Simultaneous Jet Calibration with ML including in situ JER Measurement

Journées de Rencontre des Jeunes Chercheurs

Laura Boggia

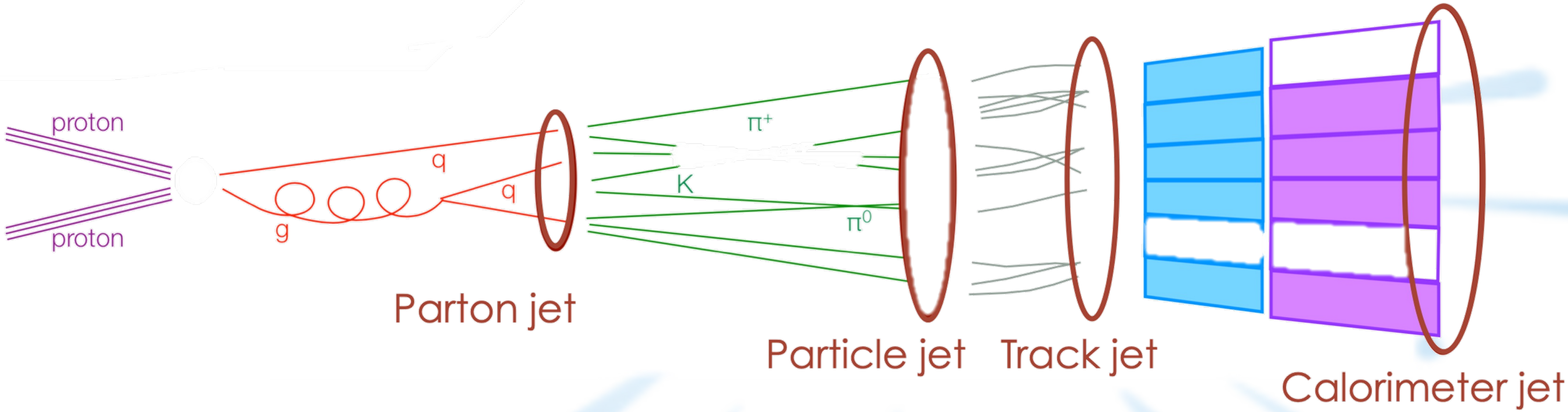
SMARTHEP is funded by the European Union's Horizon 2020 research and innovation programme, call H2020-MSCA-ITN-2020, under Grant Agreement n. 956086

Overview

- Quick summary of jet calibration
- Introduction to machine learning (ML)
- Initial machine learning model
- Modifications of machine learning model
- Intermediate results
- Outlook

Jets Physics

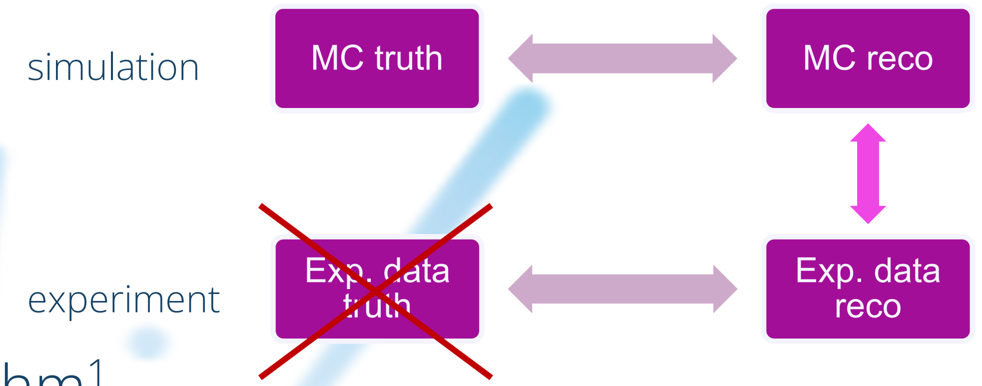
- Jets represent the shower produced by the hadronisation of a quark or gluon and is usually characterised by 4-vector: $(\vec{p}, E)$
- Its exact definition depends on the jet algorithm (often anti-kT algorithm[1])



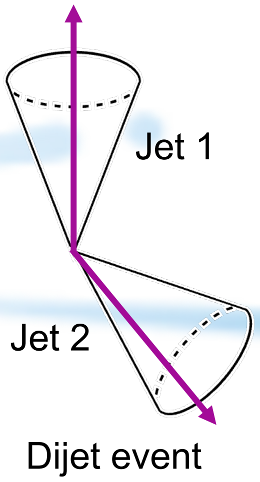
(figure from "Jet Inputs and MC Calibration", Dilia María Portillo, ATLAS collaboration, 2023)

[1] *("The anti-kt jet clustering algorithm", Cacciari et al., 2008)*

Jet Calibration



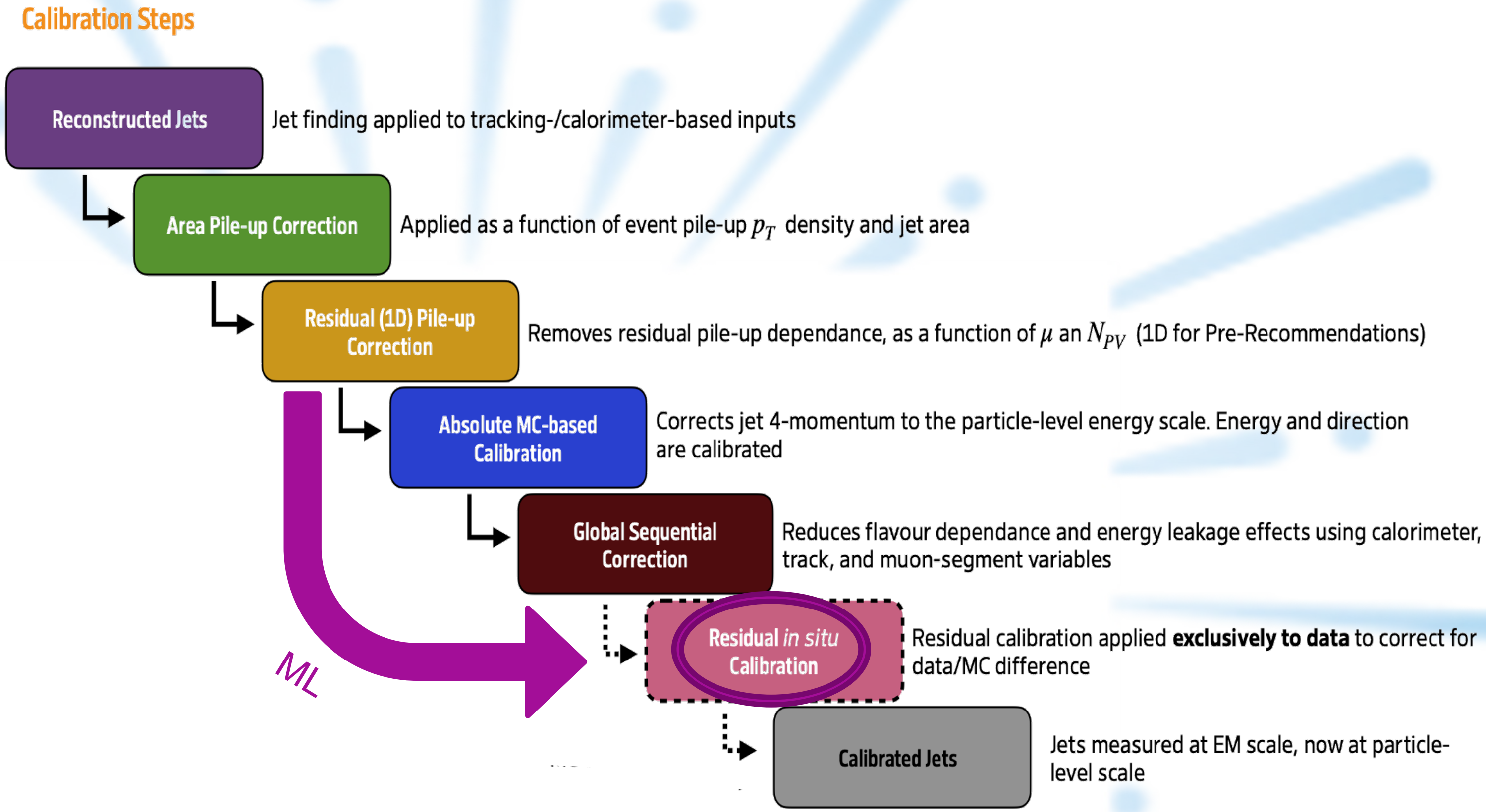
- Jet is an exp. observable defined by anti-kT algorithm[1]
  - Consist of shower of fundamental particles
  - Characterised by 4-vector: $(\vec{p}, E)$
- Calibration is essential because detector reacts differently to different kinds of particles (EM vs hadronic) → energy deposits differ depending on particle
- Truth jets:
  - "Hadron-level ('truth') jets are formed from detector-stable simulated particles…"[2]
  - Clustered using anti-kt jet algorithm
- Reco jets:
  - "Detector-level ('reco') jets are formed from topologically connected, noise-suppressed calorimeter cellclusters at the electromagnetic scale using […] the anti-kt jet algorithm…"[2]



