Estimating Elliptic Flow Coefficient in Heavy-ion Collisions using Deep Learning

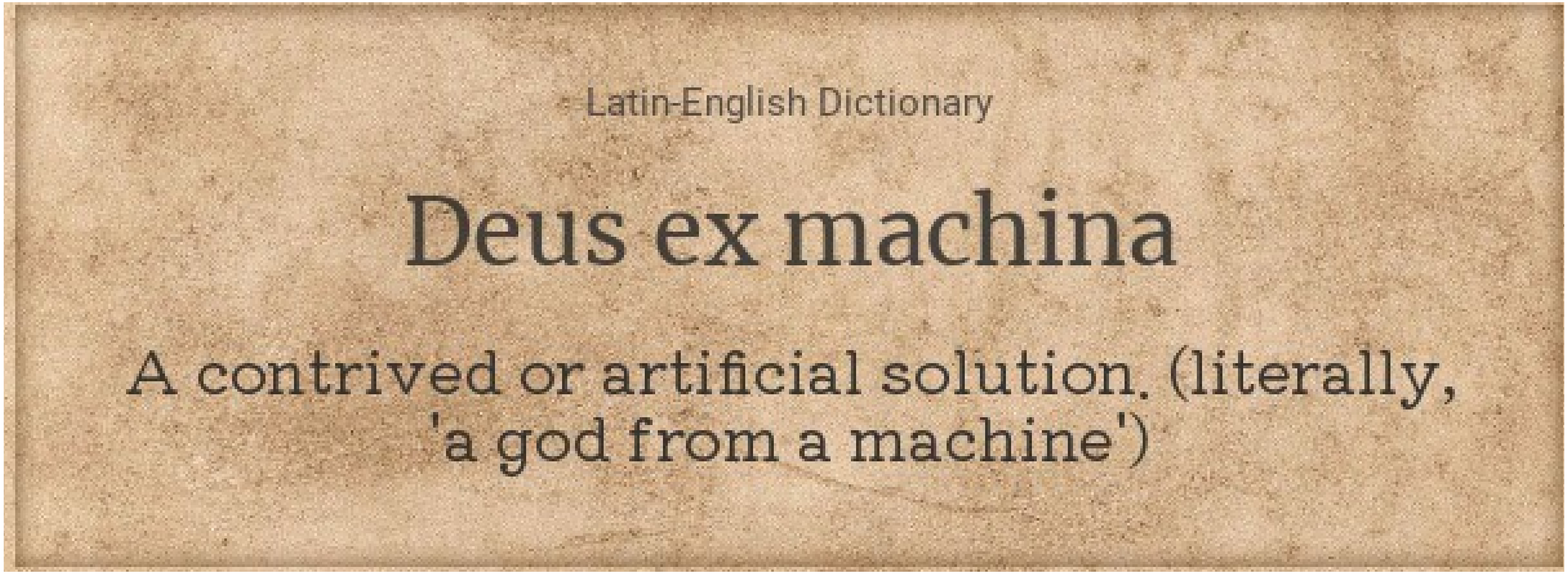
N. Mallick, A.N. Mishra, S. Pasad, R. Sahoo and **G.G. Barnaföldi**

Support: *Hungarian OTKA grants, NK123815, K135515 Wigner GPU Laboratory*
Ref: arXiv:2108.13938 (submitted to PRD)

PP2022, Budapest, 16th May 2022

Eötvös Loránd Research Network

Motivation


Latin-English Dictionary
Deus ex machina
A contrived or artificial solution. (literally,
'a god from a machine')

Motivation

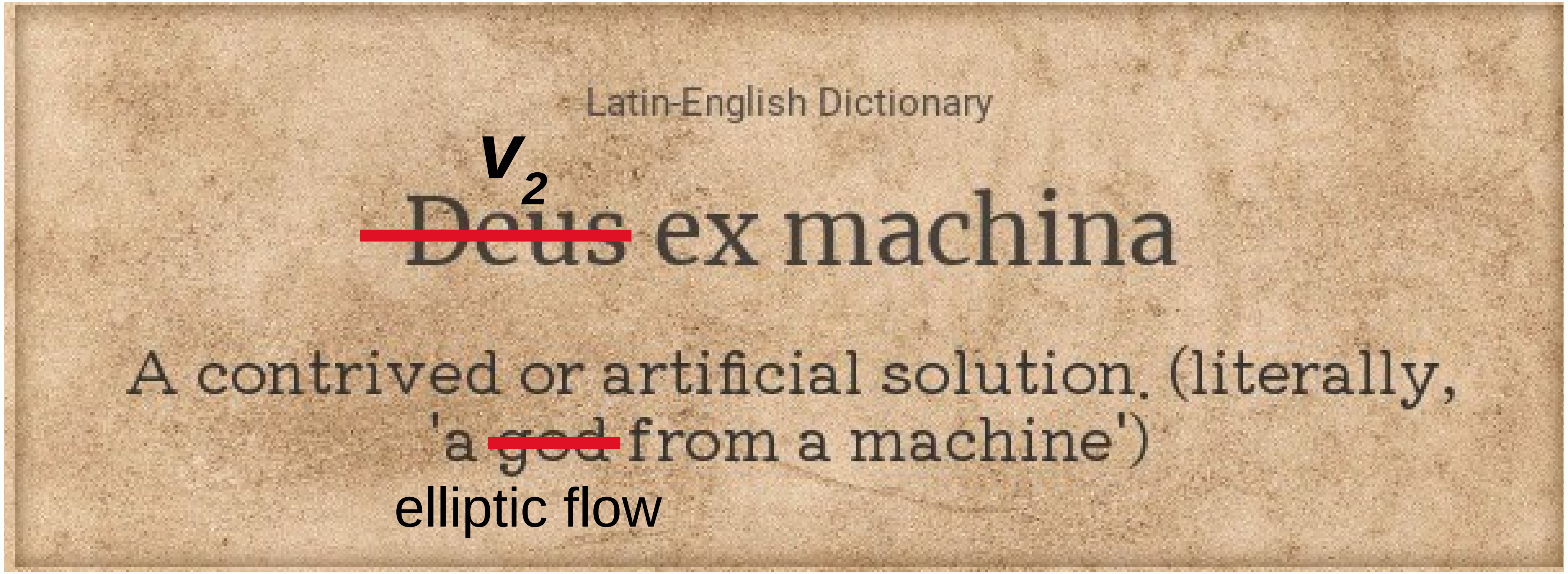
Latin-English Dictionary

v_2

~~Deus~~ ex machina

A contrived or artificial solution. (literally, 'a god from a machine')

Motivation


Latin-English Dictionary

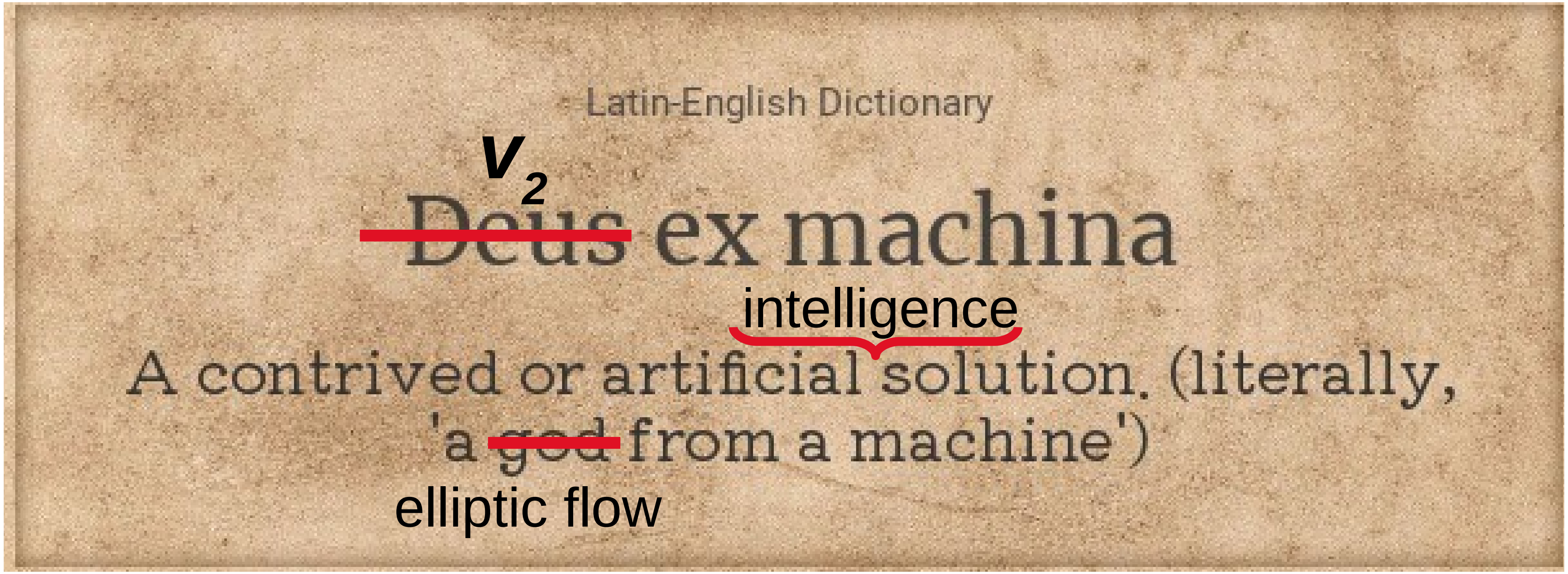
v_2

~~Deus~~ ex machina

A contrived or artificial solution. (literally, 'a ~~god~~ from a machine')

