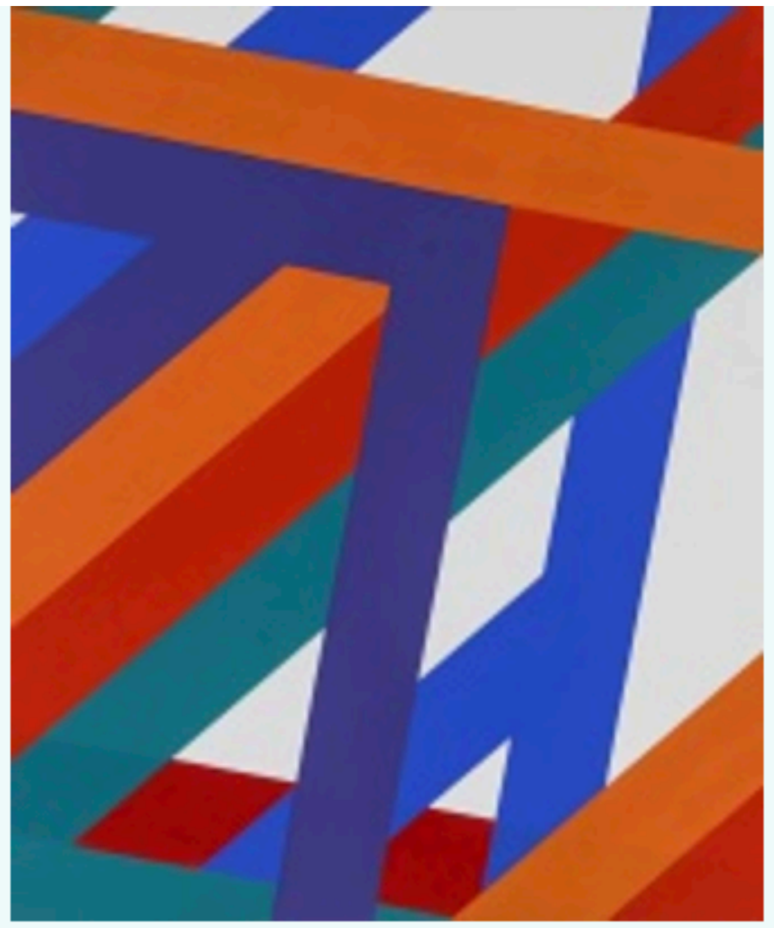
Implementation of machine learning tools in heavy-ion collisions at the LHC


D. Maurer: Relative Quasi Image

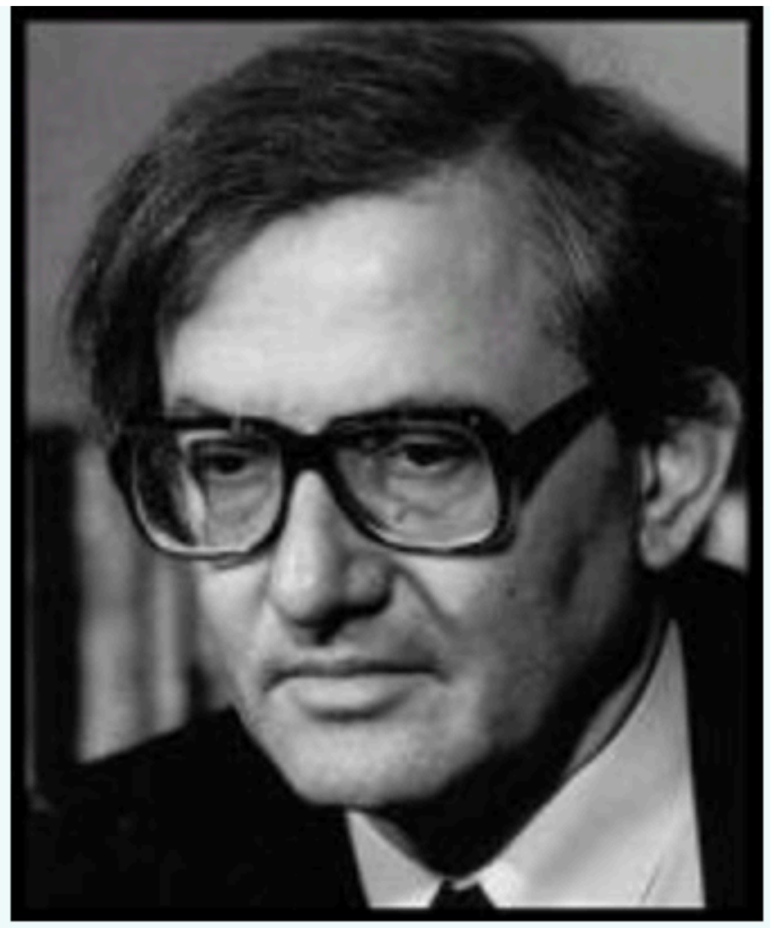
21st ZIMÁNYI SCHOOL

WINTER WORKSHOP

ON HEAVY ION PHYSICS

December 6-10, 2021

Budapest, Hungary


József Zimányi (1931 - 2006)

Neelkamal Mallick
Indian Institute of Technology Indore, India
Neelkamal.Mallick@cern.ch

Based on:

1. N. Mallick, S. Tripathy, A. N. Mishra, S. Deb, and R. Sahoo, Phys. Rev. D103, 094031 (2021)
2. N. Mallick, S. Prasad, A. N. Mishra, R. Sahoo, and G. G. Barnaföldi (In preparation)

Outline

- Introduction
- Motivation
- Methodology
- Results
- Summary

Introduction

Introduction

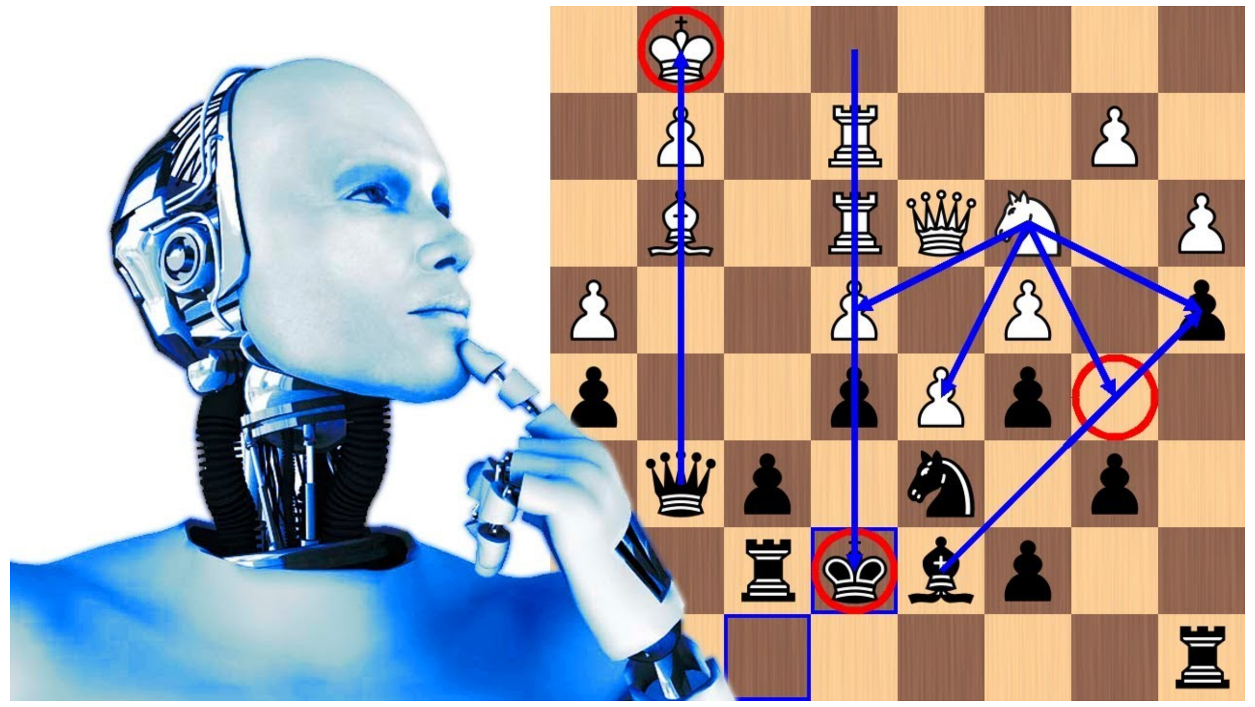
"Machine learning is the field of study that gives computers the ability to learn without being explicitly programmed."

-Arthur Samuel, 1959

Introduction

"Machine learning is the field of study that gives computers the ability to learn without being explicitly programmed."

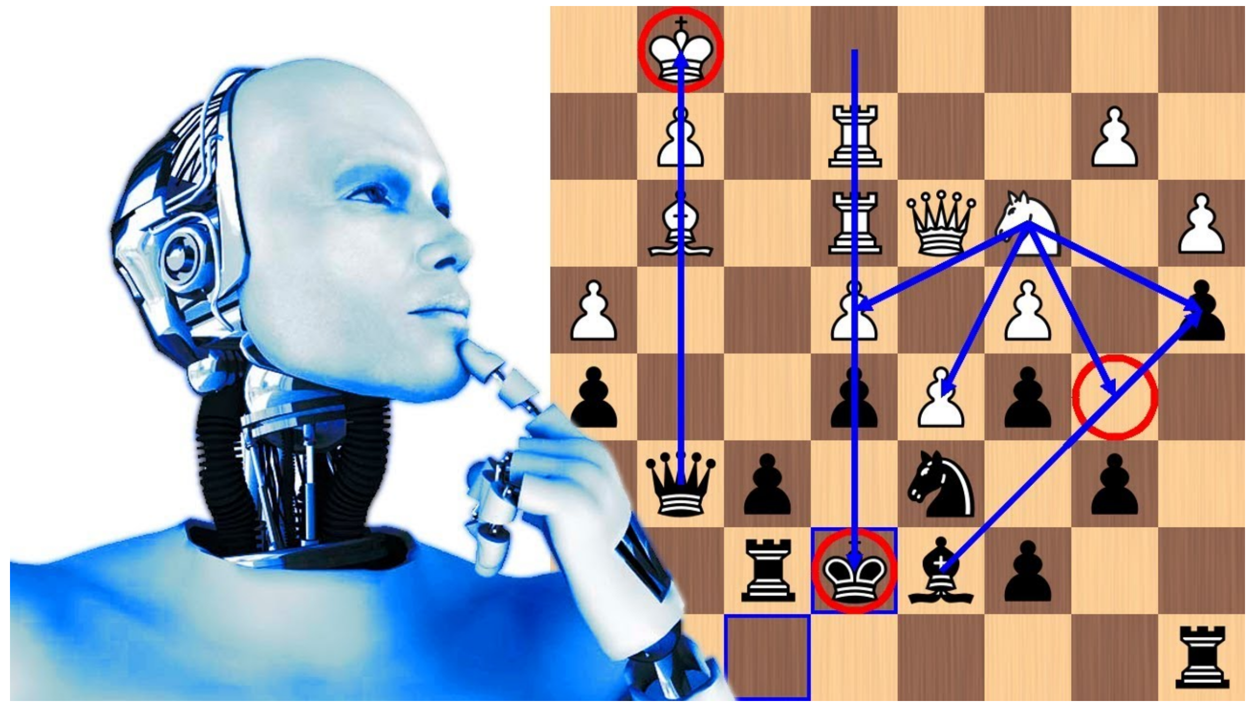
-Arthur Samuel, 1959



Introduction

“Machine learning is the field of study that gives computers the ability to learn without being explicitly programmed.”

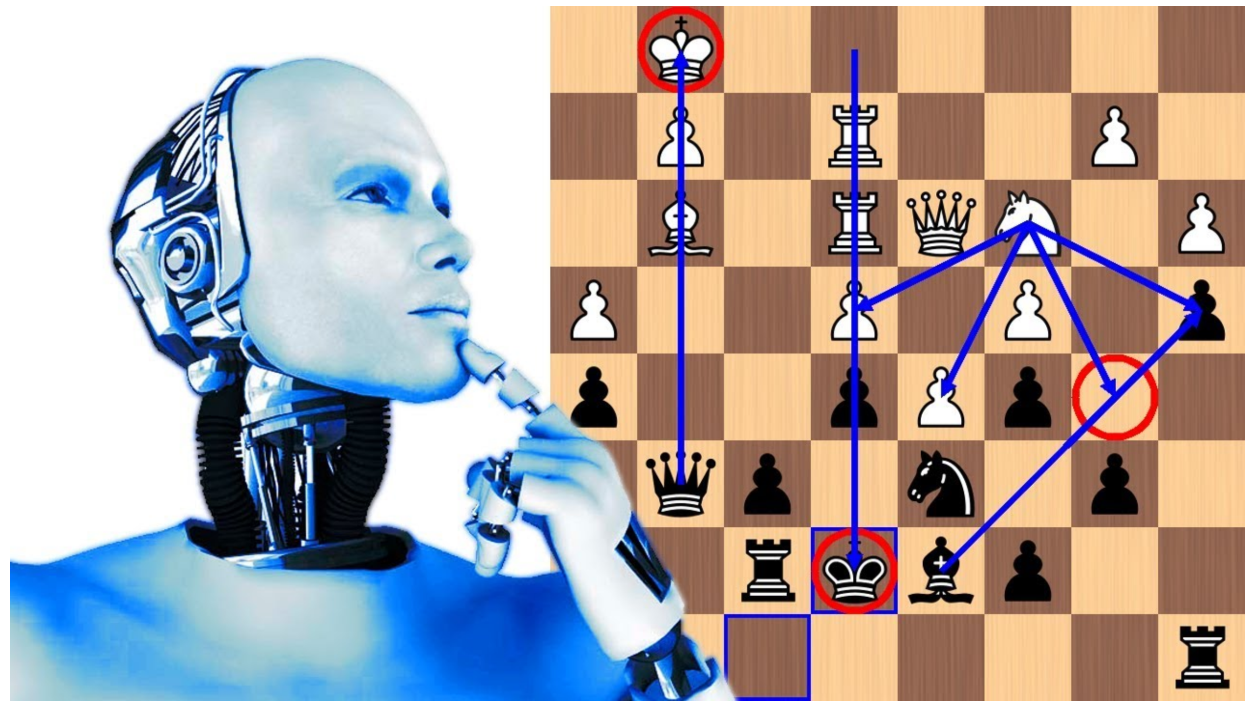
-Arthur Samuel, 1959



Introduction

“Machine learning is the field of study that gives computers the ability to learn without being explicitly programmed.”

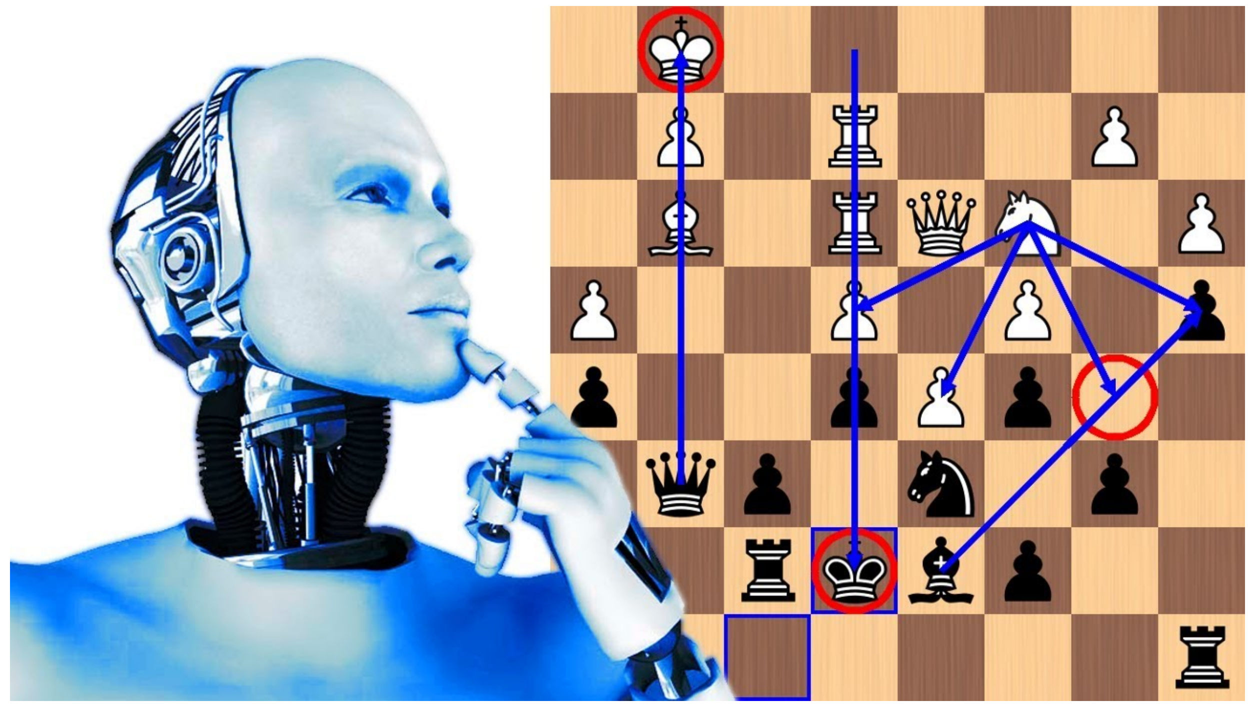
-Arthur Samuel, 1959

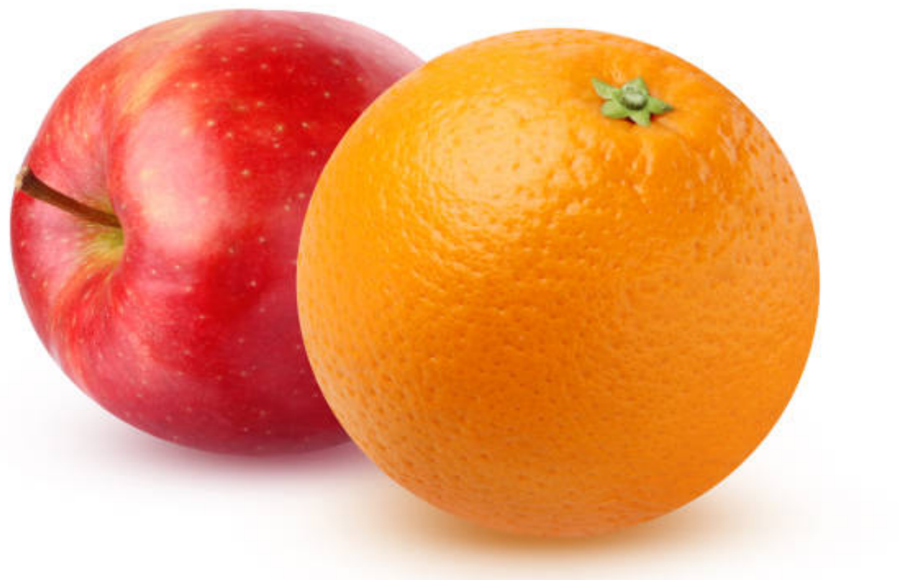


Introduction

“Machine learning is the field of study that gives computers the ability to learn without being explicitly programmed.”

-Arthur Samuel, 1959





What it needs?

- Big data
- Smart algorithm (BDT, DNN, GAN etc.)
- Knowledge from data
- Tune the parameters (Optimise the model)
- Predict!!

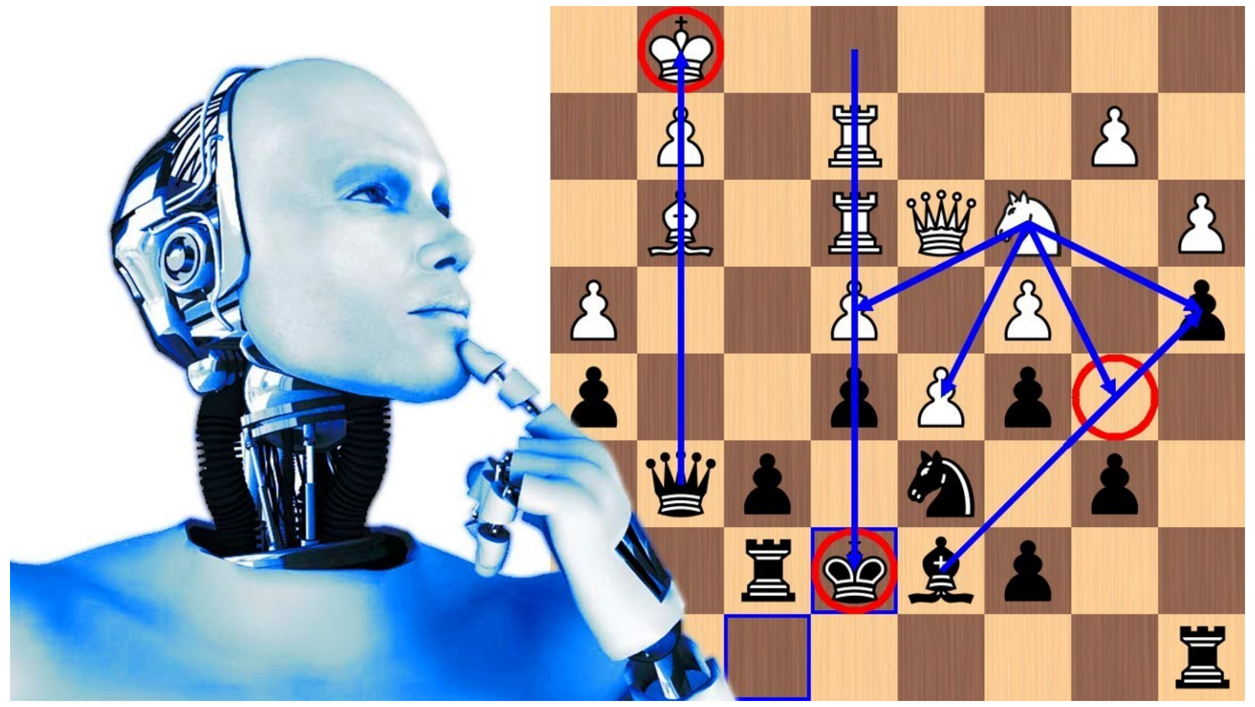
Introduction

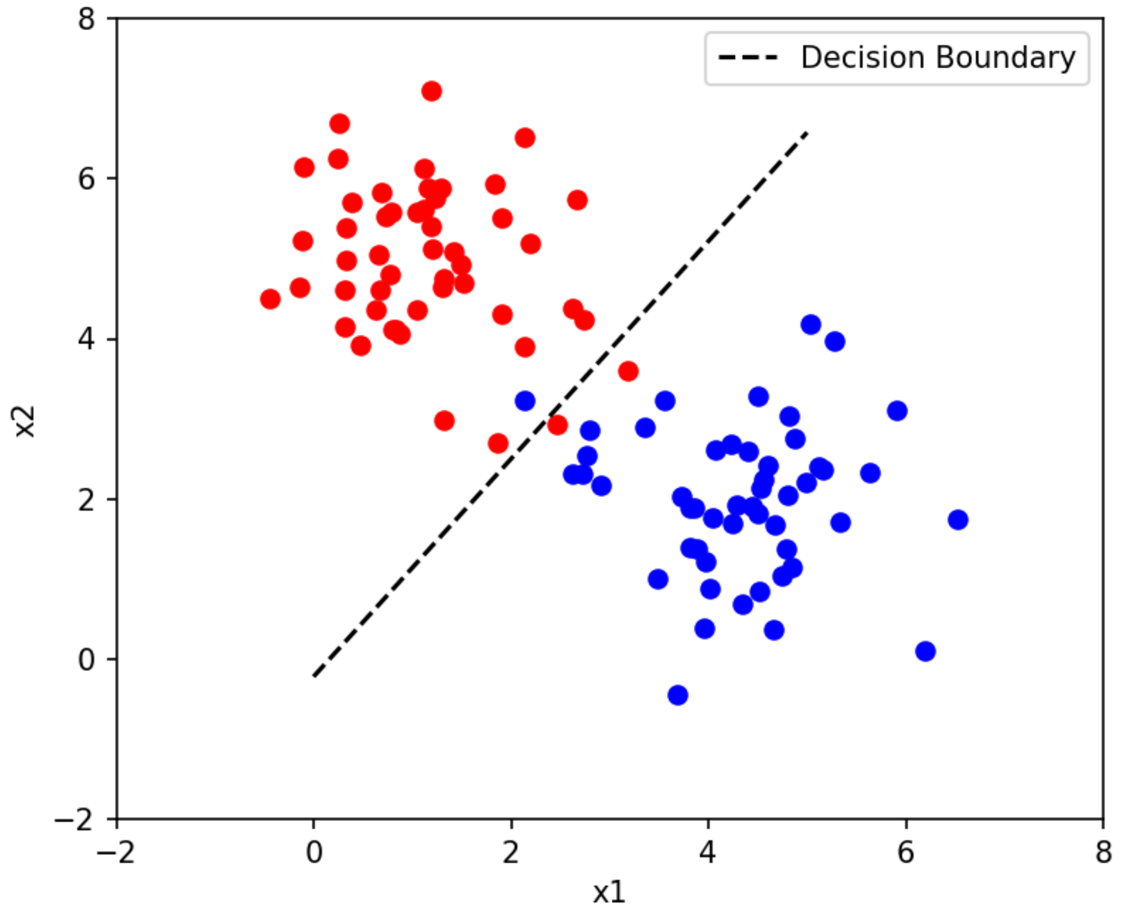
"Machine learning is the field of study that gives computers the ability to learn without being explicitly programmed."

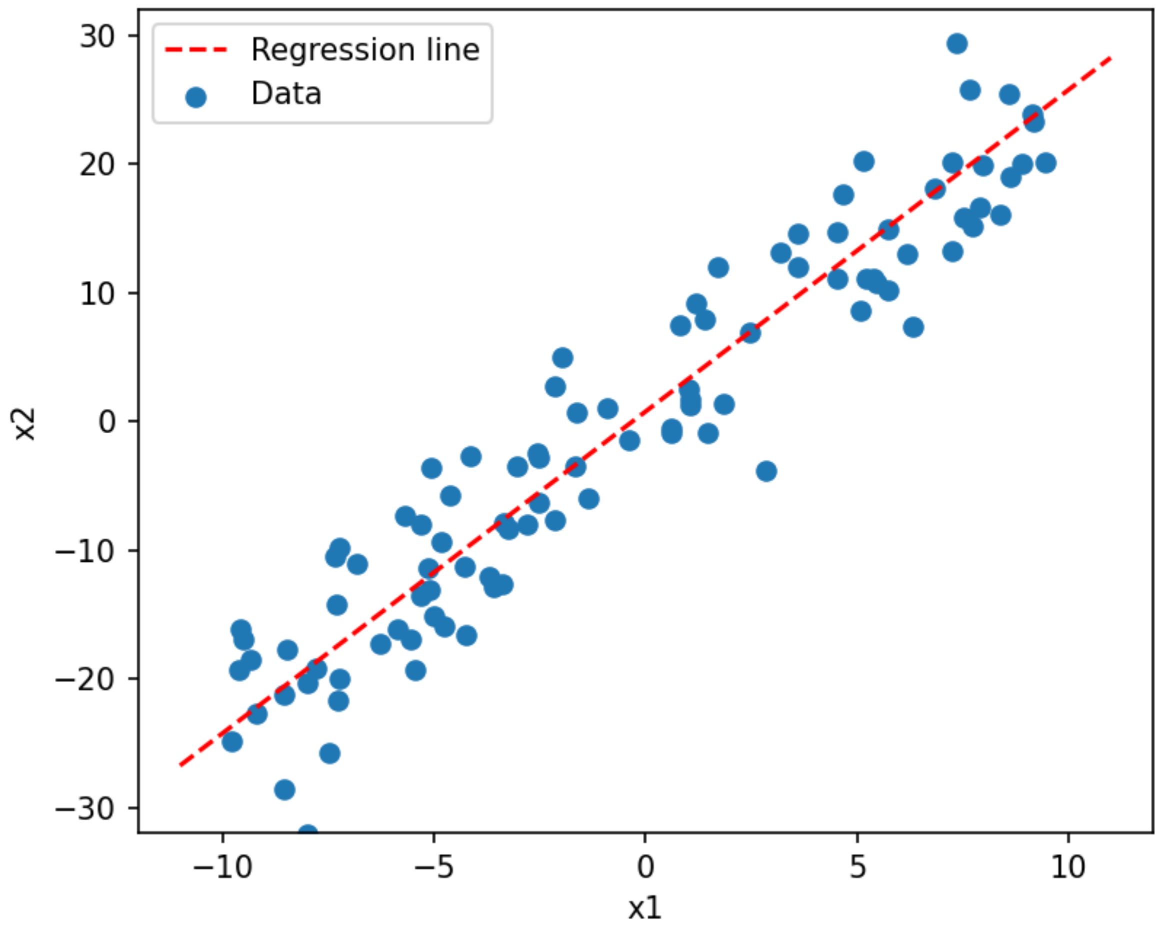
-Arthur Samuel, 1959

What it needs?

- Big data
- Smart algorithm (BDT, DNN, GAN etc.)
- Knowledge from data
- Tune the parameters (Optimise the model)
- Predict!!

