Update: Testing unfolding of jets from e-p collisions

Tanmay Pani

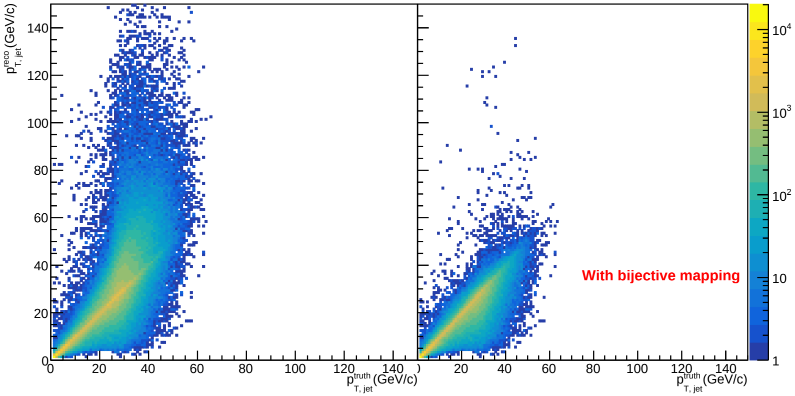
Rutgers University

June 26, 2024

Recap

- Took 1031 files from 23.12.0 campaign at:
  - root://dtneic.jlab.org/ /work/eic2/EPIC/RECO/23.12.0/epic_craterlake/DIS/NC
  - collsion energies: 18×275
  - $\min(Q^2) = 1000$ GeV
  - file suffix : tree.edm4eic.root
- Jets were clustered (anti-k_T, E-scheme, $R = 1.0$) from branches "ReconstructedParticles" for reco level and "MCParticles" branch for truth level
  - $E_{jet} > 5$ GeV
  - $|\eta_{jet}| < 2.5$
  - only for Reco jets $\Delta R(\mathrm{jet}, \mathrm{e}^-_{\mathrm{beam}}) > 1.0$
- Truth and Reco jets are matched using a proximity criteria in $\eta - \phi$ plane ($\Delta R < 0.2$)
- Added an extra criteria requiring the matching to be a bijective mapping between truth and reco level (matching truth to reco should give the same pair as matching reco to truth) to weed out fakes

Recap - $p_{\mathrm{T,jet}}$ response



Generalized angularities

$$\lambda_\beta^\kappa = \sum_{\text{const}\in\text{jet}} \overbrace{\left(\frac{p_{\text{T,const}}}{p_{\text{T,jet}}}\right)^\kappa}^{\text{soft/hard radiation}} \times \overbrace{r(\text{const},\text{jet})^\beta}^{\text{collinearity sensitive}}$$

$$r(\text{const},\text{jet}) = \sqrt{(\eta_{\text{jet}} - \eta_{\text{const}})^2 + (\phi_{\text{jet}} - \phi_{\text{const}})^2}$$

$\lambda_\beta^1 \rightarrow$ Infra-red and collinear (IRC) safe angularities

Generalized angularities

$$\lambda_\beta^\kappa = \sum_{\text{const}\in\text{jet}} \overbrace{\left(\frac{p_{\mathrm{T,const}}}{p_{\mathrm{T,jet}}}\right)^\kappa}^{\text{soft/hard radiation}} \times \overbrace{r(\text{const}, \text{jet})^\beta}^{\text{collinearity sensitive}}$$

$$r(\text{const}, \text{jet}) = \sqrt{(\eta_{\text{jet}} - \eta_{\text{const}})^2 + (\phi_{\text{jet}} - \phi_{\text{const}})^2}$$

$\langle \text{Radiation} \rangle_{\text{gluon jets}} > \langle \text{Radiation} \rangle_{\text{quark jets}}$

$\implies \langle \lambda^1_{\beta>0} \rangle_{\text{gluon jets}} > \langle \lambda^1_{\beta>0} \rangle_{\text{quark jets}}$

$\implies$ quark-gluon discrimination

$\lambda_\beta^1 \rightarrow$ Infra-red and collinear (IRC) safe angularities

Generalized angularities

$$\lambda_\beta^\kappa = \sum_{\text{const}\in\text{jet}} \overbrace{\left(\frac{p_{\mathrm{T,const}}}{p_{\mathrm{T,jet}}}\right)^{\kappa}}^{\text{soft/hard radiation}} \times \overbrace{r(\text{const},\text{jet})^{\beta}}^{\text{collinearity sensitive}}$$

$$r(\text{const},\text{jet}) = \sqrt{(\eta_{\text{jet}} - \eta_{\text{const}})^2 + (\phi_{\text{jet}} - \phi_{\text{const}})^2}$$

