Neural Network Approach to Neutrino and Extensive Air Shower Events Separation at Baikal-GVD

Matseiko Albert[1,2] , MIPT, INR RAS

This work was supported by the Russian Science Foundation grant no. 24-72-10056.

NPML 2025,
27 — 31 October 2025, Tokyo, Japan

Plan

1. **BaikalGVD experiment:**
   a) structure
   b) events
   c) data

2. **Neutrino selection** against the EAS background:
   a) Fast “pre-filter”
   b) Precise selection of events

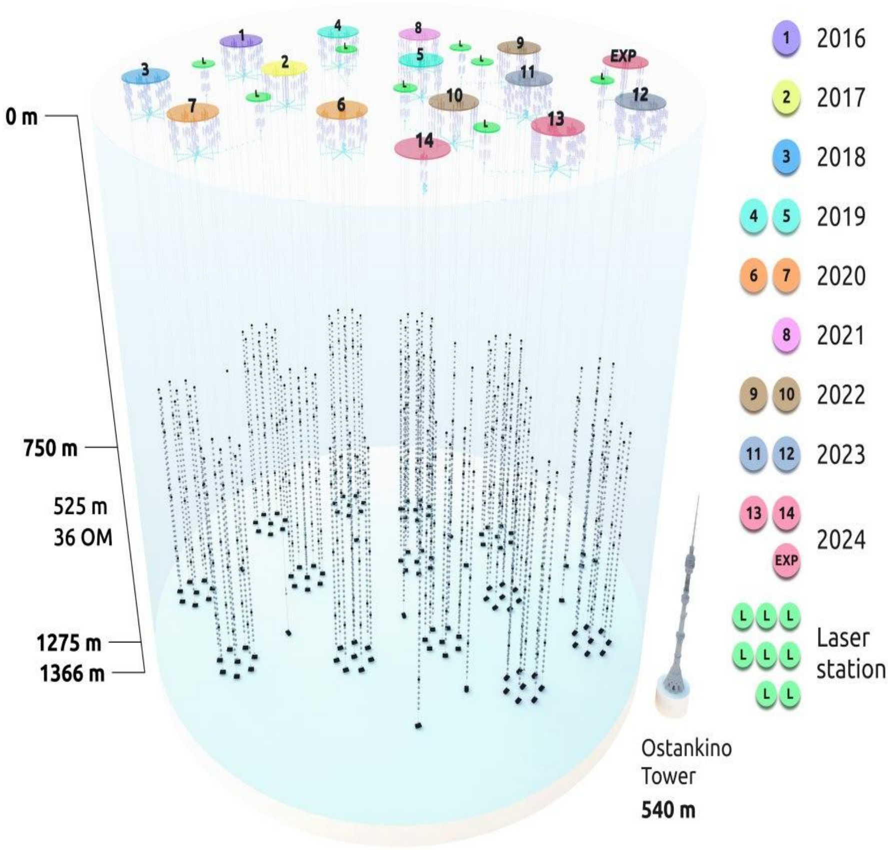
3. **Future plans**

I. Baikal-GVD

Baikal-GVD

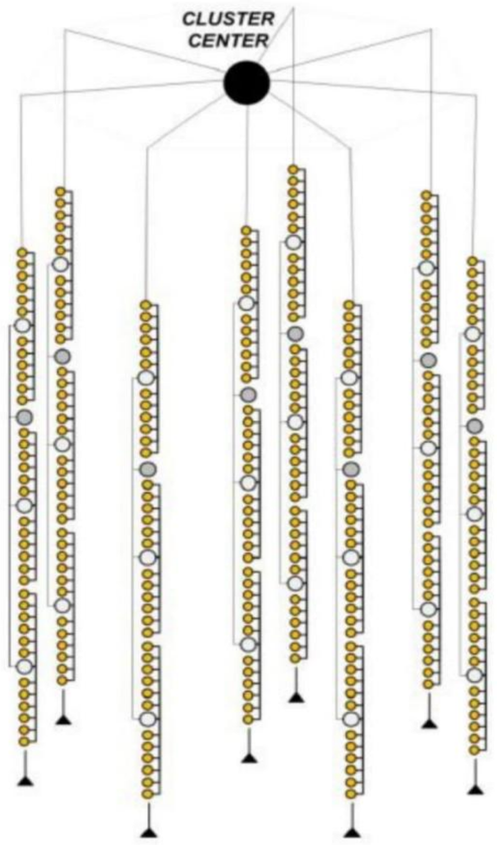
Purposes:
- To observe TeV-PeV neutrinos originating from the outside of the Solar System
- To investigate their sources

Cluster setup process

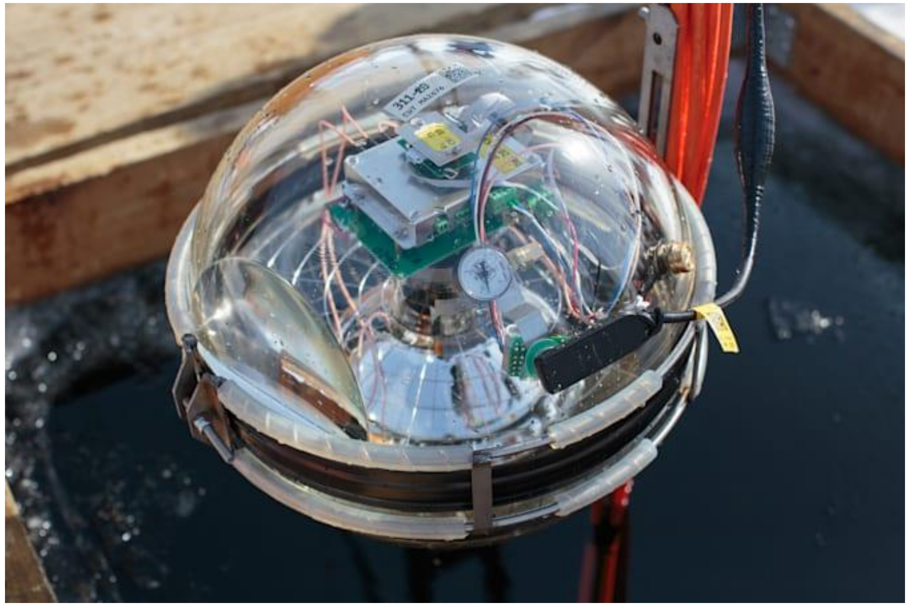


>14 clusters
are setup by now

Baikal-GVD



8 strings with 36 OM (optic modules)



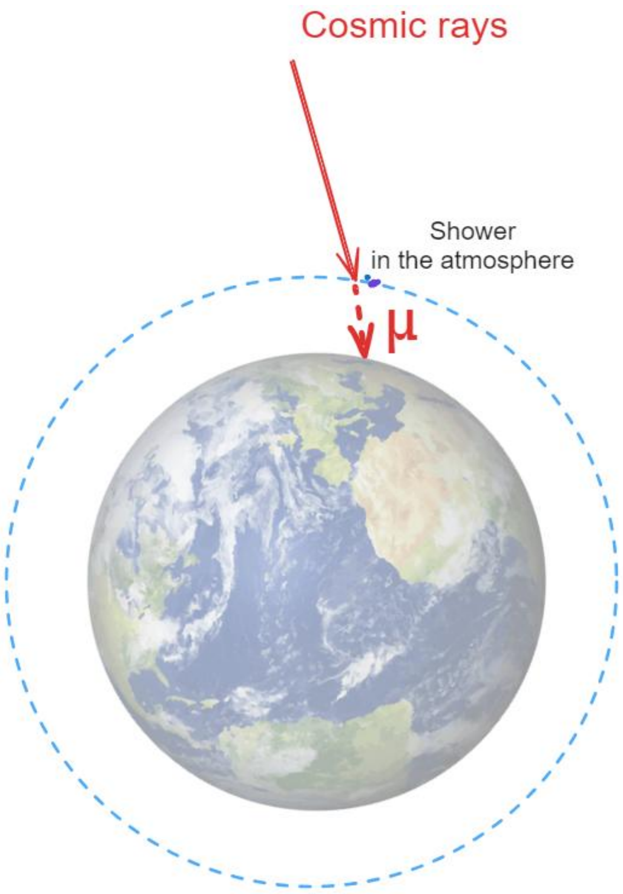
- Live calibration of OM positions (accuracy ~20 cm)
- Accuracy of determining the response time ~2 ns.

