Machine Learning for Jet Calibration in ATLAS
Generalized Numerical Inversion

Aviv Cukierman¹, on behalf of the ATLAS Collaboration
November 15, 2018
Outline

• Overview of calibration in ATLAS
• Introduction to generalized numerical inversion
• Results from demonstration of method
Overview
Current strategy of calibration at ATLAS is sequential

Advantages:
- At each step, can check dependence of response
- Easy to understand

Disadvantages:
- Difficult to add in new information and test improvement
- Ignores correlations that could be treated with non-sequential calibration
- Unclear how to use ML for learning best way to use new information

Numerical inversion

Introduction: Sequential Calibration at ATLAS

(arXiv:1703.09665)
Have to calibrate jet $p_T$ to particle-level scale in ATLAS

First idea: calibration with *maximum likelihood*:
- Make list of $(p_T^{true}, p_T^{reco})$ pairs
- Train function $M(p_T^{reco})$ that minimizes loss in training
- *Not independent* of distribution of true energies used in training
  - Edge effects particularly egregious
- *Not* what we do in ATLAS right now
• Calibration with *numerical inversion*:  
  - Train on jets with known energies $x$, take output value with maximum likelihood: $f(x) = <p_T^{reco} | p_T^{true} = x>$  
  - Invert function for unknown observed values: $p_T^{reco} \rightarrow f^{-1}(p_T^{reco})$

- *Independent* of the distribution of true energies used in training  
  • Processes in LHC have different true energy distributions!  
  • Guarantees closure* when conditioning on $p_T^{true} = x$  
- *This is what we do in ATLAS right now for MCJES and GSC*  
- *Formal details: arXiv:1609.05195*
Numerical Inversion

- **Numerical inversion:**
  \[ f(x) = \langle p_{T}^{\text{reco}}|p_{T}^{\text{true}} = x \rangle \]
  \[ p_{T}^{\text{reco}} \mapsto C(p_{T}^{\text{reco}}) \equiv f^{-1}(p_{T}^{\text{reco}}) \]

- **Sequential numerical inversion (e.g., GSC):**
  \[ f_{\theta}(x) \equiv \langle p_{T}^{\text{reco}}|p_{T}^{\text{true}} = x, \theta \rangle \]
  \[ p_{T}^{\text{reco}} \mapsto f_{\theta_{n}}^{-1} \left( \cdots f_{\theta_{2}}^{-1} \left( f_{\theta_{1}}^{-1} \left( p_{T}^{\text{reco}} \right) \right) \cdots \right) \]
Generalized Numerical Inversion

- Generalized numerical inversion:
  \[ L(x; \theta) = \langle p_T^{\text{reco}} | p_T^{\text{true}} = x; \theta \rangle \]
  - Calibration: \( p_T^{\text{reco}} \rightarrow C(p_T^{\text{reco}}; \theta) = L^{-1}(p_T^{\text{reco}}; \theta) \)

- Advantages:
  - Still independent of underlying \( p_T \) distribution
  - Easy to add in other features, unbinned
  - Can take into account correlations between features

1. Learn a neural network approximation to the function \( L(x, \theta) = \langle p_T^{\text{reco}} | p_T^{\text{true}} = x, \theta \rangle \). Note that \( L(x, \theta) : \mathbb{R}^{n+1} \rightarrow \mathbb{R} \).

2. Learn a neural network \( C(L(x, \theta), \theta) \) that tries to predict \( x \) given \( \theta \) and \( L(x, \theta) \). This is an approximation to the family of functions \( f_{\theta}^{-1}(x) \). Note that learning the inverse this way is technically simple since \( L \) is single-valued.

3. Calibrate with \( p_T^{\text{reco}} \leftrightarrow C(p_T^{\text{reco}}, \theta) \). The calibration non-closure is given the deviation of \( C(p_T^{\text{reco}}, \theta|x) \) from \( x \).
Results
Results presented at BOOST 2018

- Pub Note
- Webpage
- Poster

Generalized Numerical Inversion – Final Results

In ATLAS, collimated sprays of hadrons initiated by quarks or gluons are reconstructed as jets. The reconstructed energy of jets is not exactly the same as the truth-level energy, leading to a non-trivial response. This response needs to be corrected for in a jet calibration, while also taking into account various jet features which may have an effect on the response. Importantly, this calibration should be independent of the underlying truth distribution of jets, in order to have an unbiased calibration.

