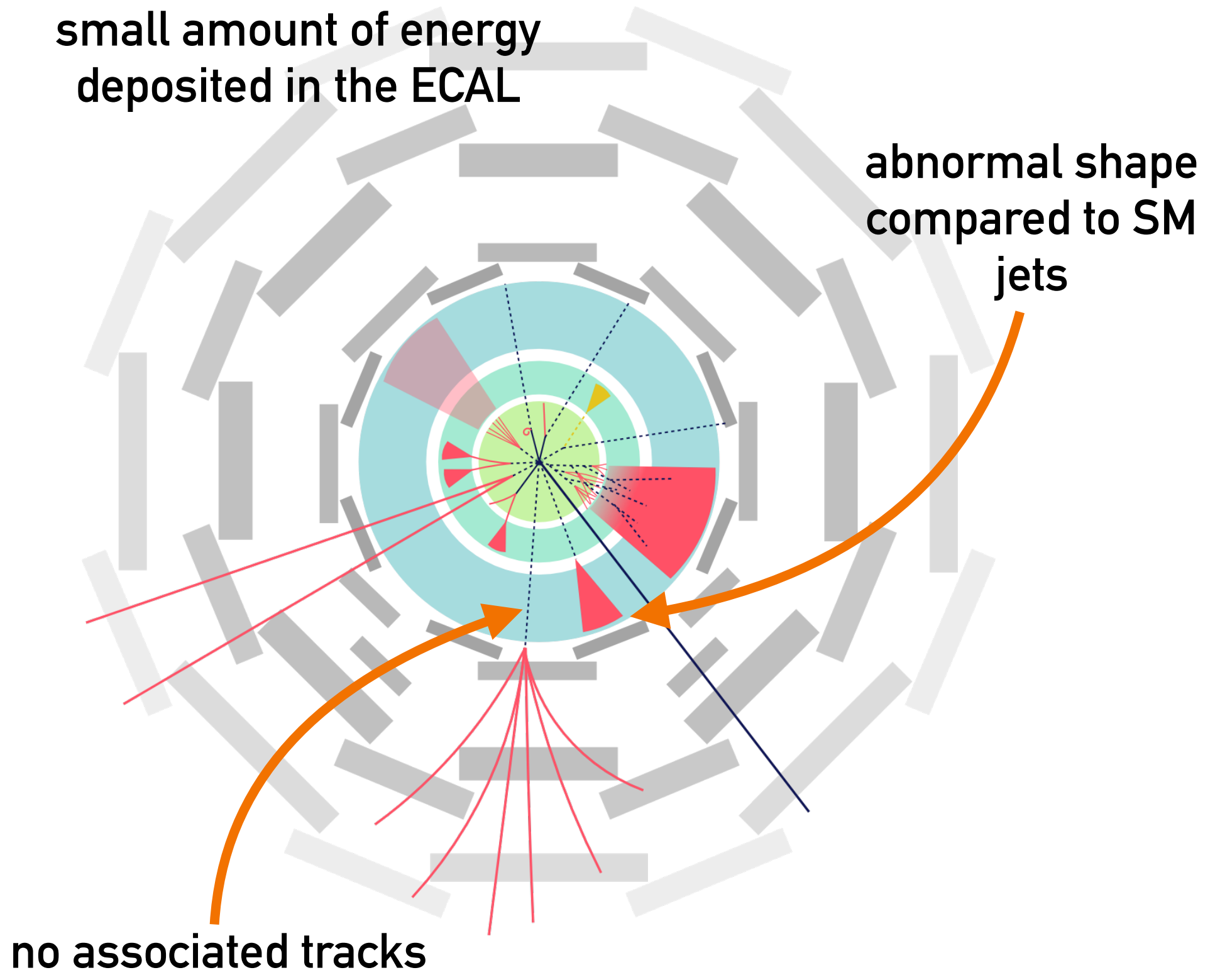
The Good, The Bad, and The Ugly: A tale of Physics, Software, and ML

Alex Golub, Gordon Watts

Motivation

- Long lived particles provide information about the hidden sector
- Hidden Sector (HS) - shares no quantum numbers with standard model
  - Can't detect until HS particle decays back into or mixes with standard model particle

Features of Long Lived Particles


small amount of energy deposited in the ECAL
abnormal shape compared to SM jets
no associated tracks

