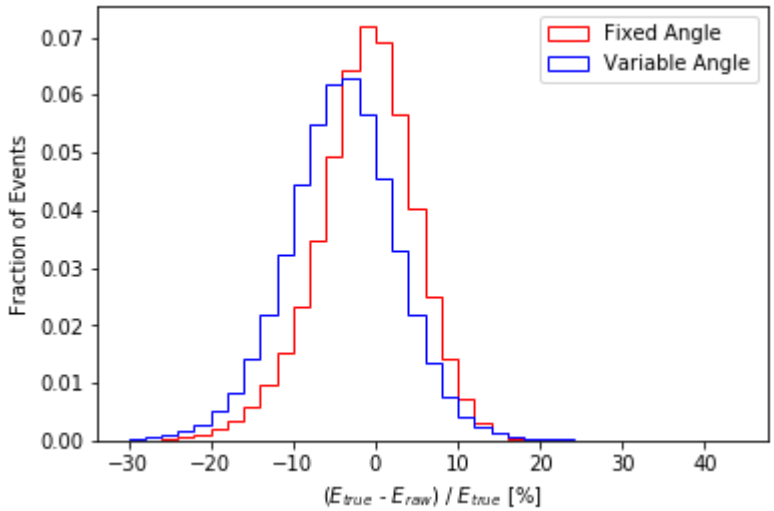
Regression: Variable Angle Electrons

Dominick Olivito (UCSD)

Overview

- Trying to understand why Variable Angle electron performance is worse than Fixed Angle

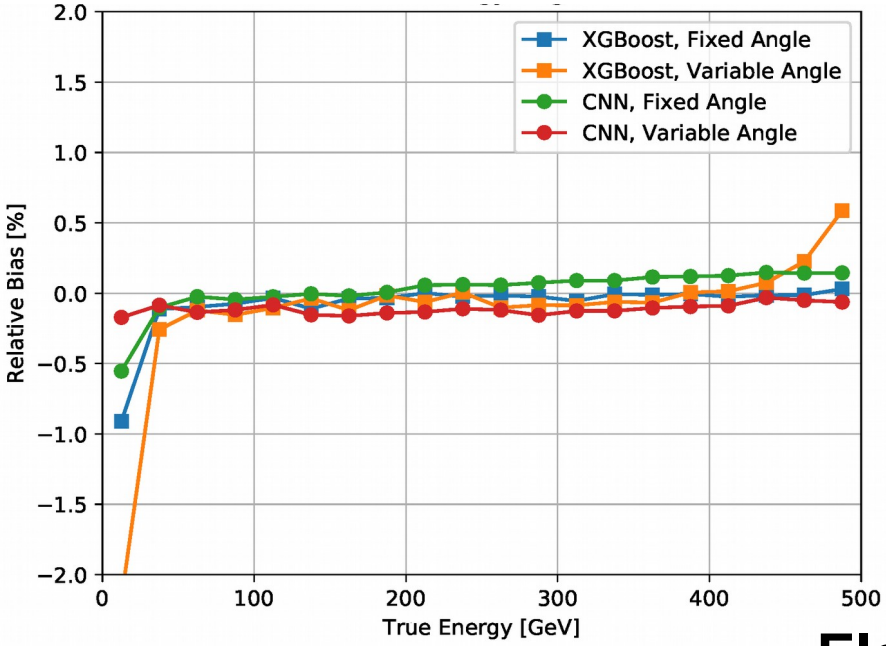
- See already that raw (ECAL+HCAL)/True response is different for variable angle vs fixed, so may expect differences at some level