Generalized Numerical Inversion

The current procedure in ATLAS for the jet calibration uses numerical inversion (NI), correcting for each feature $\theta$ in sequence:

Generalized numerical inversion generalizes NI, by learning the dependence of the response on each feature simultaneously:

The new idea allows for a multivariate regression approach to the jet calibration, which should allow for improved performance.

Results

We examine a calibration depending on $n_{\text{track}}$ and $\Delta R_{\text{track}}$:

The simultaneous calibration is able to correct for residual response dependencies that a sequential method is unable to account for:

By taking into account multiple variables, the simultaneous calibration can reduce the difference in response between quark- and gluon-initiated jets, and this effect is robust to model differences:

The response closes overall for both the simultaneous and the sequential calibrations and the resolution is reduced when taking into account the external variables:

Conclusions

Generalized numerical inversion is a new technique which allows for a regression-based jet calibration, avoids binning effects, and remains independent of the underlying jet distribution. The new method allows to correct for features simultaneously, which can correct the response dependence in all regions of the parameter space, reduce the difference between quark and gluon jets, and reduce model dependence. This method was demonstrated with neural networks, but more complicated architectures with more features should also be able to take advantage of this technique.
Generalized Numerical Inversion – Final Results

- Example usage: nTrack correction + $\Delta R_{\text{track,avg}}$ correction
  
  
  $\Delta R_{\text{track,avg}} \equiv \begin{cases} 
  \frac{1}{n_{\text{track}}+1} \sum_{\text{tracks}} (p_{T,\text{track}} / \sum_{\text{tracks}}' p_{T,\text{track}}') \times \Delta R_{\text{track,jet}} & \text{if } n_{\text{track}} > 0 \\
  -1 & \text{if } n_{\text{track}} = 0 
  \end{cases}$

- $\Delta R_{\text{track,avg}}$ chosen as a variable that contains information about width of jet

The GSC currently used in ATLAS uses instead the track width, which is the average track radius multiplied by $(n_{\text{track}}+1)$. The residual dependence of the $n_{\text{track}}$-calibrated response on the track width is negligible and is thus not useful for benchmarking generalized numerical inversion as the sequential calibration is nearly the same as the simultaneous calibration.

**Figure 1:** The dependence of the response on (a) $n_{\text{track}}$ and (b) $\Delta R_{\text{track,avg}}$ in several bins of truth jet $p_T$. 
Generalized Numerical Inversion – Final Results

• Sequential calibration (~unbinned GSC)

Figure 2: The dependence of the (a) learned response $L(p_T^{\text{true}}, \theta)/p_T^{\text{true}}$, (b) ratio of $p_T^{\text{reco}}$ to learned approximation $L(p_T^{\text{true}}, \theta)$, and (c) calibrated response $C(p_T^{\text{reco}}, \theta)/p_T^{\text{true}}$ on $n_{\text{track}}$ in several bins of truth jet $p_T$ for $\theta = \{n_{\text{track}}\}$. Also, the dependence of the (d) learned response $L(p_T^{\text{true}}, \theta)/p_T^{\text{true}}$, (e) ratio of $p_T^{\text{reco}}$ to learned approximation $L(p_T^{\text{true}}, \theta)$, and (f) calibrated response $C(p_T^{\text{reco}}, \theta)/p_T^{\text{true}}$ on $\Delta R_{\text{track,avg}}$ in several bins of truth jet $p_T$ for $\theta = \{\Delta R_{\text{track,avg}}\}$ in sequence after the $n_{\text{track}}$ correction.
Simultaneous calibration