Main Backgrounds

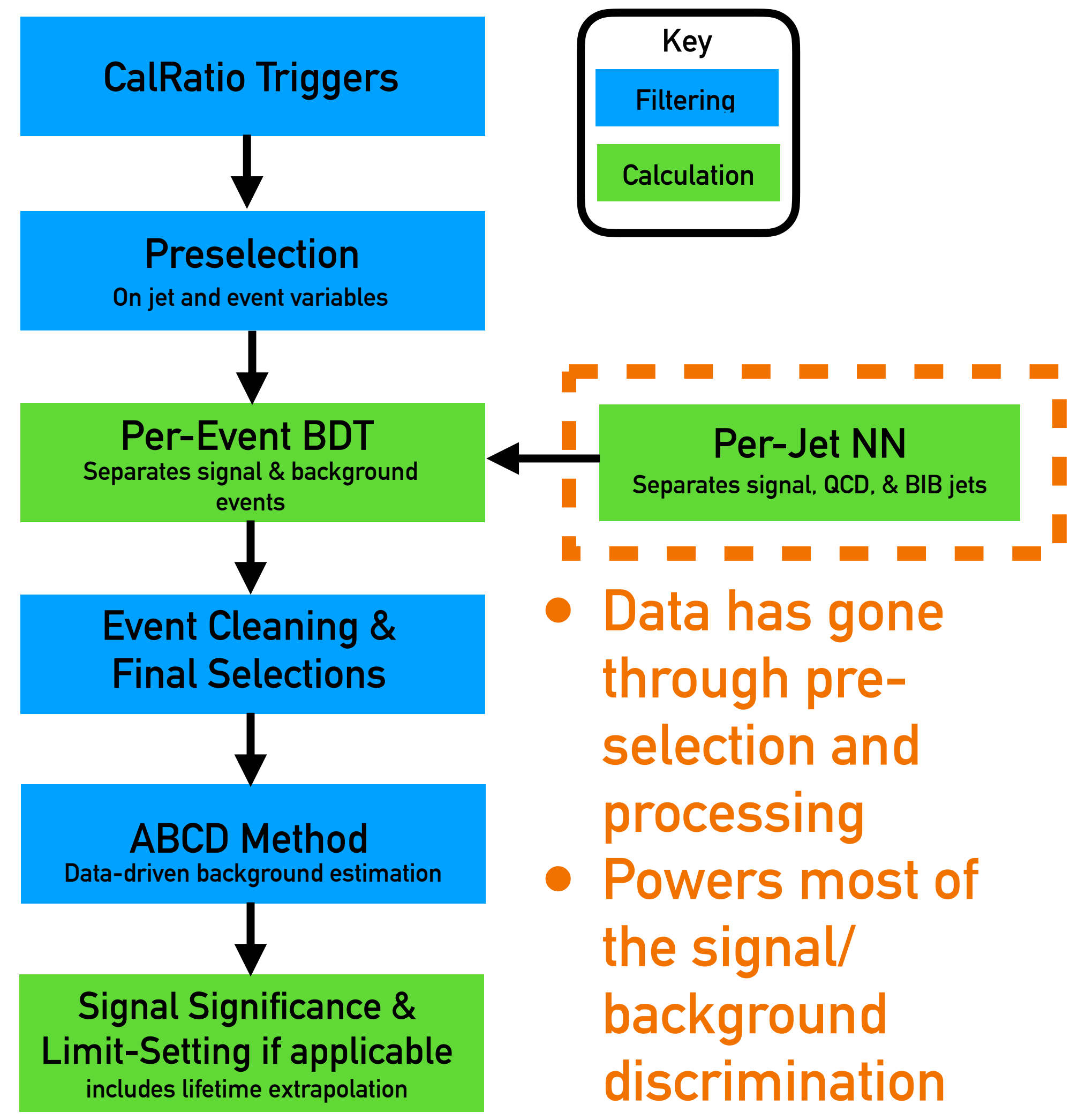
SM Multijet

- Cross section ~10^{10}
- Signal cross section ~10^{-2}
- Fluctuations in particles jet evolution creates signal-like jet

Beam Induced Background

- Doesn't occur from collision
- Ex. muon entering the detector, depositing energy in calorimeter, appearing like LLP
- Can filter out using timing differences

Analysis Overview


CalRatio Triggers

Preselection
On jet and event variables

Per-Event BDT
Separates signal & background events

Per-Jet NN
Separates signal, QCD, & BIB jets

Event Cleaning & Final Selections

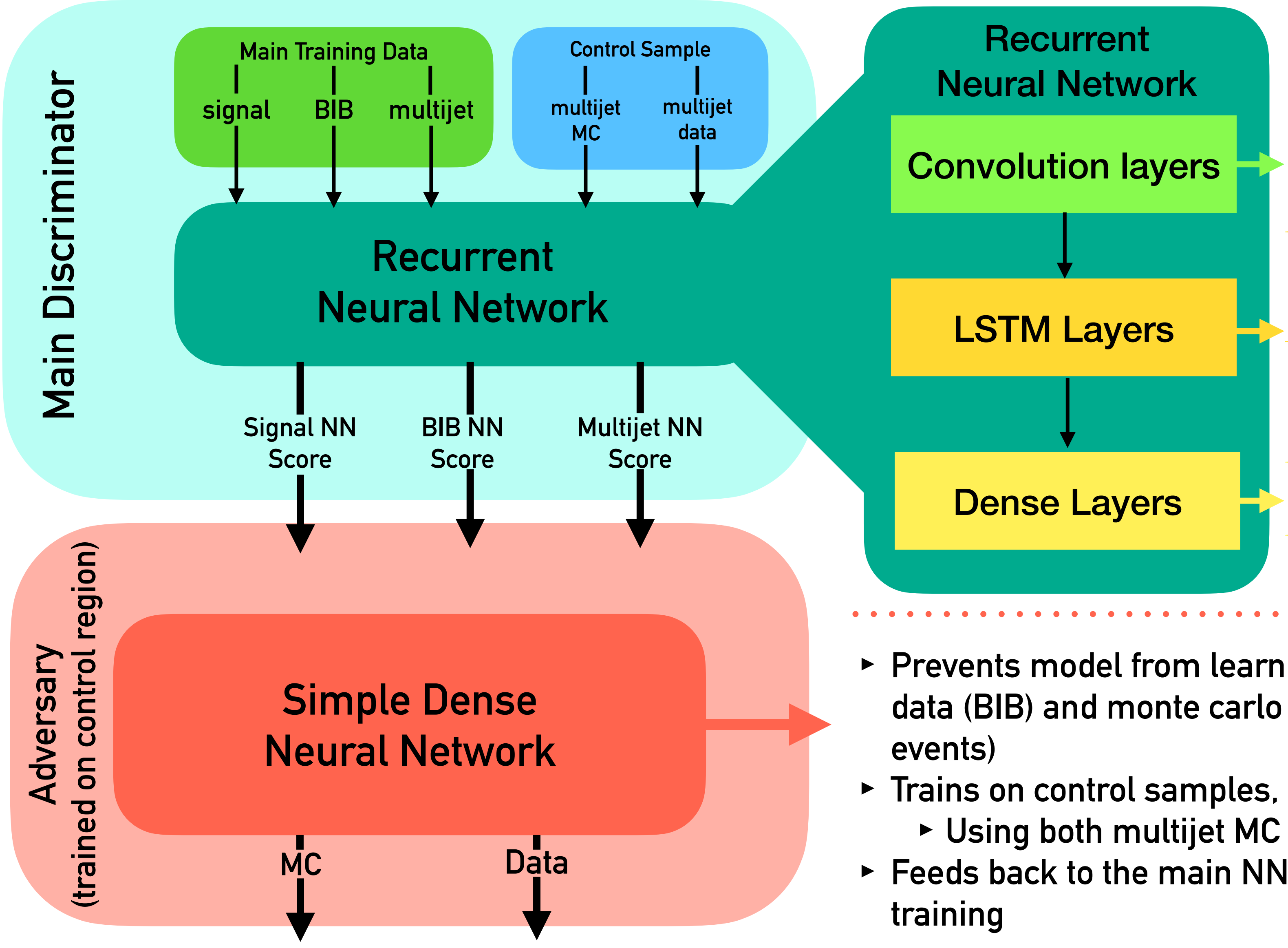
ABCD Method
Data-driven background estimation