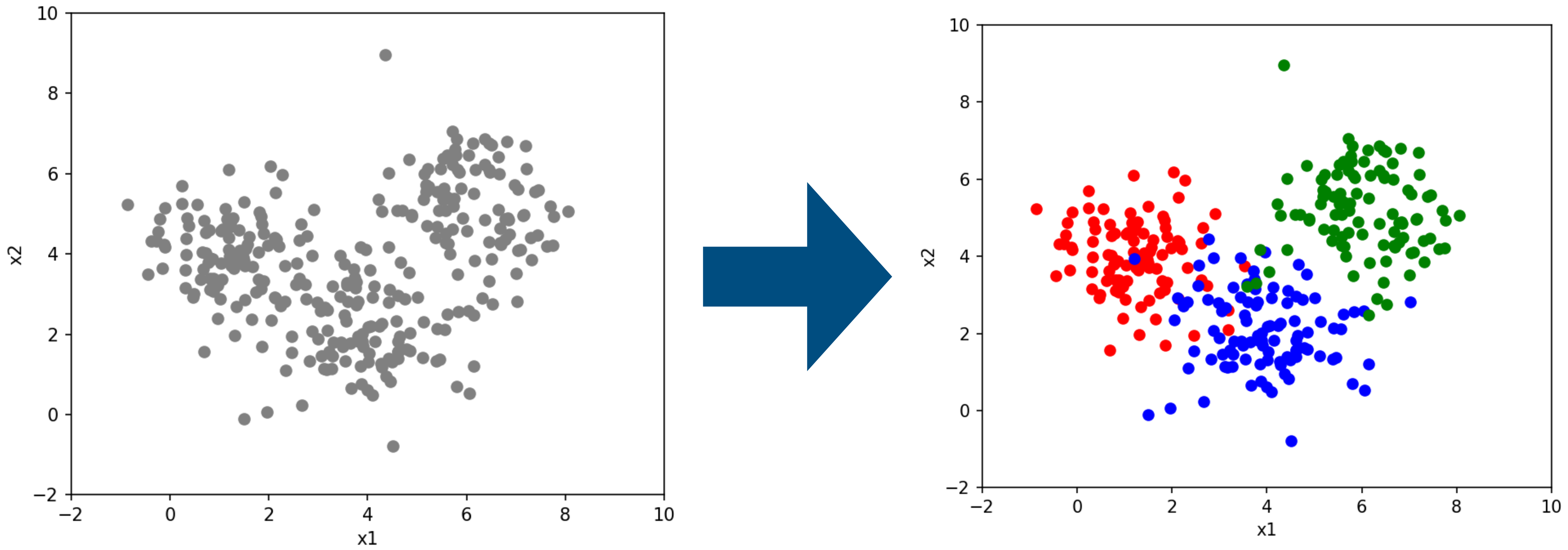






Supervised/unsupervised

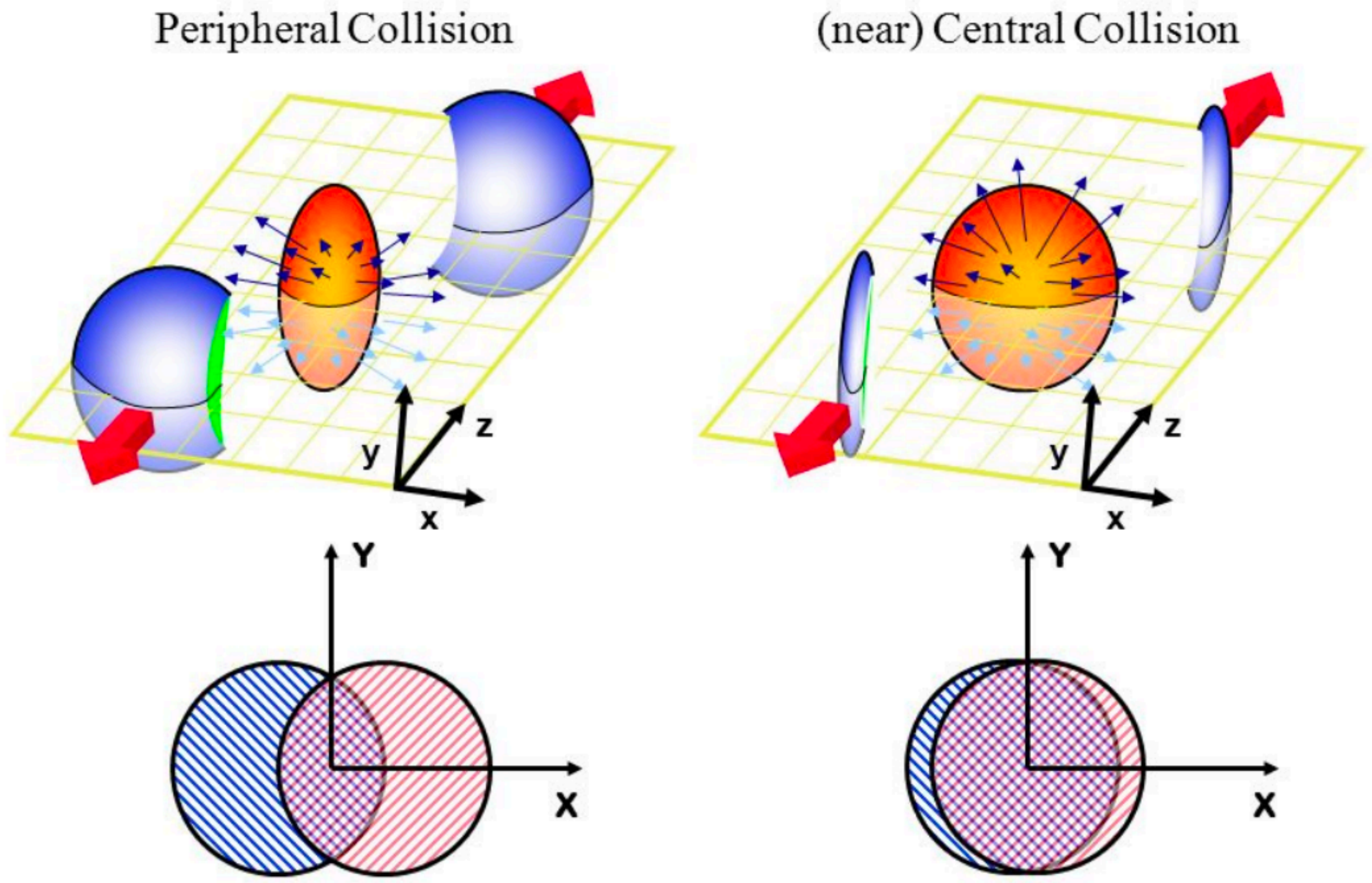
- Classification
- Regression
- Clustering
- Reinforcement learning etc.



Impact parameter (b)

Impact parameter (b)

- Transverse distance between the centres of colliding nuclei
- Initial geometry affects the final state particle production
- Order of a few fermi (10^{-15}m)
- Impossible to estimate from experiments
- Could be inferred from charged particle multiplicity distribution



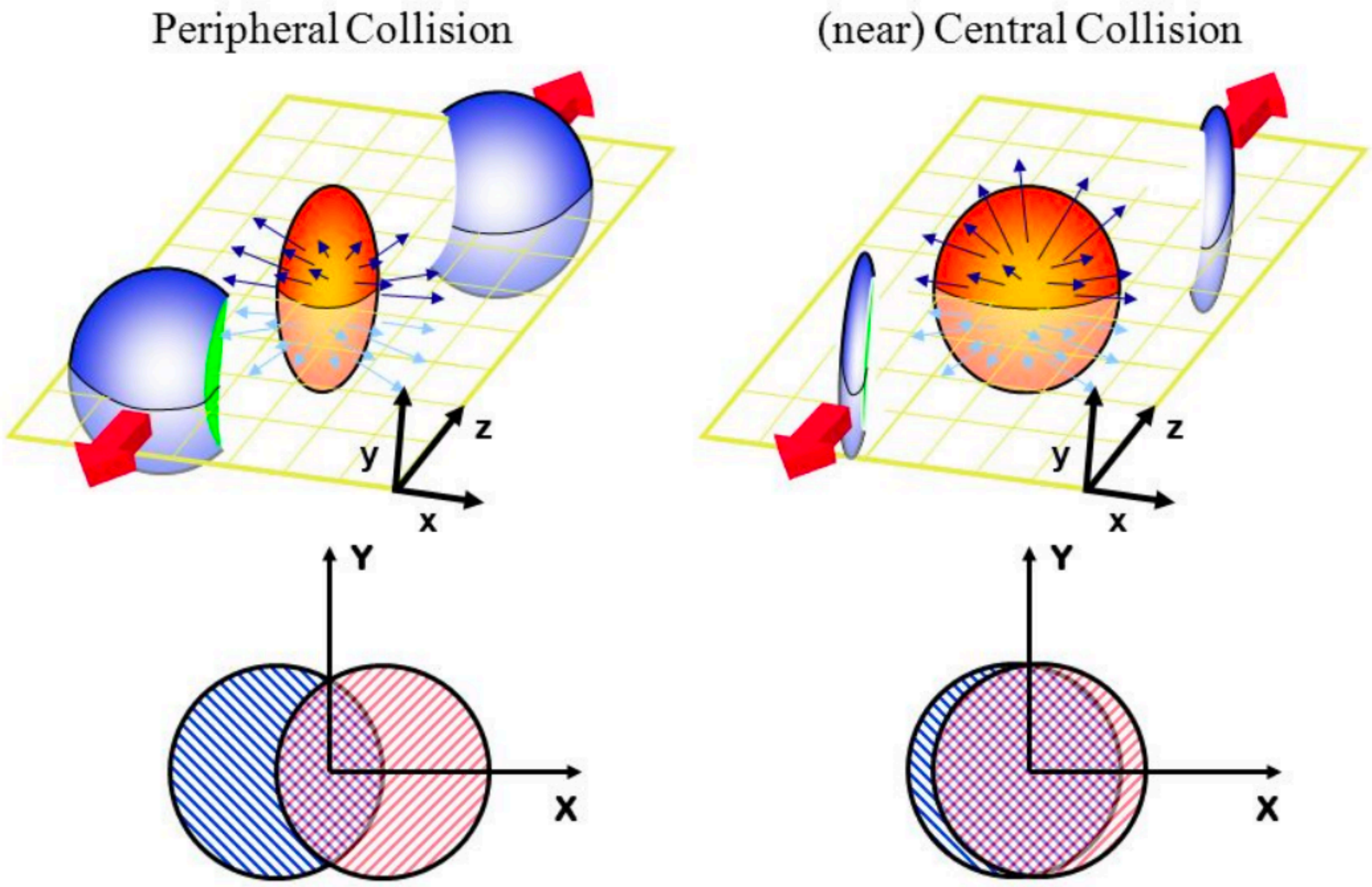
$0 \leq b \leq 2R$

Impact parameter (b)

- Transverse distance between the centres of colliding nuclei
- Initial geometry affects the final state particle production
- Order of a few fermi (10^{-15}m)
- Impossible to estimate from experiments
- Could be inferred from charged particle multiplicity distribution

Transverse Spherocity (S_0)

- Transverse Spherocity distinguishes hard and soft processes
- In pp collisions,
  1. **Jetty**: Back-to-back structure, indication of hard-QCD
  2. **Isotropic**: soft-QCD process
- Dominance of isotropic events in high multiplicity pp collisions
- $\langle p_T \rangle$ is higher for jetty events

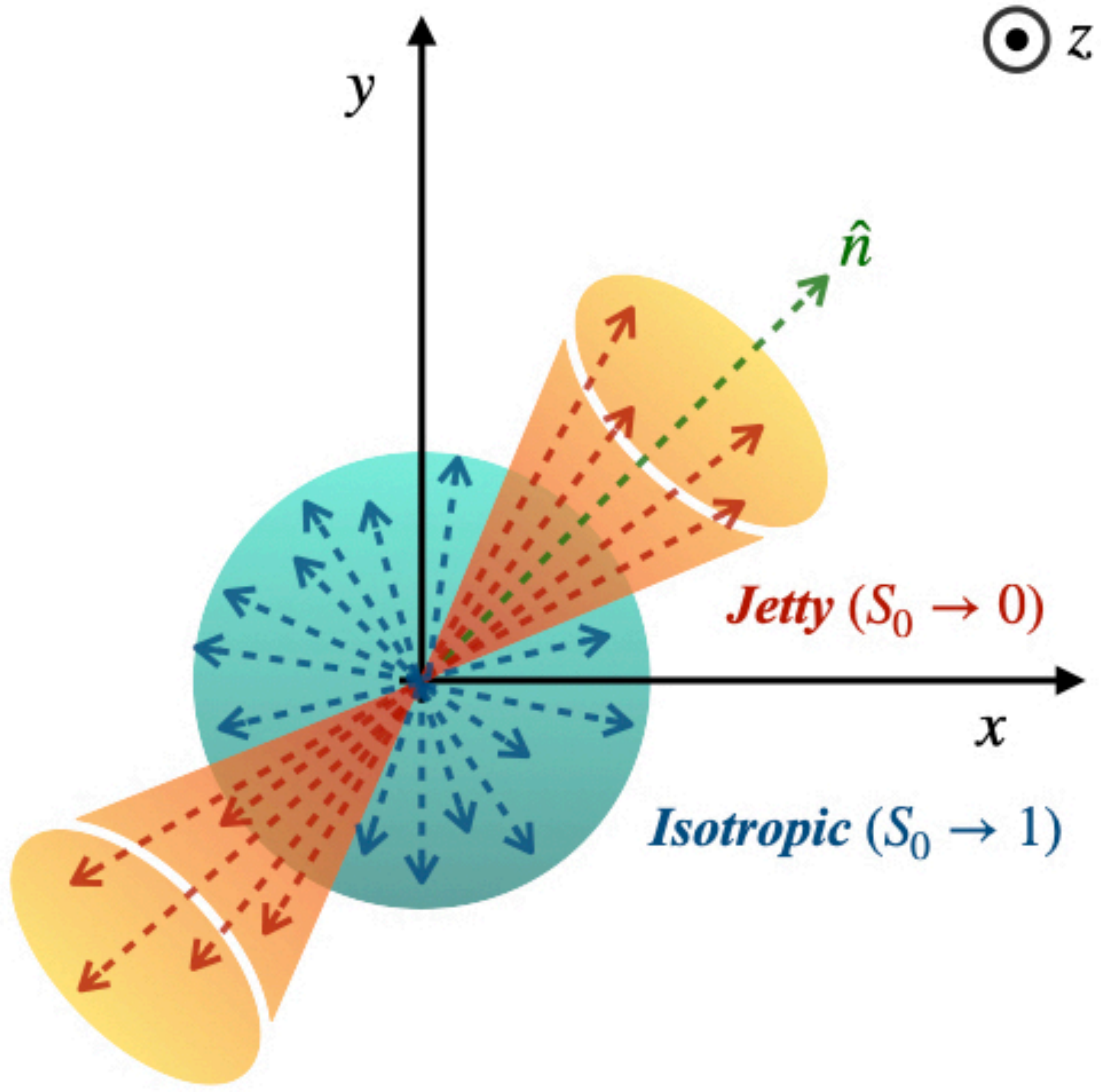


$0 \leq b \leq 2R$

$$S_0 = \frac{\pi^2}{4} \times \min_{\hat{n}=(n_x,n_y,0)} \left(\frac{\Sigma_i \, |\vec{p}_{T_i} \times \hat{n}|}{\Sigma_i \, \vec{p}_{T_i}} \right)^2$$

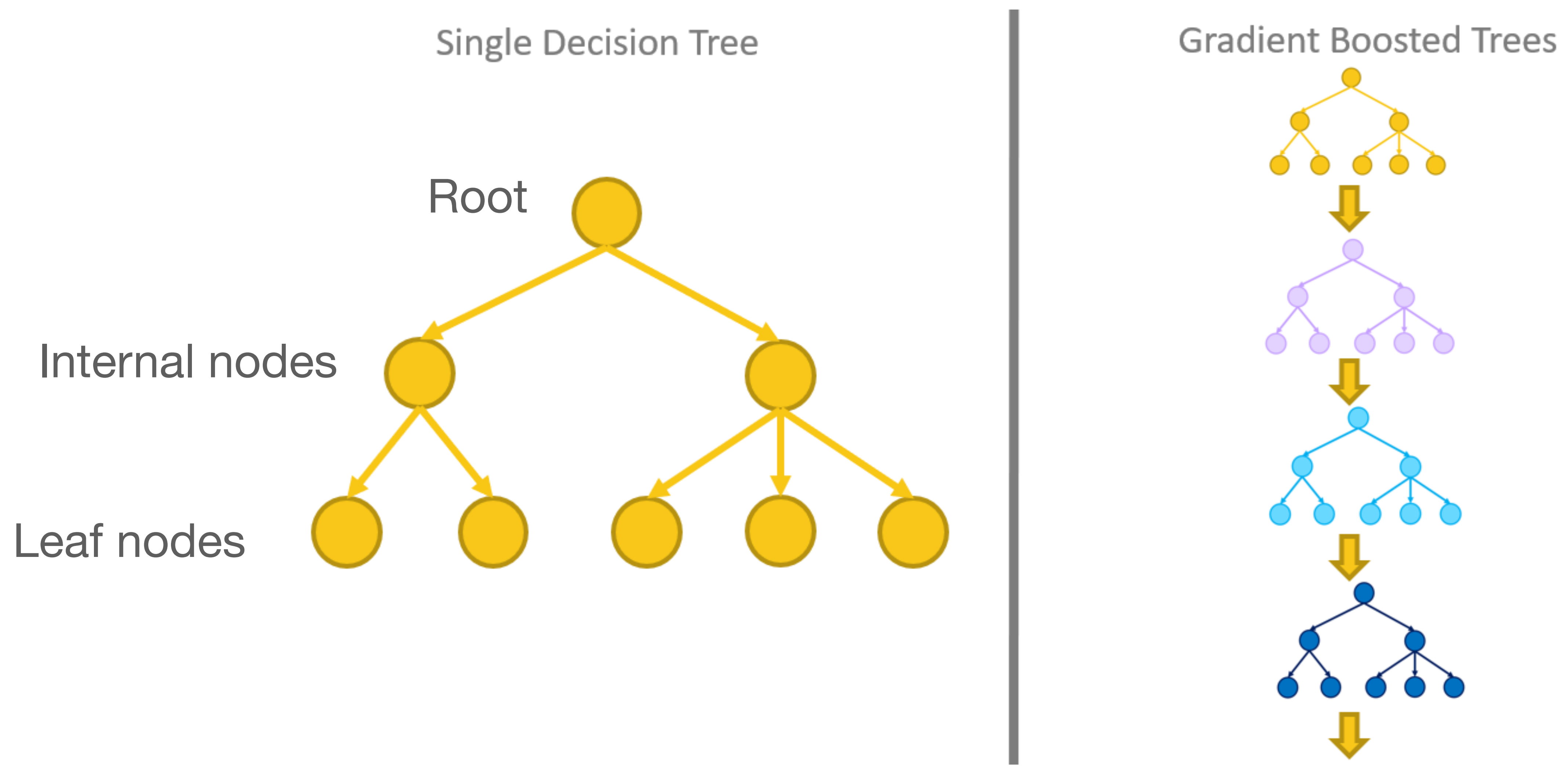
$$p_{\mathrm{T}} = \sqrt{p_x^2 + p_y^2}$$

A. Khuntia et al., J. Phys. G48, 035102



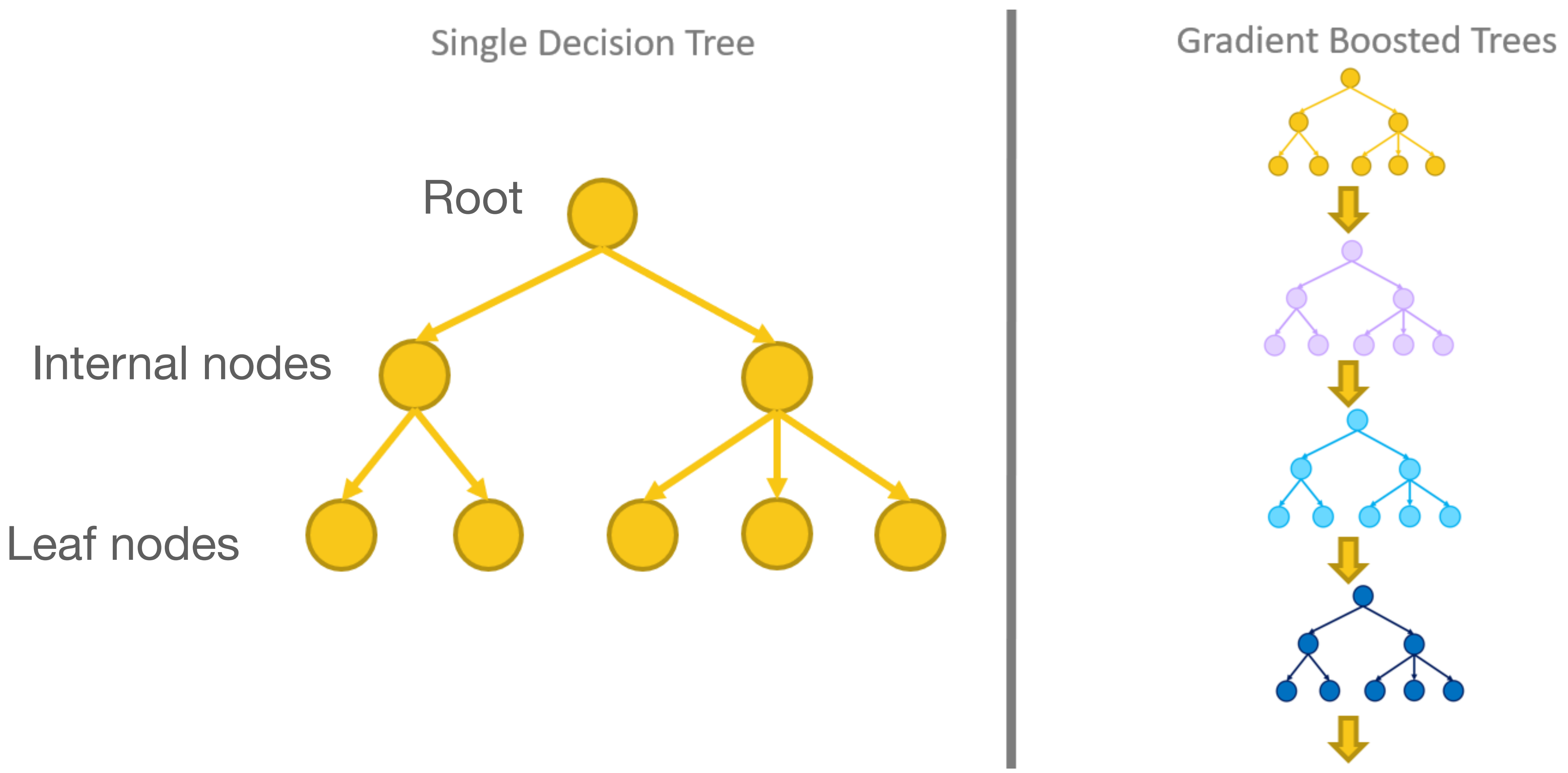
Boosted Decision Trees

Boosted Decision Trees

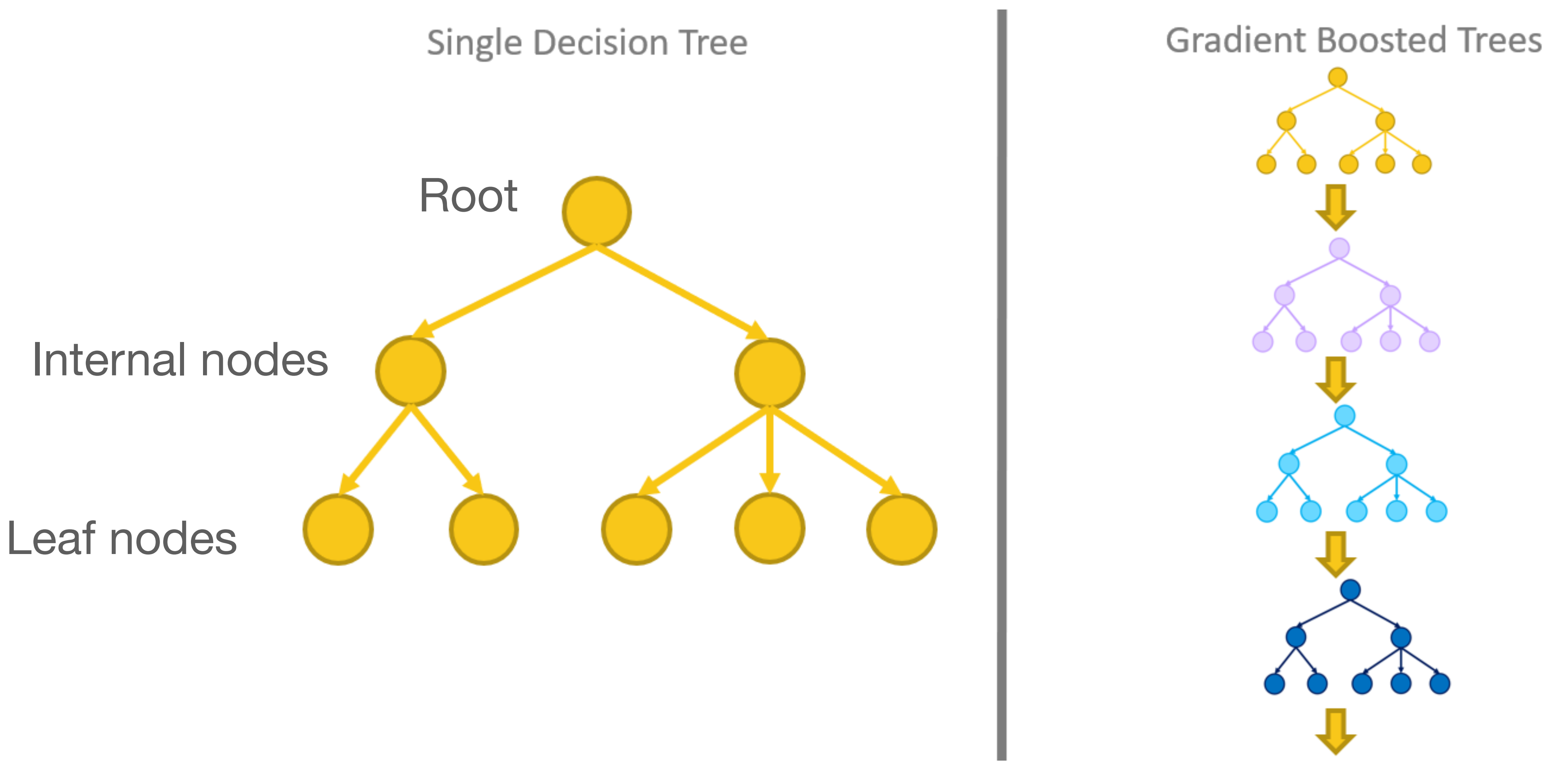


Boosted Decision Trees

- Trees are structures that takes recursive decisions
- Built in a top-down approach
- Root node: The starting point
  Internal nodes: further decision points
  Leaf nodes: End points (target class or values)
- Criteria of splitting:
  Classification: Minimise the node impurity
  Regression: Minimise the MSE
- Splitting continues till a preset (max_depth)
- Boosting: Building an additive forward staged model by combining the outcomes of all previous ones
- Boosting compensates the shortcomings
- Shortcomings are identified as the gradient



Boosted Decision Trees



- Trees are structures that takes recursive decisions
- Built in a top-down approach
- Root node: The starting point
  Internal nodes: further decision points
  Leaf nodes: End points (target class or values)
- Criteria of splitting:
  Classification: Minimise the node impurity
  Regression: Minimise the MSE
- Splitting continues till a preset (max_depth)
- Boosting: Building an additive forward staged model by combining the outcomes of all previous ones
- Boosting compensates the shortcomings
- Shortcomings are identified as the gradient

- The ***CART*** cost function for k^{th} feature with t_k threshold:

$$J(k, t_k) = \frac{m_{\text{left}}}{m} G_{\text{left}} + \frac{m_{\text{right}}}{m} G_{\text{right}}$$

- $G_{\text{left/right}}$ measures the impurity
- $m_{\text{left/right}}$ number of instances in left/right subset
- Gini impurity, cross entropy, MSE, MAE etc.