Events rate per cluster (of all origins): ~40-90 Hz

EAS and ν induced events

1) EAS

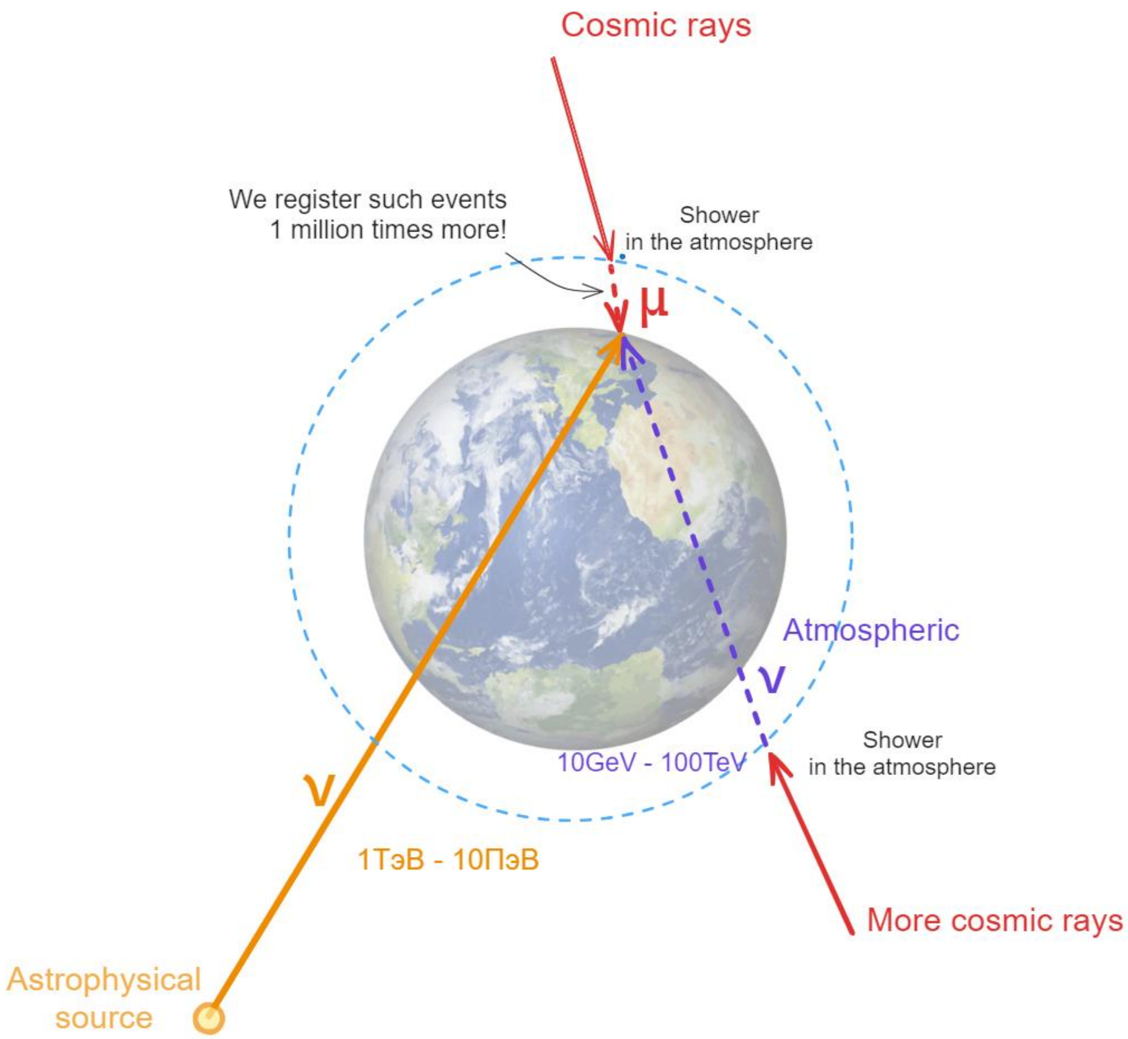
- showers from cosmic rays
- mostly muons reach the cluster
- «down-going» events



EAS and ν induced events

1) EAS
- showers from cosmic rays
- mostly muons reach the cluster
- «down-going» events

2) Neutrino
- astrophysical or atmospheric
- leptons are born in the cluster
- easily pass the Earth
  -> «up-going» events

