Artificial neural network approach to detector configuration optimization based on the impact parameter estimation problem.

Authors: Galaktionov K.A., Roudnev V.A., Valiev F.F., Feofilov G.A.

St. Petersburg University

7th International Conference on Particle Physics and Astrophysics

https://indico.particle.mephi.ru/event/436/

22.10.2024 – 25.10.2024

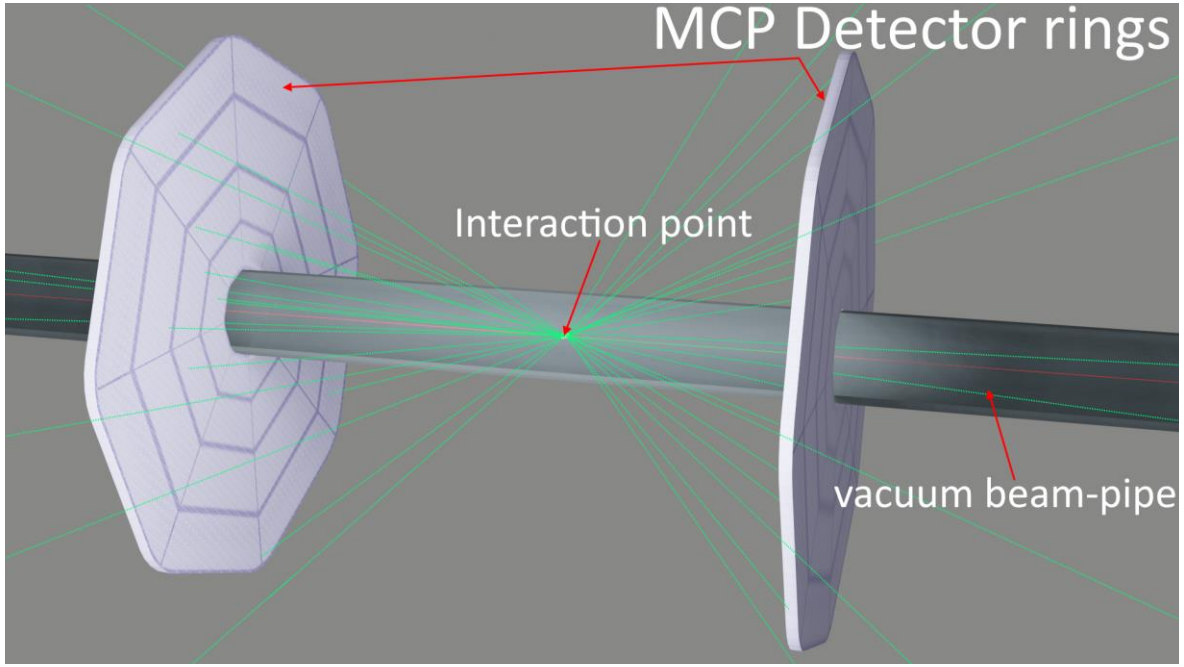
24 october 2024

The problems of event parameters estimation

St Petersburg University

Studied problems

- Estimate the value of the impact parameter
- Select head-on collisions (small impact parameter)
- Estimate the z-coordinate of the collision vertex



Scheme of investigated detectors geometry

We used MC generated data of Au+Au collisions at energies $\sqrt{s_{NN}} \approx 11$ GeV, which consists of two datasets:

- 200 000 events generated by **QGSM**[1] model
- 360 000 events generated by **EPOS**[2] model

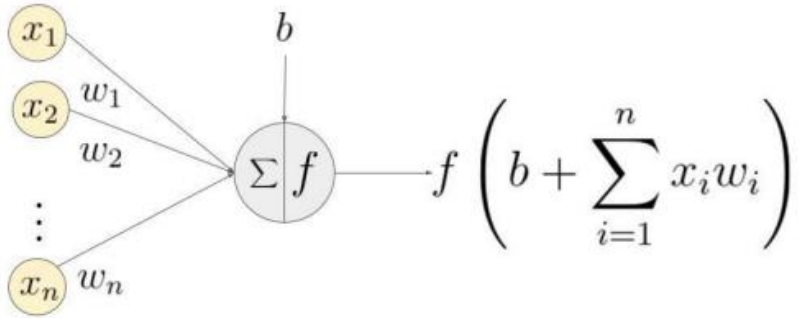
Hits information that was used for feature engineering:

- Which cell registered hit
- Number of detector ring
- Particle time of flight

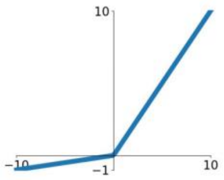
[1] Amelin N. S., Gudima K. K., Toneev V. D. Ultrarelativistic nucleus-nucleus collisions within a dynamical model of independent quark - gluon strings // Sov. J. Nucl. Phys. 1990. V. 51(6), P. 1730-1743

[2] Werner, Klaus and Liu, Fu-Ming and Pierog, Tanguy Parton ladder splitting and the rapidity dependence of transverse momentum spectra in deuteron-gold collisions at the BNL Relativistic Heavy Ion Collider // Physical Review C 2006, V. 74

Used artificial neural networks

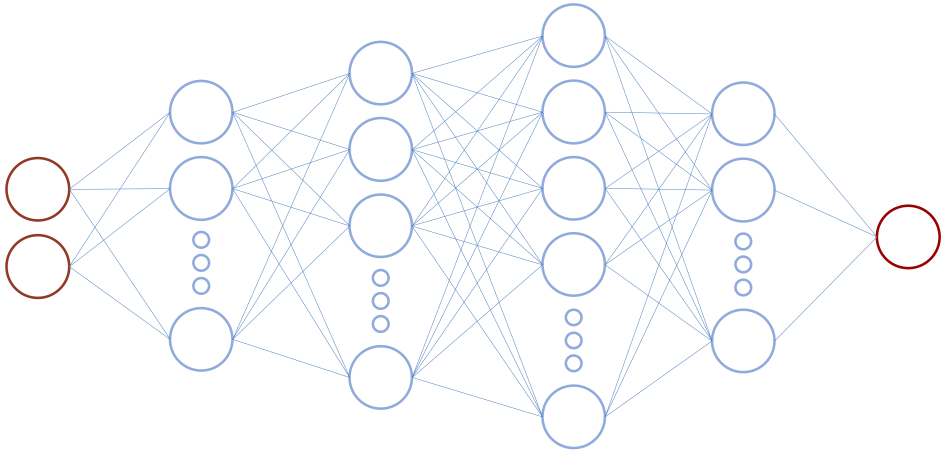


Scheme of an artificial neuron model



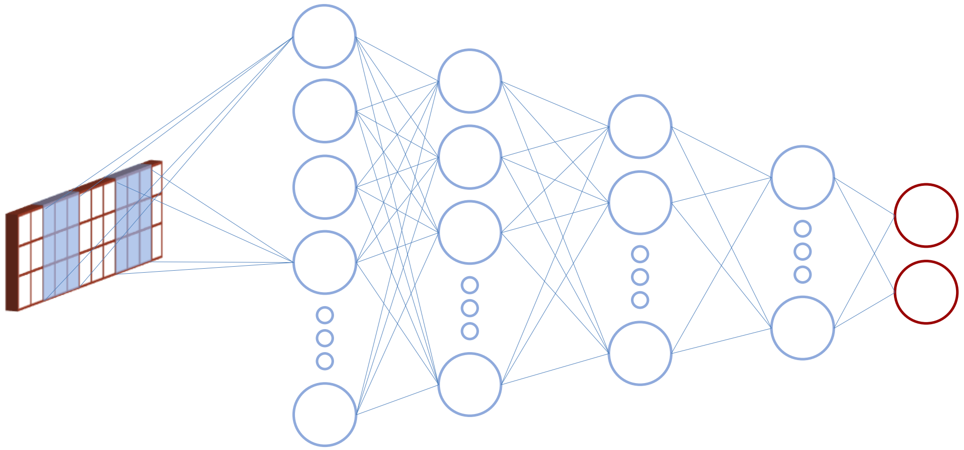
Leaky ReLU

$f(x) = \max(0.01x, x)$



Example of used dense neural network architecture, solving regression problem.

Input – 2 event features, 4 hidden layers (4, 8, 16, 4 neurons), output – 1 neuron – estimated impact parameter value



Example of used dense neural network with convolutional layer, solving classification problem.

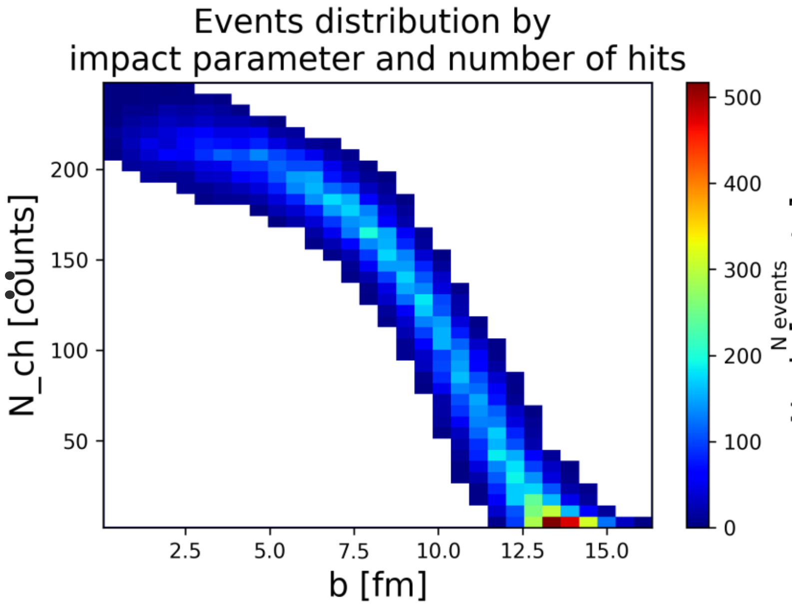
Input – Table of particles information (3x150 features), convolutional layer (16 filters 3x6), 3 hidden layers (128, 64, 32 neurons), output – 2 neurons – probabilities of an event belonging to each class.

