, wiener, gauss	UNet	6942	140
	Transformer	7032	141
	MobileNetV2	4071	72
	MobileNetV3	4410	77
wiener, gauss	UNet	6225	127
	Transformer	6267	140
	MobileNetV2	***3343***	69
	MobileNetV3	***3687***	72
none	UNet	5863	120
	Transformer	5945	133
	MobileNetV2	***3012***	69
	MobileNetV3	***3324***	72

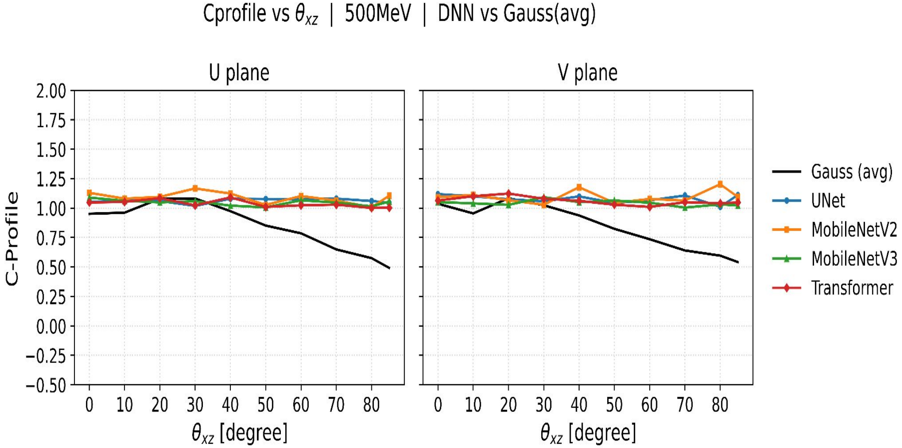


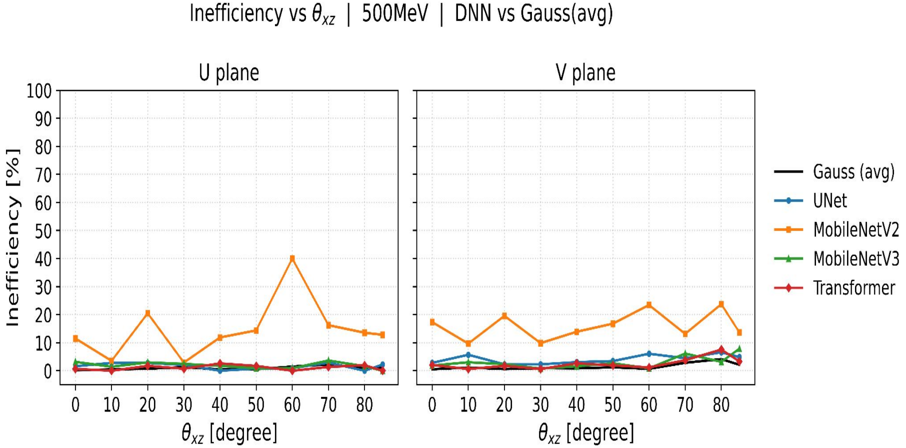


Single Shower Evaluation (500 MeV)

