Deeply Learning Deep Inelastic Scattering Kinematics

Abdullah Farhat[1,3]

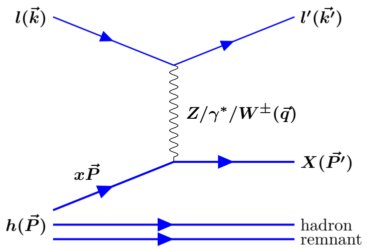
In collaboration with Yuesheng Xu[3], Markus Diefenthaler[1,2] and Andrii Verbytskyi[4]

1 EIC² EIC Center at Jefferson Lab 2 Jefferson Lab 3 Old Dominion University 4 Max-Planck-Institut für Physik (Werner-Heisenberg-Institut)

October 8, 2021

Physics case

- The studies of deep inelastic scattering (DIS) in the lepton-nucleon collisions give insight into the structure of nucleons.



$$l(\vec{k}) + h(\vec{P}) \to l'(\vec{k'}) + X(\vec{P'}),$$

- Reconstruction of collision kinematics is the key component.
- Measurements of the scattered lepton, hadronic final state and knowledge of the beam energies **over-constrain the reconstruction of event kinematics.**

Physics case

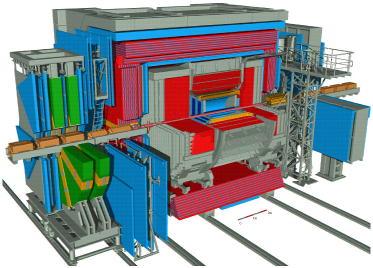
There are multiple **classical methods** aimed to reconstruct the DIS event kinematics each considering **partial information from collision event**, and being a subject to limitations related to

- bias related to QED radiation
- the need for precise measurements of the hadronic final state
- the need to measure the kinematics of the scattered lepton
- uneven performance across the desired kinematic region

To some extent, the **choice** of the reconstruction method can determine the size of systematic uncertainty.

Studies with ZEUS simulated data

We concentrate on the neutral current DIS events, i.e. those with an electron in the final state and utilize the **simulated** data of ZEUS experiment and reconstruct the four-momentum transferred to the hadronic system, $Q^2 = -\vec{q} \cdot \vec{q} = -(\vec{k} - \vec{k'})^2$ and the Bjorken scaling variable $x = \frac{Q^2}{2\vec{P} \cdot \vec{q}}$.



- Simulated and real data is available for analyses.
- Data preservation efforts led to convenient and accessible data samples.
- The documentation was appropriate to start an analysis of the ZEUS data.

Classical Reconstruction Methods

The kinematics of a DIS event has to be reconstructed by measurements related to the scattered lepton or the final state hadronic system ($\mathcal{H}$).

- the energy ($E_{l'}$) and polar angle ($\theta_{l'}$) of the scattered lepton
- the energy from the hadronic system in terms of:

$$\delta_{\mathcal{H}} = \textstyle\sum_{i\in\mathcal{H}} E_i - P_{Z,i} \qquad \& \qquad P_{T,\mathcal{H}} = \sqrt{\left(\textstyle\sum_{i\in\mathcal{H}} P_{X,i}\right)^2 + \left(\textstyle\sum_{i\in\mathcal{H}} P_{Y,i}\right)^2}$$

The hadronic energy flow is characterized by the angle $\gamma_{\mathcal{H}}$, where

$$\cos\gamma_{\mathcal{H}} = \frac{P_{T,\mathcal{H}}^2 - \delta_{\mathcal{H}}^2}{P_{T,\mathcal{H}}^2 + \delta_{\mathcal{H}}^2}$$

Classical Reconstruction Methods and NN model

Method	Requires	Pro	Contra
Electron (EL)	$E_{l'}$, $\theta_{l'}$	precise	sensitive to QED radiation
Jacques-Blondel (JB)	$\delta_{\mathcal{H}}$, $P_{T,\mathcal{H}}$	resistant to QED radiation	needs precise jet energy measurements
Double Angle (DA)	$\theta_{l'}$, $\gamma_{\mathcal{H}}$	no need for precise jet energy measurements	poor resolution at low x and low Q^2

We aim to reconstruct the kinematics of DIS events combining multiple **classical methods** with an application of deep neural networks and hereby utilizing **complete information** from the collision event.

