Penn

Jet Calibration in ATLAS Using Machine Learning

Benji Lunday (University of Pennsylvania)

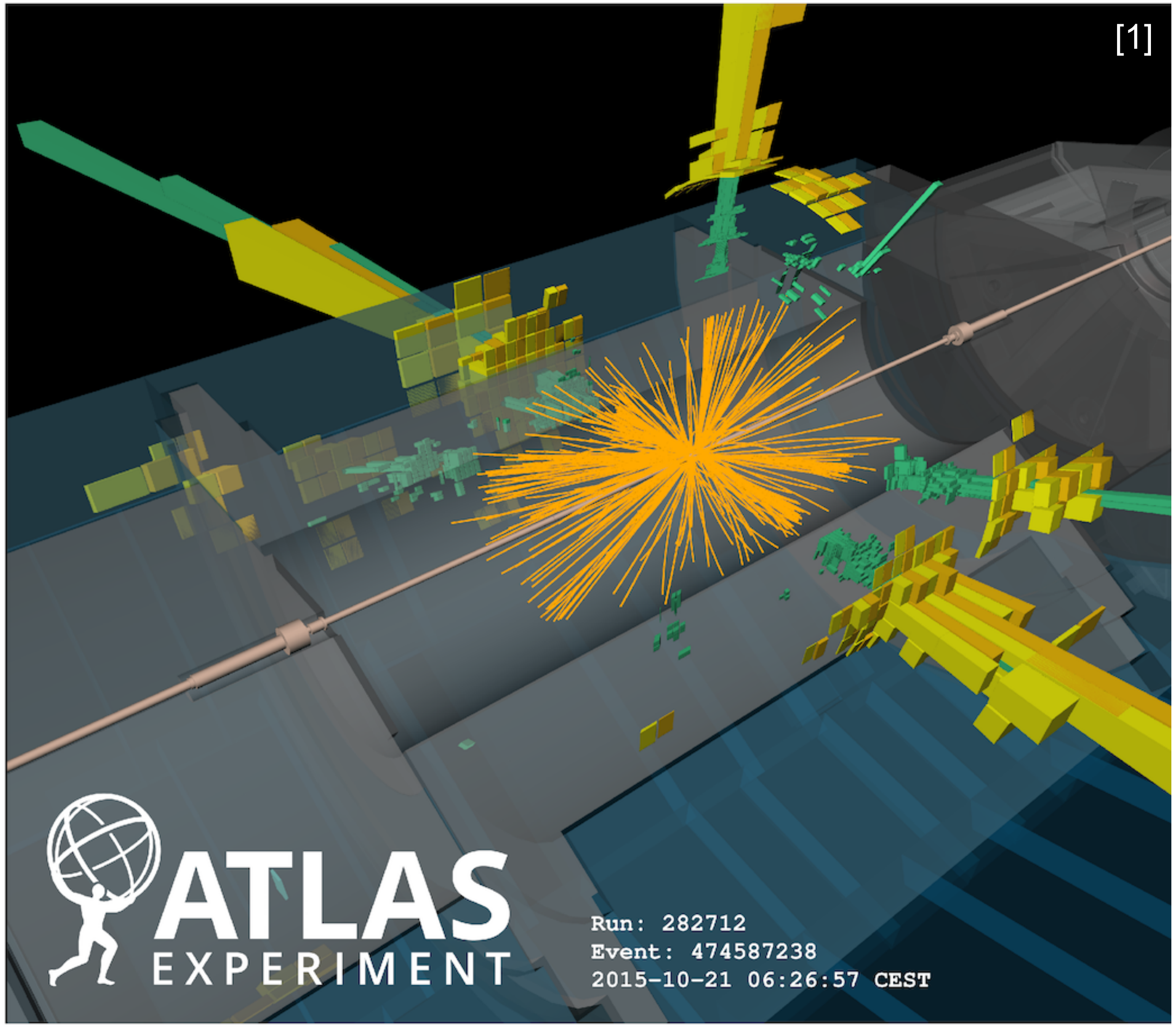
May 15th, 2024

ATLAS EXPERIMENT

Introduction: Jets in ATLAS

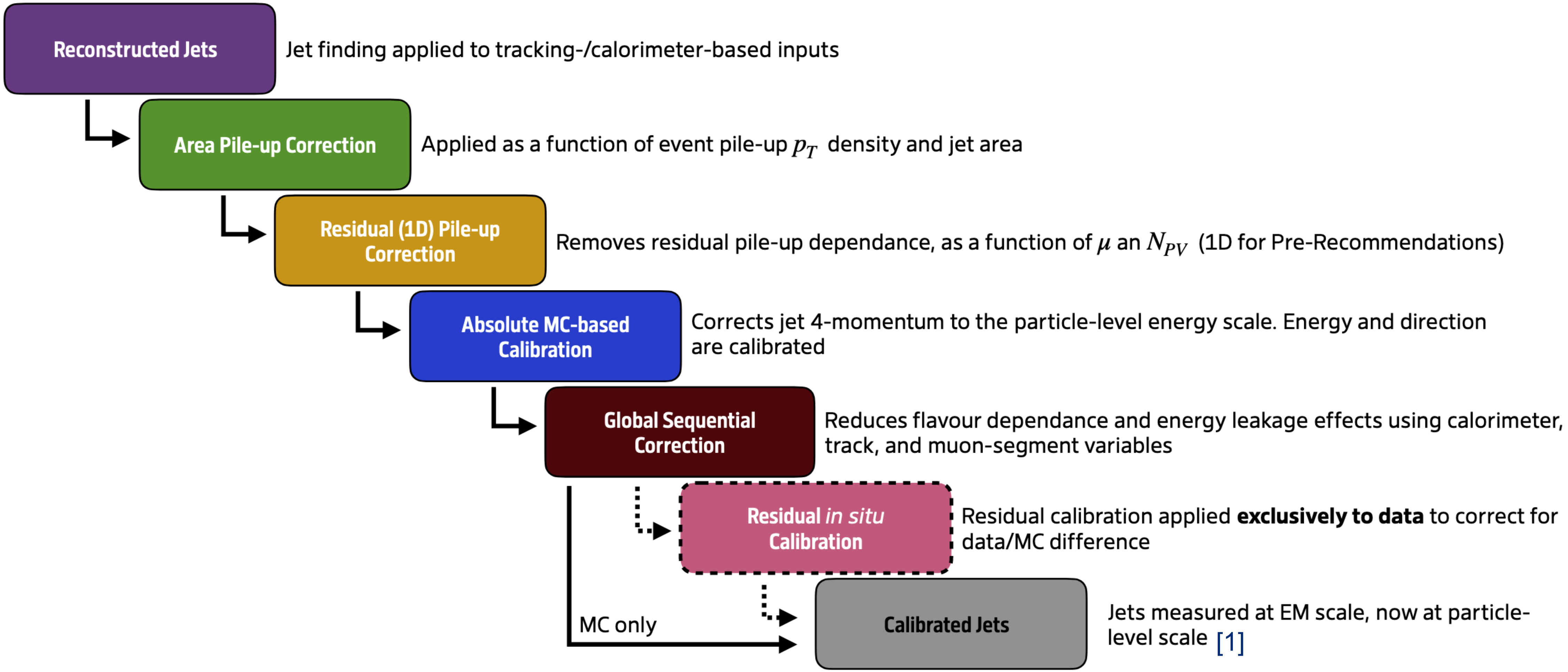
Jets in ATLAS

- Proton collisions result in high-energy particles which pass through detector
  - Jets: Collimated sprays of particles initiated by quarks and gluons
- ATLAS jets built from EM-scale calorimeter energy deposits and tracking information
  - Using anti-k_t jet algorithm with $R = 0.4$ for small-R $(1.0$ for large-R) jets
- End result: object representing best reconstruction of detected parton's energy and direction



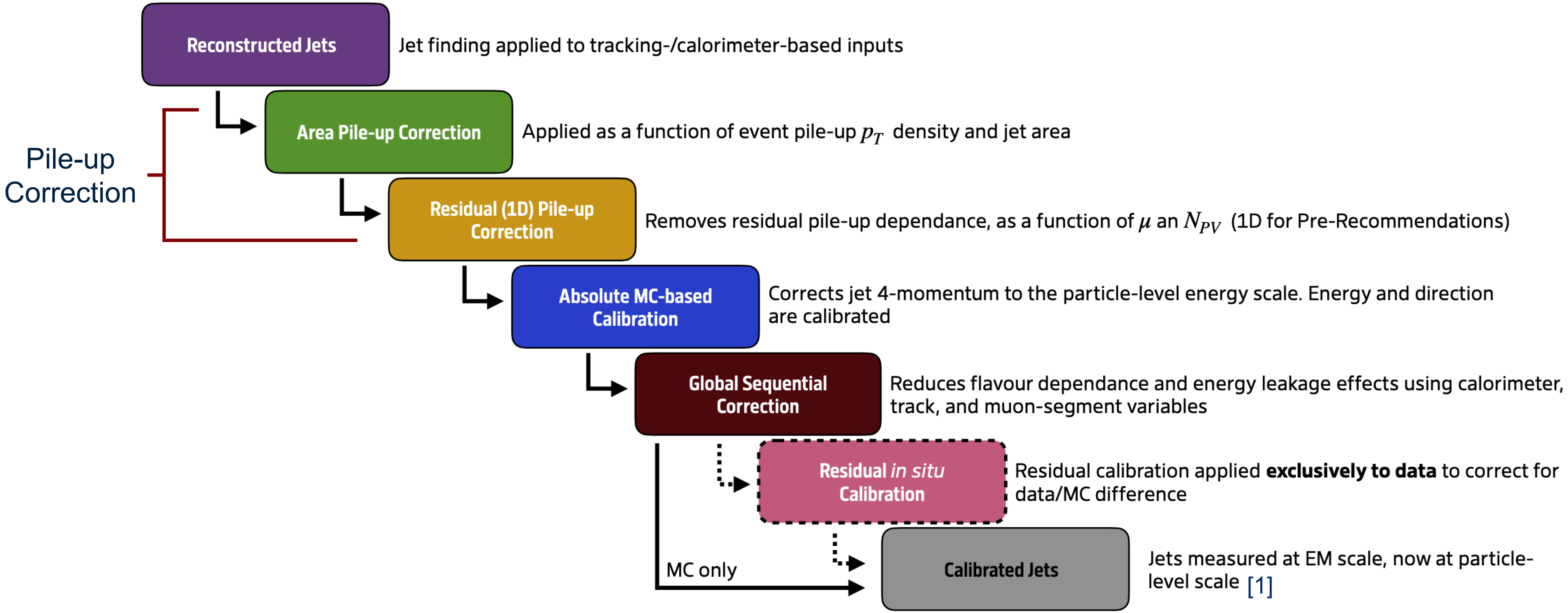
[1]: High-pT multi-jet final states at ATLAS and CMS

Current Calibration



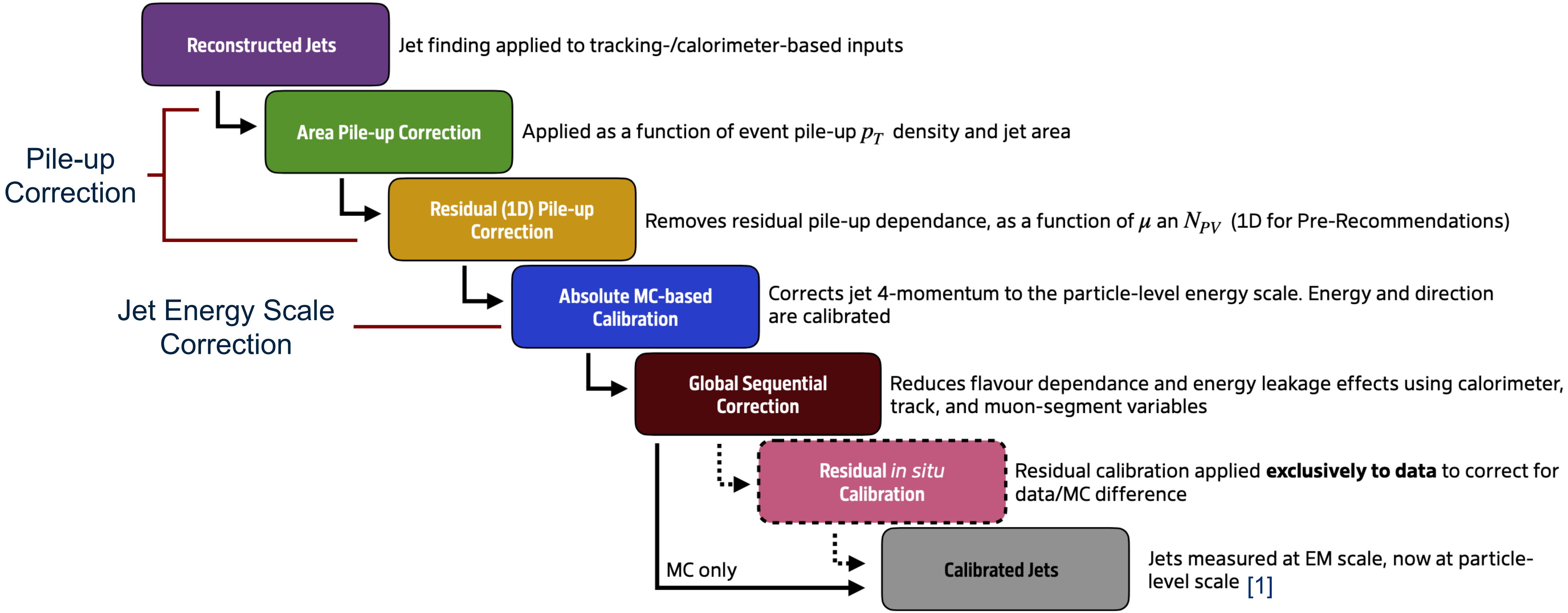
[1]: Jet energy calibration at the LHC

Current Calibration



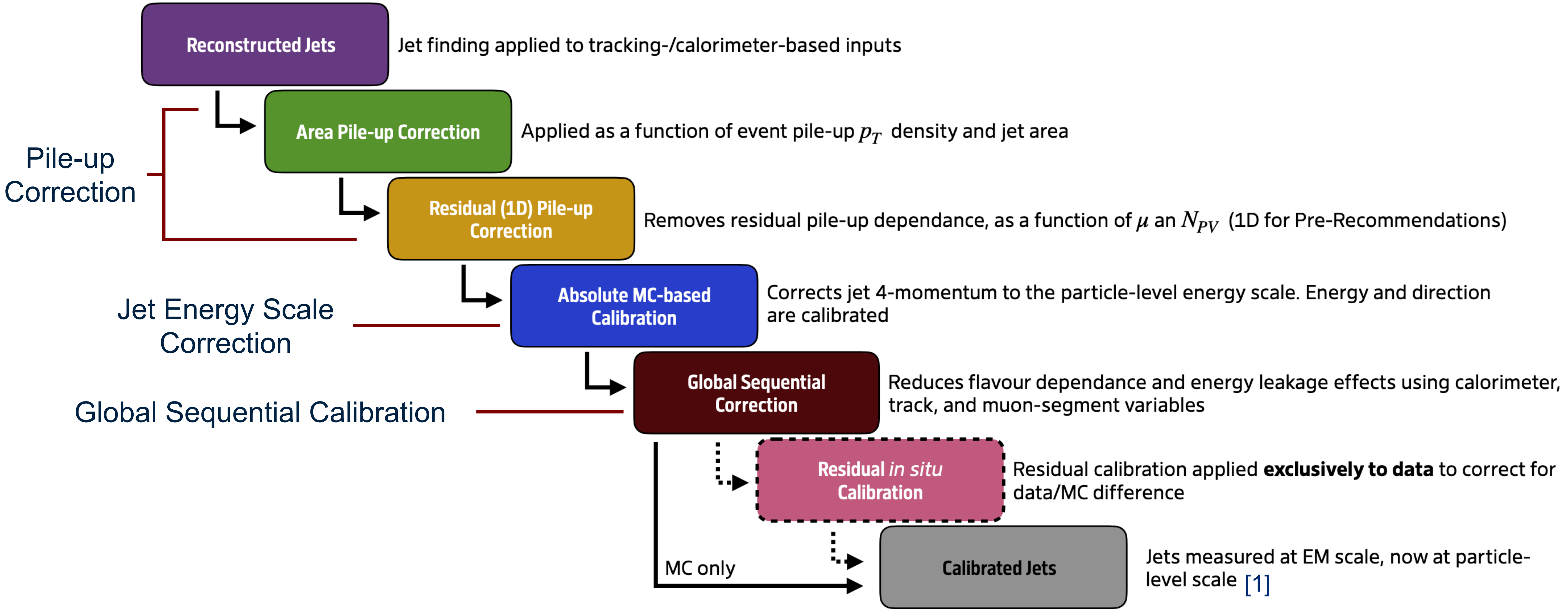
[1]: Jet energy calibration at the LHC

Current Calibration



[1]: Jet energy calibration at the LHC

Current Calibration

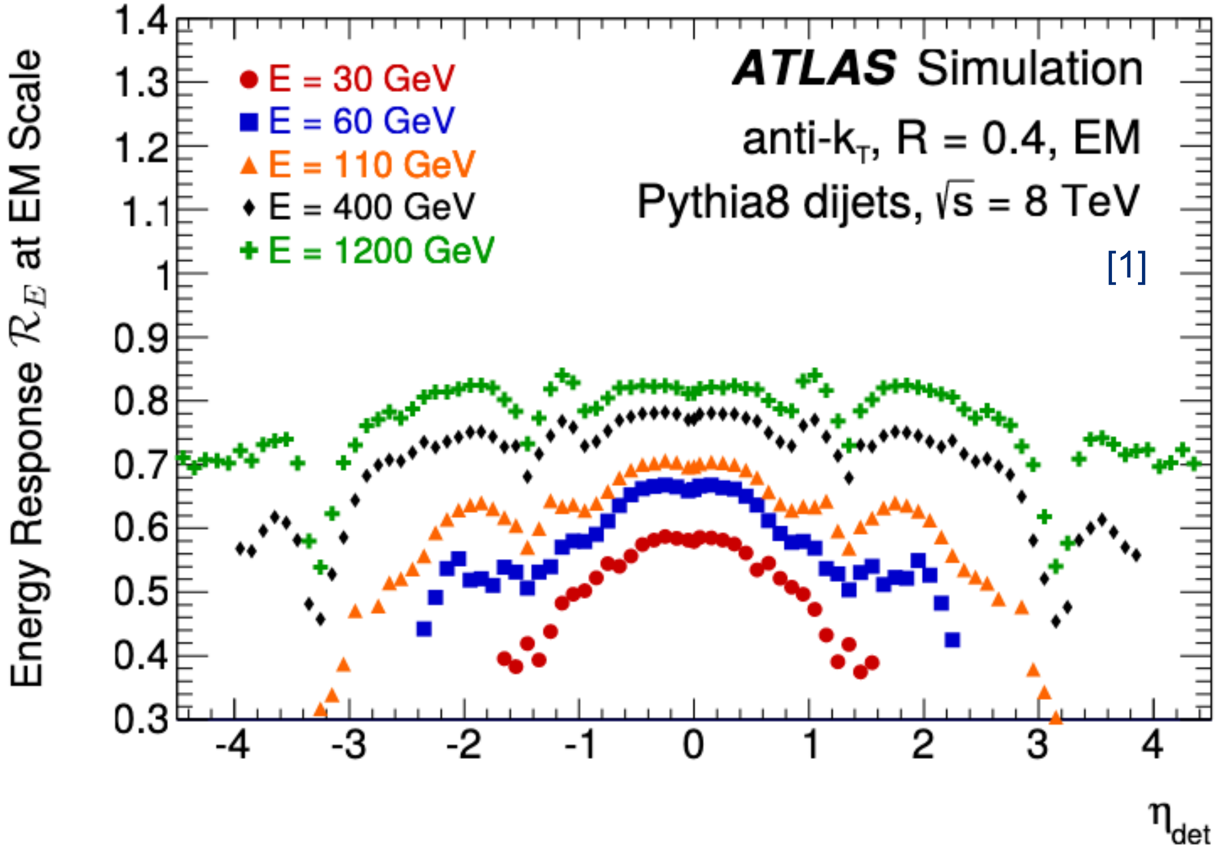


[1]: Jet energy calibration at the LHC

The Machine Learning Approach

Why Machine Learning?