$\langle\text{Radiation}\rangle_{\text{gluon jets}} > \langle\text{Radiation}\rangle_{\text{quark jets}}$

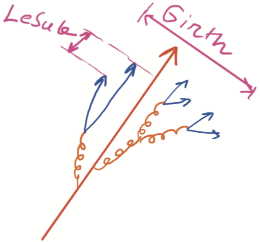
$\implies \langle\lambda^1_{\beta>0}\rangle_{\text{gluon jets}} > \langle\lambda^1_{\beta>0}\rangle_{\text{quark jets}}$

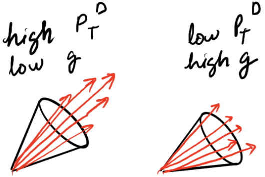
$\implies$ quark-gluon discrimination

- **Jet girth:** $\mathbf{g} = \lambda_1^1 = \frac{\sum_{\text{trk}\in\text{jet}} p_{\mathrm{T,trk}} \Delta R}{p_{\mathrm{T,jet}}}$, measure of jet broadening
- **Momentum dispersion :** $\mathbf{p_T^D} = \frac{\sqrt{\sum_{\text{trk}\in\text{jet}} (p_{\mathrm{T,trk}})^2}}{\sum_{\text{trk}\in\text{jet}} p_{\mathrm{T,trk}}}$

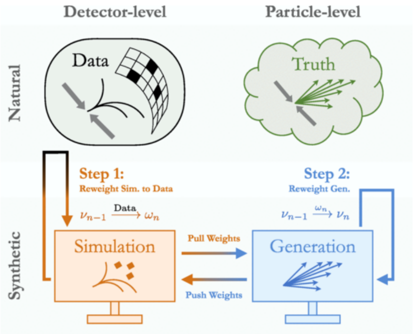
  soft/hard fragmentation $\implies$ low/high p_T^D
- **LeSub** $= p^{\text{Leading}}_{\mathrm{T,const}} - p^{\text{Subleading}}_{\mathrm{T,const}}$, proxy for hardest splitting in jet

$\lambda^1_\beta \rightarrow$ Infra-red and collinear (IRC) safe angularities





MultiFold



- Removing residual background and detector effects by mapping RECO → GEN using embedding simulation
- Simultaneously unfolding $p_{\mathrm{T,jet}}$,η_{jet}, ϕ_{jet}, $N_{contituents}$, p_T^D, LeSub and Girth through Multifolding (Phys. Rev. Lett. 124, 182001)
- Multifolding uses Dense Neural Networks (DNNs) trained on full embedding sample at the detector level and the generator level
- DNNs were implemented using PyTorch

Details

- Model architecture:

```
(0): Linear(in_features=7, out_features=100, bias=True)
(1): ReLU()
(2): Linear(in_features=100, out_features=100, bias=True)
(3): ReLU()
(4): Linear(in_features=100, out_features=100, bias=True)
(5): ReLU()
(6): Linear(in_features=100, out_features=2, bias=True)
```

- Loss function: Categorical cross entropy
- Optimizer: Adam(learning_rate = 0.001)
- 20% of the entire dataset extracted. For this set, reco level acts as a stand in for data and the gen level acts as a stand in for truth
- the remaining 80% is used only for training the neural networks

Method

- detector level NN classifies between training-reco and data (test-reco here) and assigns every jet a probability of being data (p_{data}). These probabilities are used to scale the weights of each training jets for the next step as

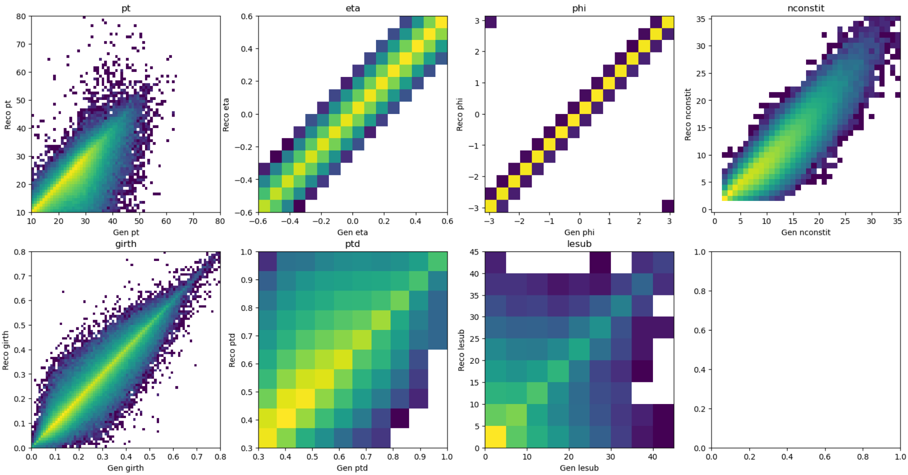
$$w_{jet} = w^0_{jet} \times \frac{p_{data}}{1 - p_{data}} \tag{1}$$