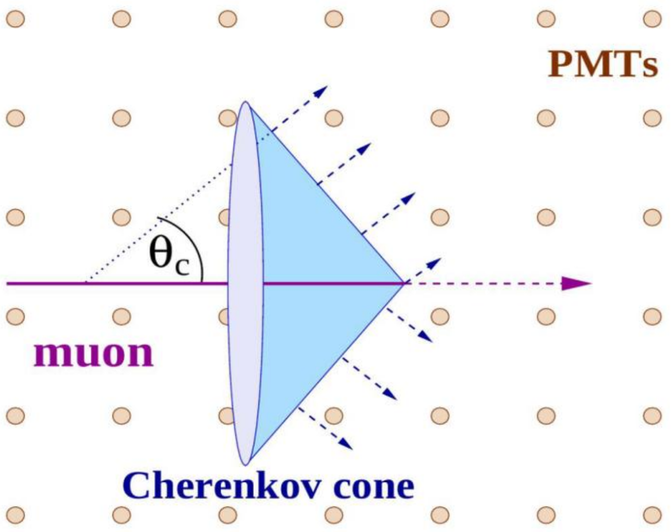


Baikal-GVD

Event options

ν_μ or atmosperic μ

Track events

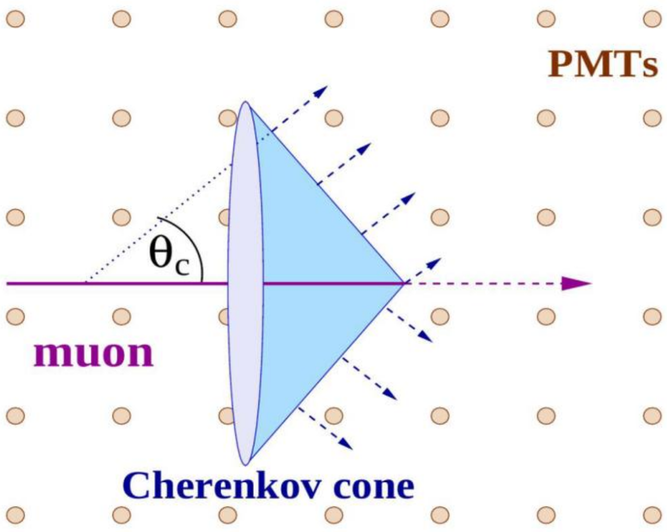


Baikal-GVD

Event options

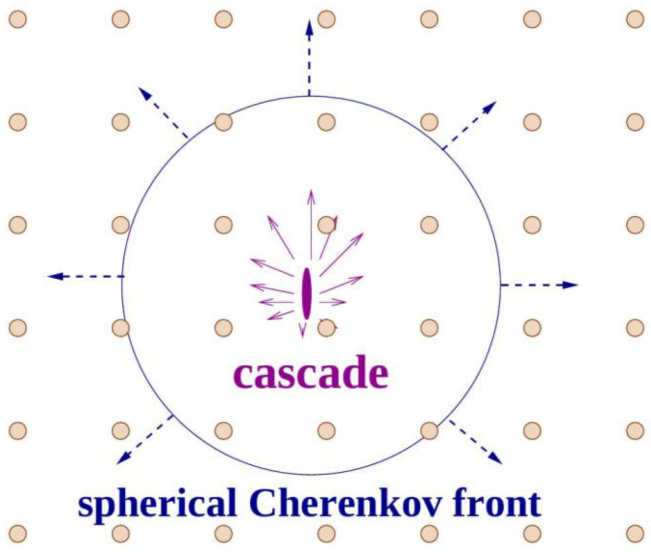
ν_μ or atmosperic μ

Track events



ν_e , ν_μ , ν_τ

Cascade events

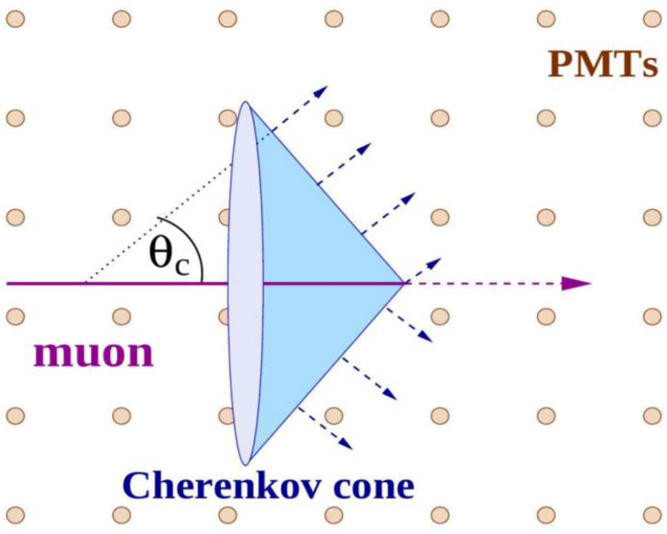


Baikal-GVD

Event options

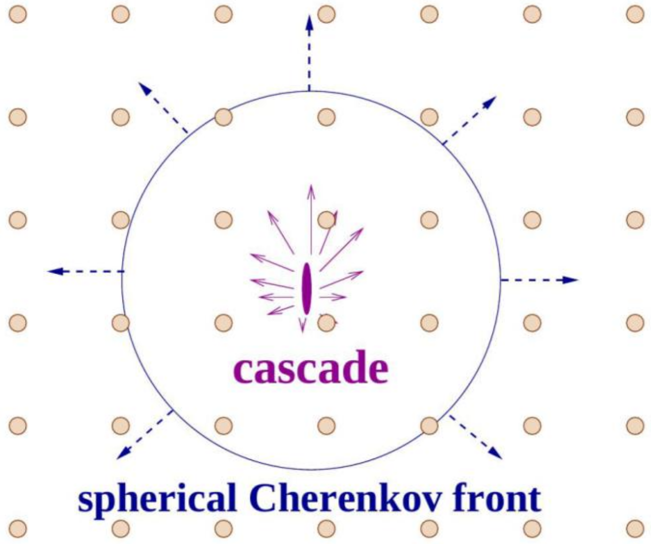
ν_μ or atmosperic μ

Track events



ν_e , ν_μ , ν_τ

Cascade events

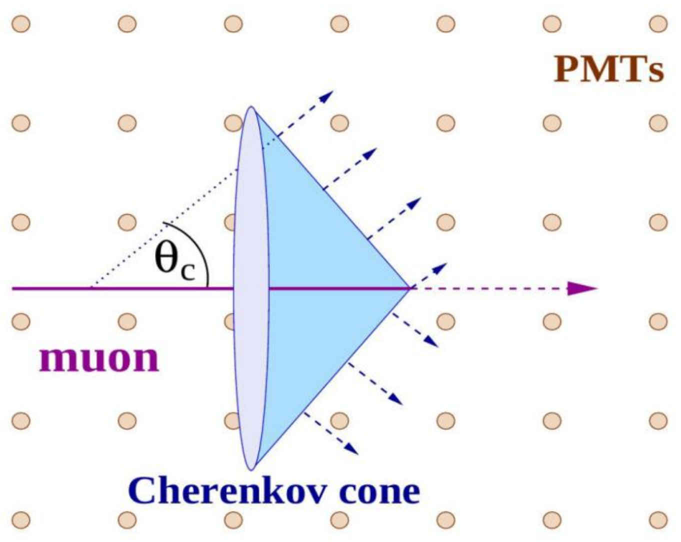


Baikal-GVD

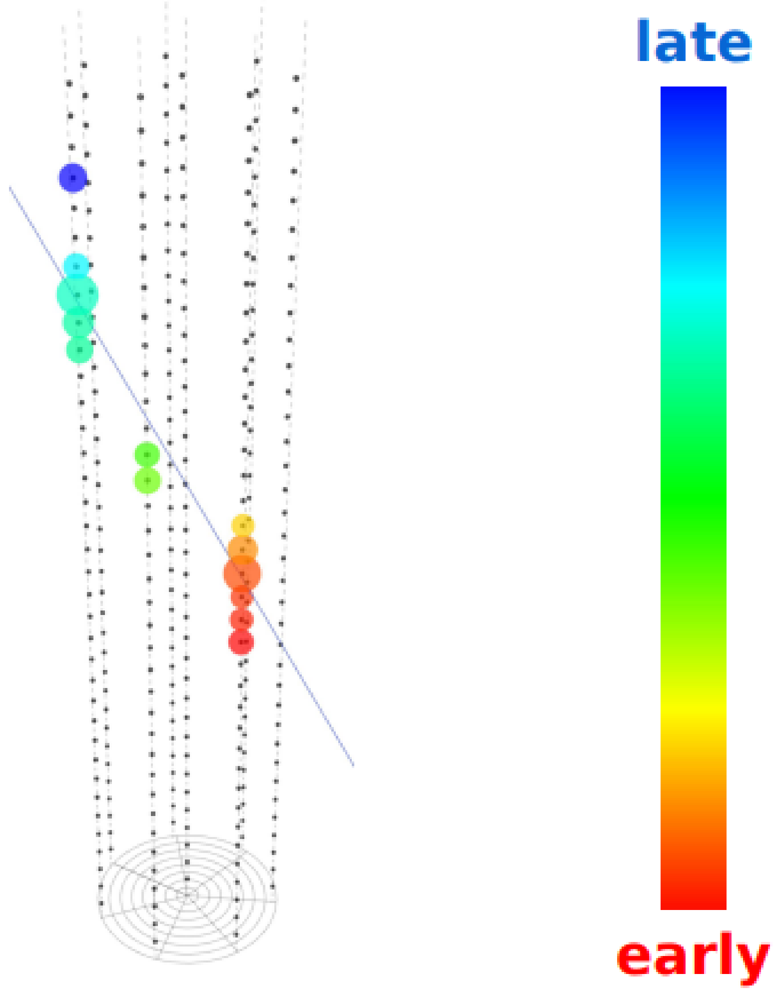
ν_μ or atmosperic μ

⇩

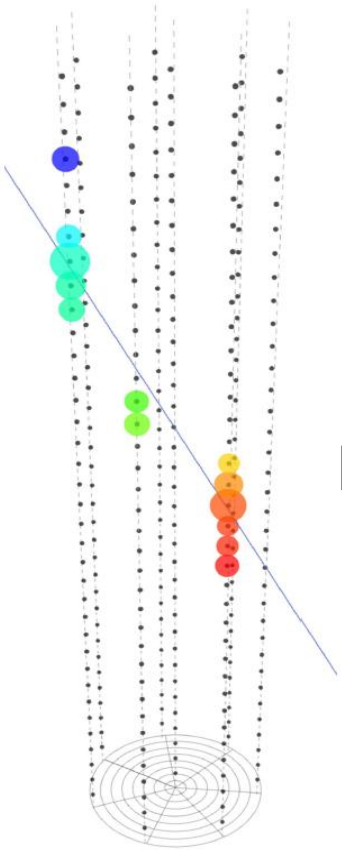
Track events



Track event picture

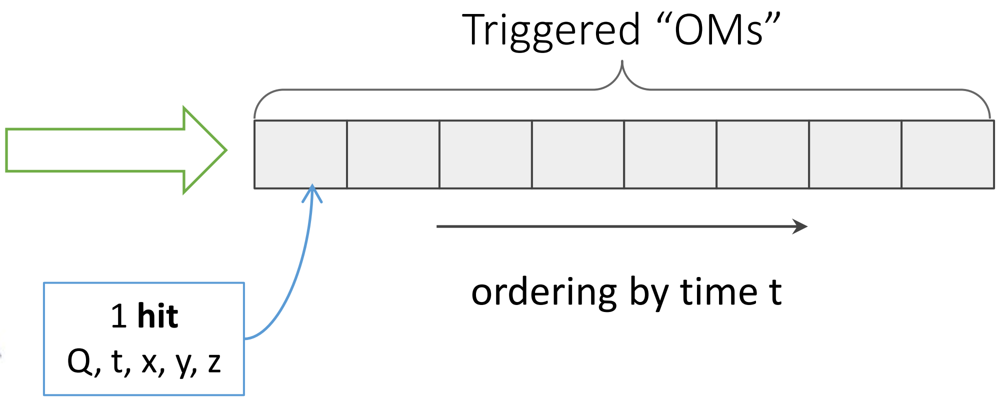


Data representation

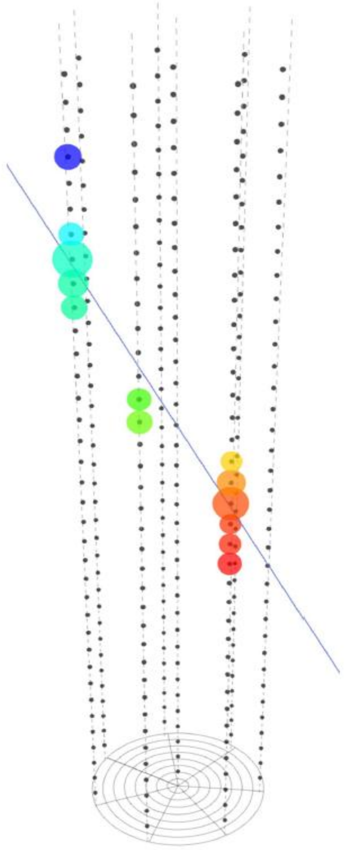


Single cluster

Causal (time-ordered)

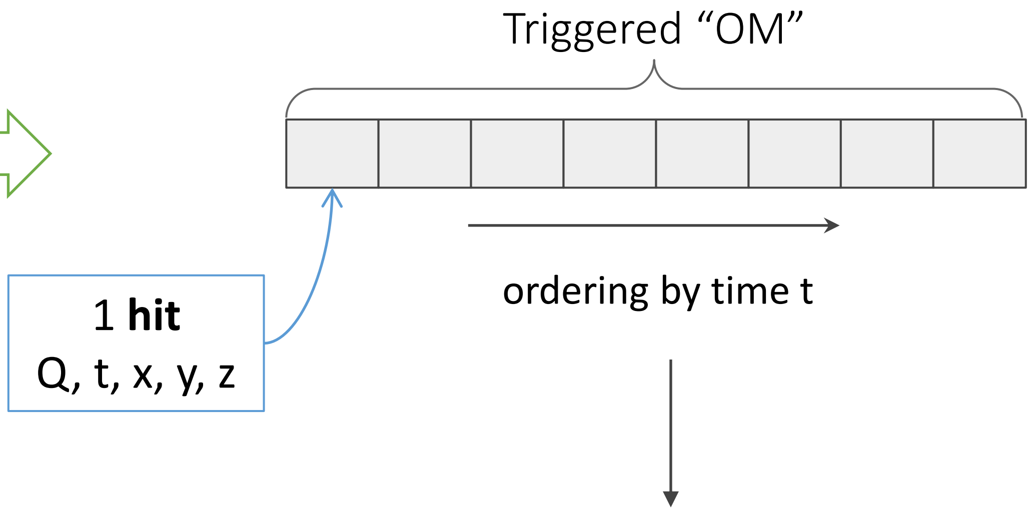


Data representation



Single cluster

Causal (time-ordered)