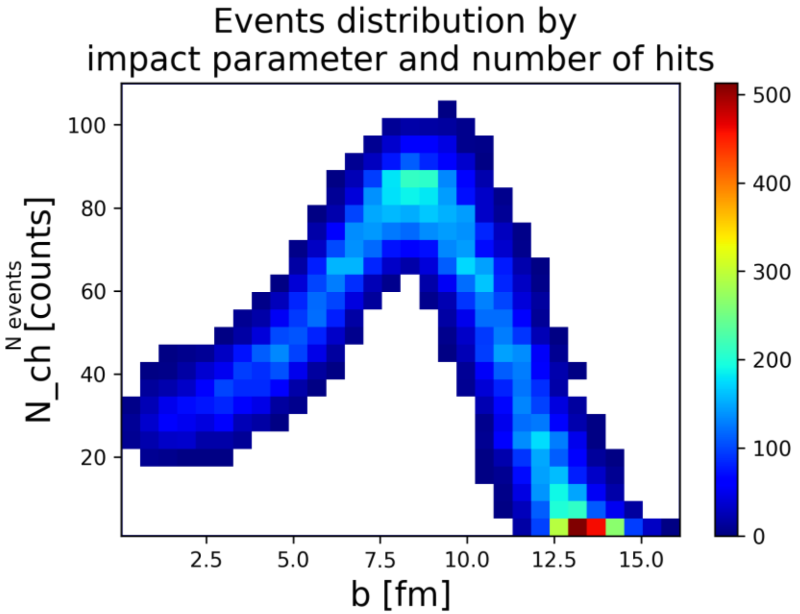
Leaky ReLU activation function. Optimizer – Adam. Optimized functions – mean squared error, binary cross entropy.

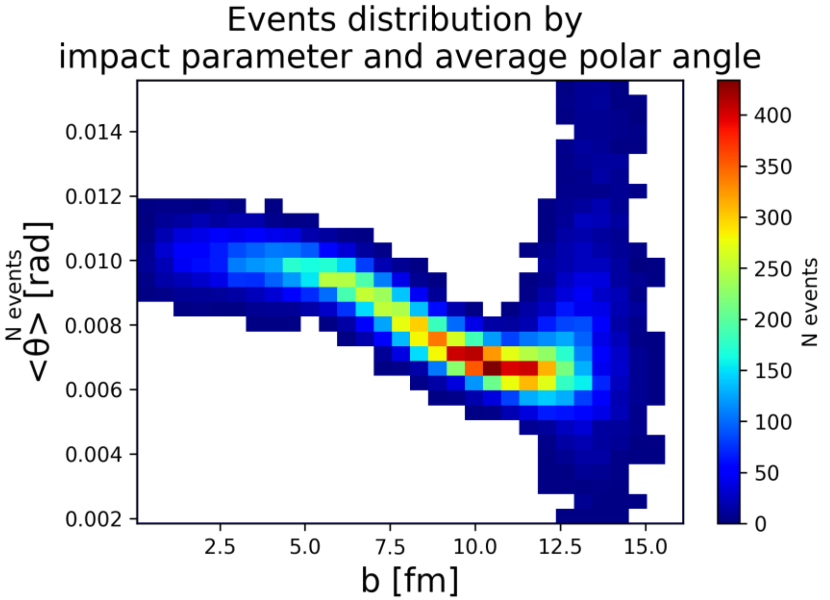
Multiplicity of charged particles and the average polar angle of trajectories

Multiplicity, R=1m | Multiplicity, R=0.25m | Average angle

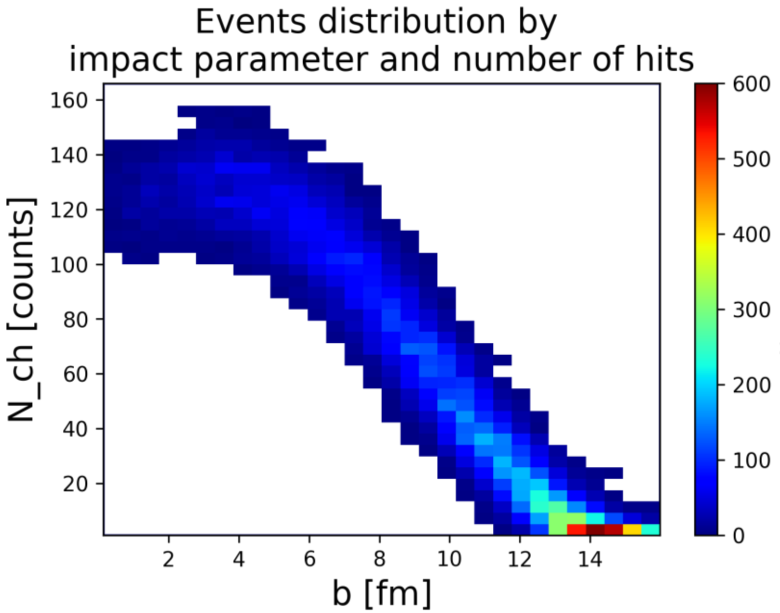
QGSM:

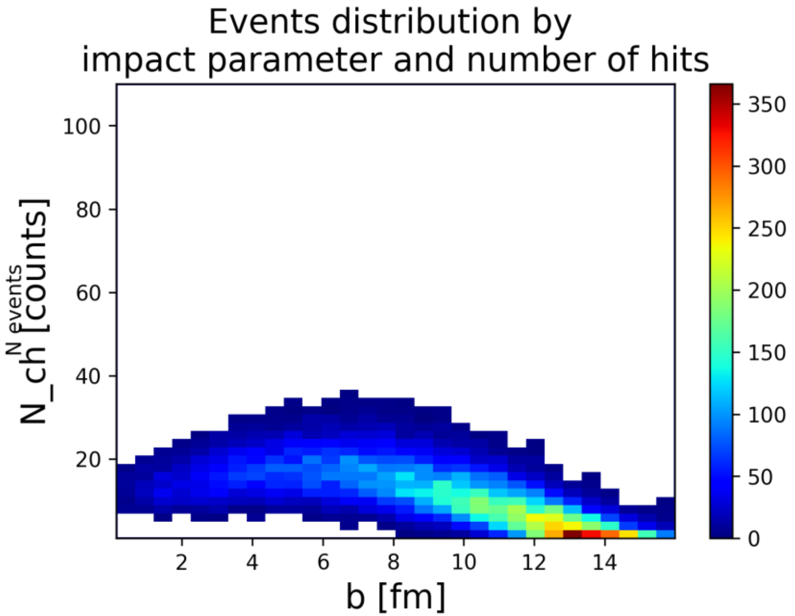


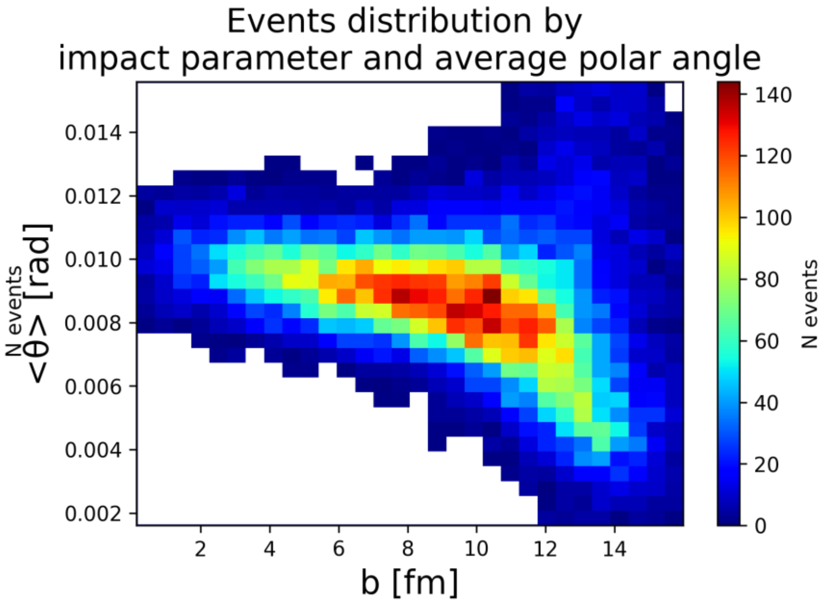




EPOS:





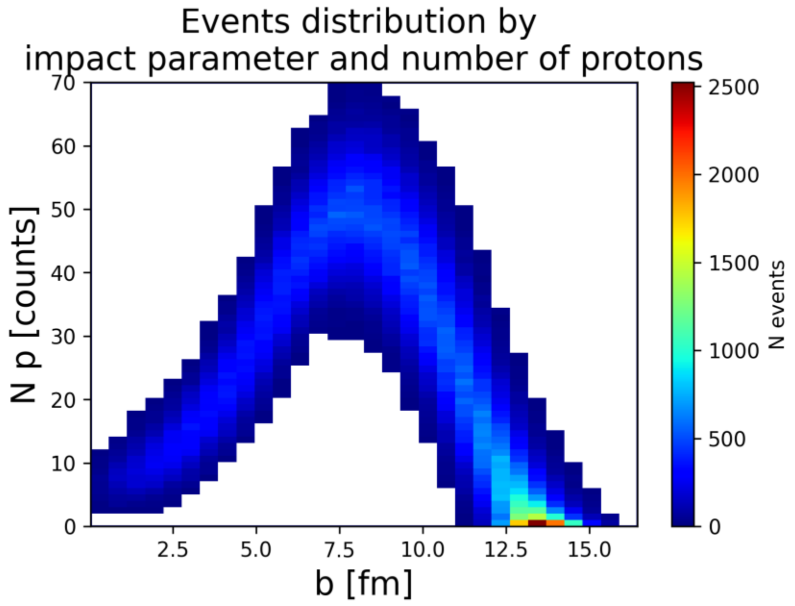


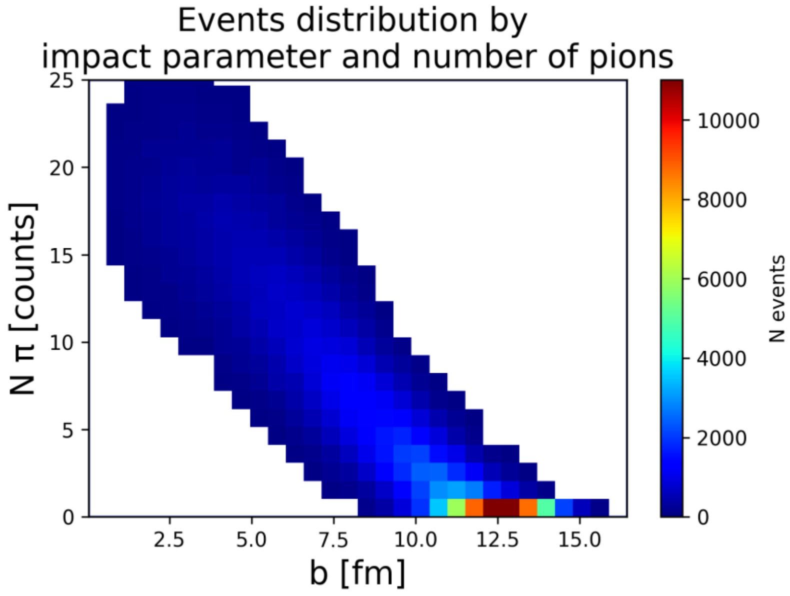
Difference in multiplicities of pions and protons