[1] ("The anti-kt jet clustering algorithm", Cacciari et al., 2008)
[2] ("Generalized Numerical Inversion: A Neural Network Approach to Jet Calibration", ATLAS, 2018)

Jet Calibration in ATLAS

- On-going studies to replace current multi-step calibration scheme by ML model[1]
  - Current research: try to merge Absolute MC-based Calibration (MCJES) and GSC for faster testing of new algorithms
  - Currently done in MonteCarlo (MC) simulations only
- My task: optimise jet energy resolution (JER) including information from exp. data (in addition to MC)



HCW slides, Gediminas Glemža, 2022

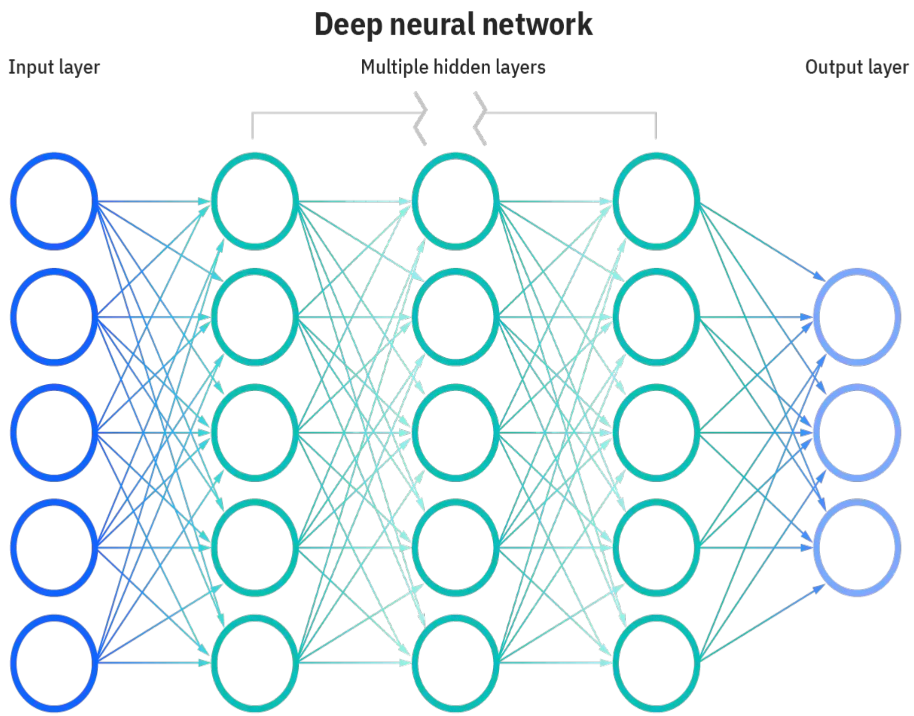
[1] (“New techniques for jet calibration with the ATLAS detector”, ATLAS collaboration, 2023)

Machine Learning

"Machine learning is the science of getting computers to act without being explicitly programmed."

(Andrew Ng, Stanford University)

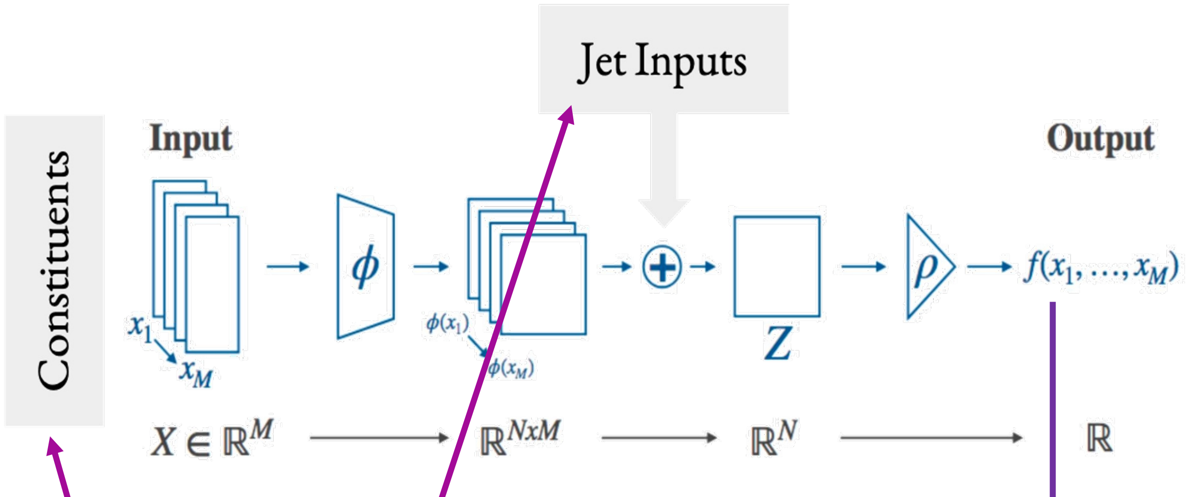
- Deep learning describes part of ML focusing on (deep) Neural Networks (NN)
- Can be used for learning more elaborate functions
- In general, learning model tries to optimise a loss function by repeatedly adjusting its own parameters
- We distinguish between supervised and unsupervised learning:
  - Supervised: we train the model by comparing the model's predictions to a known ground truth (e.g. mean-squared error)
  - Unsupervised: we don't have any ground truth to base our training on



ML Model for Jet Calibration

- Regression problem
  - Output is a probability distribution: $(\mu_{p_T}, \sigma_{p_T})$
  - Mean corresponds to calibration factor
- Deep sets[1]
  - Constructed using 2 NN, 1 for jet constituents, 1 for jet 4-vector
  - Model contains permutation invariant layer (e.g. sum layer) because order of events doesn't matter
- Supervised learning problem:
  - Compare truth μ to reco level $\mu(\theta),\ \sigma(\theta)$
  - Likelihood $\mathcal{L}(\theta) = \frac{1}{\sqrt{2\pi\sigma^2(\theta)}} \exp\left(-\frac{(\mu(\theta)-\mu)^2}{2\sigma^2(\theta)}\right)$
  - loss(θ) $= \min_\theta \left(-\log \mathcal{L}(\theta) \right)$

$$= \min_\theta \left[\frac{1}{2}\frac{(\mu(\theta)-\mu)^2}{\sigma^2(\theta)} + \log \sigma(\theta) + \text{const.}\right]$$