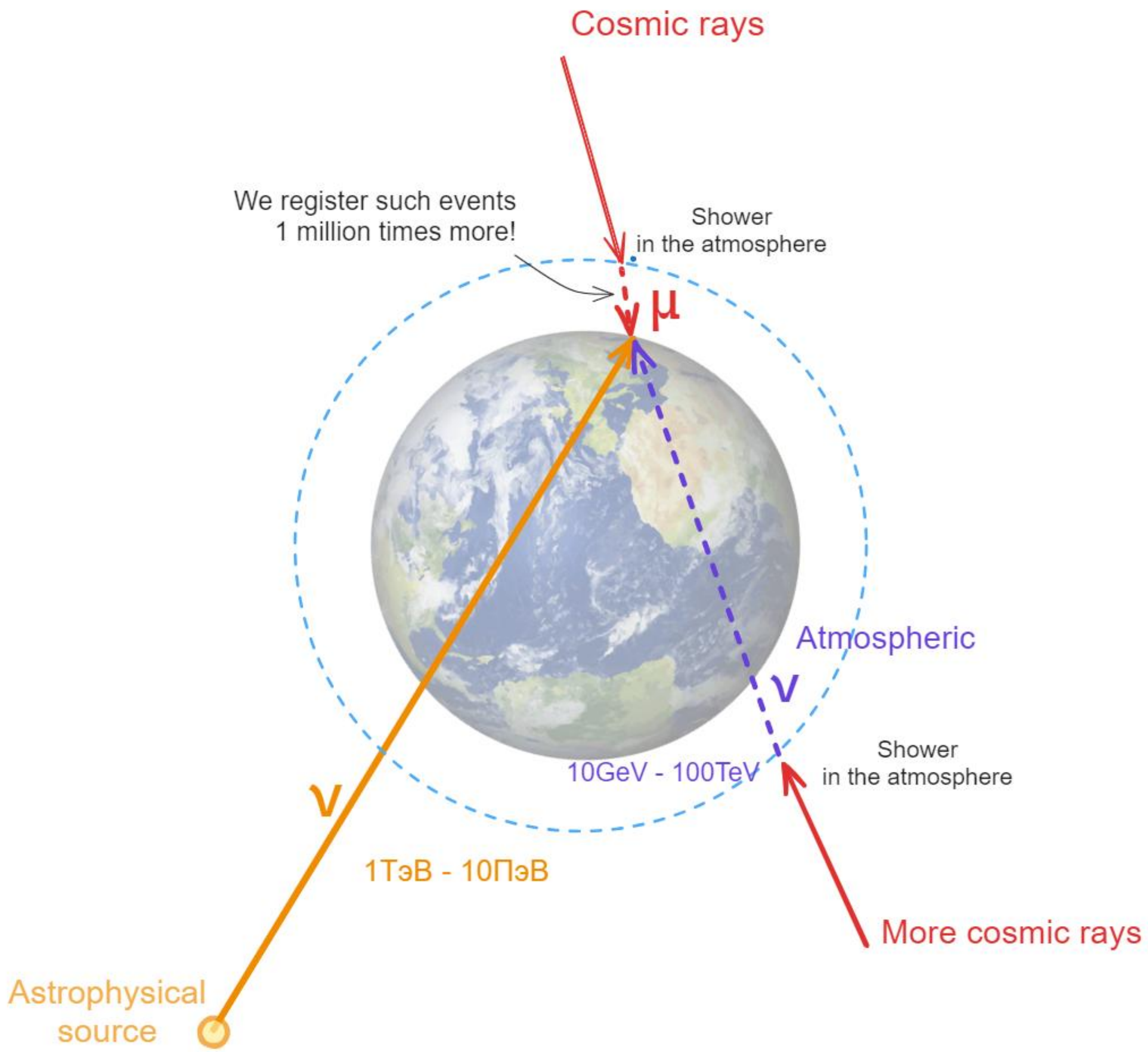


1D convolutions, recurrent networks, transformers

II. Neutrino selection against the EAS background

Motivation

- EAS to ν events ratio = 10^6-10^7.



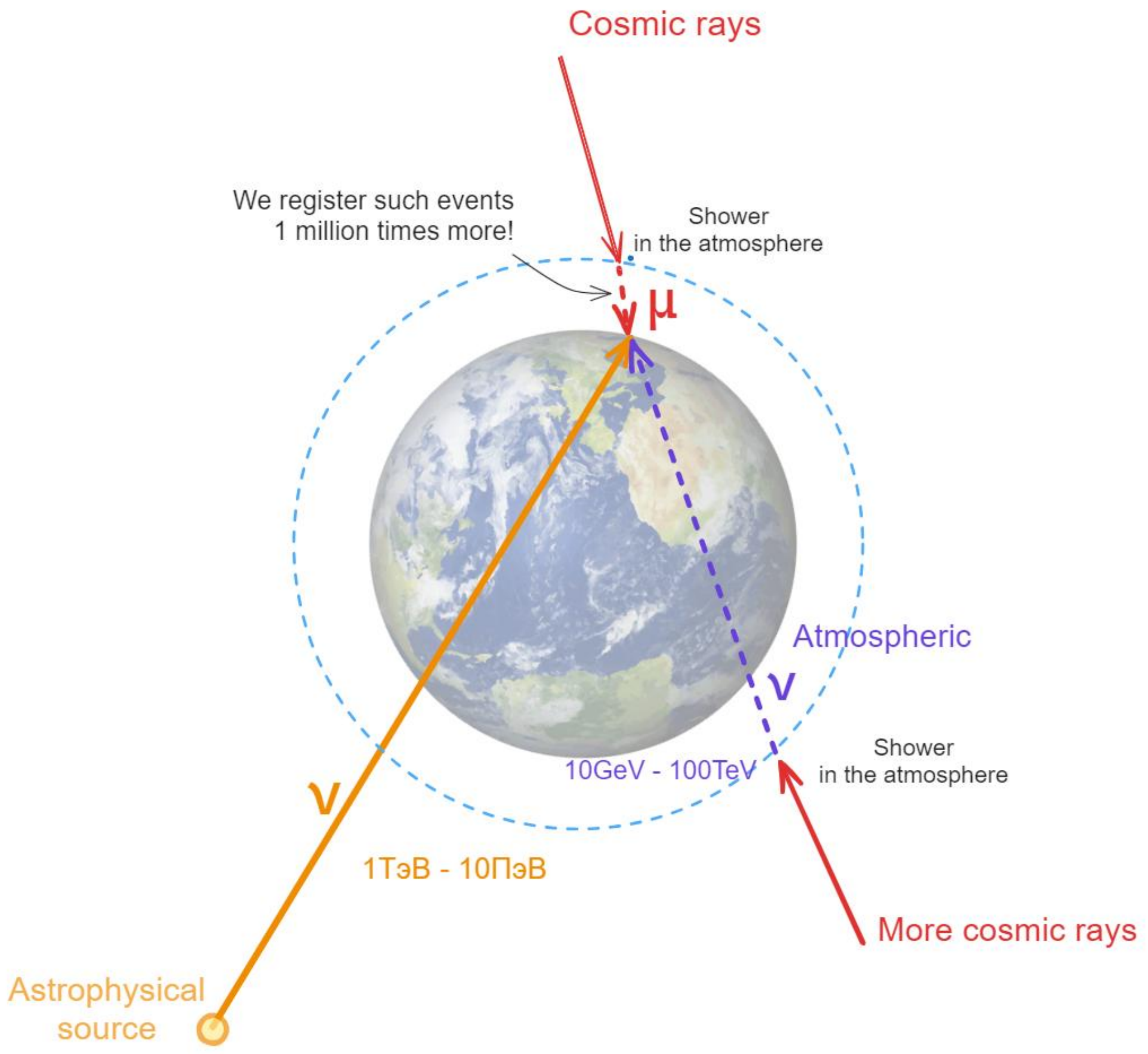
Motivation

- EAS to ν events ratio = 10^6-10^7.

- Standard approach:
  reconstruction of the zenith angle + cut.
  Computationally expensive, ~50% ν are lost.

