SMARTHEP
REAL-TIME ANALYSIS FOR SCIENCE AND INDUSTRY

Jet calibration with data-based ML training and identifying anomalies for applications in HEP and Finance

SMARTHEP Annual Meeting
27th of November 2023
Laura Boggia

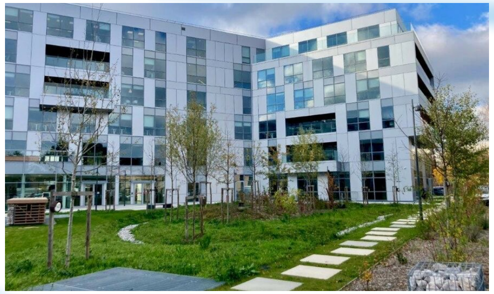
SMARTHEP is funded by the European Union's Horizon 2020 research and innovation programme, call H2020-MSCA-ITN-2020, under Grant Agreement n. 956086

Overview

- Introduction
  - About me
  - Activities during PhD
- ATLAS Qualification Task
- Fraud Detection with IBM

About Me...

- Swiss & Italian
- Grew up in Switzerland
- 2017-2020: BSc in Physics at EPFL
- 2020-2022: MSc in Physics at ETH
  - Focus on Theoretical Physics, e.g. QFT and GR
  - Thesis on Quantum ML for HEP with IBM Research Zurich
- 2022-Present: PhD with IBM Research & LPNHE at Sorbonne Université
  - Supervised by Bogdan Malaescu (LPNHE) & Shubham Gupta (IBM)
  - Also working with Anja Butter (LPNHE), Pierre Feillet (IBM) and other members of the ATLAS collaboration



Various Activities during PhD

- Workshops
  - Sep 2023: 'ATLAS Hadronic Calibration Workshop'
  - Oct 2023: 'Journées de Rencontre des Jeunes Chercheurs'
  - Jan 2024: 'Inter-experiment Machine Learning Workshop'
- Outreach
  - Oct 2022/2023: 'Fête de la Science'
    - 'My thesis in 5 minutes'
    - Guided tours of the lab for the public
- Training
  - Nov 2022: 'ATLAS Induction Day and Software Tutorial'
  - Dec 2022: 'MOOC on Scientific Integrity'
  - Jun 2023: 'Elements of Statistics'
  - Aug 2023: 'HEP C++ Essentials Course'
  - and SMARTHEP schools…

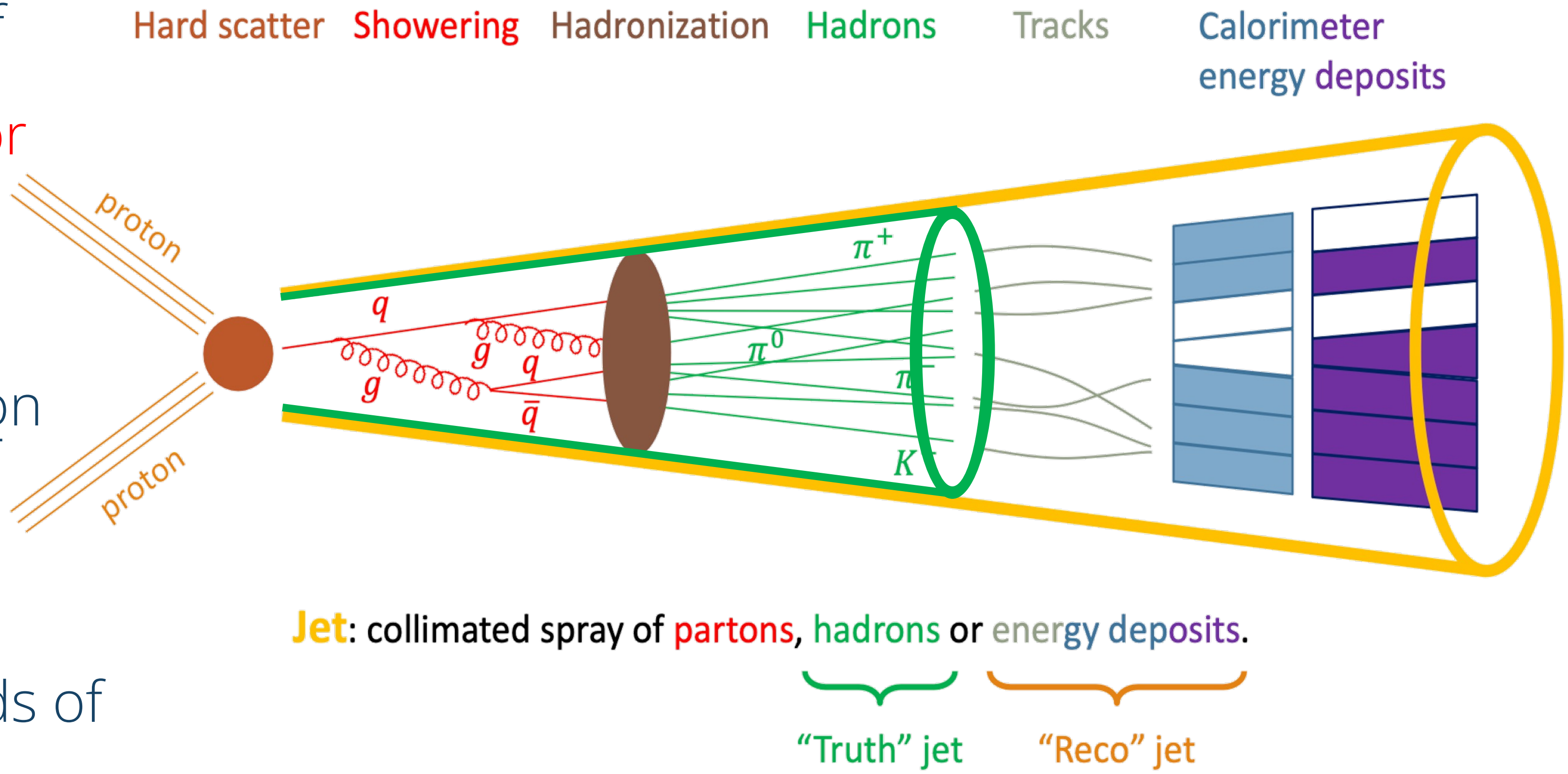
Simultaneous jet calibration with ML including in situ JER measurement

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

Jets Physics

- Jets represent the spray of particles produced by the hadronization of a quark or gluon
- Characterised by 4-vector: $(\vec{p}, E)$
- Exact definition depends on jet algorithm (often anti-kT algorithm[1])
- Calibration is essential because detector reacts differently to different kinds of particles (EM vs hadronic)