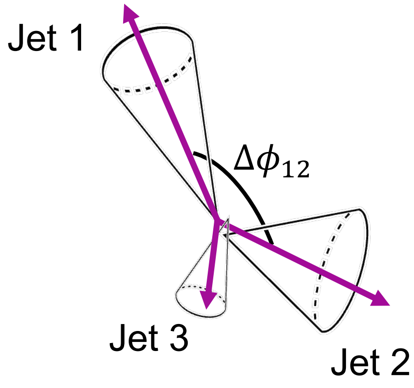
"Machine Learning the MC JES", K. Greif, C. Pollard, J. Roloff

Jet Constituents	Jet Inputs (reco)	True Jets	Outputs: calibration factor
(p_x, p_y, p_z, p_T)	(p_x, p_y, p_T, η, E)	$(p_x^{true}, p_y^{true}, p_T^{true}, \eta^{true}, E^{true})$	$(\mu_{p_T}, \log(\sigma_{p_T}))$
$(80, 4)$	$(5,)$	$(5,)$	$(2,)$

[1] ("Deep sets", Zaheer et al., 2018), ("Energy Flow Networks: Deep Sets for Particle Jets". Komiske et al., 2019)

Dijet Events

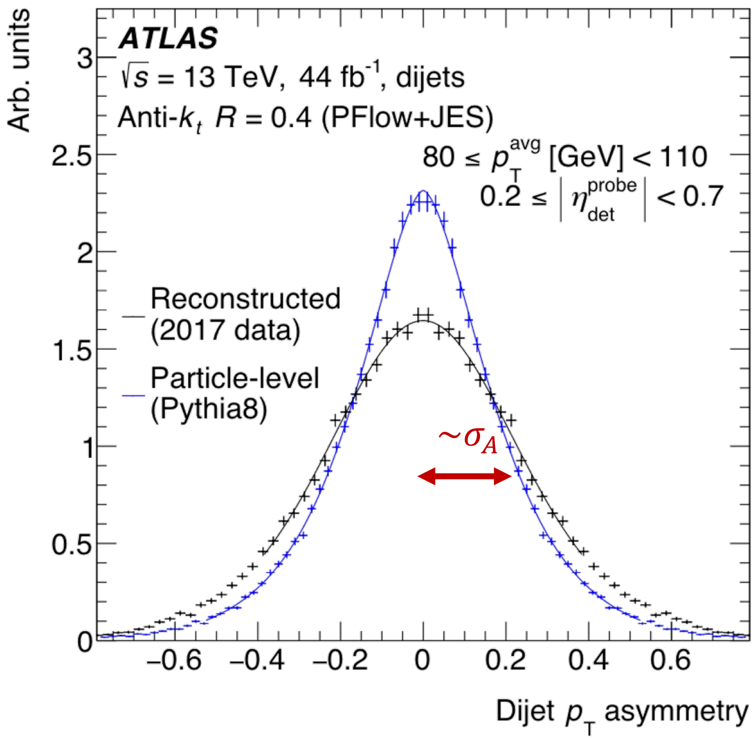
Jet Constituents	Jet Inputs (reco)	True Jets
(p_x, p_y, p_z, p_T)	(p_x, p_y, p_T, η, E)	$(p_x^{true}, p_y^{true}, p_T^{true}, \eta^{true}, E^{true})$
$(80, 4)$	$(5,)$	$(5,)$



Jet event

- Each collision event can register several jets
- Focus on events with two jets, i.e. dijet events
- Define dijet asymmetry[1]:
  - $\mathcal{A} = \frac{p_T^{ref} - p_T^{prob}}{p_T^{avg}}$, with $p_T^{avg} = \frac{p_T^{ref} + p_T^{prob}}{2}$,

  where ref and probe are randomly assigned to the two leading jets of every dijet event
  - Because of momentum conservation, this should be 0 in perfect case (i.e. no noise, reconstruction error)
  - For experimental data, we observe distribution around 0 where standard deviation (std) depends on our reconstructed jet resolution



[1] ("Jet energy scale and resolution measured in proton-proton collisions at $\sqrt{s}$ = 13 TeV with the ATLAS detector", ATLAS collaboration, 2021)

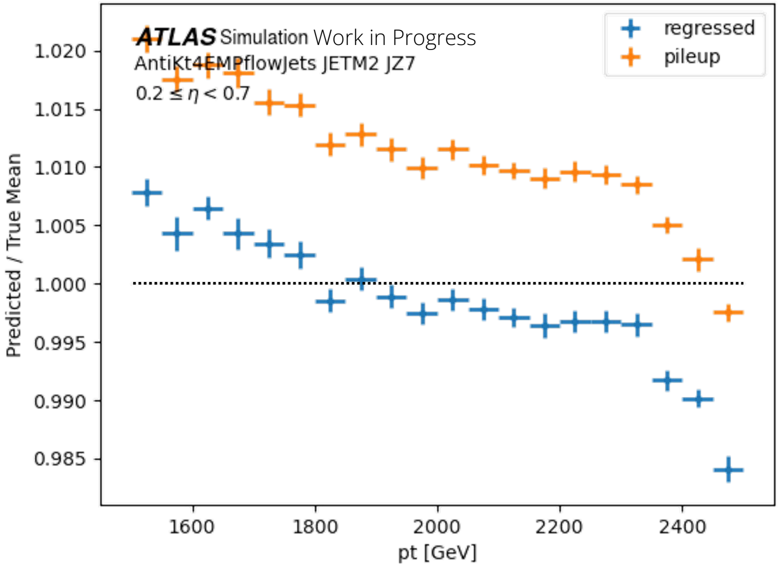
Jet Energy Resolution (JER)

- Jet energy resolution (JER) is related to std of dijet asymmetry (after subtracting the smearing from physics effects, present at hadron level):[1]
  - $(\sigma_{\mathcal{A}}^{det})^2 = (\sigma_{\mathcal{A}}^{reco})^2 - (\sigma_{\mathcal{A}}^{truth})^2$, in central part of detector
- Relative JER can be estimated from $\boldsymbol{\sigma_{\mathcal{A}}^{det}}$:[1]
  - Relative JER: $\frac{\sigma_{p_T}}{p_T} = \frac{\sigma_{\mathcal{A}}^{det}}{\sqrt{2}} \cong \frac{\sigma_{\mathcal{A}}^{reco}}{\sqrt{2}} \sim \sigma_{\mathcal{A}}^{reco}$
  - NN-based correction shouldn't impact truth, so it's sufficient to directly use $\boldsymbol{\sigma_{\mathcal{A}}^{reco}}$
  - Completely independent of true labels → useful for exp. Data
- Update loss function:
  - $\mathrm{loss}(\theta) \longrightarrow \mathrm{loss}(\theta) + f * \sigma_A{}'(\theta)$
  - ML model simultaneously minimises the JER measured in-situ and the original loss
  - No longer fully dependent on truth level, ML model is only partially supervised

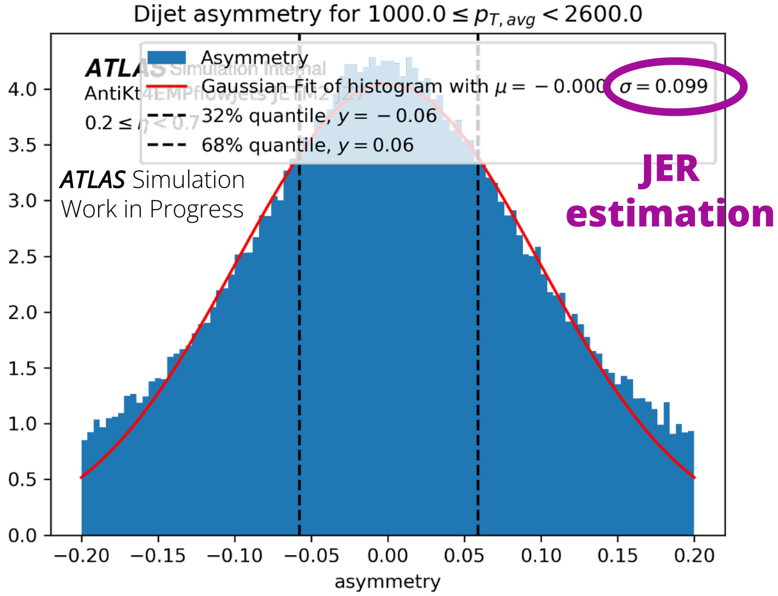
[1] (*"Jet energy scale and resolution measured in proton-proton collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector", ATLAS collaboration, 2021)*