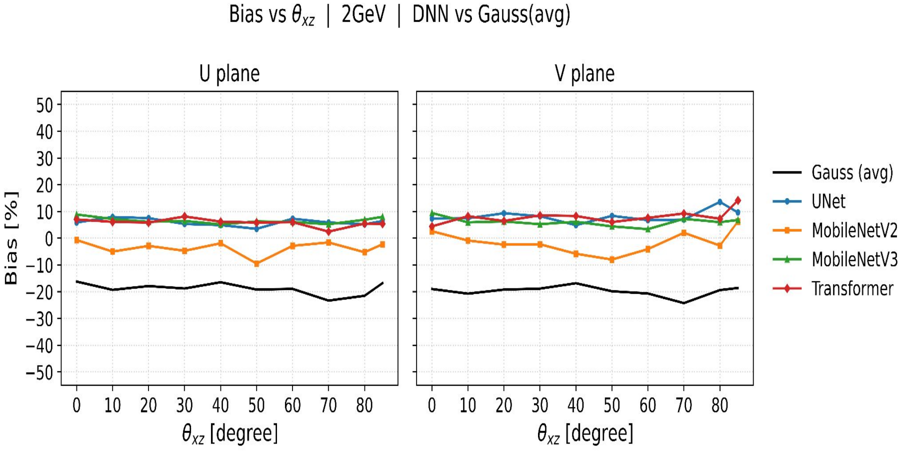


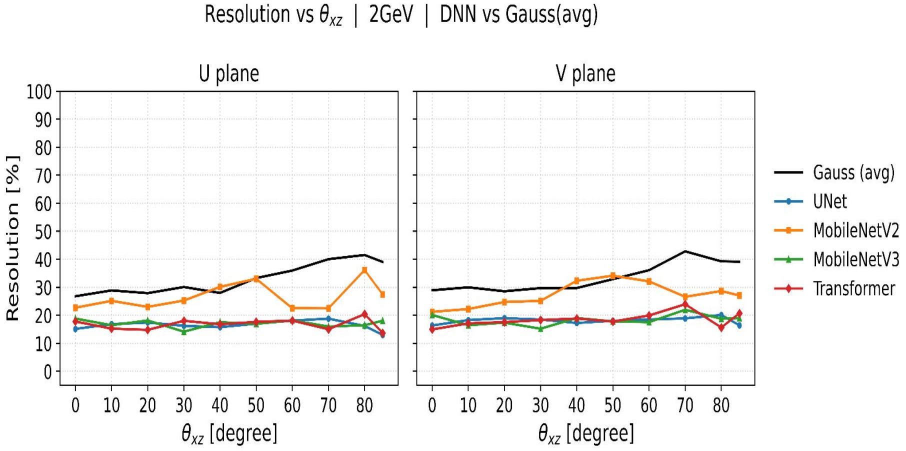