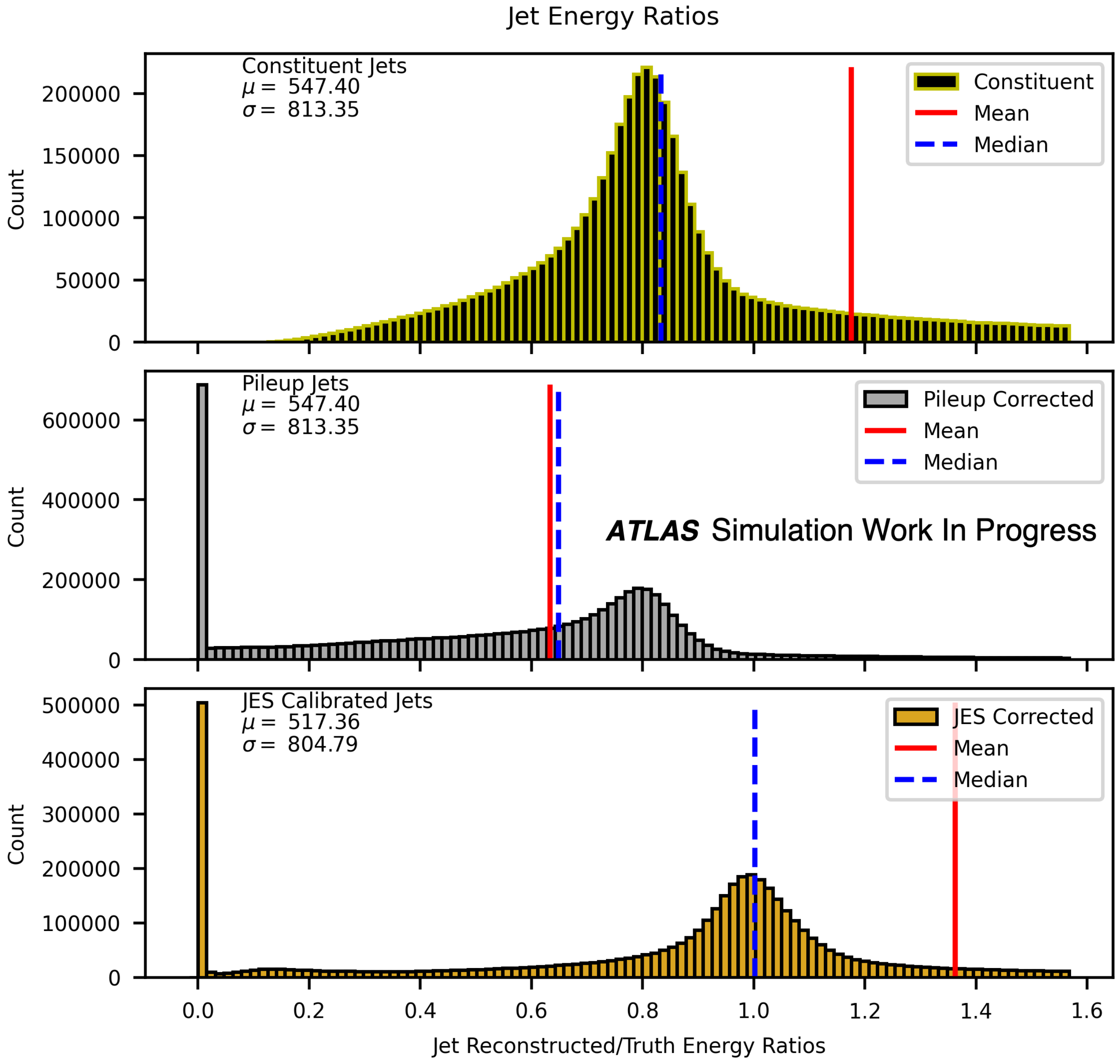
- Current calibration costly in both time and effort
  - ~1 year per full calibration
- Pile-up correction results in artifacts which must be corrected
- ML approach to GSC and large-R jets already successful
- **Goal:** Motivate and implement a ML network for small-R pile-up and JES calibrations in the HL-LHC



[1]: Determination of jet calibration and energy resolution in proton-proton collisions at √s = 8 TeV using the ATLAS detector

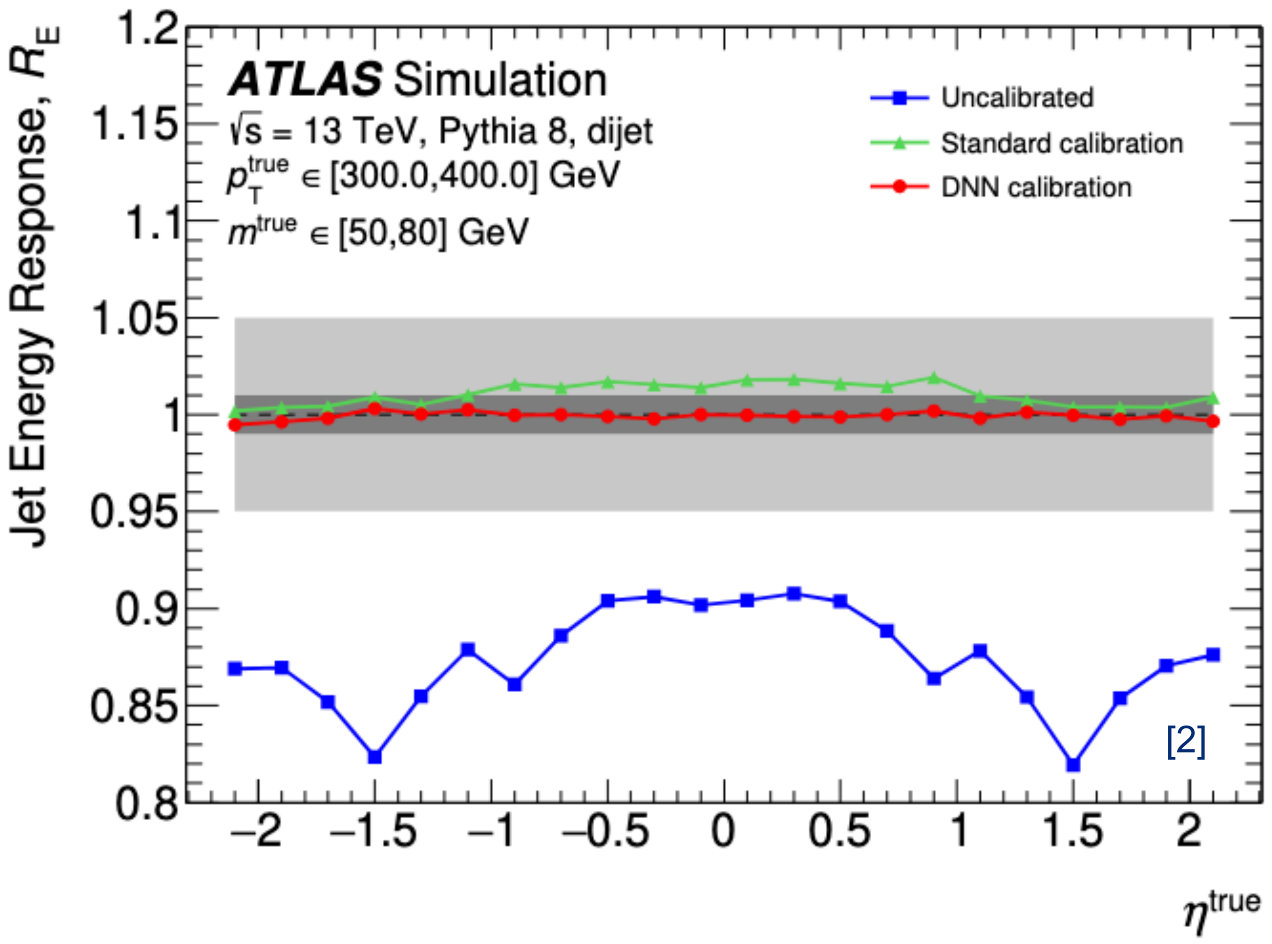
Why Machine Learning?

- Current calibration costly in both time and effort
  - ~1 year per full calibration
- Pile-up correction results in artifacts which must be corrected
- ML approach to GSC and large-R jets already successful
- **Goal:** Motivate and implement a ML network for small-R pile-up and JES calibrations in the HL-LHC



Why Machine Learning?

- Current calibration costly in both time and effort
  - ~1 year per full calibration
- Pile-up correction results in artifacts which must be corrected
- ML approach to GSC and large-R jets already successful
- **Goal:** Motivate and implement a ML network for small-R pile-up and JES calibrations in the HL-LHC



[2]: Simultaneous energy and mass calibration of large-radius jets with the ATLAS detector using a deep neural network

Network Parameters

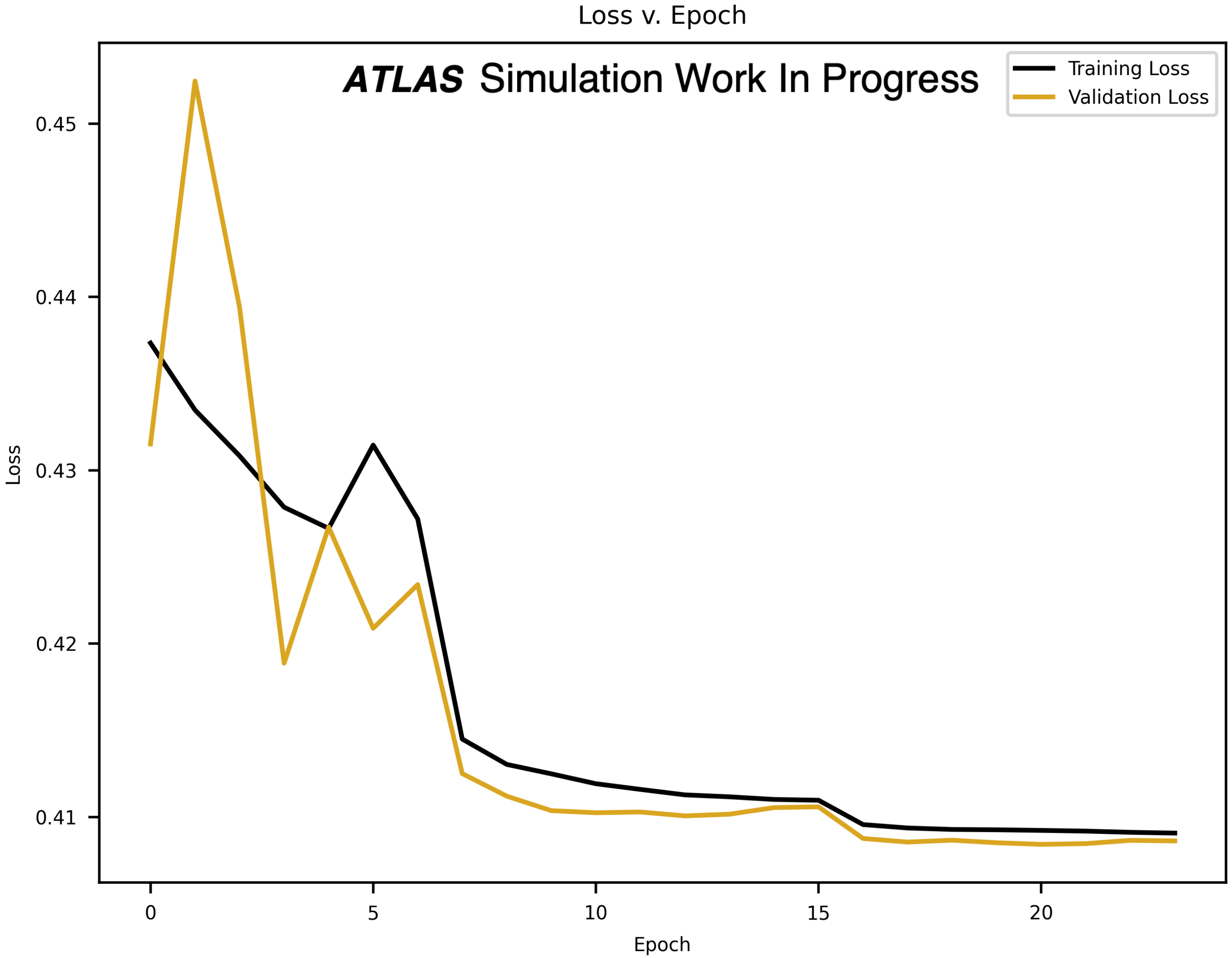
- Using MC EMTopo jets with $|\eta| \leq 4.5$
  - ~ 5 million jets
- Use mean absolute error (MAE) loss function
  - $$\mathscr{L} = \frac{1}{n}\sum_{n}^{i=1}\left|y_{i,pred} - y_{i,true}\right|$$
  - Target distribution median
  - Avoid sensitivity to outliers
  - Similar target to Large-R network

```
# Build TF model
NEPOCHS = 40
BATCH_SIZE = 4096
LR = 0.01

# Define normalization layer
norm_layer = layers.Normalization()
norm_layer.adapt(X_train)

# Define layers/node count
jet_model = tf.keras.Sequential([
  norm_layer,
  layers.Dense(32, activation='relu'),
  layers.Dense(32, activation='relu'),
  layers.Dense(64, activation='relu'),
  layers.Dense(64, activation='relu'),
  layers.Dense(128, activation='relu'),
  layers.Dense(256, activation='relu'),
  layers.Dense(128, activation='relu'),
  layers.Dense(64, activation='relu'),
  layers.Dense(64, activation='relu'),
  layers.Dense(32, activation='relu'),
  layers.Dense(32, activation='relu'),
  layers.Dense(1, activation='relu')
])

# Compile
jet_model.compile(loss = tf.keras.losses.MeanAbsoluteError(), optimizer = tf.keras.

# Add callbacks
reduce_lr = tf.keras.callbacks.ReduceLROnPlateau(monitor='val_loss', factor=0.1, pa

# Train
history = jet_model.fit(X_train, (y_train['pseudo_RT']), epochs=NEPOCHS, batch_size
```

Loss

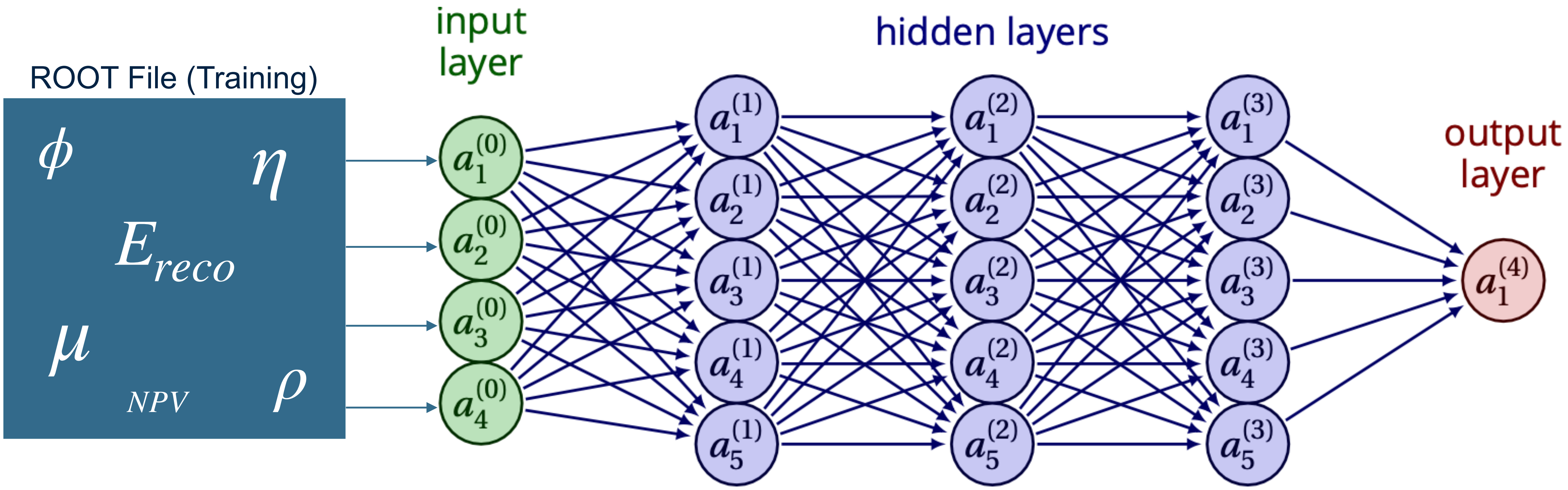
Network Parameters

- Provide most of jet 4-vector $(E_{Constit}, \eta, \phi)$ at constituent level
  - Also provide pile-up information (NPV, μ, ρ, A)
- Main correction target: $\dfrac{E_{Constit} - \rho \cdot A}{E_{Truth}}$
- Train for 25 epochs with variable learning rate
- Loss convergence for testing/validation set
- Quick training means faster development



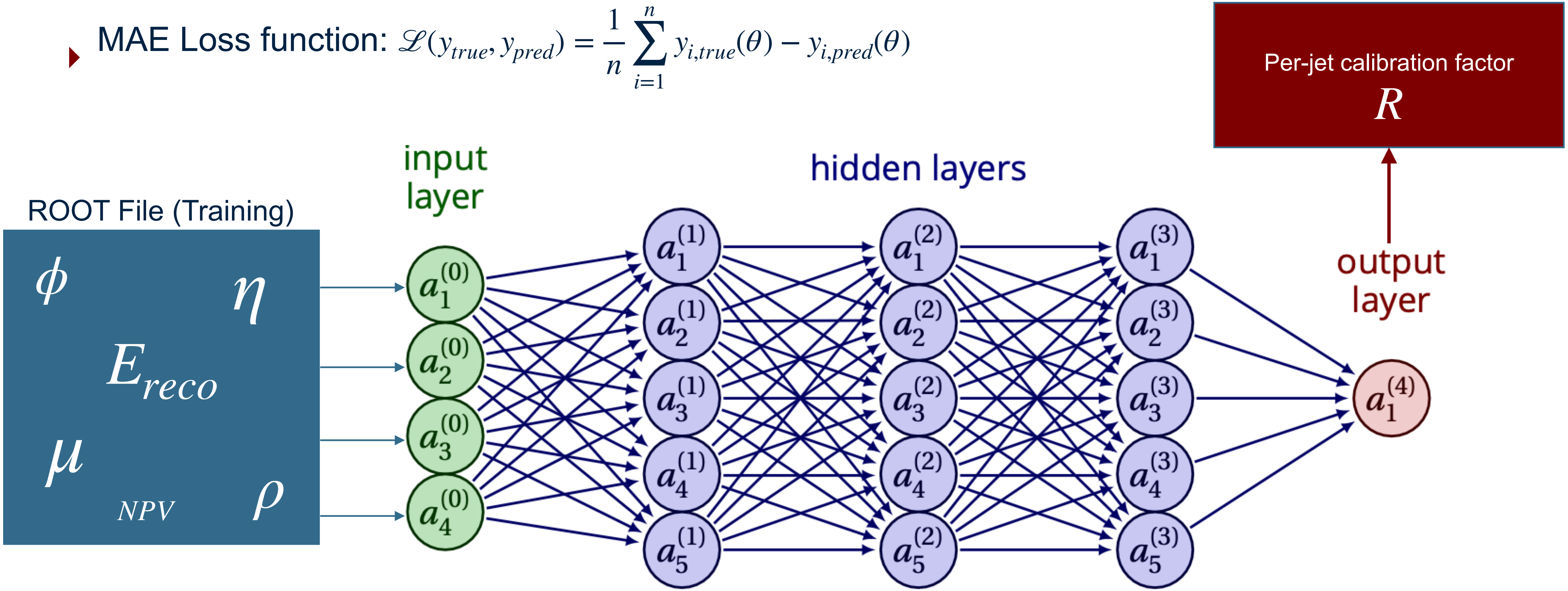
Network Structure

- Motivate simplest possible DNN to perform calibration
- MAE Loss function: $\mathscr{L}(y_{true}, y_{pred}) = \frac{1}{n}\sum_{i=1}^{n} y_{i,true}(\theta) - y_{i,pred}(\theta)$



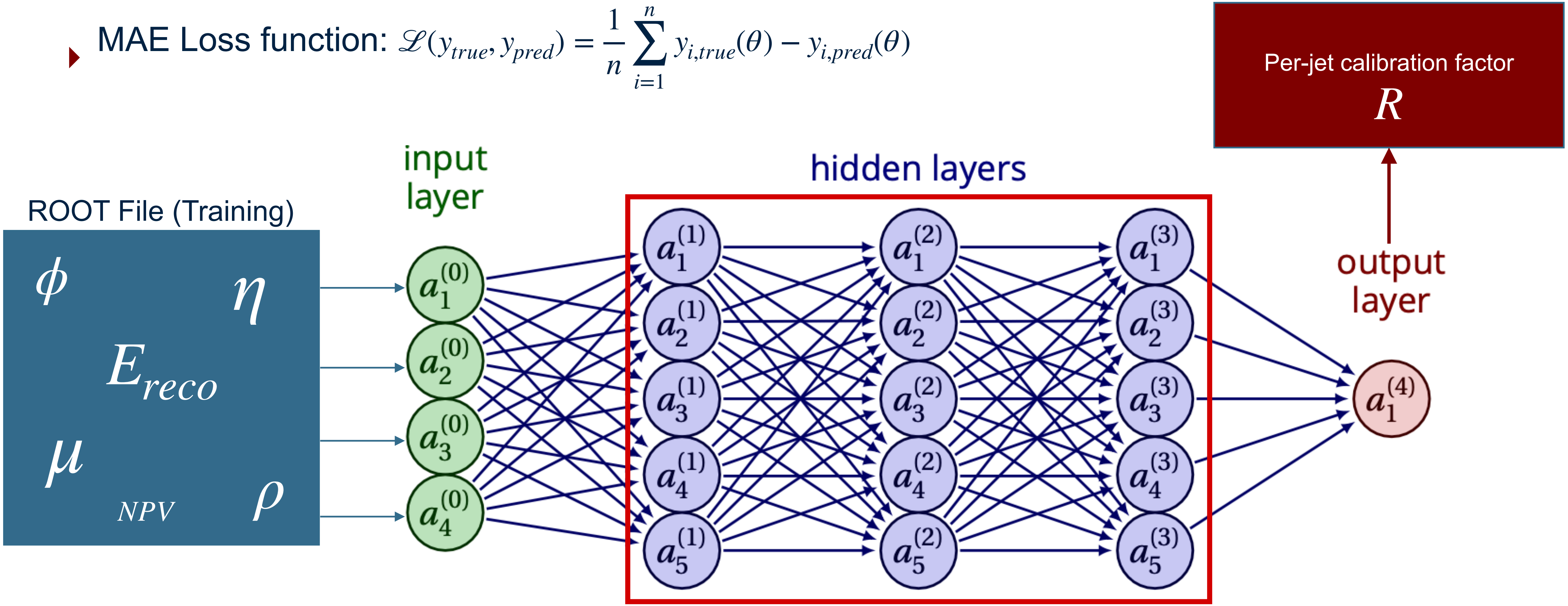
Network Structure

- Motivate simplest possible DNN to perform calibration
- MAE Loss function: $\mathcal{L}(y_{true}, y_{pred}) = \frac{1}{n}\sum_{i=1}^{n} y_{i,true}(\theta) - y_{i,pred}(\theta)$



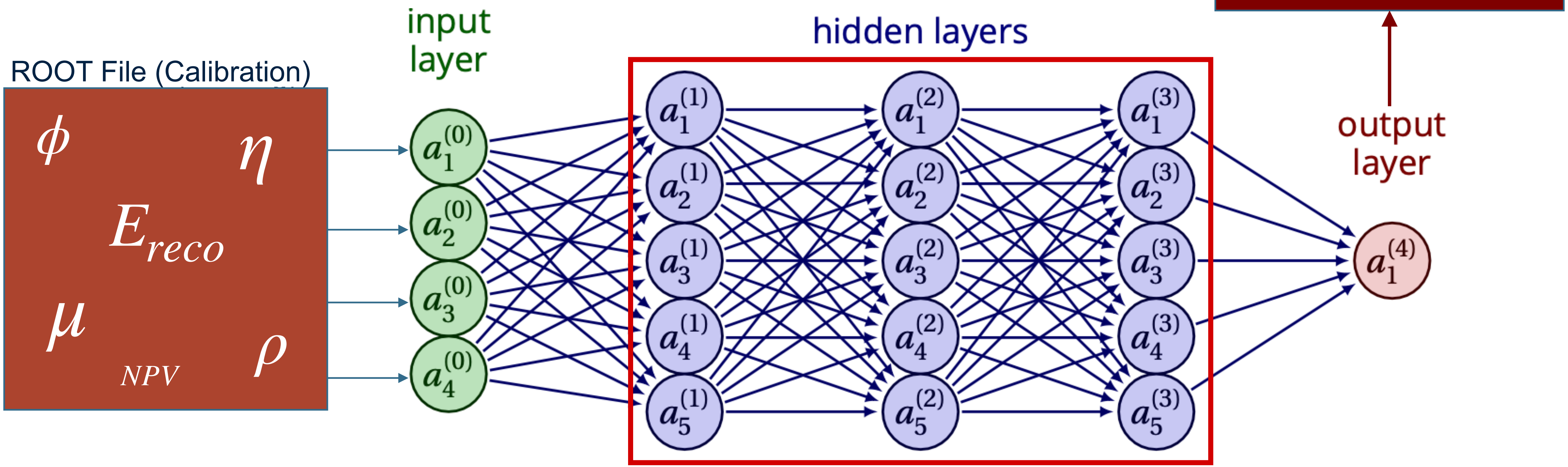
Network Structure

- Motivate simplest possible DNN to perform calibration
- MAE Loss function: $\mathscr{L}(y_{true}, y_{pred}) = \frac{1}{n}\sum_{i=1}^{n} y_{i,true}(\theta) - y_{i,pred}(\theta)$