Figure 3: The dependence of the (a,d) learned response $L(p_T^{\text{true}}, \theta)/p_T^{\text{true}}$, (b,e) ratio of $p_T^{\text{reco}}$ to learned approximation $L(p_T^{\text{true}}, \theta)$, and (c,f) calibrated response $C(p_T^{\text{reco}}, \theta)/p_T^{\text{true}}$ on (a,b,c) $n_{\text{track}}$ and (d,e,f) $\Delta R_{\text{track,avg}}$, respectively, in several bins of truth jet $p_T$ for a simultaneous calibration with $\theta = \{n_{\text{track}}, \Delta R_{\text{track,avg}}\}$. 
• Residual dependence
  - Simultaneous is flat at 0 – demonstration of method

Figure 4: The dependence of (a) $\frac{d\mathcal{R}}{dn_{\text{track}}}$ on $\Delta R_{\text{track,avg}}$ and (b) $\frac{d\mathcal{R}}{dn_{\text{track}}}$ on $n_{\text{track}}$ for: a calibration using a network with $\theta = \{n_{\text{track}}\}$ (circles); a calibration using a network with $\theta = \{\Delta R_{\text{track,avg}}\}$ employed sequentially after correcting for $n_{\text{track}}$ (squares); and a simultaneous calibration using a network with $\theta = \{n_{\text{track}}, \Delta R_{\text{track,avg}}\}$ (diamonds). Also, the closure as a function of $p_{\text{T}}^{\text{true}}$, highlighting the nonclosure due to this residual dependence in (c) a selection intended to target gluon jets; and (d) a selection intended to target quark jets.
Generalized Numerical Inversion – Final Results

• Quark-gluon response difference
  - Simultaneous is slightly better

Figure 5: The difference between the response of quarks and gluons as a function of $p_T^{\text{true}}$ for: before any $n_{\text{track}}$ or $\Delta R_{\text{track,avg}}$ correction (open circles); a calibration using a network with $\theta = \{n_{\text{track}}\}$ (circles); a calibration using a network with $\theta = \{\Delta R_{\text{track,avg}}\}$ sequentially after correcting for $n_{\text{track}}$ (squares); and a simultaneous calibration using a network with $\theta = \{n_{\text{track}}, \Delta R_{\text{track,avg}}\}$ (diamonds).
Generalized Numerical Inversion – Final Results

- Difference between generators
  - Simultaneous is slightly better

---

Figure 6: The difference between the response of jets in Pythia8 and Herwig7 as a function of $p_T^{true}$ in (a) gluon jets, and (b) quark jets, for: before any $n_{track}$ or $\Delta R_{track,avg}$ correction (open circles); a calibration using a network with $\theta = \{n_{track}\}$ (circles); a calibration using a network with $\theta = \{\Delta R_{track,avg}\}$ sequentially after correcting for $n_{track}$ (squares); and a simultaneous calibration using a network with $\theta = \{n_{track}, \Delta R_{track,avg}\}$ (diamonds).
Generalized Numerical Inversion – Final Results

- Overall closure and resolution
  - Everything mostly the same

Figure 7: The (a) closure and (b) resolution as a function of $p_T^{\text{true}}$ for: before any $n_{\text{track}}$ or $\Delta R_{\text{track,avg}}$ correction (open circles); a calibration using a network with $\theta = \{n_{\text{track}}\}$ (circles); a calibration using a network with $\theta = \{\Delta R_{\text{track,avg}}\}$ sequentially after correcting for $n_{\text{track}}$ (squares); and a simultaneous calibration using a network with $\theta = \{n_{\text{track}}, \Delta R_{\text{track,avg}}\}$ (diamonds). For the resolution, also shown is the (negative) improvement in quadrature of the resolution for a given calibration with resolution $\sigma'$ to the resolution before any correction $\sigma$. 
Summary: Generalized Numerical Inversion

• Summary:
  - New technique, allows for a regression-based jet calibration
  - Avoids binning effects
  - Remains independent of the underlying jet distribution
  - Allows to correct for features simultaneously

• Benefits:
  - Correct the response dependence in all regions of parameter space
  - Reduce model dependence
  - Reduce the difference between quark and gluon jets