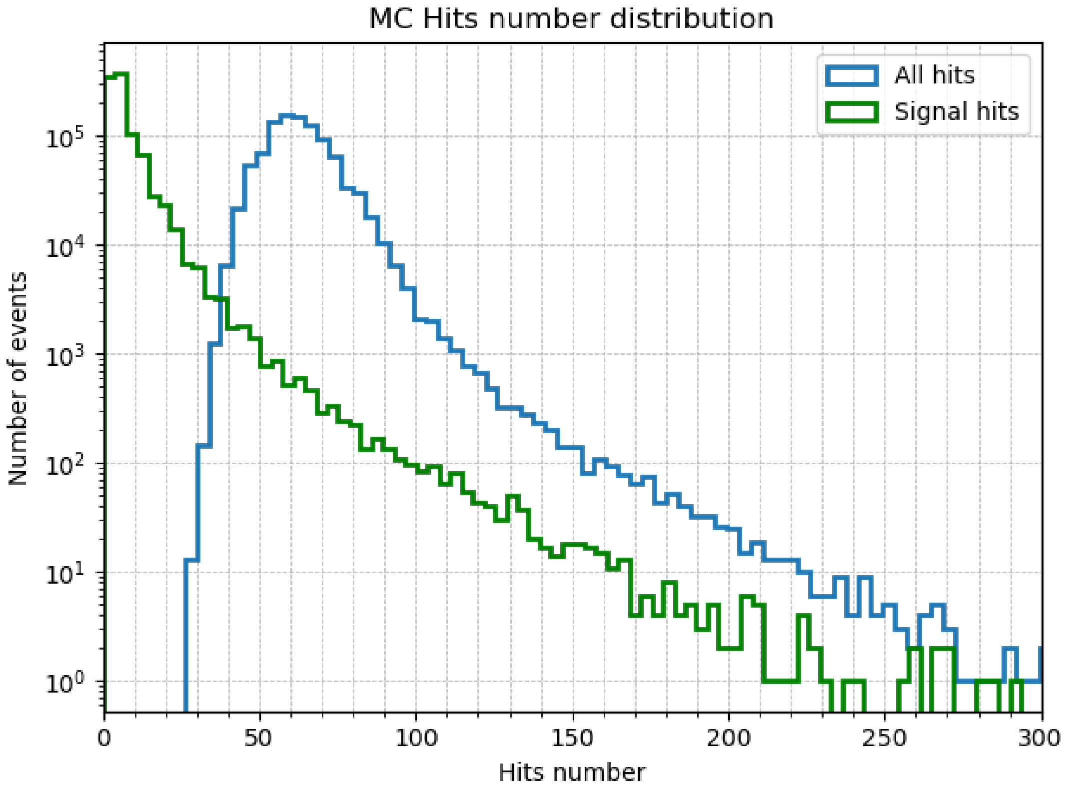
ML may help:

- filter out most of background cheap and fast:
  “pre-filter” model
- achieve better separation quality after some reconstruction:
  “precise” model



Fast pre-filter: dataset

- Using **Monte-Carlo simulation**[1]
  - EAS evolution and propagation of particles in water.
  - Random noise hits
  - Almost fixed geometry: no strings floating



- Track events only:
  1) Muons from EAS
  2) **ν_μ** (neutrinos of muon flavour)

- **Target feature** — type of particle
  Labels: 0 — EAS, 1 — neutrino

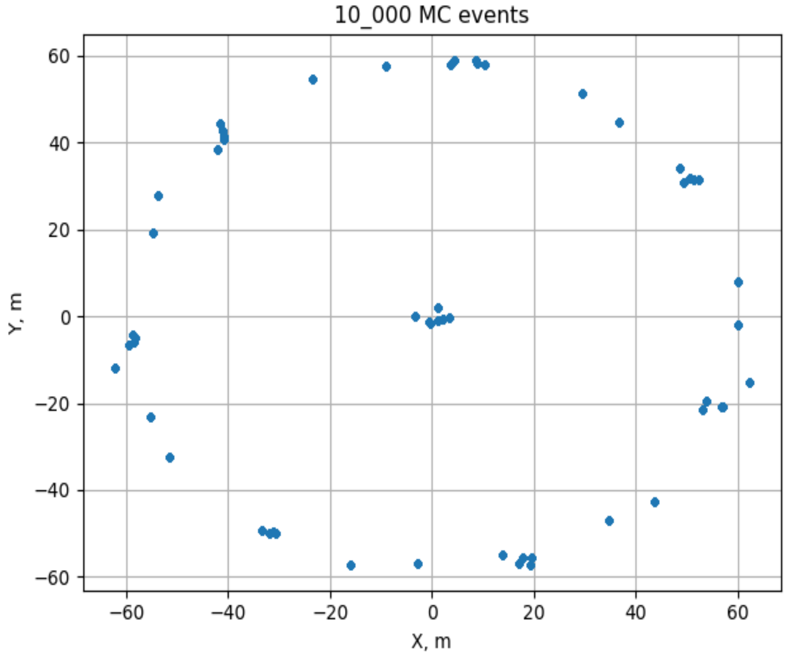
	Train	Validation	Test
EAS	$2.4*10^5$	$6*10^4$	$\approx 5*10^5$
Neutrino	$2.4*10^5$	$6*10^4$	$\approx 5*10^5$

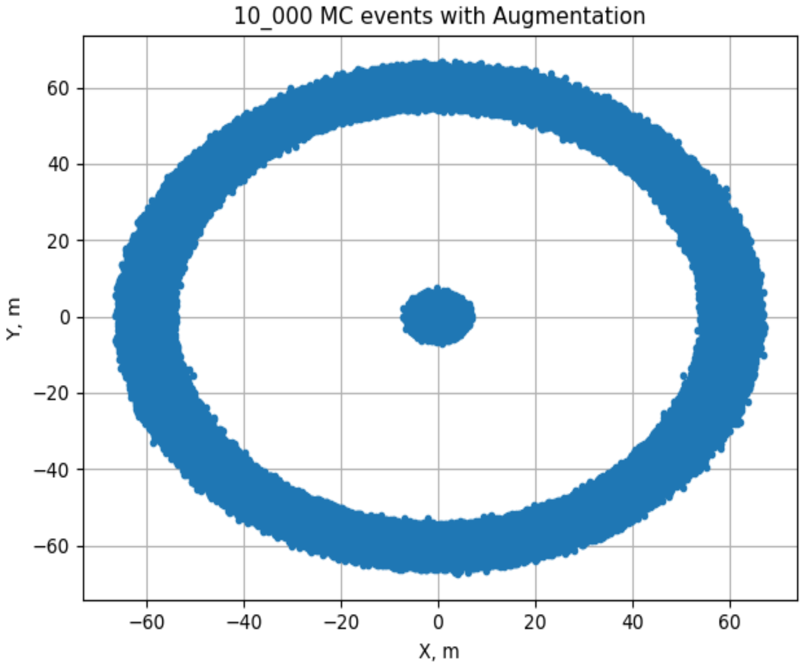
[1] arxiv.org/abs/2106.06288

Fast pre-filter: dataset

Data augmentations:

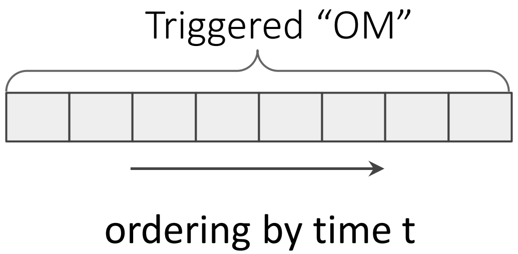
- Random gaussian noise to all inputs. STDs:
  - 1 m for **x** and **y**
  - 2m for **z**
  - 5 ns for **t**
  - 0.05 pe for **Q**
- Random rotations in x-y plane

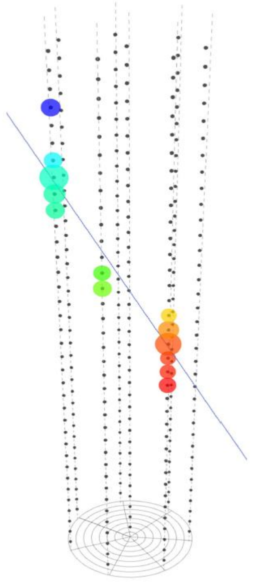




[1] arxiv.org/abs/2106.06288

Fast pre-filter: The network





Linear Projection + PE + [CLS] token
5 -> 128 embedding dim

↓

Transformer Encoder Layer
dim_model = 128,
heads = 4,
Dropout=0.1, *LayerNorm* applied

↓

MLP classifier
Hidden layers: [64, 32],
Dropout=0.1, BatchNorm applied

↓

Prediction p_ν

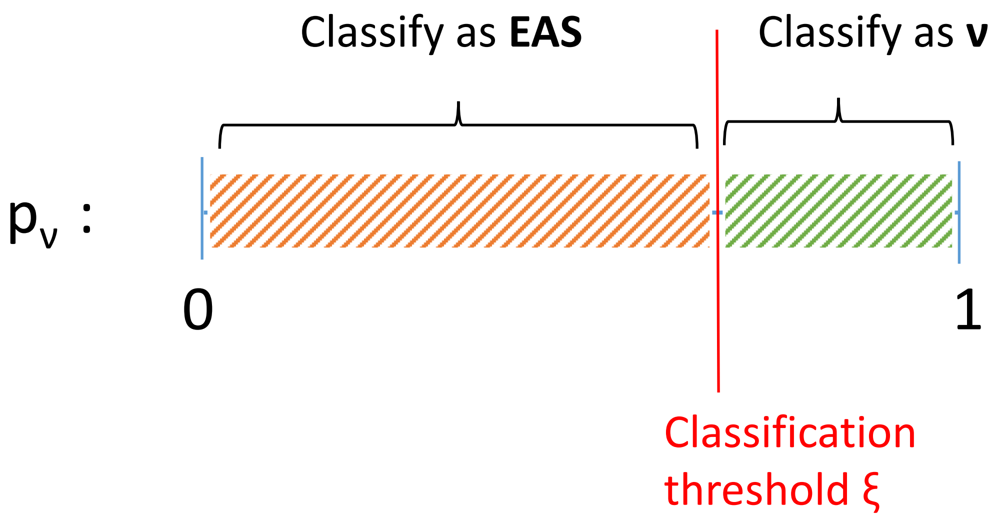
Training details:

- AdamW optimizer
- BCE loss
- LR=2e-3; exp decay with 0.01 weight
- Early stopping by validation ROC AUC, patience 15 epochs

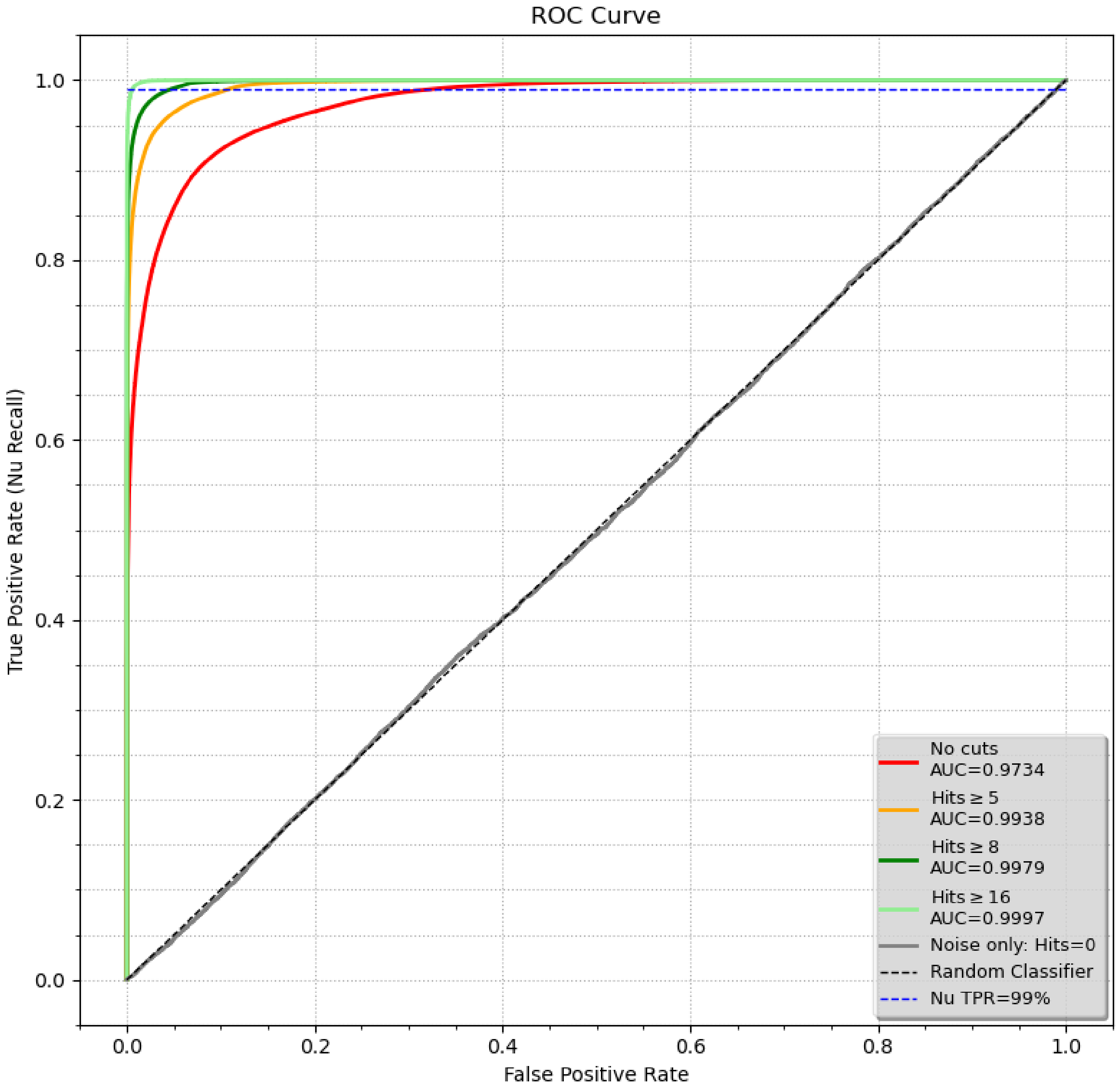
Pre-filter: Metrics

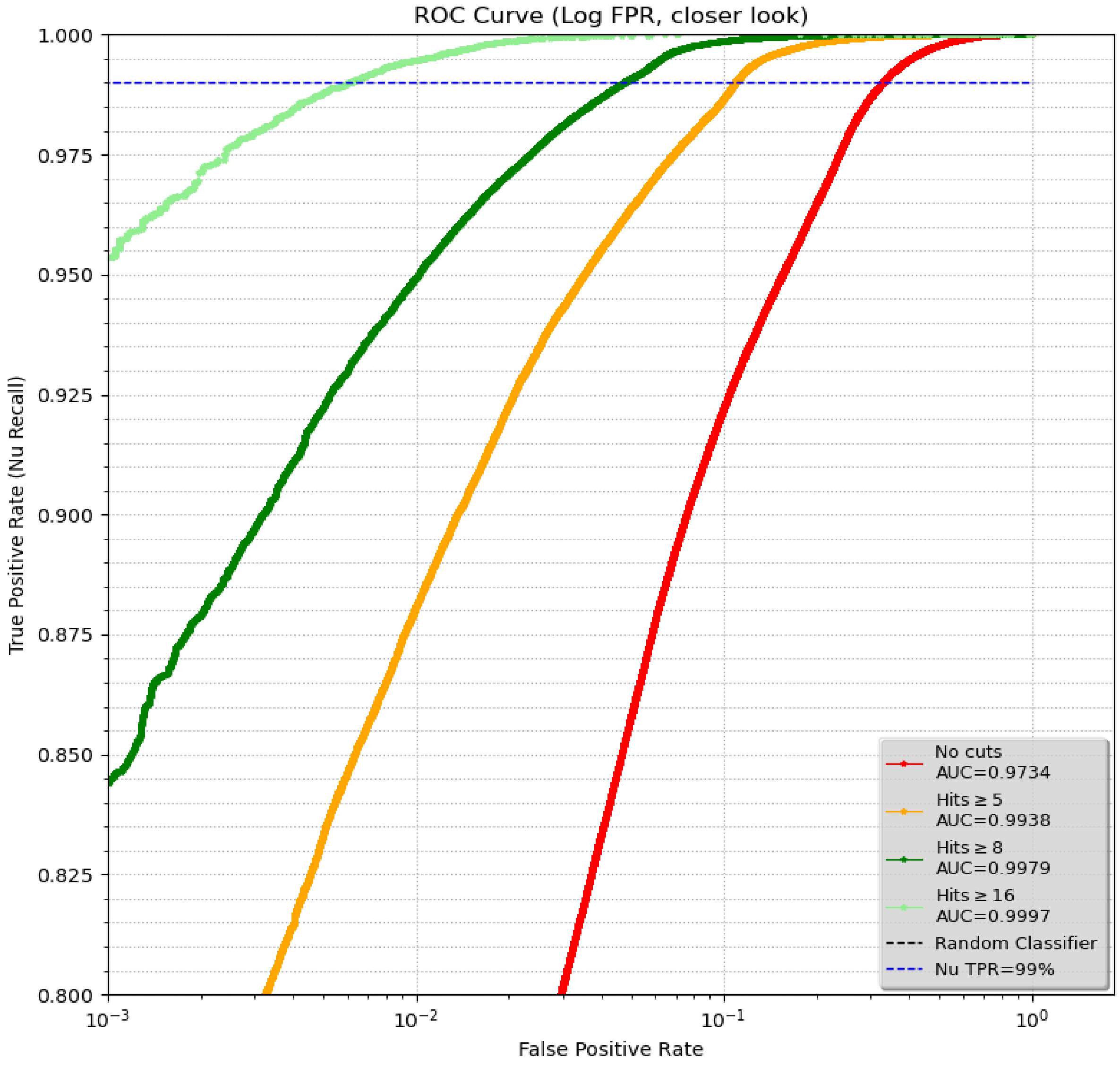
- TPR = n_ν/N_ν
  fraction of ν identified correctly
  (True Positive Rate)

- FPR = n_{EAS}/N_{EAS}
  fraction of EASs falsely assigned to ν
  (False Positive Rate)



Pre-filter: Metrics





Precise selection: dataset

- Same MC

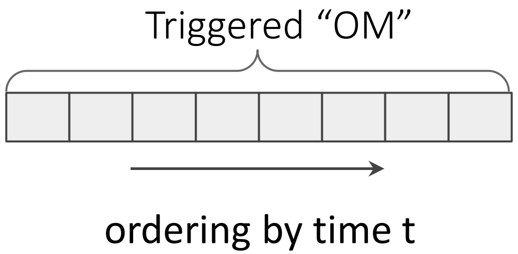
	train	test	validation
EAS	$\approx 3*10^6$	$\approx 5*10^5$	$\approx 2*10^6$
Neutrino	$\approx 6*10^5$	$\approx 10^5$	$\approx 1.3*10^7$