Protons multiplicity

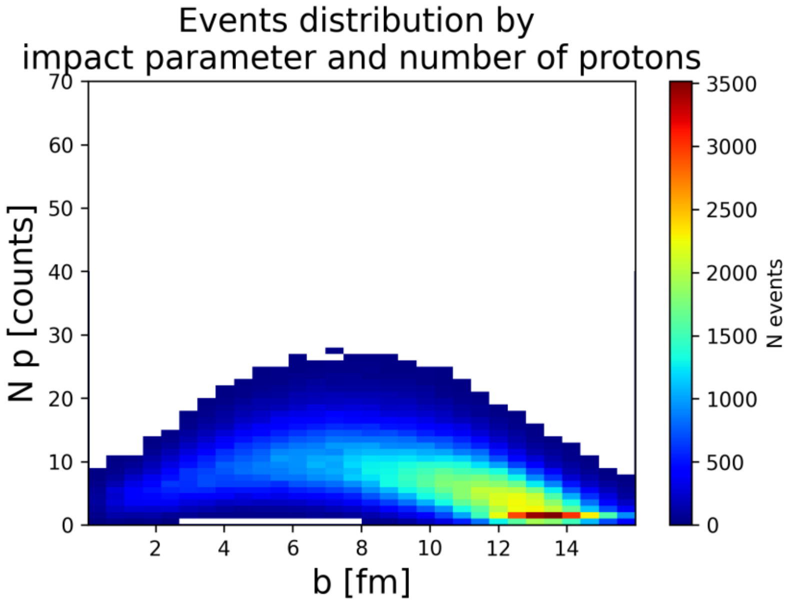
Pions multiplicity

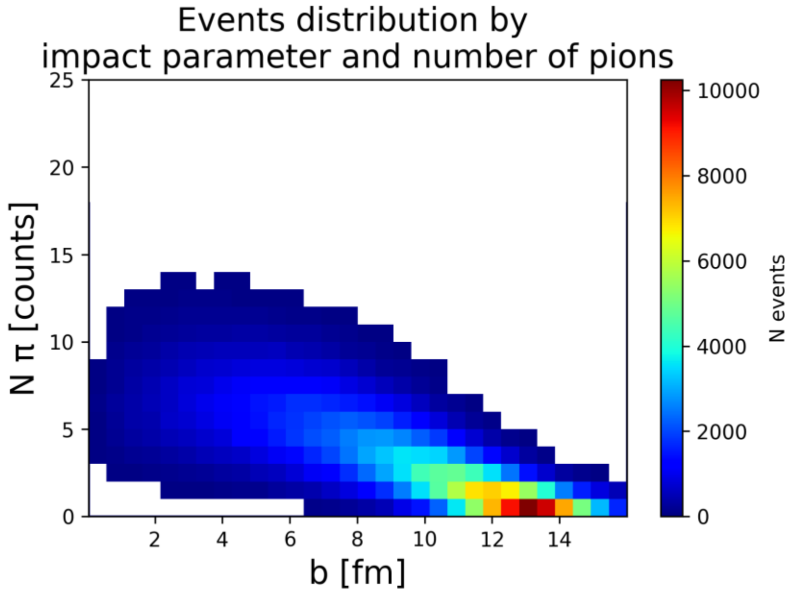
QGSM:





EPOS:





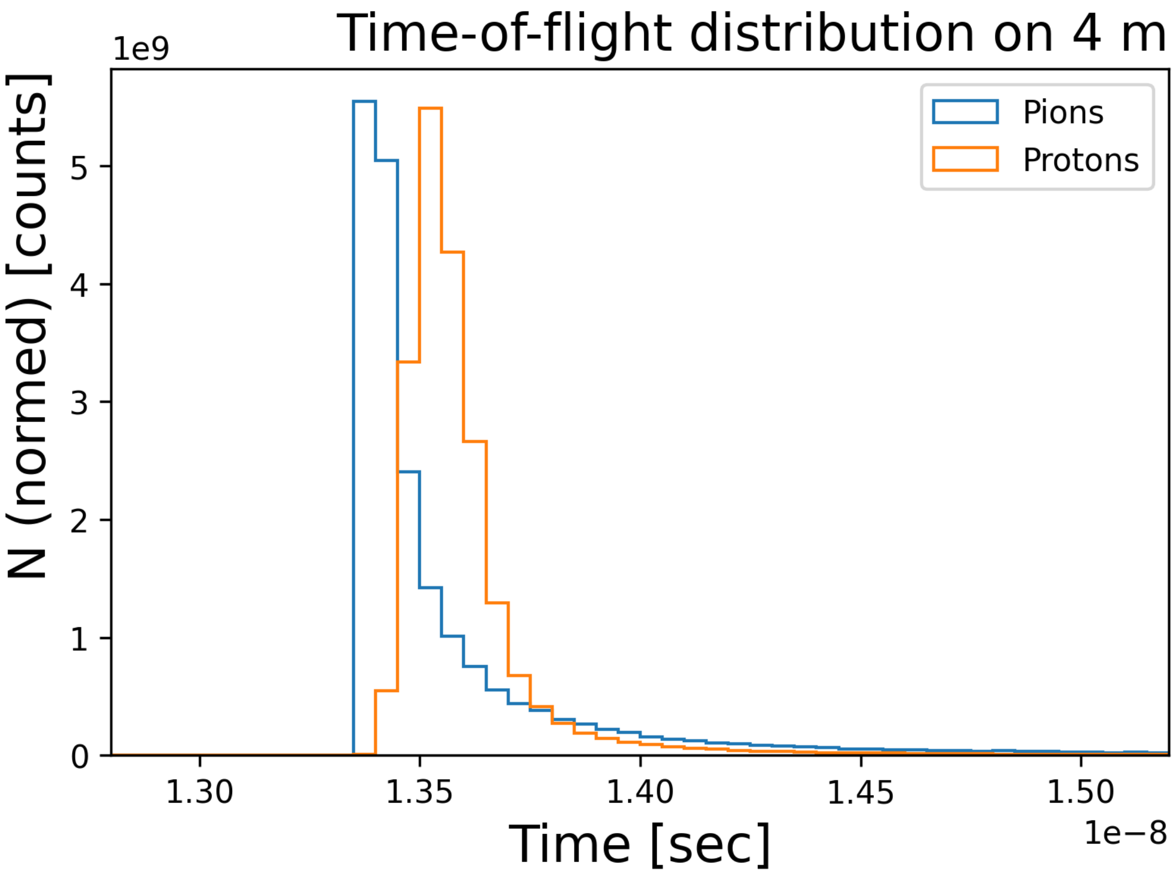
Time of flight of particles

We constructed a new hit feature from time of flight to distinguish between particle types:

$$v = \frac{1}{t - t_{0i}}$$

where t – time of flight of particle, t_{0i} - average time of flight of pions on detector number i .

QGSM:



EPOS:

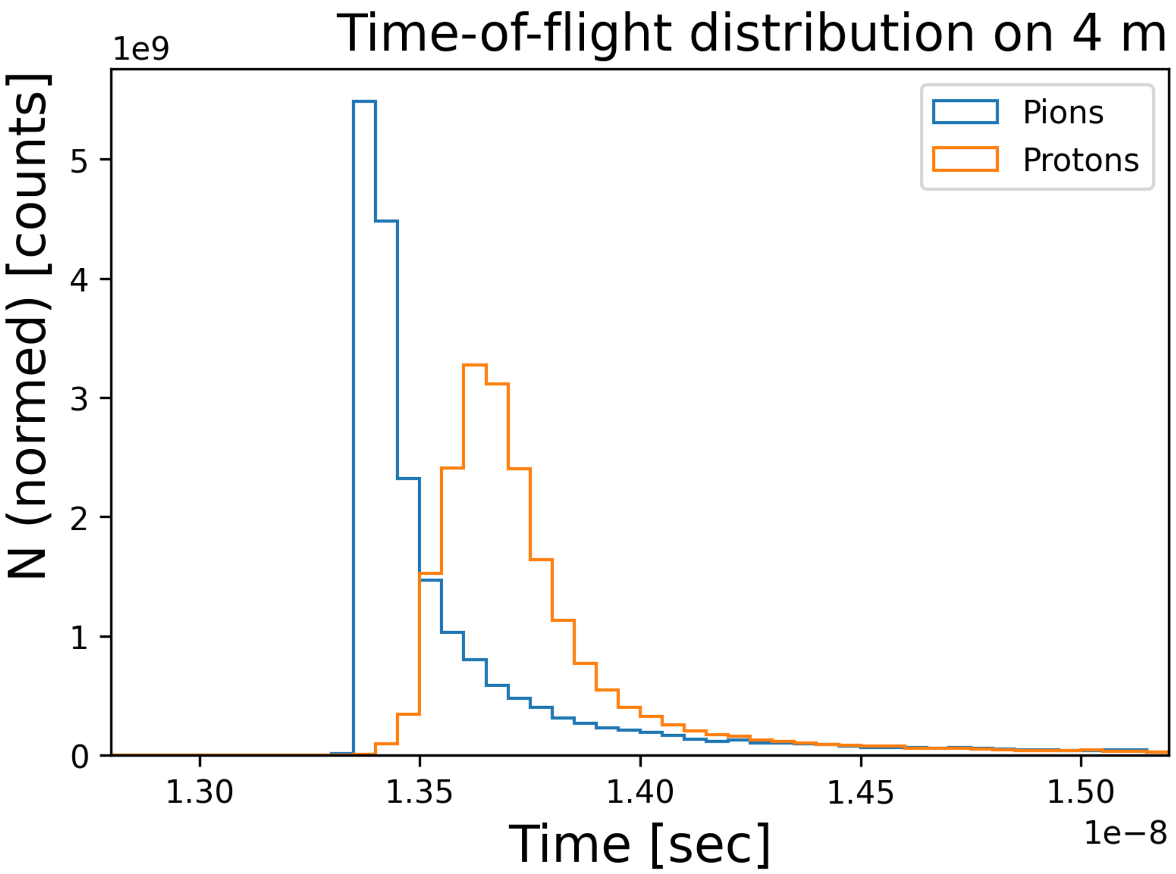


Table of NN performance

Event features (Number of features)	Binary classificatory threshold [fm]	QGSM			EPOS		
		MSE [fm]	TPR [%]	FPR [%]	MSE [fm]	TPR [%]	FPR [%]
Time of flight (3x150)	5	0,68	98,6	4,3	1,53	91,7	16,4
Time of flight (3x150)	1		90,3	6,2		94,0	17,8
Multiplicity + angle (2)	5	0,77	97,7	5,8	2,06	88,1	38,1
Multiplicity + angle (2)	1		98,9	8,8		77,2	21,1

Here we used detector system consisting of pair of rings with R=25cm, r=2.5cm, L=4m, Δt=50ps, 352 cells.

