Denoising Convolutional Networks to Accelerate Detector Simulation

S. Banerjee (FNAL), B. Cruz Rodriguez (UPRM), L. Franklin (UMD), H. Guerrero De La Cruz (UPRM), T. Leininger (Lafayette), S. Norberg (UPRM), **K. Pedro (FNAL)**, A. Rosado Trinidad (UPRM), Y. Ye (Chicago) on behalf of the CMS Collaboration

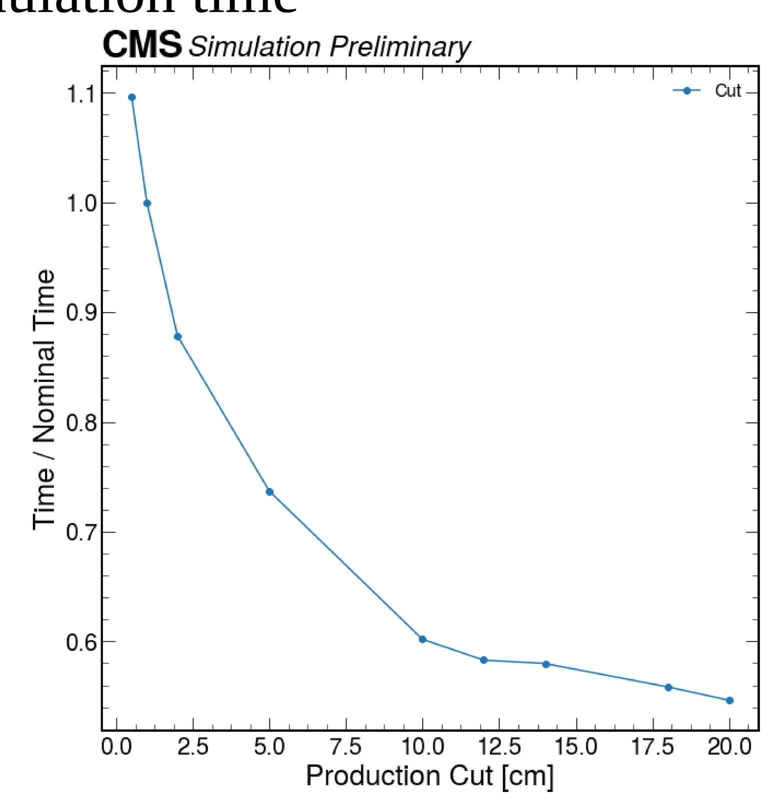
Introduction

- Detector simulation consumed 40% of grid CPU at start of LHC Run 2 [1]
- High Luminosity LHC will demand more events w/ more complex geometry & more precise physics [2]
- Use artificial intelligence (AI) to "denoise" reduced-precision output from faster simulation

➢ Get high-quality final result in less time
  - Inspired by machine learning (ML) for Monte Carlo ray-tracing in computer animation [3]