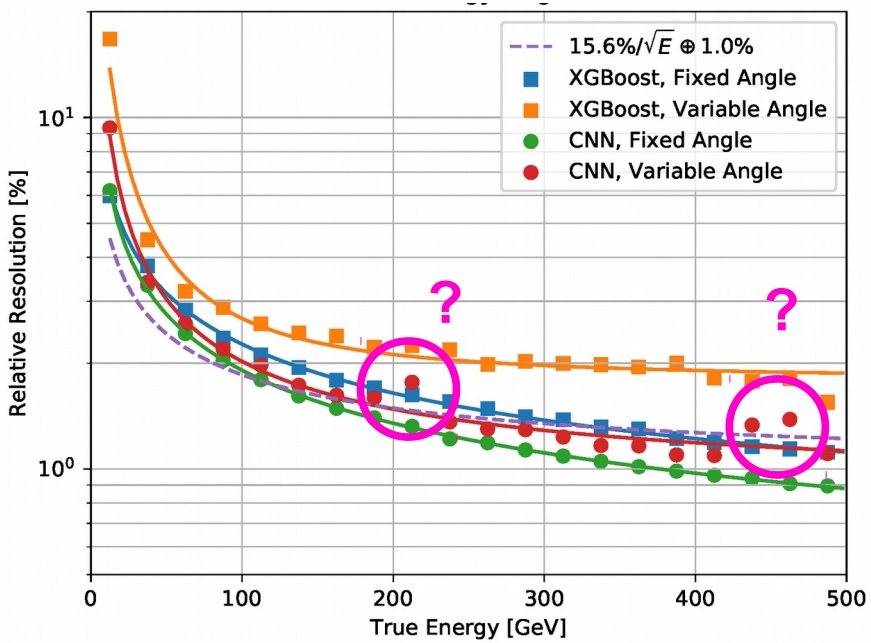


Electrons

Previous Results

- Substantial difference in performance between Fixed and Variable angle samples, especially with XGBoost
- Less difference with CNN, but suspicious outliers in resolution vs energy plot
  - Using RMS → sensitive to large outliers in (True-Pred)/True
  - Outliers come mainly from events with small fraction of true energy in window, or most of energy in HCAL(?)

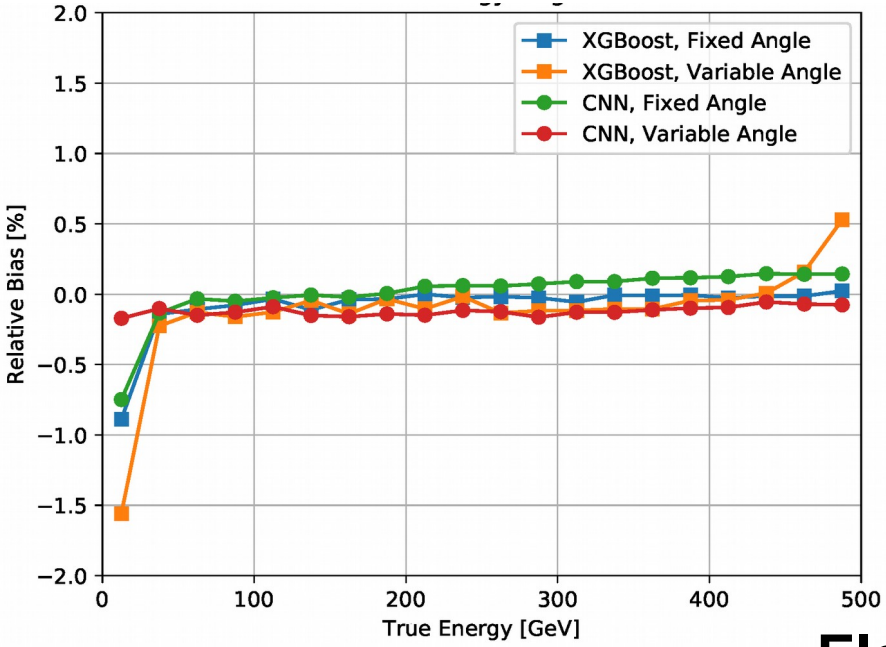


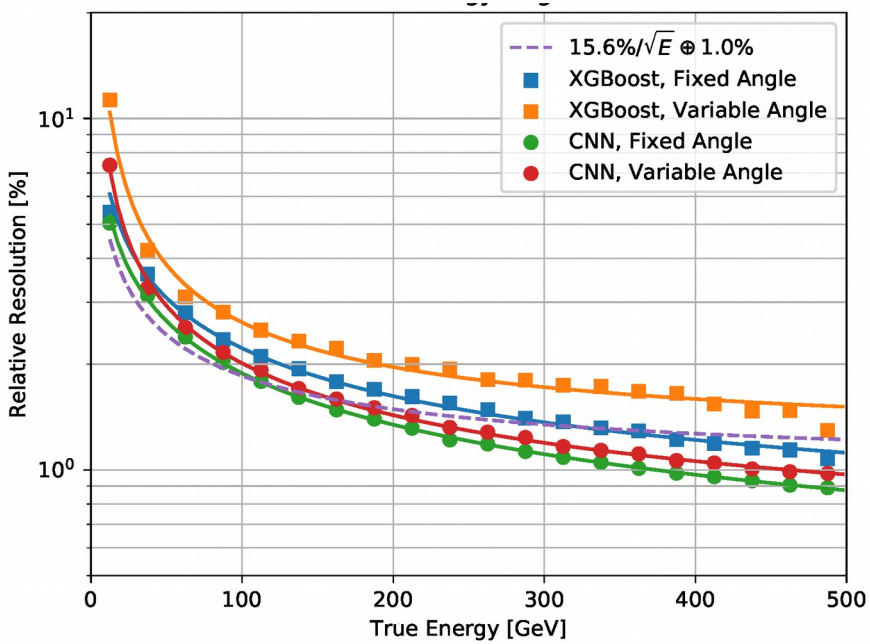


Electrons

After Outlier Removal

- In each bin, remove events that are more than 5 sigma (RMS) away from mean (True-Pred)/True
- Removes a handful of events / bin, < 0.1% overall
  - Could remove based on (ECAL+HCAL)/True, didn't have ready
  - Doesn't affect Fixed Angle sample
- Variable angle performance improves, much closer to fixed angle now → especially for CNN