Results with $f = 0$

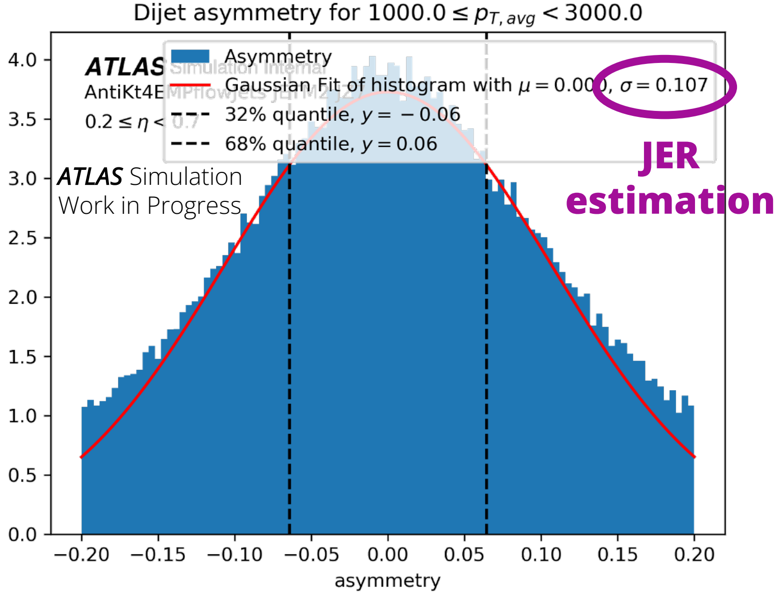
- Asymmetry factor f is fixed to 0
- ML model doesn't improve/has little effect on JER
  - JER of reco jets (at pileup level): $\sim 9.9\ \%$
  - JER of regressed jets (i.e. after applying calibration factors predicted by ML model): $\sim 10.7\ \%$
- Can JER be improved by adding asymmetry term in loss function, i.e. $f \neq 0$?



Testing set: reco jets



Testing set: regressed jets



First results: $f = 0$ vs $f \neq 0$

$f = 0$

- Asymmetry factor f is fixed to 0
- Predicted pT values:
  - $p_T^{true} \in [1100, 2600]$ GeV
  - $p_T \in [1000, 3000]$ GeV
- JER estimation:
  - JER of jets before training: $\sim 9.9\ \%$
  - JER of regressed jets (i.e. after applying calibration factors predicted by ML model): $\sim 10.7\ \%$

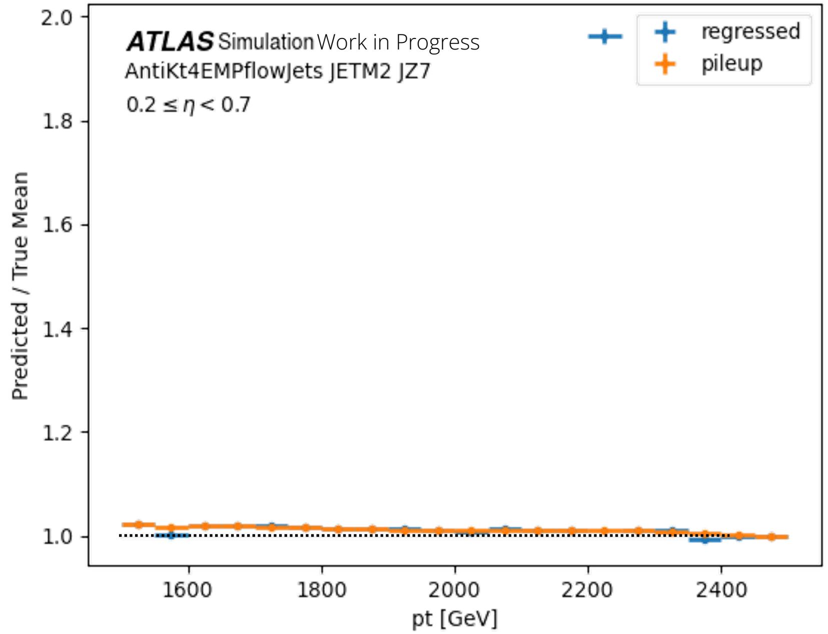
$f \neq 0$

- Asymmetry factor f is varied between 0 and 10
- Predicted pT values:
  - $p_T^{true} \in [1100, 2600]$ GeV
  - $p_T \in [-1'792'700, 394'000]$ GeV
- JER estimation:
  - JER of jets before training: $\sim 9.9\ \%$
  - JER of regressed jets (i.e. after applying calibration factors predicted by ML model): $\sim 10.2\ \%$

→ **First naive implementation failed!**

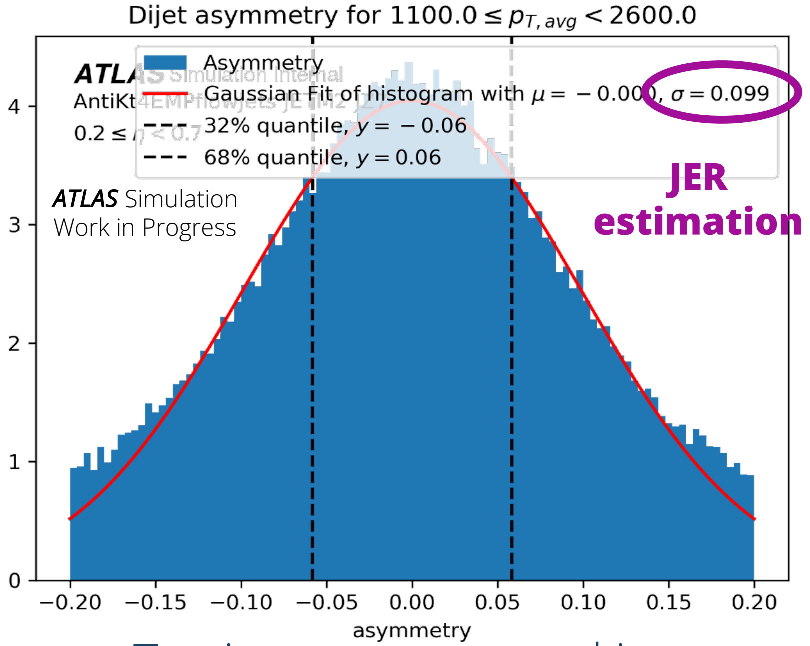
First Results with $f \neq 0$

- Predicted pT much worse
- Predicted JER slightly better:
  - JER of jets before training: $\sim 9.9\ \%$
  - JER of regressed jets (i.e. after applying calibration factors predicted by ML model): $\sim 10.2\ \%$