Input observables and correlation

N. Mallick, S. Tripathy, A. N. Mishra, S. Deb, and R. Sahoo, Phys. Rev. D103, 094031 (2021)

Input observables and correlation

- Pearsons correlation coefficient

$$\rho = \frac{\mathrm{cov}(x, y)}{\sigma_x \sigma_y}$$

- Defines the degree of correlation
- Input Variables: $\langle dN_{ch}/d\eta \rangle$, $\langle N_{ch}^{TS} \rangle$ and $\langle p_T \rangle$
  Output variable: b and S_0
- Good correlation is seen among chosen input and output variables
- The algorithm tries to understand the correlation and exploit the features to arrive on a conclusion (a number)

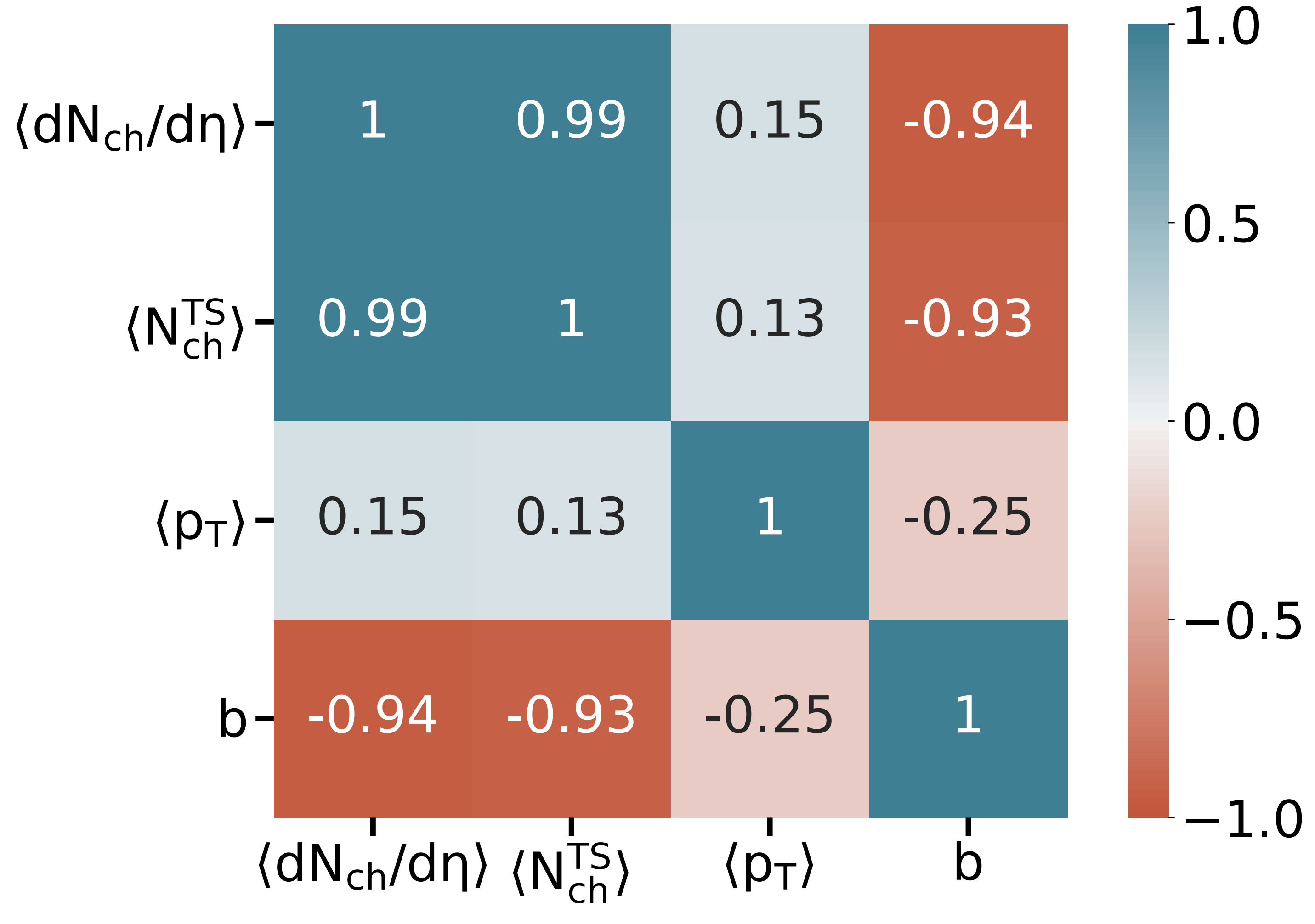
N. Mallick, S. Tripathy, A. N. Mishra, S. Deb, and R. Sahoo, Phys. Rev. D103, 094031 (2021)

Input observables and correlation

- Pearsons correlation coefficient

$$\rho = \frac{\mathrm{cov}(x, y)}{\sigma_x \sigma_y}$$

- Defines the degree of correlation
- Input Variables: $\langle dN_{ch}/d\eta \rangle$, $\langle N_{ch}^{TS} \rangle$ and $\langle p_T \rangle$
  Output variable: b and S_0
- Good correlation is seen among chosen input and output variables
- The algorithm tries to understand the correlation and exploit the features to arrive on a conclusion (a number)



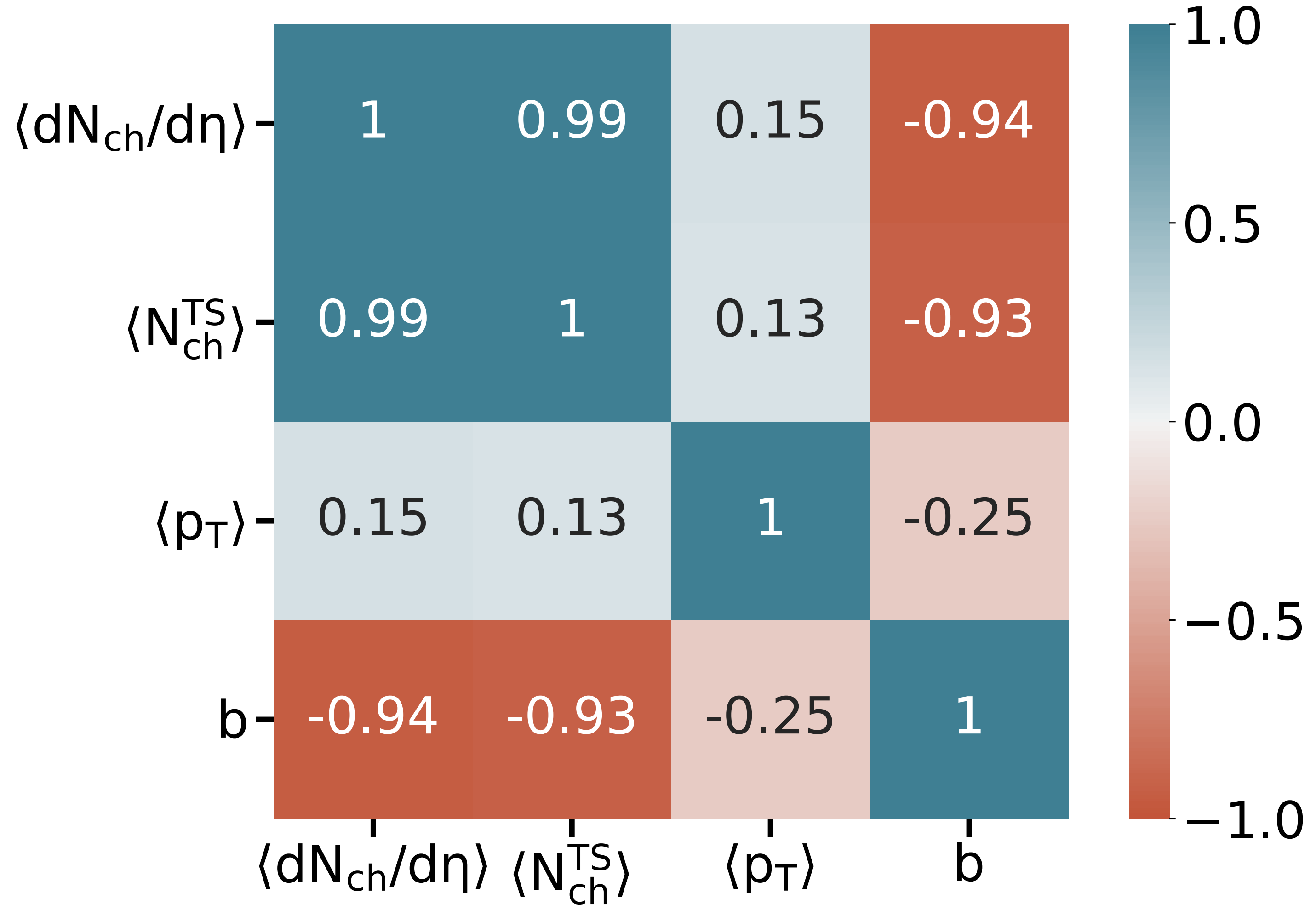
N. Mallick, S. Tripathy, A. N. Mishra, S. Deb, and R. Sahoo, Phys. Rev. D103, 094031 (2021)

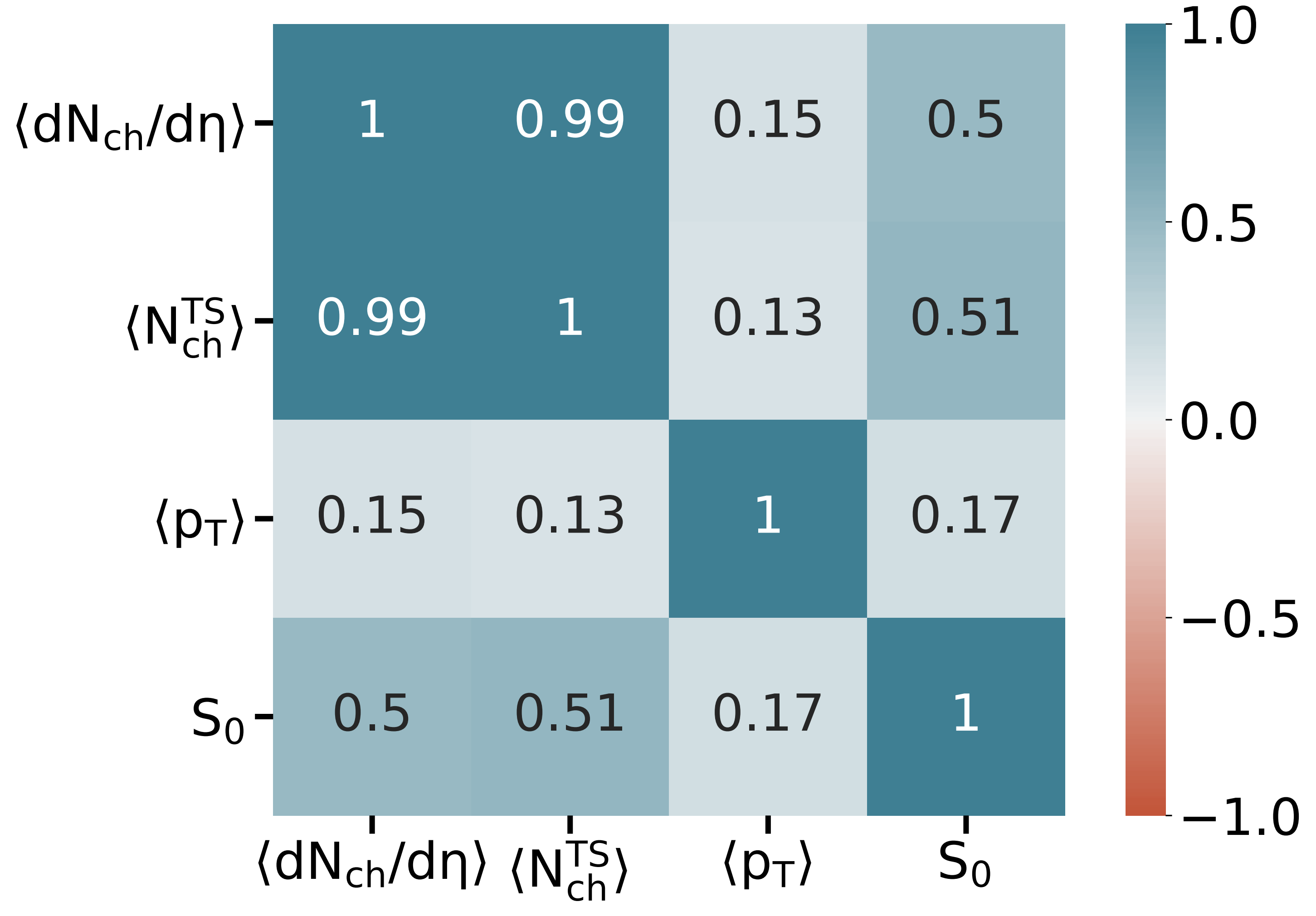
Input observables and correlation

- Pearsons correlation coefficient

$$\rho = \frac{\mathrm{cov}(x, y)}{\sigma_x \sigma_y}$$

- Defines the degree of correlation
- Input Variables: $\langle dN_{ch}/d\eta \rangle$, $\langle N_{ch}^{TS} \rangle$ and $\langle p_T \rangle$
  Output variable: b and S_0
- Good correlation is seen among chosen input and output variables
- The algorithm tries to understand the correlation and exploit the features to arrive on a conclusion (a number)





N. Mallick, S. Tripathy, A. N. Mishra, S. Deb, and R. Sahoo, Phys. Rev. D103, 094031 (2021)

Parameters and training

- Loss Function: Least Square Loss
- Small learning rate = 0.1
- Number of trees = 100
- Training Size: 60,000 events (min. bias)

$$\text{Least Sqaure loss}: l(y_i, F(\mathbf{x_i})) = \frac{1}{2}(y_i - F(\mathbf{x_i}))^2$$

$$\Delta S_0 = \frac{1}{N_{\text{events}}} \sum_{n=1}^{N_{\text{events}}} |\, S_{0_n}^{true} - S_{0_n}^{pred.} \,|$$

Size of training data		2K	10K	20K	40K	50K	60K
Δb [fm]	(Impact parameter)	0.71	0.62	0.58	0.53	0.52	0.52
ΔS_0	(Spherocity)	0.079	0.068	0.062	0.058	0.056	0.055

J. H. Friedman, Ann. Stat. 29, 1189 (2001).

L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, Classification and Regression Trees (Wadsworth & Brooks/ Cole Advanced Books & Software, Monterey, CA, 1984), p. 358, https://doi.org/10.1002/cyto.990080516.

N. Mallick, S. Tripathy, A. N. Mishra, S. Deb, and R. Sahoo, Phys. Rev. D103, 094031 (2021)

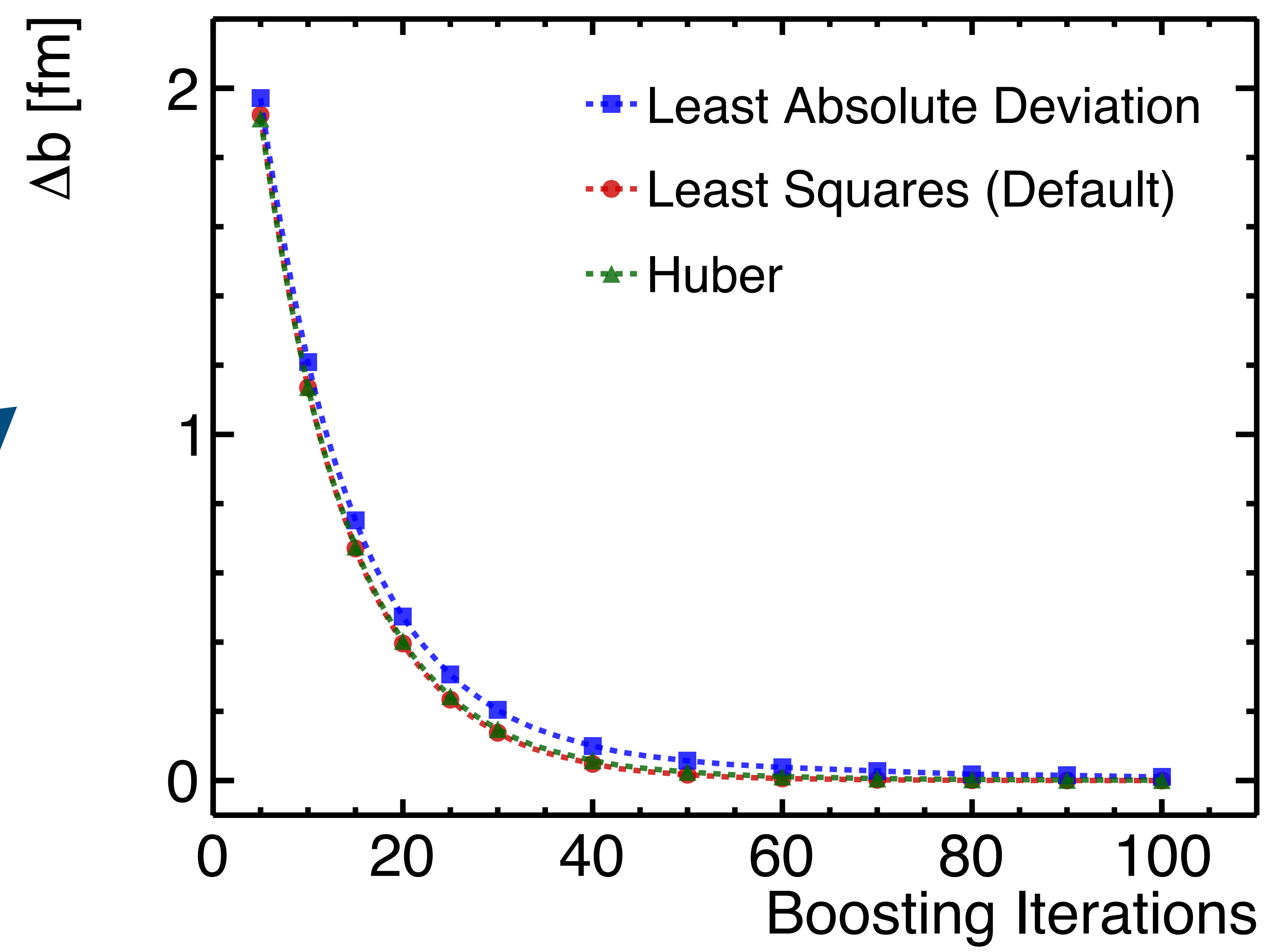
Parameters and training

- Loss Function: Least Square Loss
- Small learning rate = 0.1
- Number of trees = 100
- Training Size: 60,000 events (min. bias)

$$\text{Least Sqaure loss}: l(y_i, F(\mathbf{x_i})) = \frac{1}{2}(y_i - F(\mathbf{x_i}))^2$$

$$\Delta S_0 = \frac{1}{N_{\text{events}}} \sum_{n=1}^{N_{\text{events}}} |S_{0_n}^{true} - S_{0_n}^{pred.}|$$

Size of training data	2K	10K	20K	40K	50K	60K
Δb [fm] (Impact parameter)	0.71	0.62	0.58	0.53	0.52	0.52
ΔS_0 (Spherocity)	0.079	0.068	0.062	0.058	0.056	0.055



J. H. Friedman, Ann. Stat. 29, 1189 (2001).

L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, Classification and Regression Trees (Wadsworth & Brooks/ Cole Advanced Books & Software, Monterey, CA, 1984), p. 358, https://doi.org/10.1002/cyto.990080516.

N. Mallick, S. Tripathy, A. N. Mishra, S. Deb, and R. Sahoo, Phys. Rev. D103, 094031 (2021)

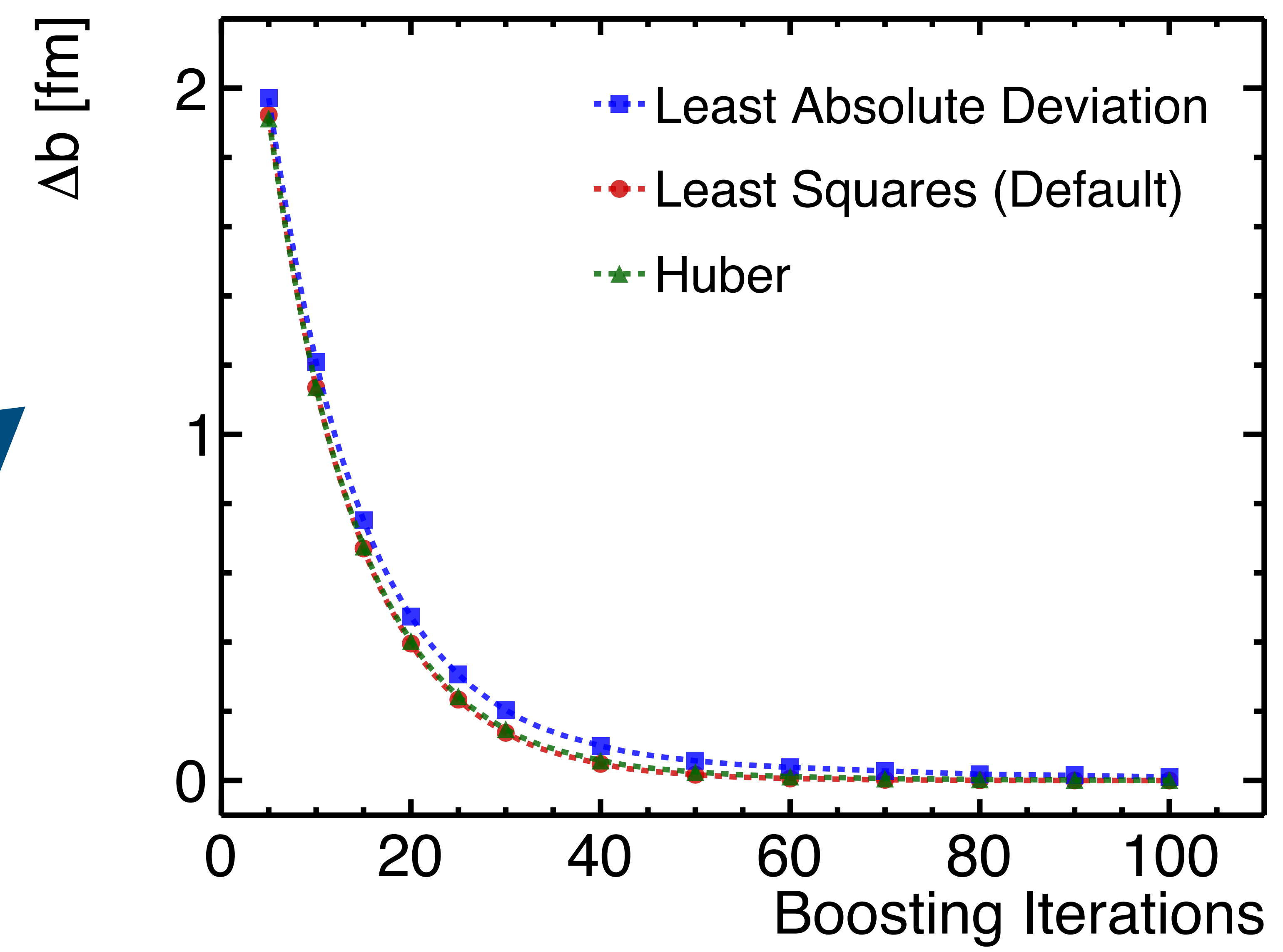
Parameters and training

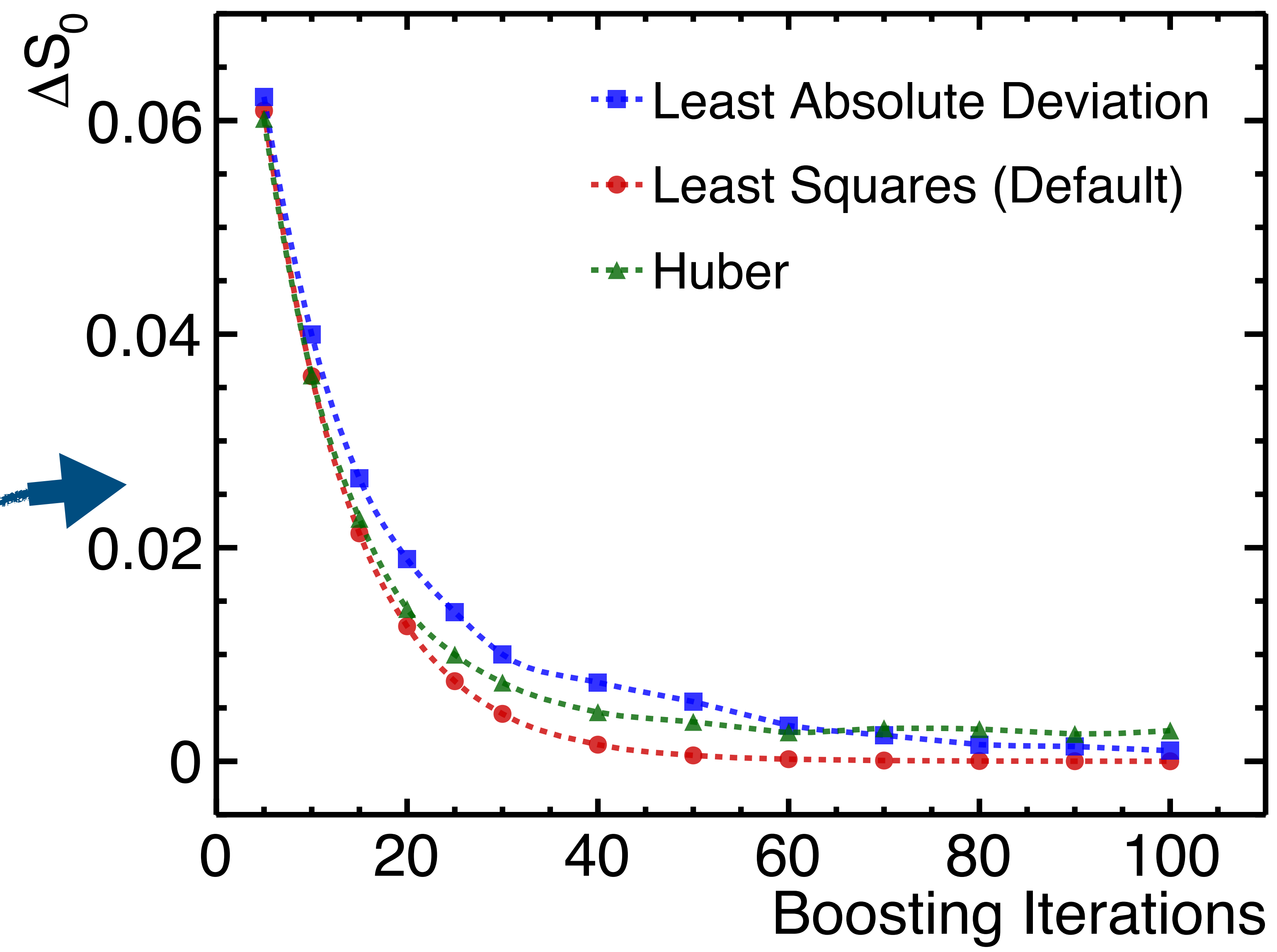
- Loss Function: Least Square Loss
- Small learning rate = 0.1
- Number of trees = 100
- Training Size: 60,000 events (min. bias)

$$\text{Least Sqare loss}: l(y_i, F(\mathbf{x_i})) = \frac{1}{2}(y_i - F(\mathbf{x_i}))^2$$

$$\Delta S_0 = \frac{1}{N_{\text{events}}} \sum_{n=1}^{N_{\text{events}}} |S_{0_n}^{true} - S_{0_n}^{pred.}|$$

Size of training data	2K	10K	20K	40K	50K	60K
Δb [fm] (Impact parameter)	0.71	0.62	0.58	0.53	0.52	0.52
ΔS_0 (Spherocity)	0.079	0.068	0.062	0.058	0.056	0.055





J. H. Friedman, Ann. Stat. 29, 1189 (2001).

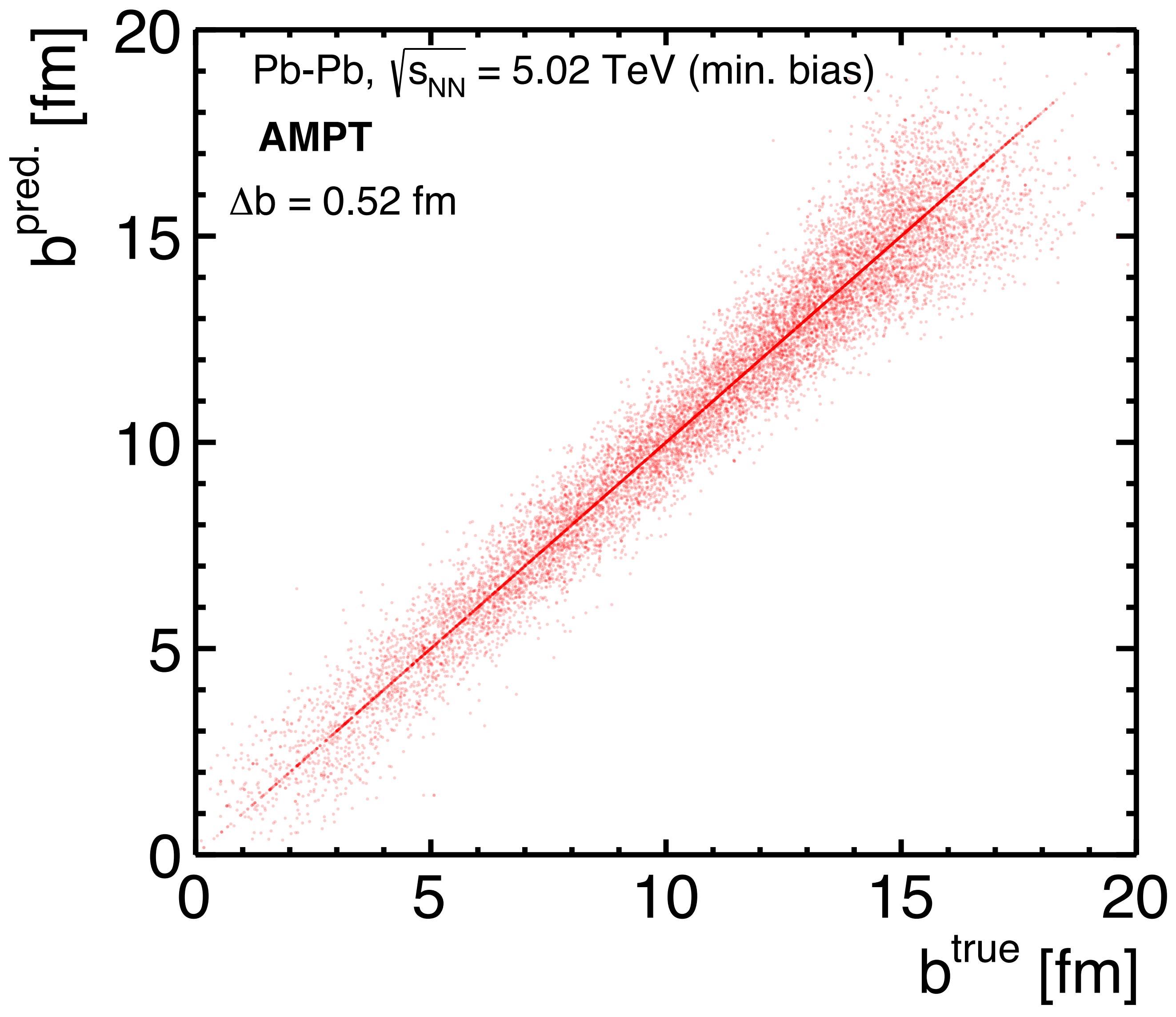
L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, Classification and Regression Trees (Wadsworth & Brooks/ Cole Advanced Books & Software, Monterey, CA, 1984), p. 358, https://doi.org/10.1002/cyto.990080516.