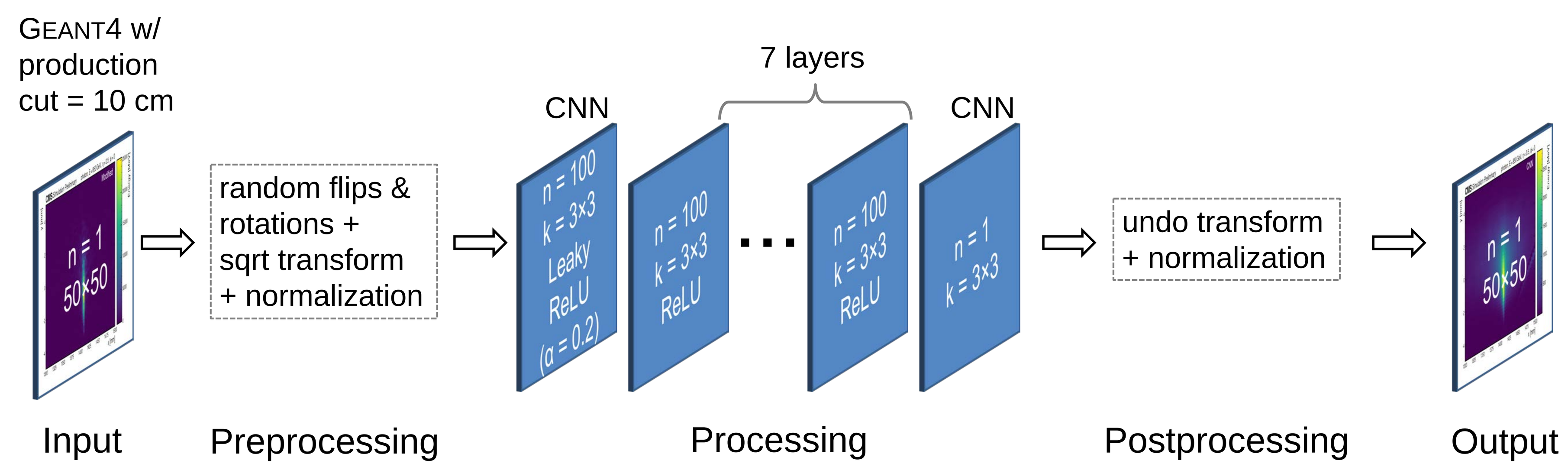
Modified Simulation

- Change GEANT4 (full simulation) parameters to increase speed (tested in $t\bar{t}$ events)
- Increase production cut (range-out distance below which particles do not produce secondary particles): reduce simulation time by almost 2×
- Magnetic field, Russian Roulette (randomly discarding low-energy particles while increasing energy of kept particles) parameters: ~5% effects on simulation time



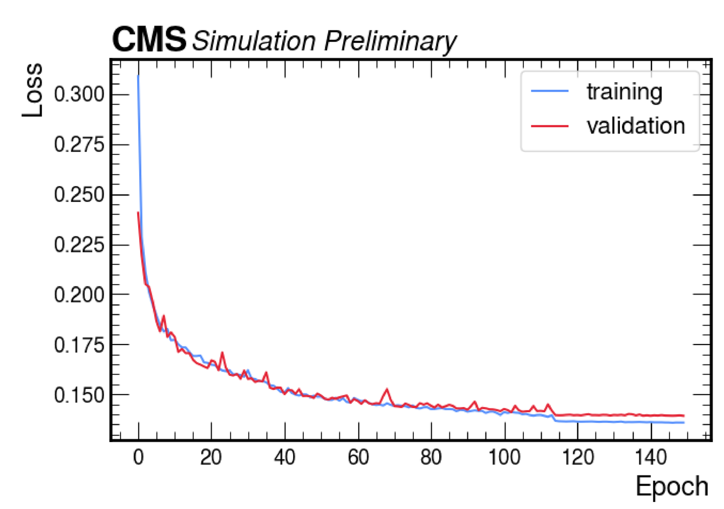
Samples

- Single photon showers, E = 850 GeV, η = 0.5, φ = 0
- Discretize GEANT4 energy deposits in ECAL barrel into 50×50 pixel images in x, y (sum over z)
- Process same generated events w/ original (high-quality) and modified (production cut = 10 cm) simulations [4]
  - Reset random seed per event (consistency)
- 5K events each for training, validation, testing