elliptic flow

Motivation


Latin-English Dictionary

v_2

~~Deus~~ ex machina

intelligence

A contrived or artificial solution. (literally, 'a ~~god~~ from a machine')

elliptic flow

Outline

1) Elliptic flow & motivation

- Motivation and definition

2) Input, test & model validation

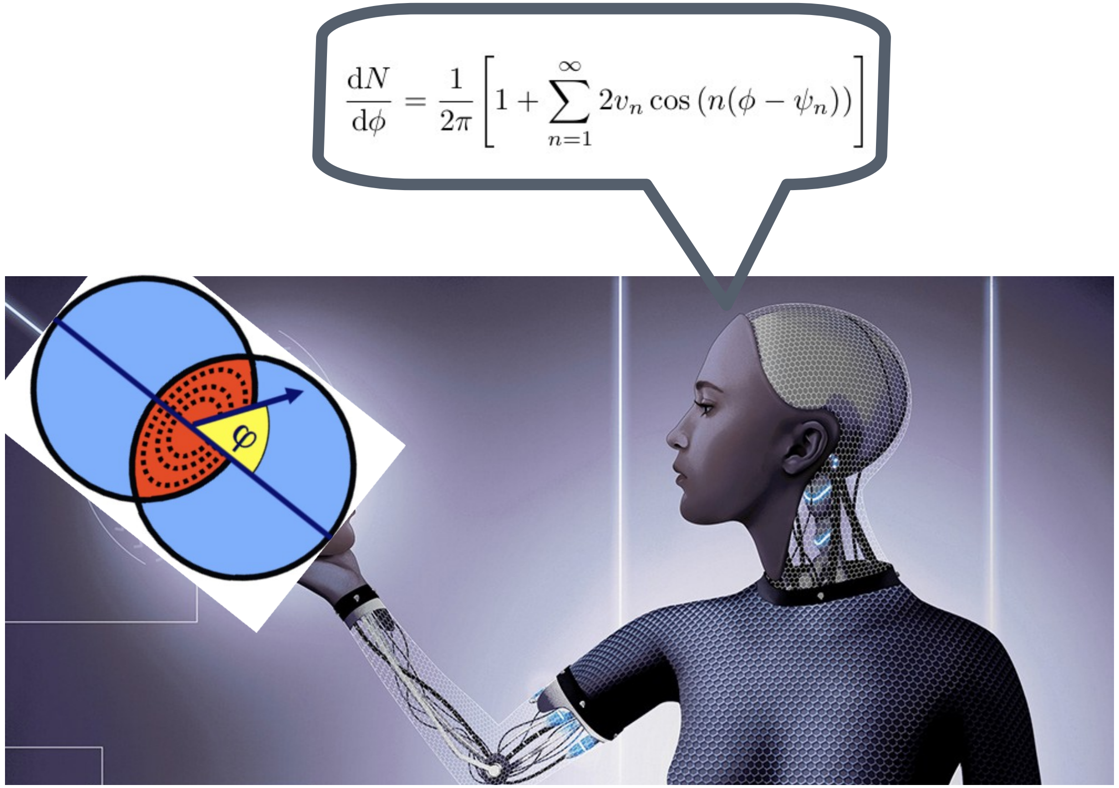
- Input data (min. bias AMPT)
- Optimalization the NN
- Test with noise, epoch

3) Results on v_2 by DNN

- Dependence on centrality, c.m. energy and p_T

Conclusions:

→ Can we estimate v_2 ex machina?

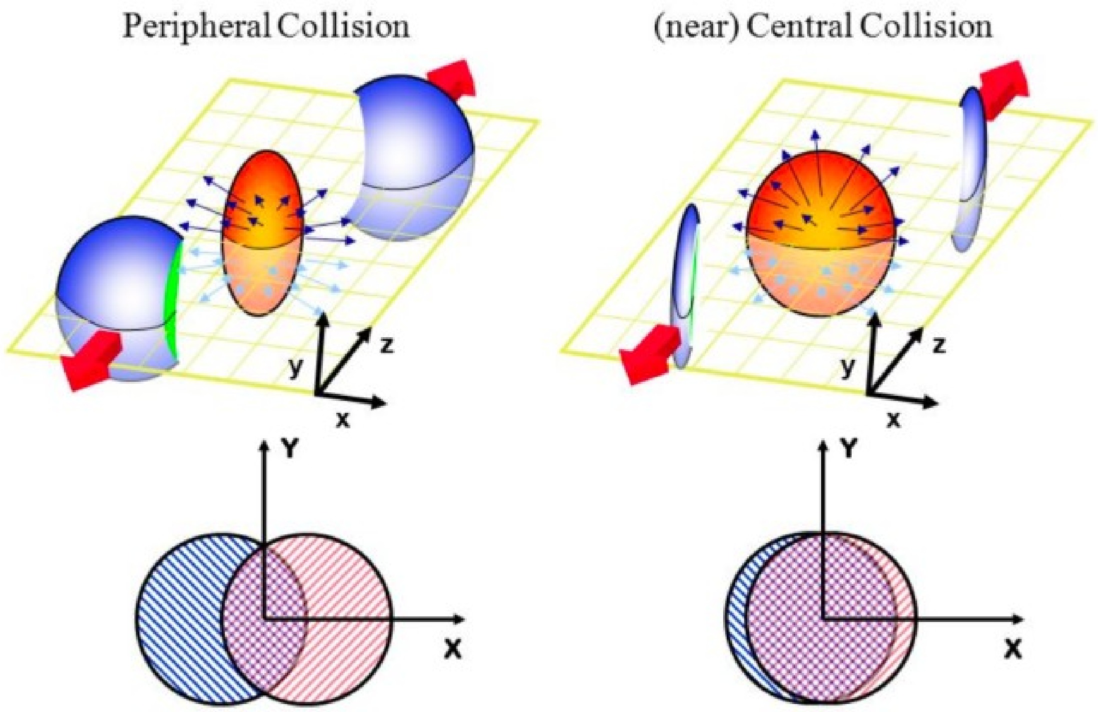


$$\frac{\mathrm{d}N}{\mathrm{d}\phi} = \frac{1}{2\pi}\left[1 + \sum_{n=1}^{\infty} 2v_n \cos\left(n(\phi - \psi_n)\right)\right]$$

Motivation & definitions

Elliptic flow (v_2) in heavy-ion collisons

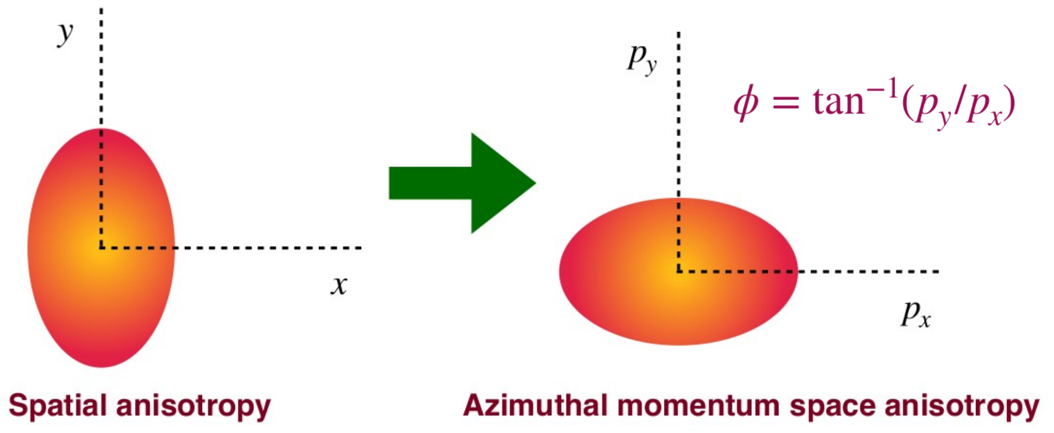
- **Experimental point:**
  - Elliptic flow describes the azimuthal momentum space anisotropy of particle emission for a non-central heavy-ion collision.