N. Mallick, S. Tripathy, A. N. Mishra, S. Deb, and R. Sahoo, Phys. Rev. D103, 094031 (2021)

Results

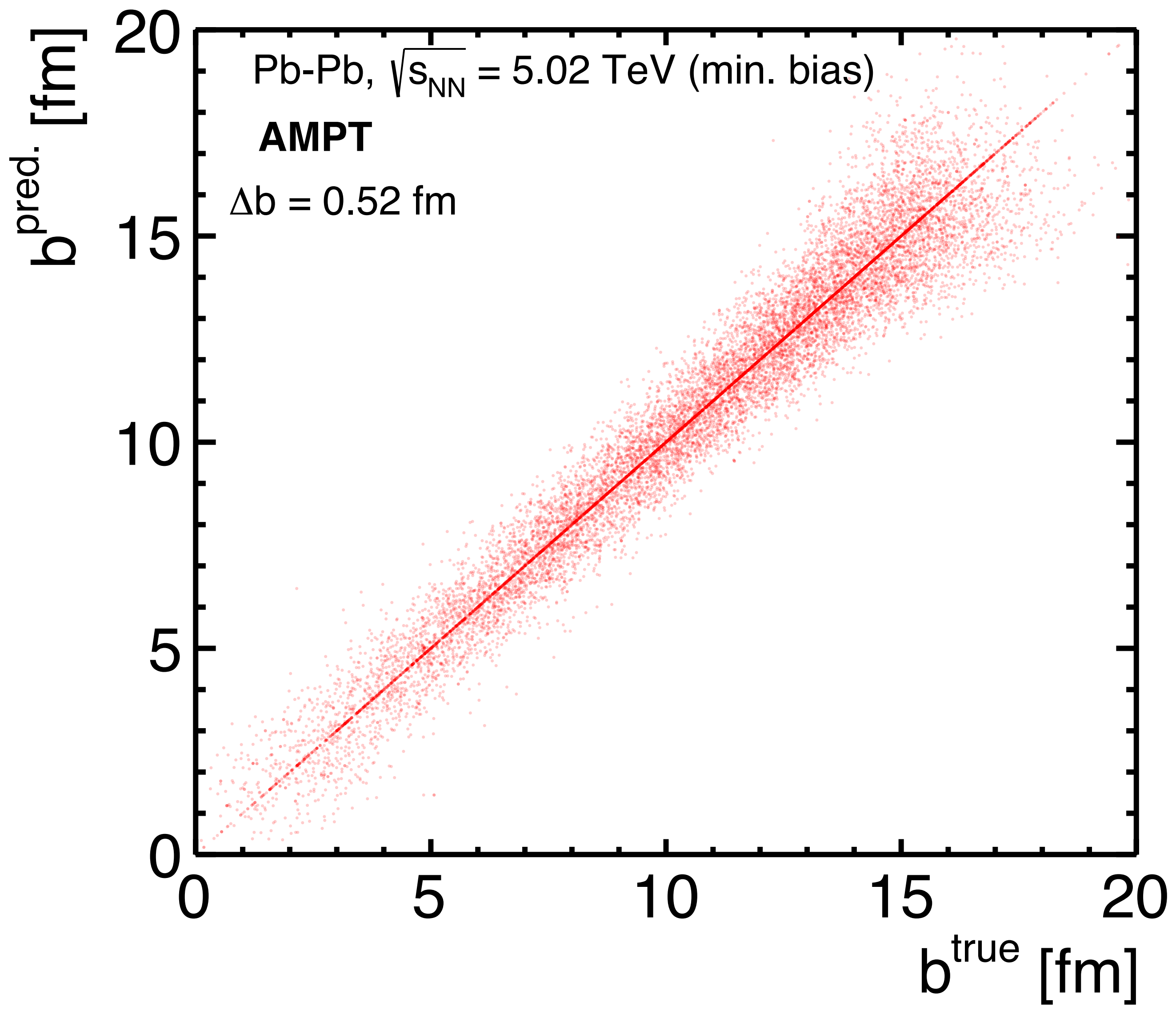
N. Mallick, S. Tripathy, A. N. Mishra, S. Deb, and R. Sahoo, Phys. Rev. D103, 094031 (2021)

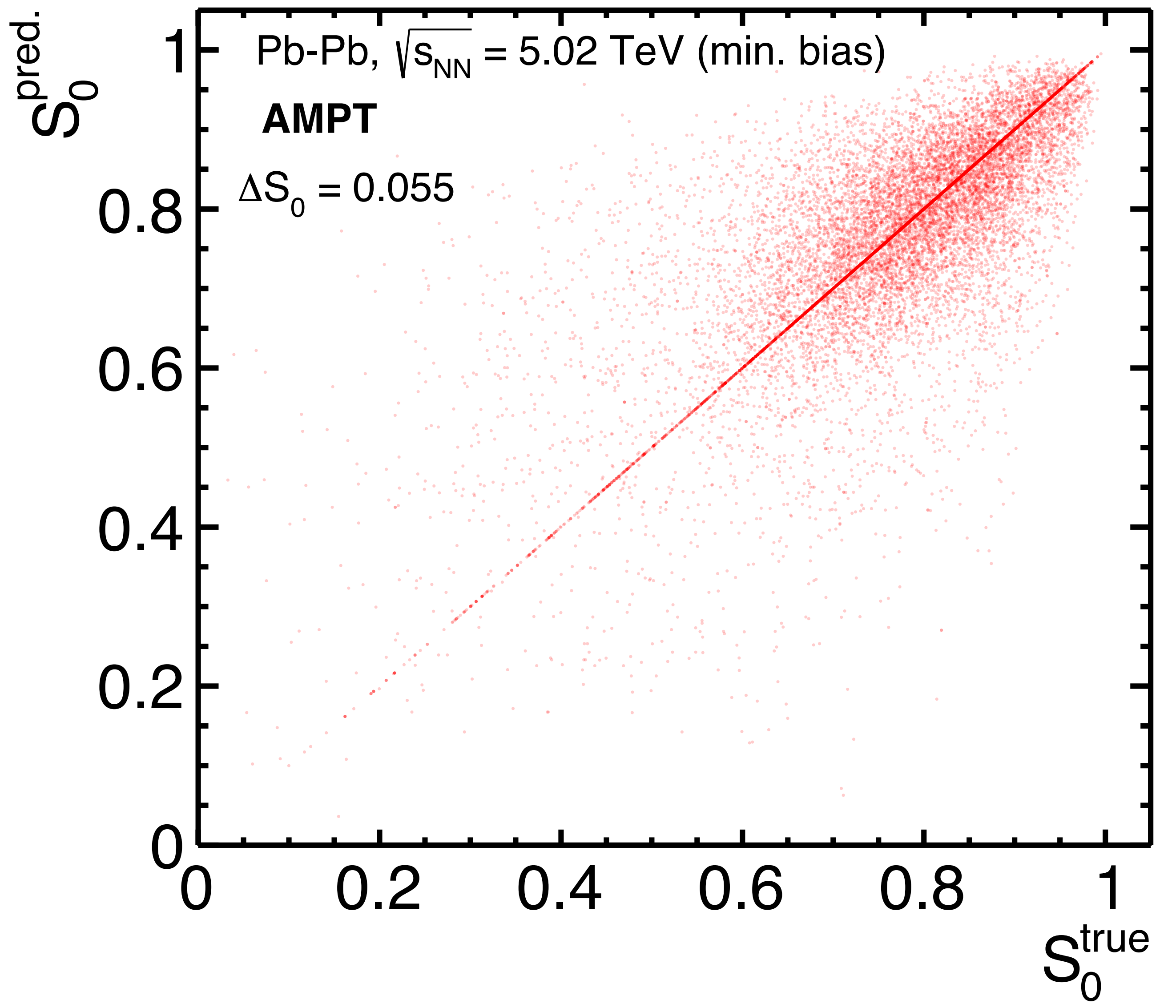
Results



N. Mallick, S. Tripathy, A. N. Mishra, S. Deb, and R. Sahoo, Phys. Rev. D103, 094031 (2021)

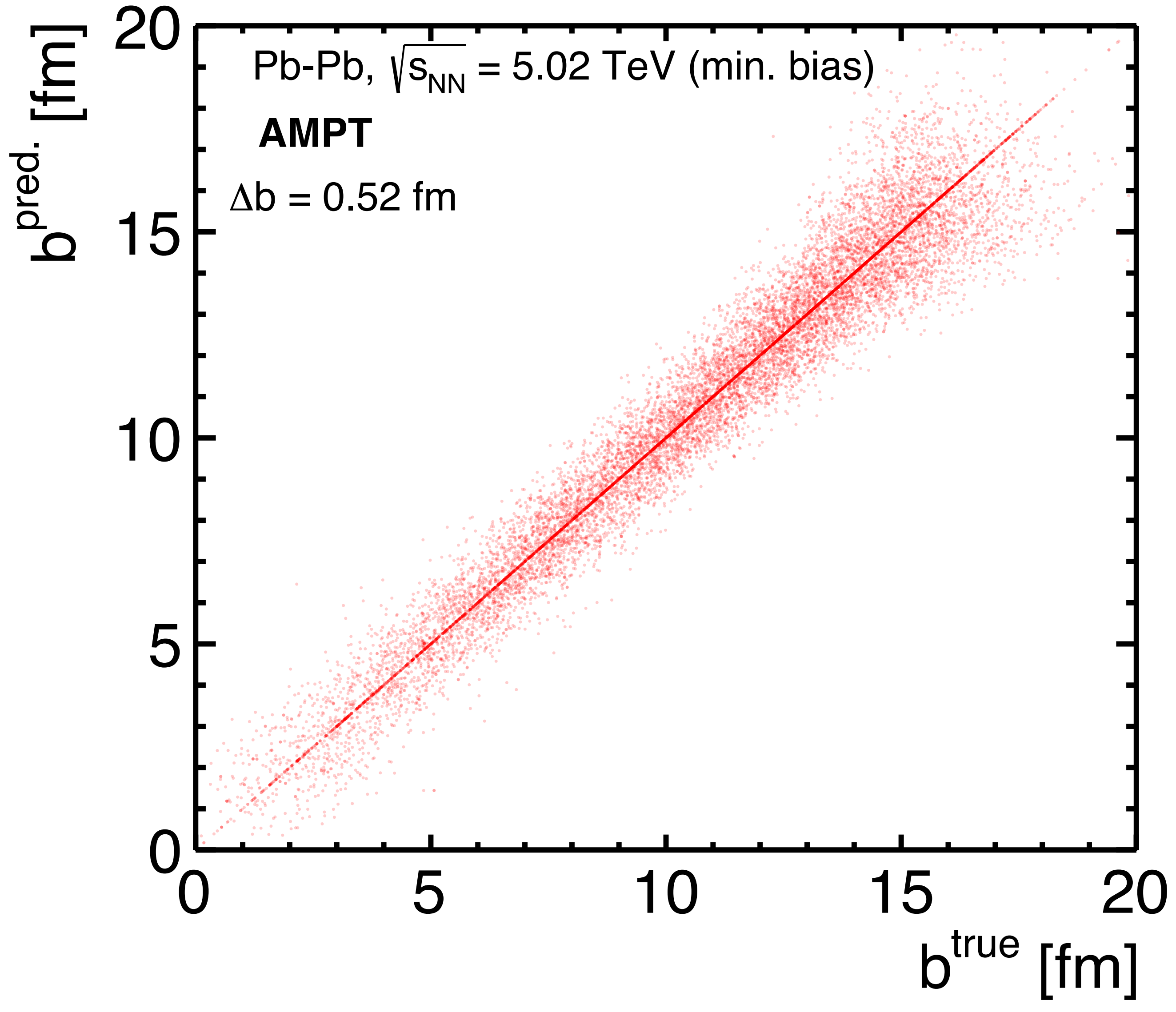
Results

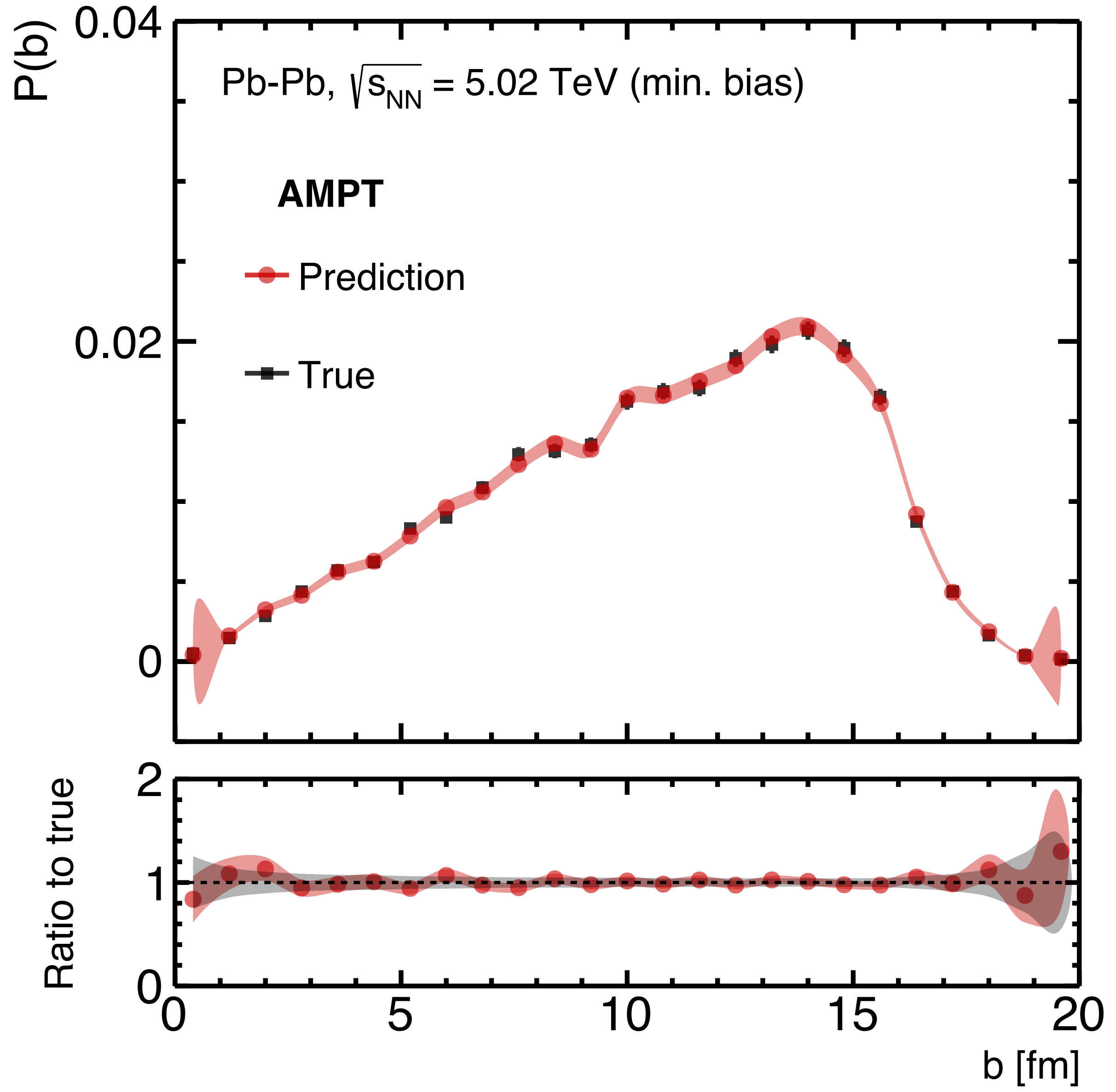


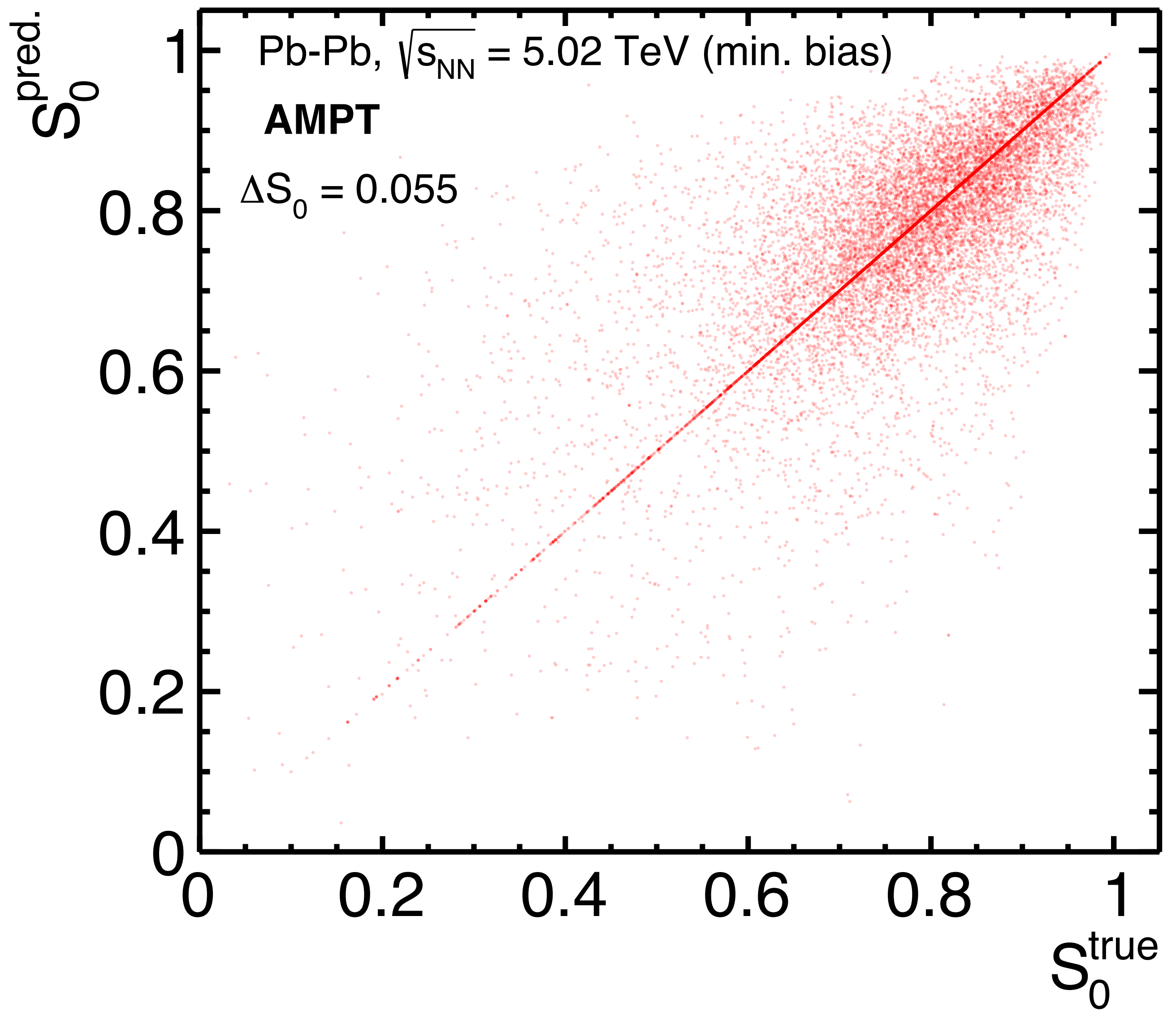


N. Mallick, S. Tripathy, A. N. Mishra, S. Deb, and R. Sahoo, Phys. Rev. D103, 094031 (2021)

Results

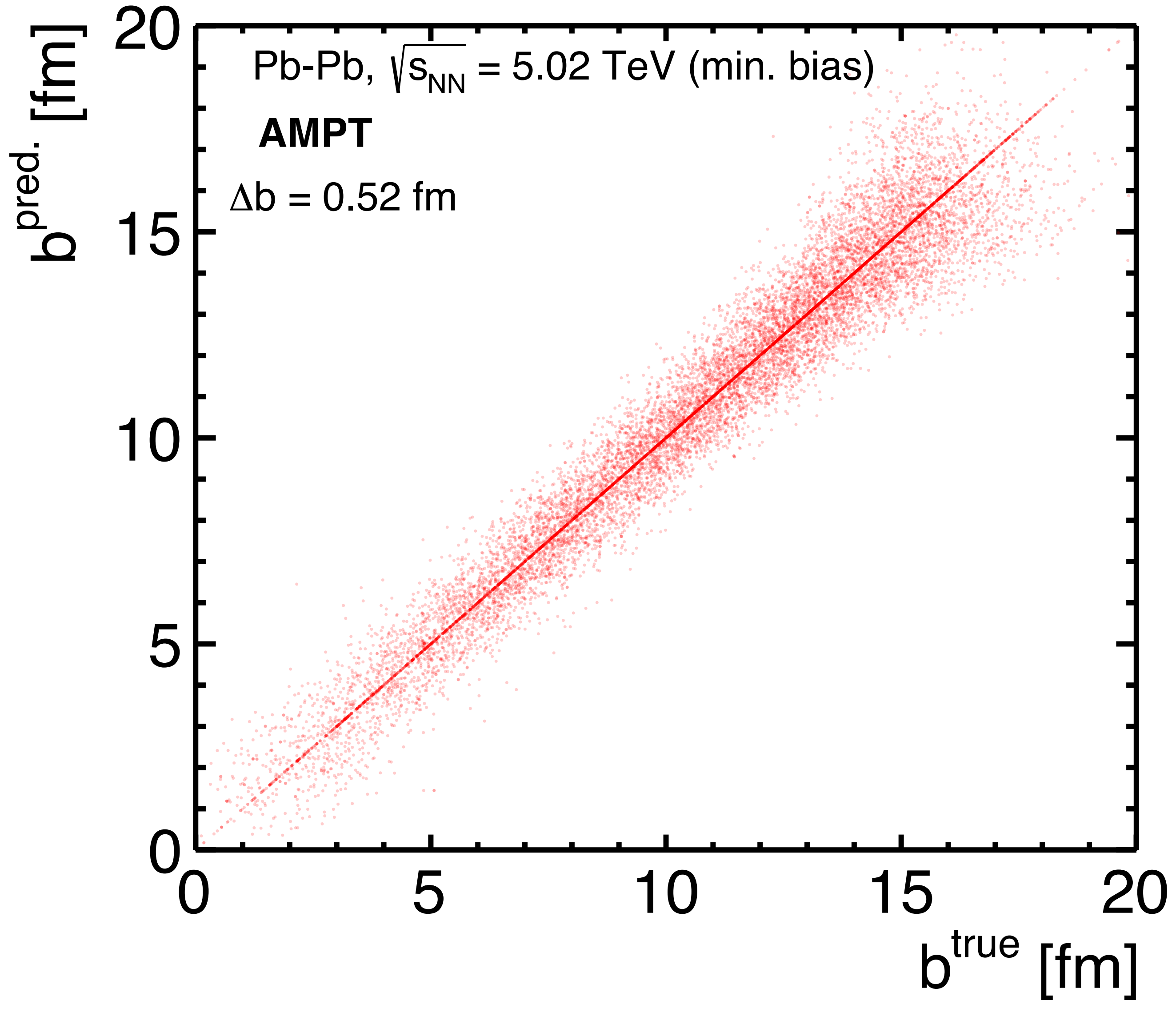


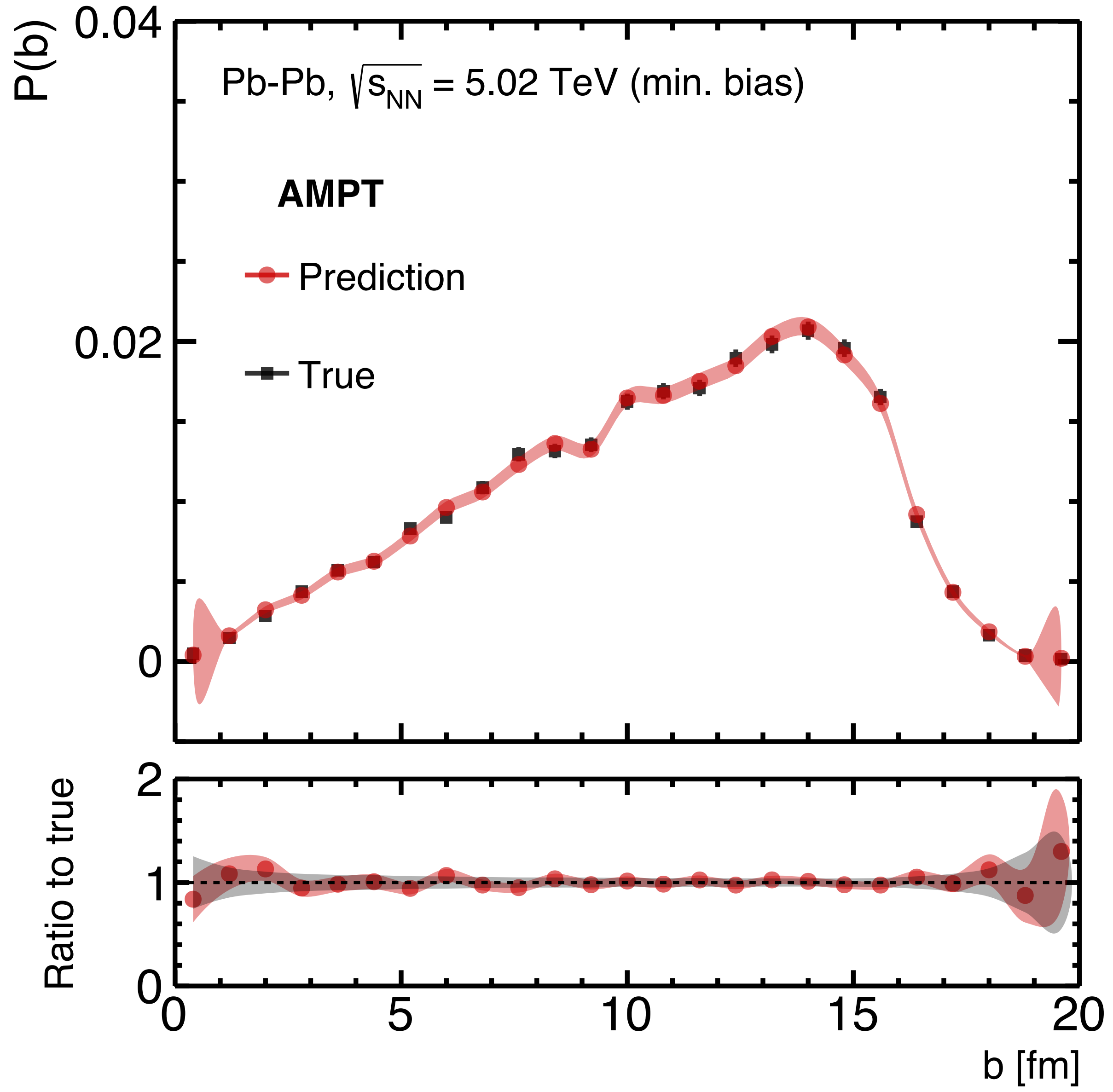


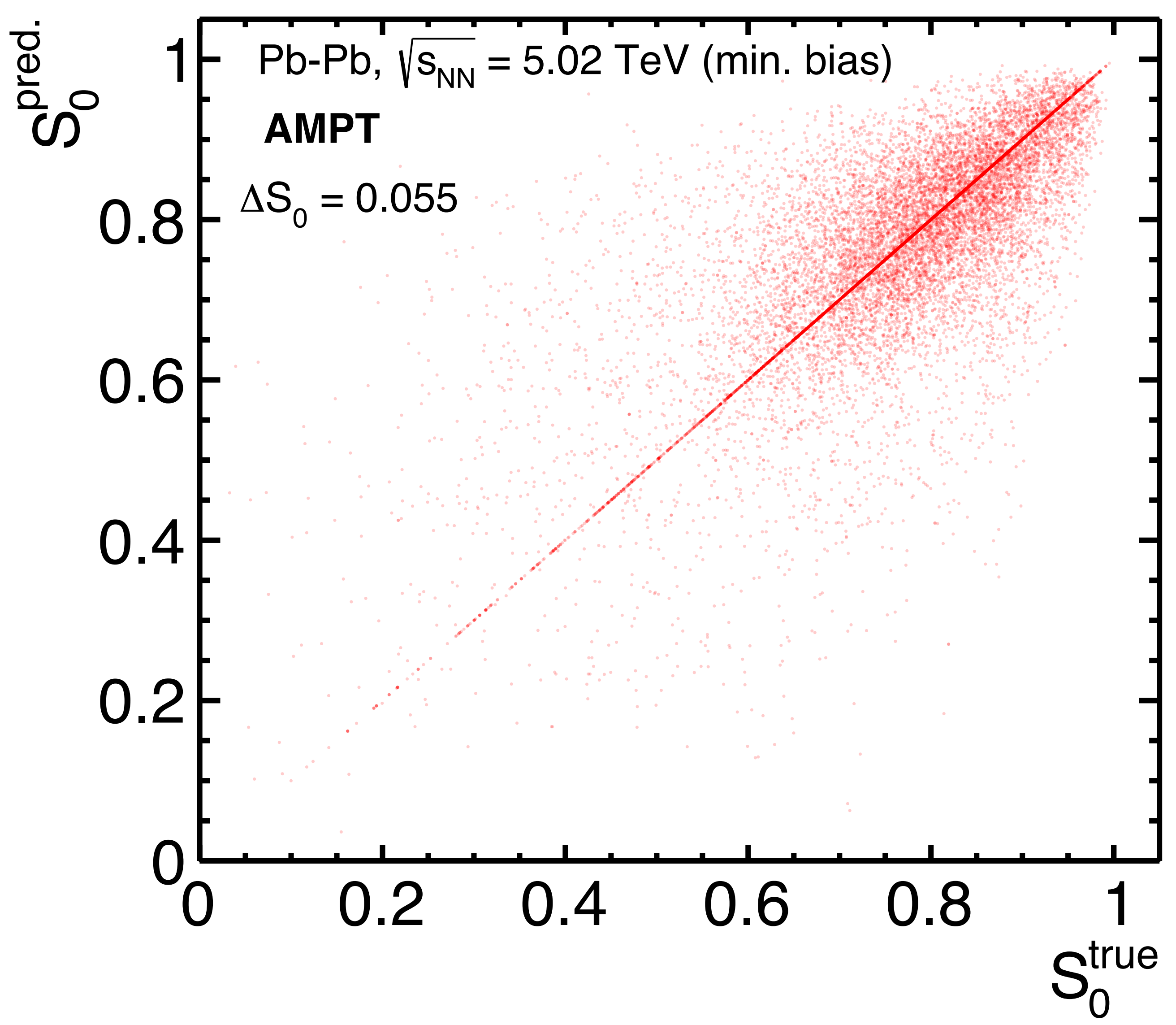


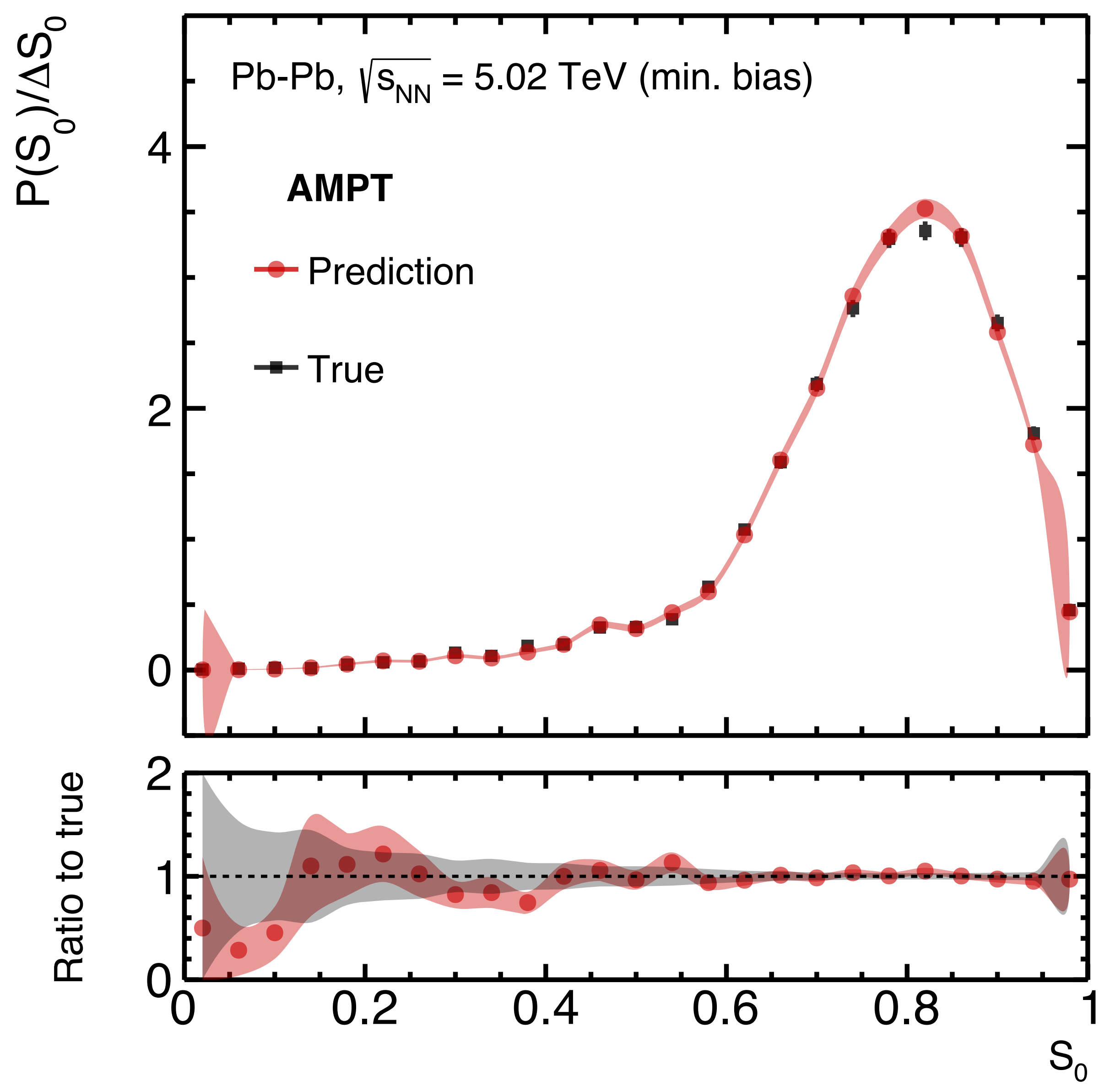
N. Mallick, S. Tripathy, A. N. Mishra, S. Deb, and R. Sahoo, Phys. Rev. D103, 094031 (2021)