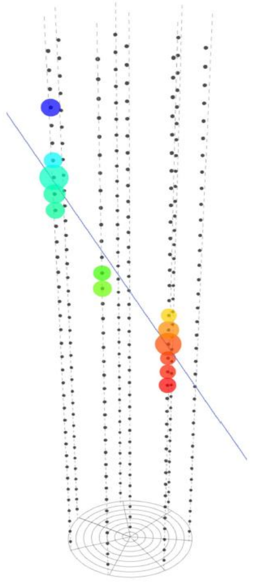
- **Cuts applied** (assuming reconstruction made)**!**
  - min 8 signal hits
  - min 2 strings triggered

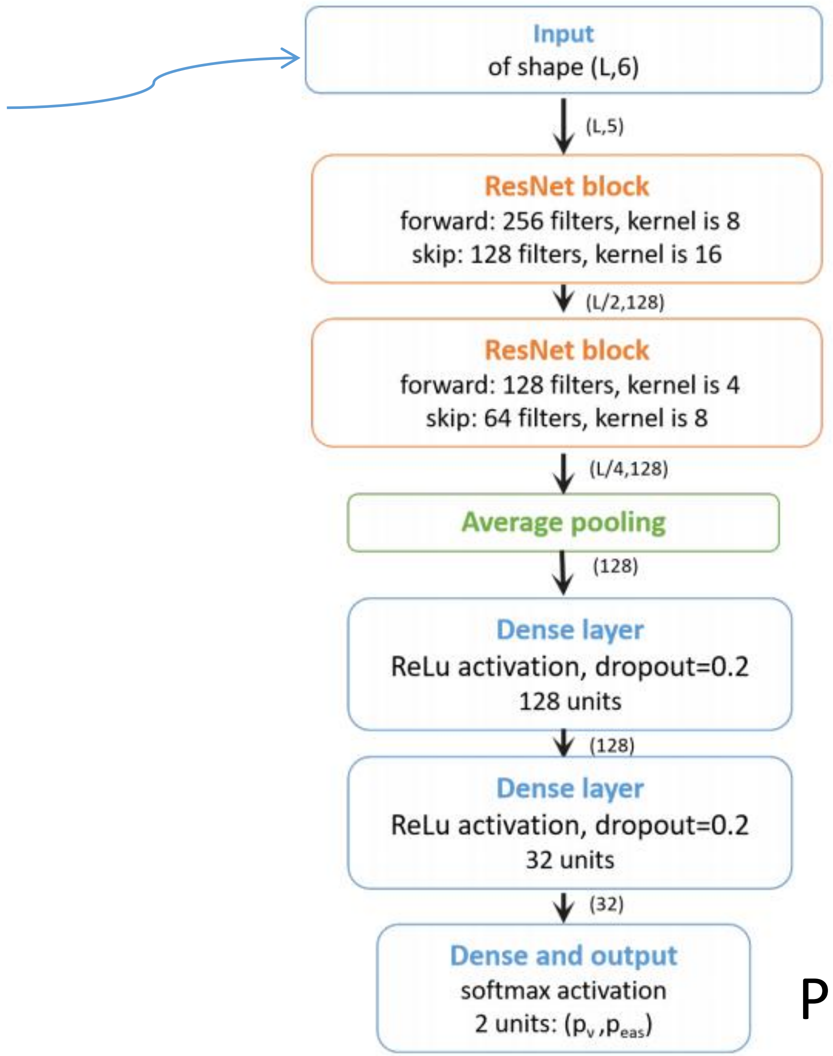
- Target feature — type of particle
  Labels: 0 — EAS, 1 — neutrino

[1] arxiv.org/abs/2106.06288

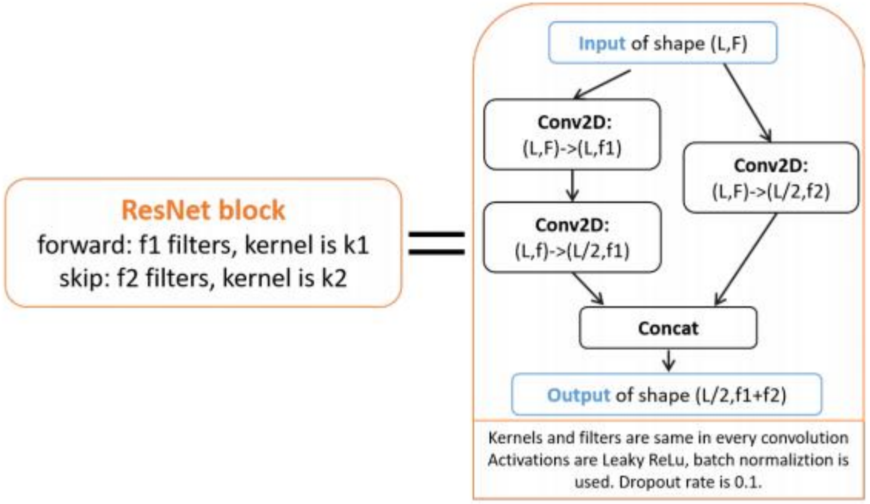
Precise selection: The network







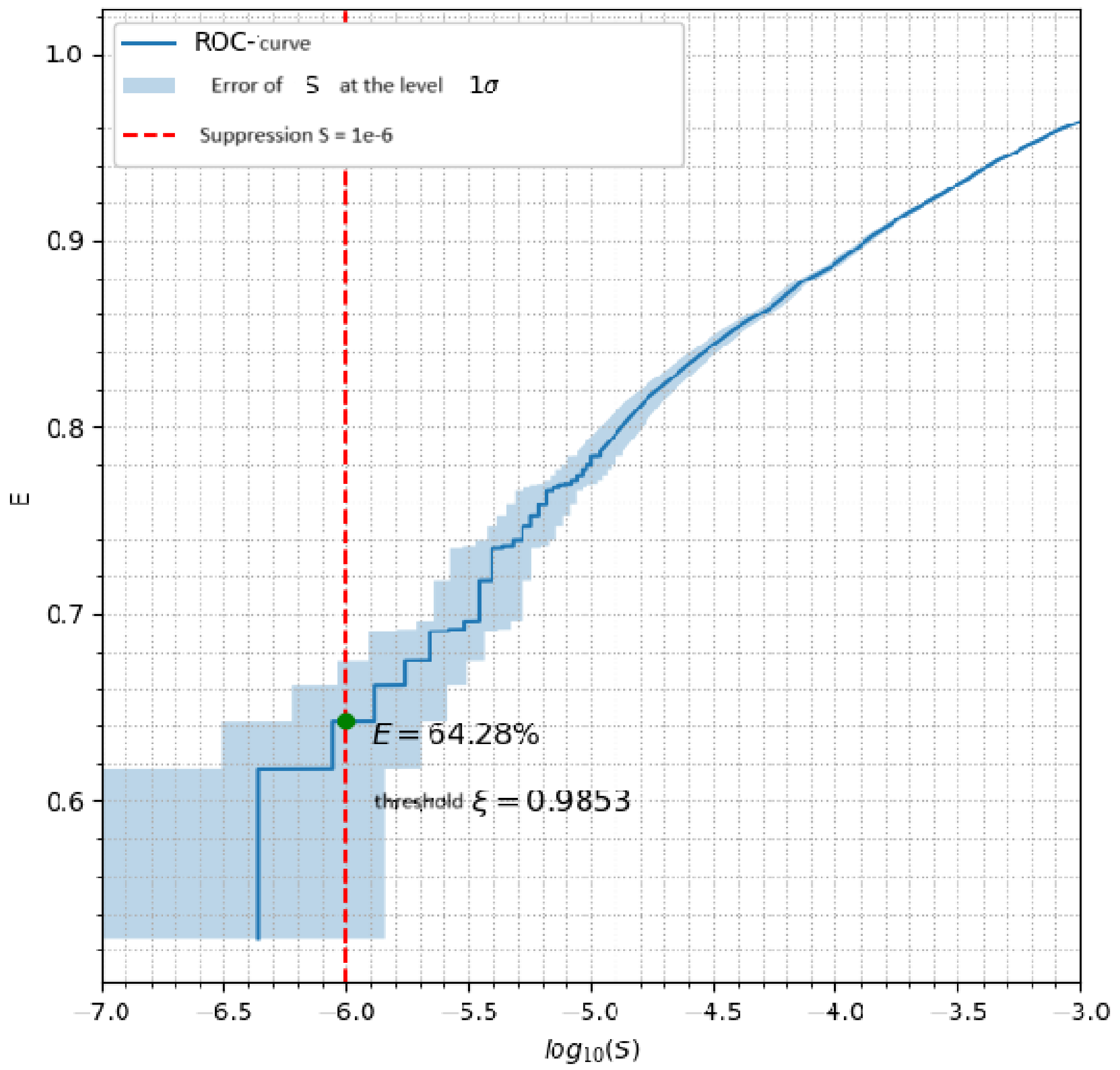
Prediction p_ν



Precise selection: Metrics

- @60% of neutrino events preserved, while suppressing background by 10^6 times!
- Results are rough, since @2e6 EAS-events were in test
- Results are preliminary:
  - no rotations in XY plane
  - ResNet should be replaced with Transformer

ROC curve (log TPR)



Precise selection: Estimation Flux Algorithm

Based on

- measured TPR and FPR, and
- knowing total number of events in the dataset N_{total},

the true number N_{ν} of ν-events can be estimated.