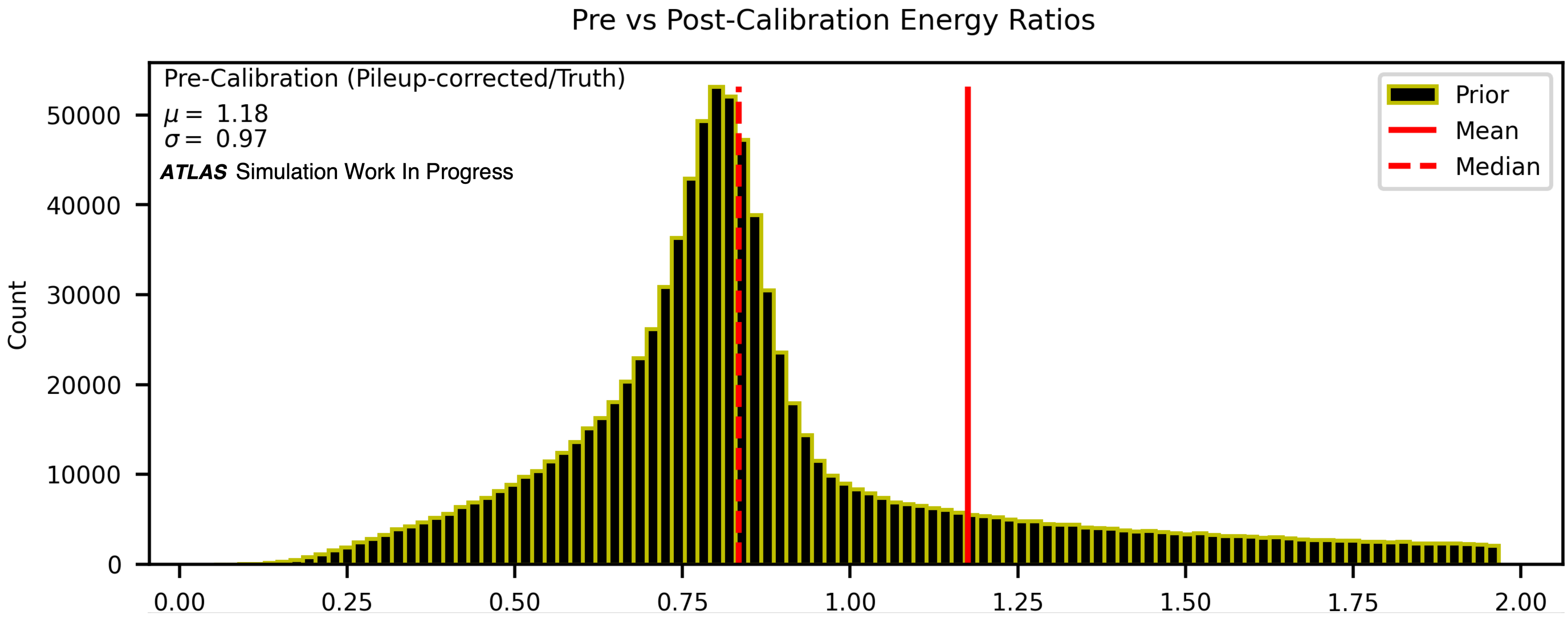
Network Structure

- Motivate simplest possible DNN to perform calibration
- MAE Loss function: $\mathscr{L}(y_{true}, y_{pred}) = \frac{1}{n}\sum_{i=1}^{n} y_{i,true}(\theta) - y_{i,pred}(\theta)$



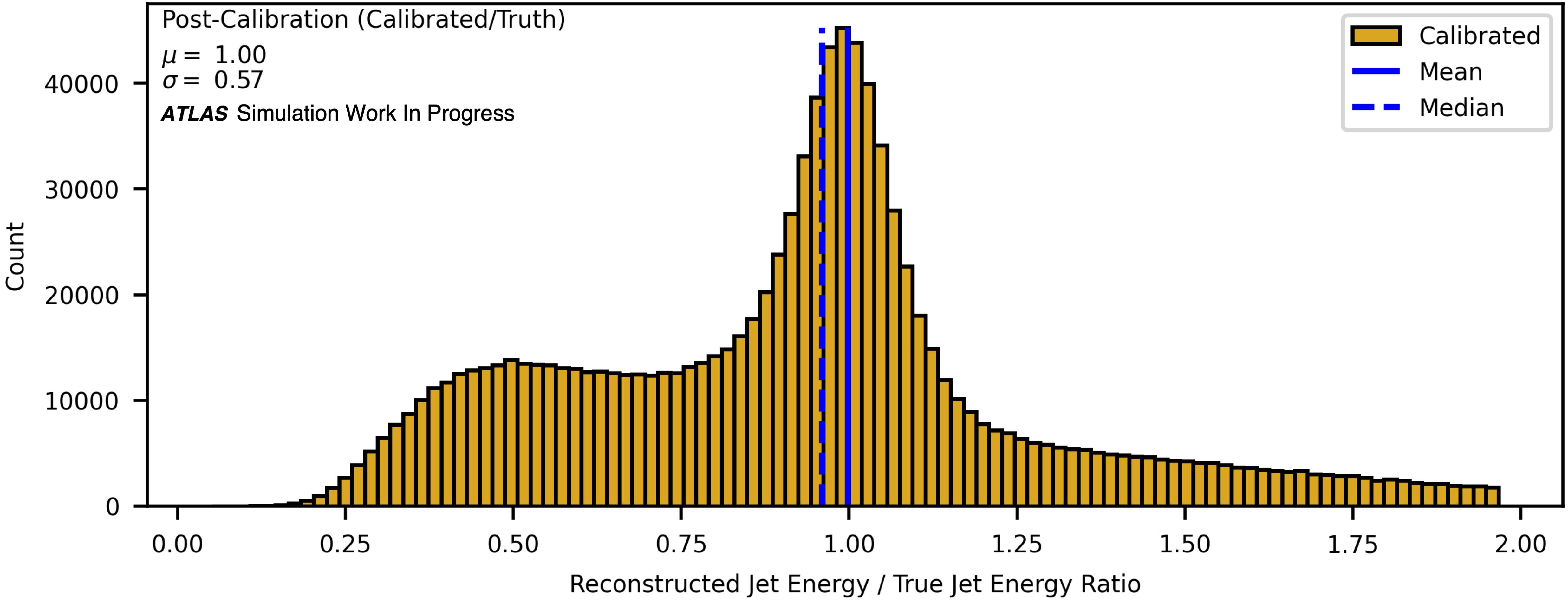
Current Results

Calibration Performance - Energy Ratios

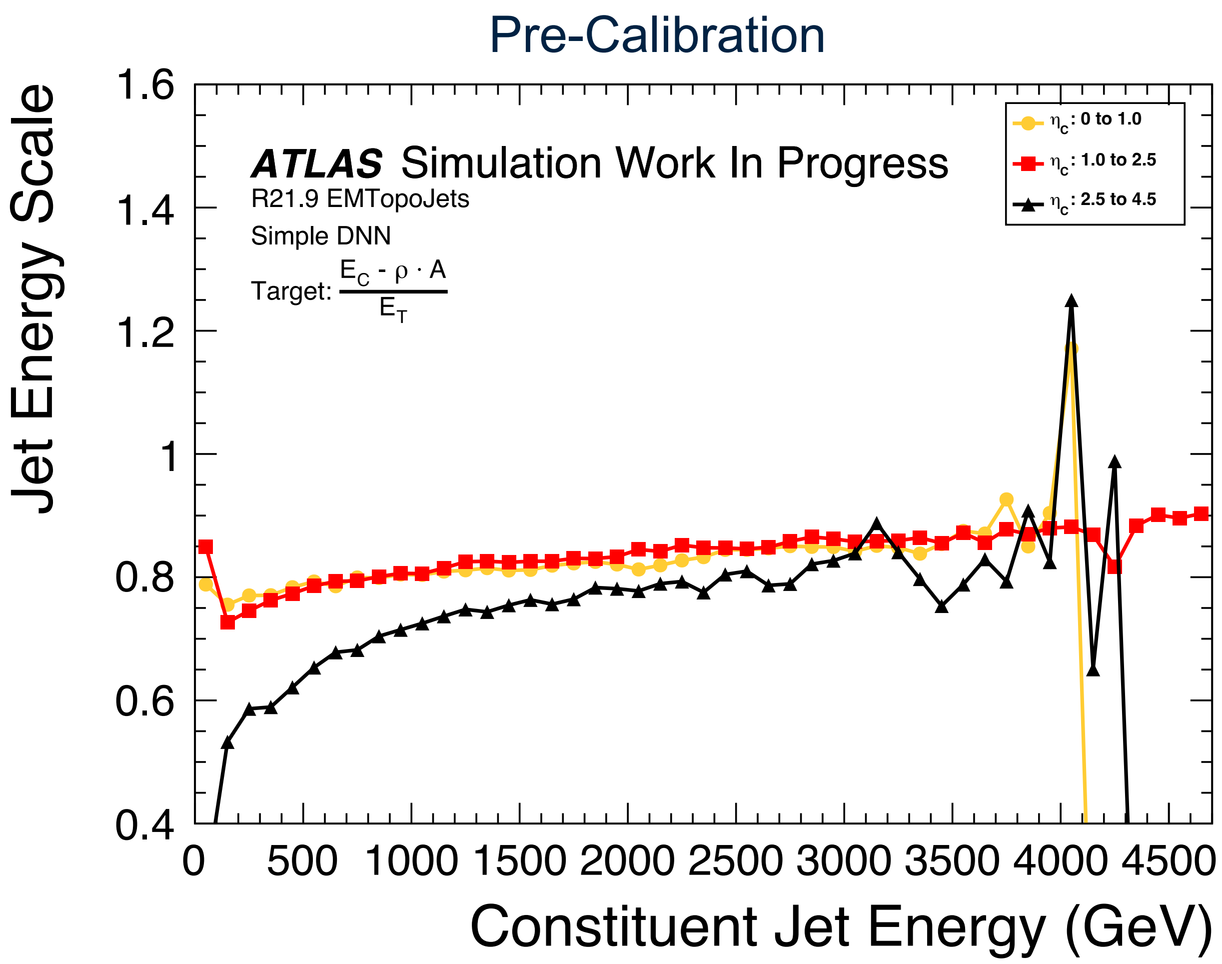


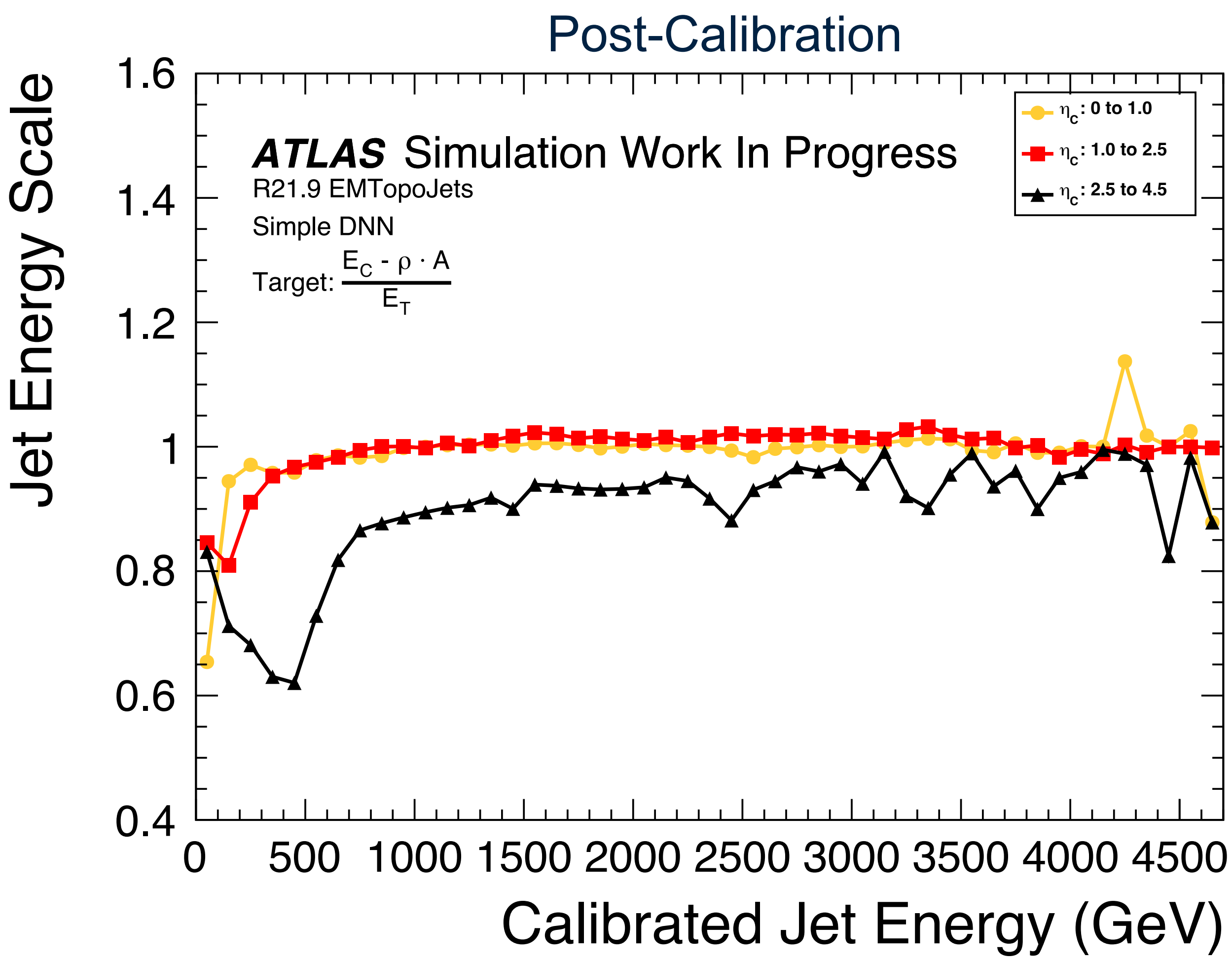
Calibration Performance - Energy Ratios

- Network can shift mean/median response and accomplish nominal task



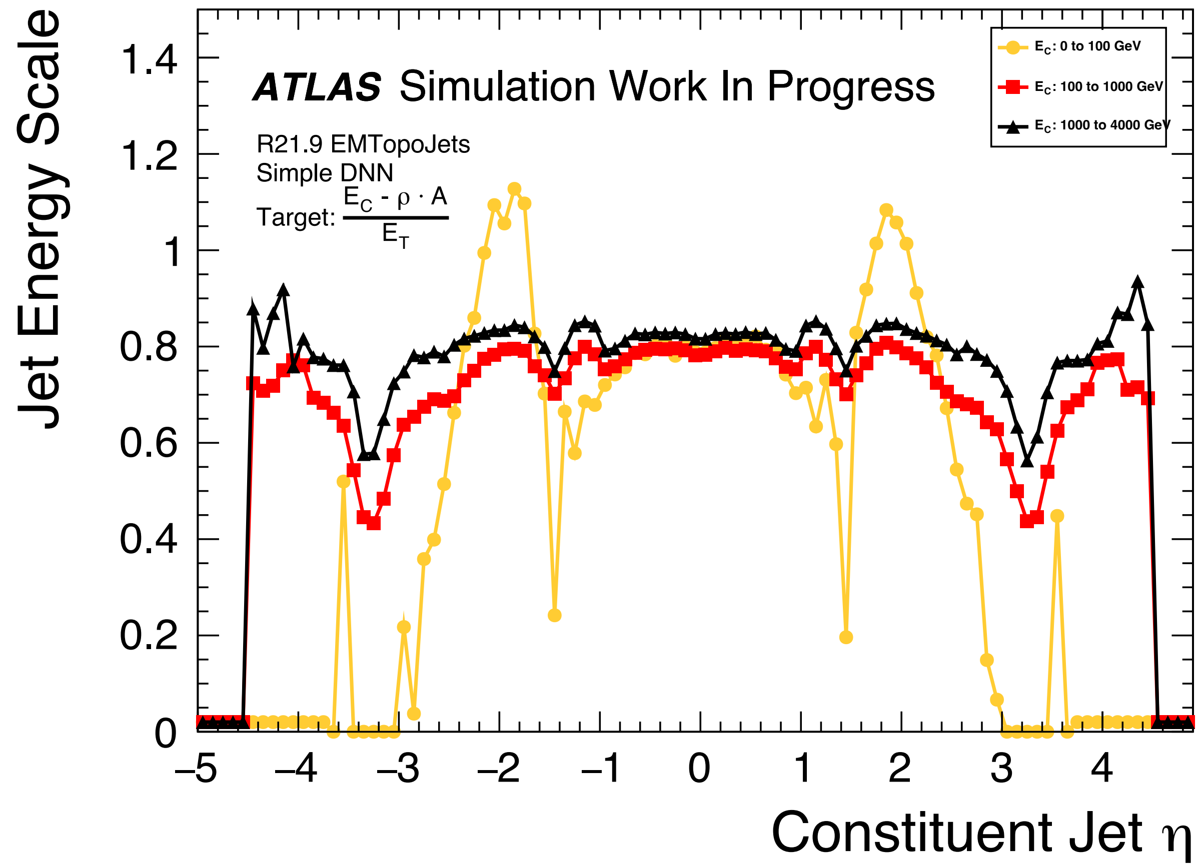
Jet Energy Scale - $E_{Constit}$ v. E_{Cal} (η Bins)



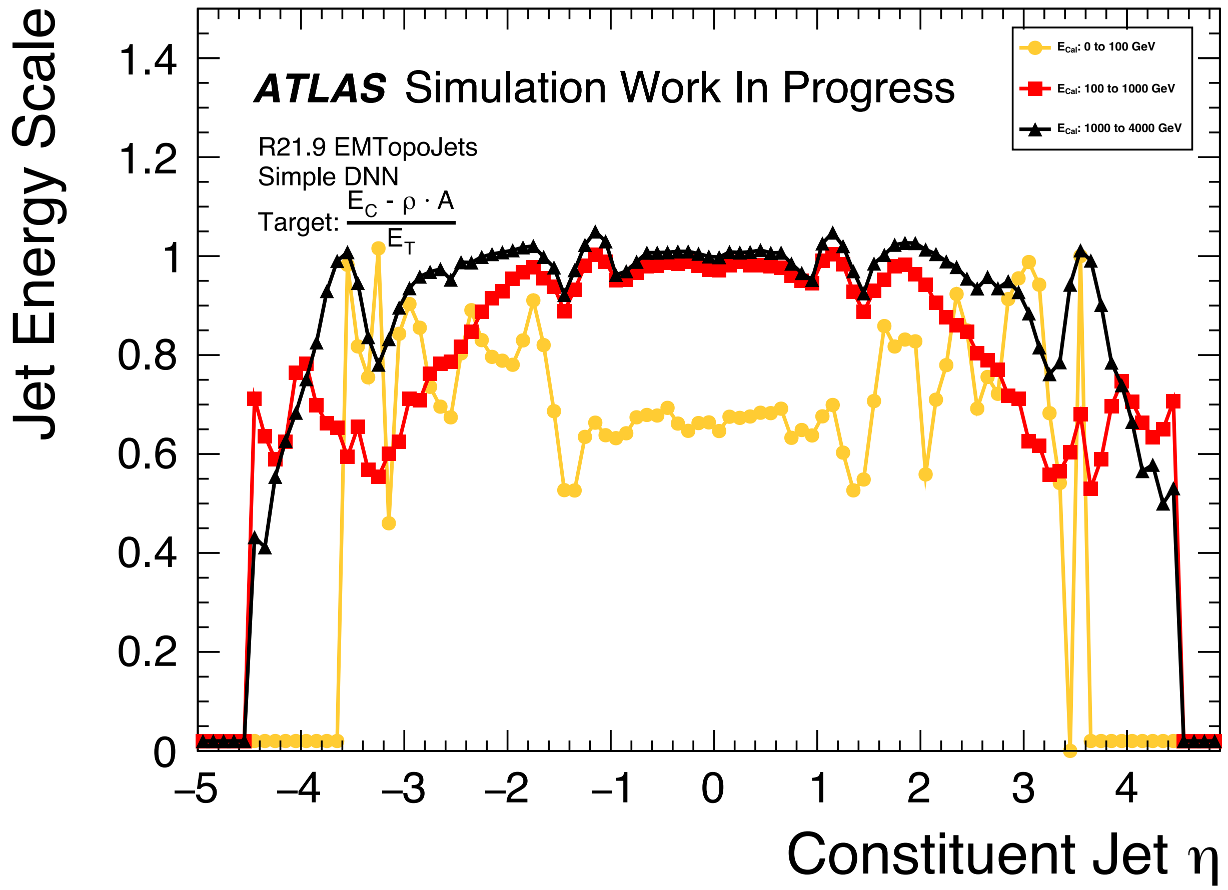


Jet Energy Scale - $E_{Constit}$ v. E_{Cal} in η (E Bins)