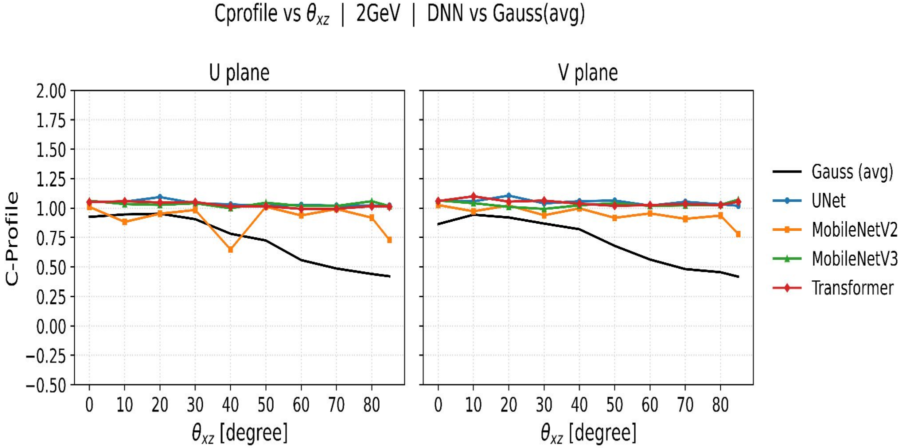


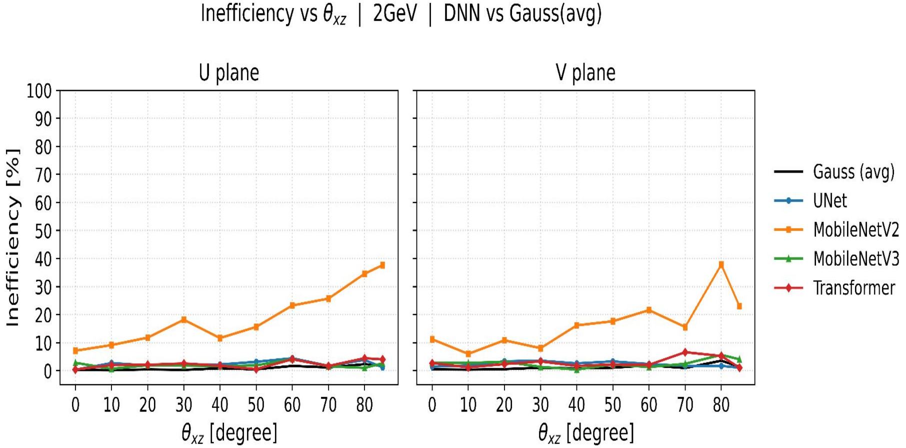


Single Shower Evaluation (2 GeV)