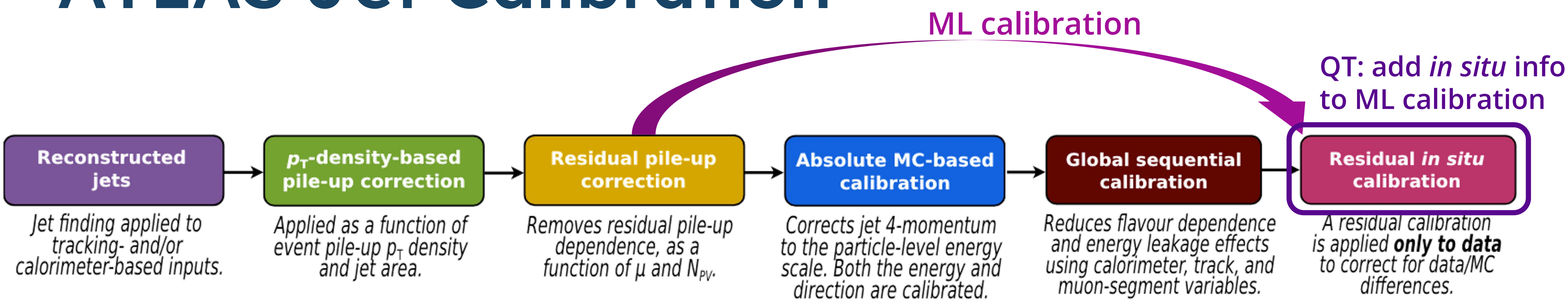
→ energy deposits differ depending on particle



(figure from Louis Ginabat, ATLAS collaboration, 2023)

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

ATLAS Jet Calibration



- 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 Global Sequential Calibration (GSC) for faster testing of new algorithms using MC samples
- My QT: optimise jet energy resolution (JER) including information from exp. data (in addition to MC samples)

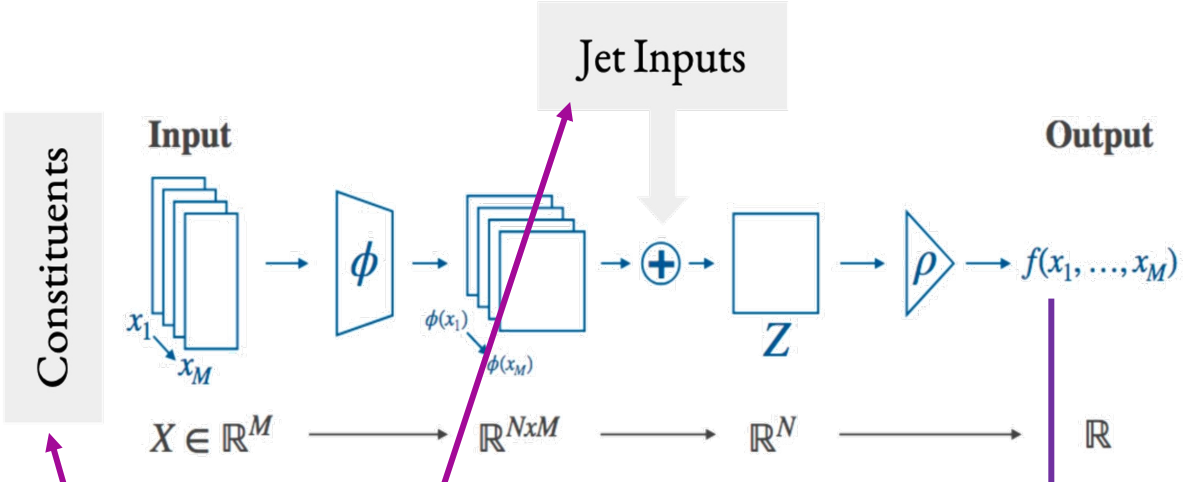
(figure from "Jet energy scale and resolution measured in proton-proton collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector", ATLAS collaboration, 2021)

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

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

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

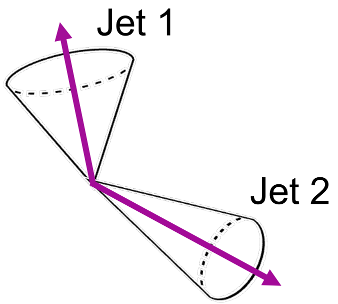


"Machine Learning the MC JES", K. Greif, C. Pollard, J. Roloff

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)

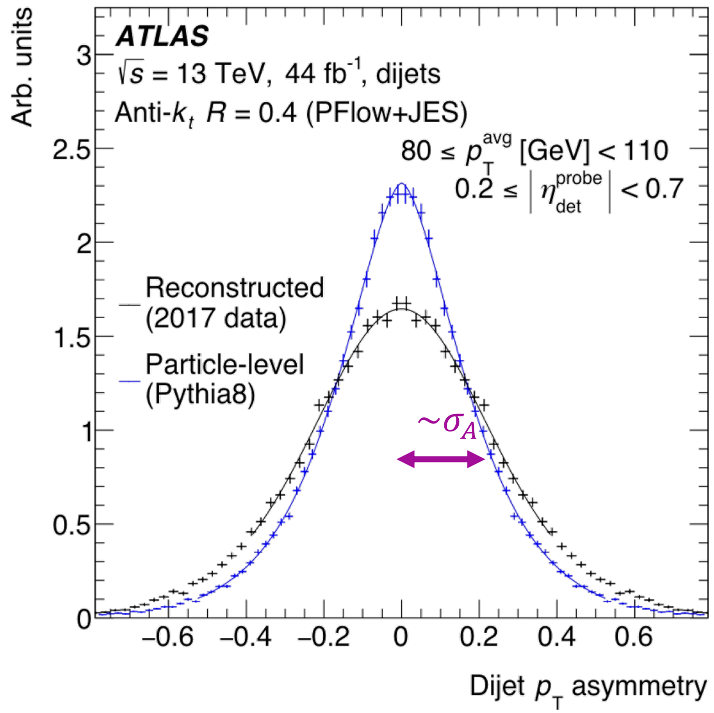
Dijet Events



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

- For events with at least two hard jets, 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 is randomly assigned to the two leading jets of every dijet event
  - Momentum conservation implies $\mathcal{A} = 0$ in ideal case (i.e. no noise, additional jets or other effects)
  - For experimental data, we observe distribution around 0 where the standard deviation (std) depends on reconstructed jet resolution (JER)



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

Minimising Jet Energy Resolution (JER)

- Relative JER can be estimated from $\sigma_{\mathcal{A}}$ (neglecting smearing from physics effects):[1] $\frac{\sigma_{p_T}}{p_T} = \frac{\sigma_{\mathcal{A}}^{det}}{\sqrt{2}} \cong \frac{\sigma_{\mathcal{A}}}{\sqrt{2}} \sim \sigma_{\mathcal{A}}$
  - Completely independent of true labels → useful for exp. data

- Update loss function:

$$\boxed{\text{loss}(\theta) = f_1 \cdot loss_G(\theta) + f_2 \cdot \sigma_{\mathcal{A}(\theta)}}$$

where $\sigma_{\mathcal{A}(\theta)}$ is the std of $\mathcal{A}(\theta)$

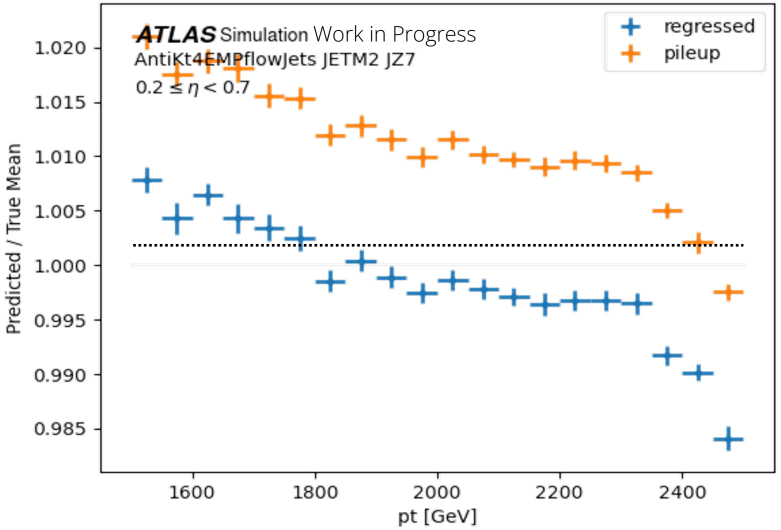
- ML model simultaneously minimises the JER measured in-situ and the original loss
- No longer fully dependent on truth level, ML model is only partially supervised