Elliptic flow (v_2) in heavy-ion collisons

- **Experimental point:**
  - 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:

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

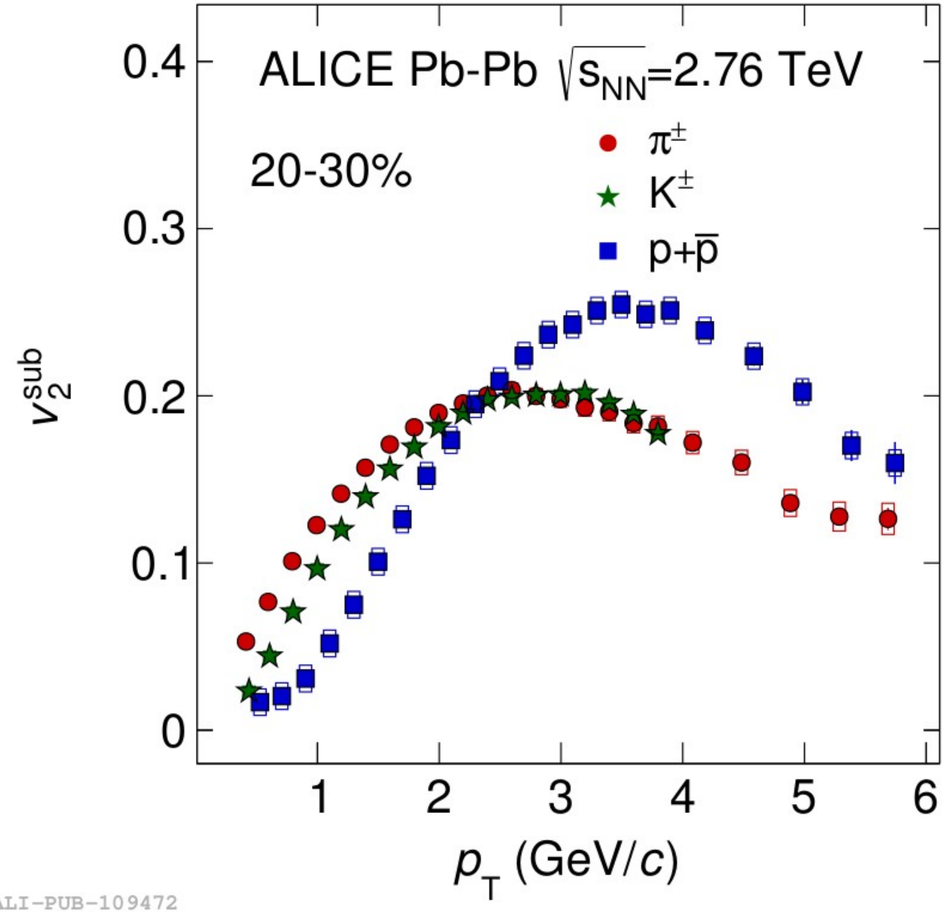


Elliptic flow (v_2) in heavy-ion collisons

- **Experimental point:**
  - 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:

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

  - The $v_2(p_T, y) = \langle \cos(2(\phi - \psi_2)) \rangle$ directly reflects the initial spatial anisotropy of the nuclear overlap region in the transverse plane.



Input, test, and model validation

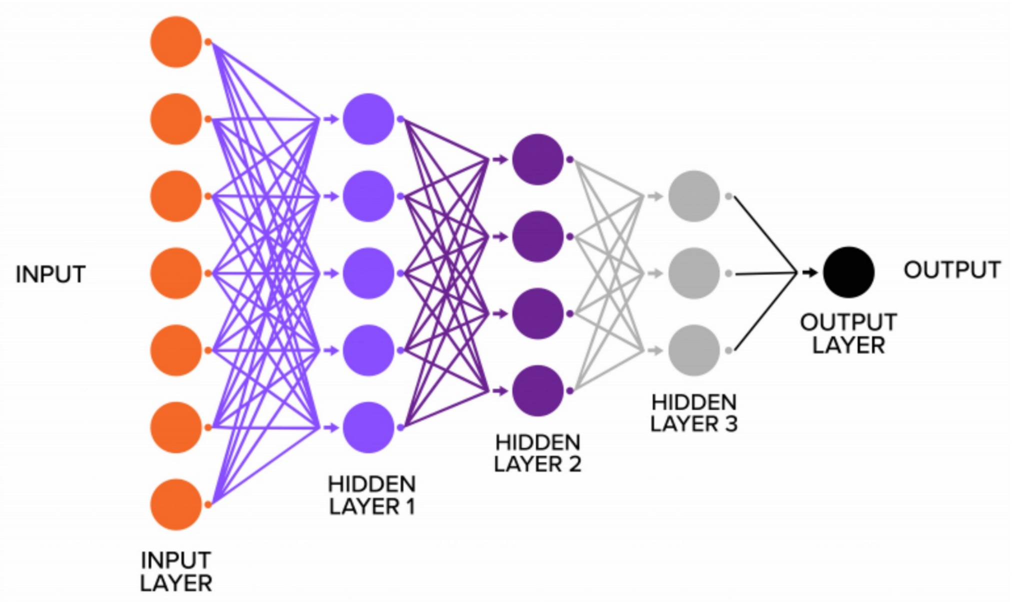
The AMPT model for Pb-Pb collisions

- **A Multi-phase transport model (AMPT):** MC event generator for simulating p-A and A-A collisions from RHIC to LHC energies.
  - **Fluctuating initial conditions:** Initialization of collision is done by obtaining the spatial and momentum distributions of the hard minijet partons and soft string excitations from the HIJING model. The inbuilt Glauber model is used to calculate and convert the cross-section of the produced mini-jets from pp to AA.
  - **Zhang’s parton cascade (ZPC)** model is used to perform the partonic interactions and parton cascade which currently includes the two-body scatterings with cross-sections obtained from the pQCD with screening masses.
  - **Hadronization mechanism:** Lund string fragmentation model is used to recombine the partons with their parent strings and then the strings are converted to hadrons, whereas, in the string melting mode the transported partons are hadronized using a quark coalescence mechanism.
  - **Hadron cascade**: scattering among the produced hadrons are performed using a relativistic transport model (ART) by meson-meson, meson-baryon and baryon-baryon interactions.

Building up the ML structure

Three key layers

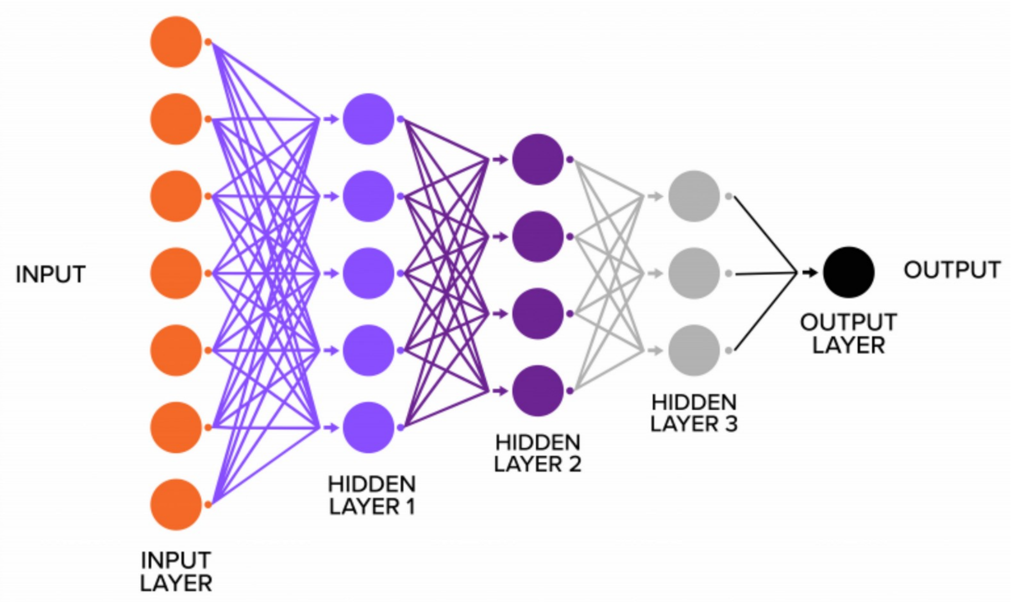
- **Input**: Takes the features as input
- **Hidden layers**: Connects to each neuron through different weights
- **Output**: Gives the result as a number or class