Results



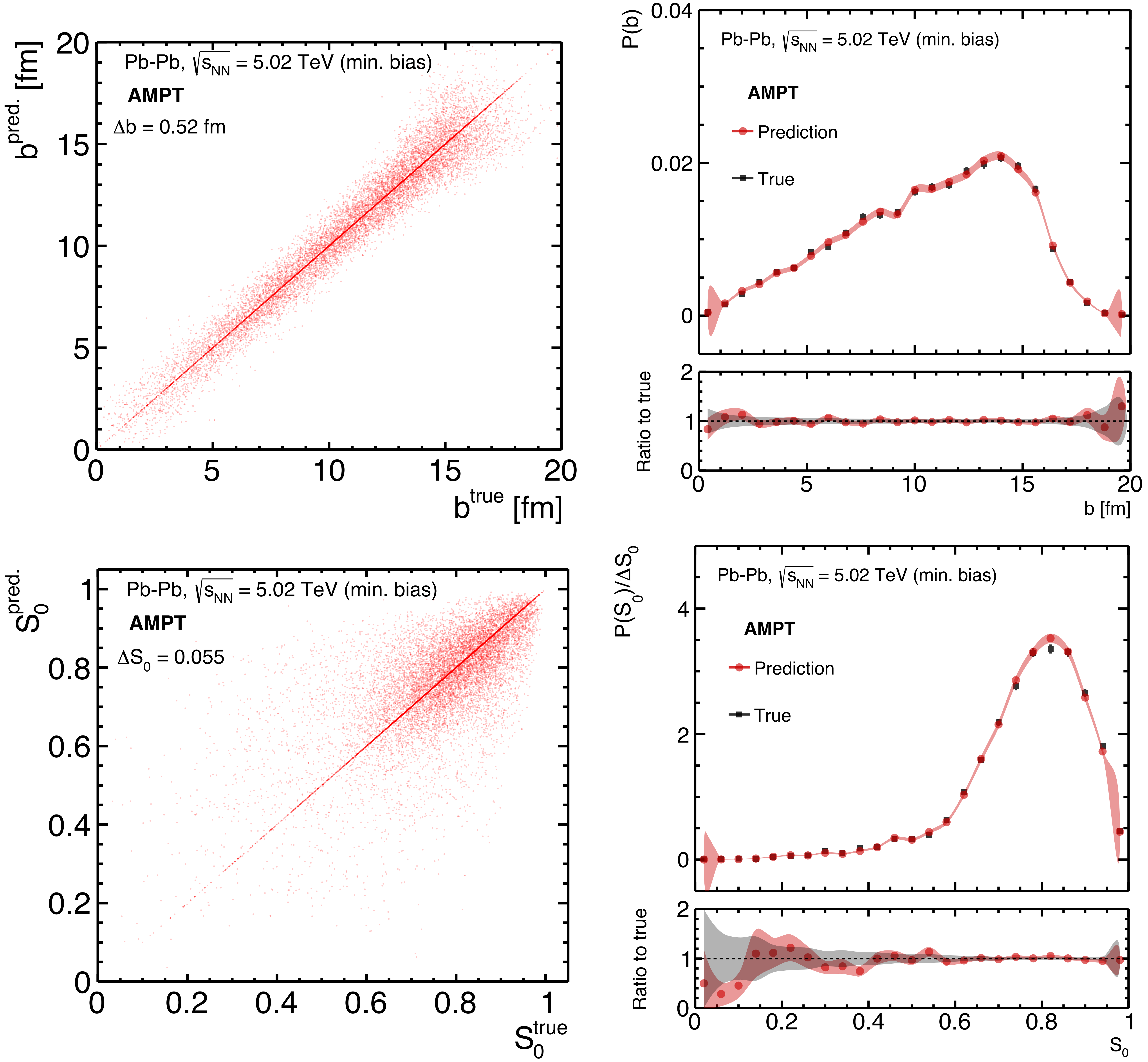






N. Mallick, S. Tripathy, A. N. Mishra, S. Deb, and R. Sahoo, Phys. Rev. D103, 094031 (2021)

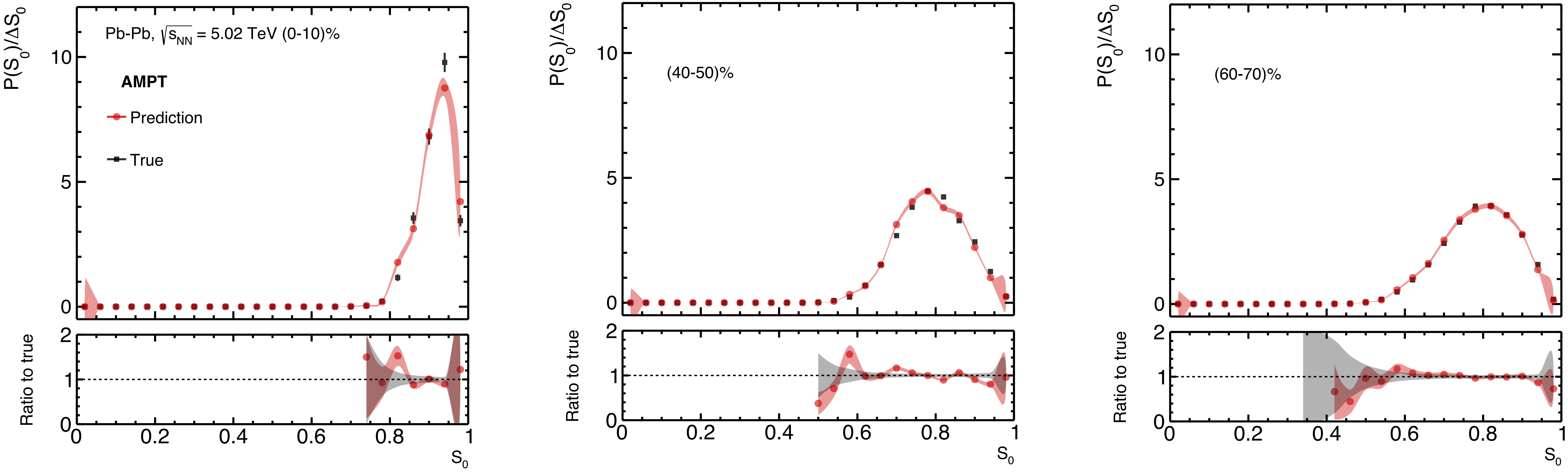
Results



- The ML model trained with 5.02 TeV minimum bias simulated data
- Most of the points populate the straight line inclined at an angle $45°$ with the *x*-axis
- The predictions for both impact parameter and spherocity distributions are in good agreement with the simulated data

N. Mallick, S. Tripathy, A. N. Mishra, S. Deb, and R. Sahoo, Phys. Rev. D103, 094031 (2021)

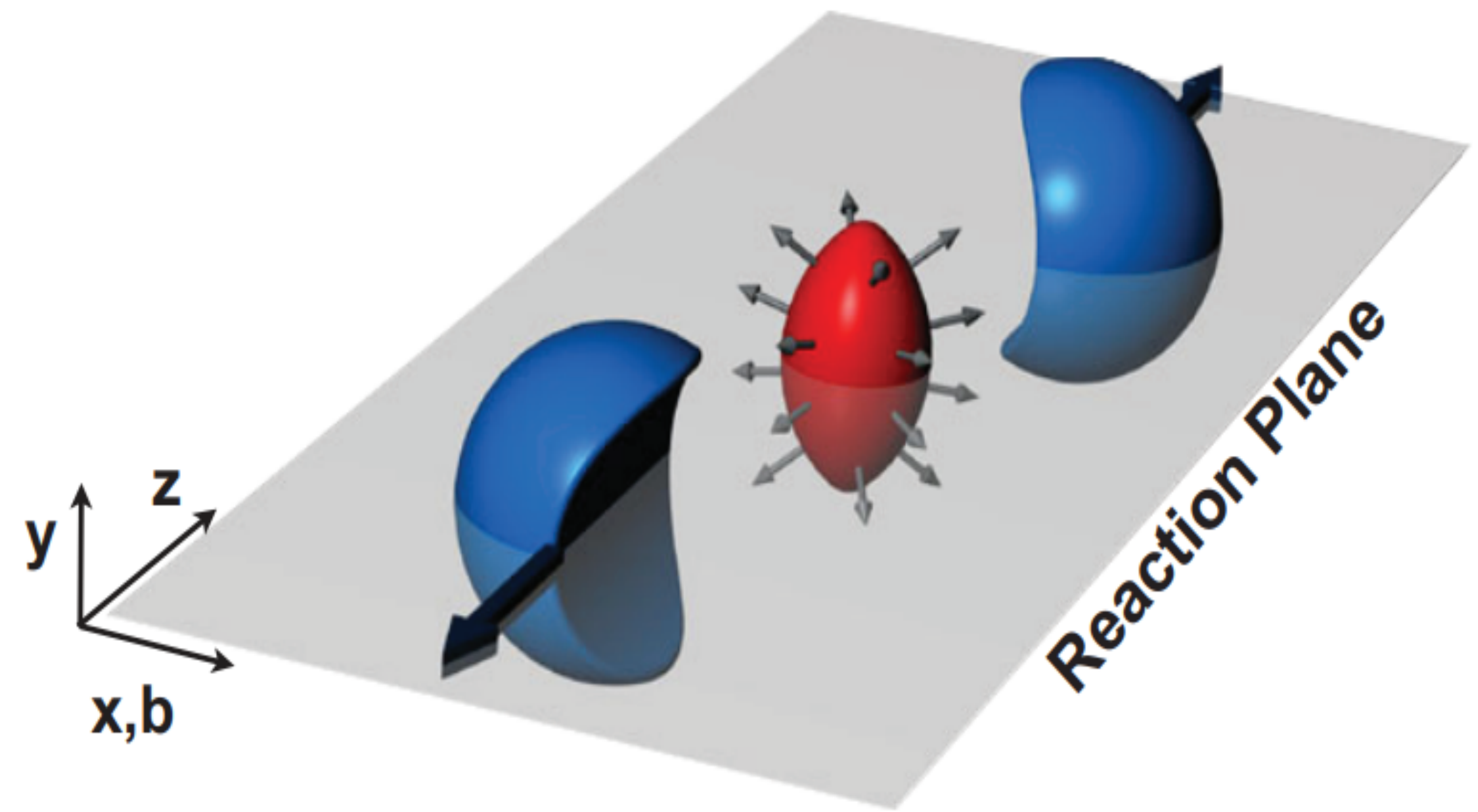
Results



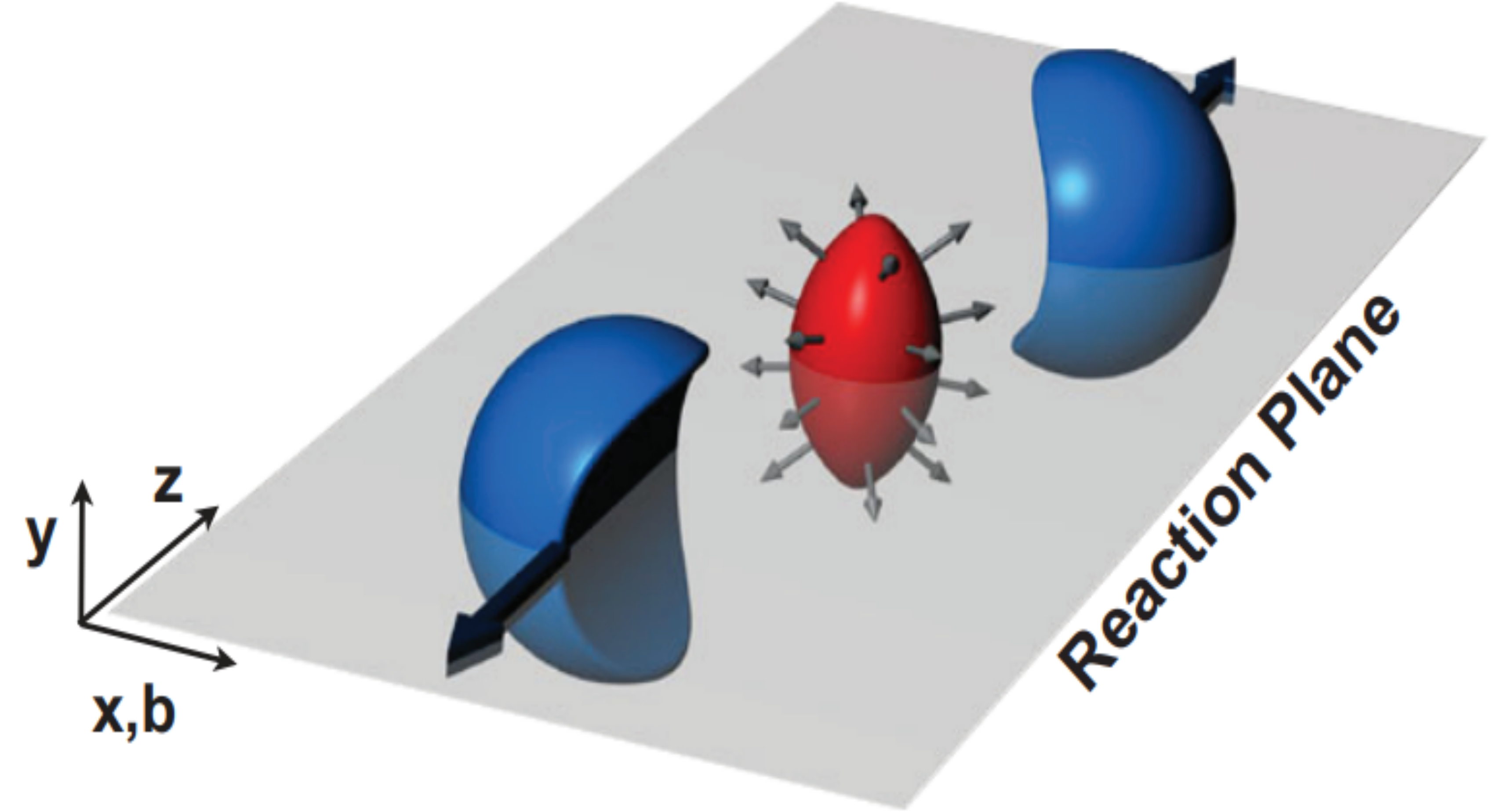
- Centrality wise spherocity distributions
- Training is done using minimum bias simulated data
- **BDT preserves the centrality (or multiplicity) dependence**

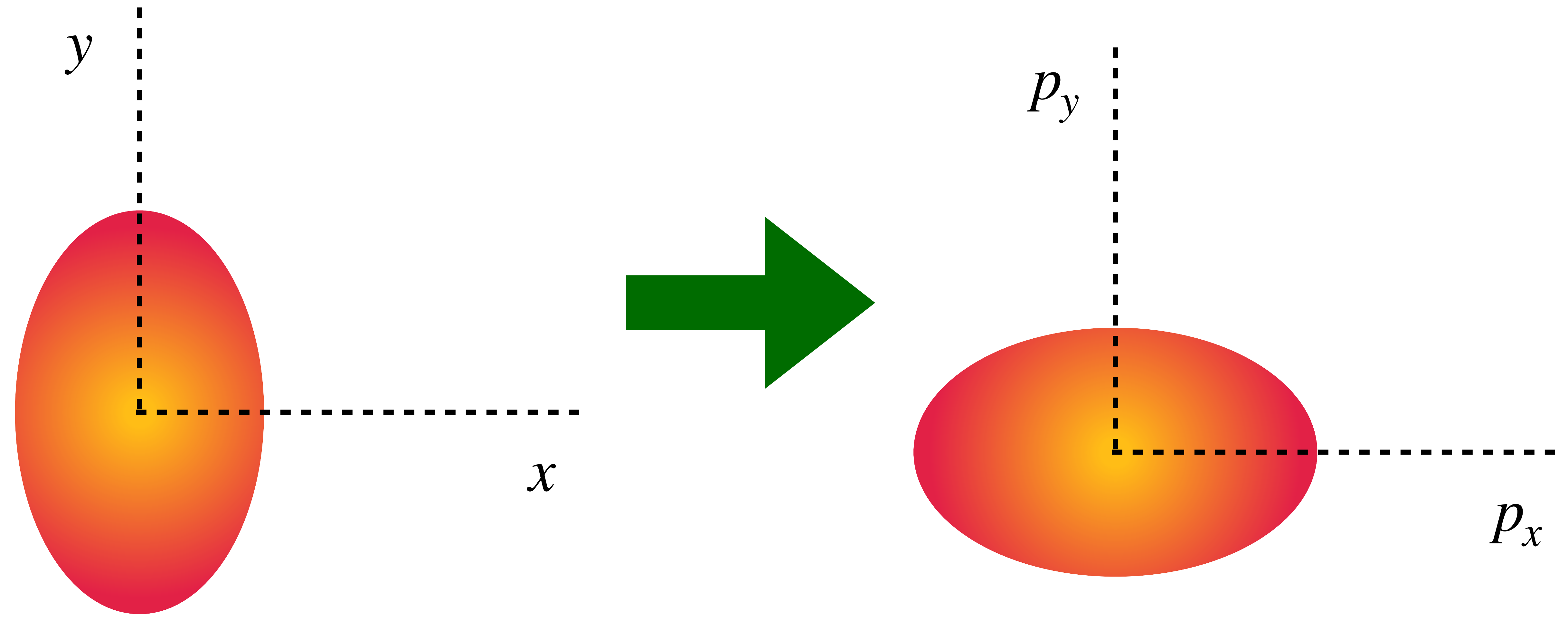
Elliptic Flow (v_2)

Elliptic Flow (v_2)



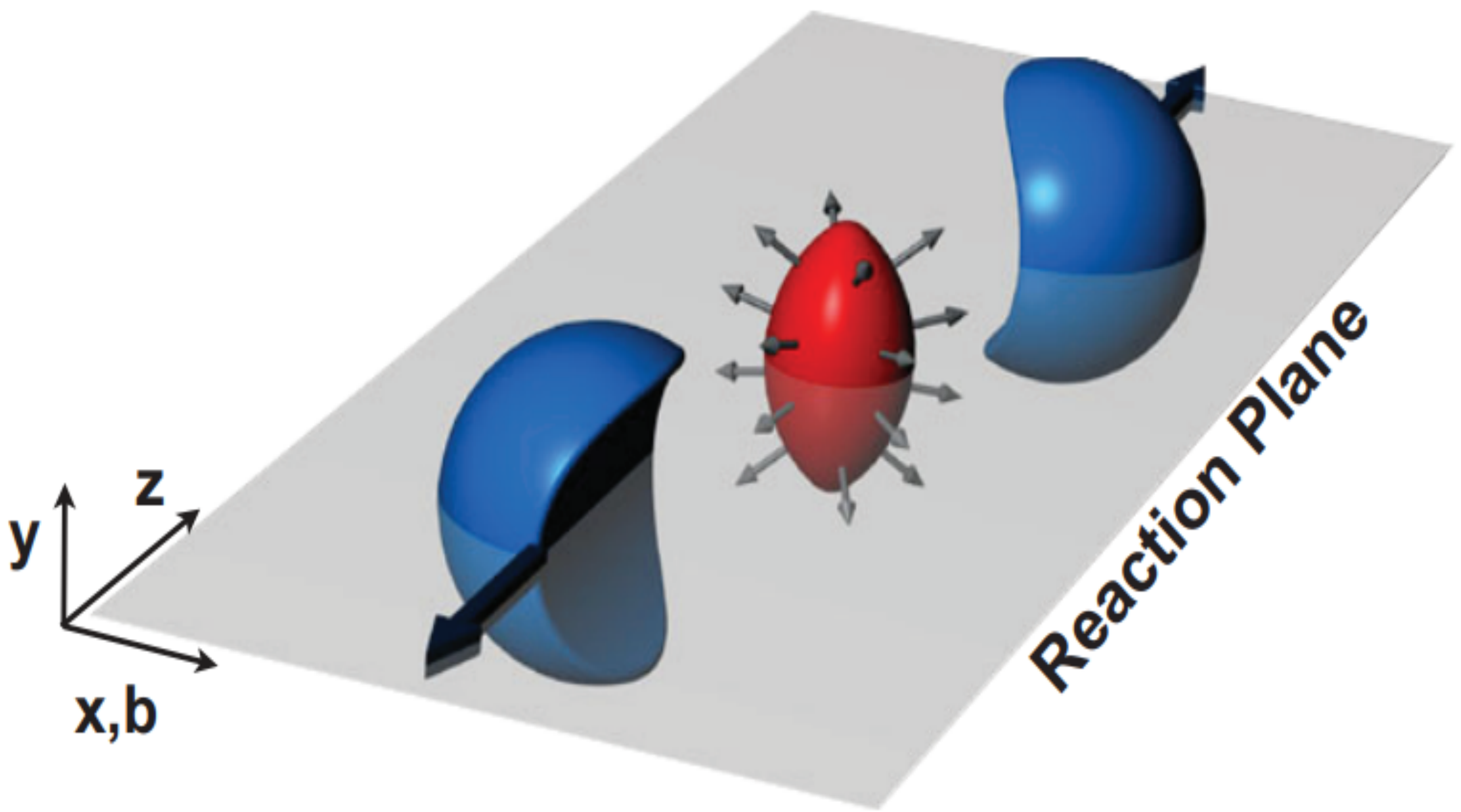
Elliptic Flow (v_2)

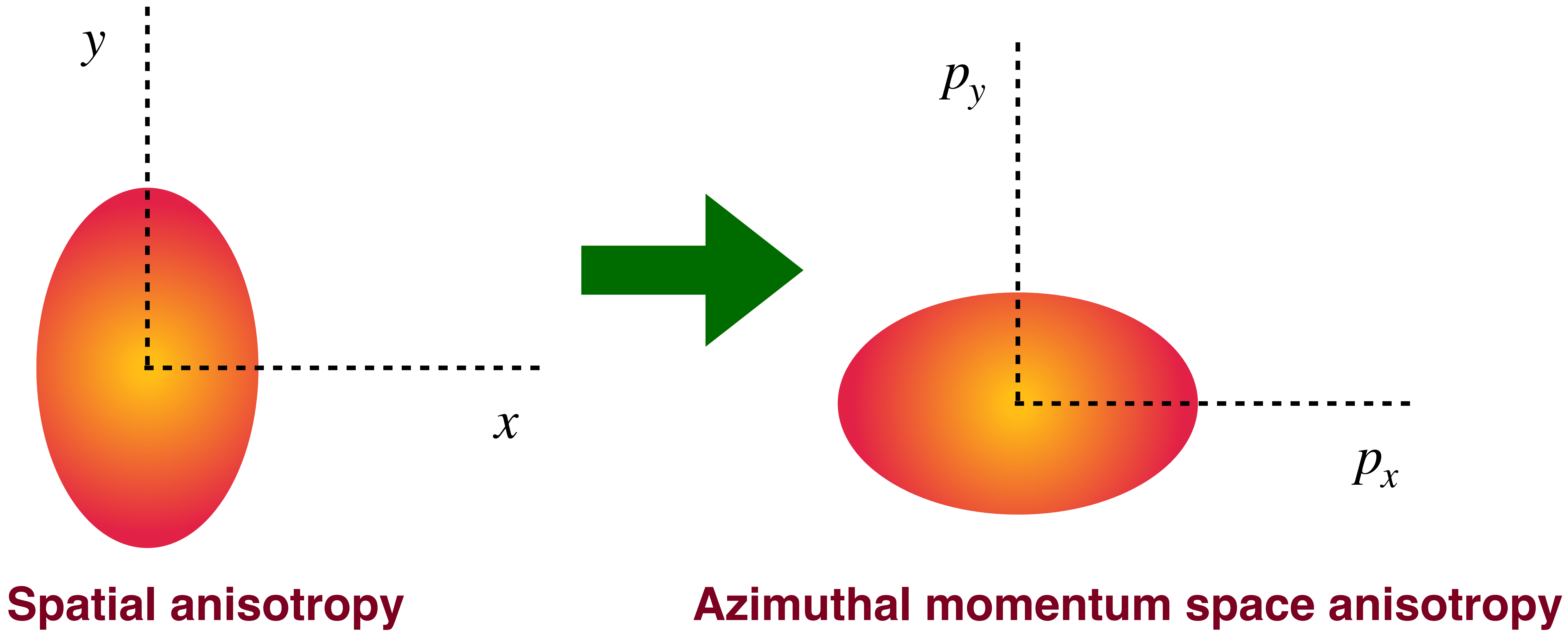




Spatial anisotropy **Azimuthal momentum space anisotropy**

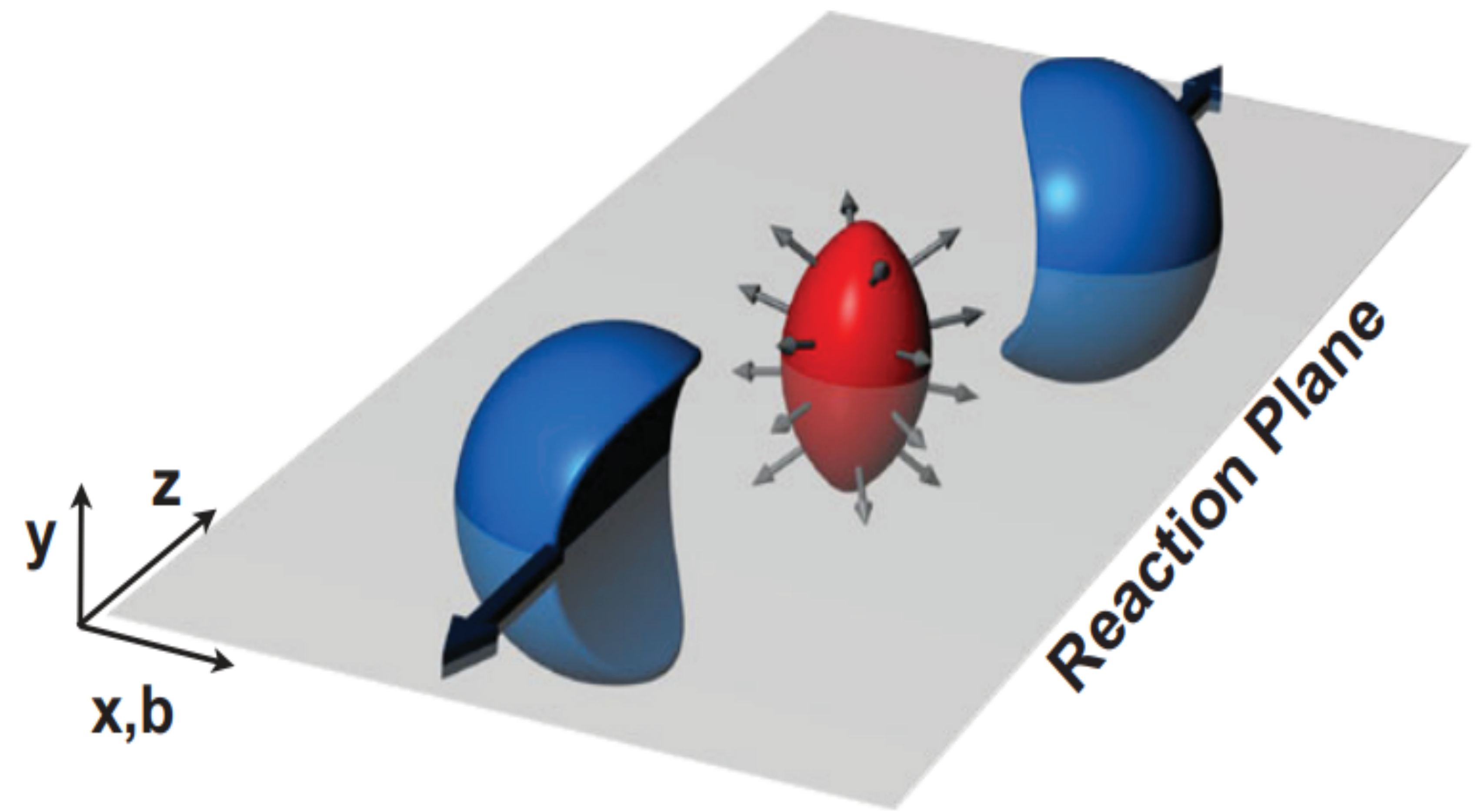
Elliptic Flow (v_2)