TPR – true positive rate, FPR – false positive rate.

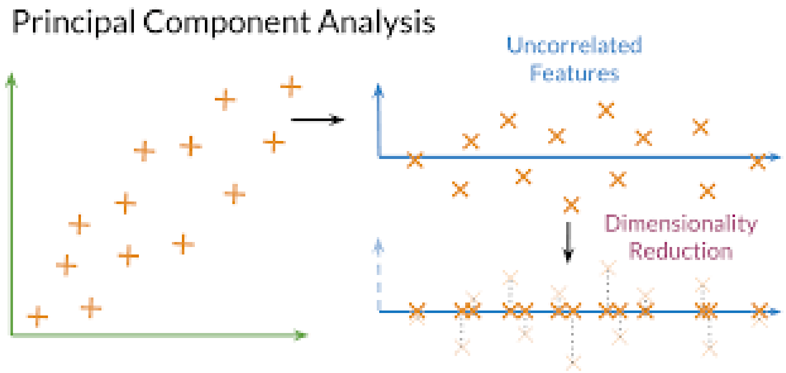
Search for universal event characteristics

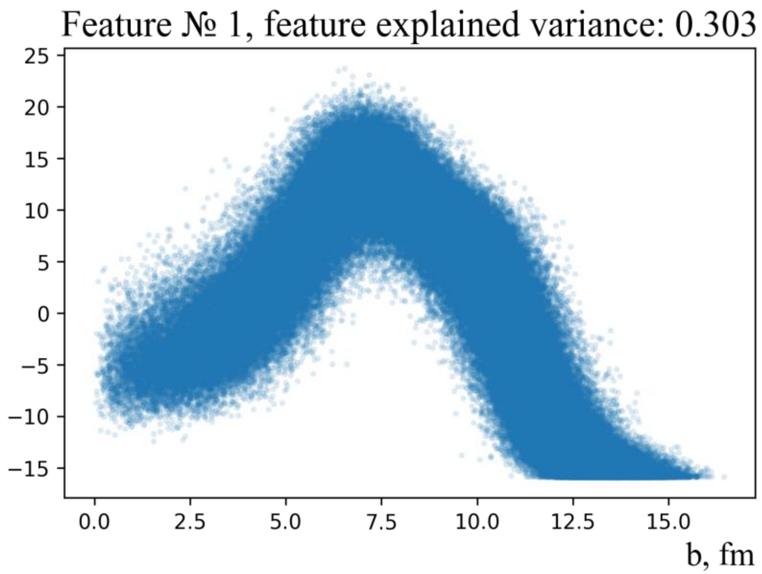
The idea of method

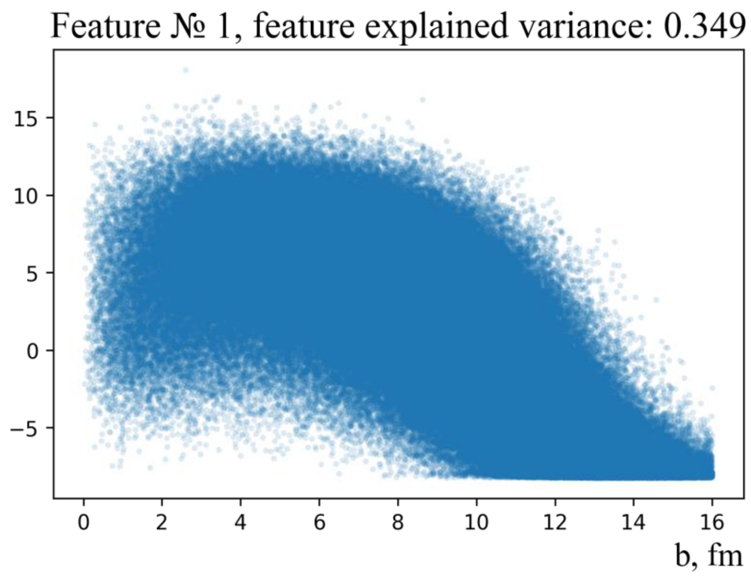
QGSM:

EPOS:

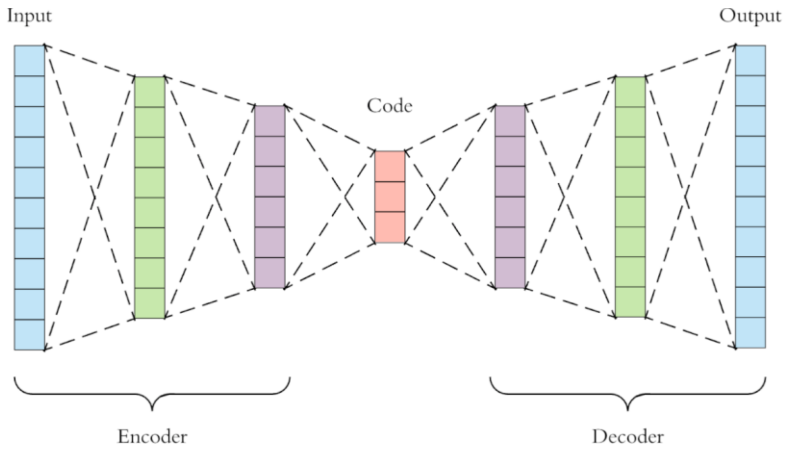
PCA:

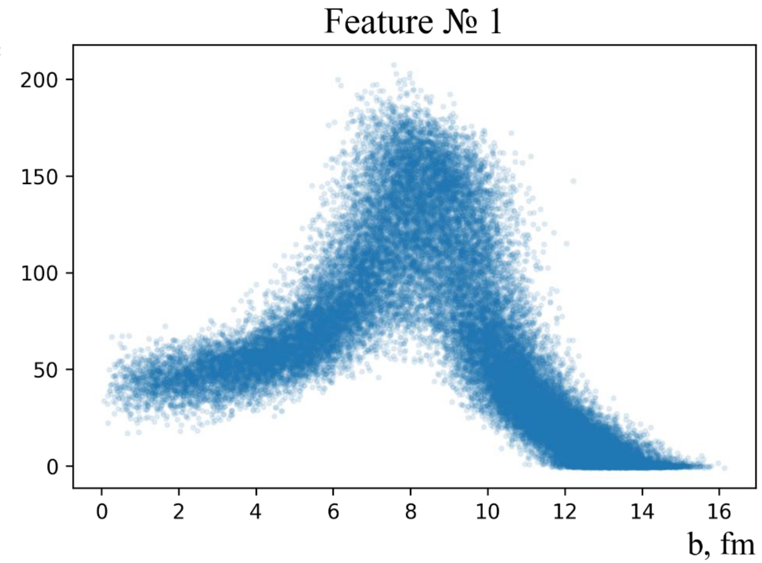


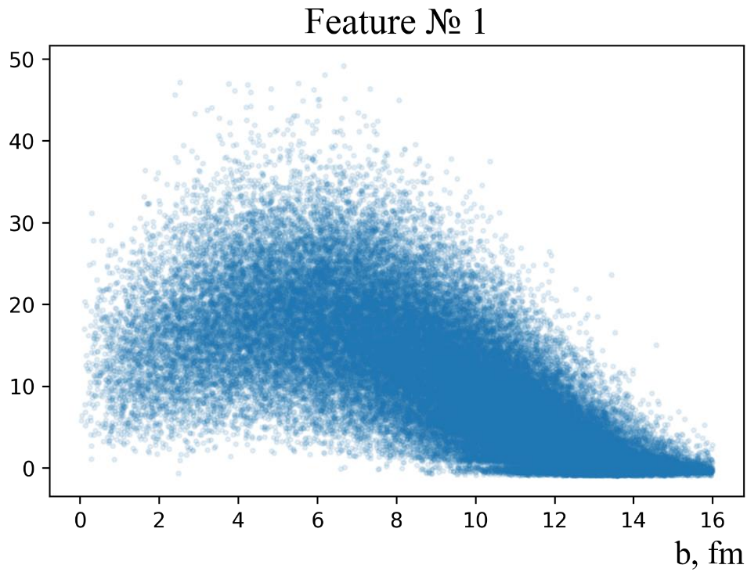




Auto-encoder:

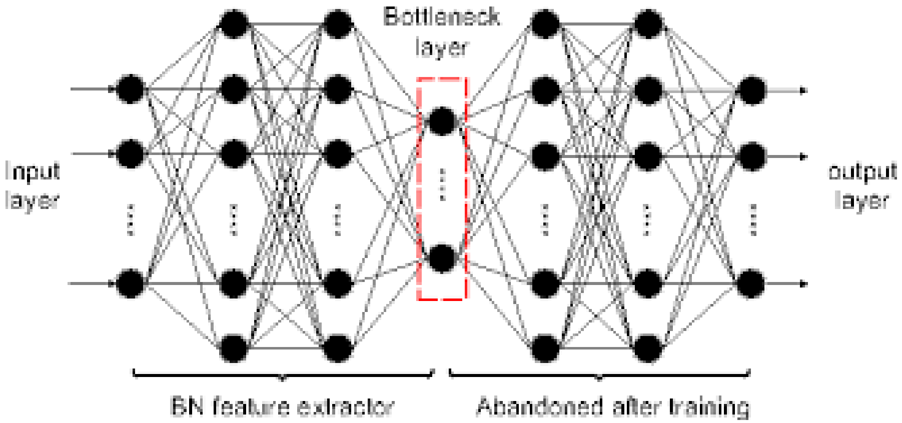


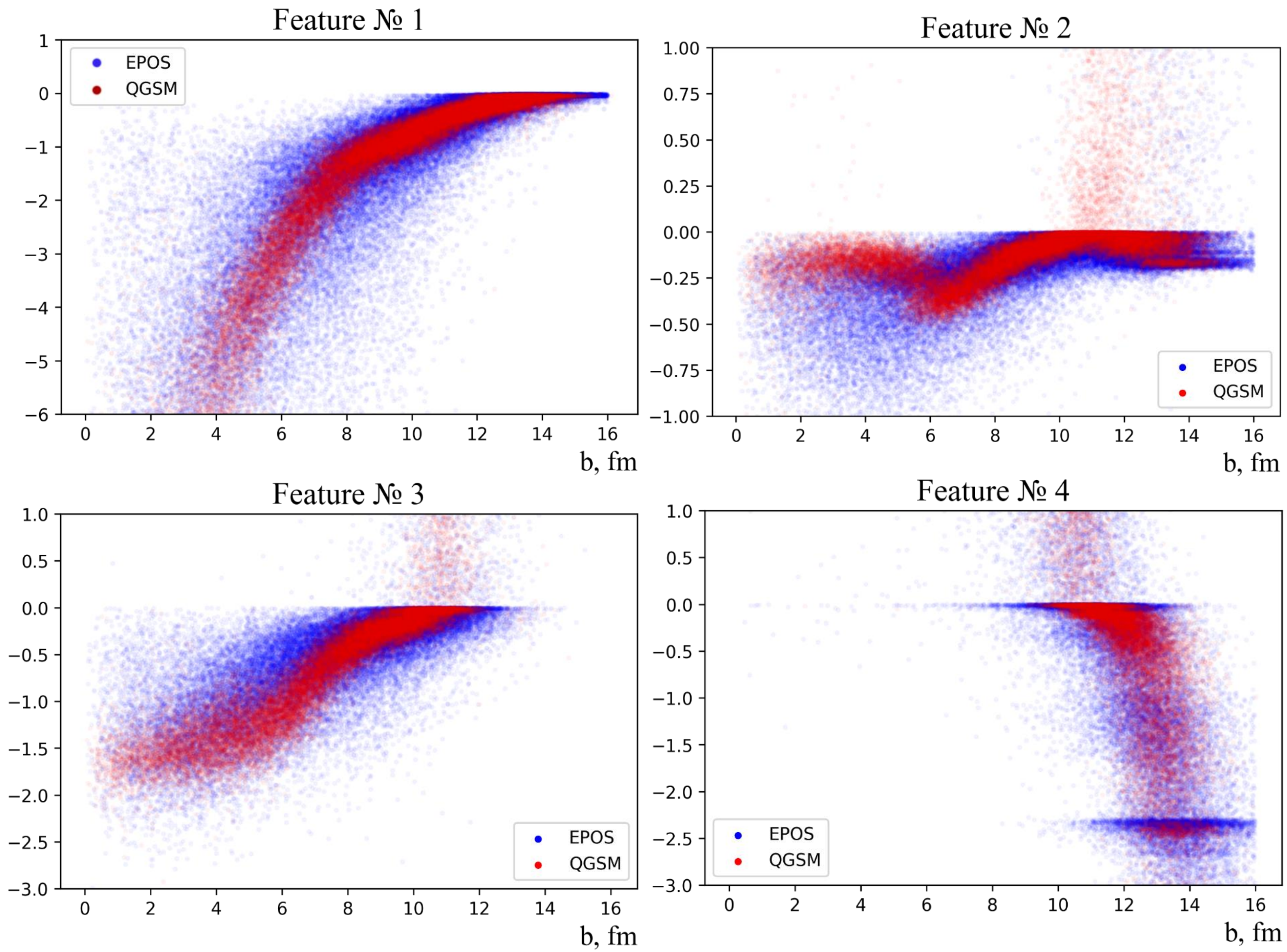




Universal model trained on mixed dataset

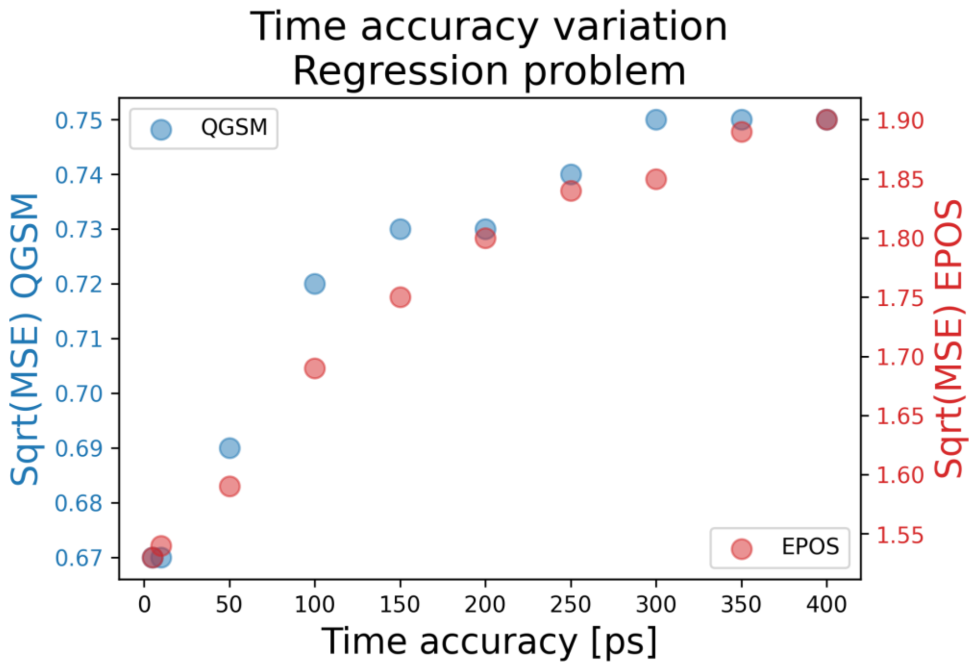
Here we present a view of the internal representation of event features in the neural network extracted using the “bottleneck” method.

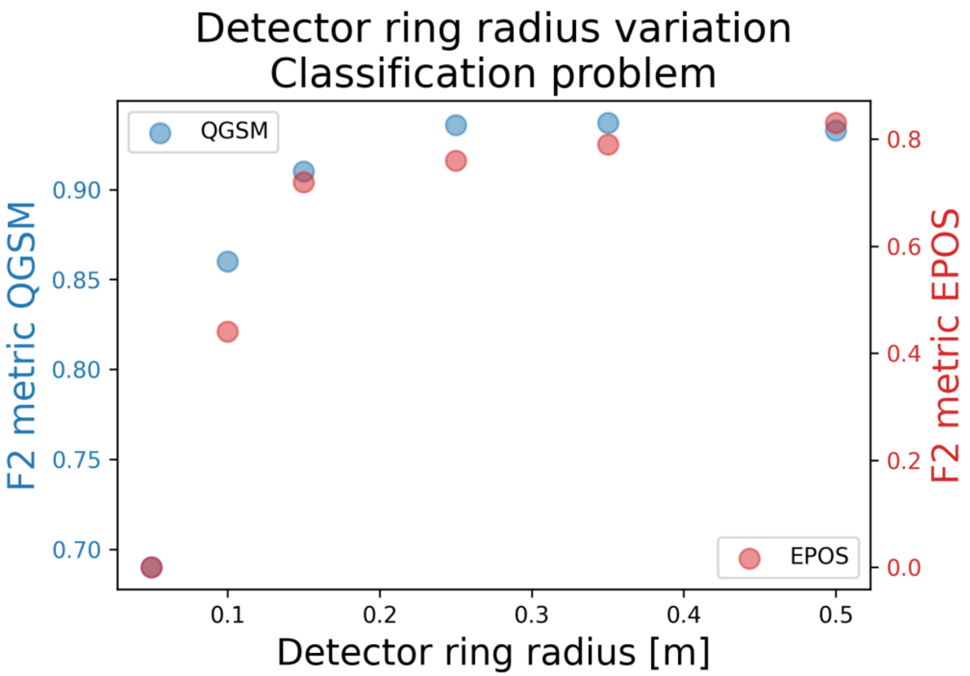


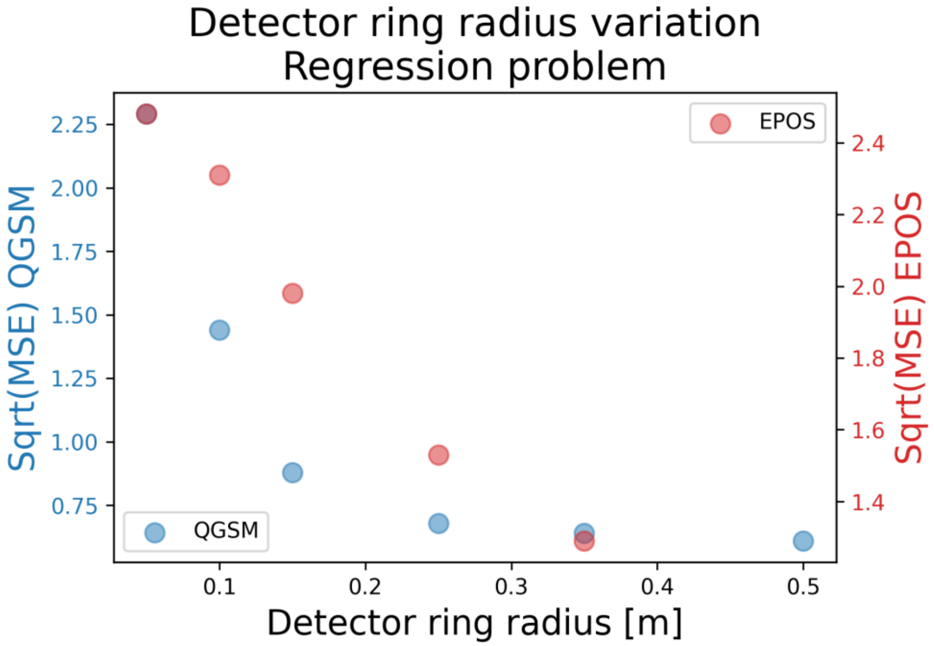


Optimization

The main idea of the optimization method is to vary one of the geometry parameters, leaving the others constant, and then to train the model to solve impact parameter estimation problem in each variation. The quality of results will change and will be different for the two datasets, but the patterns and key points will remain the same.