Building up the ML structure

Math behind

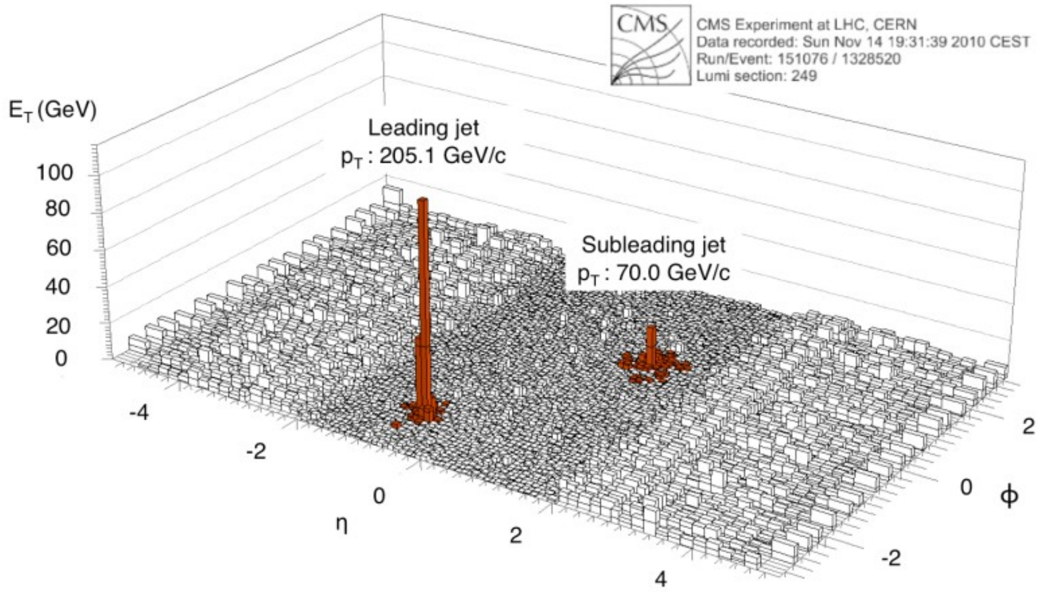
- **Weights** dictate the importance of an input
  → more important features get more weights
- **Activation function**: mathematical function that guides the outcome at each node
  → Standardize the values
- **Cost function**: Evaluates the accuracy between machine prediction and true value
- **Optimizer**: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



Building up the ML structure

Estimation of elliptic flow using DNN

- Elliptic flow → Event property
- Inputs → Track property
- (η-ϕ) space is the primary input space

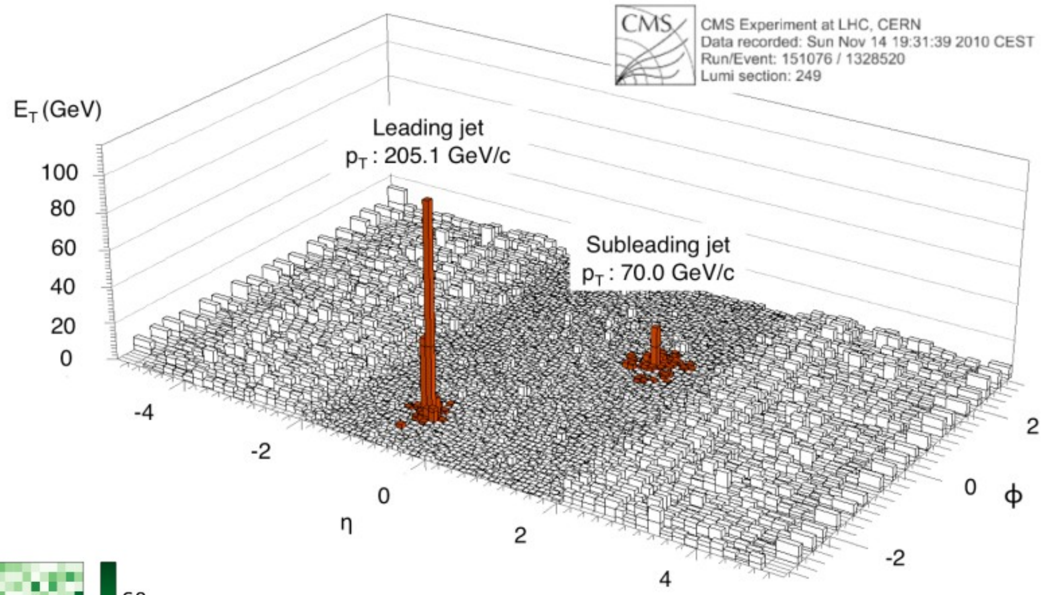


Serguei Chatrchyan et al., Phys.Rev.C 84 (2011), 024906

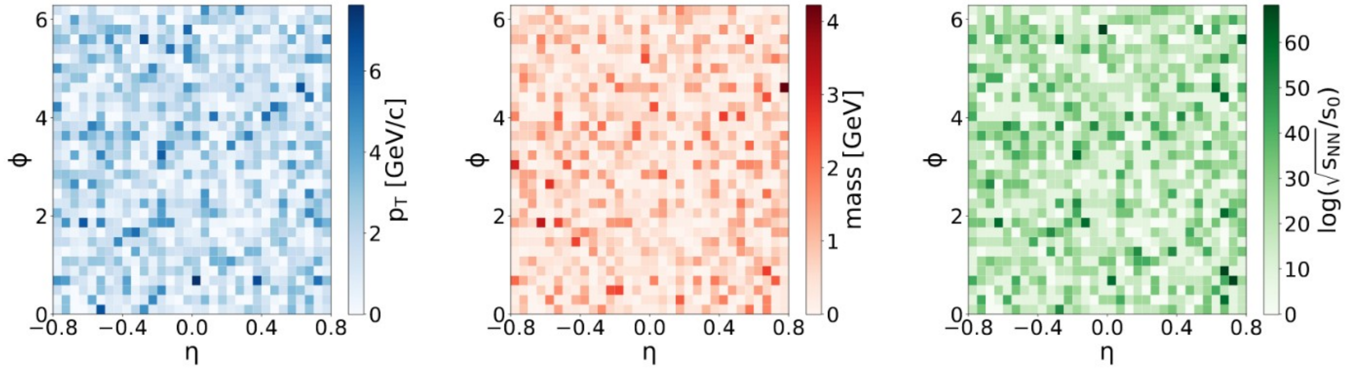
Building up the ML structure

Estimation of elliptic flow using DNN

- Elliptic flow → Event property
- Inputs → Track property
- (η-ϕ) space is the primary input space
- Three layers having different weights: p_T, mass and $log(s_{NN}/s_0)$ weighted layers serve as the secondary input space