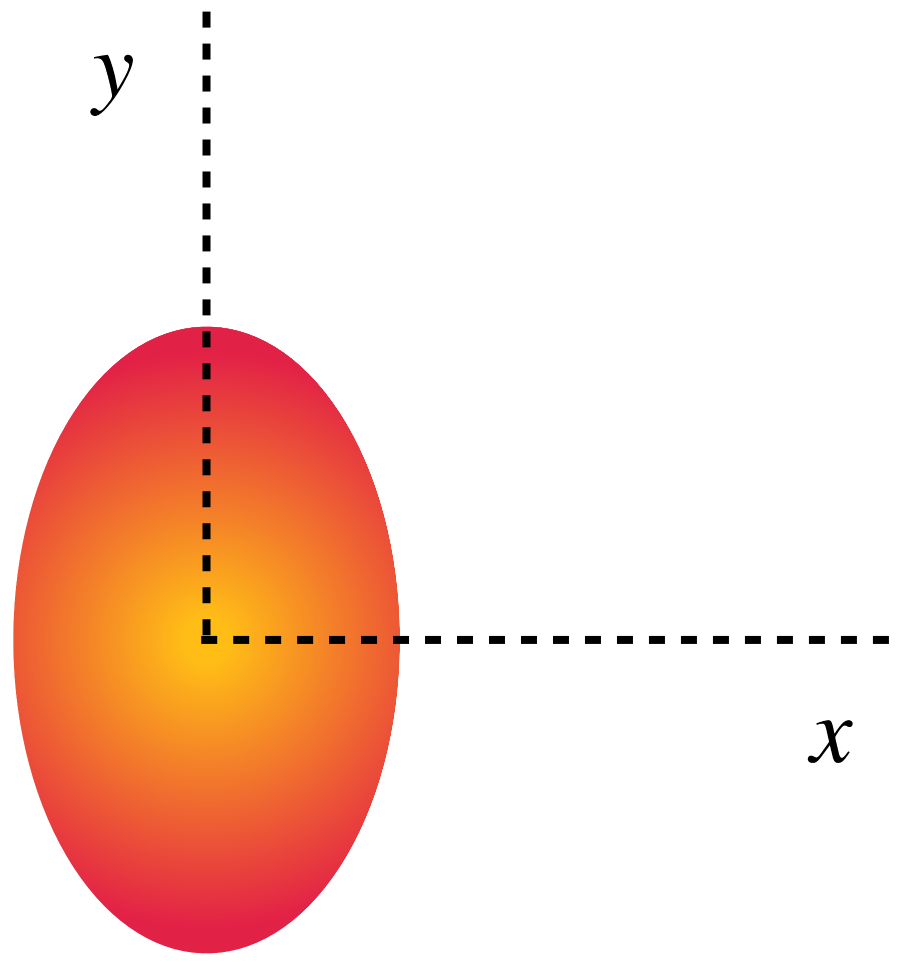




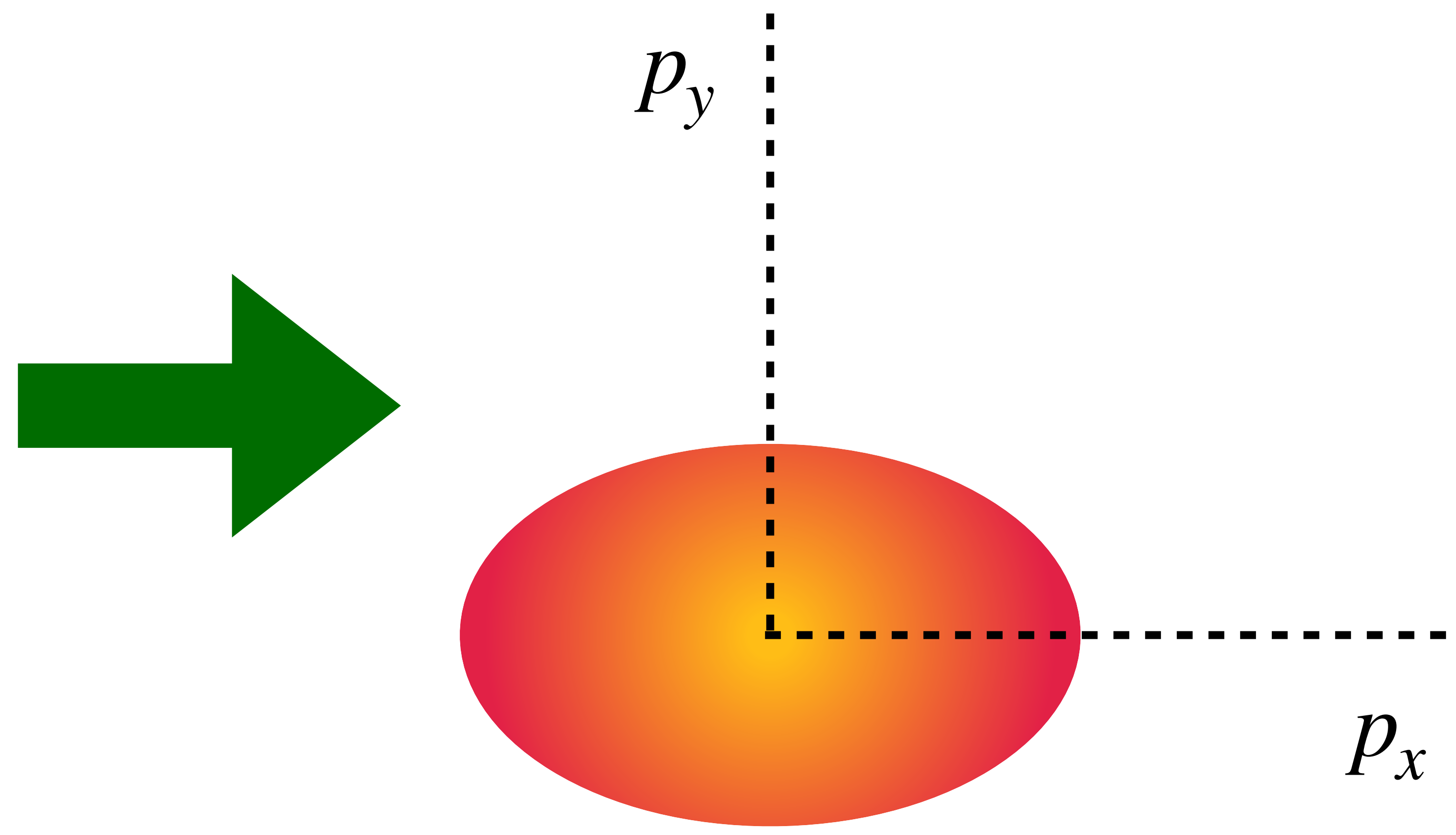
- Elliptic flow describes the azimuthal momentum space anisotropy of particle emission for a non-central heavy-ion collision
- The 2nd harmonic coefficient of the Fourier expansion of azimuthal momentum distribution $(dN/d\phi)$
- Directly reflects the initial spatial anisotropy of the nuclear overlap region in the transverse plane

Elliptic Flow (v_2)





Spatial anisotropy



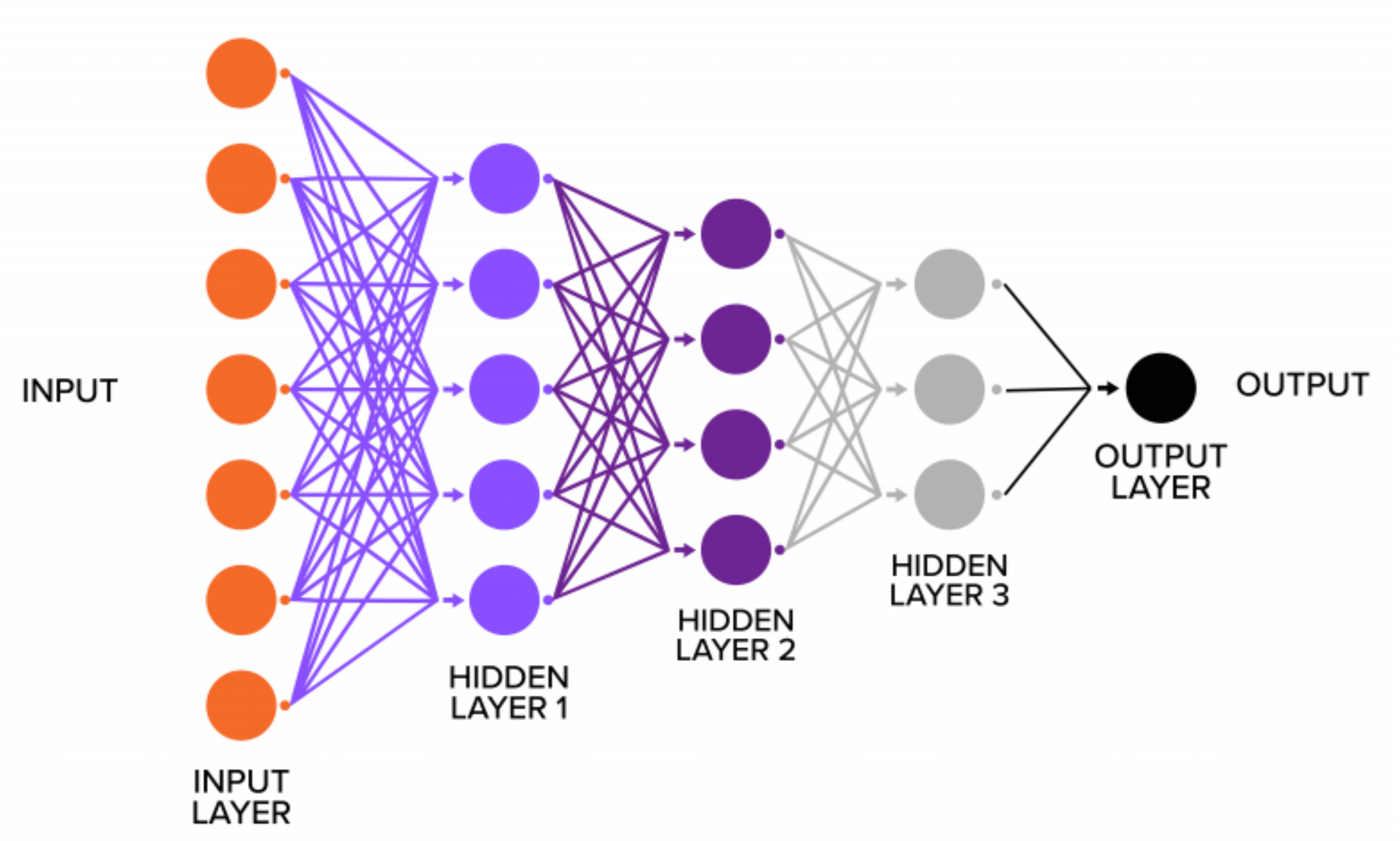
Azimuthal momentum space anisotropy

- Elliptic flow describes the azimuthal momentum space anisotropy of particle emission for a non-central heavy-ion collision
- The 2nd harmonic coefficient of the Fourier expansion of azimuthal momentum distribution ($dN/d\phi$)
- Directly reflects the initial spatial anisotropy of the nuclear overlap region in the transverse plane

$$E\frac{d^3N}{dp^3} = \frac{d^3N}{p_{\mathrm{T}}dp_{\mathrm{T}}dyd\phi} = \frac{d^2N}{p_{\mathrm{T}}dp_{\mathrm{T}}dy}\frac{1}{2\pi}\left(1 + 2\sum_{n=1}^{\infty} v_n \cos[n(\phi - \psi_n)]\right)$$

$$v_2(p_{\mathrm{T}}, y) = \langle \cos(2(\phi - \psi_2))\rangle$$
$$\phi = \tan^{-1}(p_y/p_x)$$

Deep Neural Network (DNN)



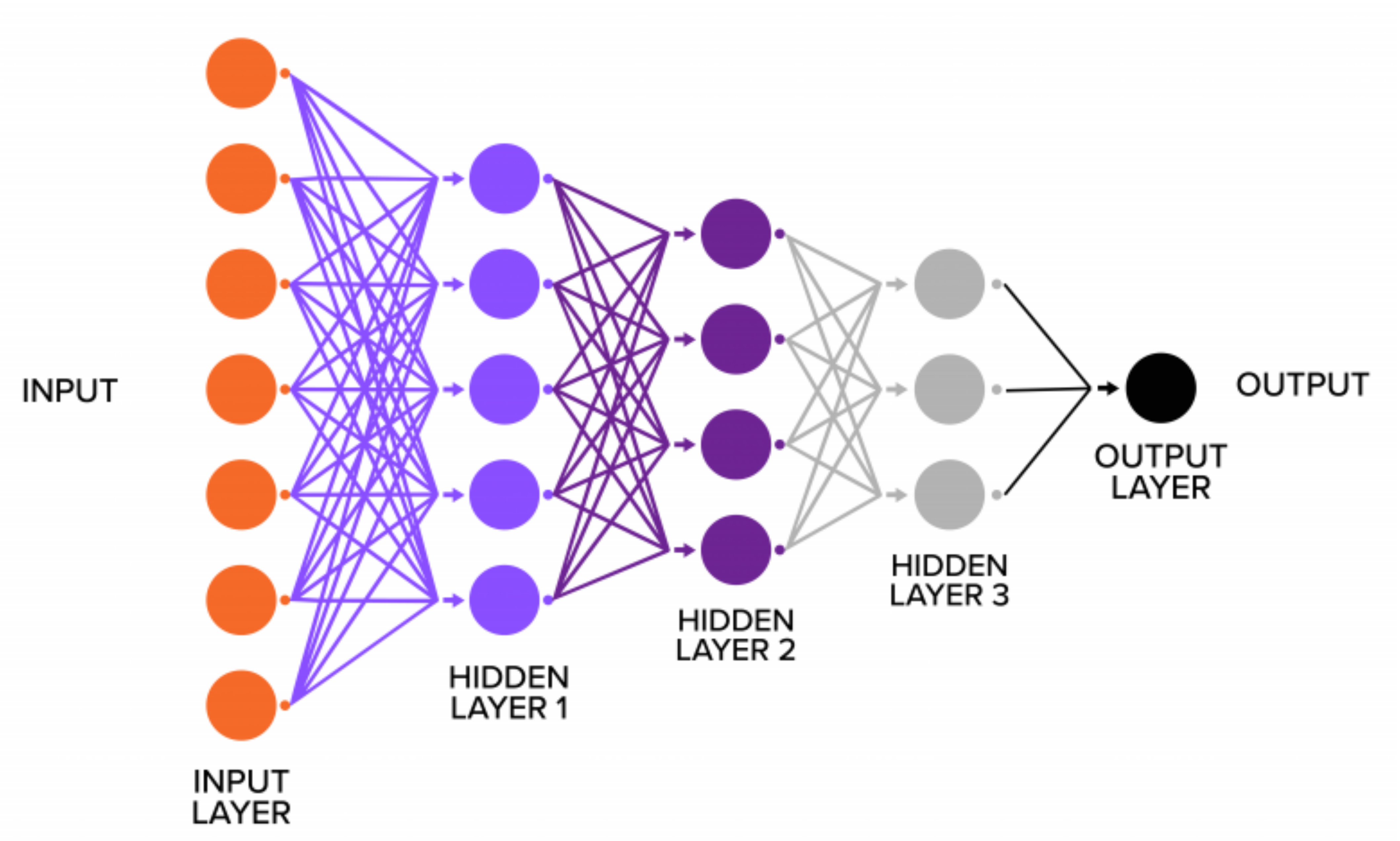
Deep Neural Network (DNN)

- ML Algorithm inspired from neurons in animal brains
- Three key layers

  Input: Takes the features as input

  Hidden layers: Connects to each neuron through different weights

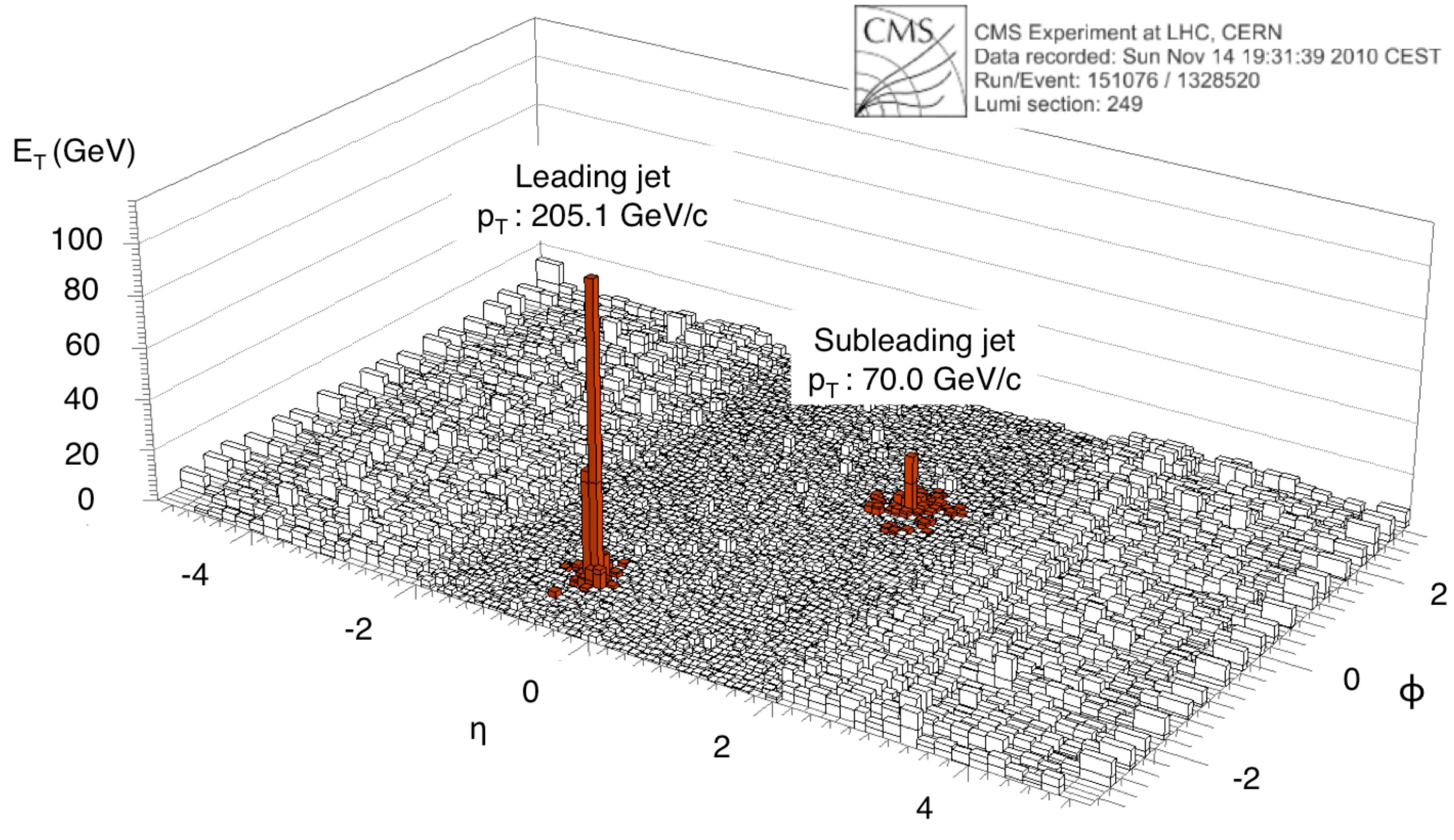
  Output: Gives the result as a number or class
- **Weights** dictate the importance of an input → more important features get more weights
- **Activation function**: mathematical function that guides the outcome at each node → Standardize the values
- **Cost function**: Evaluates the accuracy between machine prediction and true value
- **Optimizer**: Method (or algorithm) that minimizes the cost function by automatically updating the weights



Estimation of elliptic flow (v_2)

N. Mallick, S. Prasad, A. N. Mishra, R. Sahoo, and G. G. Barnaföldi (In preparation)

Estimation of elliptic flow (v_2)

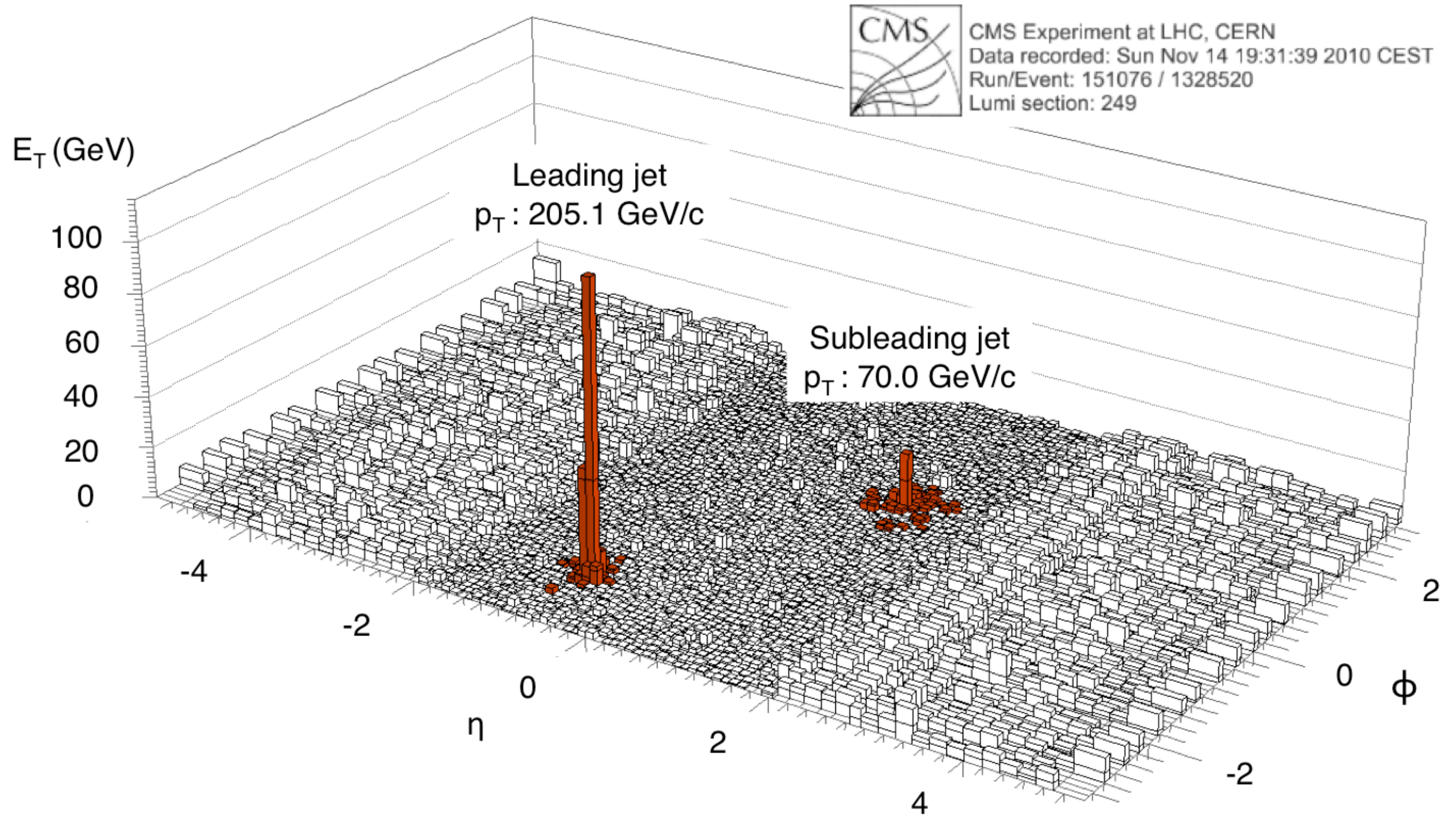


Serguei Chatrchyan et al., Phys.Rev.C 84 (2011), 024906

N. Mallick, S. Prasad, A. N. Mishra, R. Sahoo, and G. G. Barnaföldi (In preparation)

Estimation of elliptic flow (v_2)

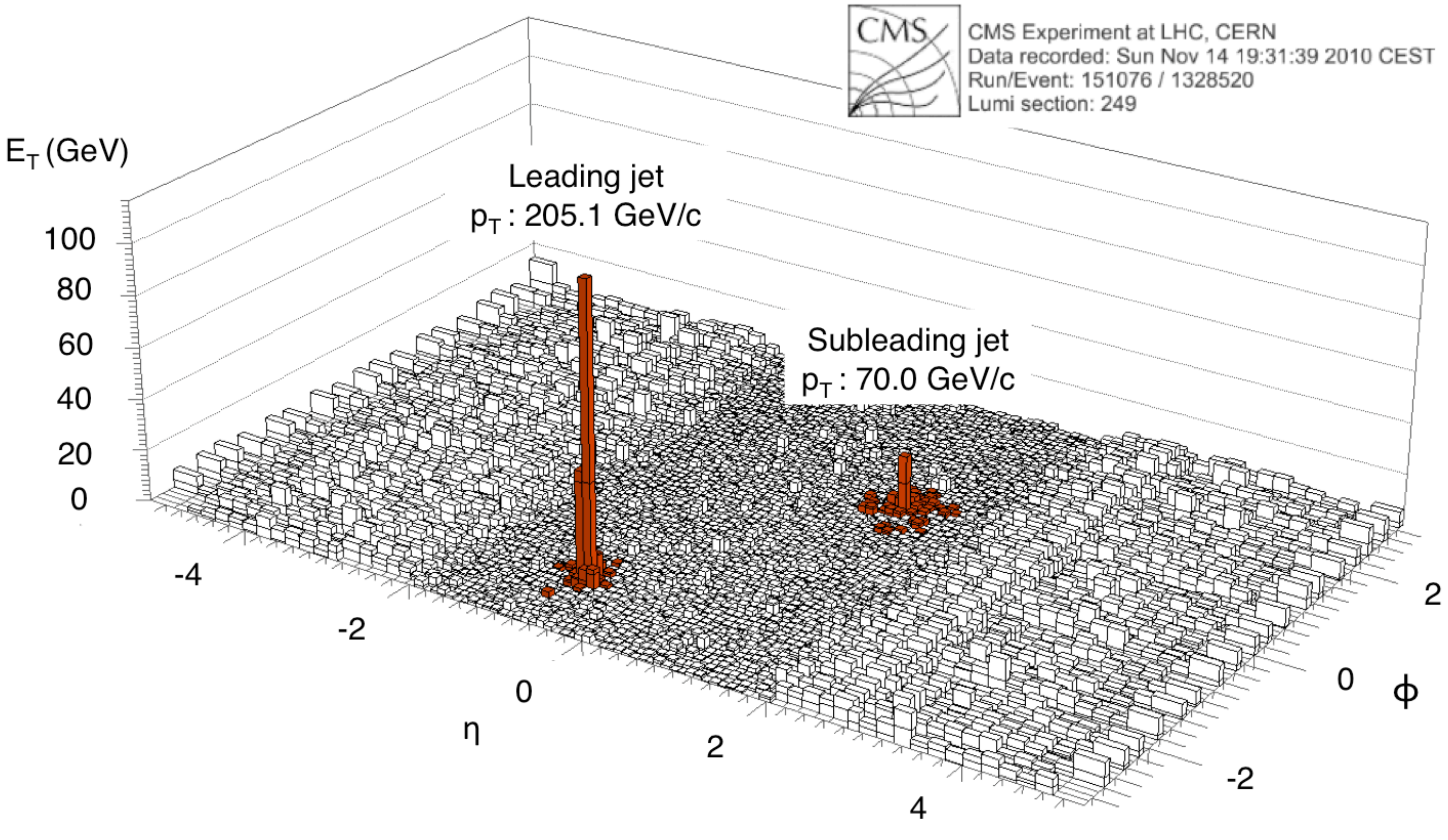
- Estimation of elliptic flow using Deep Neural Network
- Elliptic flow -> Event property
- Inputs ->Track property
- $(\eta - \phi)$ space could be taken as the primary input space
- Three layers having different weights
- $p_{\rm T}$, mass and $\log(\sqrt{s_{\rm NN}/s_0})$ weighted layers serve as the secondary input space