Signal Significance & Limit-Setting if applicable
includes lifetime extrapolation

Key
Filtering
Calculation

- Data has gone through pre-selection and processing
- Powers most of the signal/background discrimination

Model Overview



- Feature extractor
- Exploits correlations between variables

- Remembers information between subsequent inputs
- Exploits correlations between cluster/tracks/muon segments

- Information concatenated with jet inputs
- Outputs predictions on classification

- Prevents model from learning difference between data (BIB) and monte carlo (signal and multijet events)
- Trains on control samples, separate from main data
  - Using both multijet MC and multijet data
- Feeds back to the main NN's loss function during training

Leveraging Modern Development Techniques

Testing & Continuous Integration

- Making sure that edge cases are handled in the code
- Paying attention to errors outputted by TensorFlow - they're there for a reason!
- Tests provided a way to check if changes impacted model performance
- Also provide feedback and continuous integration in GitHub repository

Cleaning Up the Code

- Initial code was written organically
  - Lots of redundant code that is no longer needed
- Large amounts of errors put out by TensorFlow made it impossible to tell what was important and what was not
- Complete working code located in multiple branches in the repository
- Running the code was more more difficult than required

Improving Training File Generation

- Original training files were created in a convoluted library for our data (pandas)
- Rewrote generation code to use awkward awkward arrays
  - Better suited for non rectangular data common in HEP analysis
- Code to create training files now exists in the repository with main network code

Modernization Improves

Sustainability

- Code is now easily accessible on a public GitHub page
  - All necessary code is on one branch
- Cleaned up code leads to easier modification without risk of breaking it

Functionality

- Command line interface makes running the training easier
  - No need to look through multiple python files to run training
- Added new functions to analyze training results and performance

Versatility

- New training file creation code allows us to use new data for training the model
- Ease of training model allows future analyses to train the network on new data