• Future:
  - More complicated architectures
    • Jet substructure, jet images
  - More features
    • Pileup suppression, track jets, b-jet calibration, etc.
Backup
Introduction: Jets and Calibration

- At LHC, quarks and gluons produced from proton-proton collisions
  - Hadronize and form jets in calorimeter

- Reconstructed energy is not the same, on average, as truth energy of originating particle
  - Energy must be calibrated
• Train function $M(p_T^{reco})$ that minimizes loss in training
• Not independent of distribution of true energies used in training

Edge effect (network learns $p_T>20$ GeV always) -> Nonclosure!

Learning prior effect (network learns lower $p_T$s are more likely than higher $p_T$s) -> Nonclosure!
Learning the Prior – Uniform pT Distribution

- Train function $M(p_T^{\text{reco}})$ that minimizes loss in training
- Proposed solution: Use uniform underlying distribution to mitigate dependence on prior
- Can show leads to closure (away from edges) if:
  - Uniform underlying distribution
  - Linear $f(x) = \mathbb{E}[p_T^{\text{reco}}|p_T^{\text{true}}=x]$ \(\times\)
  - Constant $\sigma(x) = \sigma[p_T^{\text{reco}}|p_T^{\text{true}}=x]$ \(\times\)
- If any assumption is violated, then closure is not necessarily achieved!
• Train function $M(p_T^{\text{reco}})$ that minimizes loss in training
• Uniform distribution, but $f(x)$ not linear!

Edge effect (network learns $p_T>20$ GeV always) -> Nonclosure!

Learning prior effect (network learns all $p_T$s are equally likely away from edges) -> Nonclosure because $f(x)$ not linear
Learning the Prior – Uniform pT Distribution

- Train function $M(p_T^{\text{reco}})$ that minimizes loss in training
- Proposed solution: Use uniform underlying distribution

Edge effect (network learns $p_T>20$ GeV always) -> Nonclosure!

Learning prior effect (network learns all pTs are equally likely away from edges) -> Closure?

[Graphs showing data and M(p_T^{reco}) against true pT values]
Learning the Prior – Uniform pT Distribution

- Train function $M(p_T^{\text{reco}})$ that minimizes loss in training
- Uniform distribution, $f(x)$ linear, but $\sigma(x)$ not constant!

Edge effect (network learns $p_T>20$ GeV always) -> Nonclosure!

Learning prior effect (network learns all pTs are equally likely away from edges) -> Nonclosure because $\sigma(x)$ not constant
Train function \( M(p_T^{\text{reco}}) \) that minimizes loss in training

- Can show leads to closure (away from edges) if:
  - Uniform underlying distribution
  - Linear \( f(x) = E[p_T^{\text{reco}}|p_T^{\text{true}}=x] \)
    - GSC is done after MCJES, so \( f(x) \) is in principle linear
    - Doesn’t work for doing MCJES all the way through
  - Constant \( \sigma(x) = \sigma[p_T^{\text{reco}}|p_T^{\text{true}}=x] \)
    - N.B. If any assumption is violated, then closure is not necessarily achieved!

- Proposed solution: Do residual numerical inversion after to correct
  - N.B.: If looking at GSC, then steps are MCJES -> (ML) GSC -> MCJES again
  - N.B. If given an already calibrated collection, learning the prior uncalibrates it
Learning the Prior – Uniform pT Distribution

- Train function $M(p_T^{reco})$ that minimizes loss in training
- Do residual numerical inversion after to correct

Edge effect
(network learns $p_T>20$ GeV always)
-> Nonclosure!

Fit to averages: $f'(x) \to 0$
-> Numerical inversion fails!
($f^{-1}(20$ GeV) does not exist)
Train function $M(p_T^{reco})$ that minimizes loss in training

Can show leads to closure (away from edges) if:
- Uniform underlying distribution
- Linear $f(x) = E[p_T^{reco}|p_T^{true}=x]$ 
  - GSC is done after MCJES, so $f(x)$ is in principle linear
  - Doesn’t work for doing MCJES all the way through
- Constant $\sigma(x) = \sigma[p_T^{reco}|p_T^{true}=x]$ 
  - If any assumption is violated, then closure is not achieved!