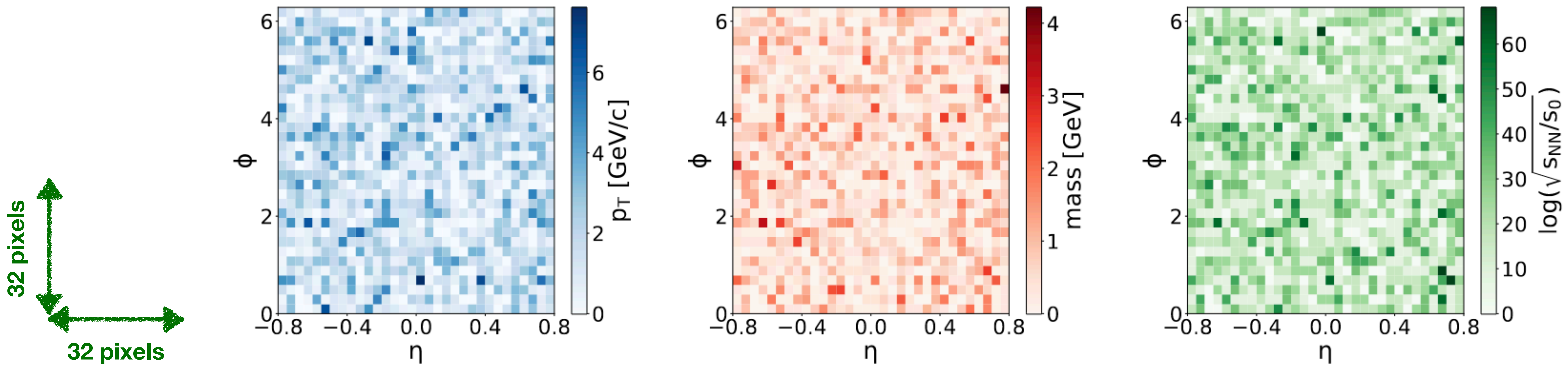
Serguei Chatrchyan et al., Phys.Rev.C 84 (2011), 024906

Estimation of elliptic flow (v_2)

- Estimation of elliptic flow using Deep Neural Network
- Elliptic flow -> Event property
- Inputs ->Track property
- $(\eta - \phi)$ space could be taken as the primary input space
- Three layers having different weights
- $p_{\rm T}$, mass and $\log(\sqrt{s_{\rm NN}/s_0})$ weighted layers serve as the secondary input space



Serguei Chatrchyan et al., Phys.Rev.C 84 (2011), 024906



Pb-Pb, $\sqrt{s_{\rm NN}} = 5.02$ TeV, **AMPT Simulation**

N. Mallick, S. Prasad, A. N. Mishra, R. Sahoo, and G. G. Barnaföldi (In preparation)

DNN Model:

DNN Model:

- Each space has 32×32 pixels (grids)
- Total number of pixel points = $32 \times 32 \times 3 = 3072$ for each event
- DNN with the following architecture

  Input Layer: 128 Nodes

  Three hidden layers: 256 Nodes each

  Final layer : 1 node (v_2)

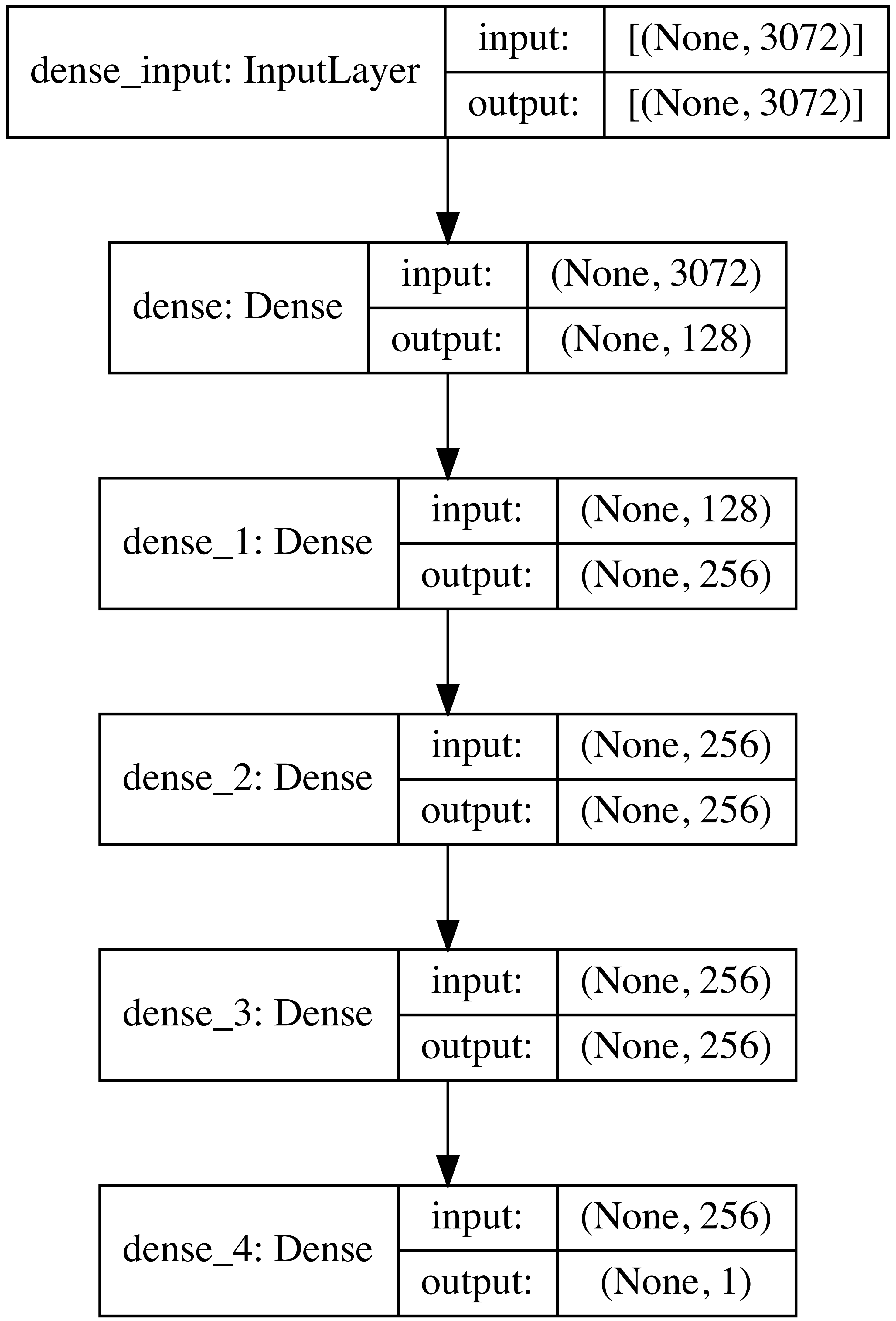
DNN Model:

- Each space has 32×32 pixels (grids)
- Total number of pixel points = $32 \times 32 \times 3 = 3072$ for each event
- DNN with the following architecture

  Input Layer: 128 Nodes

  Three hidden layers: 256 Nodes each

  Final layer : 1 node (v_2)



DNN Model:

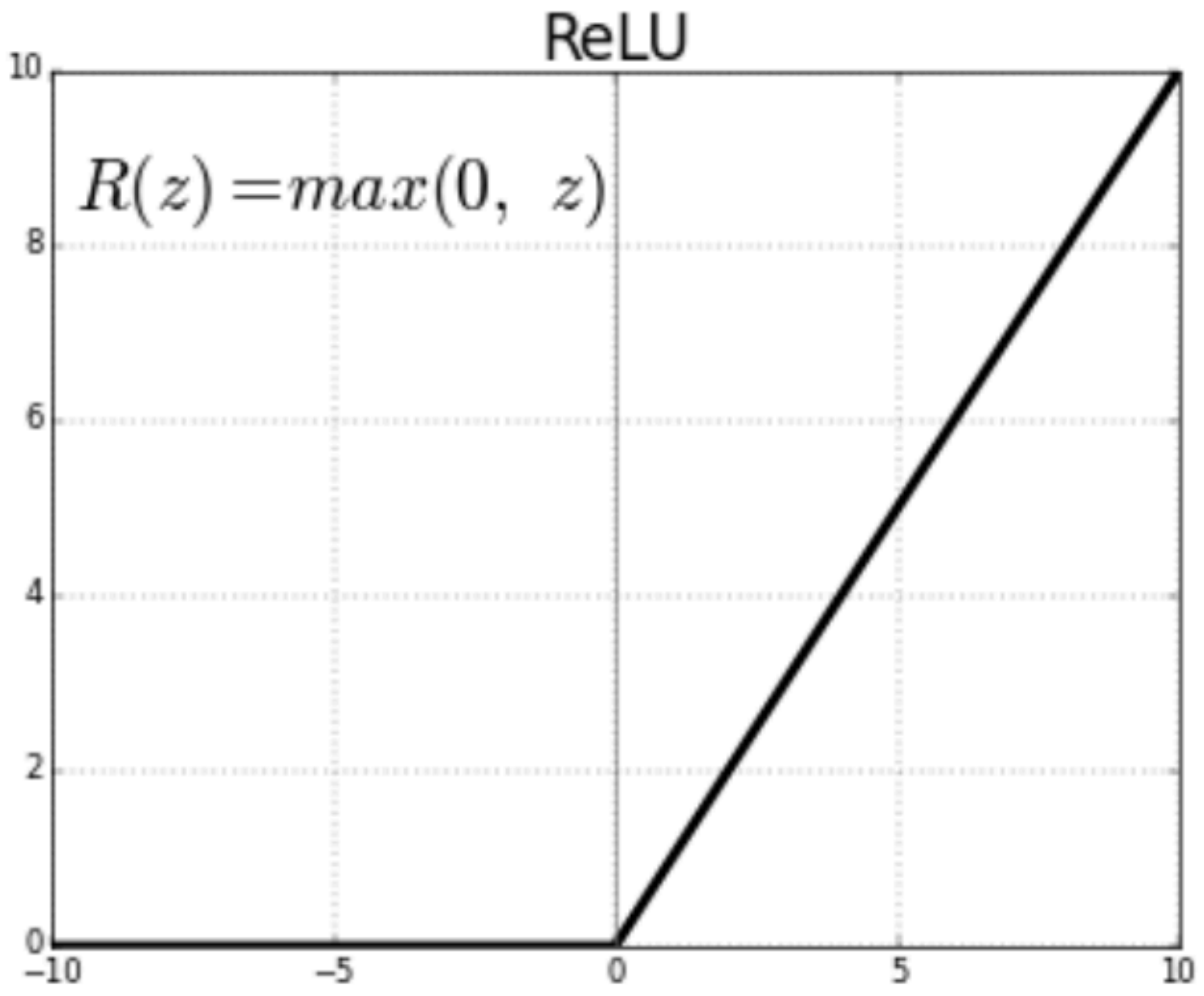
- Each space has 32×32 pixels (grids)
- Total number of pixel points = $32 \times 32 \times 3 = 3072$ for each event
- DNN with the following architecture

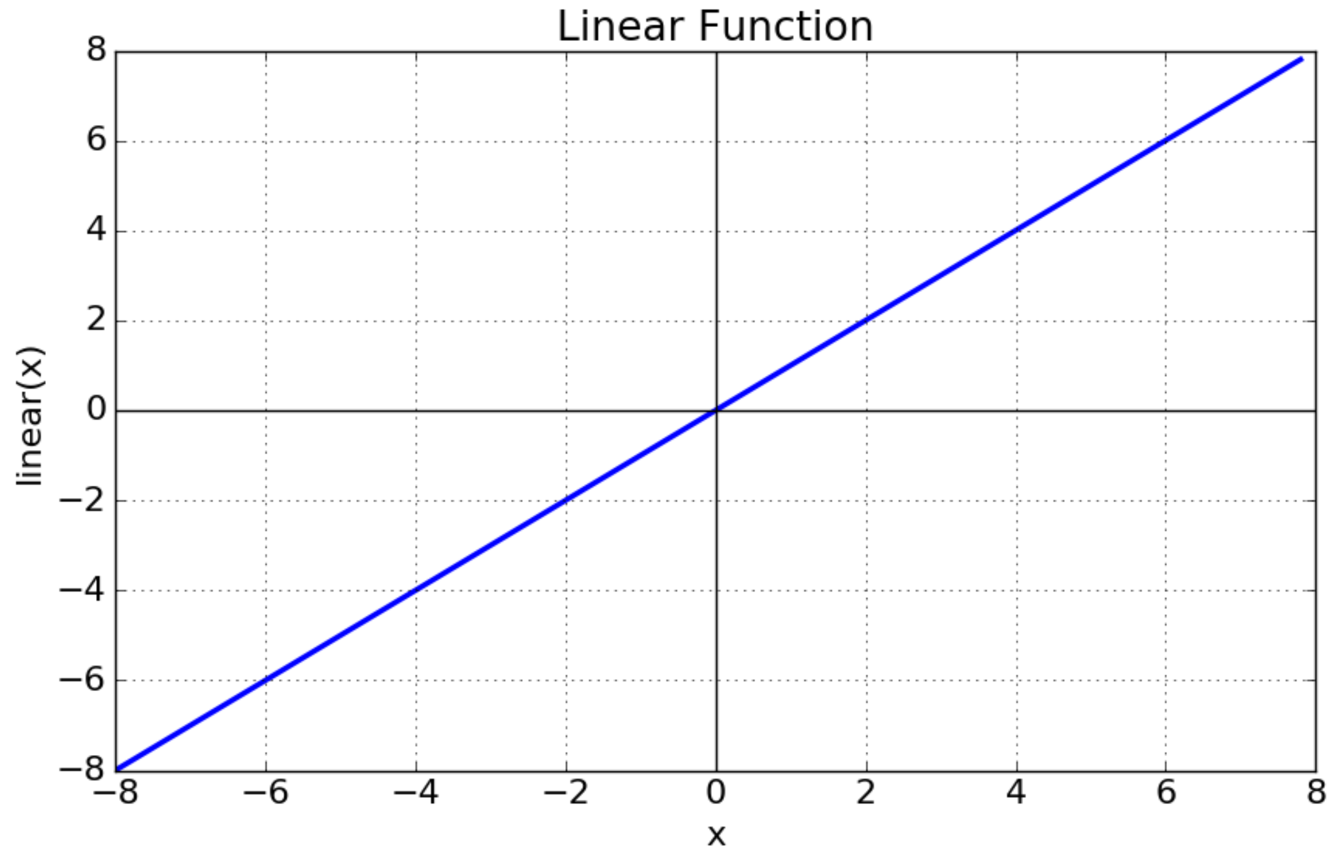
  Input Layer: 128 Nodes

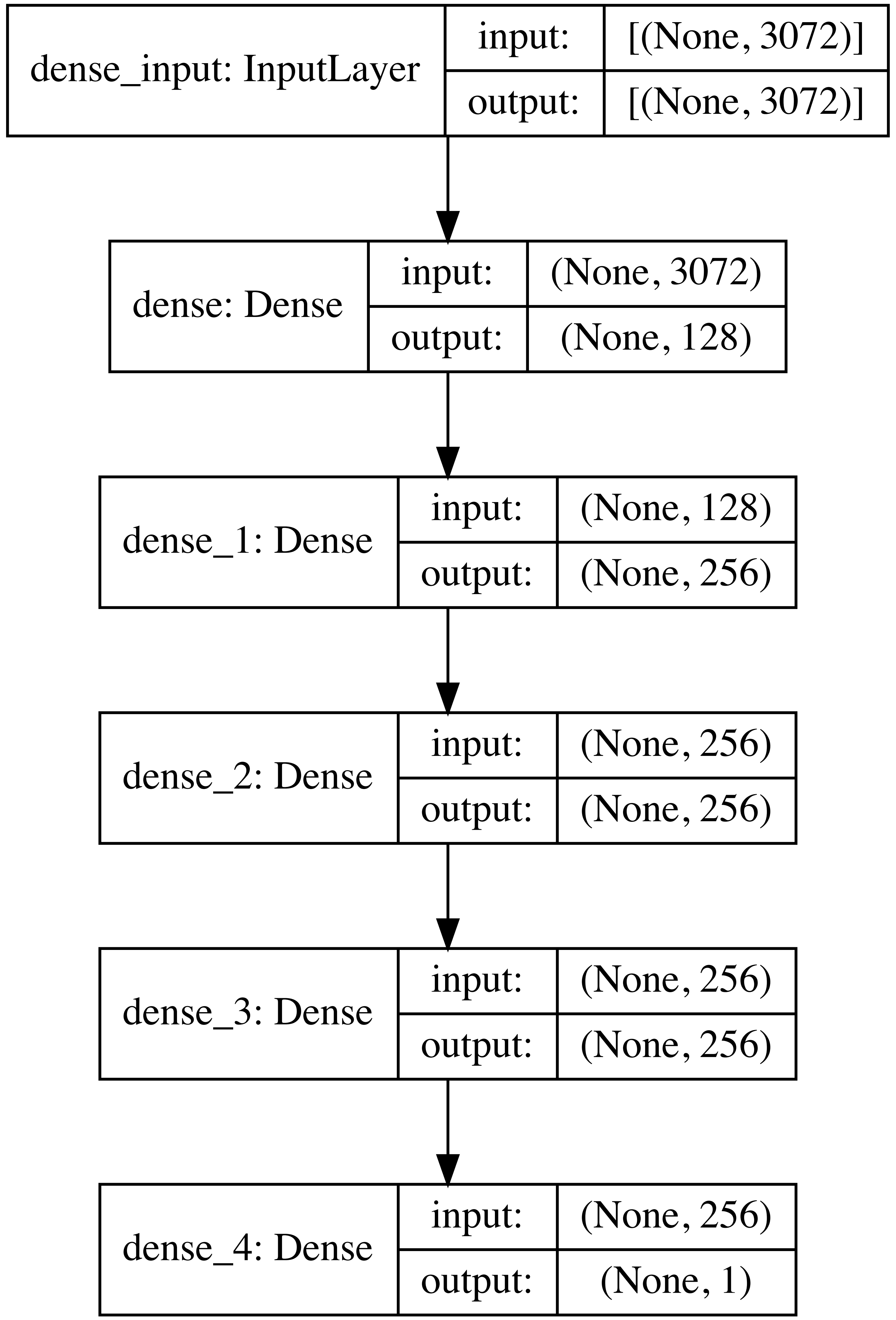
  Three hidden layers: 256 Nodes each

  Final layer : 1 node (v_2)

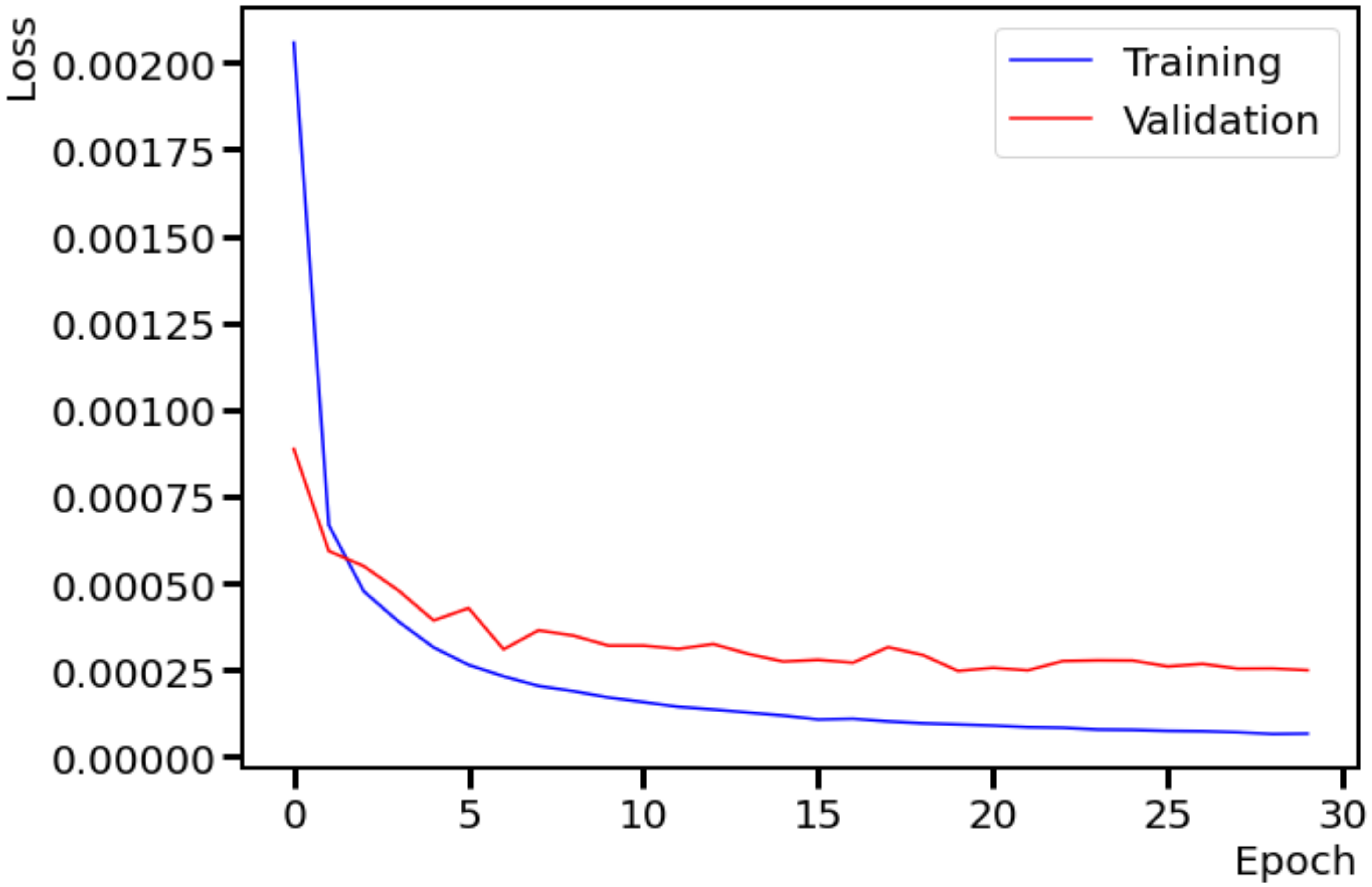
- Input and hidden layers have *ReLu* Activation
- Output layer has *Linear* activation
- Optimzer: $adam$, Loss function: mse
- Max epoch: 100, Batch Size: 32
- Training: 2×10^5 Events (~60 GB)
- Validation: $10\,\%$ Events

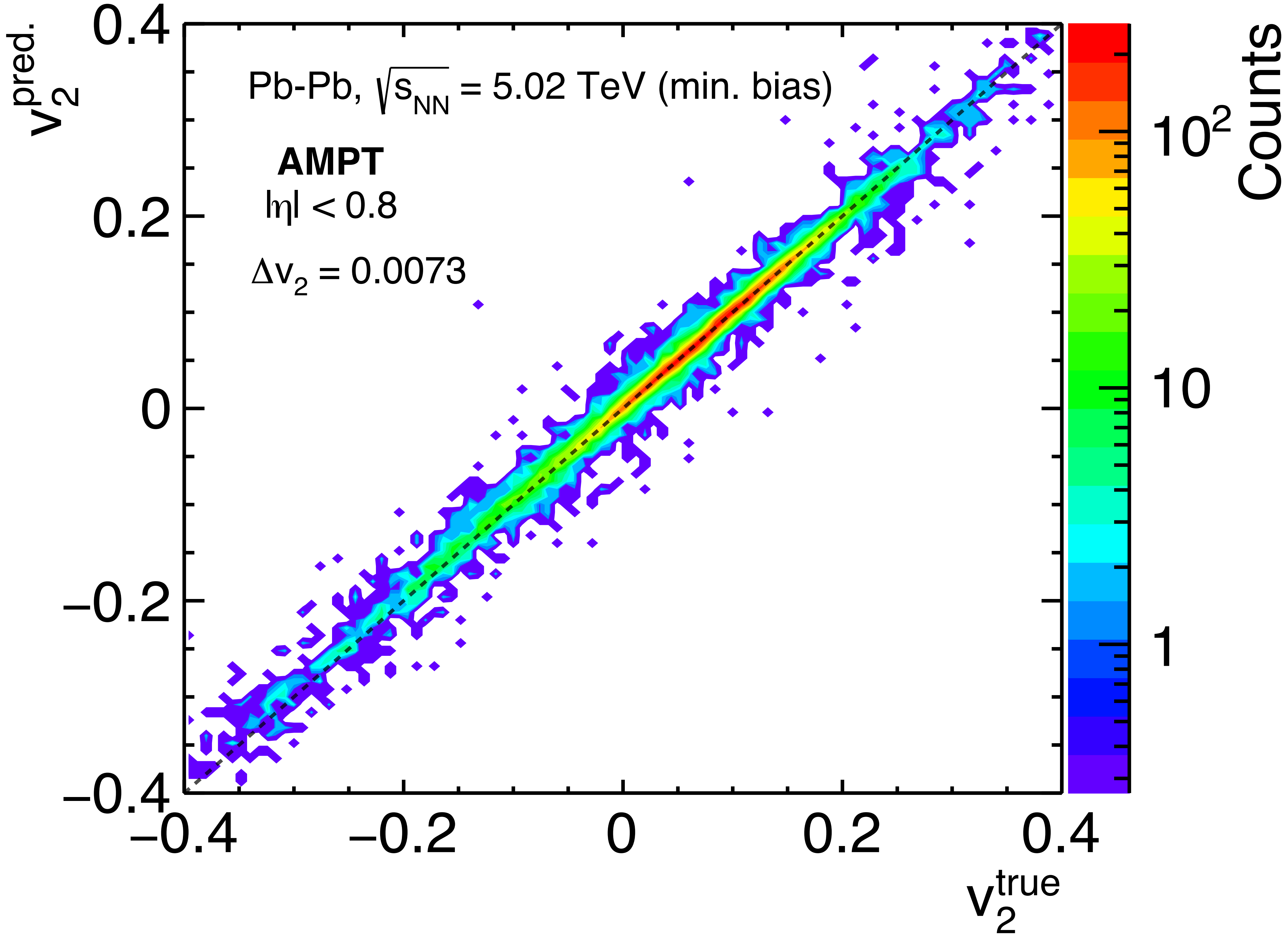






Results

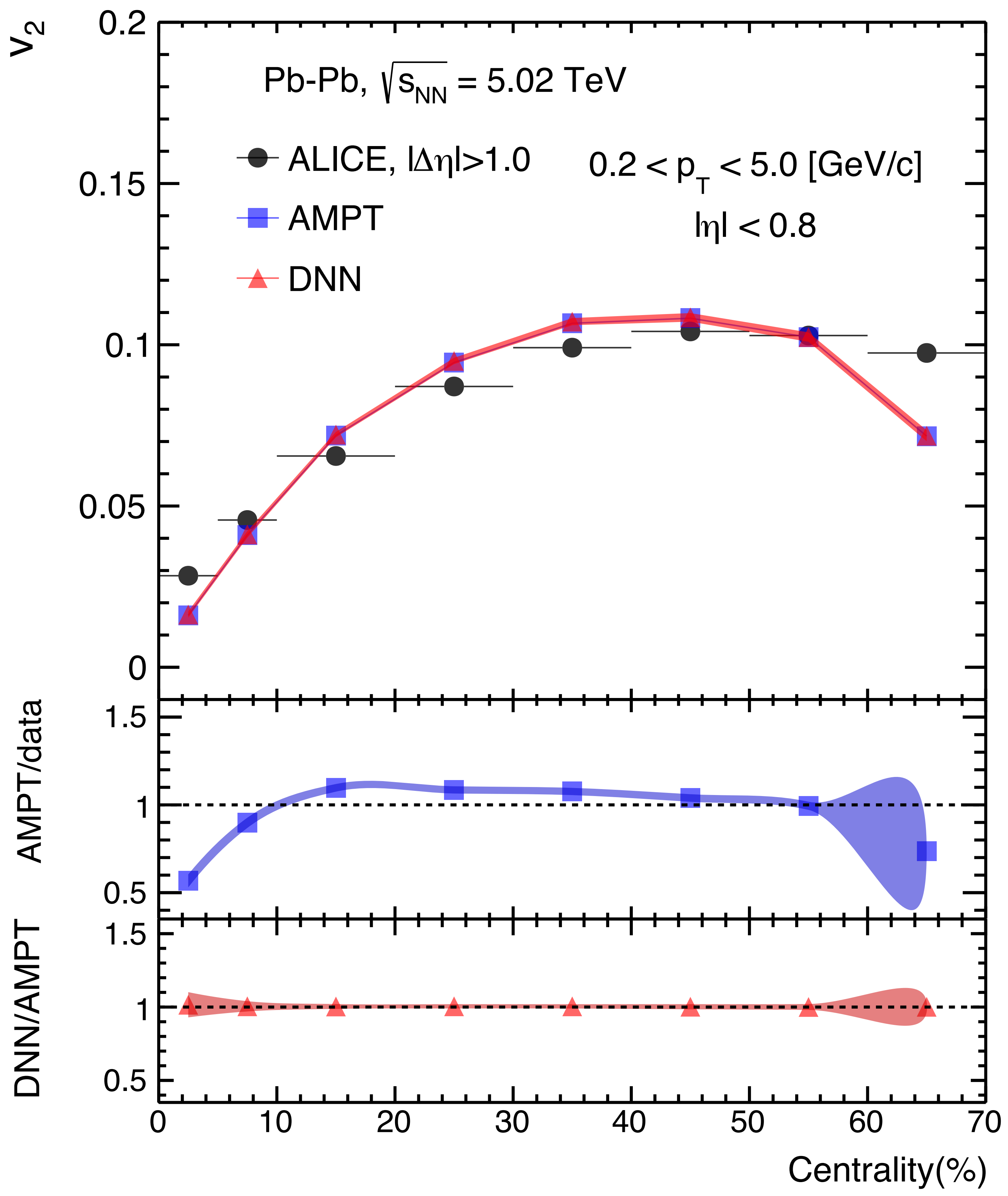


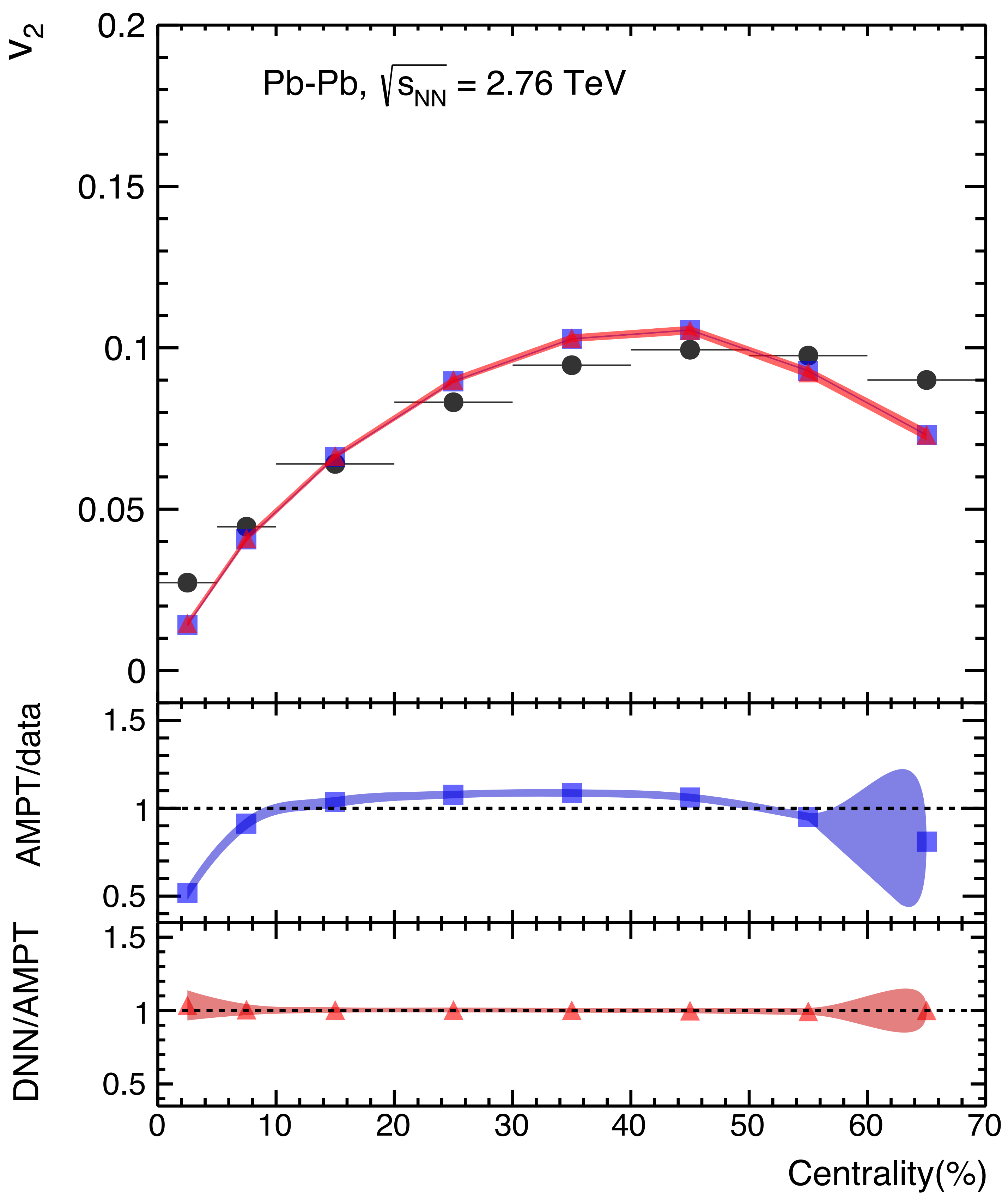


$$\Delta v_2 = \frac{1}{N_{\text{events}}} \sum_{n=1}^{N_{\text{events}}} | v_{2_n}^{true} - v_{2_n}^{pred.} |$$