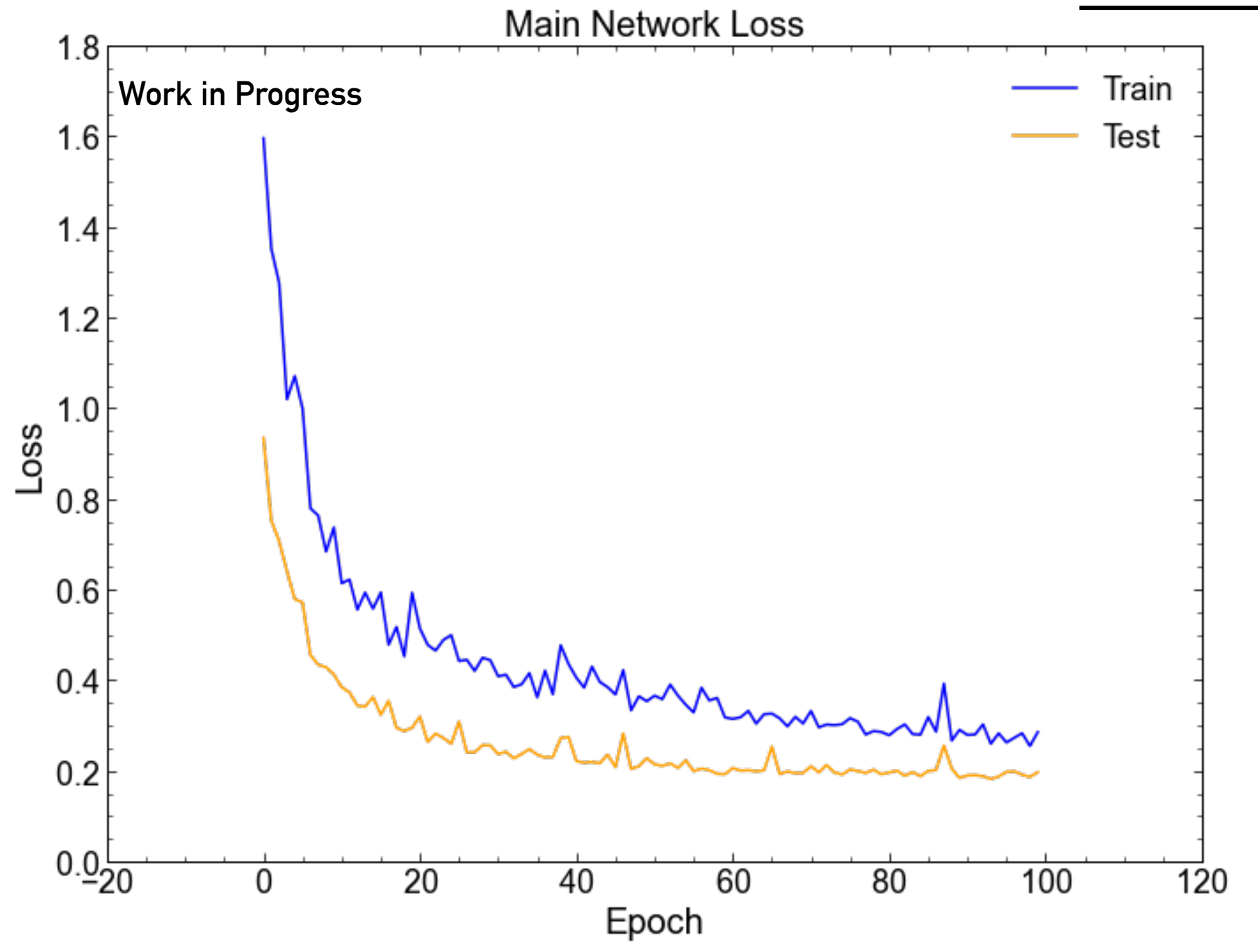
Performance

- Training time reduced substantially
  - Originally over 24 hours to complete training, now ~2 hours
- Less GPU memory used in training
  - Can train on an 11gb GPU, originally required 40gb

Model Performance

- Began with the original version of the model which was written in TensorFlow 1.12
- Updated to TensorFlow 2.11
- Led problems with the model
  - Memory leak and poor network performance
  - Ultimately due to mismatch of how to call the optimizer function in TF 1.12 vs 2.11

Loss Plot



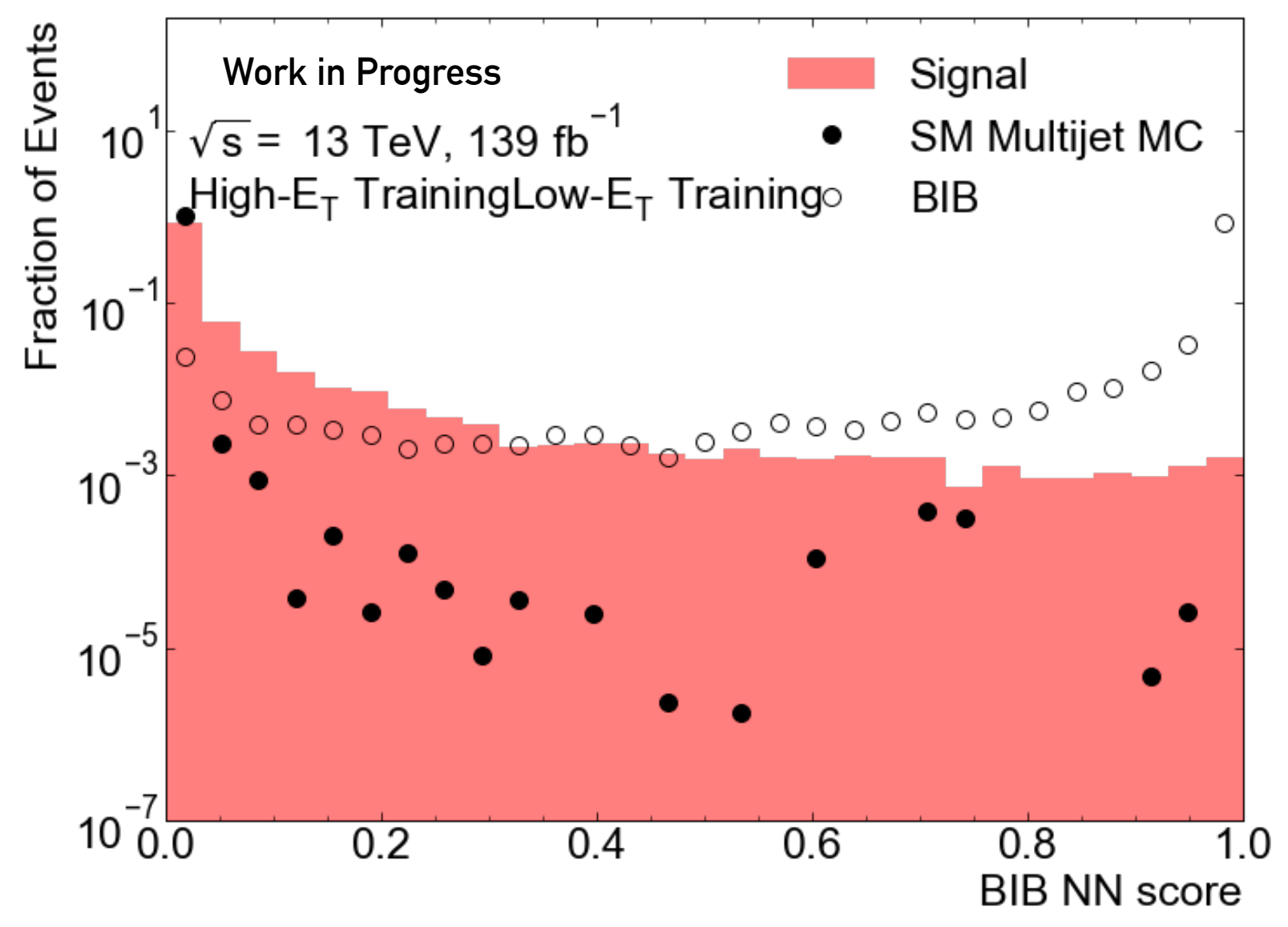
- Loss plot shows how far off the model is from being correct
  - During training, model attempts to minimize loss
- Want to have a loss as close to 0 as possible, ideally < 1
  - Loss here quickly drops below 1
- Plateaus around 0.3, indicates a good performance of the network