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

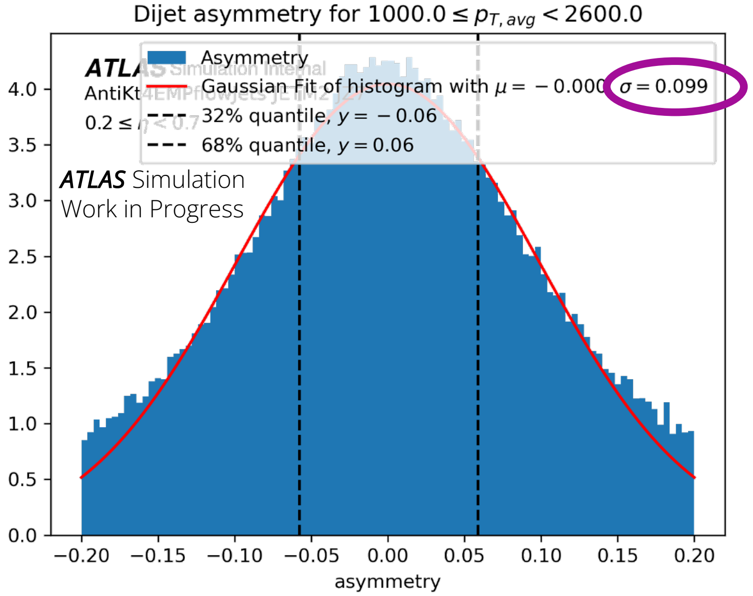
Results with $f_2 = 0$

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

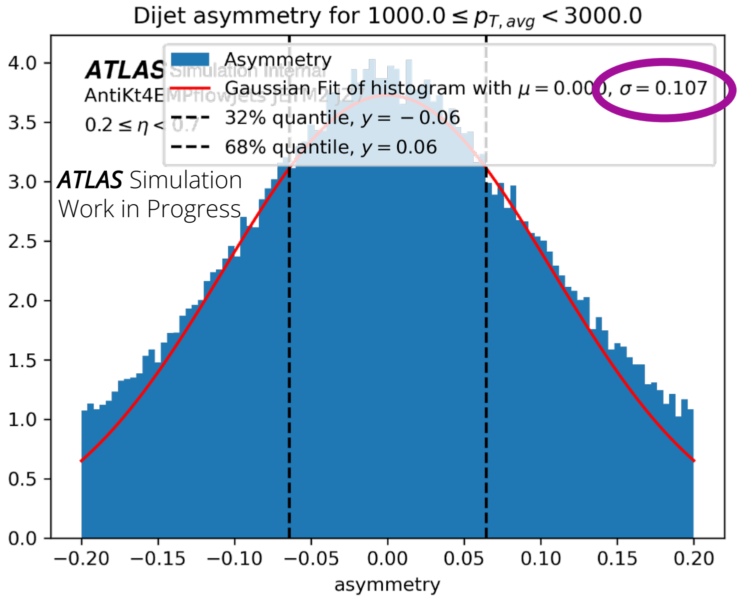
$$\text{loss}(\theta) = loss_G(\theta)$$



Testing set: reco jets



Testing set: regressed jets



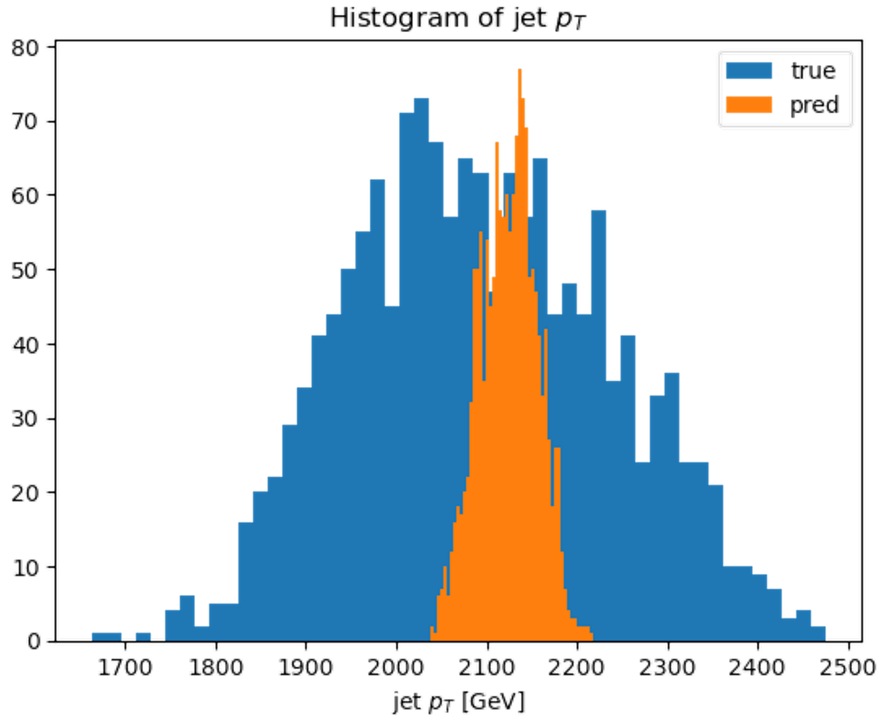
Challenges with $f_1 = 0$

- Trivial solution: model pushes all pT predictions towards one constant value which minimises std of asymmetry
- PROBLEM: very unphysical solution, we want the average jet pT to stay invariant

→Introduce constraints

- Possible constraints $C(\theta)$:
    - Keep batch mean invariant (predicted vs. initial pT)
    - Introduce bins in pT and keep bin mean invariant

$$\text{loss}(\theta) \longrightarrow f_1 \cdot loss_G(\theta) + f_2 \cdot \sigma_{\mathcal{A}(\theta)} + f_3 \cdot C(\theta)$$