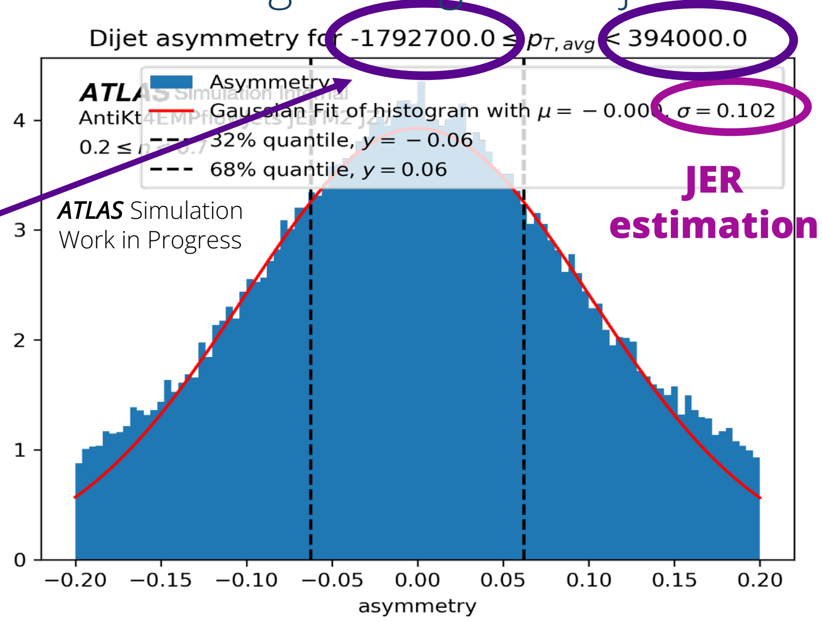


Problem: Why do we have negative calibration factors?

Testing set: reco jets



Testing set: regressed jets

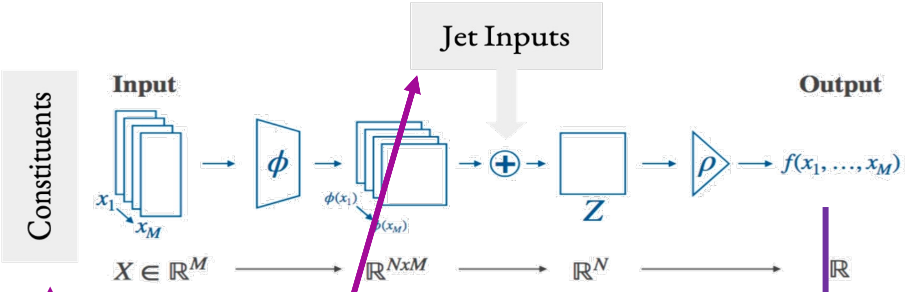


What's next

- Naive approach doesn't work immediately
- It seems the two loss terms contradict/work against each other
  - Add softplus layer to restrict outputs of NN to positive values[1]
  - Introduce penalty term that forbids unphysical solution
  - Standardise truth targets
- Use **GSC variables**[2] (which are known to improve JER) in addition to jet 4-vector as jet inputs

NEW!

GSC variables
Energy fractions, tracking, detector eta, muon segments, pileup etc.
$(20,)$



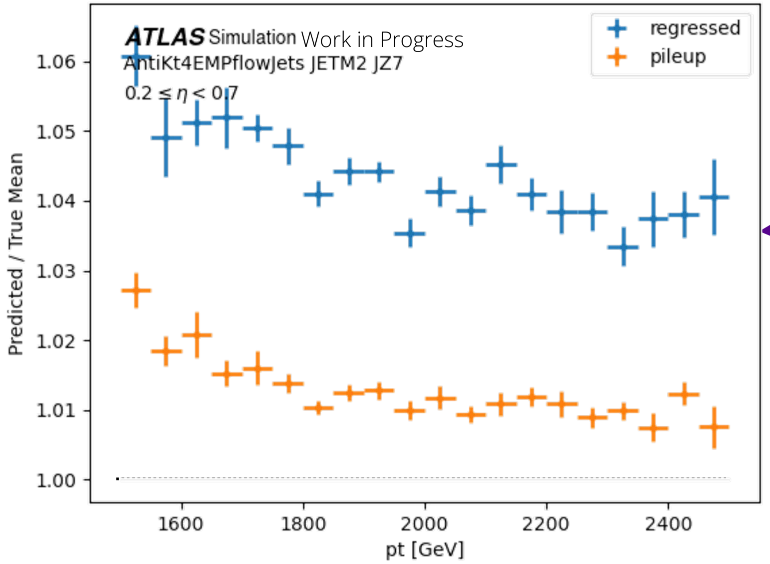
"Machine Learning the MC JES", K. Greif, C. Pollard, J. Roloff

Jet Constituents	Jet Inputs (reco)	True Jets	Outputs: calibration factor
(p_x, p_y, p_z, p_T)	(p_x, p_y, p_T, η, E)	$(p_x^{true}, p_y^{true}, p_T^{true}, \eta^{true}, E^{true})$	$(\mu_{p_T}, \log(\sigma_{p_T}))$
$(80, 4)$	$(5,)$	$(5,)$	$(2,)$

[1] ("tf.math.softplus", TensorFlow, September 2022),
[2] ("New techniques for jet calibration with the ATLAS detector", ATLAS collaboration, 2023)

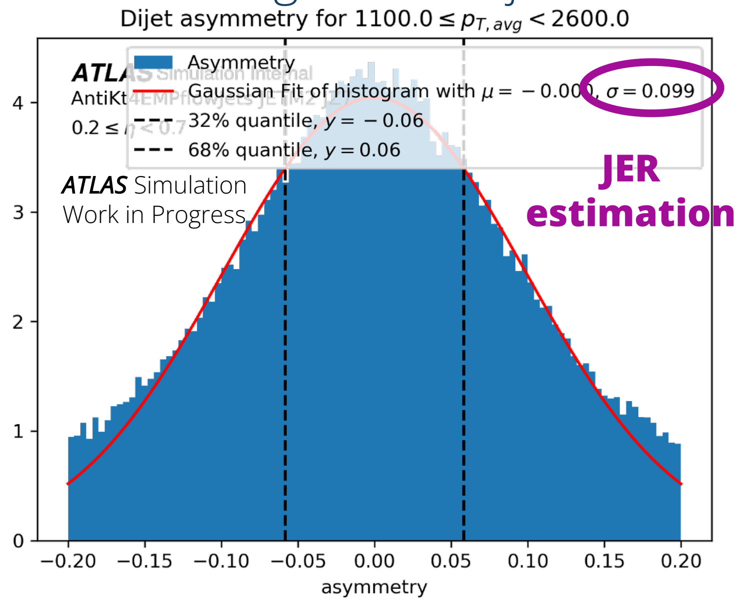
More results with $f \neq 0$

- New variables added
- Softplus layer applied
- Predicted / True ratio pf pT is getting closer to 1 but JER is worse
    - JER of reco jets: $\sim 9.9\ \%$
    - JER of regressed jets (i.e. after applying calibration factors predicted by ML model): $\sim 12.7\ \%$

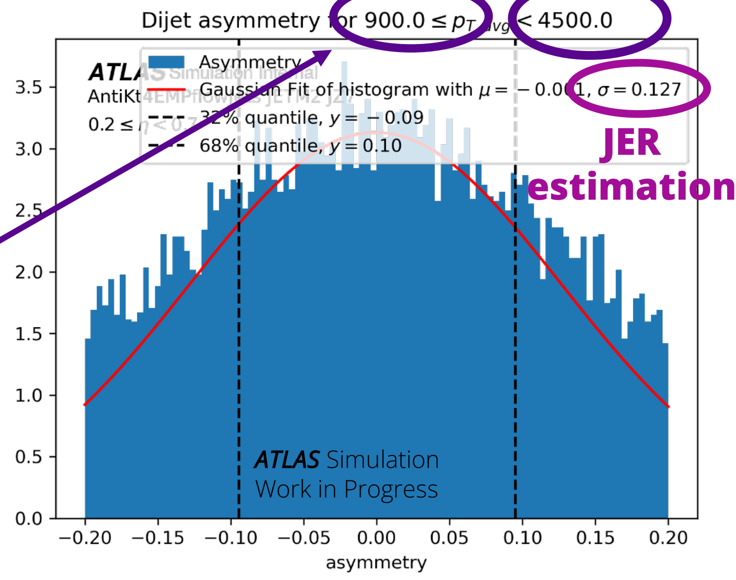


Problem: pT predictions are still off

Testing set: reco jets



Testing set: regressed jets



Thank you for your attention!