Pre-Calibration

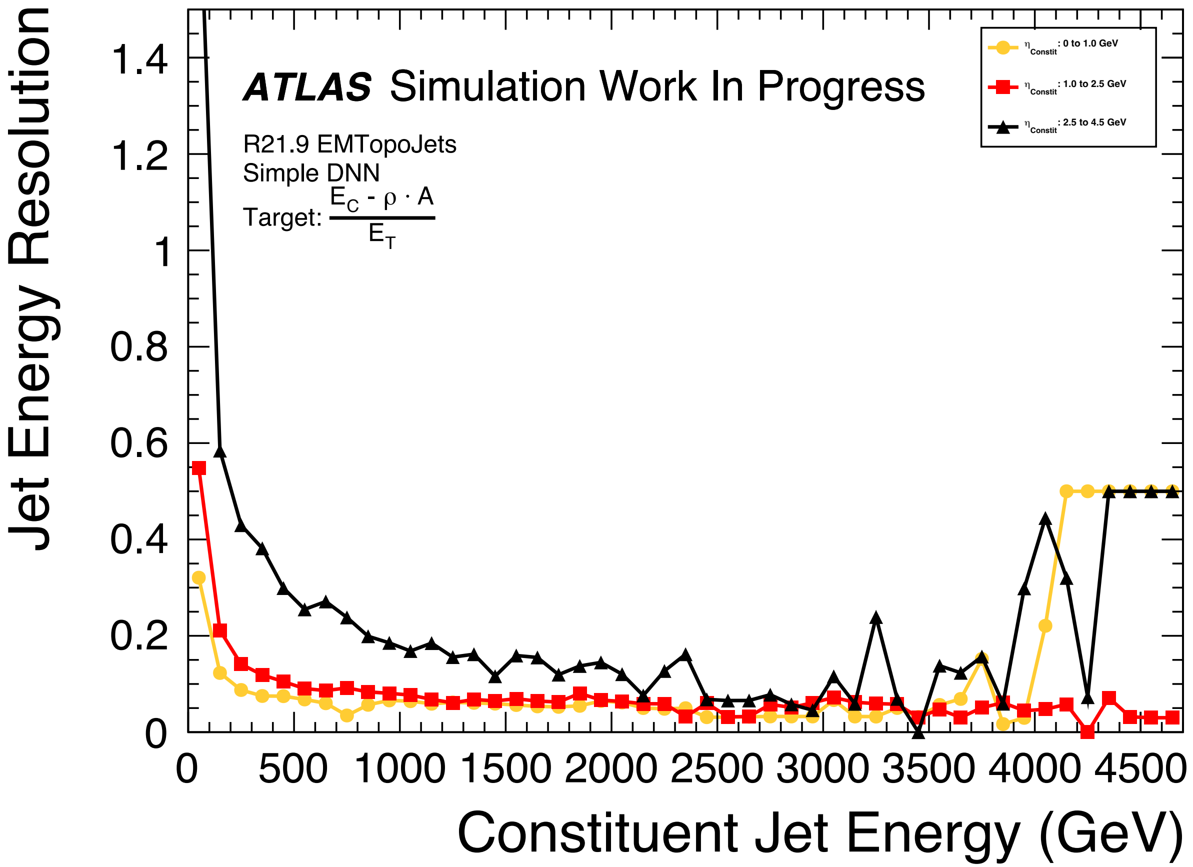


Post-Calibration

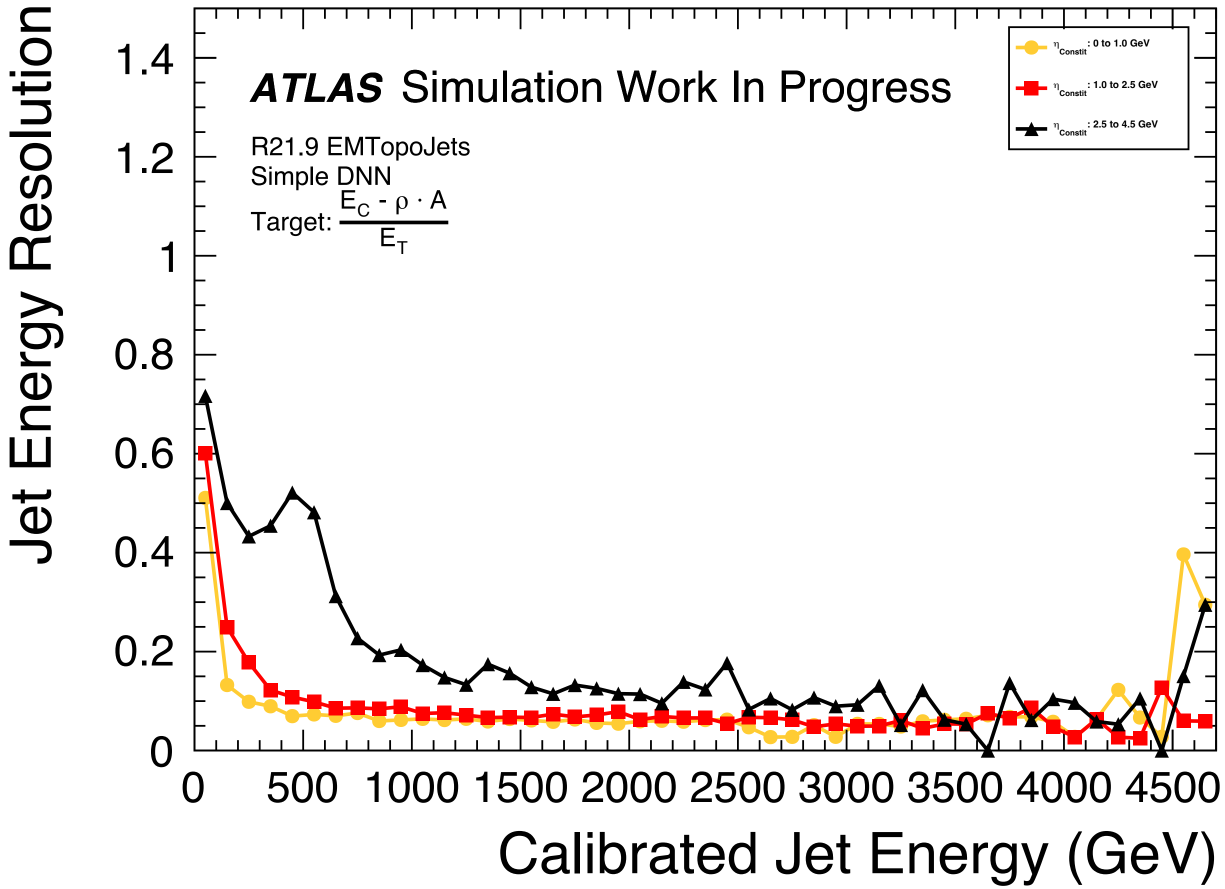


Jet Energy Resolution - $E_{Constit}$ v. E_{Cal} (η Bins)

Pre-Calibration



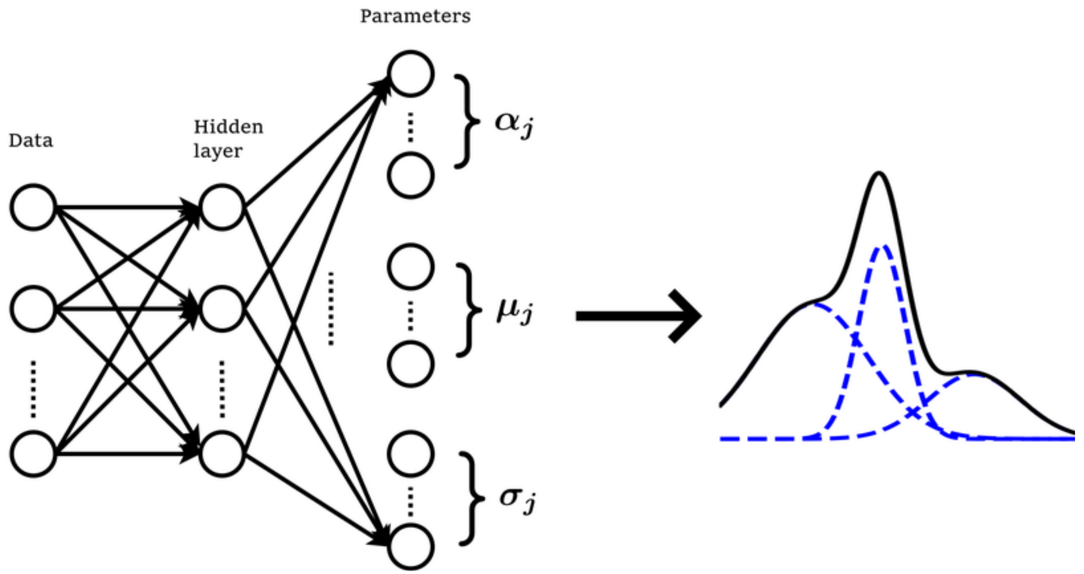
Post-Calibration



Future Studies and Considerations

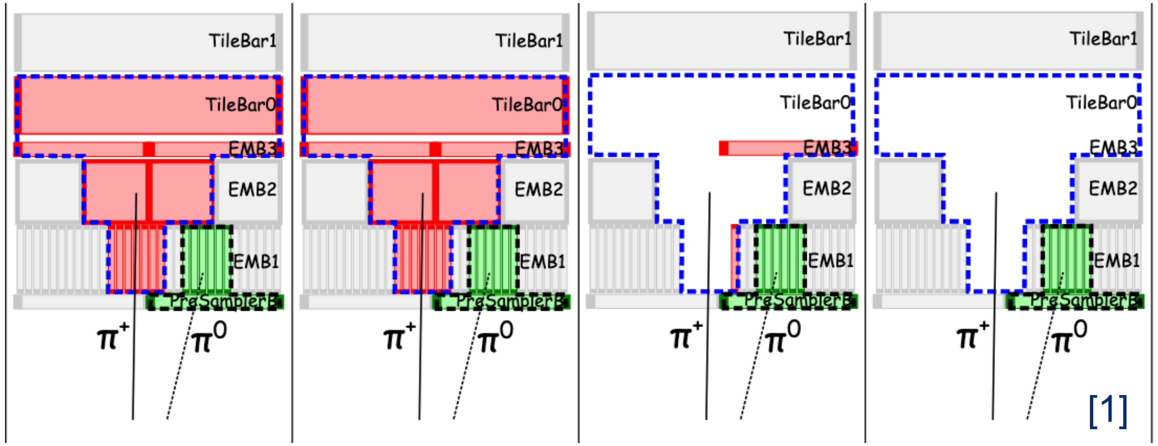
Loss Function Extensions

- Several groups exploring Mixture Density Networks (MDNs)[APS]
- Fit K Gaussians to input distribution
  - Return amplitude, μ, σ for each
- Effectively generate PDF for possible corrections



PFlow Jets

- Associate inner tracker information with EMTopo clusters forming each jet
- Improves low-p_T jet resolution, reconstruction efficiency
  - Combats pile-up instability



[1]: Jet reconstruction and performance using particle flow with the ATLAS Detector

Penn ATLAS EXPERIMENT

Questions?

Backup

Overview

- Overall: Develop ML-based MCJES calibration for upcoming R24 HL-LHC MC samples
  - Initially built on Run 3 framework developed by Kevin Greif in coordination with Chris Pollard & Jennifer Roloff [1]
- Develop/cross-check new ML calibration performance against existing 21.9 EMTopo jet calibration
  - Use same inputs & evaluate performance against Jingjing Pan's R21.9 EMTopo jet calibration (residual pileup + MCJES corrections)[2]
- Network output: set of calibrated weights which generate all-in-one scalar jet correction $R(X_{reco}, \theta)$

Current jet calibration

Stage I: Pileup Correction

$$p_{corr} = p_{reco} - \rho \times A - \alpha \times (N_{PV} - 1) - \beta \times \mu$$

Stage II: JES Correction

$$E_{corr} = \mathscr{R}(E_{reco}) * E_{reco} \approx \mathscr{R}\left(N\left(E^{reco}/E^{true}\right)\right) * E_{reco}$$

Stage II: GSC Correction

$$E_{corr} = \mathscr{R}(f_{charged}, f_{Tile0}, w_{trk}\ldots) * E_{reco}$$

ML-based calibration

Stage I: Train

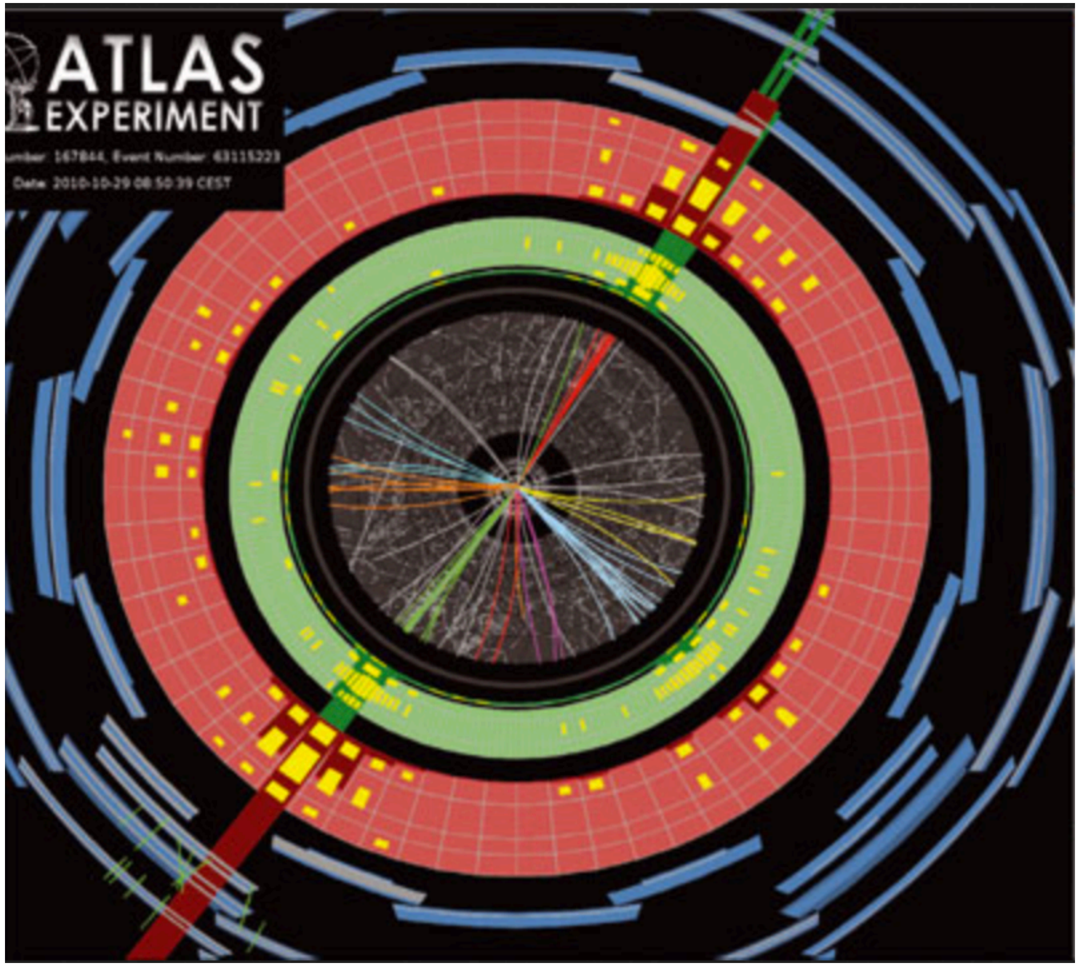
$$R(X_{reco}, \theta) = \left(\frac{X_{reco}}{X_{true}}\right) * f(\theta)$$

Stage II: Calibrate

$$X_{calib} = \frac{1}{R} X_{Reco} = \frac{X_{true}}{X_{reco}} X_{reco} \approx X_{true}$$

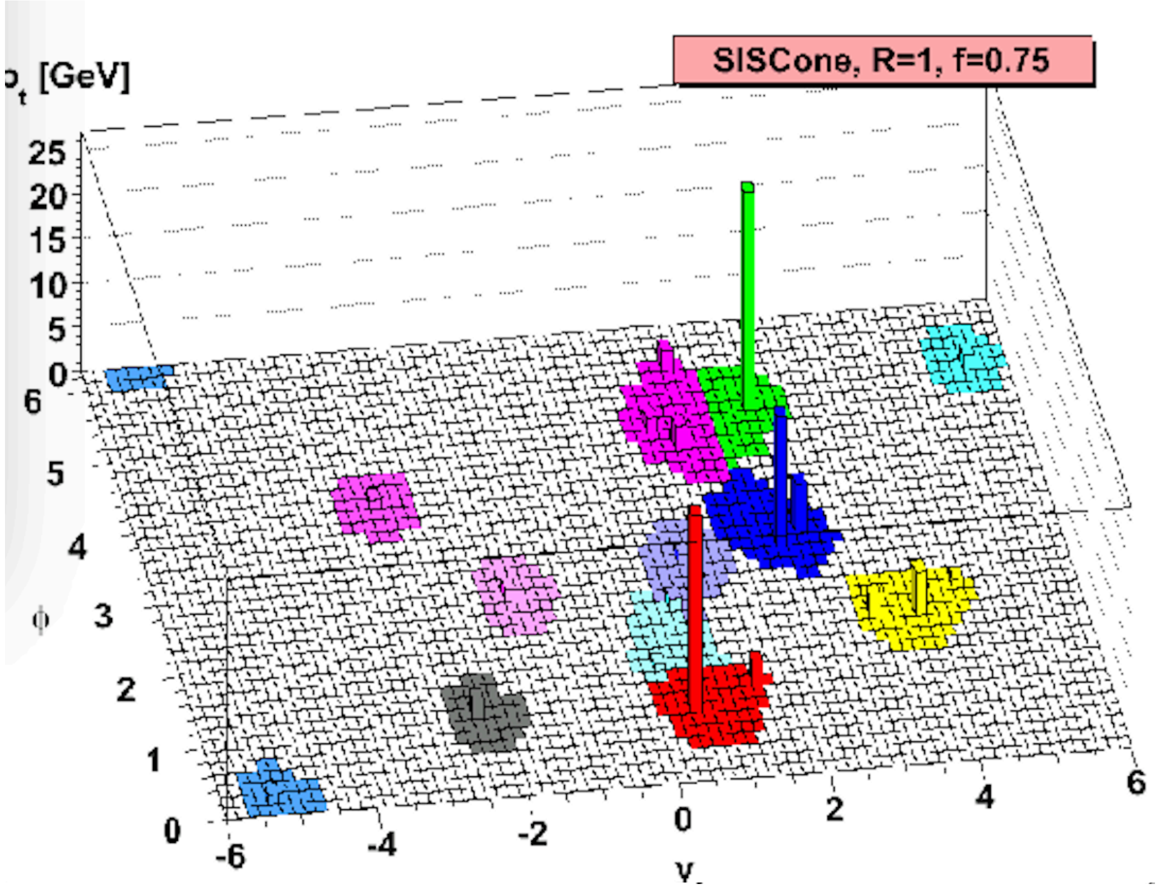
What is a jet?

- Jet = closest physics object to original parton
  - Offer multiplicity, p_T, and substructure signatures
- Defined by **parameter(s)** and **recombination scheme**
- Must nominally meet Snowmass Conditions
  - "Simple" to use in theory/experiment
  - Yields finite, hadronization-insensitive σ
- Definition choice heavily dependent on use-case
  - "No single optimal way of defining jets"
- Upcoming R3/HL-LHC demand high performance across various aspects
  - Energy resolution, pileup correction, readout time…
- "…no single jet definition will work optimally for the whole range of LHC phenomena"



Sequential Recombination (Anti$-k_t$)

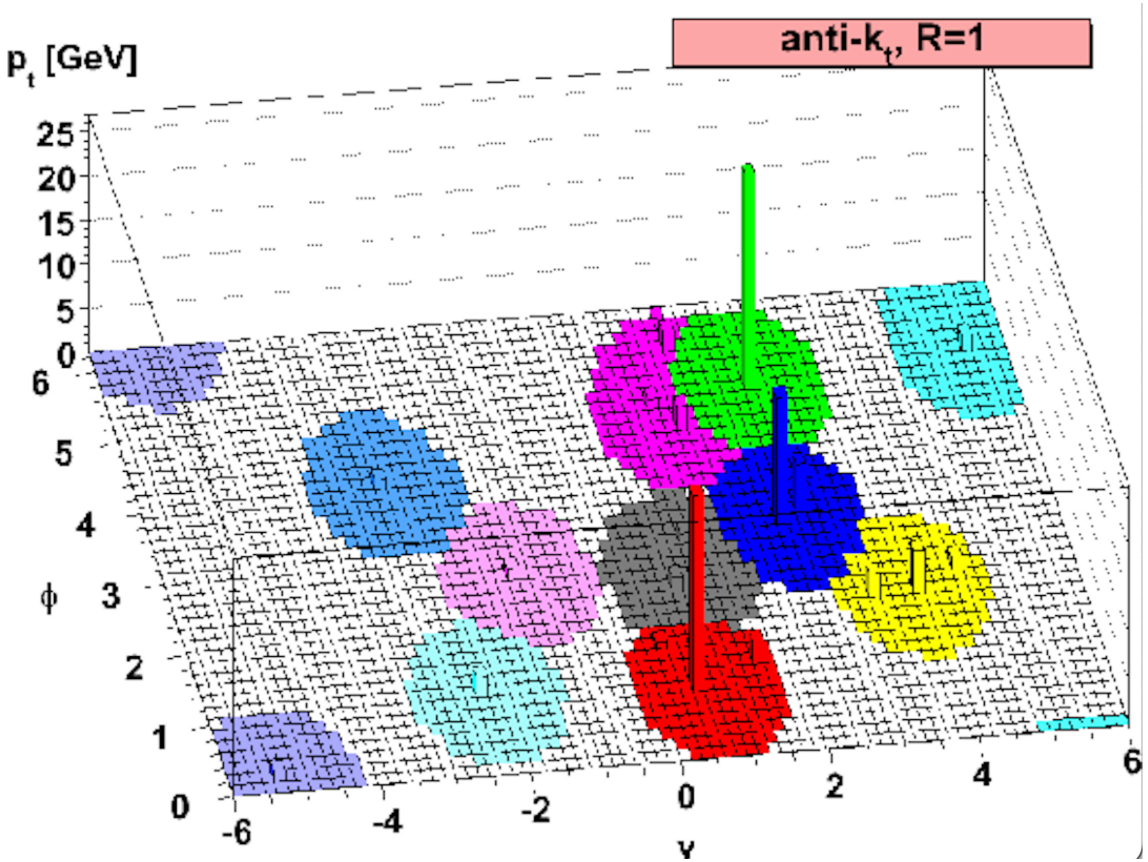
- Bottom-up jet construction
  - Build jets on shared metric, not from singular seed
  - Assign clustering sequence to jet substructure
- For set of particles $\{n\}$:
  - Find all distance measures d_{ij}
  - Locate pair $\{i, j\}$ corresponding to min$\{d_{ij}\}$
  - IF($d_{iB} = d_{min}$): Declare i final-state jet and repeat
  - ELIF($d_{min} > d_{cut}$): merge $\{i, j\}$ into single protojet
  - If particles remain: repeat procedure
  - ELSE: Assign all remaining objects to be jets and terminate
- Jets built out around harder seeds
- Fully inclusive, relatively fast [$\mathcal{O}(N\sqrt{n})$], and **IRC-safe**



$$d_{ij} = \min\left[p_{T,i}^{-2}, p_{T,j}^{-2}\right]\frac{\Delta R_{ij}^2}{R^2} \qquad d_{iB} = p_{T,i}^{-2}$$