Electrons

Summary / To Do

- Some of the bad Variable Angle Electron performance was coming from events with either:
  - Very small fraction of true energy reconstructed in window, or
  - Large fraction of reco energy in HCAL (instead of 0-5%)
- Doesn't really occur in Fixed Angle sample, not sure what the difference is

- To do:
  - Check resolution vs eta
  - Check other particles

Bonus Slides

Samples / Details

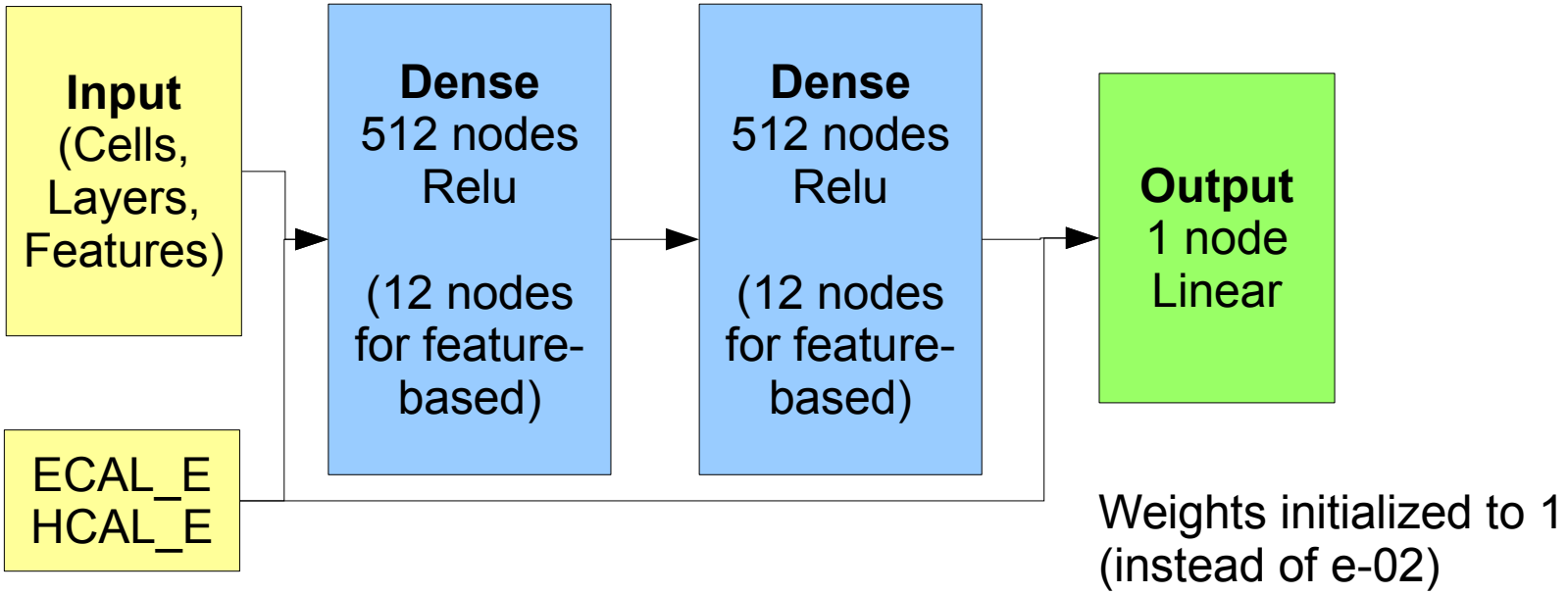
- Samples: new larger window samples, fixed angle, with features
  - On culture-plate at caltech:
    - /data/shared/LCDLargeWindow/fixedangle/*Escan/*.h5
    - /data/shared/LCDLargeWindow/varangle/*Escan/*.h5
  - Slimmed versions with only features (no images):
    - /data/shared/LCDLargeWindow/fixedangle/*Escan/merged_featuresonly/
    - /data/shared/LCDLargeWindow/varangle/*Escan/merged_featuresonly/
  - ~800k events, 70% train, 30% test

- Running XGBoost in python with:
  - maxdepth 3, up to 1000 rounds
  - Early stopping if test loss doesn't improve for 10 rounds

- Running DNNs / CNNs in pytorch, python3 using Triforce
  - Dropout 0.2
  - Adam, learning rate 0.001
  - L2 regularization 0.01 (“decayRate”)
  - Train for 5-10 epochs depending on window size

Skip Connections

- Basic idea: hardcode **Identity function** into network, to make other layers learn residual correction to identity
- Appropriate for our case: we know linear regression in ECAL_E, HCAL_E gets close to the right answer
- Performance is similar, training converges faster
  - For Feature Based NN, in 10-20 epochs instead of 40



NIPS CNN Architecture

- Modifications:
  - 51x51x25 ECAL input
  - 11x11x60 HCAL input
  - 3 conv filters for HCAL
  - Skip connections for ECAL_E, HCAL_E