Laura Boggia
laura.boggia@cern.ch

SMARTHEP is funded by the European Union's Horizon 2020 research and innovation programme, call H2020-MSCA-ITN-2020, under Grant Agreement n. 956086

Backup

SMARTHEP is funded by the European Union's Horizon 2020 research and innovation programme, call H2020-MSCA-ITN-2020, under Grant Agreement n. 956086

Deep Sets Model

- Model contains permutation invariant layer (e.g. sum layer)
- Why do we want permutation invariance for jet physics?
  - Order of events doesn't matter, each collision event happens independently
  - Can guarantee infrared and collinear (IRC) safety which is important for comparing QCD theory predictions to experimental results

IRC-Safe Observable Decomposition. *An IRC-safe observable $\mathcal{O}$ can be approximated arbitrarily well as:*

$$\mathcal{O}(\{p_1, \ldots, p_M\}) = F\left(\sum_{i=1}^{M} z_i \Phi(\hat{p}_i)\right), \tag{1.2}$$

where z_i is the energy (or p_T) and $\hat{p}_i$ the angular information of particle i.

Approximate functions F, Φ with neural networks

[1] ("Deep sets", Zaheer et al., 2018),
("Energy Flow Networks: Deep Sets for Particle Jets". Komiske et al., 2019)

ML Model for Jet Calibration

- Regression problem

  Output is a probability distribution: $(\mu_{p_T}, \sigma_{p_T})$

  Mean corresponds to calibration factor

- Deep sets[1]

  Constructed using 2 NN, 1 for jet constituents, 1 for jet 4-vector

  Model contains permutation invariant layer (e.g. sum layer) because order of events doesn't matter

- Supervised learning problem:

  Compare truth μ to reco level $\mu(\theta),\ \sigma(\theta)$

  Likelihood $\mathcal{L}(\theta) = \frac{1}{\sqrt{2\pi\sigma^2(\theta)}} \exp\left(-\frac{(\mu(\theta)-\mu)^2}{2\sigma^2(\theta)}\right)$

  $$\text{loss}(\theta) = \min_\theta \left(-\log \mathcal{L}(\theta) \right)$$

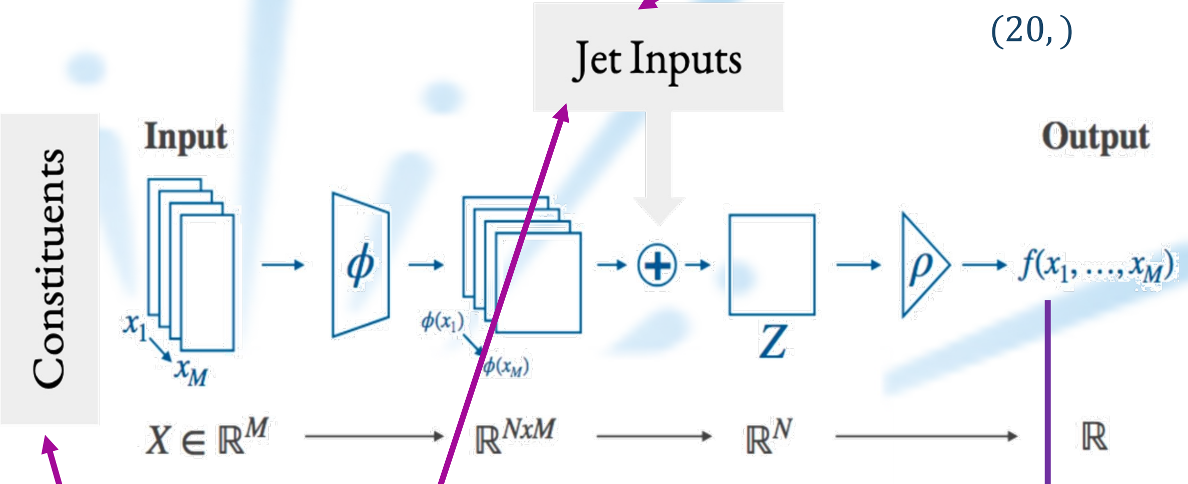
  $$= \min_\theta \left[\frac{1}{2}\frac{(\mu(\theta)-\mu)^2}{\sigma^2(\theta)} + \log\sigma(\theta) + \text{const.}\right]$$

NEW!

GSC variables

Energy fractions, tracking, detector eta, muon segments, pileup etc.

$(20,)$



"[Machine Learning the MC JES](#)", K. Greif, C. Pollard, J. Roloff

Jet Constituents	Jet Inputs (reco)	True Jets	Outputs: calibration factor
(p_x, p_y, p_z, p_T)	(p_x, p_y, p_T, η, E)	$(p_x^{true}, p_y^{true}, p_T^{true}, \eta^{true}, E^{true})$	$(\mu_{p_T}, \log(\sigma_{p_T}))$
$(80, 4)$	$(5,)$	$(5,)$	$(2,)$

[1] ("Deep sets", Zaheer et al., 2018), ("Energy Flow Networks: Deep Sets for Particle Jets". Komiske et al., 2019)

Add GSC variables

Calorimeter	$f_{\mathrm{LAr}0-3*}$	The E_{frac} measured in the 0th-3rd layer of the EM LAr calorimeter
	$f_{\mathrm{Tile}0*-2}$	The E_{frac} measured in the 0th-2nd layer of the hadronic tile calorimeter
	$f_{\mathrm{HEC},0-3}$	The E_{frac} measured in the 0th-3rd layer of the hadronic end cap calorimeter
	$f_{\mathrm{FCAL},0-2}$	The E_{frac} measured in the 0th-2nd layer of the forward calorimeter
	$N_{90\%}$	The minimum number of clusters containing 90% of the jet energy
Jet kinematics	$p_{\mathrm{T}}^{\mathrm{JES}}$ *	The jet p_{T} after the MCJES calibration
	η^{det}	The detector η
Tracking	w_{track}*	The average p_{T}-weighted transverse distance in the η-ϕ plane between the jet axis and all tracks of $p_{\mathrm{T}} > 1$ GeV ghost-associated with the jet
	N_{track}*	The number of tracks with $p_{\mathrm{T}} > 1$ GeV ghost-associated with the jet
	f_{charged}*	The fraction of the jet p_{T} measured from ghost-associated tracks
Muon segments	N_{segments}*	The number of muon track segments ghost-associated with the jet
Pile-up	μ	The average number of interactions per bunch crossing
	N_{PV}	The number of reconstructed primary vertices

Table 1: List of variables used as input to the GNNC. Variables with a * correspond to those that are also used by the GSC.