Sequential Recombination (Anti$-k_t$)

- Bottom-up jet construction
  - Build jets on shared metric, not from singular seed
  - Assign clustering sequence to jet substructure
- For set of particles $\{n\}$:
  - Find all distance measures d_{ij}
  - Locate pair $\{i, j\}$ corresponding to min$\{d_{ij}\}$
  - IF($d_{iB} = d_{min}$): Declare i final-state jet and repeat
  - ELIF($d_{min} > d_{cut}$): merge $\{i, j\}$ into single protojet
  - If particles remain: repeat procedure
  - ELSE: Assign all remaining objects to be jets and terminate
- Jets built out around harder seeds
- Fully inclusive, relatively fast [$\mathcal{O}(N\sqrt{n})$], and **IRC-safe**

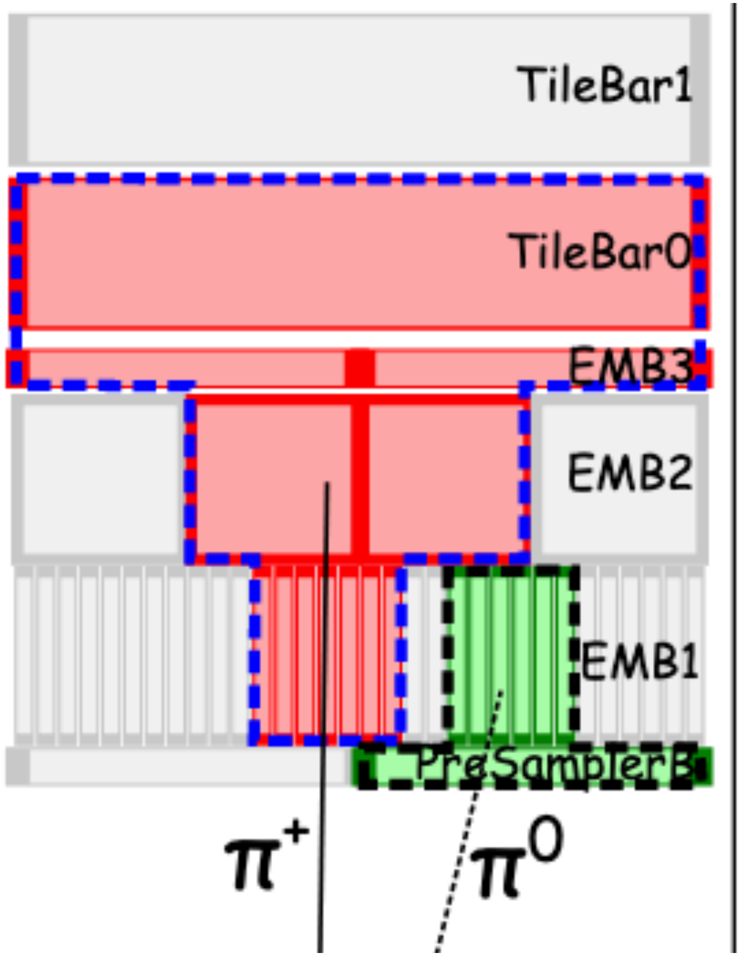


$$d_{ij} = \min\left[p_{T,i}^{-2}, p_{T,j}^{-2}\right]\frac{\Delta R_{ij}^2}{R^2} \qquad d_{iB} = p_{T,i}^{-2}$$

Penn ATLAS EXPERIMENT

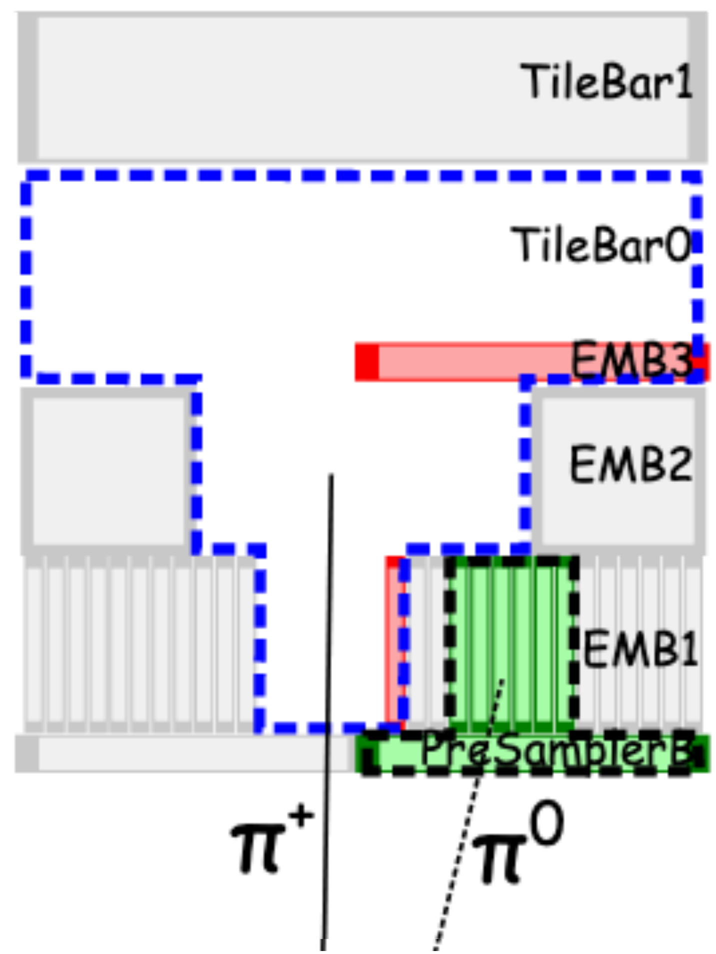
ATLAS Jet Inputs

- Through Run 2: **EMTopo** Jets
  - Massless clustering of calorimeter cells (topo-clusters)
  - Cut on deposited energy/noise ratio with vertex correction
- Moving forward: **Particle Flow** (PFlow) Jets
  - Combine calorimeter towers with tracking data
    - Link EMTopo cluster to low-p_T tracks
    - Remove EMTopo energy/replace with particle p_T
    - Leave remnant EMTopo clusters + hard tracks
  - Better resolution (E, ϕ), pileup stability, reco. Efficiency
    - Better captures low-p_T regime (< 40 GeV)
- Jet inputs passed to anti$-k_t$ algorithm with $R = 0.4$ $(1.0$ for fat jets)



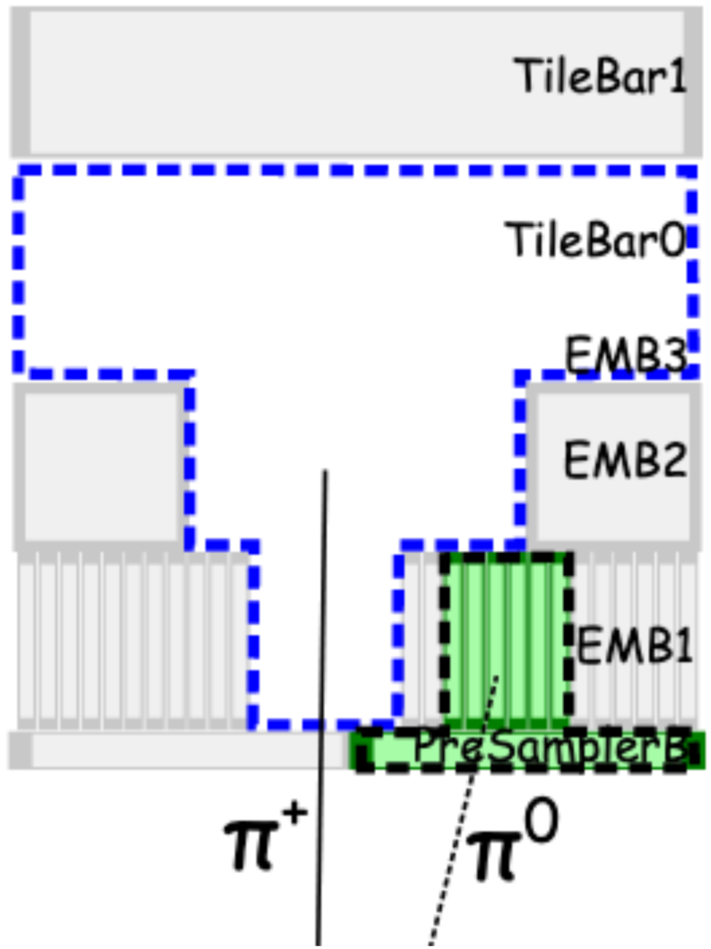
ATLAS Jet Inputs

- Through Run 2: **EMTopo** Jets
  - Massless clustering of calorimeter cells (topo-clusters)
  - Cut on deposited energy/noise ratio with vertex correction
- Moving forward: **Particle Flow** (PFlow) Jets
  - Combine calorimeter towers with tracking data
    - Link EMTopo cluster to low-p_T tracks
    - Remove EMTopo energy/replace with particle p_T
    - Leave remnant EMTopo clusters + hard tracks
  - Better resolution (E, ϕ), pileup stability, reco. Efficiency
    - Better captures low-p_T regime (< 40 GeV)
- Jet inputs passed to anti$-k_t$ algorithm with $R = 0.4$ $(1.0$ for fat jets)



ATLAS Jet Inputs

- Through Run 2: **EMTopo** Jets
  - Massless clustering of calorimeter cells (topo-clusters)
  - Cut on deposited energy/noise ratio with vertex correction
- Moving forward: **Particle Flow** (PFlow) Jets
  - Combine calorimeter towers with tracking data
    - Link EMTopo cluster to low-p_T tracks
    - Remove EMTopo energy/replace with particle p_T
    - Leave remnant EMTopo clusters + hard tracks
  - Better resolution (E, ϕ), pileup stability, reco. Efficiency
    - Better captures low-p_T regime (< 40 GeV)
- Jet inputs passed to anti$-k_t$ algorithm with $R = 0.4$ $(1.0$ for fat jets)

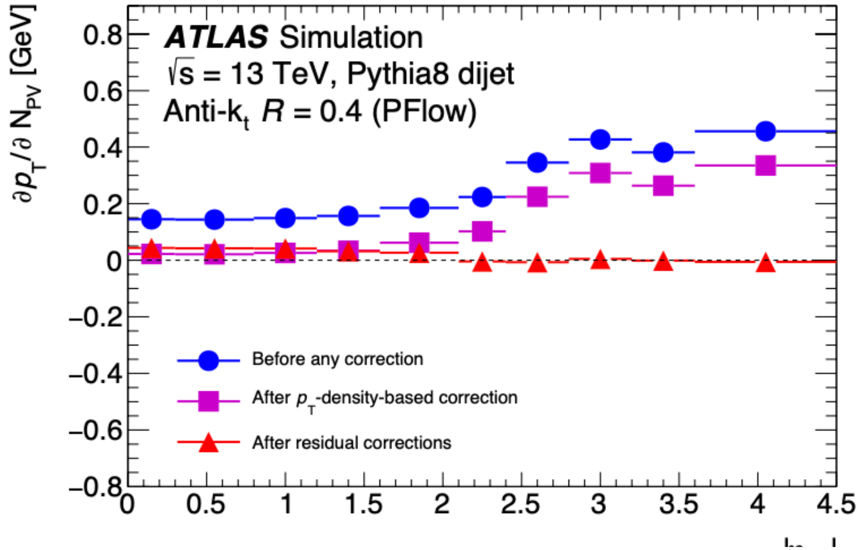


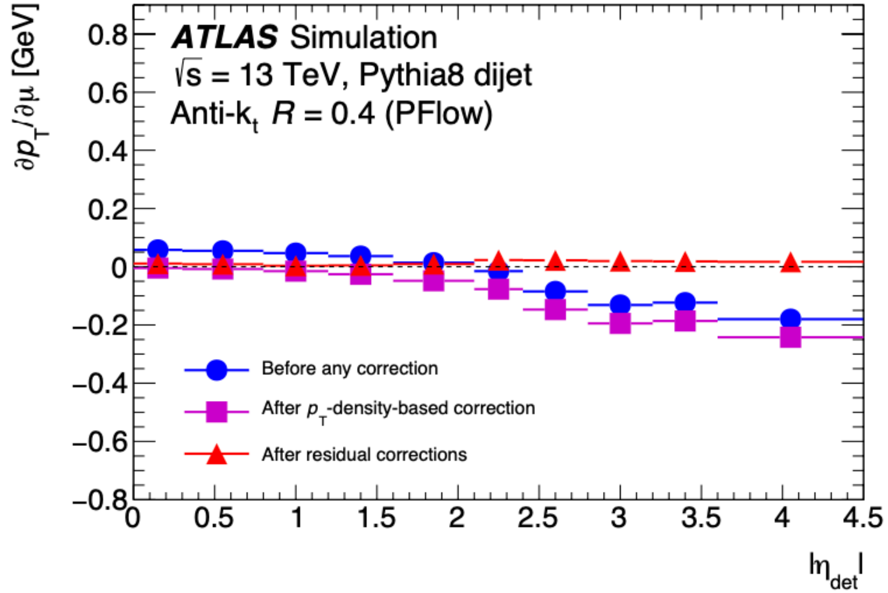
ATLAS Jet Inputs

- Through Run 2: **EMTopo** Jets
  - Massless clustering of calorimeter cells (topo-clusters)
  - Cut on deposited energy/noise ratio with vertex correction
- Moving forward: **Particle Flow** (PFlow) Jets
  - Combine calorimeter towers with tracking data
    - Link EMTopo cluster to low-p_T tracks
    - Remove EMTopo energy/replace with particle p_T
    - Leave remnant EMTopo clusters + hard tracks
  - Better resolution (E, ϕ), pileup stability, reco. Efficiency
    - Better captures low-p_T regime (< 40 GeV)
- Jet inputs passed to anti$-k_t$ algorithm with $R = 0.4$ $(1.0$ for fat jets)

Residual Pileup Correction

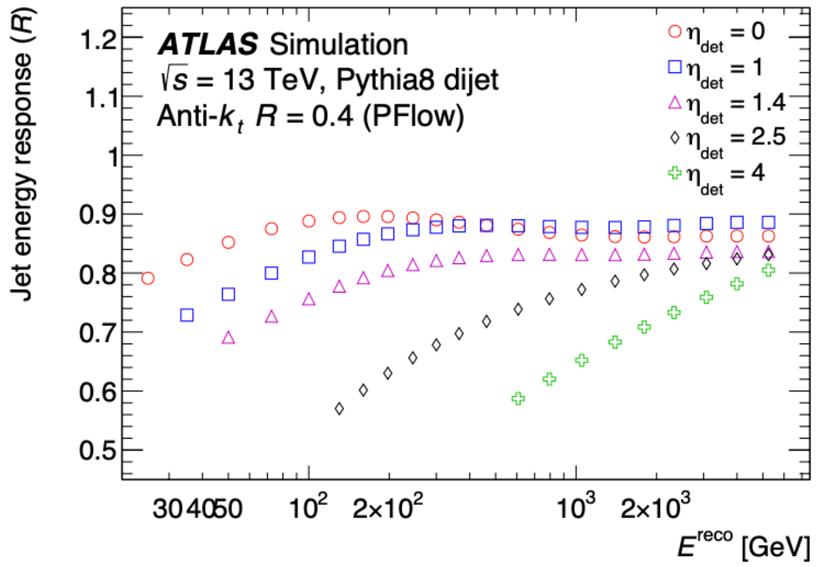
- First (central, low-occupancy) correction: reduce added p_T due to pileup using k_T-based density measure
  - Determine **passive** jet area a using “ghosts”
  - Calculate p_T density $\rho = \langle \frac{p_T}{A} \rangle$ in $y - \phi$ with $|\eta| < 2$
    - Best measure of soft pileup background
  - Scale jet $(\vec{E}, \vec{p})$ by ρ-subtracted p_T to original p_T ratio
- Second (forward, high-occupancy) correction: match $p_{T,reco}$ to $p_{T,truth}$
  - Function of N_{PV} and μ
- Final correction given by $p_{corr} = p_{reco} - \rho \times A - \alpha \times (N_{PV} - 1) - \beta \times \mu$
  - Fit in bins of $|\eta_{det}|$

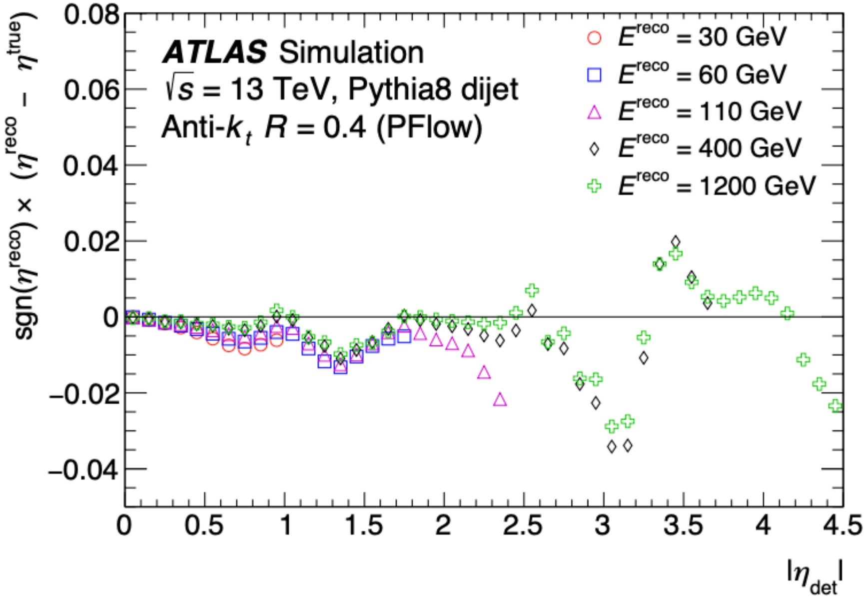




MCJES/η Correction

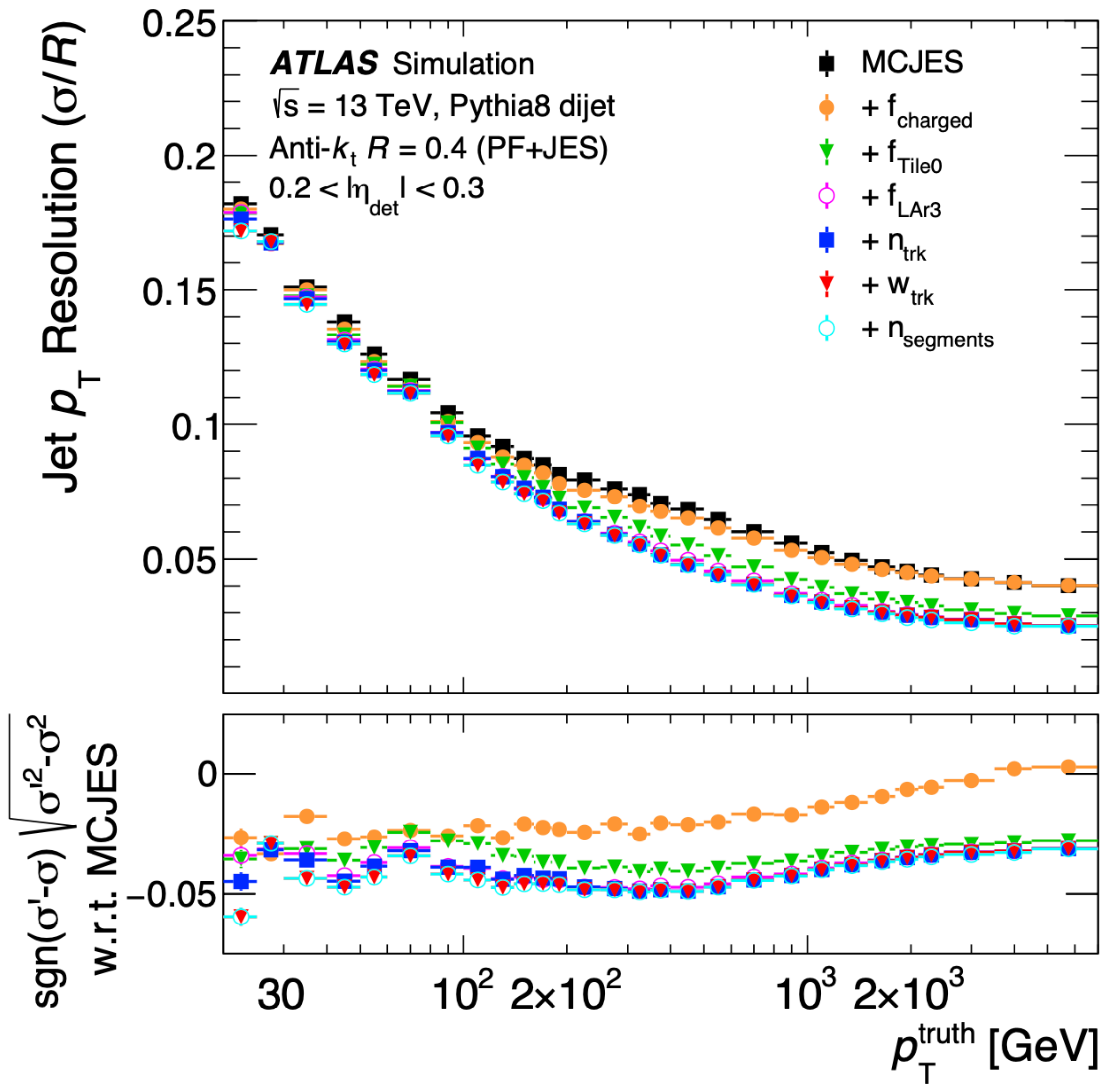
- Jet Energy Scale (JES) accounts energy loss within the detector
  - Match truth jets to isolated reco. Jets within $\Delta R = 0.3$
  - Define jet energy response $\mathscr{R}$ as mean of $N\left(E^{reco}/E^{true}\right)$
  - Numerically invert distribution to find $\mathscr{R}(E^{reco})$
  - Scale jet four-momentum accordingly
- η correction accounts for calorimeter edges/energy responses
  - Similar methodology
  - Only alters $\vec{p}$ and η measurements, not four-vector