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



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

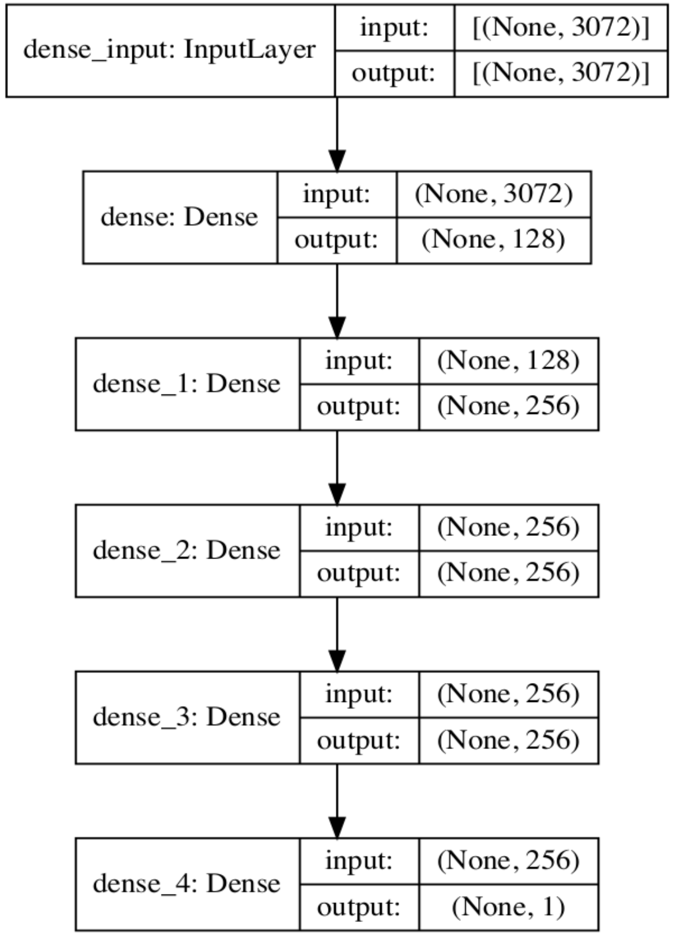
Building up the ML structure

Input "pictures" for DNN

- Each space has 32 × 32 pixels (grids)
- Total number of pixel points = 32 × 32 × 3 = 3072 for each event

DNN with the following architecture

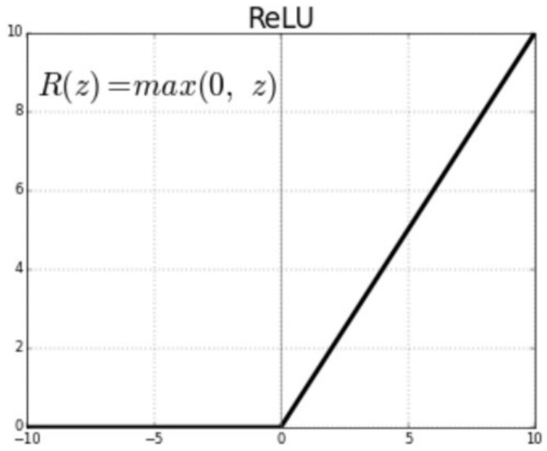
- Input Layer: 128 Nodes
- Three hidden layers: 256 Nodes each
- Final layer : 1 node (v_2)

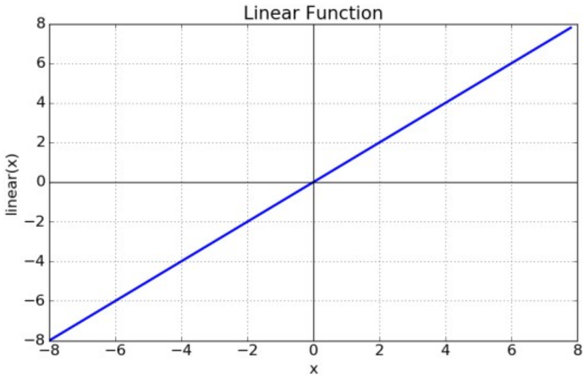


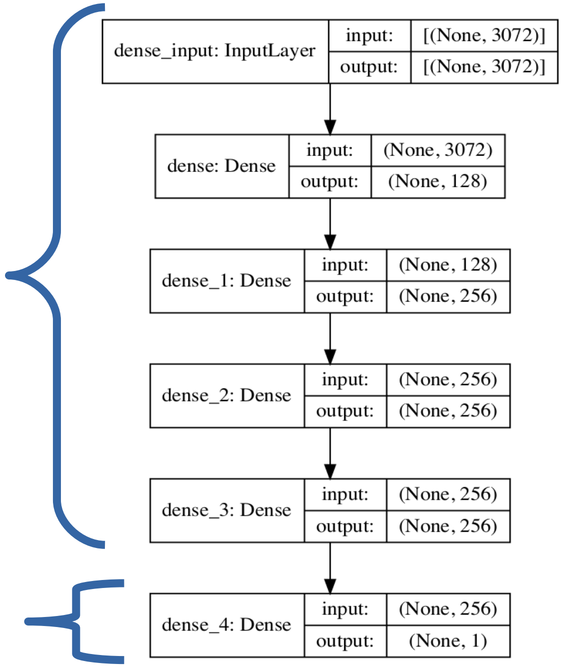
Building up the ML structure

Activation, optimalization, validation

- Input and hidden layers have ReLu Activation
- Output layer has Linear activation
- Optimizer: adam , Loss function: mse





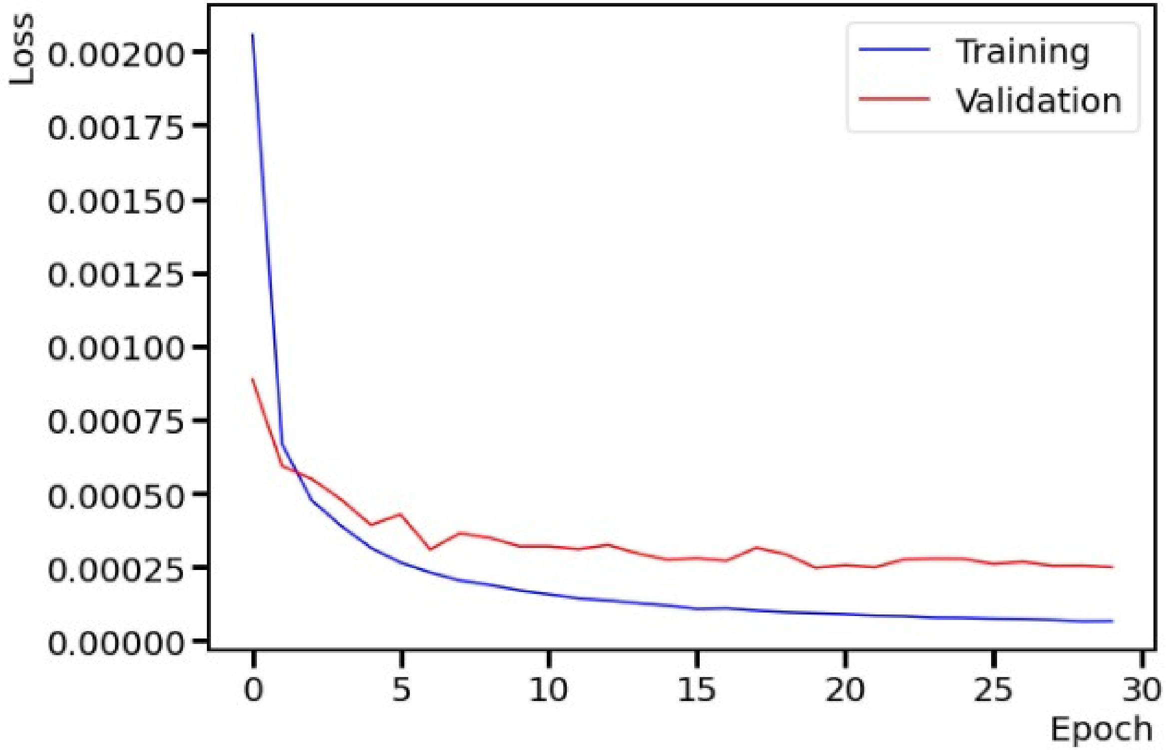


Testing the ML structure

Activation, optimalization, validation

- Input and hidden layers have ReLu Activation
- Output layer has Linear activation
- Optimizer: adam , Loss function: mse
- Epoch: 30, Batch Size: 32x32
- Training: 10^8 Events (~25 GB)

Bin size	Input neurons	MAE	Epoch	$\frac{\text{Time (sec)}}{\text{Epoch}}$	Trainable parameters
8×8	192	0.0292	18	1.679	189,569
16×16	768	0.0171	28	1.909	263,297
32×32	3072	0.0102	30	2.684	558,209
64×64	12288	0.0113	60	6.001	1,737,857