[1] *(see table 1 in "New techniques for jet calibration with the ATLAS detector", ATLAS collaboration, 2023)*

Bayesian Optimisation of Hyperparameters

- Training set:
  - JETM2 JZ7
  - Initially 3Mio events but after selection cuts (dijet & η) only ca. 677k
  - Unflattened (because current resampling seems bad for training)
- Bayesian optimisation of hyperparameters
  - 10 trials with 10 different validation folds

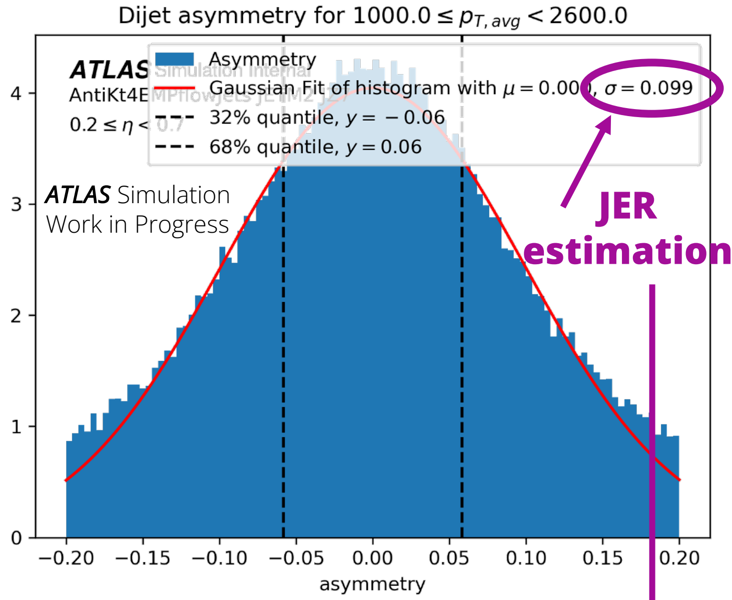
Hyperparameter Search Space

Use $\log(p_T)$	Dropout cluster	Dropout jet	Learning rate	Factor asymmetry term
[False, True]	[0.0, 0.1, 0.2, 0.3, 0.4, 0.5]	[0.0, 0.1, 0.2, 0.3, 0.4, 0.5]	[0.0001, 0.01]	$f \in [0, 10]$

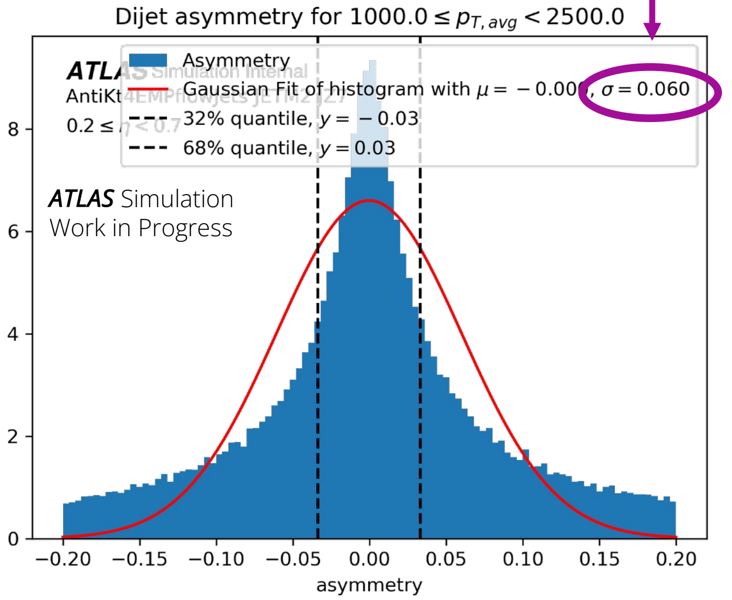
Dijet Asymmetry of JETM2 JZ7 (before Training)

- Truth dijet asymmetry has non-Gaussian tails
  - Use Gaussian as a first approximation
  - Can be improved by fitting convolution of exponential and Gaussian function[1]
- Goal is to minimise JER
  - Cannot get better than truth level
  - True asymmetry is limited by smearing from physics effect
- After training:
  - Apply predicted calibration factors to uncalibrated test samples
  - Check their p_T distribution, dijet asymmetry & estimate the JER from it
  - Call them 'regressed jets'

Testing set: reco jets

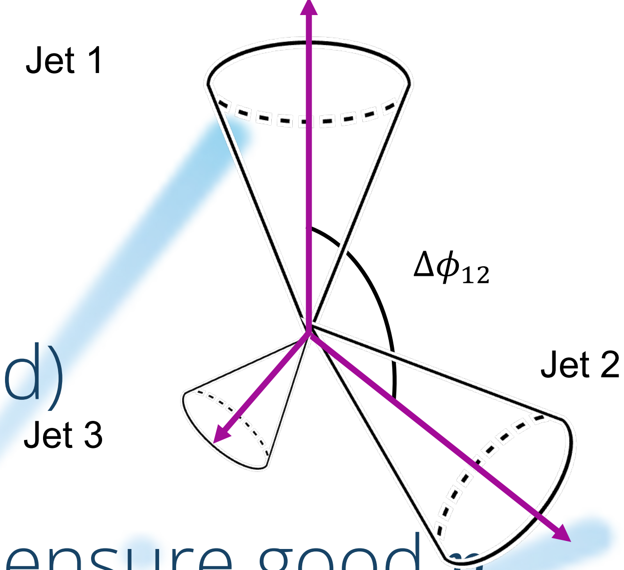


Testing set: true jets



[1] ("Jet energy scale and resolution measured in proton-proton collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector", ATLAS collaboration, 2021)

Input: Selection Criteria



- Central jets (to simplify problem, will be extended)

$$|\eta| \in [0.2, 0.7]$$

- Apply dijet topology cuts[1] on jet components to ensure good p_T balance between leading jets

$$\Delta\phi_{12} > 2.7 \text{ rad}$$
$$p_{T_3} < \max(25 \text{ GeV}, 0.25 \cdot p_{T,avg})$$

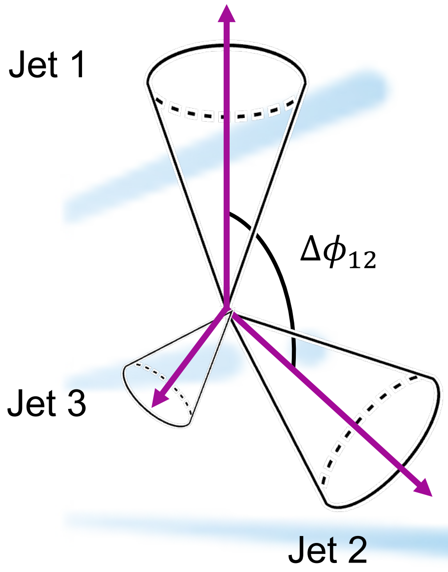
- pT between 800 and 2800 GeV because using JZ7
  - Later add more JZ slices

[1] ("Jet energy scale and resolution measured in proton-proton collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector", ATLAS collaboration, 2021)

Input: MC Samples

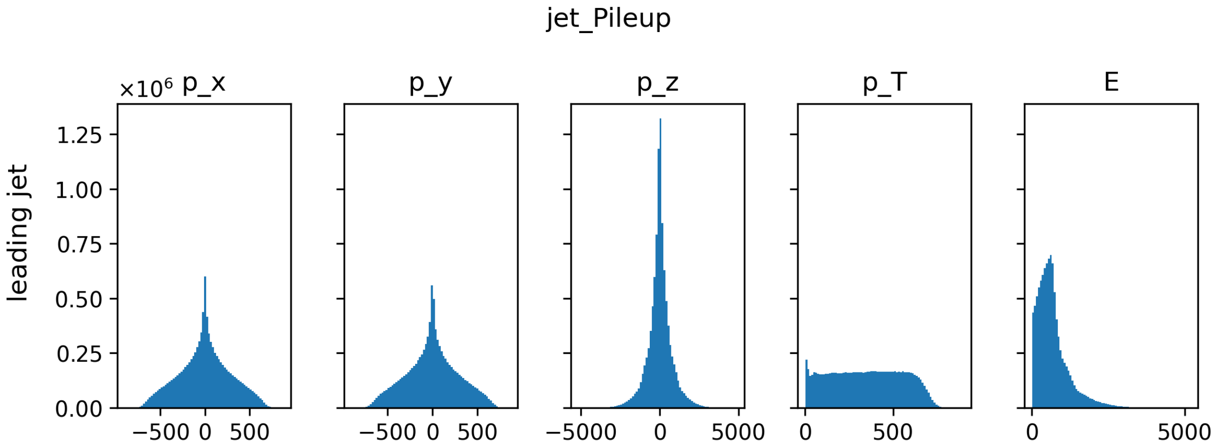
Input data	Jet Constituents	Jet Inputs
old	(p_x, p_y, p_z, p_T)	(p_x, p_y, p_z, p_T, E)
new	$(p_{x_i}, p_{y_i}, p_{T_i}, \eta_i)$, $i \in \{1, 2, 3\}$	(p_{T_i}), $i \in \{1, 2, 3\}$