Kinematic Reconstruction with Deep Neural Networks

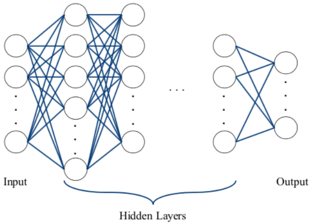
The idea is to build a neural network method that combines the quantities determined with the classical methods in a way:

$$Q^2_{NN} = A_{Q^2}\left(Q^2_{EL}, Q^2_{DA}, Q^2_{JB}\right) + L_{Q^2}\left(A_{Q^2}, E_{l'}, \theta_{l'}\right) + H_{Q^2}\left(A_{Q^2}, \delta_{\mathcal{H}}, P_{T,\mathcal{H}}\right)$$

And reconstruct x, with Q^2_{NN} as an input, in the form:

$$x_{NN} = A_x\left(x_{EL}, x_{DA}, x_{JB}\right) + L_x\left(A_x, Q^2_{NN}, E_{l'}, \theta_{l'}\right) + H_x\left(A_x, Q^2_{NN}, \delta_{\mathcal{H}}, P_{T,\mathcal{H}}\right)$$

Neural networks can be used to reconstruct the kinematics by weighting classical reconstructions and using all four of the measured quantities as corrections.



Deep Neural Networks

A neural network is the sequential application of affine transformations and a fixed nonlinear function.

Call $\mathbf{\Psi}^D_{m,n}(\alpha) :=$ the set of neural networks with D hidden layers that map $\mathbb{R}^m \to \mathbb{R}^n$ with activation function α.

If $\psi \in \mathbf{\Psi}^D_{m,n}$, then

$$\begin{cases} h^0(\mathrm{x}) = \mathrm{x}, \\ h^{i+1}(\mathrm{x}) = \alpha\left(A_i h^i(\mathrm{x})\right) \text{ for } 0 \leq i < D \\ \psi(\mathrm{x}) = A_D h^D(\mathrm{x}) \end{cases} \tag{1}$$

with affine transformations A_i.

A Particular Subclass of Deep Neural Networks

Consider a particular subclass of neural networks, call it: $\mathbf{\Phi}^D_{m,n}(\alpha)$.

If a function $\phi \in \mathbf{\Phi}^D_{m,n}(\alpha)$, then

$$\begin{cases} h^1(\mathrm{x}) = \alpha\left(A_0\mathrm{x}\right), \\ h^{i+1}(\mathrm{x}) = \alpha\left(A_i h^i(\mathrm{x})\right) \text{ for } 0 \leq i < D \\ \phi(\mathrm{x}) = W_0\mathrm{x} + \sum_{i=1}^{D} W_i h^i(\mathrm{x}) \end{cases} \tag{2}$$

with affine transformations A_i and matrices W_i.

Properties About the Subclass

- **Effective**
  Universal approximation capability: can approximate any continuous function to arbitrary accuracy
- **Robust**
  increasing the depth of the network (i.e. the number of terms in the sum) necessarily reduces the error
- **Computationally Efficient**
  structure avoids "vanishing" gradients arising in the backpropagation algorithm

Optimization Methods

The optimal function from the class $\mathbf{\Phi}^{D}_{m,n}(\alpha)$ is defined by a collection of parameters (ω) that minimizes the generalization error.

With a randomly sampled data set $\{\mathrm{x}_i, \mathrm{y}_i\}_{i=1}^{N}$, if N is sufficiently large, then the generalization error can be approximated by an empirical error (fidelity term):

$$\frac{1}{N}\sum_{i=1}^{N} \ell(\omega, \mathrm{x}_i, \mathrm{y}_i) \tag{3}$$

where ℓ is some loss function measuring the discrepancy between the observed output and the neural network output.

We used the logarithmic mean square error for the fidelity term by selecting for the loss function:

$$\ell(\omega, \mathrm{x}_i, \mathrm{y}_i) = \| \log \mathrm{y}_i - \log \phi_\omega(\mathrm{x}_i) \|_2^2 \tag{4}$$

Regularization

Due to the universality of neural networks, there is a model that can achieve zero empirical error. In the presence of noise, then, the model can overfit to the data sample and lose it generalizability.