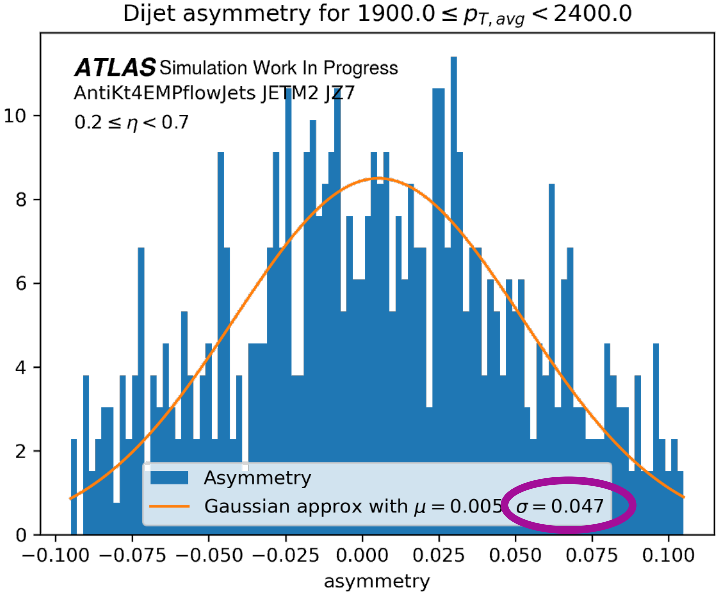


Results with $f_1 = 0$

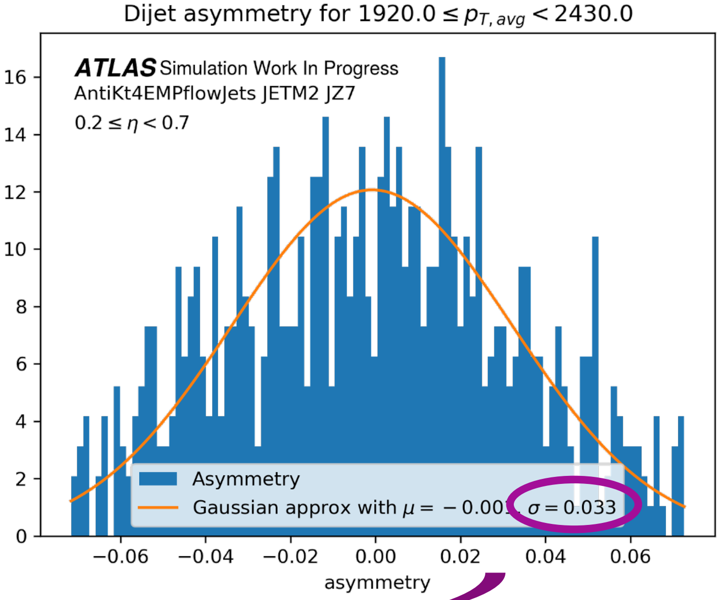
- With constraints for each p_T bin, the model's predictions start to look more physical:
  - $\sigma_{\mathcal{A}}$ (and JER) decrease noticeably
  - Predicted and initial jet p_T very similar (per bin)

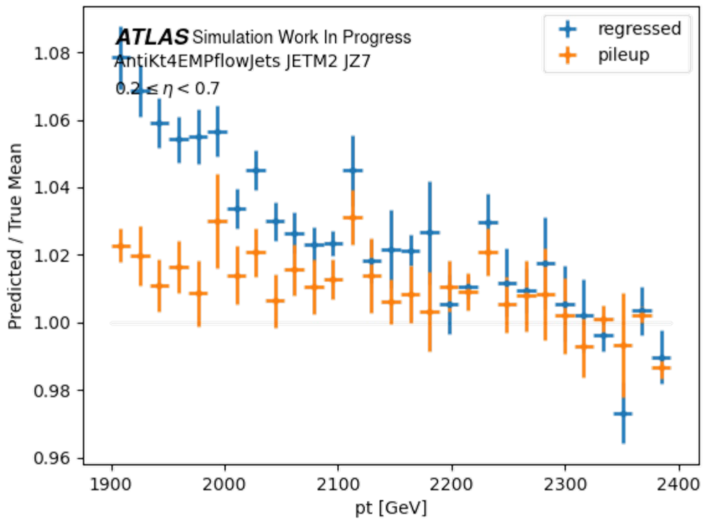
$$\text{loss}(\theta) = \sigma_{\mathcal{A}(\theta)} + 3 \cdot \mathrm{C}(\theta)$$

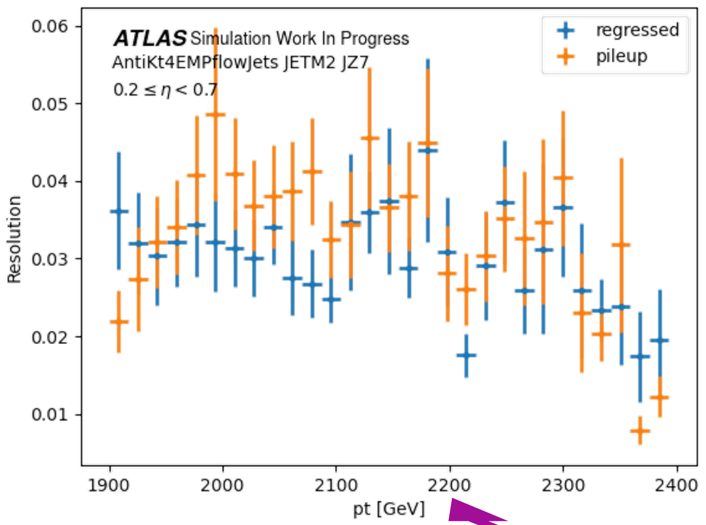
Testing set: reco jets



Testing set: regressed jets







Fraud Detection with IBM Research

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

Project with IBM: Fraud Detection

- Fraud detection in financial transactions
  - High input rate: ~1.5 billion of transactions / day
  - Highly imbalanced data: anomalies are very rare but should be correctly classified
  - Essential to understand/explain decisions of model
- New kind of frauds might appear → anomaly detection
- No data available for confidentiality reasons:
  - Develop anomaly detection methods for anomalous jet events
  - Adapt those methods to fraud detection

SMARTHEP
REAL-TIME ANALYSIS FOR SCIENCE AND INDUSTRY

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

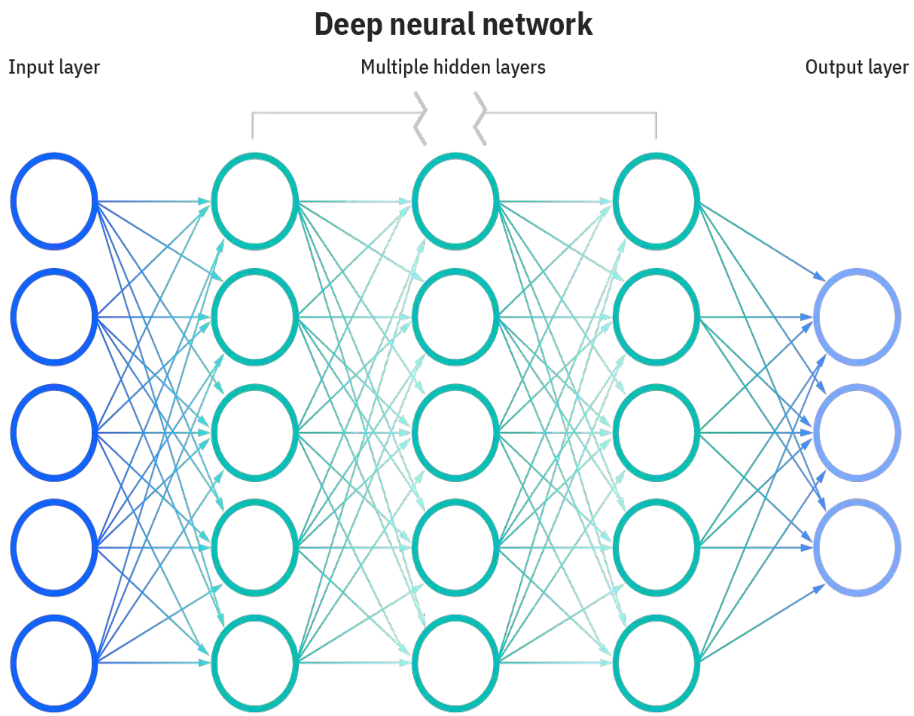
SMARTHEP is funded by the European Union's Horizon 2020 research and innovation programme, call H2020-MSCA-ITN-2020, under Grant Agreement n. 956086