- generator level NN now classifies between reweighted jets (with weights w_{jet}) and the un-reweighted jets (with weights w^0_{jet}) and then assigns each jet a probability of being reweighted (p_{rew}). Weights are changed as,

$$w^0_{jet} = w_{jet} \times \frac{p_{rew}}{1 - p_{rew}} \tag{2}$$

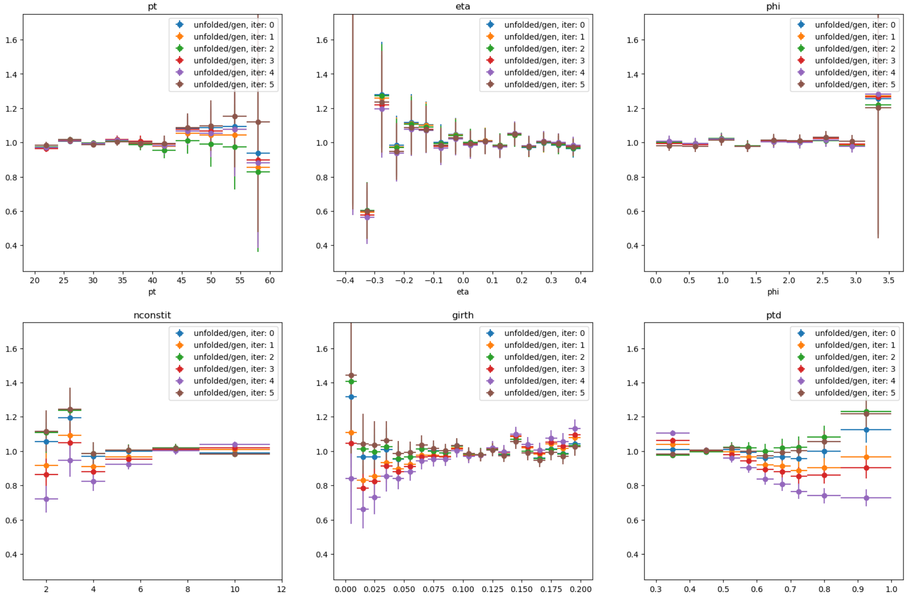
After certain iterations, the detector level NN should not be able to distinguish between reco and data jets. At that point unfolded distributions can be obtained by histograms with reweighted jet weights (h- Fill(jet, w^0_{jet}))

Responses

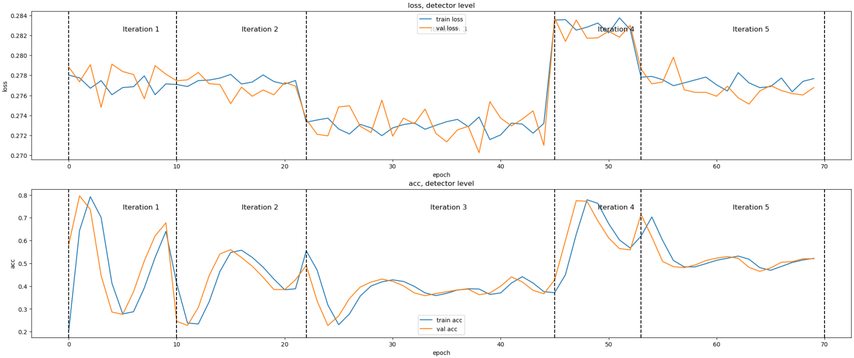


pt
Gen pt
Reco pt
eta
Gen eta
Reco eta
phi
Gen phi
Reco phi
nconstit
Gen nconstit
Reco nconstit
girth
Gen girth
Reco girth
ptd
Gen ptd
Reco ptd
lesub
Gen lesub
Reco lesub

Closure


pt
eta
phi
nconstit
girth
ptd
unfolded/gen, iter: 0
unfolded/gen, iter: 1
unfolded/gen, iter: 2
unfolded/gen, iter: 3
unfolded/gen, iter: 4
unfolded/gen, iter: 5
0.4
0.6
0.8
1.0
1.2
1.4
1.6
20 25 30 35 40 45 50 55 60
−0.4 −0.3 −0.2 −0.1 0.0 0.1 0.2 0.3 0.4
0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5
2 4 6 8 10 12
0.000 0.025 0.050 0.075 0.100 0.125 0.150 0.175 0.200
0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0

Training - detector level



Training - generator level