There is good evidence that minimizing the ℓ^1 norm provides sparse optimal solutions with a minimal number of nonzero elements.

The final optimization problem is:

$$\min_{\omega} \frac{1}{N} \sum_{i=1}^{N} \ell(\omega, \mathrm{x}_i, \mathrm{y}_i) + \lambda \cdot \|\omega\|_1 \tag{5}$$

This optimization problem is solved using stochastic gradient methods.

Training Neural Network Models

- The training of the neural network model was performed on the Monte Carlo generated DIS events that were passed through the ZEUS detector simulation and reconstructed with the standard ZEUS software, see details in backup slides.
- The selection of the events on the detector level (after the reconstruction) was chosen to be close to the selection used in the published ZEUS analyses of NC DIS process, see details in backup slides.
- For the training all the phase space was used, but the final performance is presented for the specific kinematic regions (bins). It has been measured from a similar, but statistically independent samples. See more details on the training in the backup slides.

Distribution of events used for training

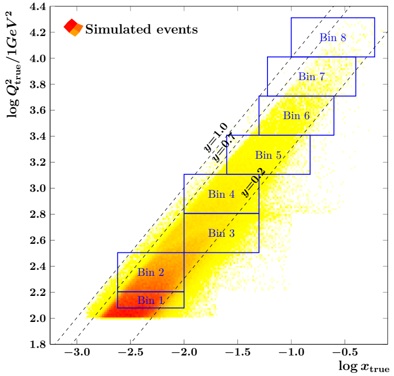
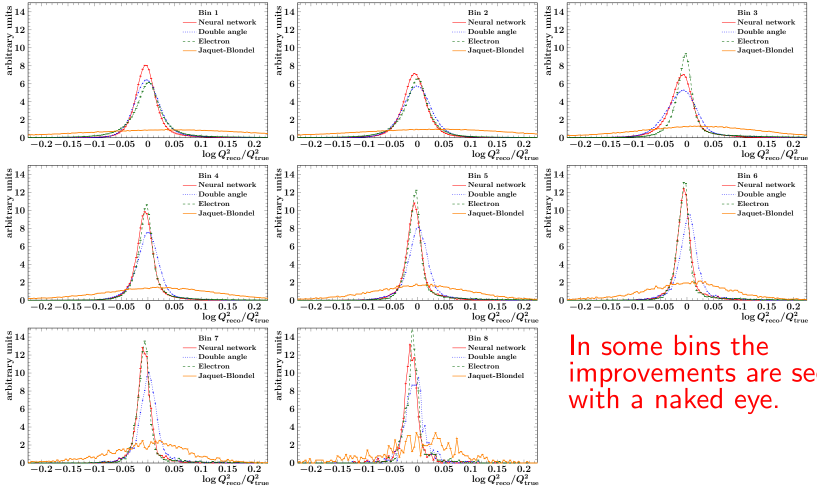


Figure: Distribution of (x, Q^2) for the training set and boundaries of bins.

Bin	$Q^2\ (GeV^2)$	x
1	120 - 160	0.0024 - 0.010
2	160 - 320	0.0024 - 0.010
3	320 - 640	0.01 - 0.05
4	640 - 1280	0.01 - 0.05
5	1280 - 2560	0.025 - 0.150
6	2560 - 5120	0.05 - 0.25
7	5120 - 10240	0.06 - 0.40
8	10240 - 20480	0.10 - 0.60