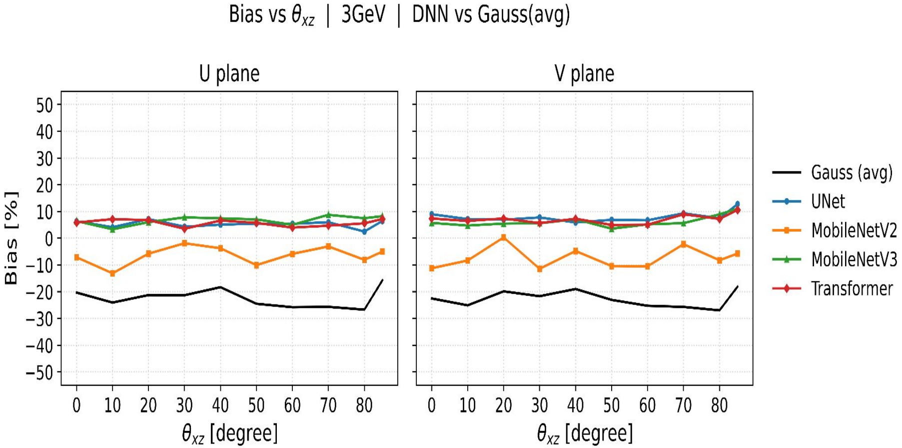


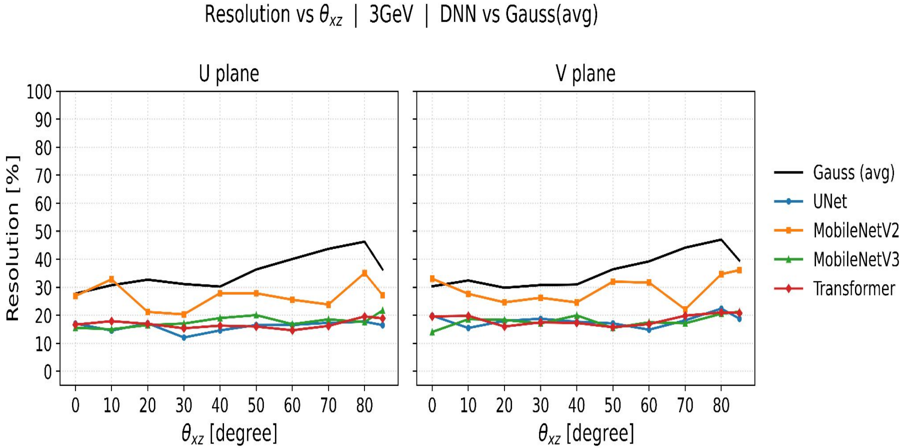


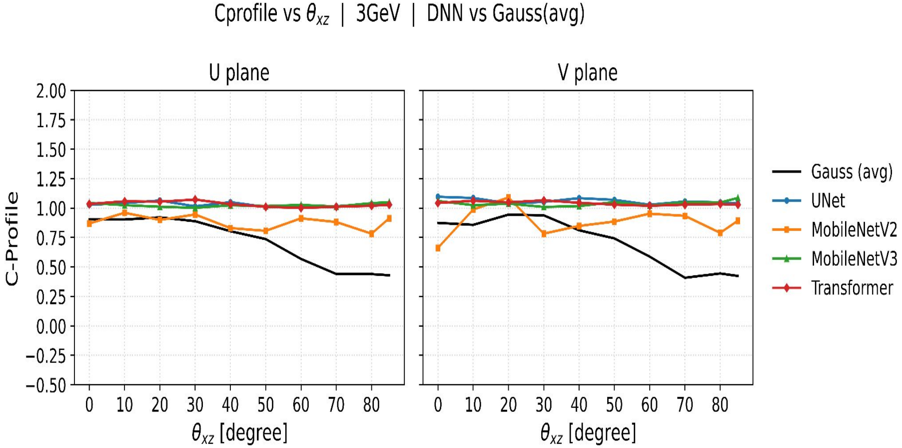


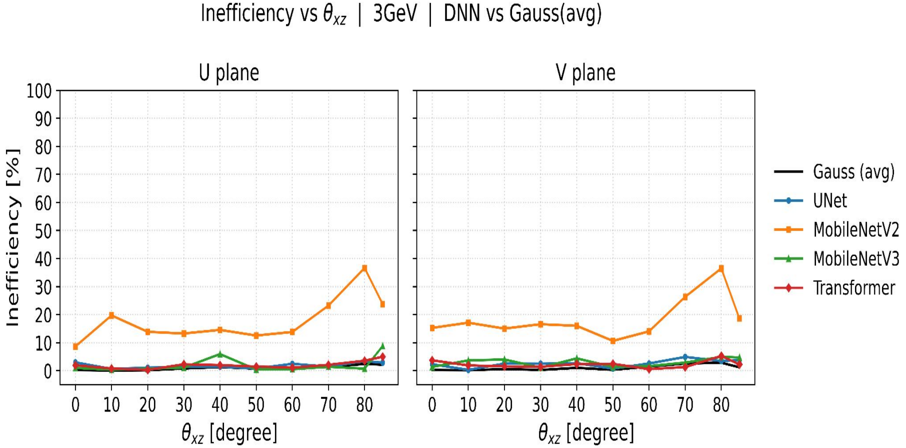


Single Shower Evaluation (3 GeV)