- Old input samples:
  - Per event: 1-2 leading jets, no event info
  - All jets are treated independently
  - Isolated jets, lots of monojet events
  - Empty entries are filled with mask value: 0
  - Info about masking will be passed on to NN
- Modify format of input samples:
  - Keep event info of 3 leading jets
  - Empty entries are filled with new mask value: -10k
- Motivation: apply dijet topology cuts on jet components to ensure good p_T balance between leading jets

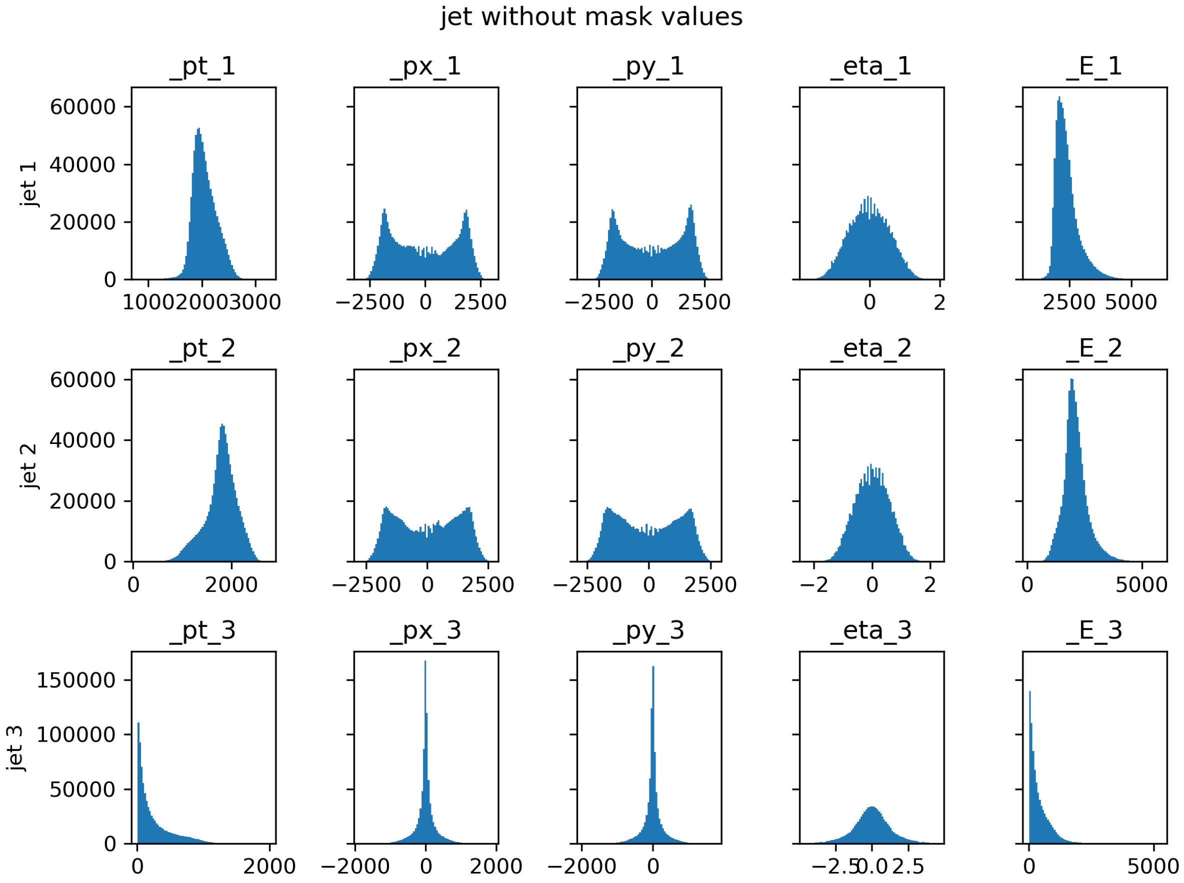


Input: Jet Components

Old MC samples



New MC samples



→ Note that p_T distribution on LHS has been flattened by resampling

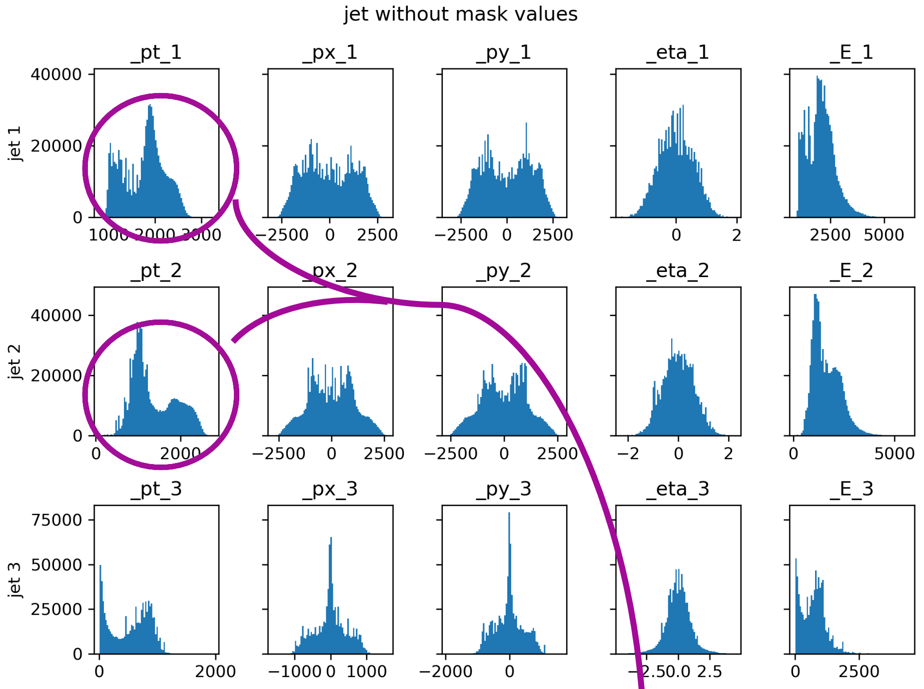
→ On RHS no resampling/flattening

Produced from mc20a, JETM2, JZ7

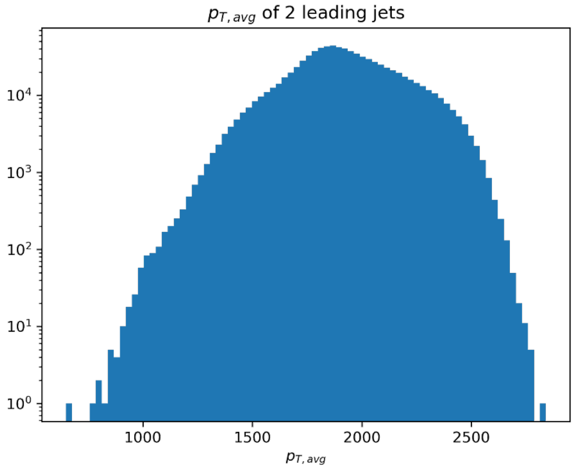
Input: Jet Components

- Events have been resampled to flatten distribution of $\log p_T^{avg}$ where $p_T^{avg} = (p_{T_1} + p_{T_2})/2$
    - This approach was chosen because $\log p_T^{avg}$ is physically significant
- PROBLEM:
    - Resampling assigns some very large weights to certain events
    - Weights differ by several orders of magnitude

New MC samples: resampled



Before resampling



With resampling