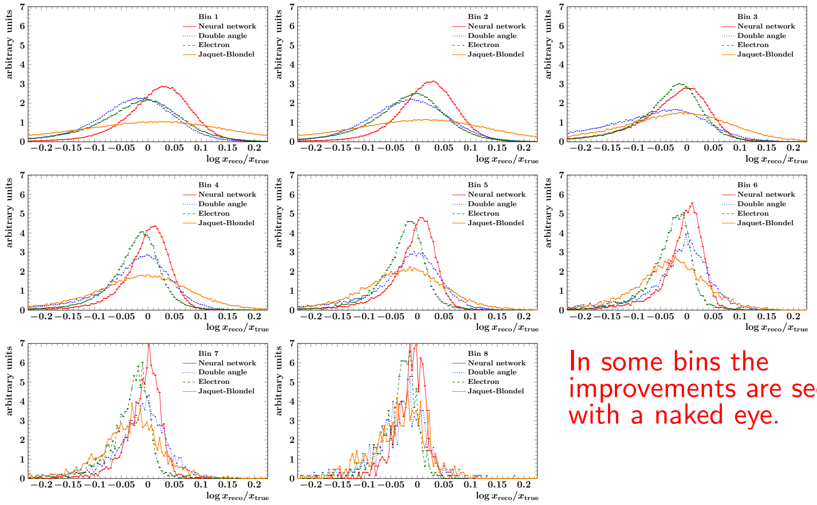
Table: Kinematic bins in x and Q^2 used for performance comparisons. The bins were chosen to be close to the bins used in the analyses of hadronic final state in the ZEUS experiment.

Results: distribution of $\log Q^2_{\text{reco}} - \log Q^2_{\text{true}}$



In some bins the improvements are seen with a naked eye.

Results: distribution of $\log x_{\mathrm{reco}} - \log x_{\mathrm{true}}$



In some bins the improvements are seen with a naked eye.

Results: RMSE in bins of x and Q^2 for different methods

Bin	Events	Resolution of $\log x$		Resolution of $\log Q^2/1GeV^2$	
1	301780	**NN: 0.070**	EL: 0.083	**NN: 0.035**	EL: 0.035
		JB: 0.180	DA: 0.103	JB: 0.203	DA: 0.062
2	350530	**NN: 0.069**	EL: 0.082	**NN: 0.040**	EL: 0.043
		JB: 0.167	DA: 0.096	JB: 0.192	DA: 0.064
3	138456	**NN: 0.098**	EL: 0.130	NN: 0.055	**EL: 0.053**
		JB: 0.138	DA: 0.100	JB: 0.150	DA: 0.077
4	74844	**NN: 0.067**	EL: 0.084	**NN: 0.044**	EL: 0.046
		JB: 0.117	DA: 0.077	JB: 0.138	DA: 0.063
5	31043	**NN: 0.064**	EL: 0.091	**NN: 0.036**	EL: 0.041
		JB: 0.102	DA: 0.073	JB: 0.117	DA: 0.053
6	11475	**NN: 0.053**	EL: 0.079	**NN: 0.033**	EL: 0.036
		JB: 0.083	DA: 0.061	JB: 0.100	DA: 0.045
7	3454	**NN: 0.050**	EL: 0.069	**NN: 0.036**	EL: 0.038
		JB: 0.074	DA: 0.055	JB: 0.093	DA: 0.042
8	624	**NN: 0.036**	EL: 0.055	**NN: 0.033**	EL: 0.037
		JB: 0.067	DA: 0.045	JB: 0.095	DA: 0.041

Smaller = better

Full details can be found on arXiv (https://arxiv.org/abs/2108.11638).

Discussion

In addition to a better resolution, the reconstruction with the DNN has two important advantages over the classical methods or any simple combination of them:

- allows an extension of the model with various physics observables in the most robust way
- allows the use of the desired definition of the kinematic observables, avoiding the intrinsic biases of the other methods

Interesting to admit:

- Despite all the used samples were dominated by the events with lower Q^2 and x it has not prevented the NN approach from reasonable performance at higher Q^2 and x.

Outlook and Conclusions

- With the appropriate selection of the training set, data enhances the DNNs sufficiently to outperform all classical reconstruction methods on most of the kinematic range considered.
- To effectively do this comparison, we established methods to evaluate the accuracy and robustness of a reconstruction method.
- The next step is to extend the methods to do reconstruction of kinematic observables in semi-inclusive and exclusive processes.
- We attribute the improvements to the advantages in modern scientific computing and machine learning methods as well as the accessibility to computing resources.

Acknowledgements

This work was partially supported by the Center for Nuclear Femtography (CNF), administrated by the Southeastern Universities Research Association under an appropriation from the Commonwealth of Virginia and a CNF graduate fellowship CY 2021 to Abdullah Farhat.