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



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)

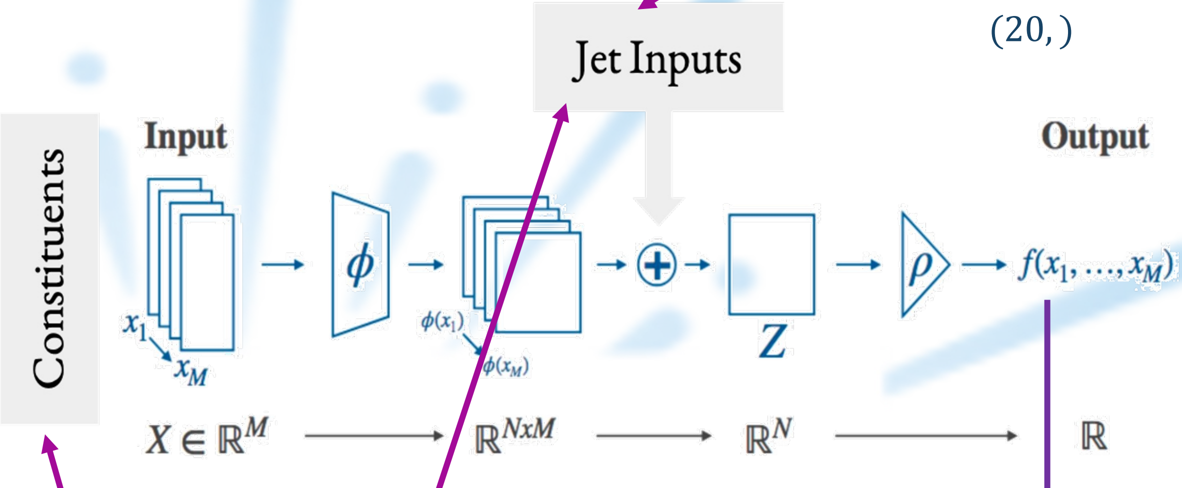
ML Model for Jet Calibration

GSC variables

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

$(20,)$

NEW!



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

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

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

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

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	$f_{\text{LAr}0-3*}$	The E_{frac} measured in the 0th-3rd layer of the EM LAr calorimeter
	$f_{\text{Tile}0*-2}$	The E_{frac} measured in the 0th-2nd layer of the hadronic tile calorimeter
	$f_{\text{HEC},0-3}$	The E_{frac} measured in the 0th-3rd layer of the hadronic end cap calorimeter
	$f_{\text{FCAL},0-2}$	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	$p_{\text{T}}^{\text{JES}}$ *	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 $p_{\text{T}} > 1$ GeV ghost-associated with the jet
	N_{track}*	The number of tracks with $p_{\text{T}} > 1$ 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

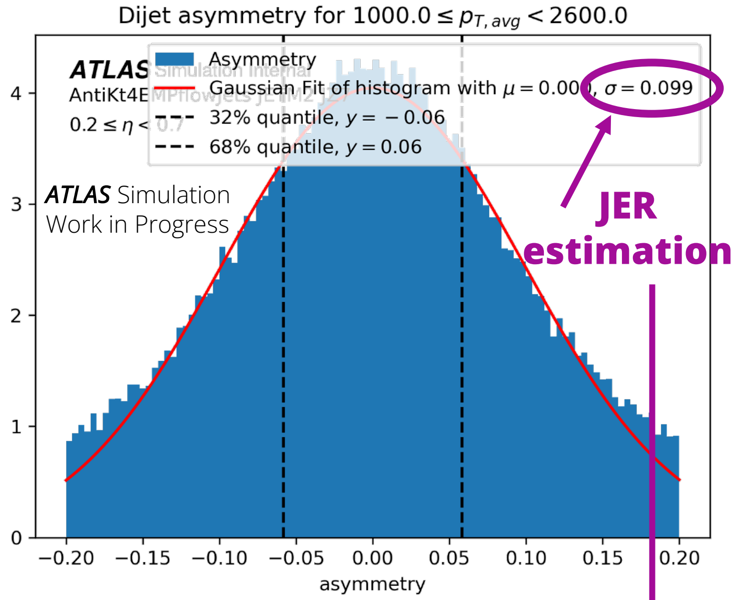
Table 1: List of variables used as input to the GNNC. Variables with a * correspond to those that are also used by the GSC.

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

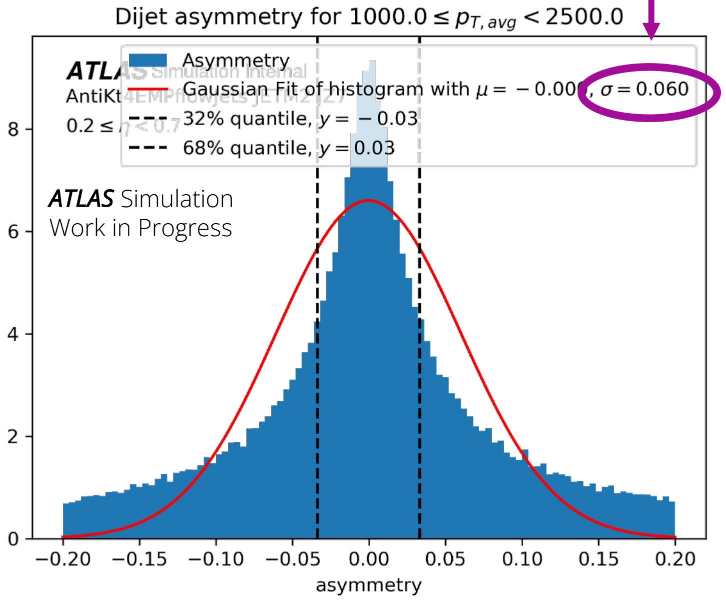
Dijet Asymmetry of JETM2 JZ7 (before Training)

- 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

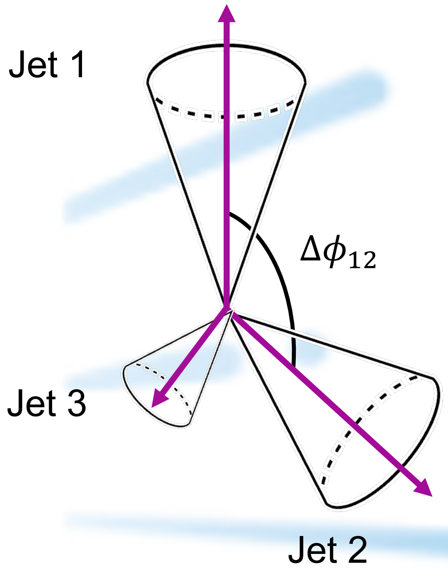


[1] ("Jet energy scale and resolution measured in proton-proton collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector", ATLAS collaboration, 2021)

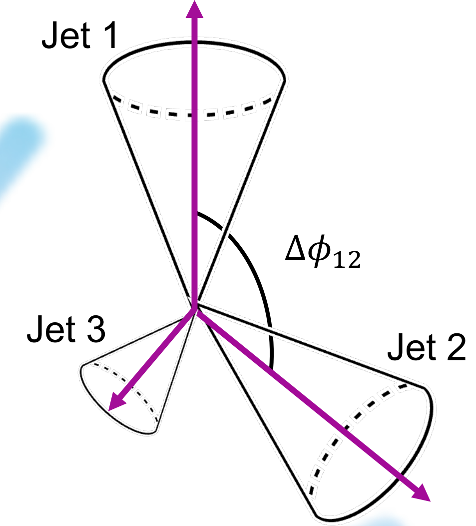
Input: MC Samples

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
- Modified input samples:
  - Keep event info of 3 leading jets
  - Empty entries are filled with same mask value
  - Additional features: GSC variables (22 add. Variables)
- Motivation: apply dijet topology cuts on jet components to ensure good p_T balance between leading jets