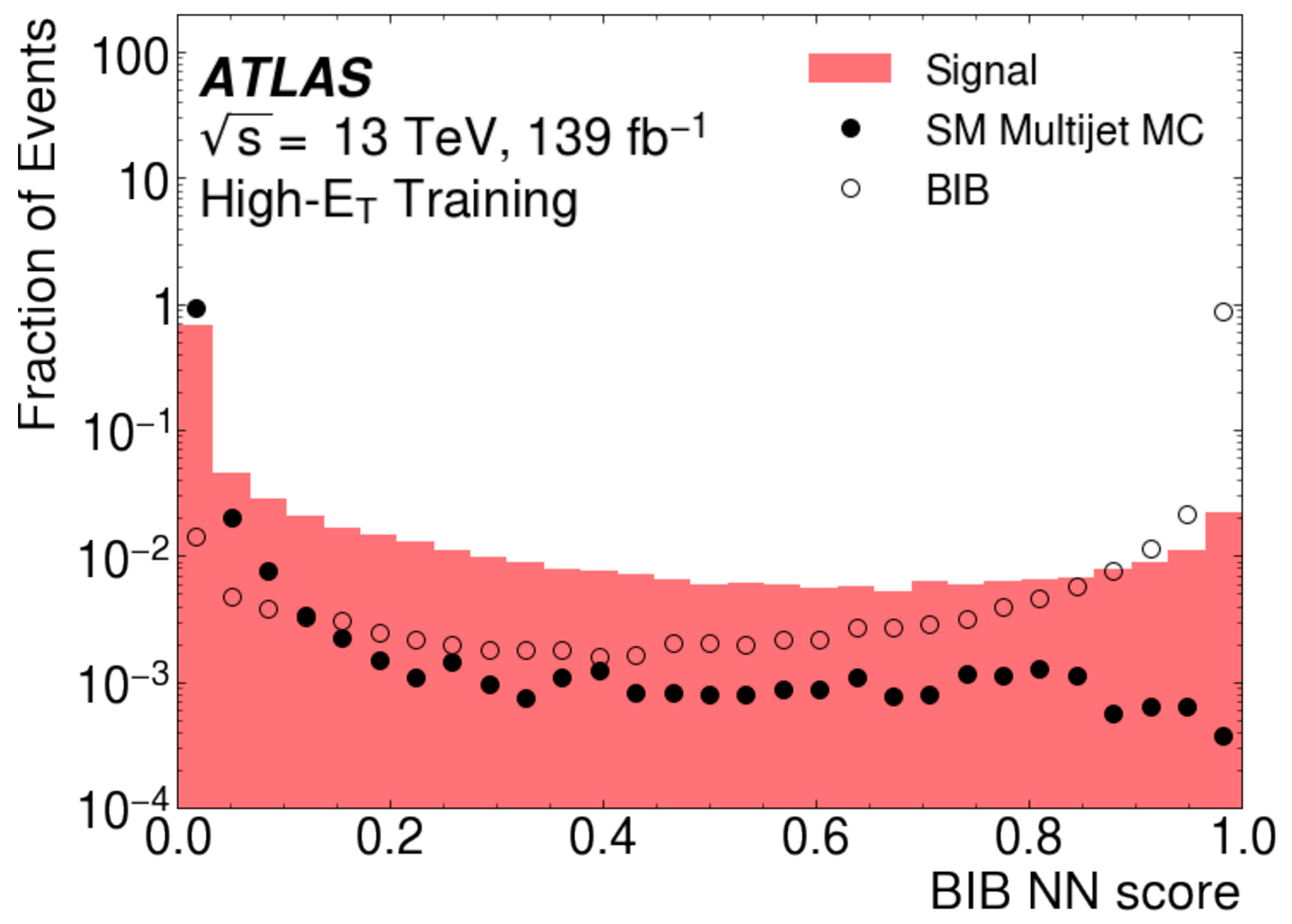
New vs Original Neural Net Score Histograms