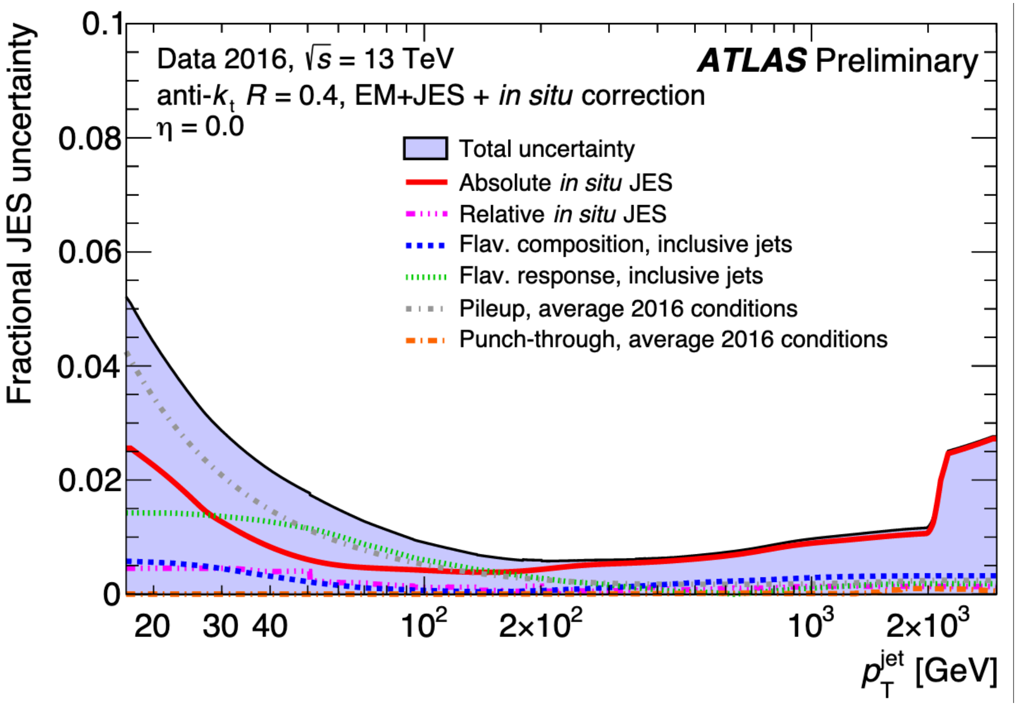
Global Sequential Calibration

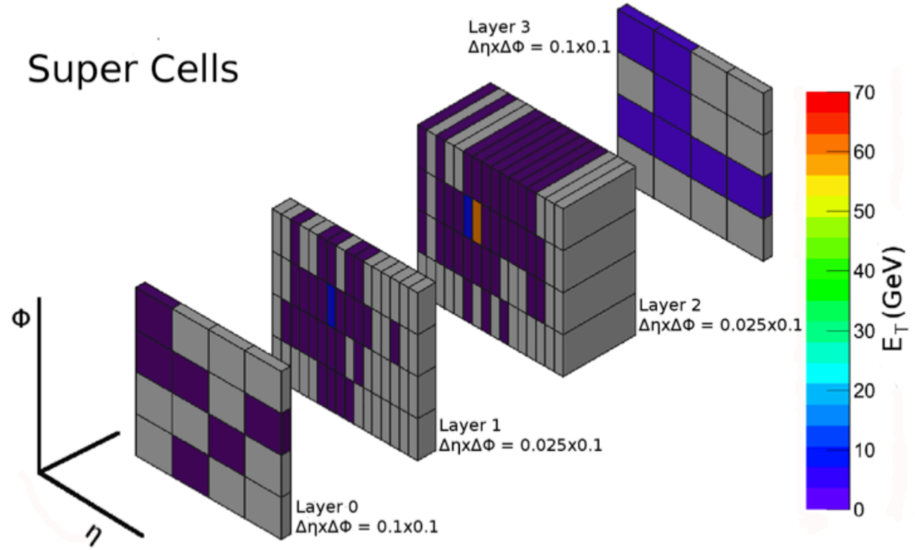
- Accounts for remaining jet physics which bias detector response
  - Quark vs. gluon jets: hard hadron signals vs. soft, transverse profile
  - Quark flavor/energy distribution bias reconstruction as well
  - Goal: improve jet resolution [$\sigma_{\mathscr{R}} \leftarrow N(p_T^{reco}/p_T^{true})$] while maintaining JER
- Six independent scaling parameters derived for:
  - $f_{charged} = \{p_T > 500M, |\eta_{det}| < 2.5\}$
  - $f_{Tile0} = \{$first tile layer, $|\eta_{det}| < 1.7\}$
  - $f_{LAr3} = \{$third LAr layer, $|\eta_{det}| < 3.5\}$
  - n_{trk} = # of associated 1-GeV tracks
  - w_{trk} = average transverse distance between jet axis and all associated 1-GeV tracks
  - n_{seg} = # of associated muon track segments
- Derivation follows MCJES inversion-based procedure



Looking Forward: HL-LHC

- Main challenge: pileup up to $<\mu> = 200$
  - Dominant systematic for low-p_T (< 40 GeV) jets
  - Few studies on anticipated HL-LHC jet resolution
- Understanding upgraded detector effects
  - Improved calorimeter resolution
    - More localized energy deposits = better EMTopo clusters
  - Improved forward region tracking
  - Improved timing w/ HGTD
  - 1 MHz triggering
- Overall: need to simulate and understand jet performance under HL-LHC conditions

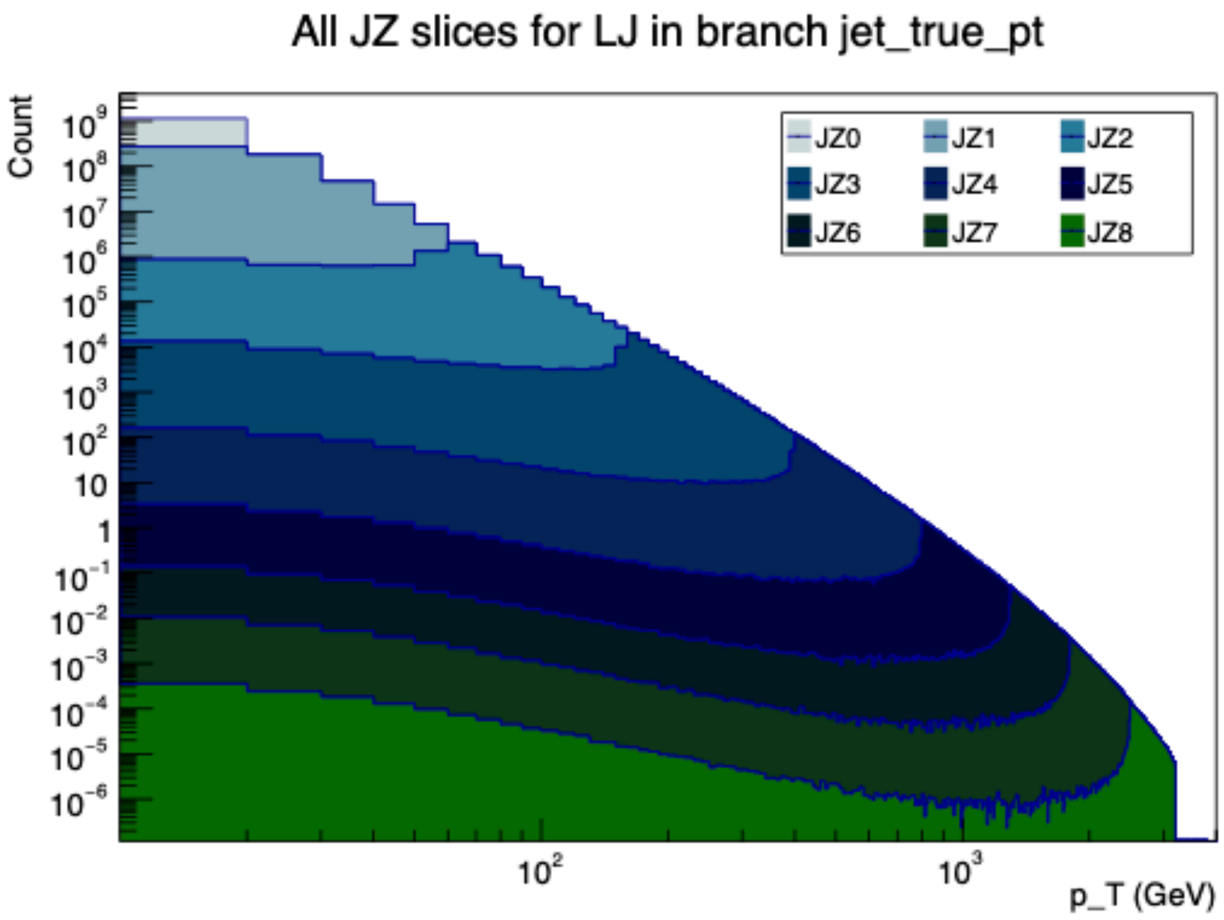


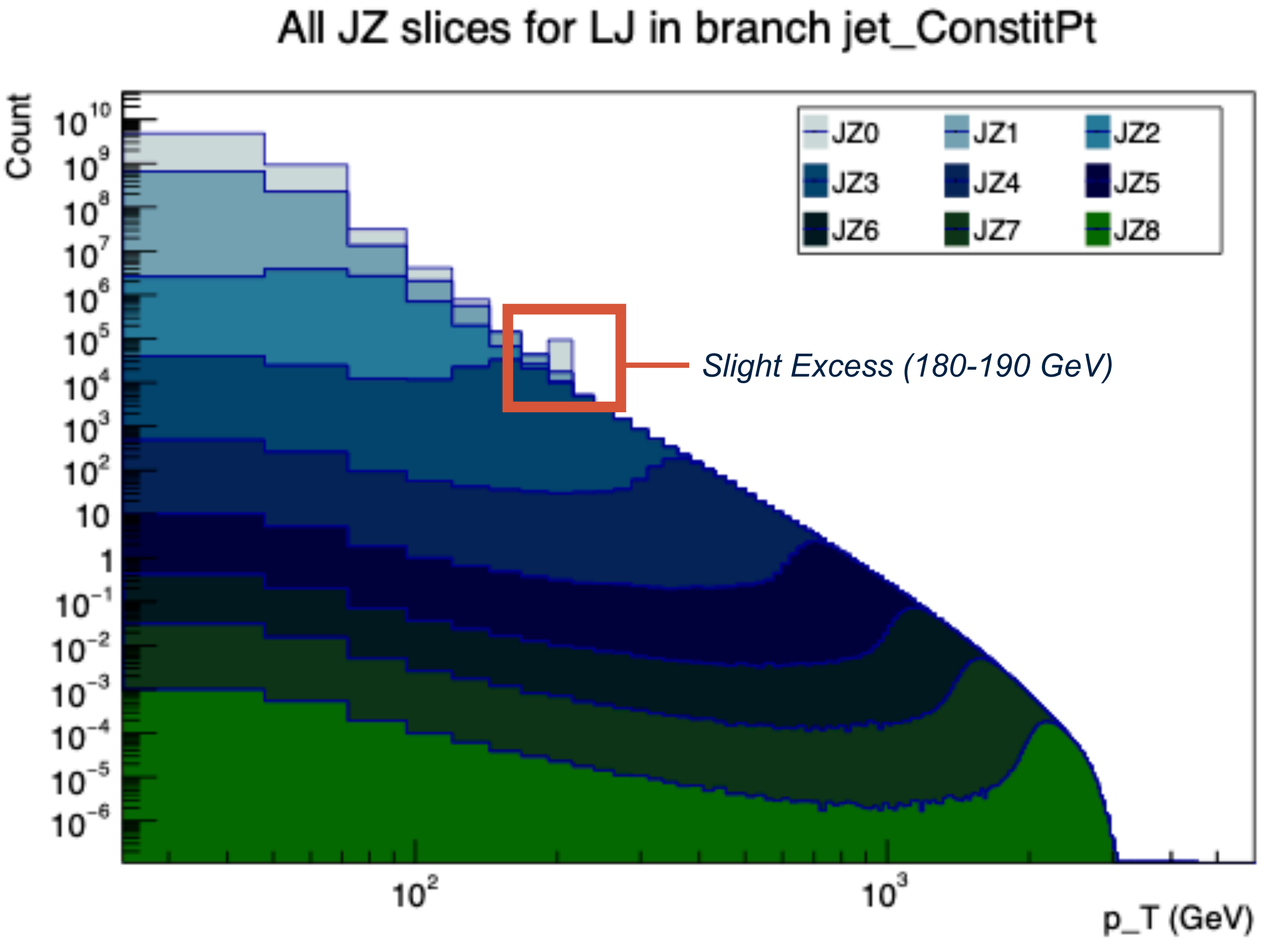


Current Dijet Samples

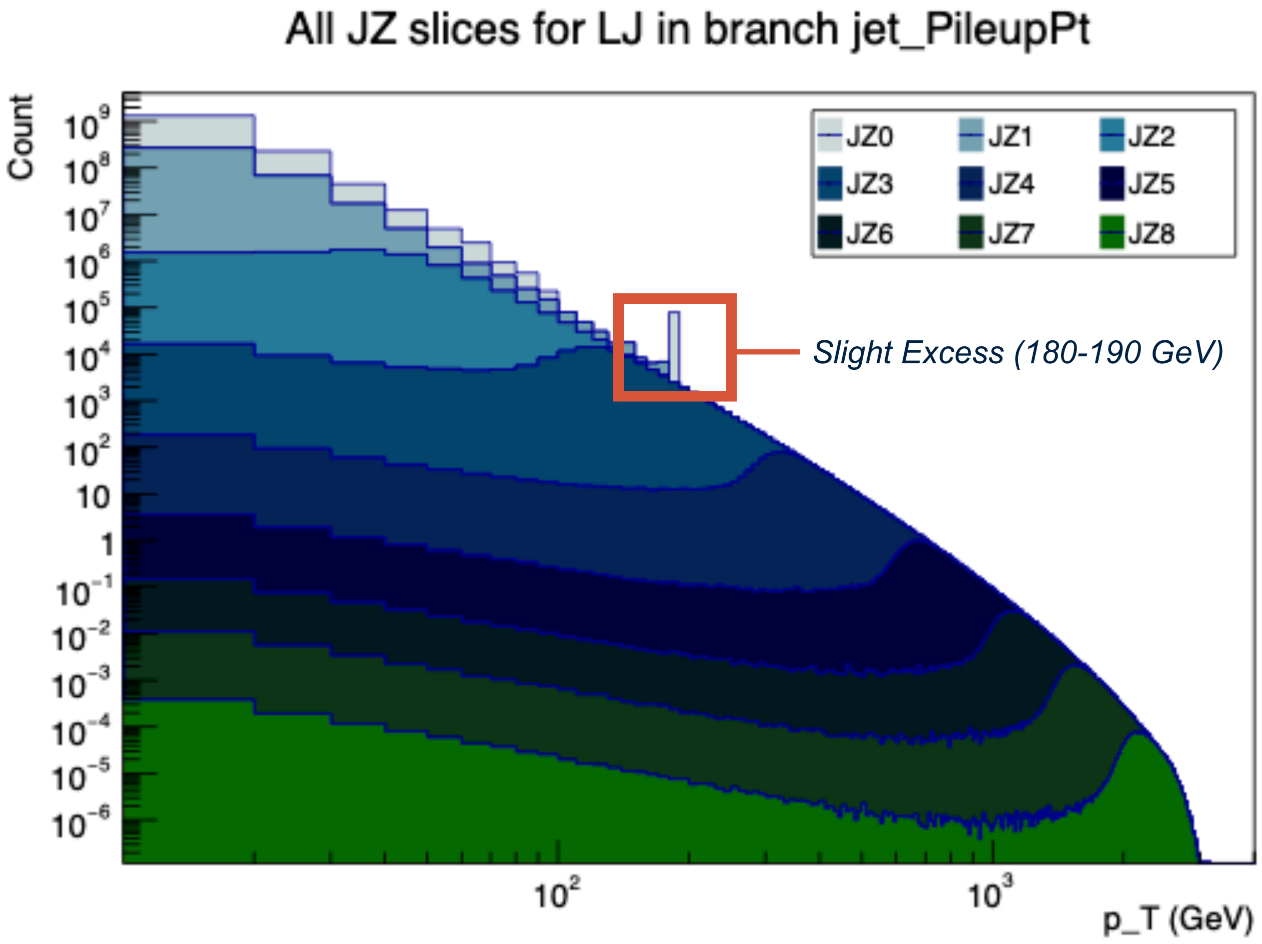
- R21.9
  - mc15_14TeV.800292.Py8EG_A14NNPDF23LO_jetjet_JZ2**WithSW**.recon.AOD.e8185_s3770_s3773_r13619
- R23
  - mc21_14TeV.801165.Py8EG_A14NNPDF23LO_jj_JZ0.deriv.DAOD_PHYSVAL.e8481_s4038_r14362_p5608

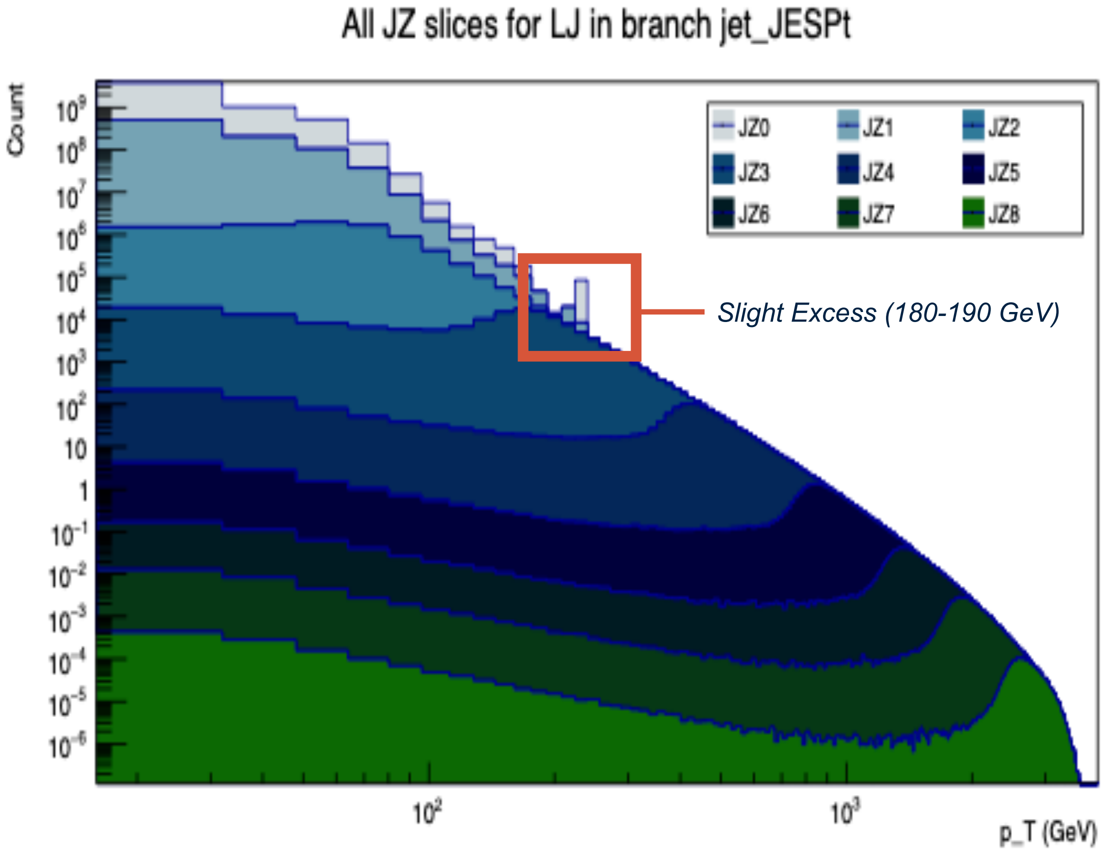
Leading Jet p_T Distributions - Truth + Constit





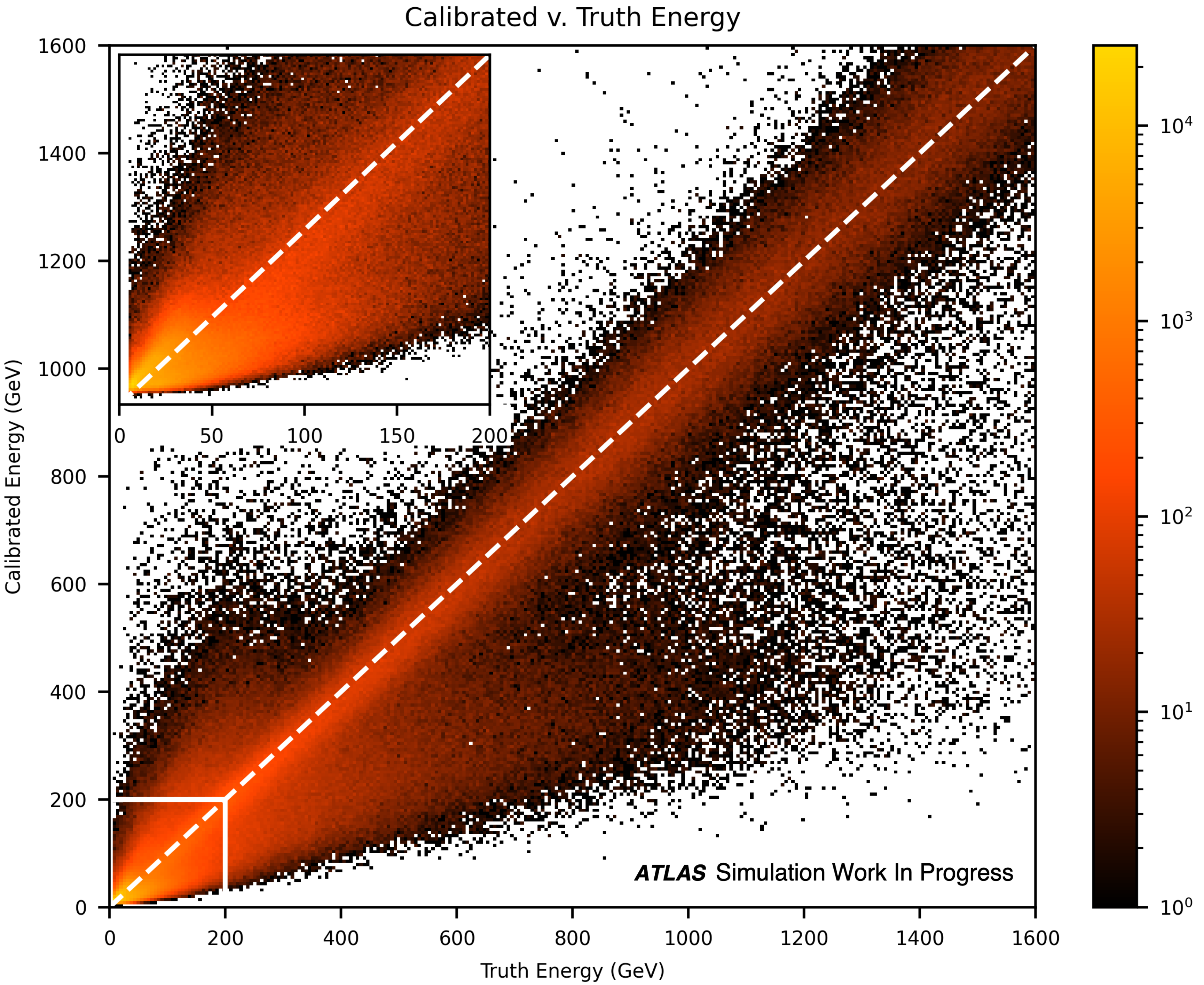
Leading Jet p_T Distributions - Pileup + JES