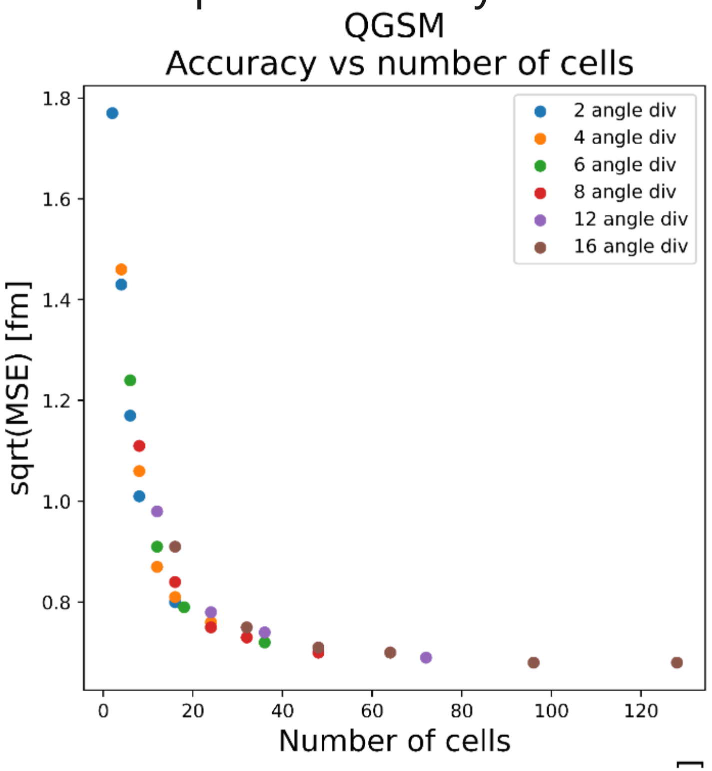


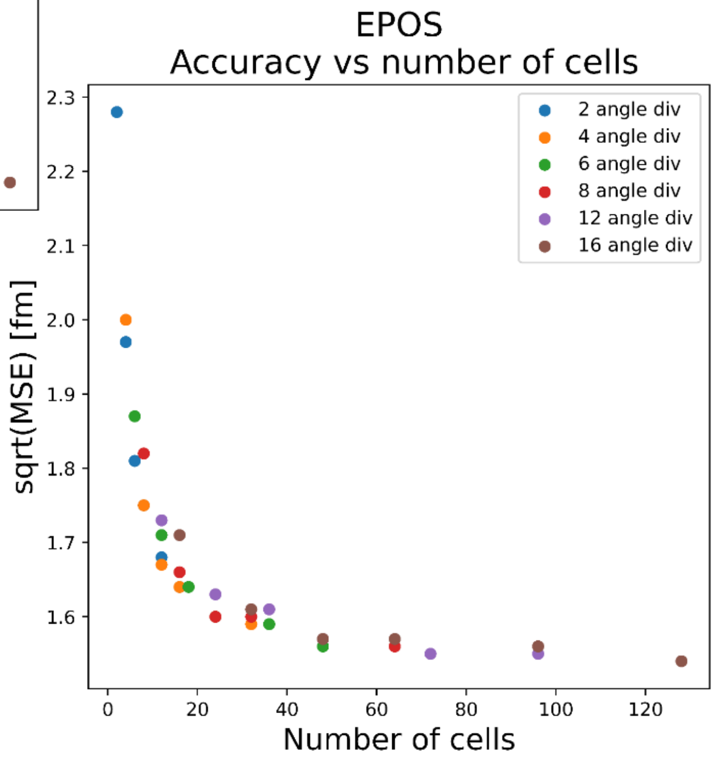
$$F_\beta = (1+\beta^2) * \frac{precision * recall}{\beta^2 * precision + recall}$$

Used to calculate F2 metric for classification problem

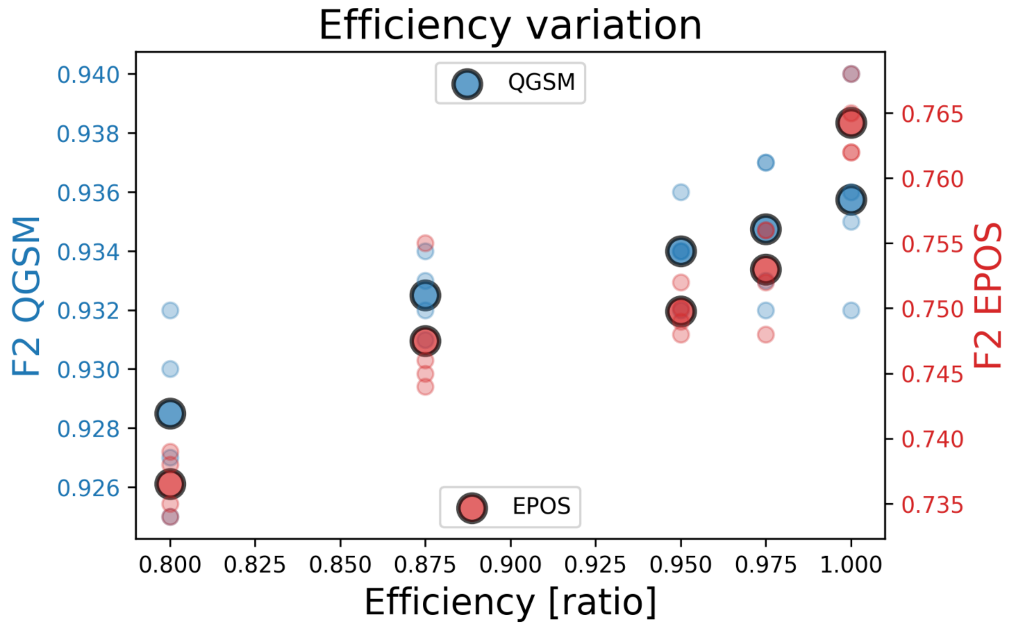
Optimization

Optimization by the number of detector cells

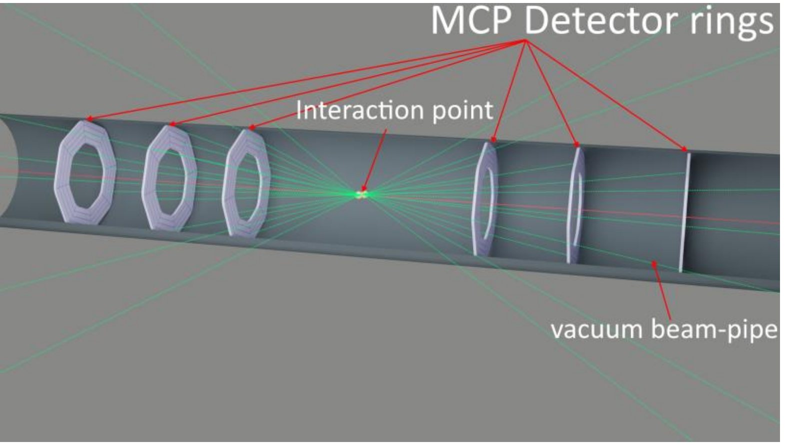




Optimization by the detector efficiency

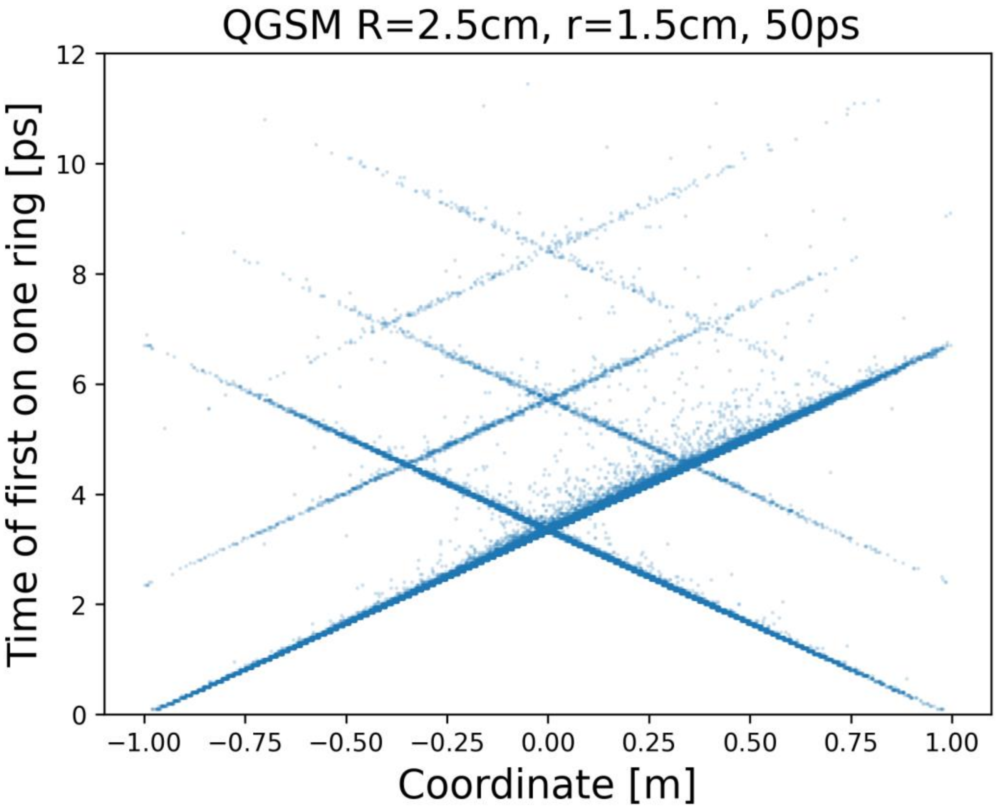


Coordinate prediction

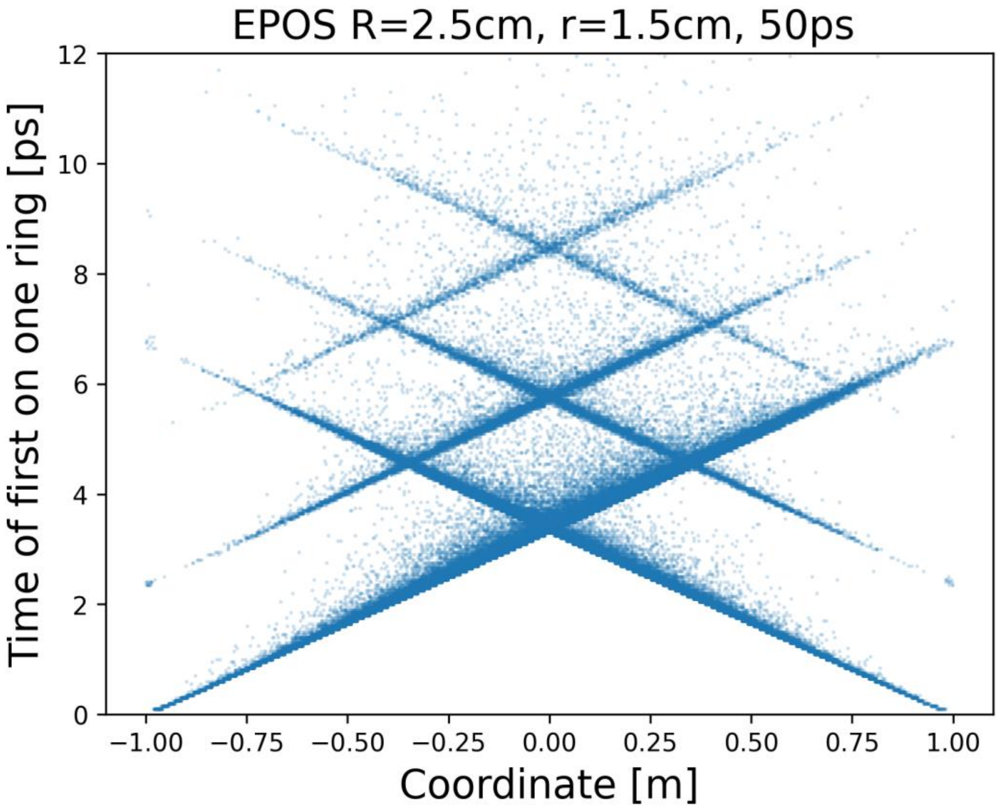


Vertex coordinate was taken from normal distribution with $\mu=0$, $\sigma=30$cm.
Mainly the prediction is done using time of flight of the fastest particle.
Time of flight dependence on coordinate is linear on each ring.