N. Mallick, S. Prasad, A. N. Mishra, R. Sahoo, and G. G. Barnaföldi (In preparation)

Results

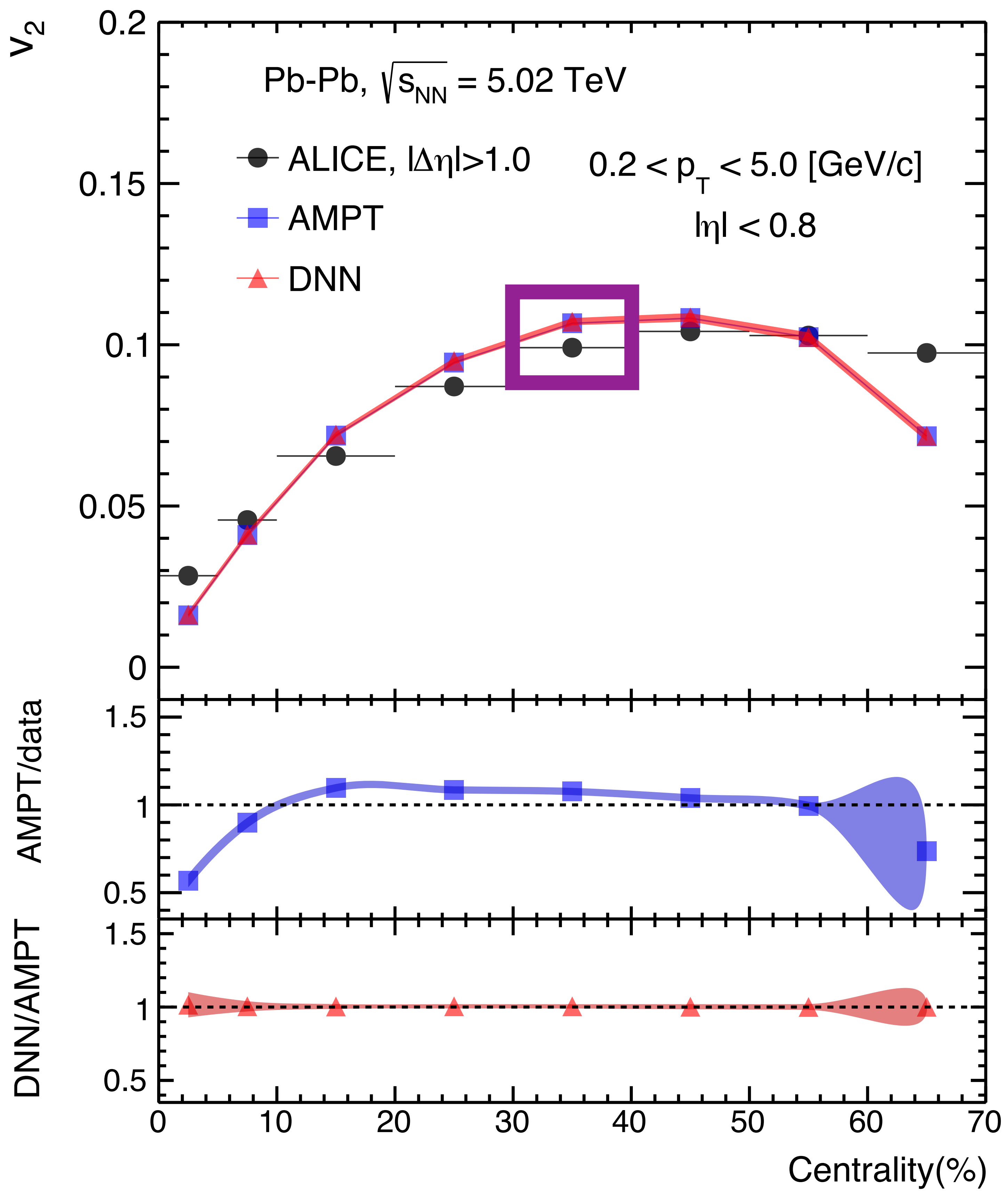


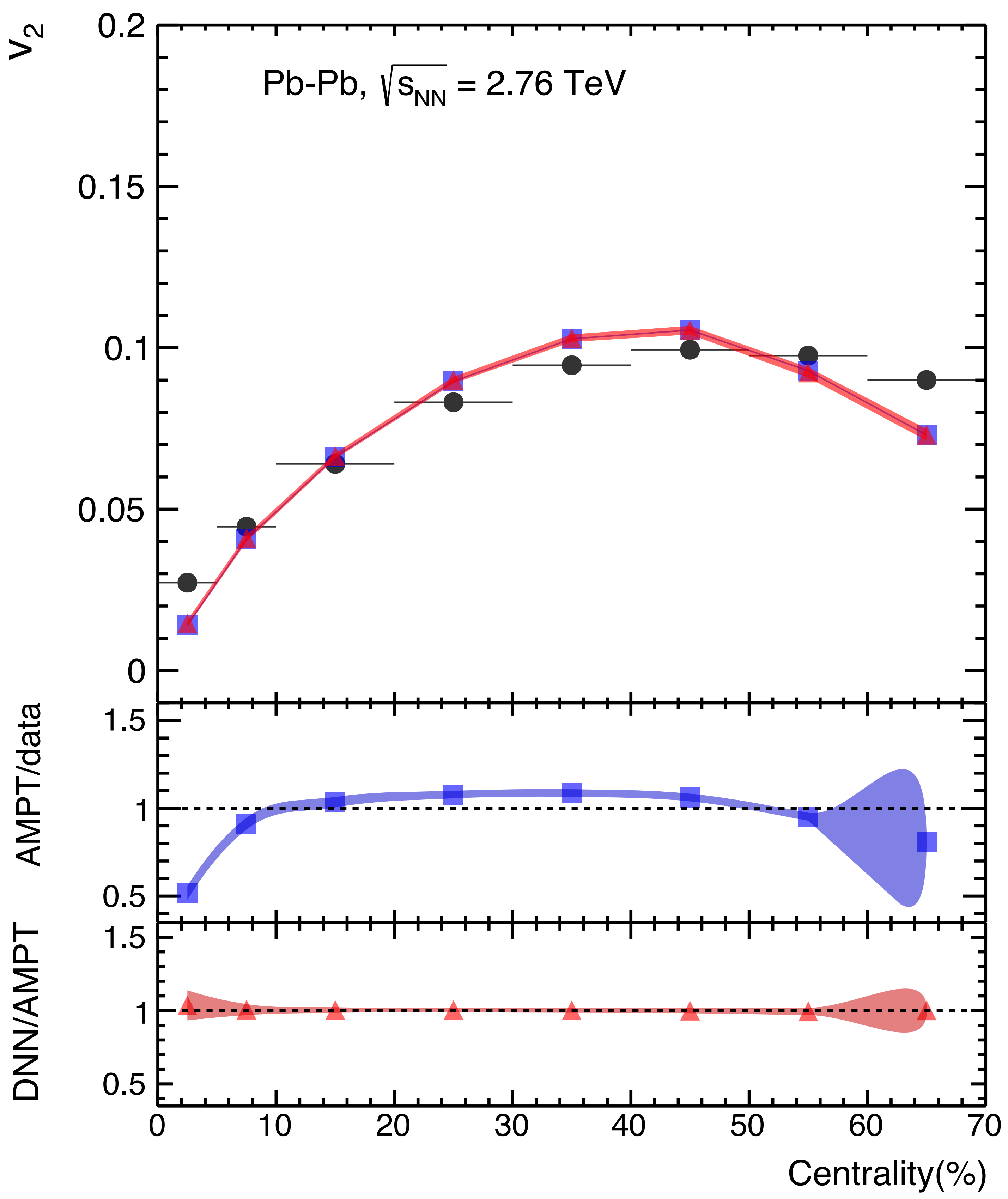


- DNN trained with 5.02 TeV minimum bias simulated data
- Good agreement between the simulated and predicted values of v_2
- ML model applied to lower energy
- **DNN preserves energy dependence of v_2**

N. Mallick, S. Prasad, A. N. Mishra, R. Sahoo, and G. G. Barnaföldi (In preparation)

Results

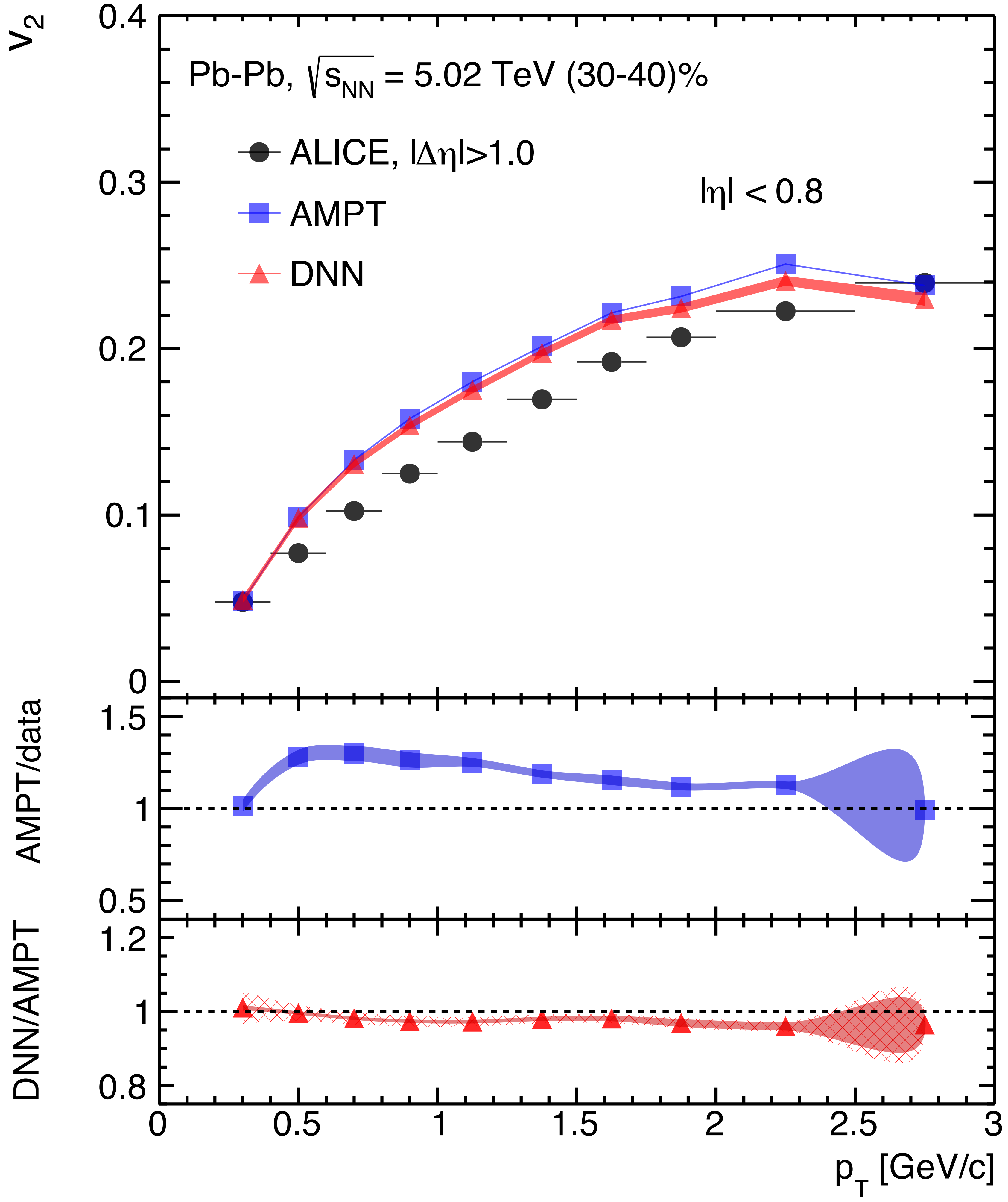




- DNN trained with 5.02 TeV minimum bias simulated data
- Good agreement between the simulated and predicted values of v_2
- ML model applied to lower energy
- **DNN preserves energy dependence of v_2**

N. Mallick, S. Prasad, A. N. Mishra, R. Sahoo, and G. G. Barnaföldi (In preparation)

Results



- Training done in the range: $0.2 < p_T < 5.0$ [GeV/c]
- Applied to different slices of p_T-bins:

  [0.2, 0.4, 0.6, 0.8, 1.0, 1.25, 1.5, 1.75, 2.0, 2.5, 3.0]
- Elliptic flow as a function of transverse momentum
- **DNN preserves the p_T dependence of v_2**

N. Mallick, S. Prasad, A. N. Mishra, R. Sahoo, and G. G. Barnaföldi (In preparation)

Summary

Problem?$\rightarrow$Learn$\rightarrow$Experience$\rightarrow$Decision!

Summary

Problem?→Learn→Experience→Decision!

- Report the first implementation of ML tools for the estimation of impact parameter, transverse spherocity, and v_2 in heavy-ion collisions at the LHC
- Minimum bias training predicts centrality wise distributions
- ML preserves the centrality and energy dependence of particle production
- Final state particle information is used
- Training is resource hungry → application is faster and economic
- A learning process → Scope for improvements

Summary

Problem?→Learn→Experience→Decision!

- Report the first implementation of ML tools for the estimation of impact parameter, transverse spherocity, and v_2 in heavy-ion collisions at the LHC
- Minimum bias training predicts centrality wise distributions
- ML preserves the centrality and energy dependence of particle production
- Final state particle information is used
- Training is resource hungry → application is faster and economic
- A learning process → Scope for improvements

Thank you!

Back up

Machine learning in HEP

https://iml-wg.github.io/HEPML-LivingReview/

Machine learning in HEP

- ML needs a big data to achieve good accuracy and perform predictions
- Big data experiments such as the LHC, CERN provides ample opportunity to apply ML techniques to High Energy Physics (HEP)
- The data flow from all experiments in RUN2 was about 25 GB/s
- A factor of 10x particle flow rate is expected in the high luminosity LHC era
- Requirements for faster simulation
- Faster computing, GPUs, FPGAs and ML are key to the future

https://iml-wg.github.io/HEPML-LivingReview/

Machine learning in HEP

- ML needs a big data to achieve good accuracy and perform predictions
- Big data experiments such as the LHC, CERN provides ample opportunity to apply ML techniques to High Energy Physics (HEP)
- The data flow from all experiments in RUN2 was about 25 GB/s
- A factor of 10x particle flow rate is expected in the high luminosity LHC era
- Requirements for faster simulation
- Faster computing, GPUs, FPGAs and ML are key to the future

- PID, reconstruction and triggering
- Simulation
- Data Quality Monitoring
- Unfolding Techniques
- Jet tagging
- Neutrino Detectors
- **Heavy-ion physics and QGP phenomenology**

https://iml-wg.github.io/HEPML-LivingReview/

Machine learning in HEP

- ML needs a big data to achieve good accuracy and perform predictions
- Big data experiments such as the LHC, CERN provides ample opportunity to apply ML techniques to High Energy Physics (HEP)
- The data flow from all experiments in RUN2 was about 25 GB/s
- A factor of 10x particle flow rate is expected in the high luminosity LHC era
- Requirements for faster simulation
- Faster computing, GPUs, FPGAs and ML are key to the future

https://iml-wg.github.io/HEPML-LivingReview/

- PID, reconstruction and triggering
- Simulation
- Data Quality Monitoring
- Unfolding Techniques
- Jet tagging
- Neutrino Detectors
- **Heavy-ion physics and QGP phenomenology**

ROOT Data Analysis Framework

scikits learn machine learning in Python

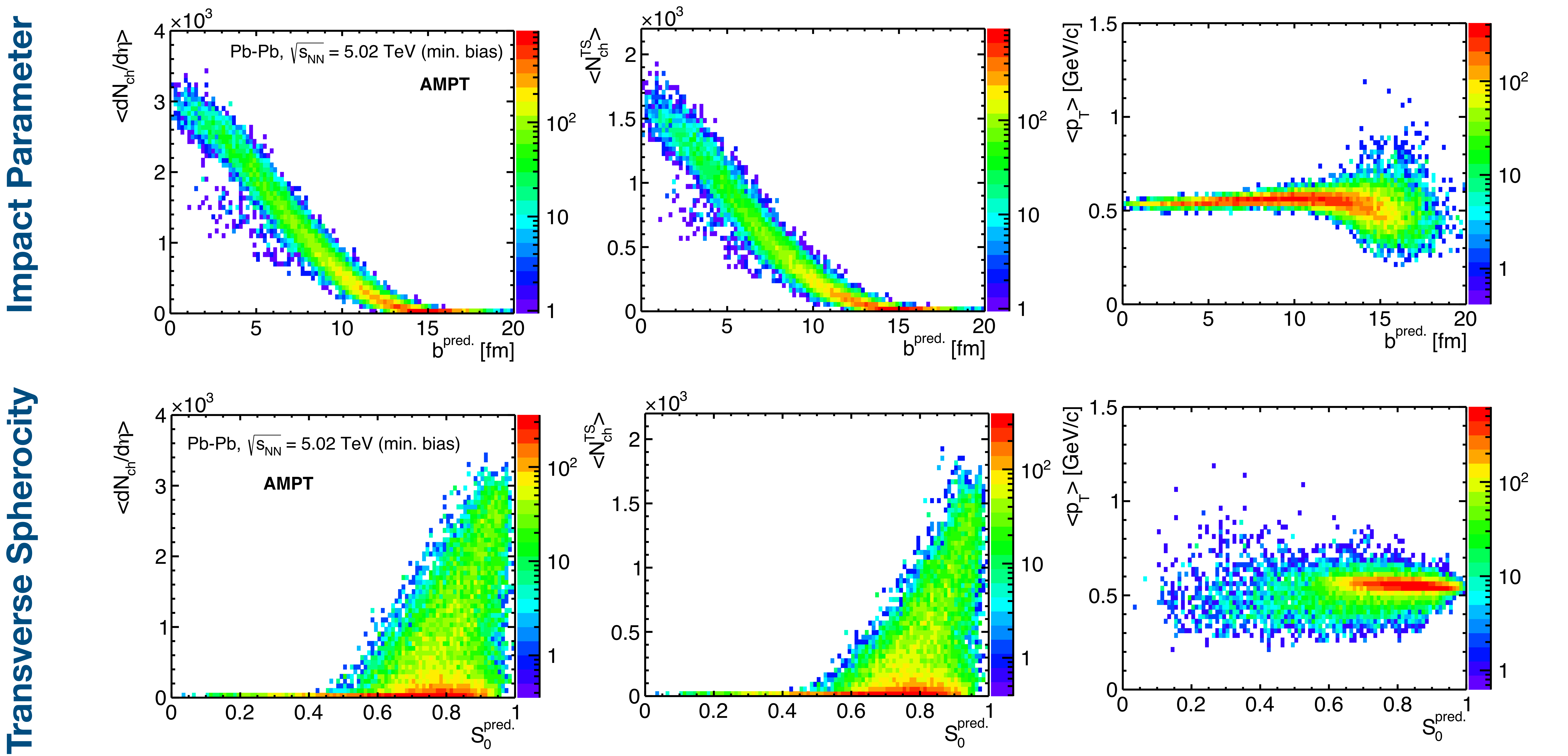
TMVA

TensorFlow

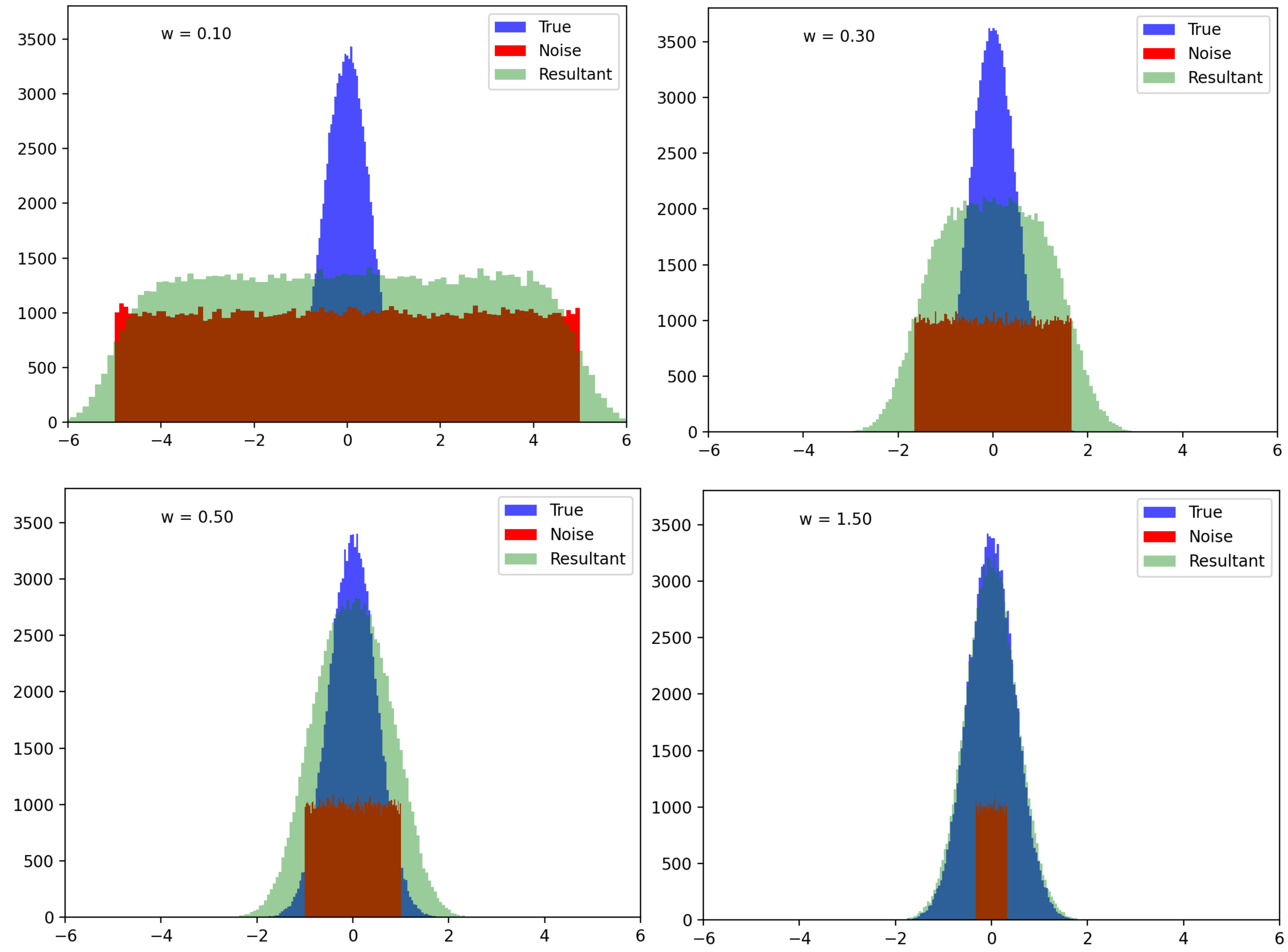
Keras

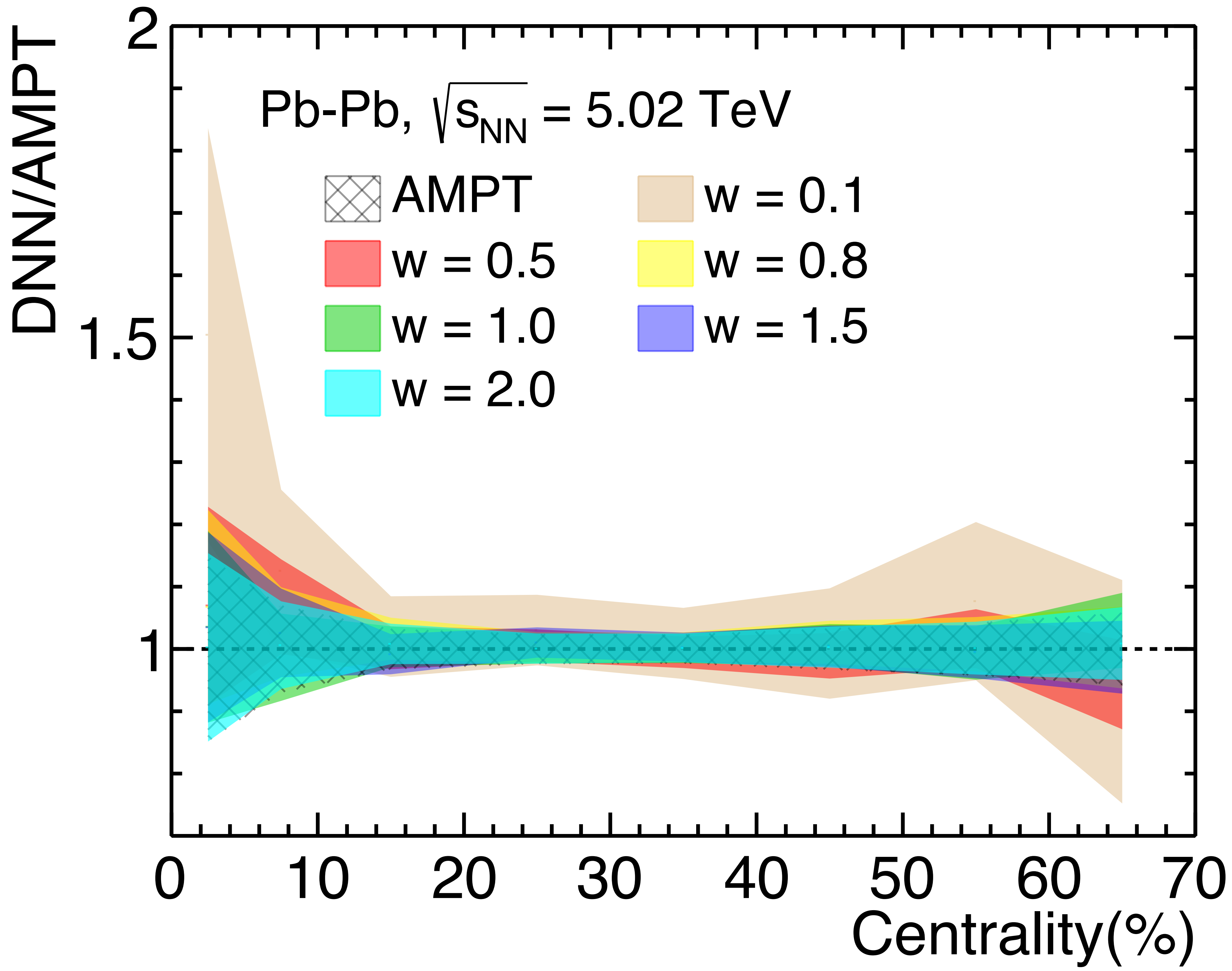
https://root.cern/
https://root.cern/manual/tmva/
Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011
https://keras.io/
https://www.tensorflow.org/

Average Multiplicity | Transverse Multiplicity | Average Transverse Momentum



N. Mallick, S. Tripathy, A. N. Mishra, S. Deb, and R. Sahoo, Phys. Rev. D103, 094031 (2021)


w = 0.10
w = 0.30
w = 0.50
w = 1.50
True
Noise
Resultant


Pb-Pb, $\sqrt{s_{NN}}$ = 5.02 TeV
AMPT
w = 0.1
w = 0.5
w = 0.8
w = 1.0
w = 1.5
w = 2.0
DNN/AMPT
Centrality(%)