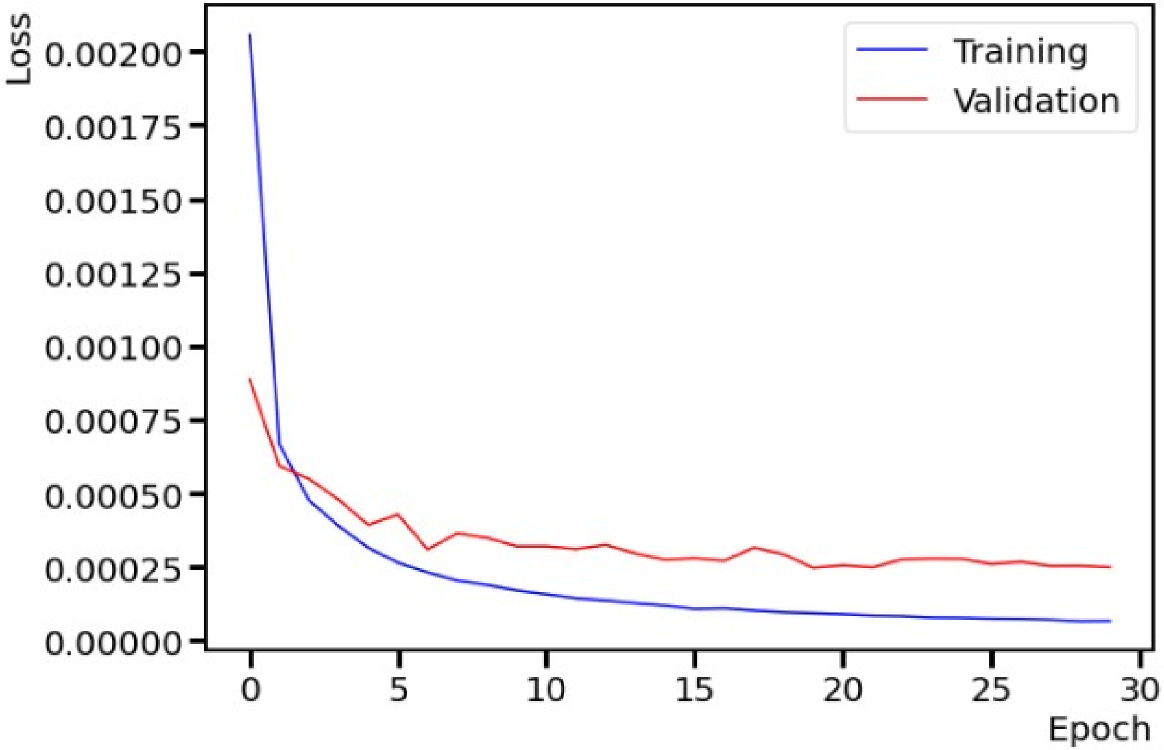


Testing the ML structure

Activation, optimalization, validation

- Input and hidden layers have ReLu Activation
- Output layer has Linear activation
- Optimizer: adam , Loss function: mse
- Epoch: 30, Batch Size: 32x32
- Training: 10^8 Events (~25 GB)

Bin size	Input neurons	MAE	Epoch	$\frac{\text{Time (sec)}}{\text{Epoch}}$	Trainable parameters
8×8	192	0.0292	18	1.679	189,569
16×16	768	0.0171	28	1.909	263,297
32×32	3072	0.0102	30	2.684	558,209
64×64	12288	0.0113	60	6.001	1,737,857

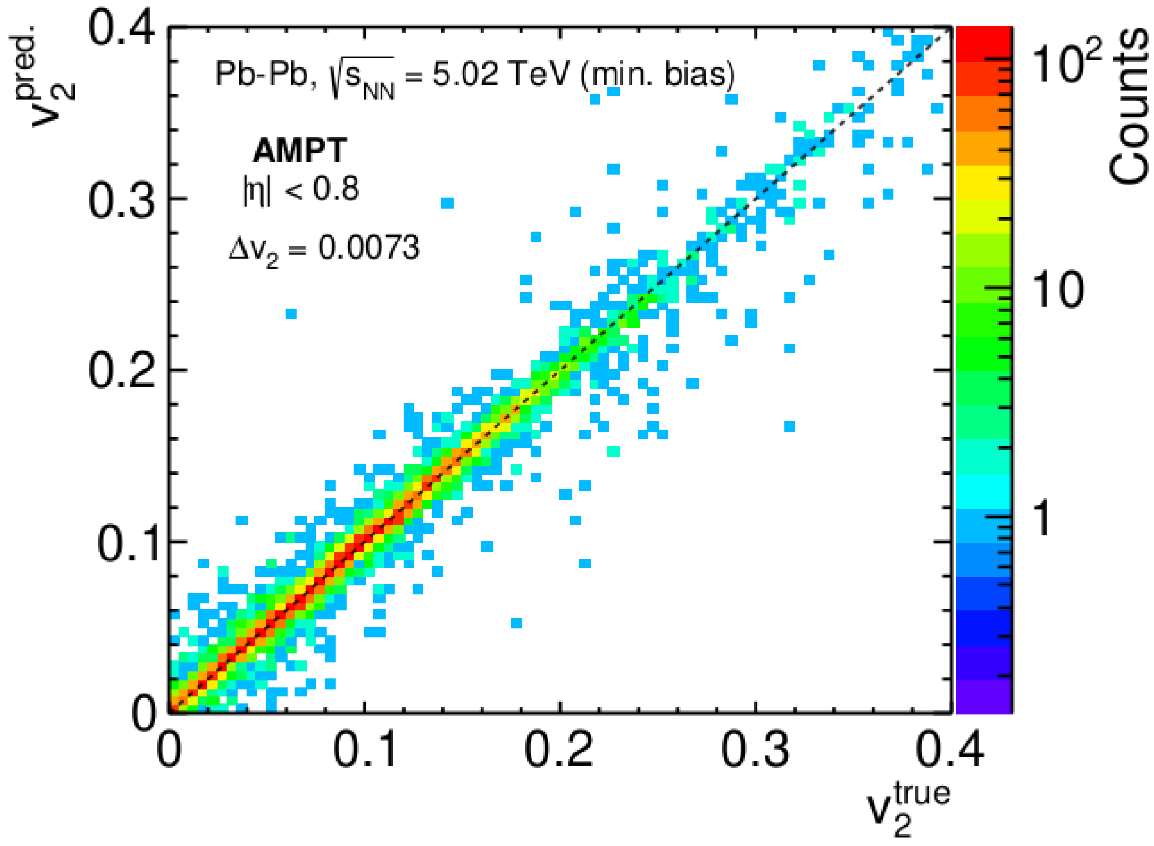


Testing the ML structure

Activation, optimalization, validation

- Input and hidden layers have ReLu Activation
- Output layer has Linear activation
- Optimizer: adam , Loss function: mse
- Epoch: 30, Batch Size: 32x32
- Training: 10^8 Events (~25 GB)
- Validation: 10^4 Events