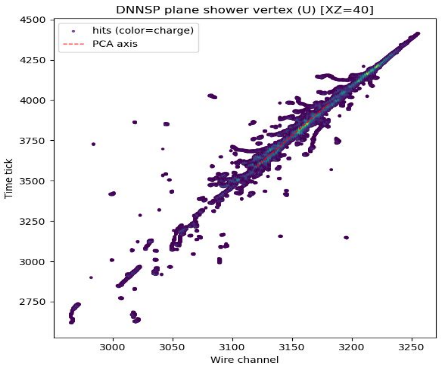


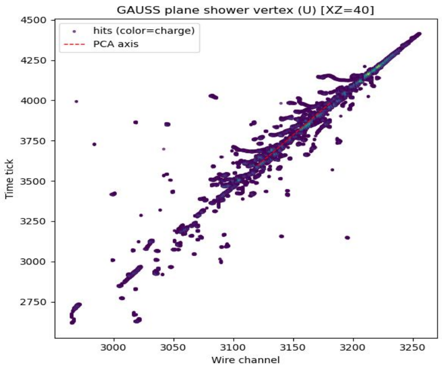


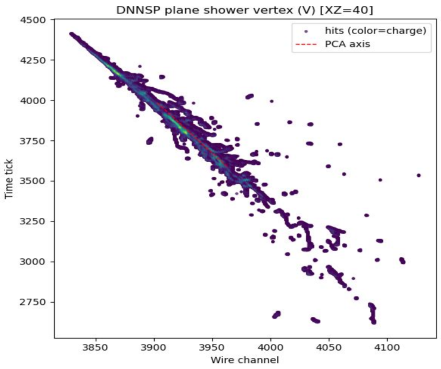


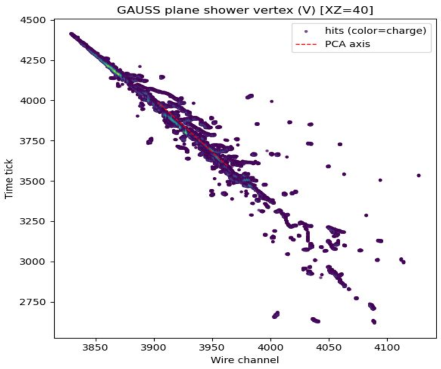


Angle between reco and true direction (E = 5 GeV, XZ = 40)



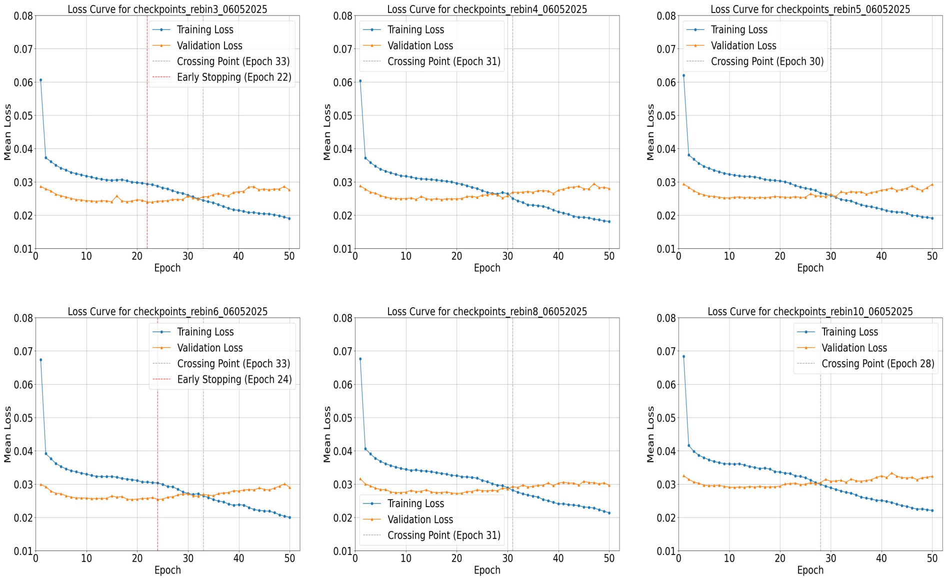






	Angle diff	
	Gauss (deg.)	DNN (deg.)
U plane	0.74	0.82
V plane	0.52	0.51
W plane	1.22	

DNN Training with rebining - UNet



- Re-trained the UNet with rebinning in time tick
- High resolution in time (y-axis) contributes better training loss: 0.0299 → 0.0245
- Loss: Binary Cross-Entropy, Epoch = 50, Learning Rate = 0.1, Momentum = 0.9, train vs val split = 90:10

Memory and Time consumption on the dunegpvm

Server	WCT	Resource	DNN ROI	Mem (MB)	Time (s)	Mem Ratio	Time Ratio
dunegpvm	dunesw	None	None	1841.24	59.67	1.00	1.00
dunegpvm	dunesw	CPU	UNet	7399.78	155.31	4.02	2.60
dunegpvm	dunesw	CPU	MobileNetV2	4575.95	97.3	2.49	1.63

- PyTorch Model file
  - ❏ PD-HD data: run026763_0008, run027673_0000, run027673_0001, run027673_0002, run028588_0019
  - UNet: unet-cosmic390-newwc-depofluxsplat-pdhd.ts (Jay)
  - MobileNetV2: CP49.ts (from my training, rebin: 10)

- MobileNetV2 showed a more efficient balance between memory and time compared to UNet