QGSM:



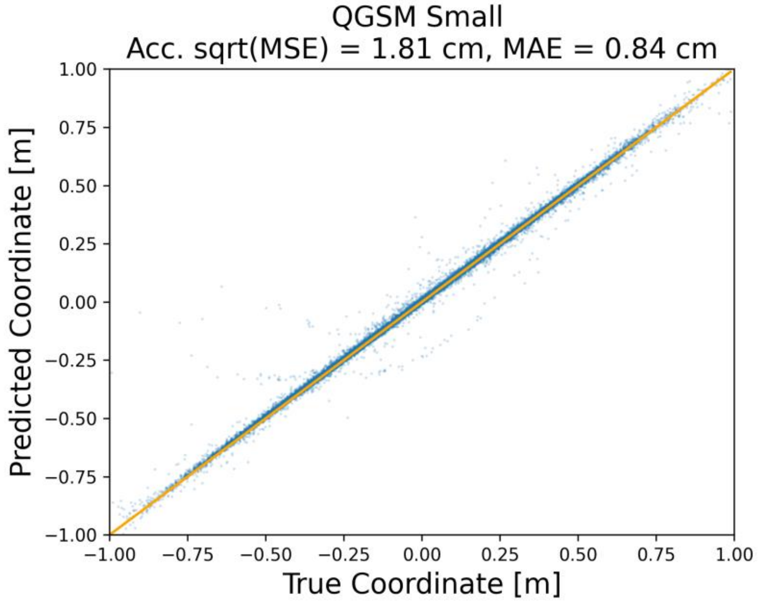
EPOS:

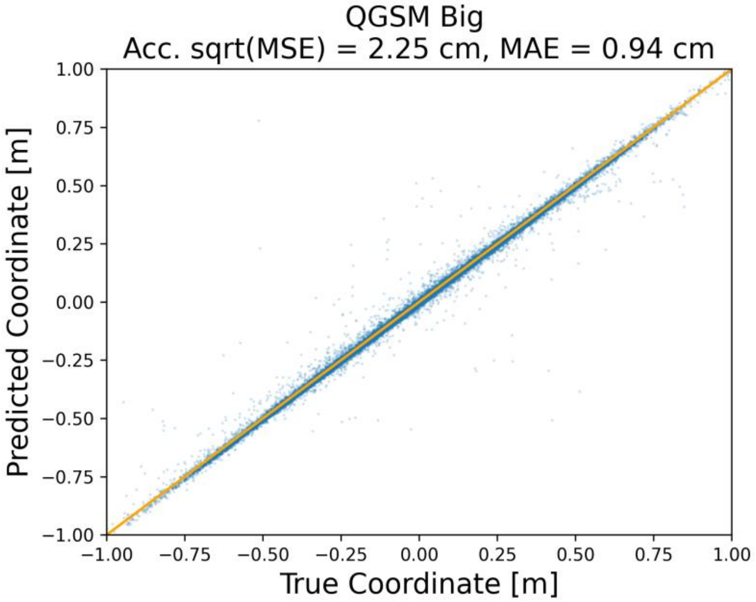


Coordinate prediction

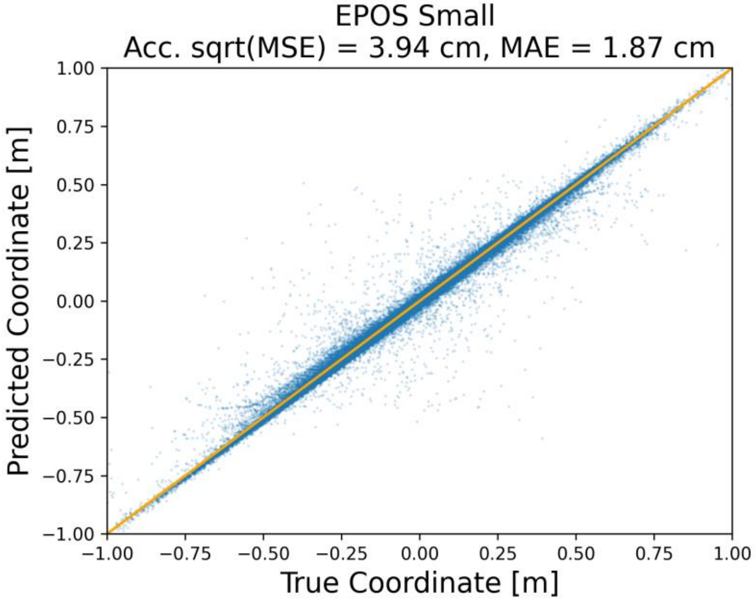
Small detectors configuration | Big detectors configuration

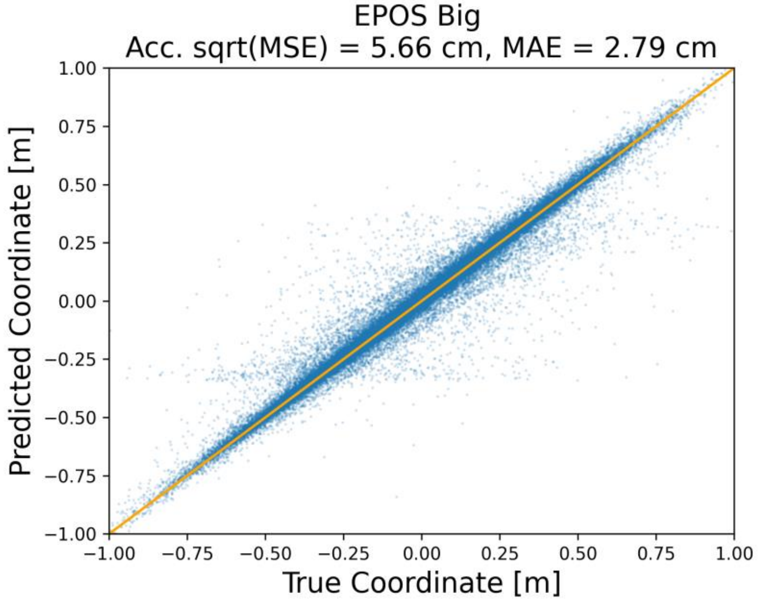
QGSM:





EPOS:

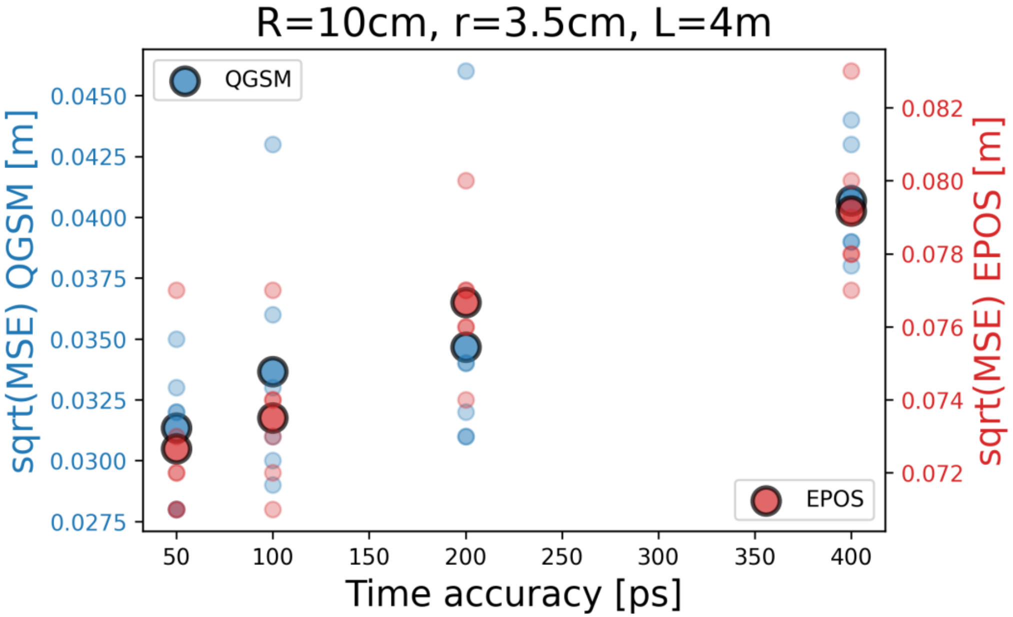


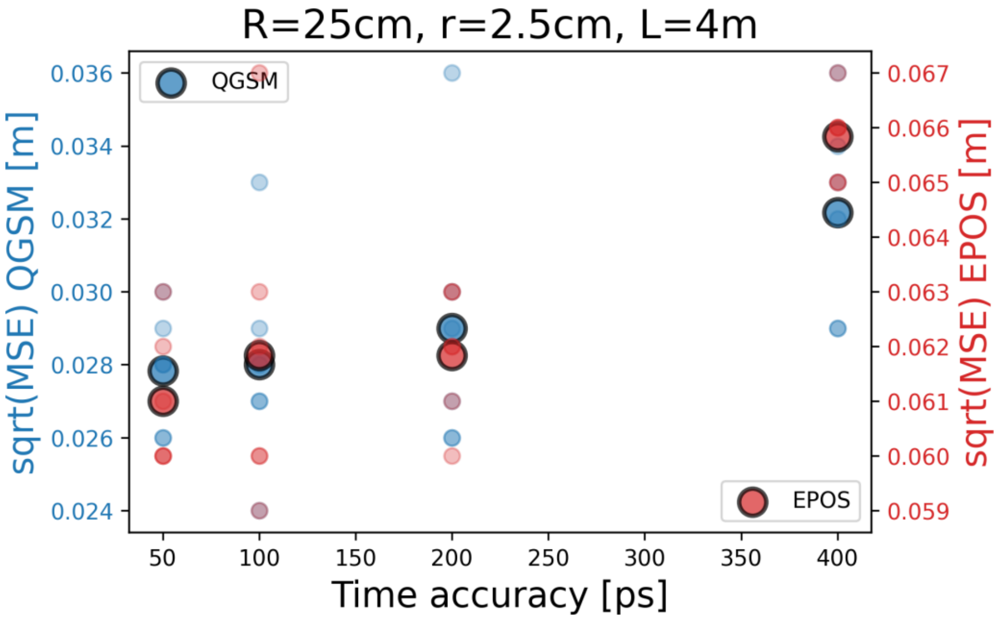


St Petersburg University

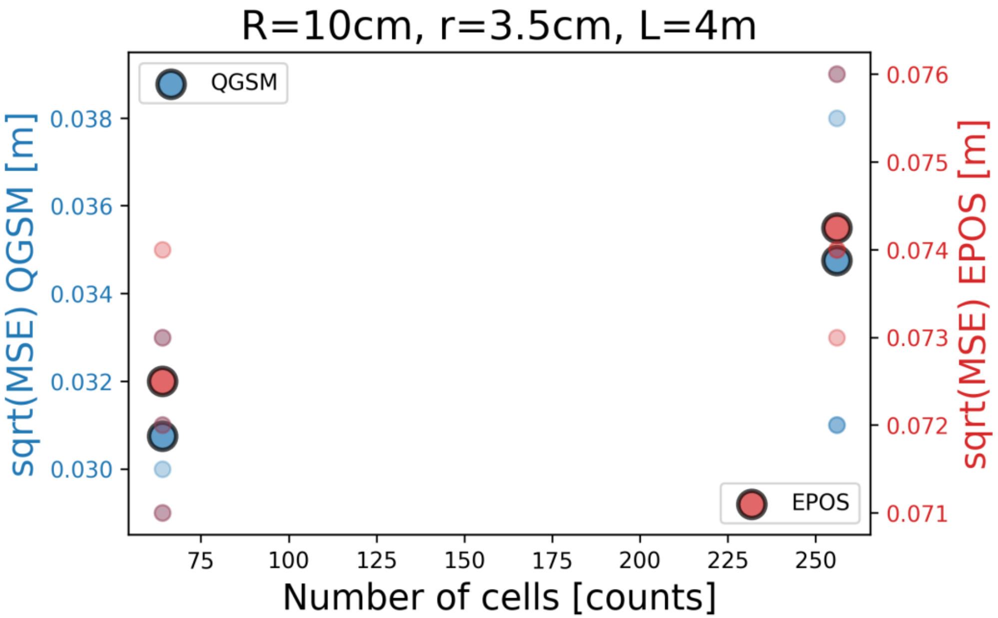
Optimization based on coordinate prediction

Time accuracy:

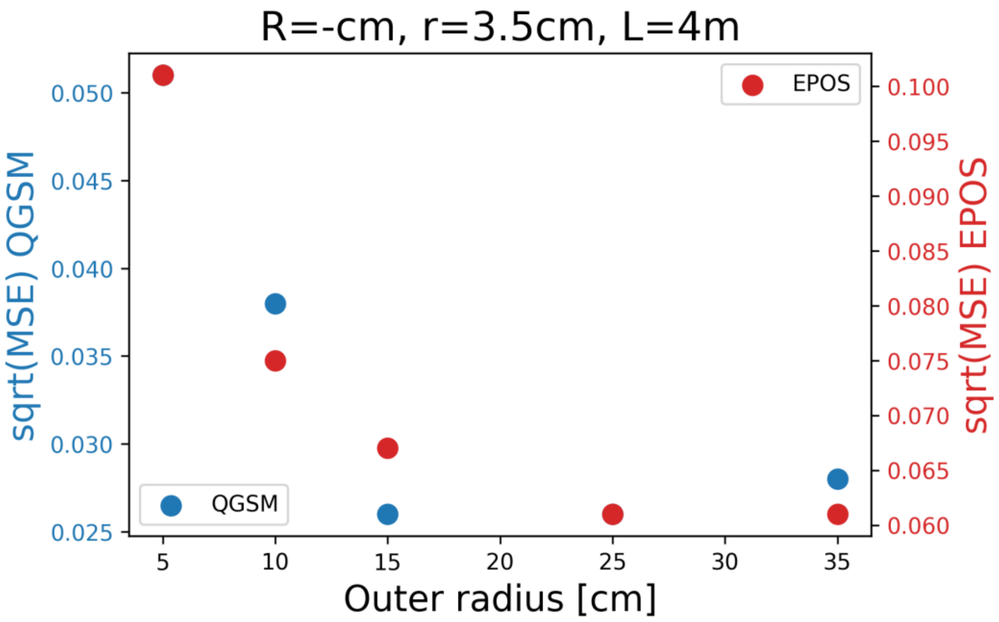




Amount of cells:



Rings radius:



Conclusions

St Petersburg University

New method

Investigated method is capable of extracting information from non-obvious event features.

Hidden dependencies

With the help of artificial neural networks, it has become possible to extract hidden patterns in data from different sources.

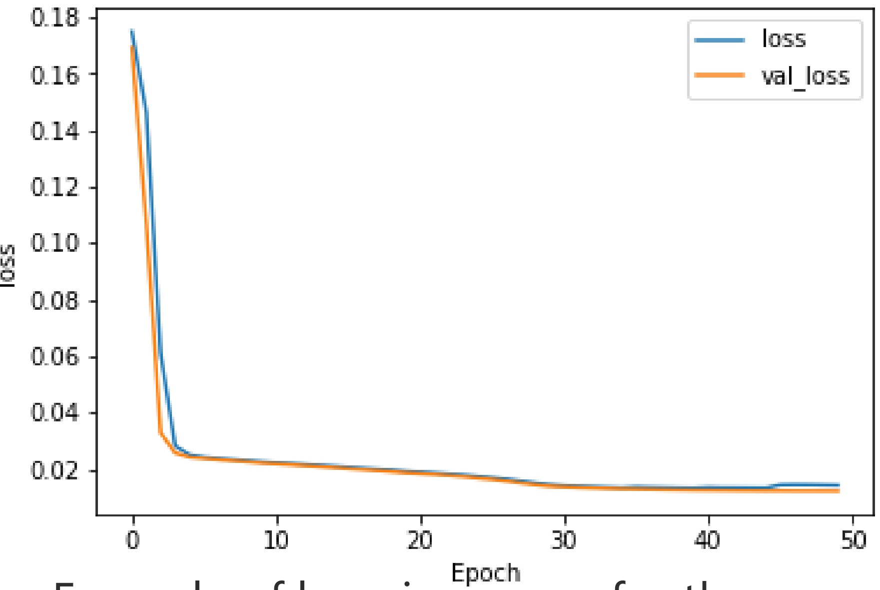
Optimization technique

The proposed optimization method uses extracted data similarities and, despite differences in the characteristics of events and the quality of problem solving, makes it possible to evaluate the best properties of the detector systems.

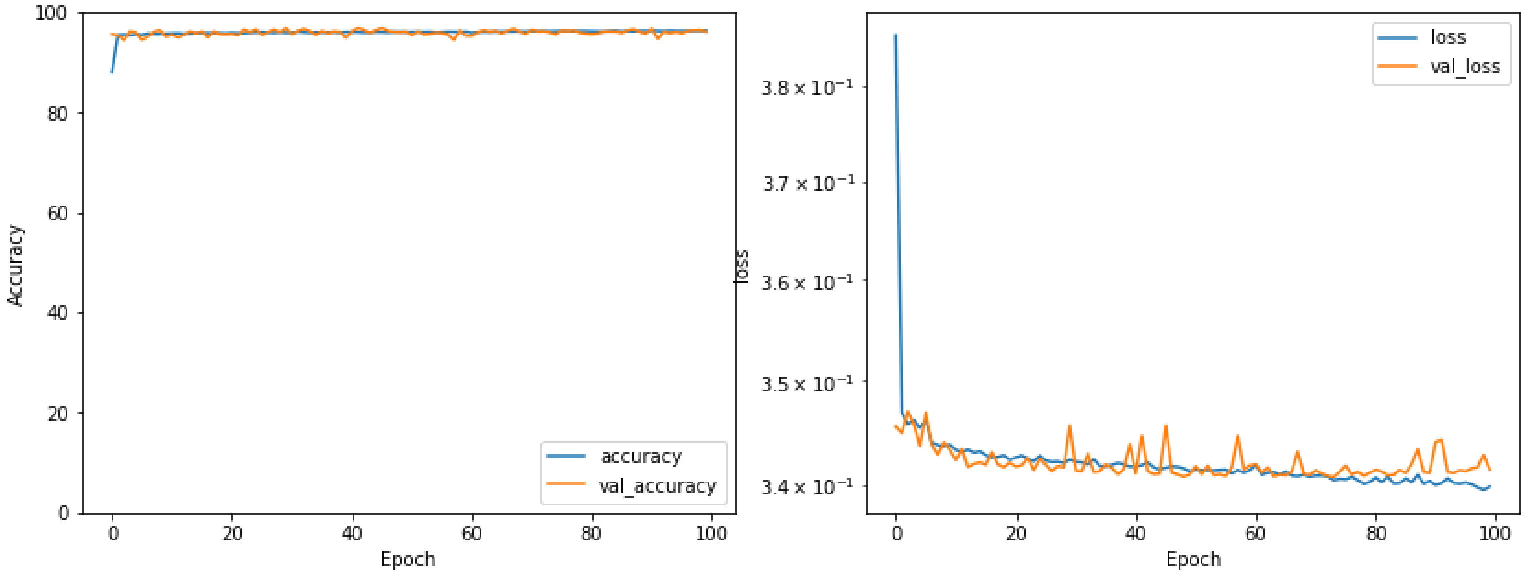
The authors acknowledge Saint-Petersburg State University for a research project 95413904

Backup slides – network training

The goal of network training is to iteratively change the network weights in order to minimize a metric (MSE for regression, cross entropy for classification). At each step, the network results are calculated, and the weights changes are determined using methods based on gradient descent.



Example of learning curve for the regression problem model. (blue – training set, orange – test set)



Example of learning curve for the binary classification problem model. Left – increasing accuracy. Right – decreasing cross entropy. (blue – training set, orange – test set)

Learning was stopped at the certain epoch, where there is no significant decrease in optimizing metric.
Early stopping of training is also a regularization method that helps deal with the problem of network overfitting.