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

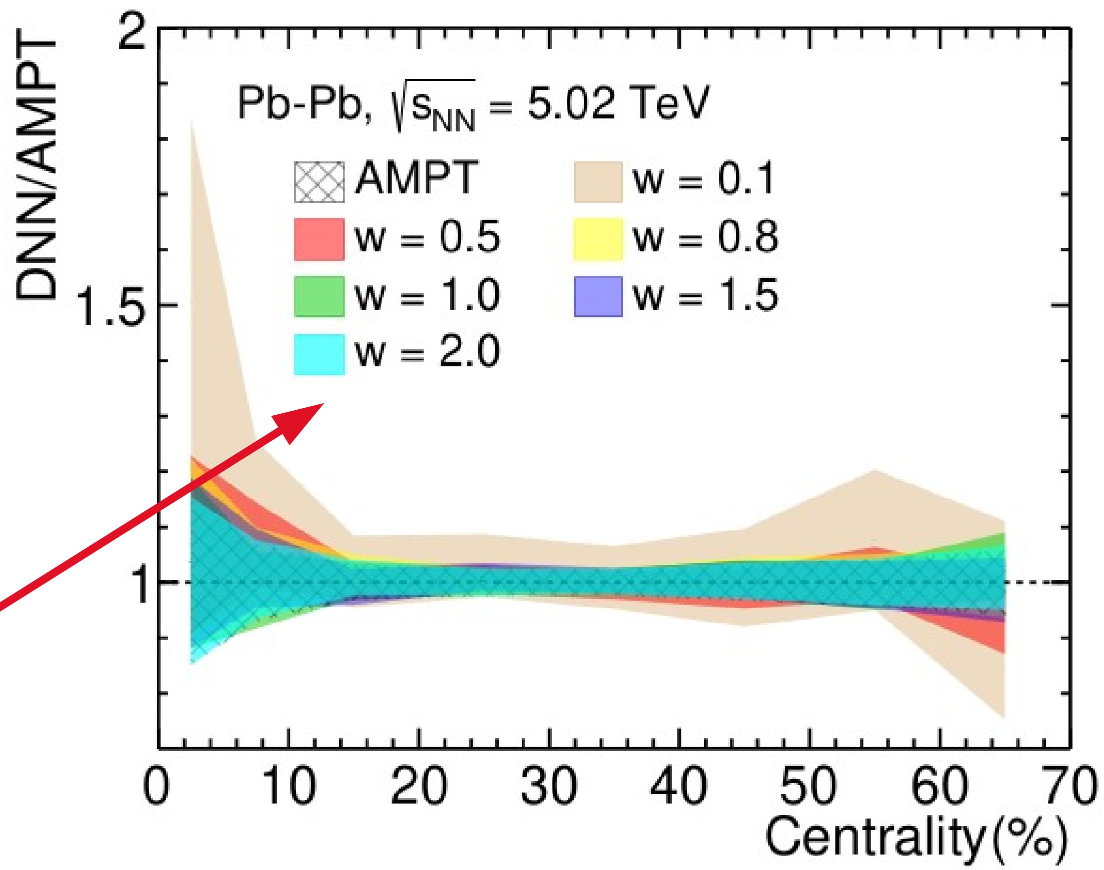


Testing the ML structure

Activation, optimalization, validation

- Input and hidden layers have ReLu Activation
- Output layer has Linear activation
- Optimizer: adam , Loss function: mse
- Epoch: 30, Batch Size: 32x32
- Training: 10^8 Events (~25 GB)
- Validation: 10^4 Events
- Error: effect of uncorrelated noise

$$F_{i,j} = F_{i,j} + X_{i,j}/\boxed{w}$$



v_2 ex machina

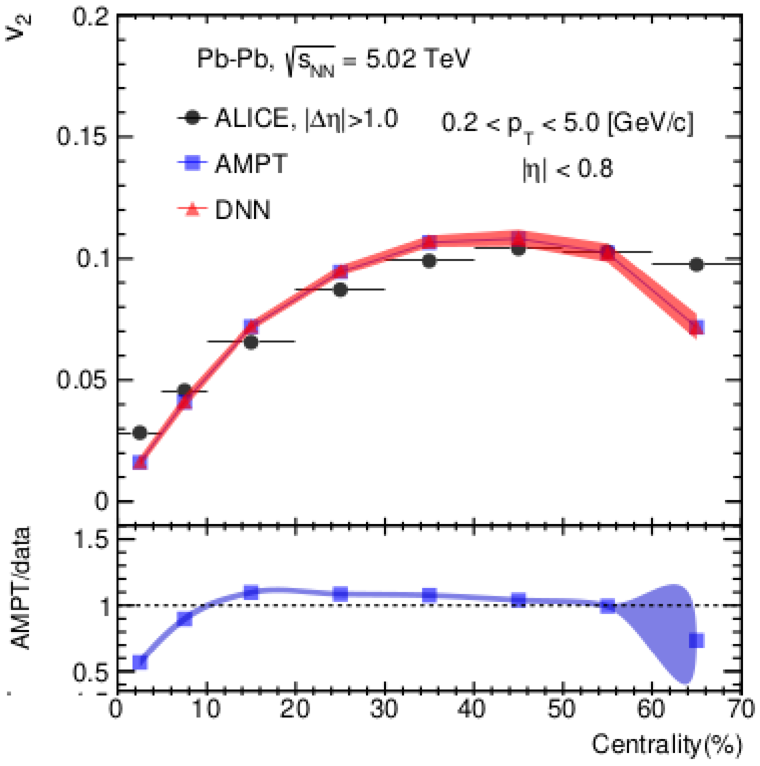
Results on v_2 *vs* centrality

Results on the training data & sets

– **AMPT simulation**: 5.02 TeV Pb-Pb

→ works well [10%:60%] centrality

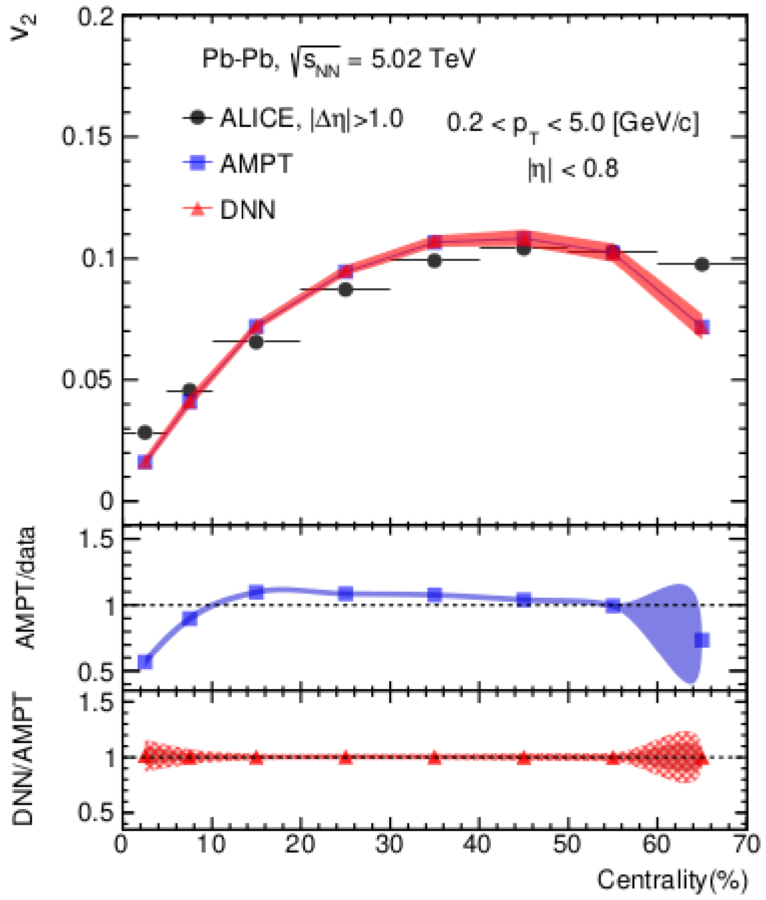
→ low statistics/v_2 values out of this



Results on v_2 *vs* centrality

Results on the training data & sets

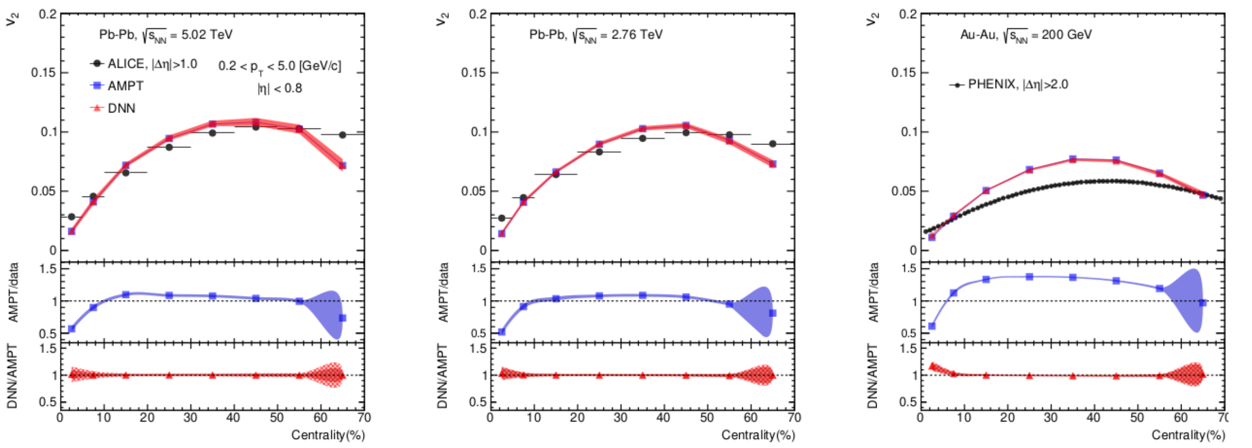
- **AMPT simulation**: 5.02 TeV Pb-Pb
  → works well [10%:60%] centrality
  → low statistics/v_2 values out of this
- **DNN simulation**: same parameters
  → Follows well the AMPT
  → Even including noise *w=0.5*