A. Farhat and Y. Xu were supported in part by National Science Foundation under grant DMS-1912958.

Backup slides

Theorem regarding the Subclass

Theorem

With fixed non-zero natural numbers D, m, and n, continuous function $f : \mathcal{X} \subseteq \mathbb{R}^m \to \mathbb{R}^n$, for any $\epsilon > 0$, there exists some $\phi \in \mathbf{\Phi}^D_{m,n}(\alpha)$ such that $\rho(f, \phi) < \epsilon$.

Moreover, if $\phi(\mathrm{x}) = W_0\mathrm{x} + \sum_{i=1}^{D} W_i h^i(\mathrm{x})$, as defined as in equation 2, and for natural number $1 \leq k \leq D$, define $\epsilon_k = \rho(f, W_0 + \sum_{i=1}^{k} W_i h^i(\cdot))$, then ϕ can be selected in a way in which $\epsilon_{k-1} \leq \epsilon_k$.

Particular NN models

$Q^2_{NN} = A_{Q^2}\left(Q^2_{EL}, Q^2_{DA}, Q^2_{JB}\right) + L_{Q^2}\left(A_{Q^2}, E_{l'}, \theta_{l'}\right) + H_{Q^2}\left(A_{Q^2}, \delta_{\mathcal{H}}, P_{T,\mathcal{H}}\right)$
where A_{Q^2}, L_{Q^2}, H_{Q^2} are networks in $\mathbf{\Phi}^5_{3,1}(\alpha)$, with each hidden layer of the networks containing 2000 nodes.

$x_{NN} = A_x\left(x_{EL}, x_{DA}, x_{JB}\right) + L_x\left(A_x, Q^2_{NN}, E_{l'}, \theta_{l'}\right) + H_x\left(A_x, Q^2_{NN}, \delta_{\mathcal{H}}, P_{T,\mathcal{H}}\right)$
where A_x is a network in $\mathbf{\Phi}^{20}_{3,1}(\alpha)$ with with each hidden layer containing 1000 nodes, and L_x, H_x are networks in $\mathbf{\Phi}^{10}_{4,1}(\alpha)$, with each hidden layer containing 500 nodes.

Stochastic Gradient Methods

Stochastic gradient methods on batches of the data, accelerated using classical momentum methods can iteratively solve the problem.

Randomly select ω^0, choose learning rates η^k that diminish to zero, select a batch $I \subseteq \{1, ..., N\}$, and choose momentum parameter μ. Define:

$$L_I(\omega) = \frac{1}{|I|} \sum_{i \in I} \ell(\omega, x_i, y_i) + \lambda \cdot \|\omega\|_1 \tag{6}$$

$$v^{k+1} = \mu v^k - \eta^k \nabla_\omega L_I(\omega^k), \tag{7}$$

$$\omega^{k+1} = \omega^k + v^{k+1}, \tag{8}$$

Then the sequence ω^k converges to a set of parameters defining the optimal neural network.

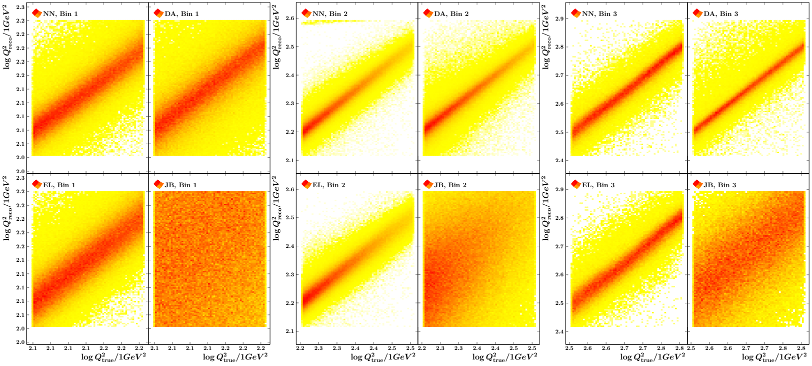
Reconstruction methods

A more detailed description of the Reconstruction methods can be found in `https://old.inspirehep.net/record/332514?ln=en`