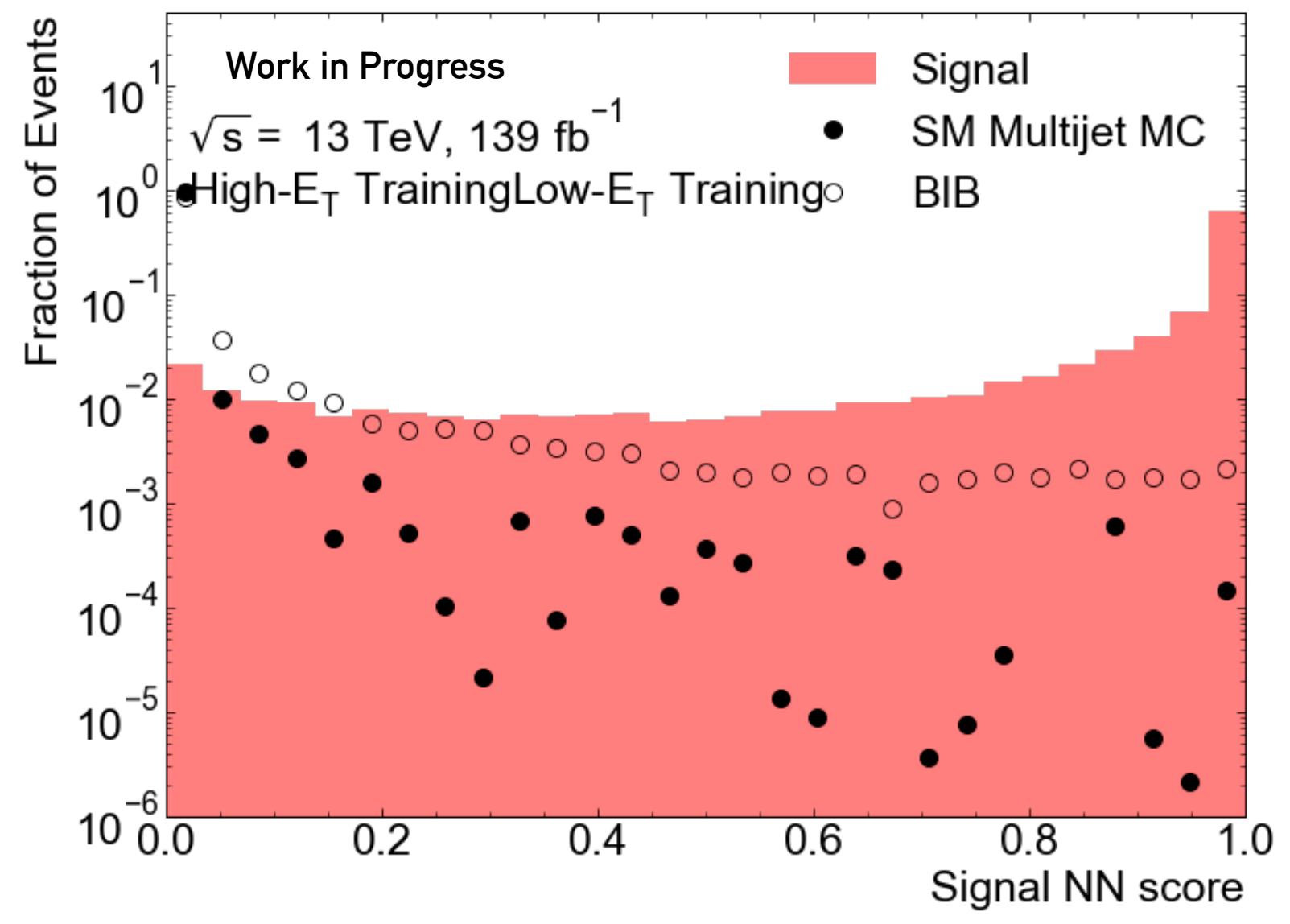
- Histograms show model confidence in classifying BIB/signal/multijet appropriately
- Want BIB/signal/multijet datapoints to have a BIB/signal/multijet NN score mostly at 1
  - NN score is the numeric value corresponding to the classification confidence
- New results generally match the previous results, most of the events being classified correctly.