Results on v_2 *vs* c.m. energy

Results on the training data & sets

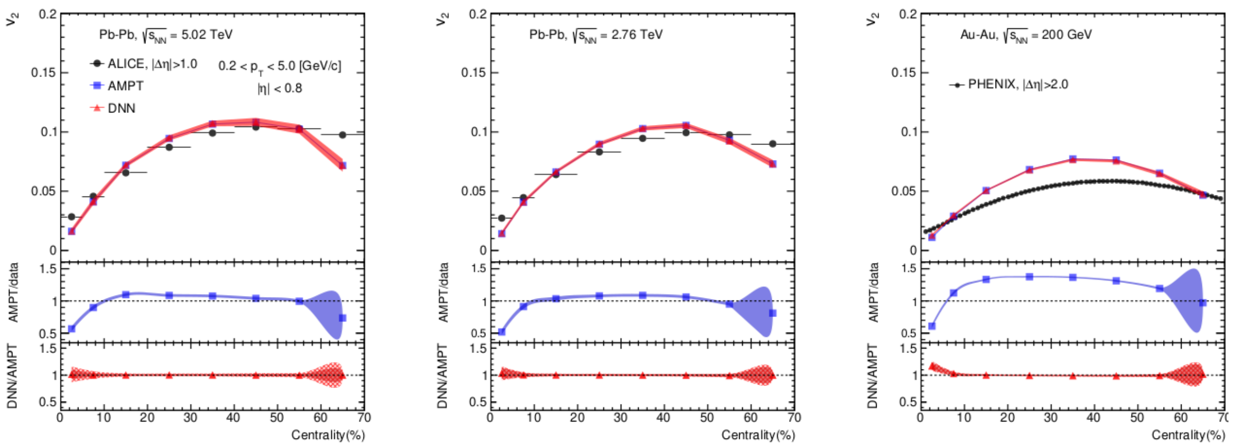
- **AMPT simulation**: 5.02 TeV Pb-Pb
  - → works well [10%:60%] centrality
  - → low statistics/v_2 values out of this
- **DNN simulation**: same parameters
  - → Follows well the AMPT
  - → Even including noise *w=0.5*
- **Predictions for other energies**
  - → similar trends as on the training
  - → AMPT tune for 200 GeV is different

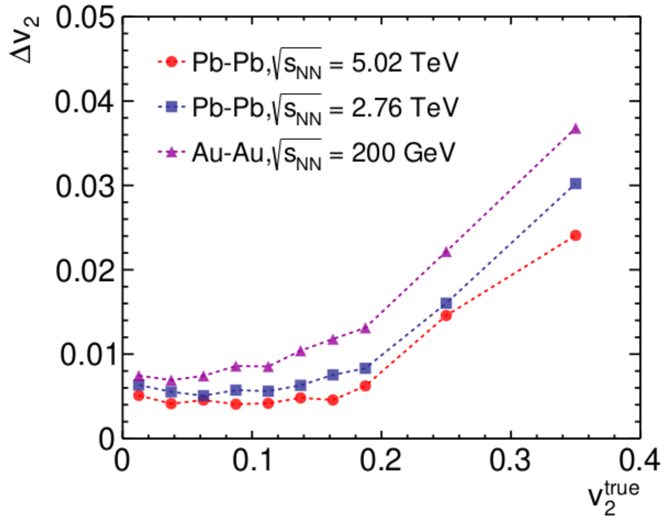


Results on v_2 *vs* c.m. energy

Results on the training data & sets

- **AMPT simulation**: 5.02 TeV Pb-Pb
  - → works well [10%:60%] centrality
  - → low statistics/v_2 values out of this
- **DNN simulation**: same parameters
  - → Follows well the AMPT
  - → Even including noise *w=0.5*
- **Predictions for other energies**
  - → similar trends as on the training
  - → AMPT tune for 200 GeV is different





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

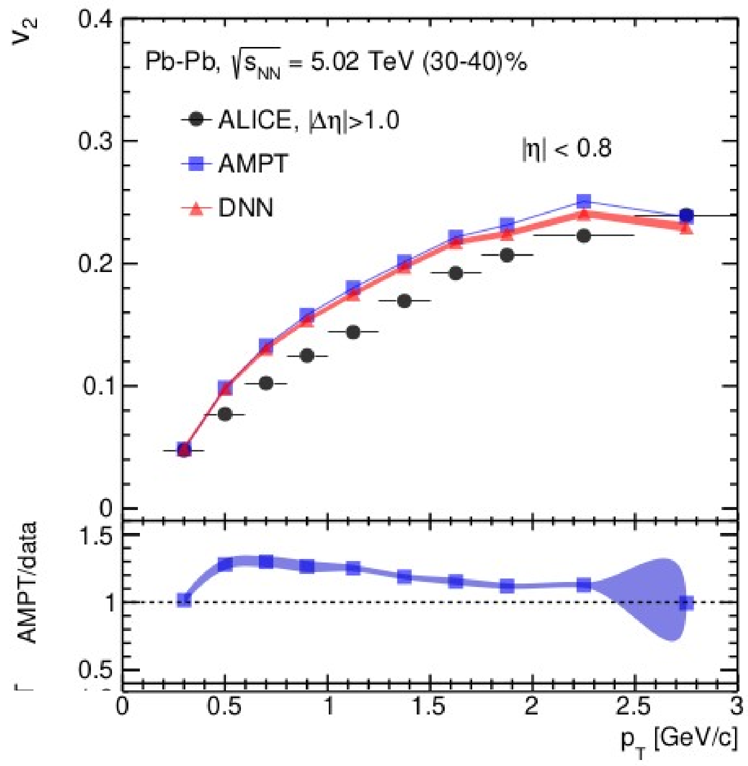
Results on v_2 vs p_T

Results on the training data & sets

– **AMPT simulation**: 5.02 TeV Pb-Pb

→ works well at 30%-40% centrality

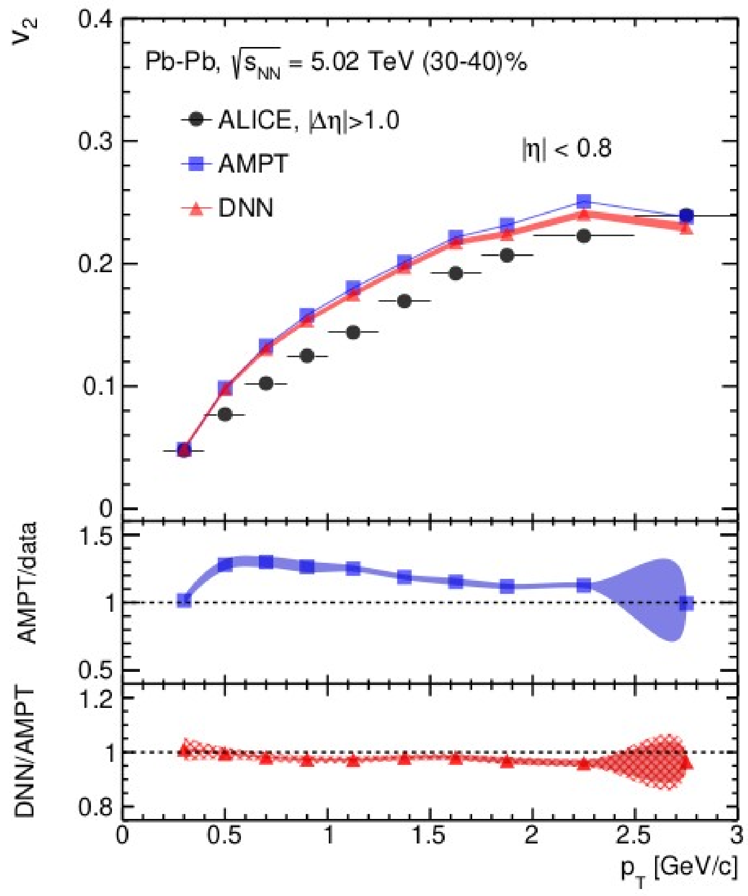
→ low statistics at high p_T



Results on v_2 vs p_T

Results on the training data & sets

- **AMPT simulation**: 5.02 TeV Pb-Pb
  → works well at 30%-40% centrality
  → low statistics at high p_T
- **DNN simulation**: same parameters
  → Follows well the AMPT
  → Even including noise $w=0.5$



Conclusions

- **Is it possible to estimate the elliptic flow by ML?**
  - Get best Min. Bias. Monte Carlo simulation data and train the well-designed DNN system...
    - → More sophisticated NN, the less epoch needs
    - → Un-correlated noise can be even *w=1*
    - → AMPT & DNN correlates well for all centrality
    - → Best correlation is for the highest statistic
    - → Energy scaling is well preserved (non-linear)
    - → The $v_2(p_T)$ is also preserved
- **What is missing...**
  - Test of correlated noise (detector setup, etc)
  - Train with real data

BACKUP