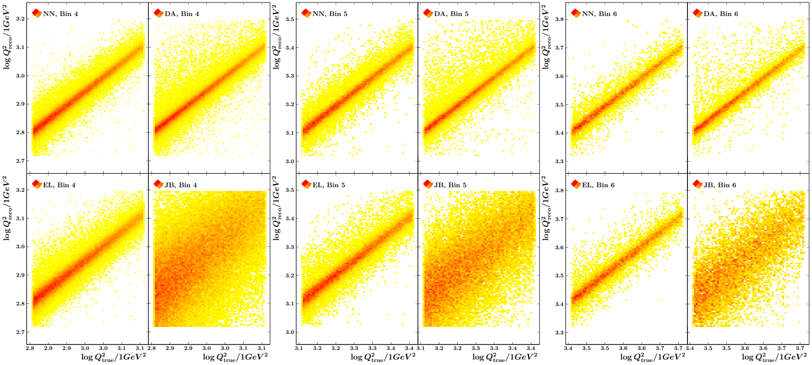
Event selection (most important cuts)

- **Detector status:** It was required that for all the events the detector was functional.
- **Electron energy:** At least one electron candidate with energy greater than $10GeV$
- **Electron identification probability:** The SINISTRA probability of lepton candidate being the DIS lepton was required to be greater than 90%.
- **Electron isolation:** The fraction of the energy not associated to the lepton was required to be less than 10% over the total energy deposited within a cone around the lepton candidate. The cone is defined with a radius of 0.7 units in the pseudorapidity-azimuth plane around the lepton momentum direction.
- **Electron track matching:** The tracking system covers the region of polar angles restricted to $0.3 < \theta < 2.85$. If the lepton candidate was within the tracking system acceptance region, there must be a matched track. This track must have a distance of closest approach between the track extrapolation point at the front surface of the CAL and the cluster center-of-gravity-position of less than 10 cm. The track energy must be greater than 3GeV.
- **Electron position:** To remove regions poorly described by Monte Carlo simulations, additional requirements on the position of the electromagnetic shower were imposed. The events in which the lepton was found in the following regions were rejected: RCAL where the depth was reduced due to the cooling pipe for the solenoid, regions in-between calorimeter sections, regions close the the beam pipe.
- **Primary vertex position:** It was required that the reconstructed primary was close to the central part of the detector, implying $-28.5 < Z_{\mathrm{vtx}} < 26.7cm$.
- **Energy-longitudinal momentum balance:** To suppress photoproduction and beam-gas interaction background events and poor Monte Carlo simulations, restrictions are put on the energy-longitudinal momentum balance. This quantity is defined as:
  $\delta = \delta_e + \delta_{\mathcal{H}} = (E_{e'} - P_{z,e'}) + (E_{\mathcal{H}} - P_{z,\mathcal{H}}) = \sum_i (E_i - P_{z,i})$ where the final summation index runs over all energy deposits in the detector. In this analysis we've applied a condition $38 < \delta < 65$ GeV.
- **Missing transverse energy :** To remove the beam-related background and the cosmic-ray events an cut on the missing energy was imposed. $P_{T,miss}/\sqrt{E_T} < 2.5GeV^{1/2}$, where $P_{T,miss}$ is the missing transverse momentum as measured with the CAL and E_T is the total transverse energy in the CAL.