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

$$\Delta\phi_{12} > 2.7 \text{ rad}$$

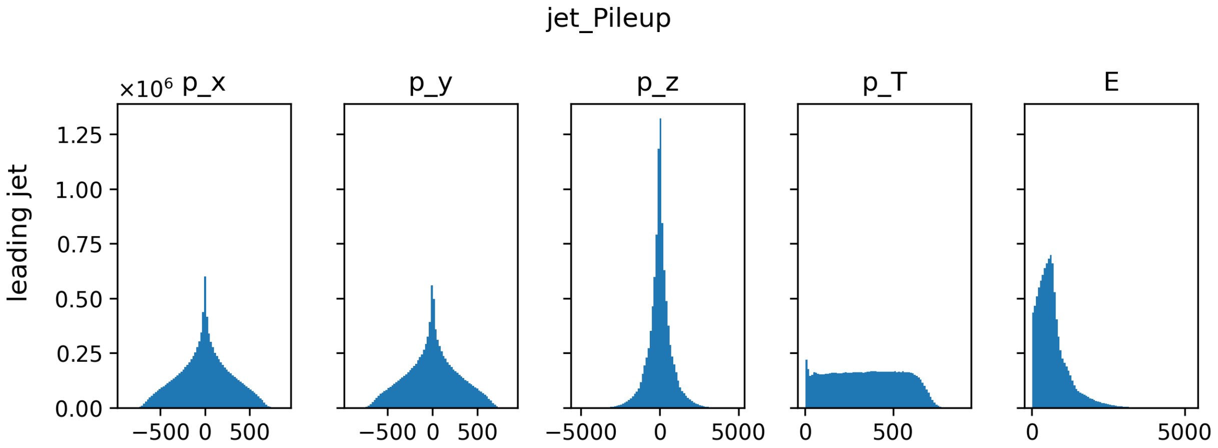
$$p_{T_3} < \max(25 \text{ GeV}, 0.25 \cdot p_{T,avg})$$

- pT between 1800 and 2400 GeV because using JZ7
  - Later add more JZ slices, e.g. study lower pT region
- Cut outliers (i.e. badly reconstructed jets)

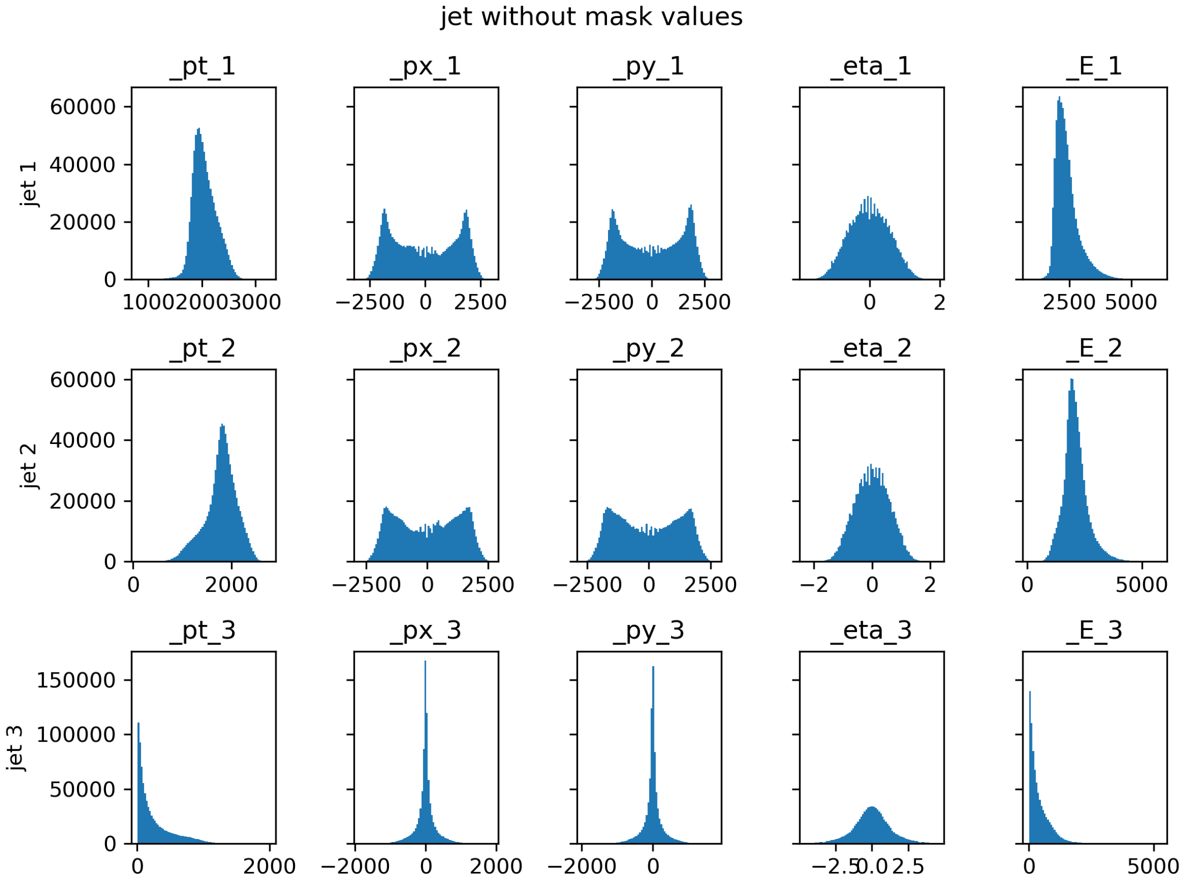
[1] ("Jet energy scale and resolution measured in proton-proton collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector", ATLAS collaboration, 2021)

Input: Jet Components

Old MC samples



New MC samples



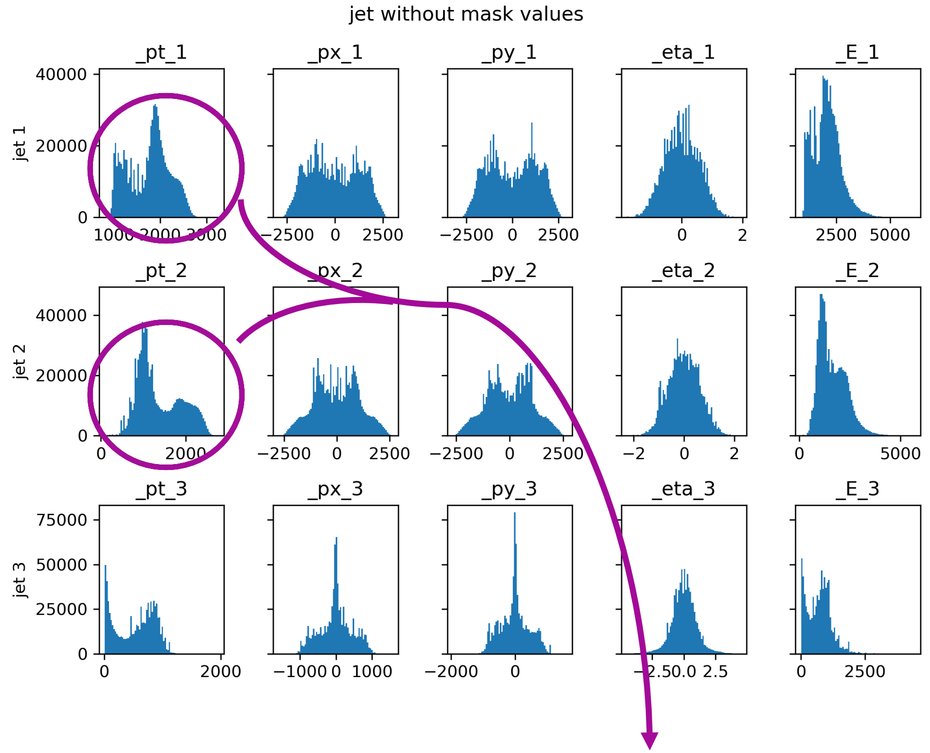
→ Note that p_T distribution on LHS has been flattened by resampling
→ On RHS no resampling/flattening