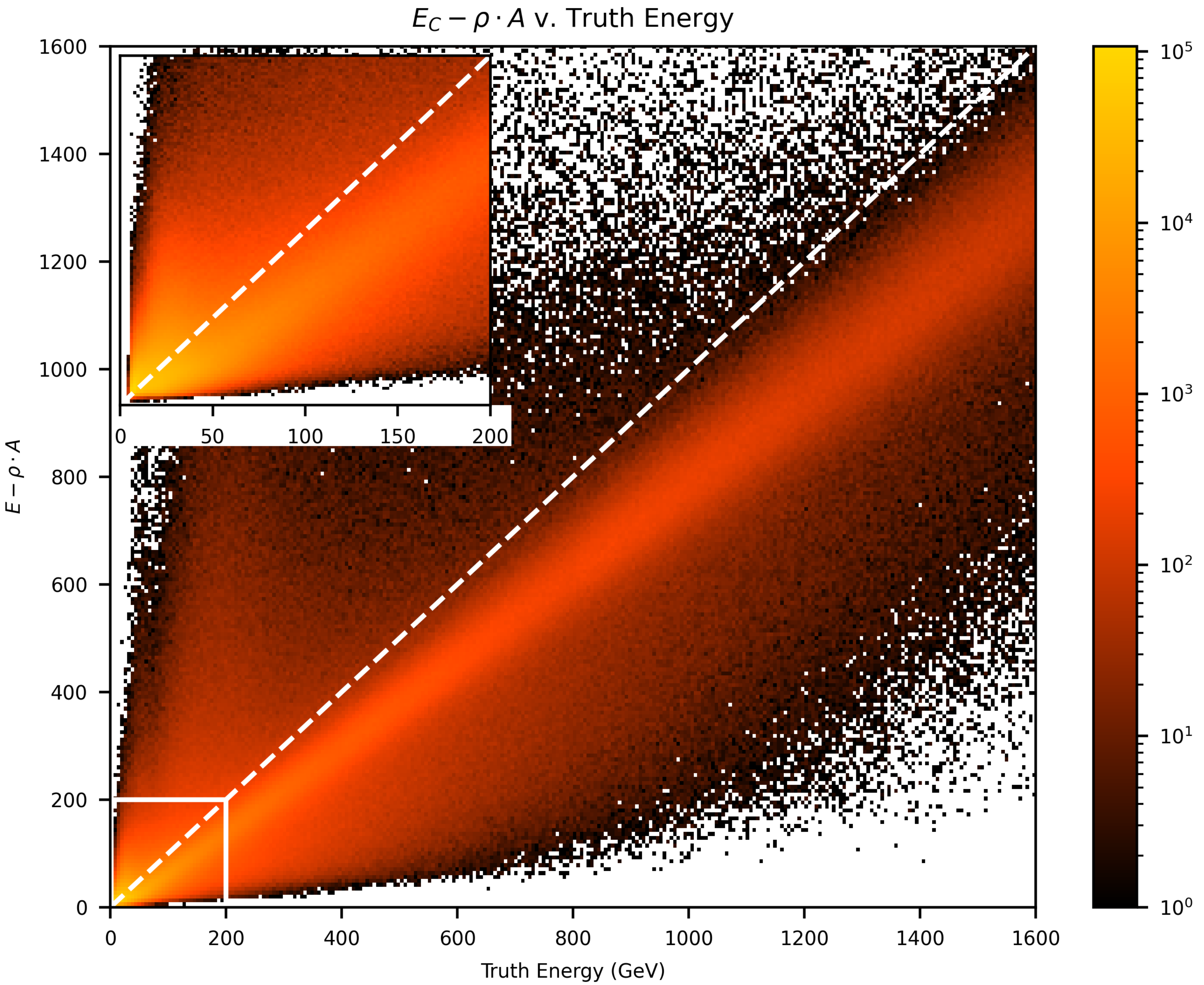
Calibrated v. True Energy

- ~100 GeV spread in calibrated energy for given truth value
- Exacerbated in pileup region



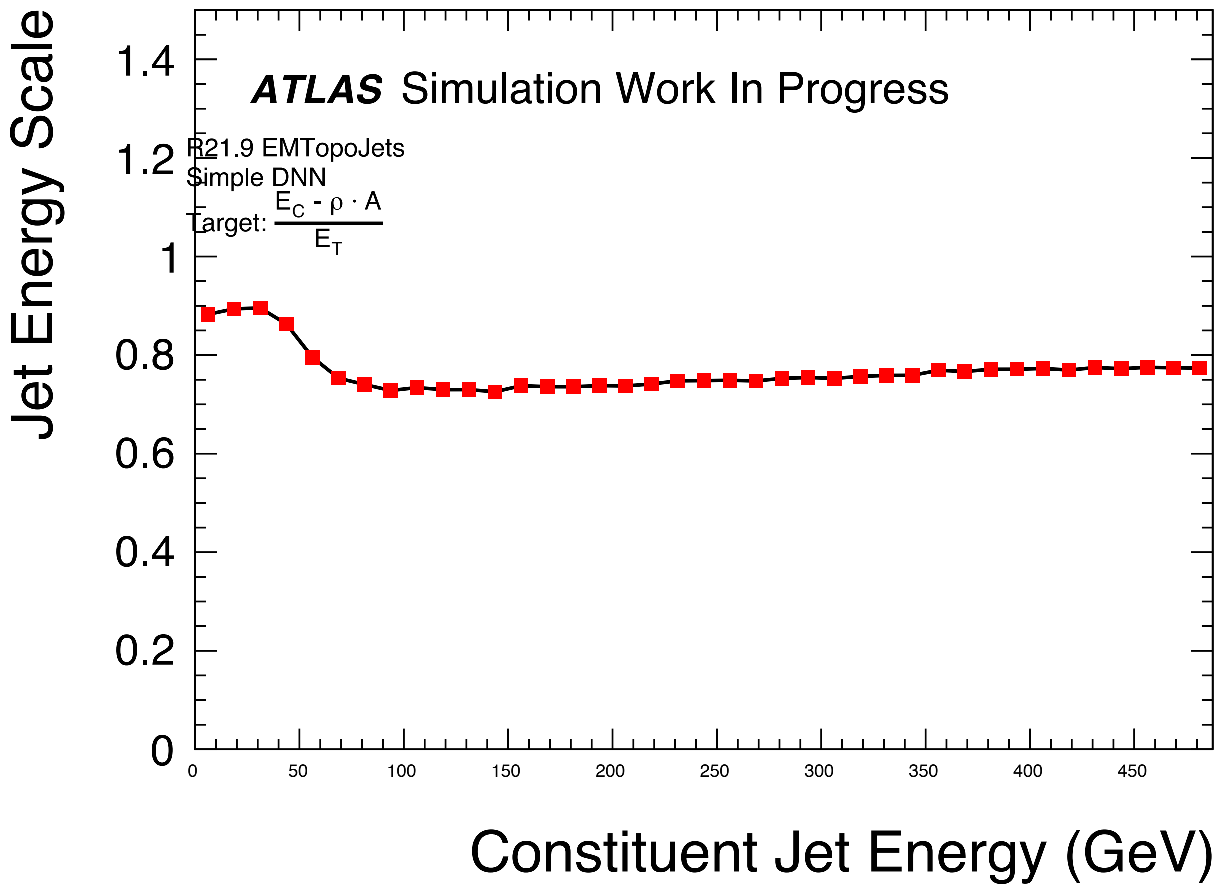
$E - \rho \cdot A$ Distribution

- Confirm similar low-E distribution within 21.9 samples
  - Not a network feature
- Network unable to pull out constituent calibration when inherent to distribution
- Reinforces this is a sample issue, needs to be addressed for further attempts at ML calibration

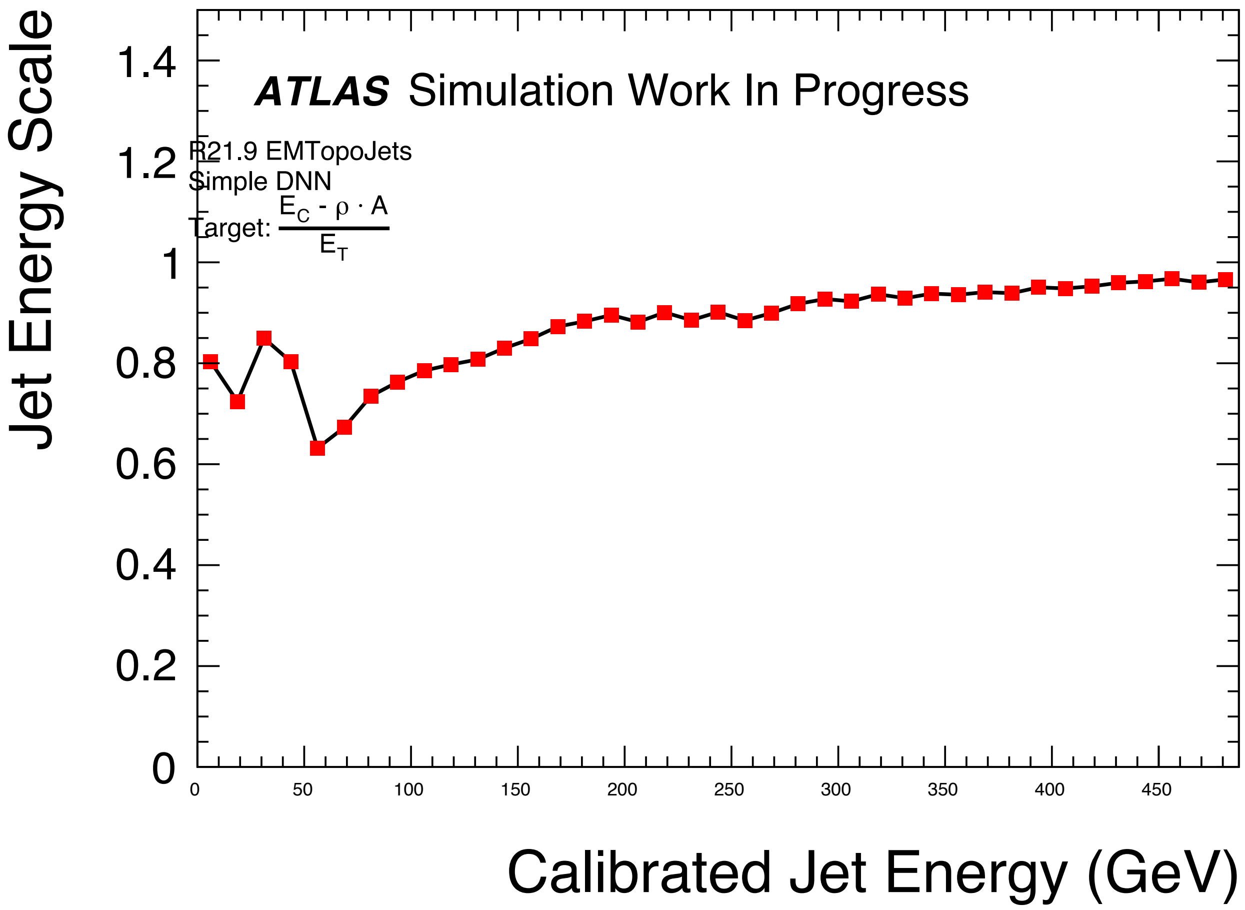


Jet Energy Scale - $E_{Constit}$ v. E_{Cal} - 0 to 500 GeV

Pre-Calibration

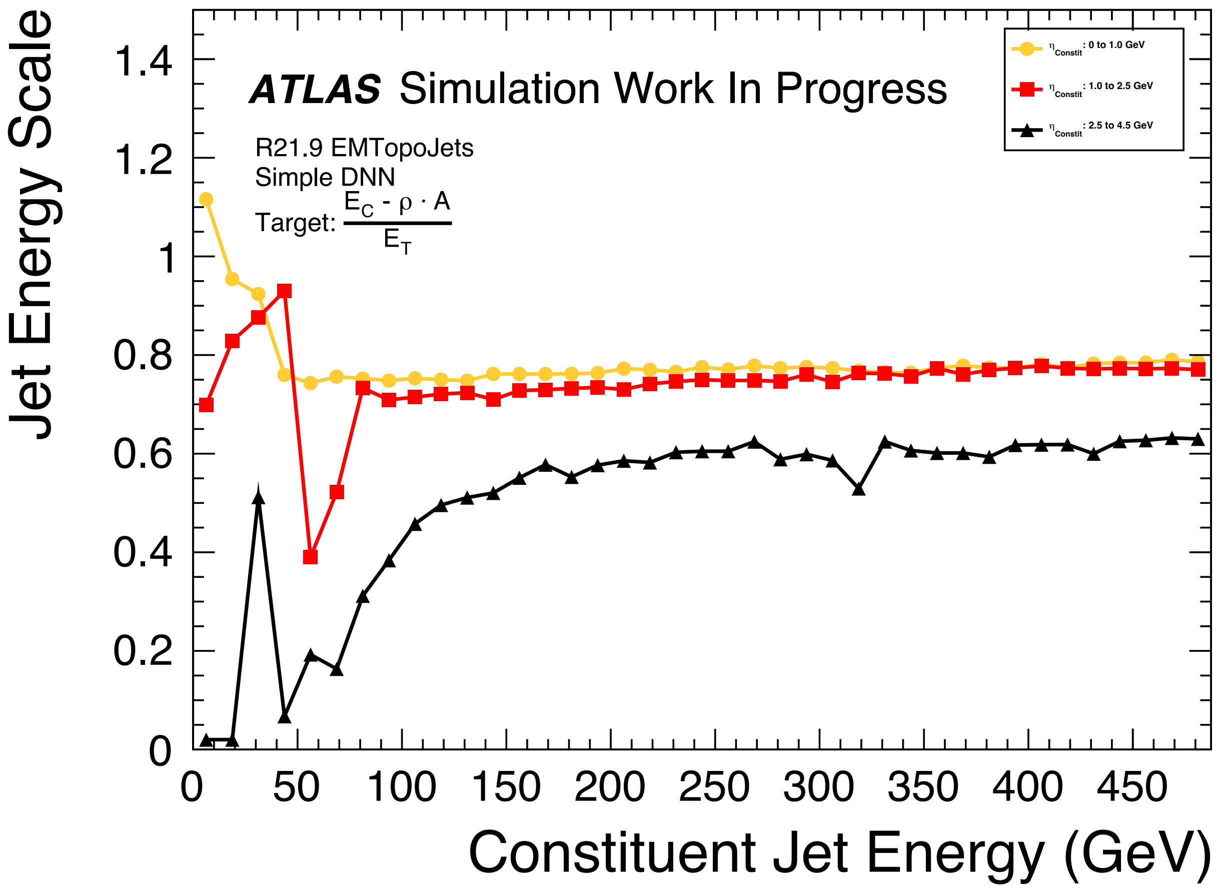


Post-Calibration

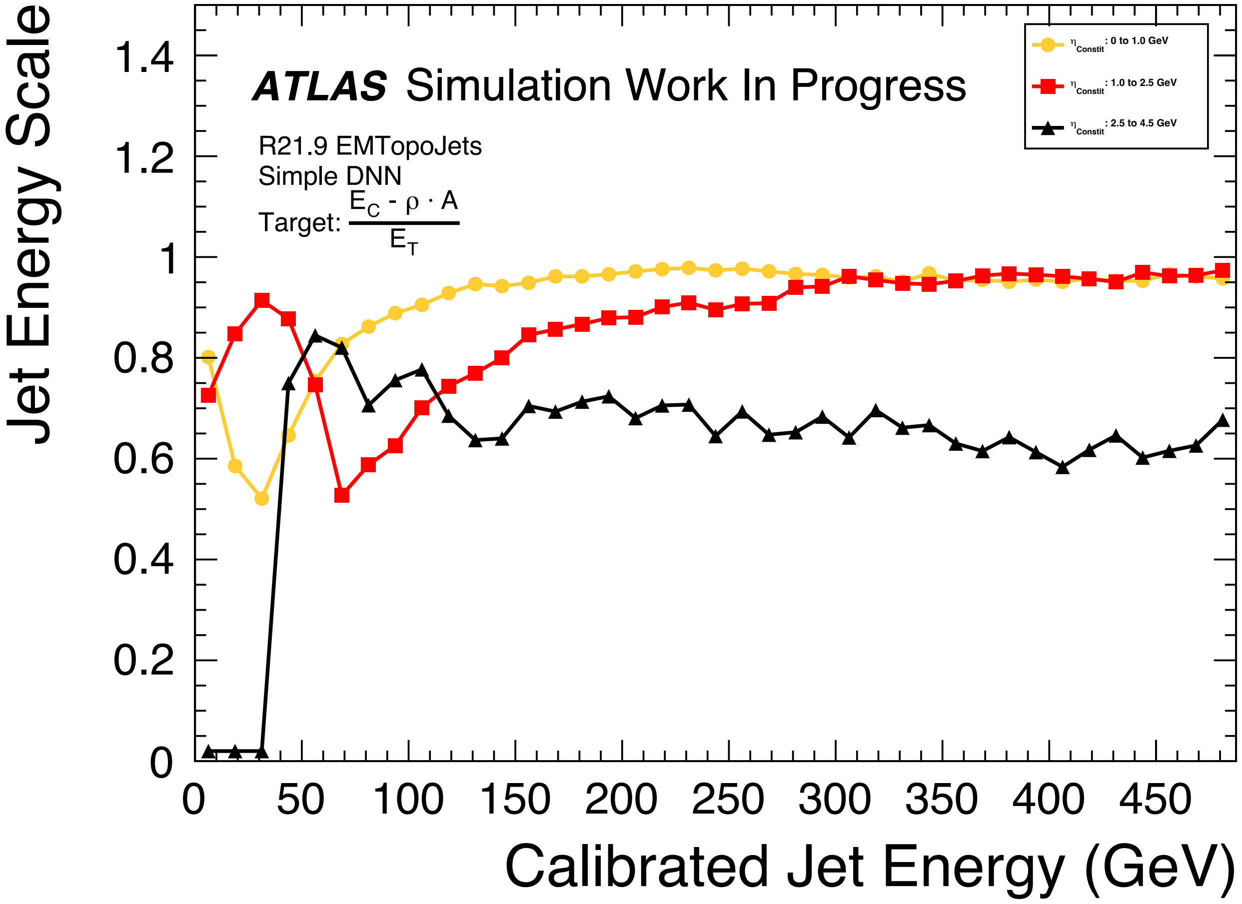


Jet Energy Scale - $E_{Constit}$ v. E_{Cal} (η Bins) - 0 to 500 GeV

Pre-Calibration

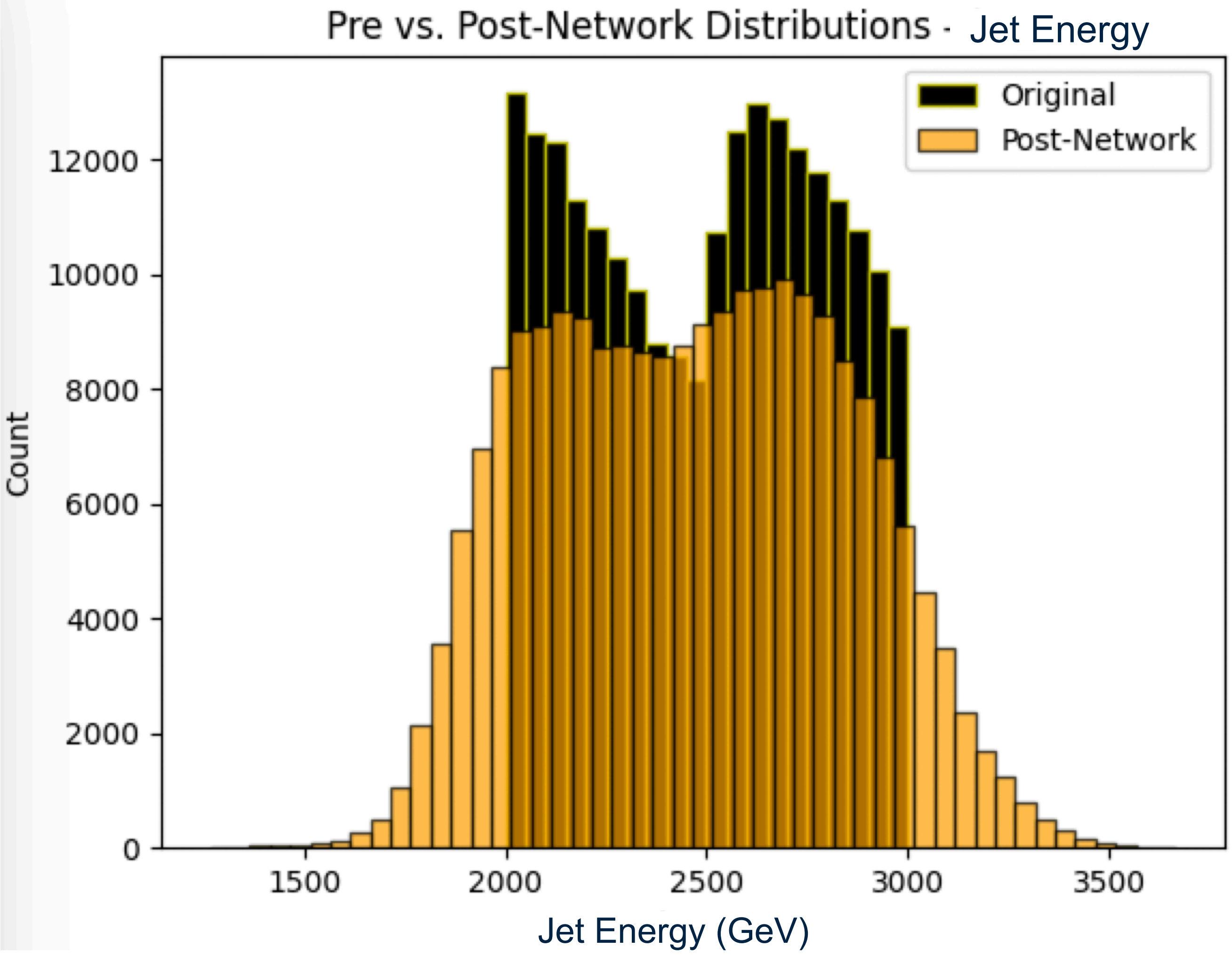


Post-Calibration



Loss Function

- Evaluate possible loss functions starting from mean square error
  - Goal: motivate the simplest possible network structure without compromising on performance
- Default TF options do well but still lacking
- End up back where we started at a Mixture Density Network (MDN)
  - Simpler this time!
- Still developing optimal implementation
  - 4.9M jets ~ 1 hour training, currently