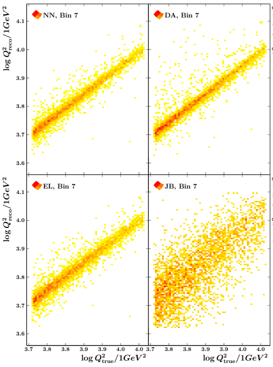
2-D plots for Q^2, bins 1-3

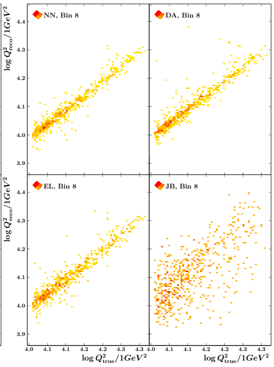


2-D plots for Q^2, bins 4-6

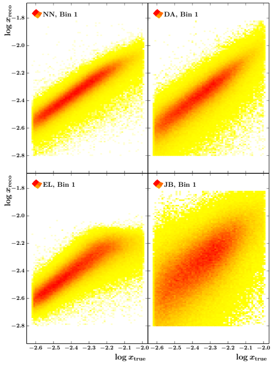


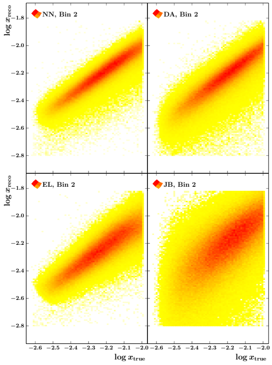
2-D plots for Q^2, bins 7-8

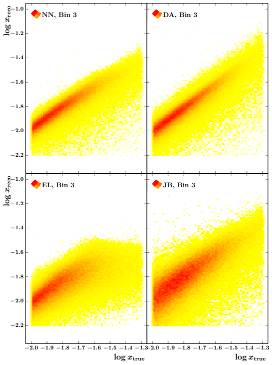




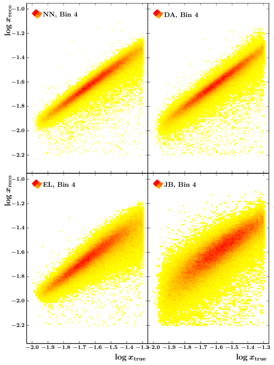
2-D plots for x, bins 1-3

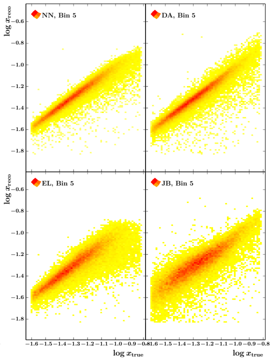


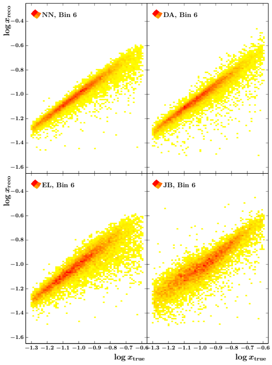




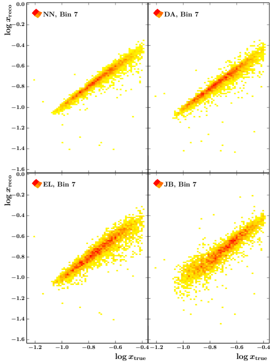
2-D plots for x, bins 4-6

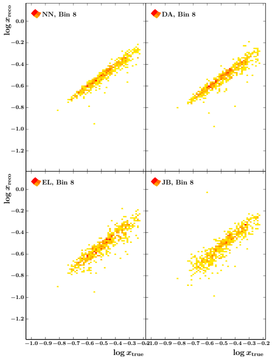






2-D plots for x, bins 7-8





Event shapes

$$T = \frac{\sum_i |\overrightarrow{p_i} \cdot \overrightarrow{n}|}{\sum_i |\overrightarrow{p_i}|}, \tag{9}$$

$$B = \frac{\sum_i |\overrightarrow{p_i} \times \overrightarrow{n}|}{2\sum_i |\overrightarrow{p_i}|}, \tag{10}$$

$$M^2 = \frac{\left(\sum_i E_i\right)^2 - \left|\sum_i \overrightarrow{p_i}\right|^2}{\left(2\sum_i E_i\right)^2}, \tag{11}$$

$$C = \frac{3\sum_{ij} |\overrightarrow{p_i}||\overrightarrow{p_j}| \sin^2\left(\theta_{ij}\right)}{2(\sum_i |\overrightarrow{p_i}|)^2}, \tag{12}$$

where $\overrightarrow{p_i}$ is the momentum of the final-state particle/object i used in the calculations.

The used software

The ROOT package of version 6.22 was used to read the ZEUS data, analyze it and prepare plain text files with selected information to be used with the ML tools. The selected information from the plain text files was piped using the pandas package into Keras interface to tensorflow 2.3.0 library to train the ML models. The packages Eigen, frugally-deep, FunctionalPlus, JSON for Modern C++ were used for execution of the trained models after these were converted into frugally-deep model format. The dependencies for the tensorflow were supplied from the PyPi repository. To speedup the training process the CUDA framework of version 10.1 was used.
The analysis codes were compiled and executed on the Linux system with $x86\_64$ architecture using gcc of version 7.3 and python of version 3.6.8.

The figures with the final results were produced with the PGFPlots package.