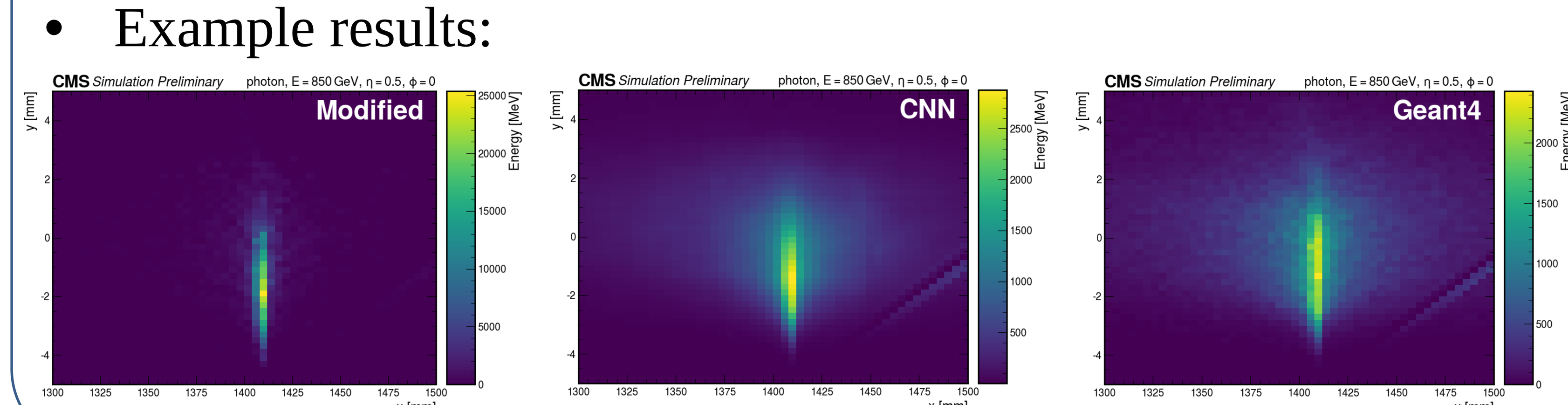


Training

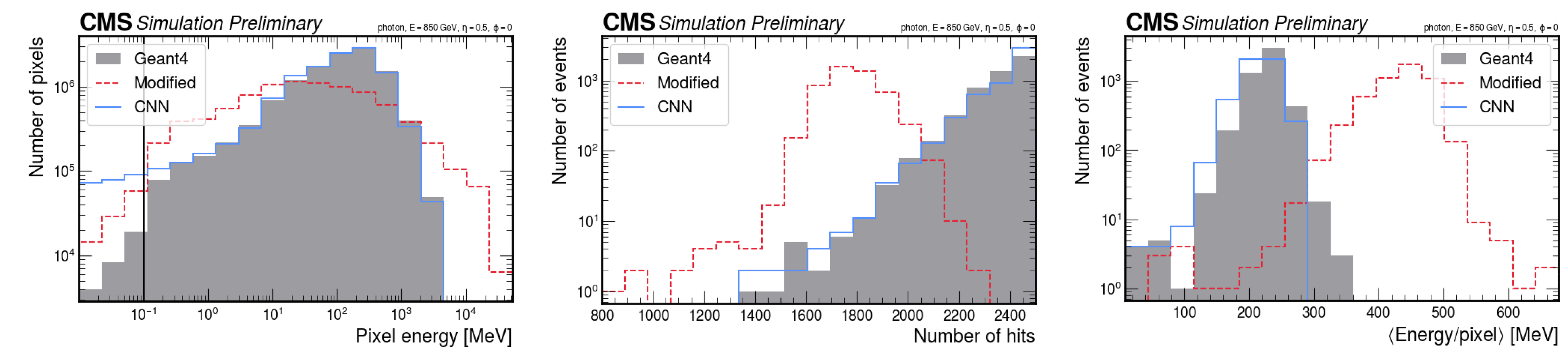
- Architecture: convolutional neural network (CNN), 9 layers, 100 features, 3×3 kernels, ReLU activation (PyTorch) [5]
  - Regression: predict energy value for each pixel
- Improve low-energy fidelity: use $\sqrt{E}$ as input, LeakyReLU activation in first CNN layer
- Normalize input as $(\sqrt{E} - \mu)/\sigma$, w/ mean and standard deviation computed per-event from modified simulation



- Hyperparameters: batch size 50, learning rate 0.001 (schedule: reduce on plateau), 150 epochs
- Loss function: L1 (mean absolute error)
- Example results:

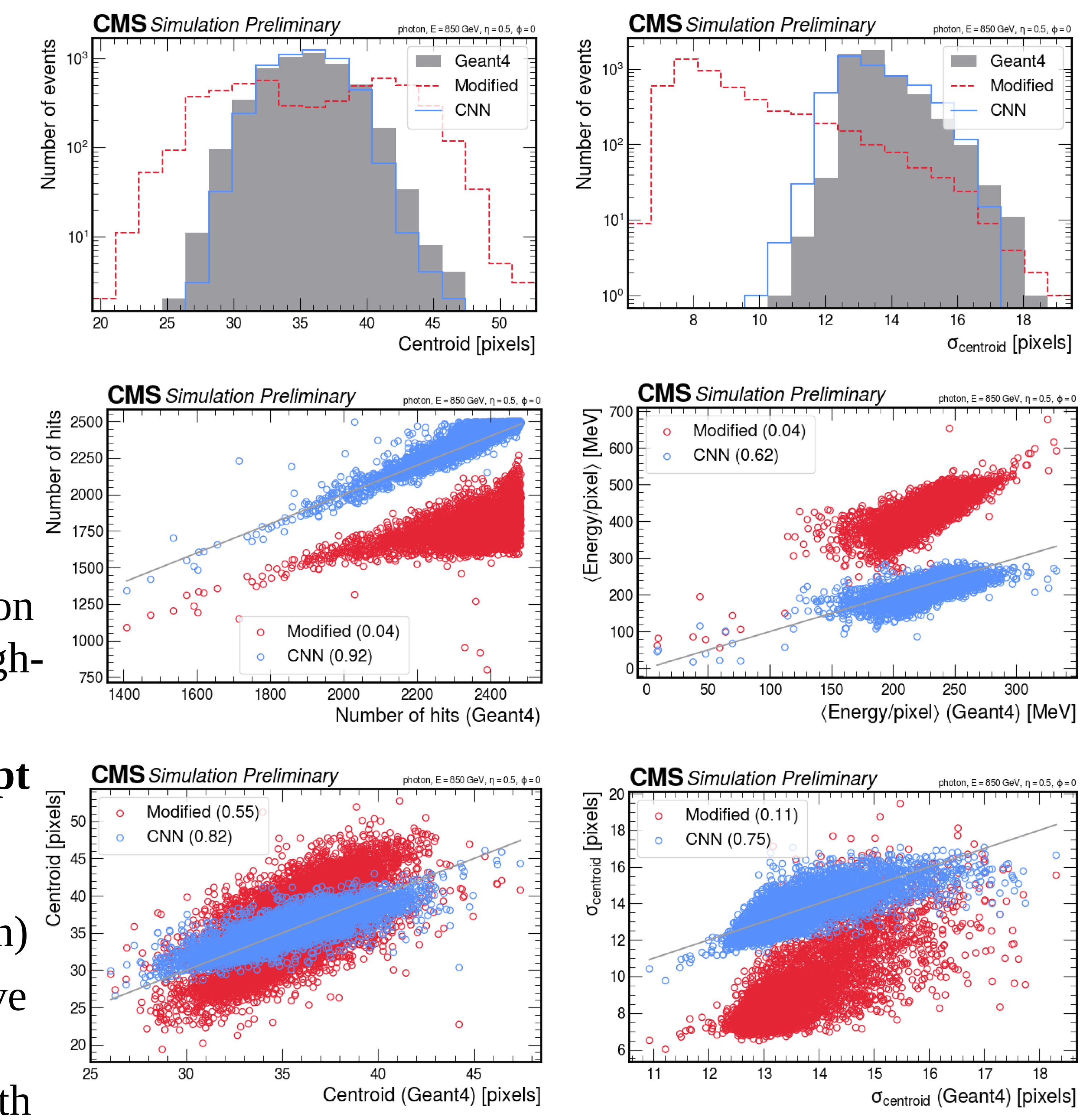


Results



- Good pixel energy agreement down to low values
- Number of hits, ‹energy/pixel› shown after threshold: pixel energy > 0.1 MeV
- Good agreement in shower centroid and width (2nd moment)
- Per-event comparisons: high concordance correlation between CNN output & high-quality simulation

➢ **Successful proof of concept** for AI denoising approach (regression-based ML, built on classical simulation)
  - Qualitatively competitive results vs. other AI approaches (e.g. [6]) with much simpler CNN architecture



Technical Performance

- Nvidia P100 GPU
- Training: 2GB memory, ~15 s/epoch
- Inference:
  - Batch 1: 800 evt/s
  - Batch 100: 8200 evt/s

Future Work

- Expand to more energy values, particle types, subdetectors, 3D images, etc.
- Extract additional input features from GEANT4
- Optimize NN architecture, loss function(s), hyperparameters
- Explore different simulation engines (e.g. CMS FastSim)
- Implement inference in CMS software

References

[1] arXiv:1803.04165
[2] Eur. Phys. J. Web Conf. 214 (2019) 02036
[3] ACM Trans. Graph 36 (2017) 97
[4] GitHub:cms-denoising/SimDenoising
[5] GitHub:cms-denoising/SimDenoising_training
[6] Comput. Softw. Big Science 5 (2021) 13