Proposed solution: Generalized numerical inversion
Calibration: Other Features

- Numerical inversion:
  - $f(x) = <p_T^{reco}|p_T^{true}=x>$
  - Calibration: $p_T^{reco} \rightarrow f^1(p_T^{reco})$
  - Want to add in other features $\theta$
  - Bin in $\theta$: $f_1(x) = <p_T^{reco}|p_T^{true}=x, \theta=\theta_1>$, $f_2(x) = <p_T^{reco}|p_T^{true}=x, \theta=\theta_2>$, …
    - Biases from binning; N times resources for each new feature; etc.
- Sequential calibration:
  - Correct for dependence on $\theta$ over all jets first, then proceed as normal
  - Used in ATLAS:

![Diagram of jet calibration process](image-url)

**Numerical inversion**

- EM-scale jets: Jet finding applied to topological clusters at the EM scale.
- Origin correction: Changes the jet direction to point to the hard-scatter vertex. Does not affect $E$.
- Jet area-based pile-up correction: Applied as a function of event pile-up $p_T$ density and jet area.
- Residual pile-up correction: Removes residual pile-up dependence, as a function of $\mu$ and $N_{PV}$.
- Absolute MC-based calibration: Corrects jet 4-momentum to the particle-level energy scale. Both the energy and direction are calibrated.
- Global sequential calibration: Reduces flavor dependence and energy leakage effects using calorimeter, track, and muon-segment variables.
- Residual in situ calibration: A residual calibration is derived using in situ measurements and is applied only to data.
Generalized Numerical Inversion - Toy Example (0 features)

- Toy example: Just energy calibration
  - Will add in more features later
  - Same model for response as used in arXiv:1609.05195
Generalized Numerical Inversion - Toy Example (0 features)

- Toy example: Just energy calibration
  - Will add in more features later
  - Simple nonlinear response
- $L(x; \theta) = \langle p_{T}^{\text{reco}} | p_{T}^{\text{true}} = x; \theta \rangle$
  - Use fact that NN with 1 hidden layer can learn arbitrary function
Generalized Numerical Inversion - Toy Example (0 features)

- Calibration: $p_T^{\text{reco}} \rightarrow C(p_T^{\text{reco}}; \theta) = L^{-1}(p_T^{\text{reco}}; \theta)$
- Technical challenge:
  - $C$ only learns in range $[\min(L(p_T^{\text{true}}; \theta)), \max(L(p_T^{\text{true}}; \theta))]$
  - $C$ has to extrapolate to values outside this range
- Solution:
  - Do linear extrapolation (for given $\theta$) above and below range learned on
Generalized Numerical Inversion - Toy Example (0 features)

- Calibration: $p_T^{\text{reco}} \rightarrow C(p_T^{\text{reco}}; \theta) = L^{-1}(p_T^{\text{reco}}; \theta)$
- Test ”Closure” ($C(L(p_T^{\text{true}}; \theta))/p_T^{\text{true}}$)
  - Perfect! (depends on architecture)
- Test Closure ($<C(p_T^{\text{reco}}; \theta)/p_T^{\text{true}}>)$
  - Looks good!
  - Nonclosure –may result from necessary nonclosure in NI
Generalized Numerical Inversion - Inversion

- Calibration: $p_T^{reco} \rightarrow C(p_T^{reco};\theta) = L^{-1}(p_T^{reco};\theta)$
- How to invert?
  - Can invert numerically
    - Resource intensive for each new value of feature $\theta$
  - Better: Use fact that NN with 1 hidden layer can learn arbitrary function
    - Train with $X=(L(p_T^{true};\theta),\theta)$, $Y=p_T^{true}$
    - Since $L$ is single-valued, avoid problem of dependence on prior

![Graph showing $L(p_T^{true})$ vs $p_T^{true}$ and $L(p_T^{true})$ vs $C(L(p_T^{true}))$]