Produced from mc20a, JETM2, JZ7

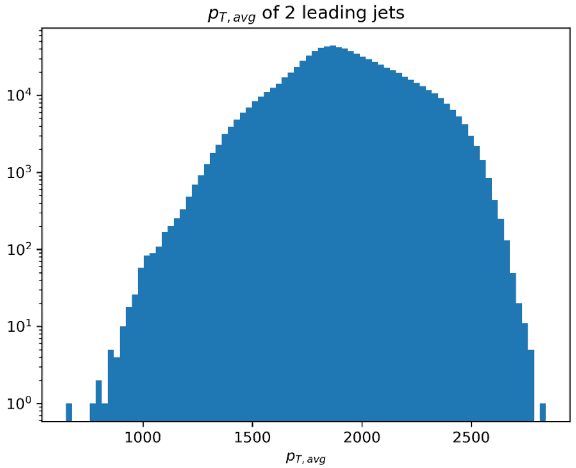
Input: Jet Components

- 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

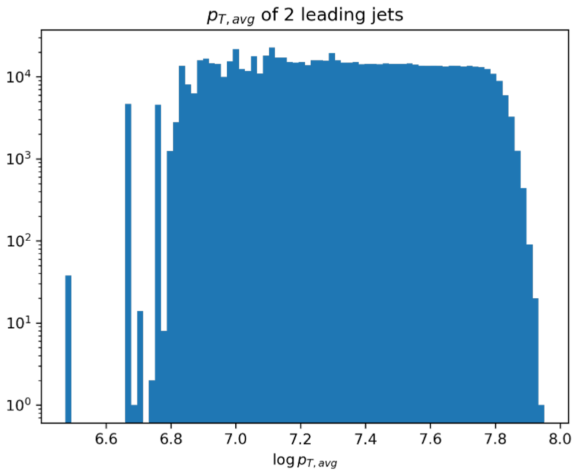
New MC samples: resampled



Before resampling



With resampling



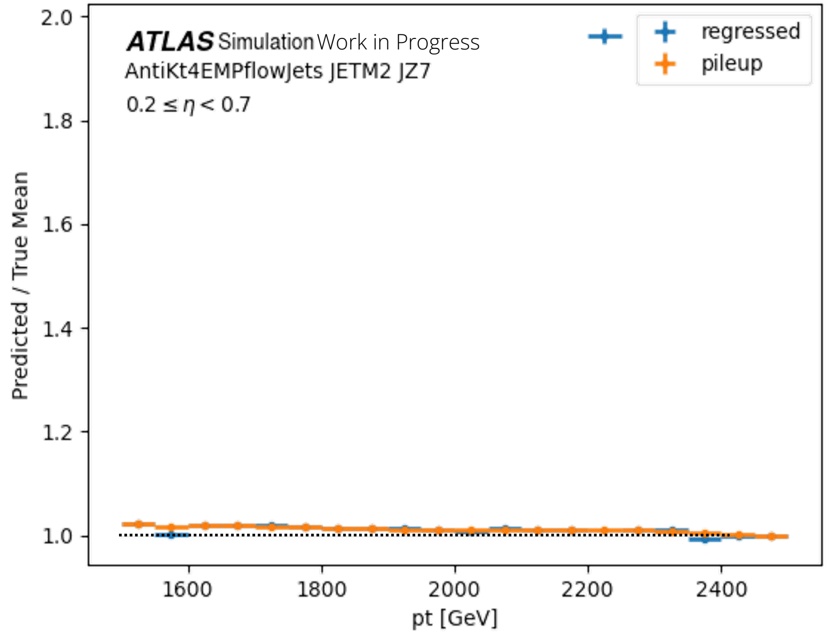
First results: $f = 0$ vs $f \neq 0$

$f = 0$	$f \neq 0$
• Asymmetry factor f is fixed to 0	• Asymmetry factor f is varied between 0 and 10
• Predicted pT values: • $p_T^{true} \in [1100, 2600]$ GeV • $p_T \in [1000, 3000]$ GeV	• 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): $\sim 10.7\ \%$	• 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!**