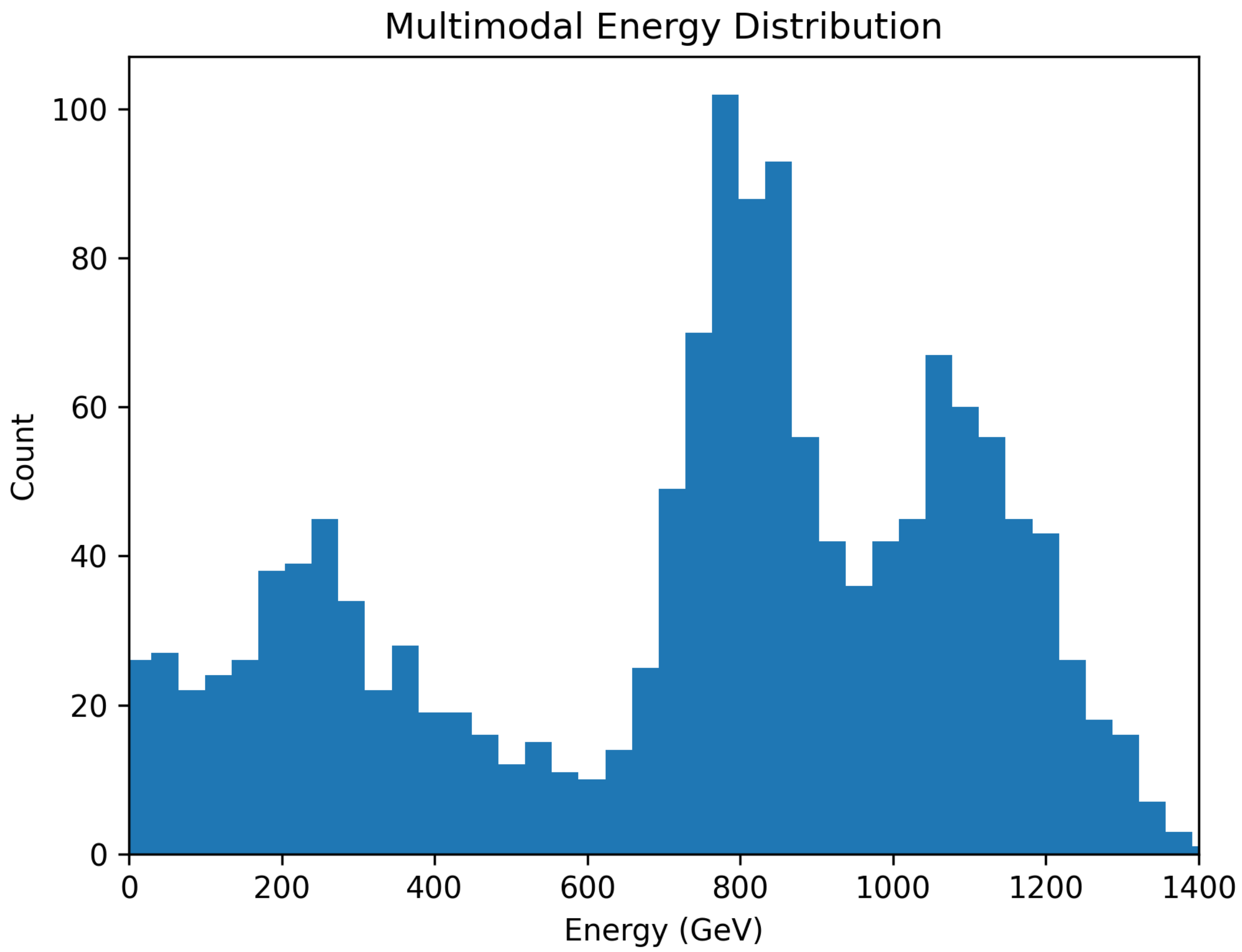
Jet truth (black) and predicted (yellow) energy values using the RMSLE loss function

Loss Function

- Evaluate possible loss functions starting from mean square error
  - Goal: motivate the simplest possible network structure without compromising on performance
- Default TF options do well but still lacking
- End up back where we started at a Mixture Density Network (MDN)
  - Simpler this time!
- Still developing optimal implementation
  - 4.9M jets ~ 1 hour training, currently

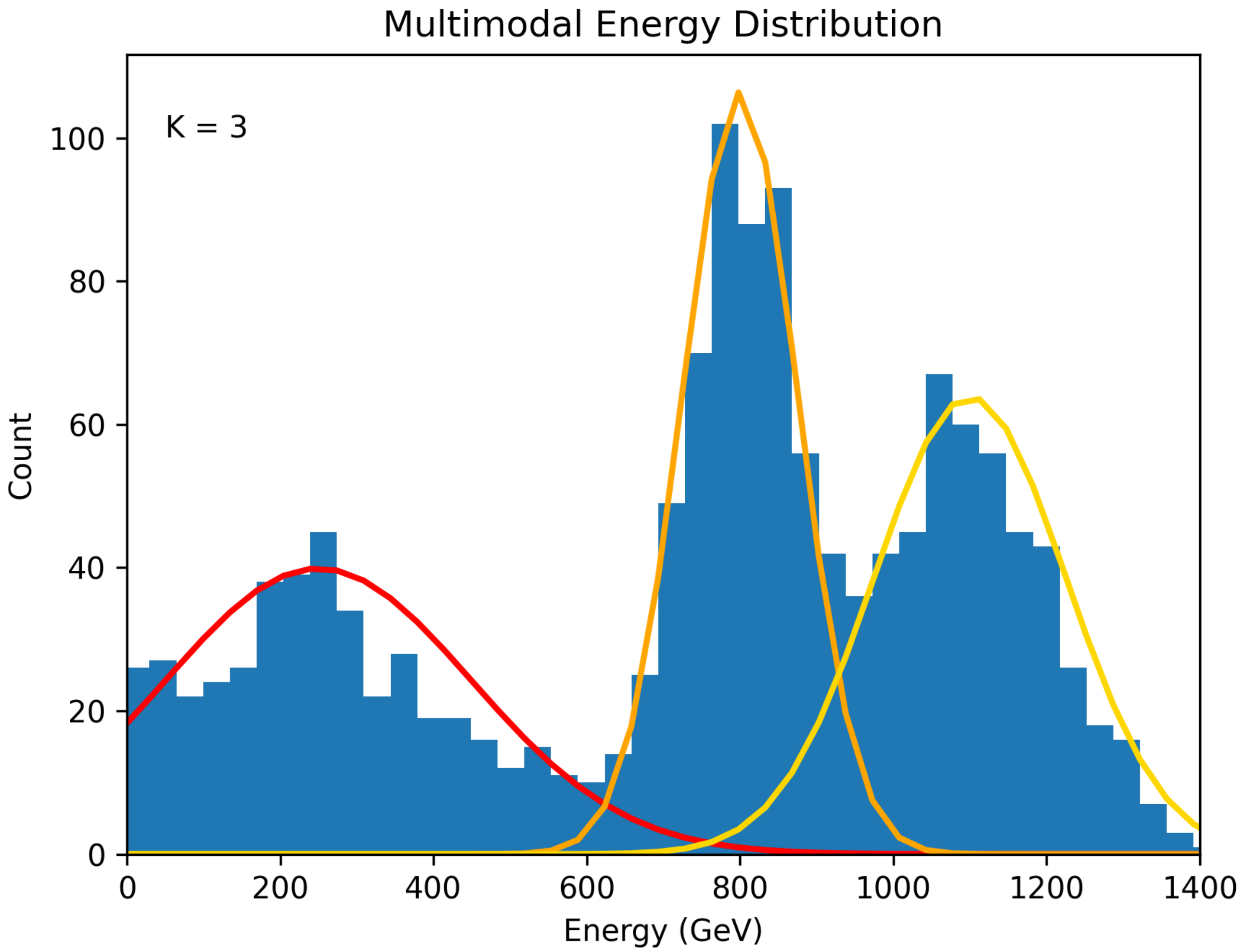
Loss Function

- Evaluate possible loss functions starting from mean square error
  - Goal: motivate the simplest possible network structure without compromising on performance
- Default TF options do well but still lacking
- End up back where we started at a Mixture Density Network (MDN)
  - Simpler this time!
- Still developing optimal implementation
  - 4.9M jets ~ 1 hour training, currently



Loss Function

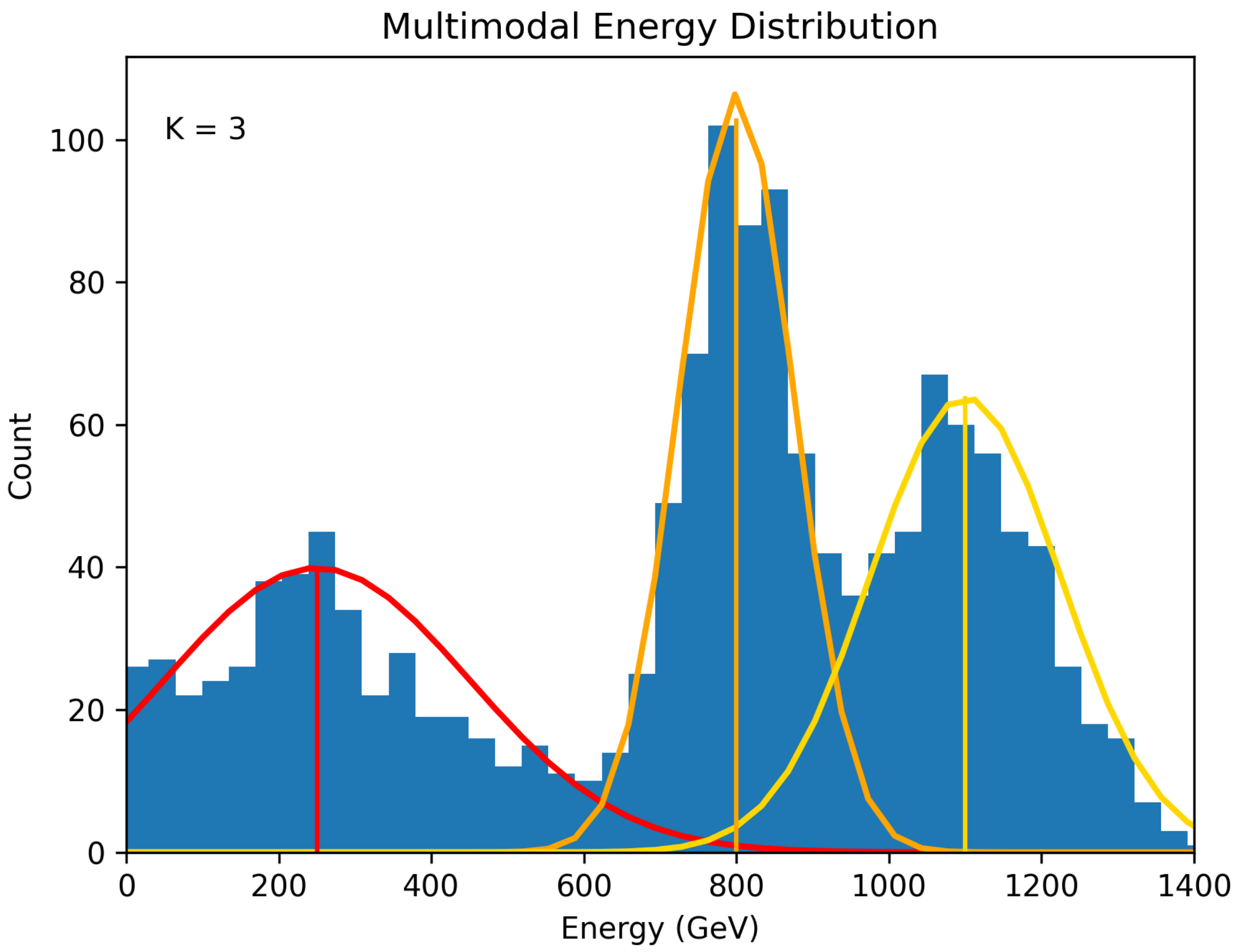
- Evaluate possible loss functions starting from mean square error
  - Goal: motivate the simplest possible network structure without compromising on performance
- Default TF options do well but still lacking
- End up back where we started at a Mixture Density Network (MDN)
  - Simpler this time!
- Still developing optimal implementation
  - 4.9M jets ~ 1 hour training, currently



$$P(Y = y \mid X = x) = \sum_{k=0}^{K-1} \Pi_k(x)\phi\left(y, \mu_k(x), \sigma_k(x)\right)$$

Loss Function

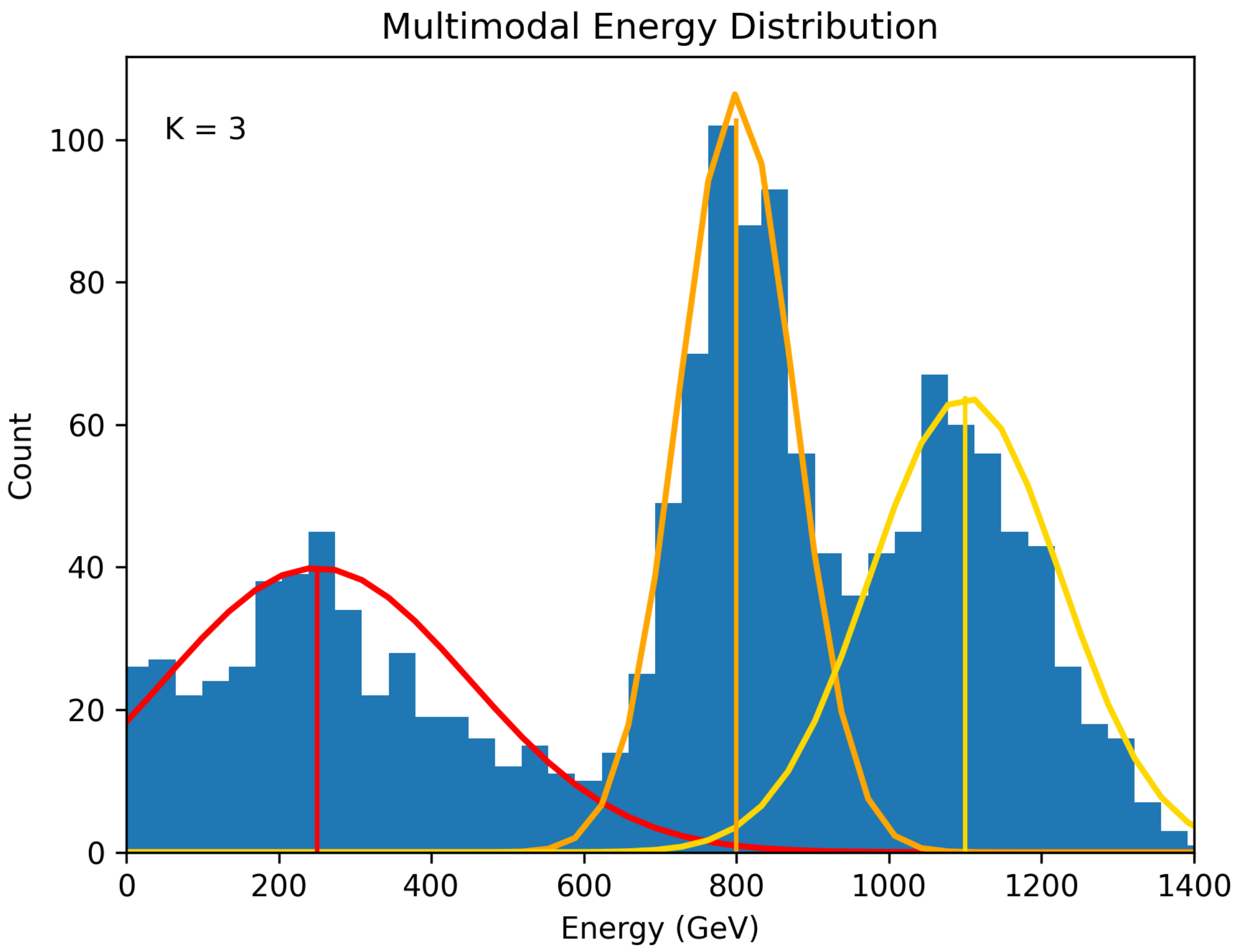
- Evaluate possible loss functions starting from mean square error
  - Goal: motivate the simplest possible network structure without compromising on performance
- Default TF options do well but still lacking
- End up back where we started at a Mixture Density Network (MDN)
  - Simpler this time!
- Still developing optimal implementation
  - 4.9M jets ~ 1 hour training, currently



$$P(Y = y \mid X = x) = \sum_{k=0}^{K-1} \Pi_k(x)\phi\left(y, \mu_k(x), \sigma_k(x)\right)$$

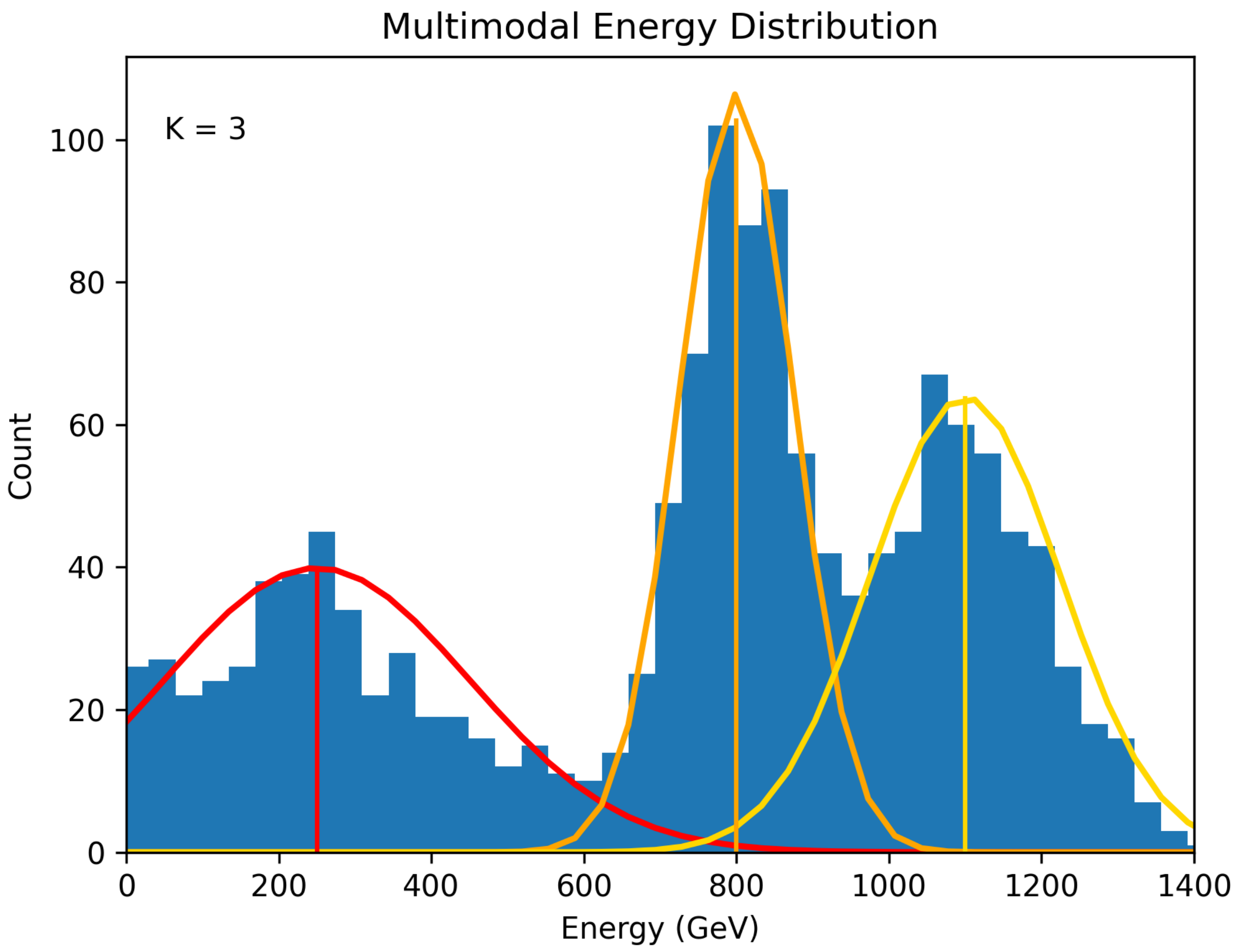
Loss Function

- Evaluate possible loss functions starting from mean square error
  - Goal: motivate the simplest possible network structure without compromising on performance
- Default TF options do well but still lacking
- End up back where we started at a Mixture Density Network (MDN)
  - Simpler this time!
- Still developing optimal implementation
  - 4.9M jets ~ 1 hour training, currently



Loss Function

- Evaluate possible loss functions starting from mean square error
  - Goal: motivate the simplest possible network structure without compromising on performance
- Default TF options do well but still lacking
- End up back where we started at a Mixture Density Network (MDN)
  - Simpler this time!
- Still developing optimal implementation
  - 4.9M jets ~ 1 hour training, currently



$$\mathscr{L}(y \mid x) = -\log\left[\sum_{k}^{K} \Pi_k(x)\phi(y, \mu(x), \sigma(x))\right]$$