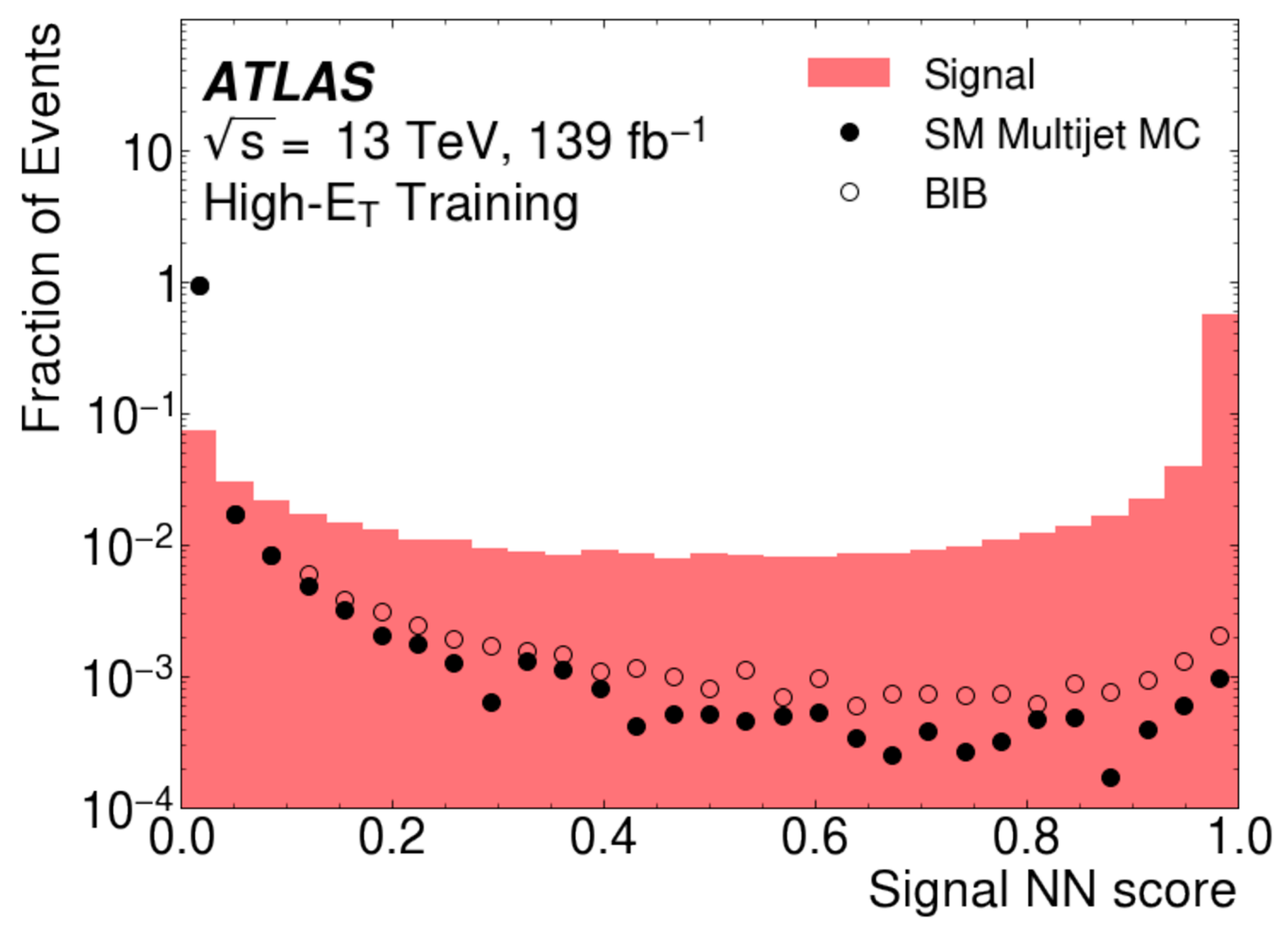
New

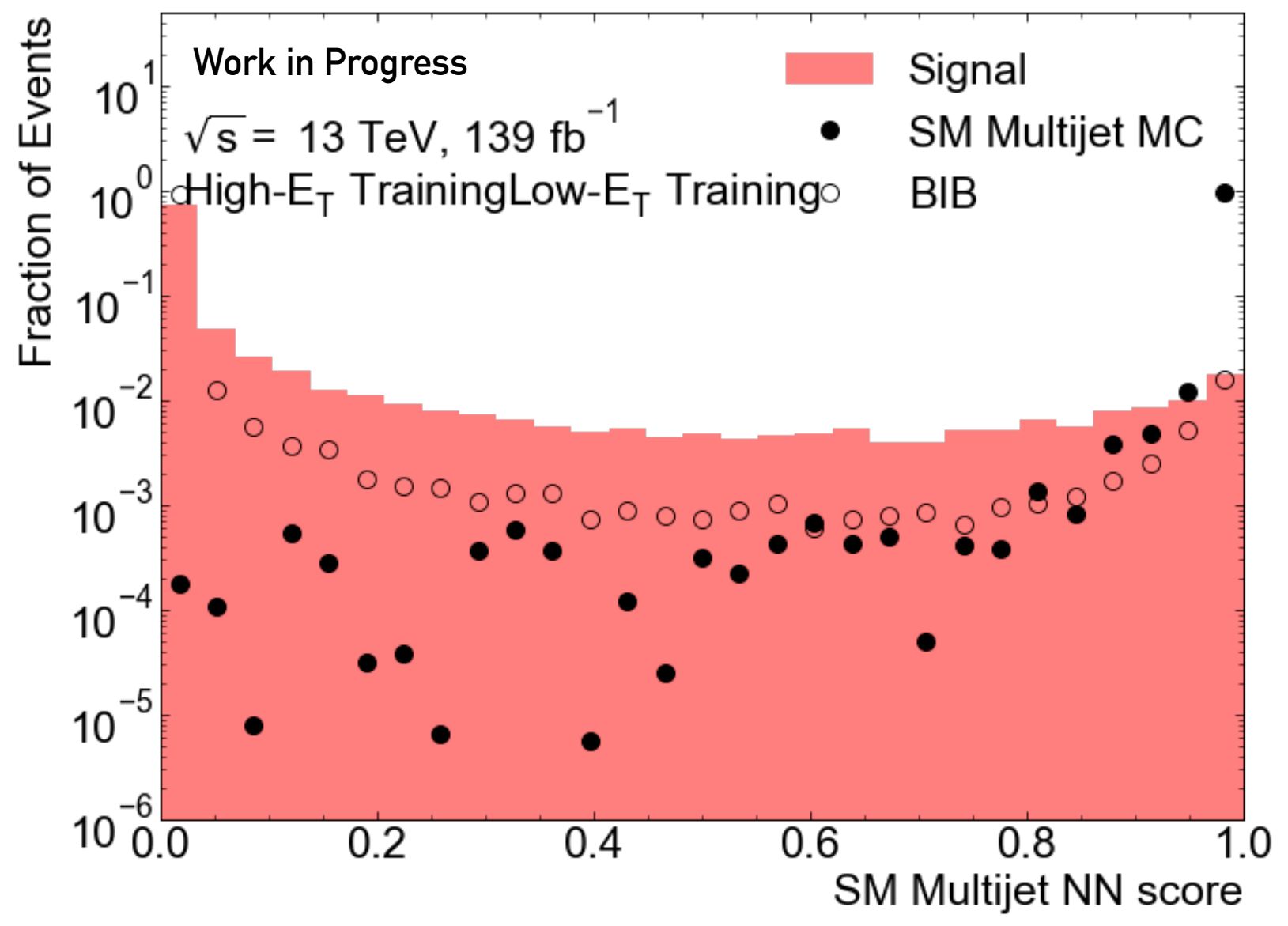
Original

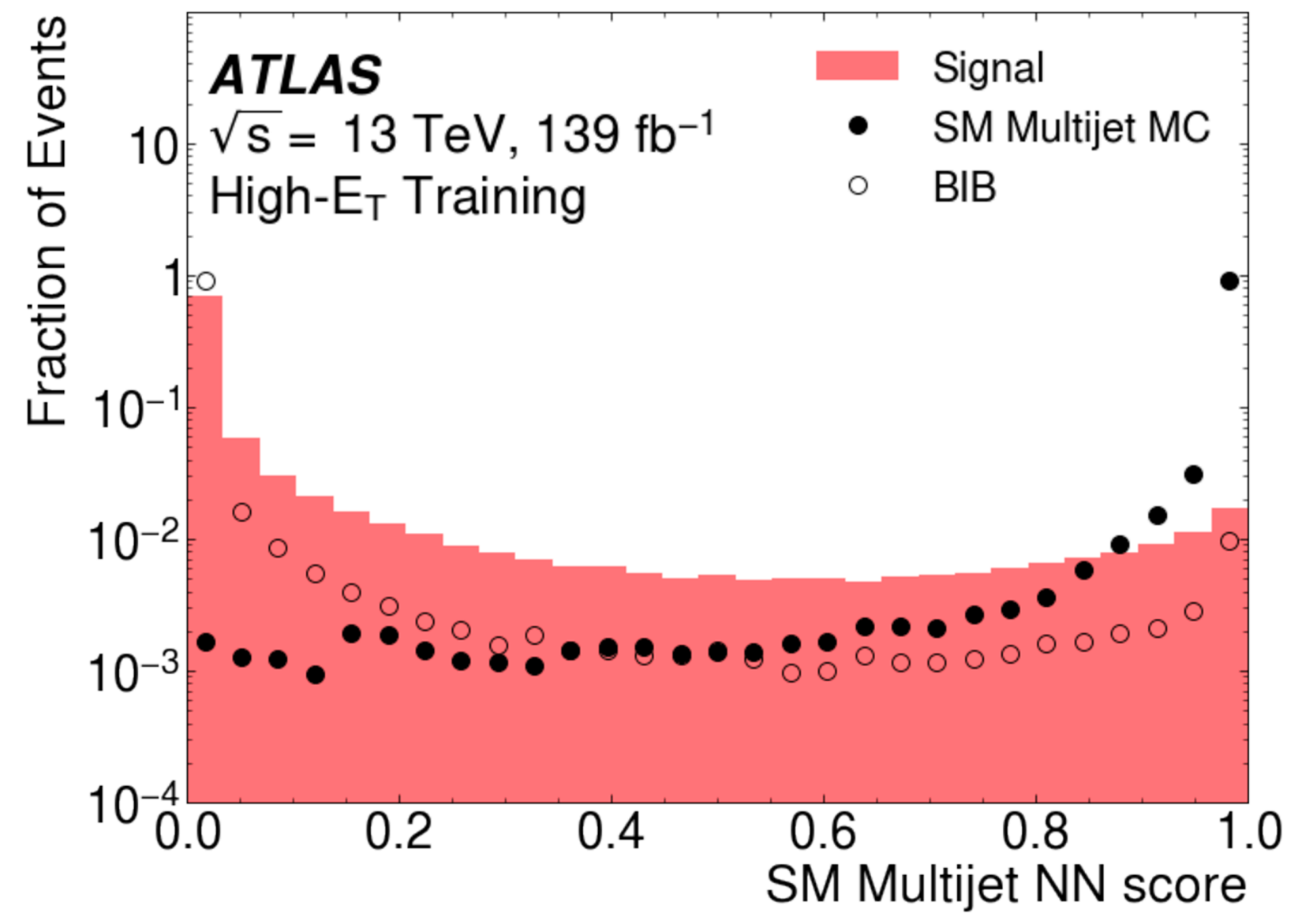

Work in Progress
$\sqrt{s}$ = 13 TeV, 139 fb^{-1}
High-E_T TrainingLow-E_T Training
Signal
SM Multijet MC
BIB
Fraction of Events
BIB NN score
10^{1}
10^{-1}
10^{-3}
10^{-5}
10^{-7}


ATLAS
$\sqrt{s}$ = 13 TeV, 139 fb^{-1}
High-E_T Training
Signal
SM Multijet MC
BIB
Fraction of Events
BIB NN score
10^{-1}
10^{-2}
10^{-3}
10^{-4}


Work in Progress
$\sqrt{s}$ = 13 TeV, 139 fb^{-1}
High-E_T TrainingLow-E_T Training
Signal
SM Multijet MC
BIB
Fraction of Events
Signal NN score
10^{1}
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}


ATLAS
$\sqrt{s}$ = 13 TeV, 139 fb^{-1}
High-E_T Training
Signal
SM Multijet MC
BIB
Fraction of Events
Signal NN score
10^{-1}
10^{-2}
10^{-3}
10^{-4}


Work in Progress
$\sqrt{s}$ = 13 TeV, 139 fb^{-1}
High-E_T TrainingLow-E_T Training
Signal
SM Multijet MC
BIB
Fraction of Events
SM Multijet NN score
10^{1}
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}


ATLAS
$\sqrt{s}$ = 13 TeV, 139 fb^{-1}
High-E_T Training
Signal
SM Multijet MC
BIB
Fraction of Events
SM Multijet NN score
10^{-1}
10^{-2}
10^{-3}
10^{-4}