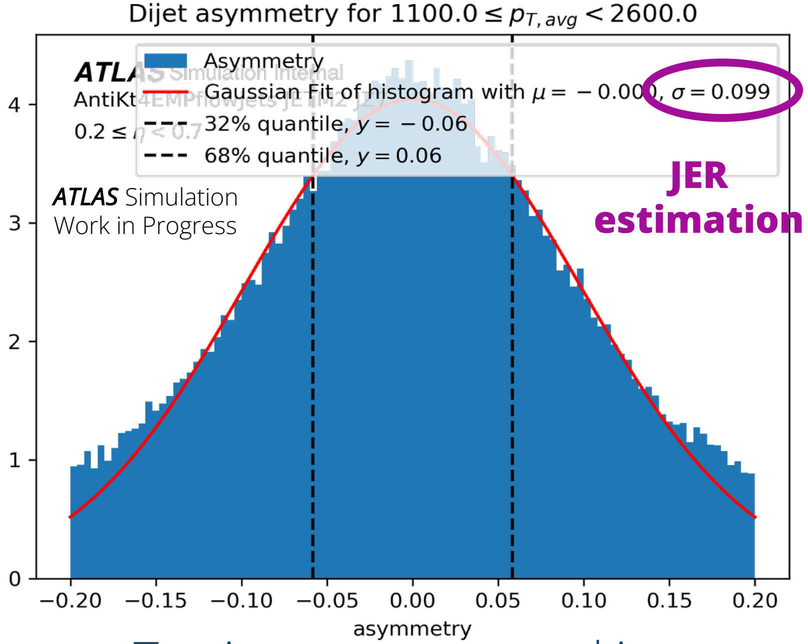
First Results with $f \neq 0$

- 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): $\sim 10.2\ \%$

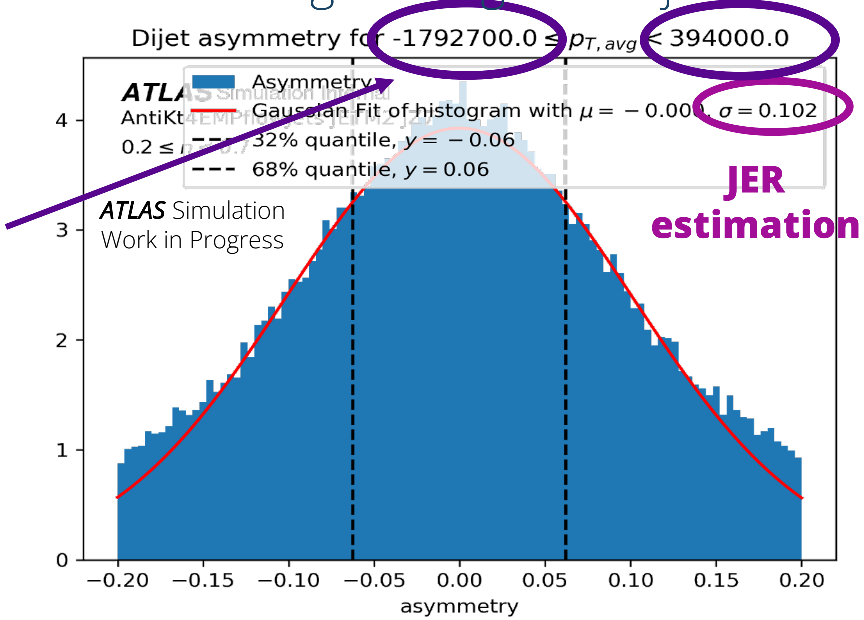


Problem: Why do we have negative calibration factors?

Testing set: reco jets



Testing set: regressed jets



What’s next

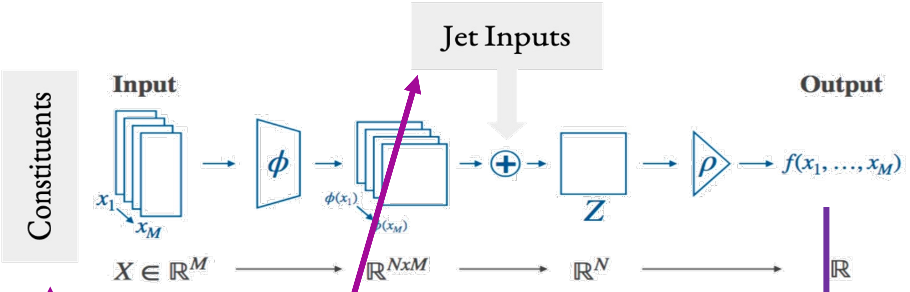
- Naive approach doesn’t work immediately
- It seems the two loss terms contradict/work against each other
    - Add softplus layer to restrict outputs of NN to positive values[1]
    - Introduce penalty term that forbids unphysical solution
    - Standardise truth targets
- Use **GSC variables**[2] (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.

$(20,)$



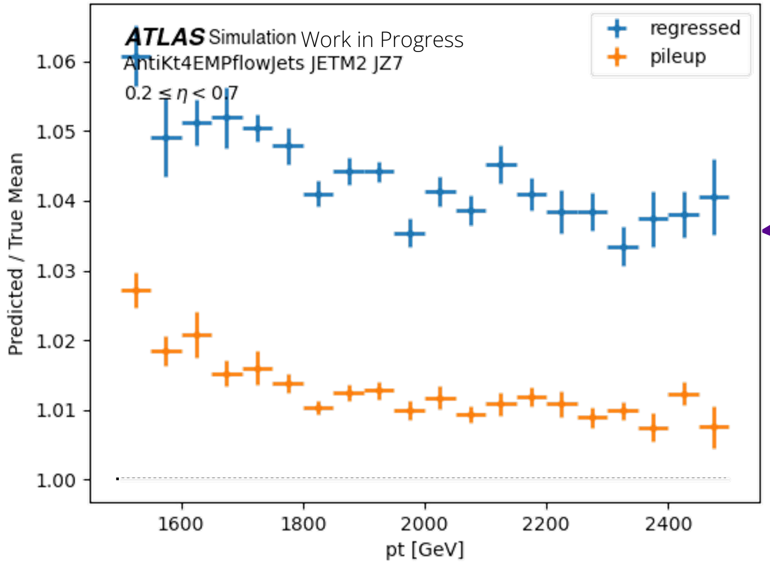
”Machine Learning the MC JES”, K. Greif, C. Pollard, J. Roloff

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)

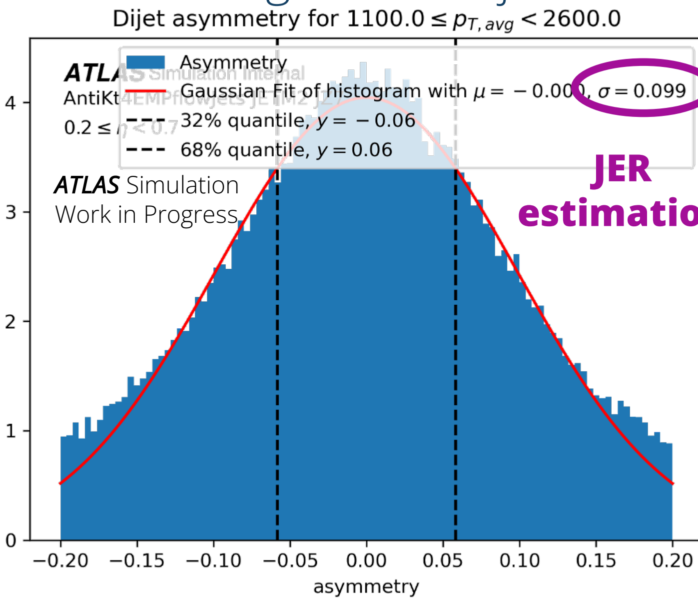
More results with $f \neq 0$

- 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): $\sim 12.7\ \%$



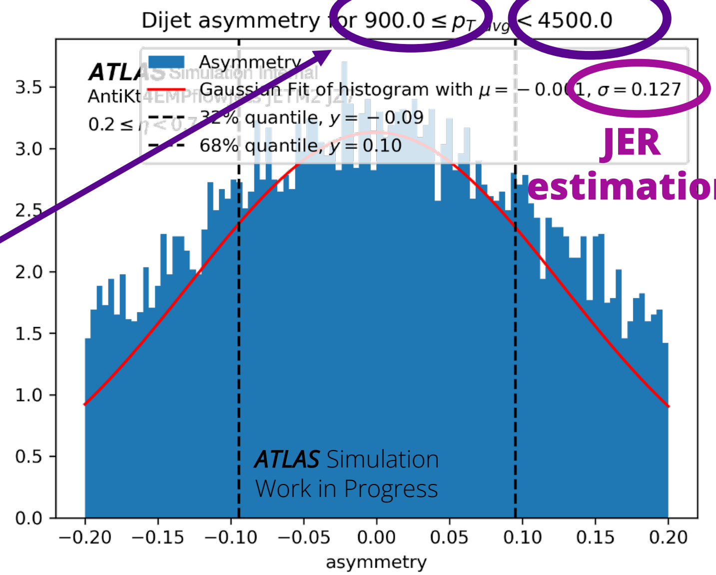
Problem: pT predictions are still off

Testing set: reco jets



JER estimation

Testing set: regressed jets



JER estimation

Explainable Machine Learning

- Despite success of Neural Networks (NN) their approach raises interpretability and explainability challenges
  - Very hard to understand how/why they reach a conclusion
  - For critical applications you cannot blindly trust NN models (e.g. fraud detection, medical decisions)
- Various approaches to make ML models more interpretable
  - Combine statistical with explainable models, e.g. rule induction