Precise selection: Estimation Flux Algorithm

Based on

- measured TPR and FPR, and
- knowing total number of events in the dataset N_{total},

the true number N_ν of ν-events can be estimated.

Idea:

1) Treat the NN as a *black box*, performing a binomial process:
   - If the event is ν, the NN assigns it to ν class with probability p_ν.
   - If the event is EAS, the NN assigns it to ν class with probability p_{EAS}
2) Then, expected number of events classified as ν is:

$$N_{pred} = p_\nu N_\nu + p_{EAS}(N_{total} - N_\nu),$$

from which N_ν can be derived.

Precise selection: Estimation Flux Algorithm

Based on
- measured TPR and FPR, and
- knowing total number of events in the dataset N_{total},

the true number N_ν of ν-events can be estimated.

Idea:

1) Treat the NN as a *black box*, performing a binomial process:
   - If the event is ν, the NN assigns it to ν class with probability p_ν.
   - If the event is EAS, the NN assigns it to ν class with probability p_{EAS}

2) Then, expected number of events classified as ν is:

$$N_{pred} = p_\nu N_\nu + p_{EAS}(N_{total} - N_\nu),$$

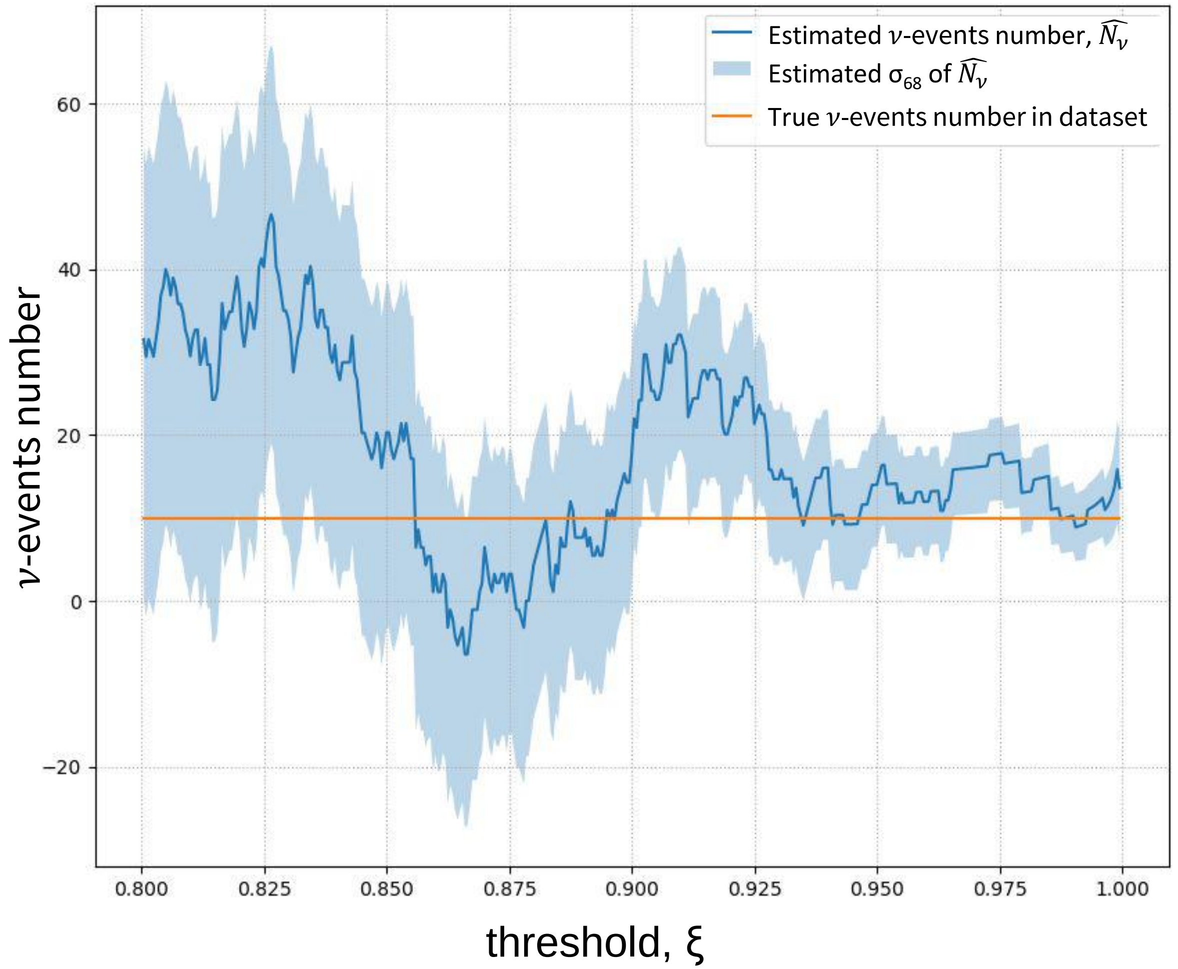
from which N_ν can be derived.

We have estimations for p_ν and p_{EAS}: TPR and FPR!
Also, we know N_{pred} and N_{total}!
Then, the following estimation is correct:

$$\widehat{N_\nu} \approx \frac{N_{pred} - \boldsymbol{FPR} \cdot N_{total}}{\boldsymbol{TPR} - \boldsymbol{FPR}}$$

Then, we can calculate its statistical uncertainty σ_{68} !

Preliminary results for
10 ν-events against 100_000 EAS background



III.
Future Challenges

1. Adaptation to **experimental data domain**

The problem:

- MC deviates from reality
  → Hard to rely on NN predictions

1. Adaptation to **experimental data domain**

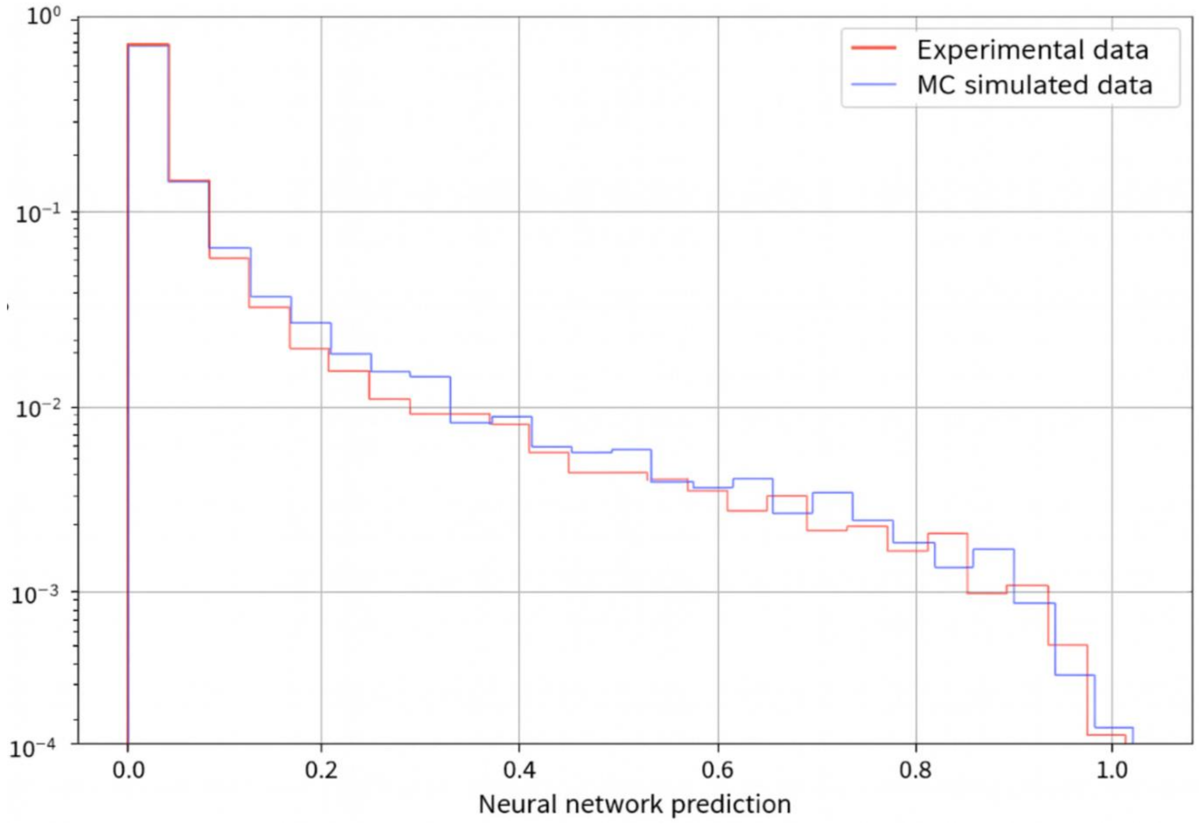
The problem:

- MC deviates from reality
  → Hard to rely on NN predictions

Solutions:

- To validate models using well-reconstructed events
- **To compare distributions** of model scores in MC and Experimental domains

Distributions for **"pre-filter"** model prediction:
MC VS Experimental data.
Signal Cuts (experiment events have been reconstructed): hits≥8, strings ≥2.



1. Adaptation to **experimental data domain**

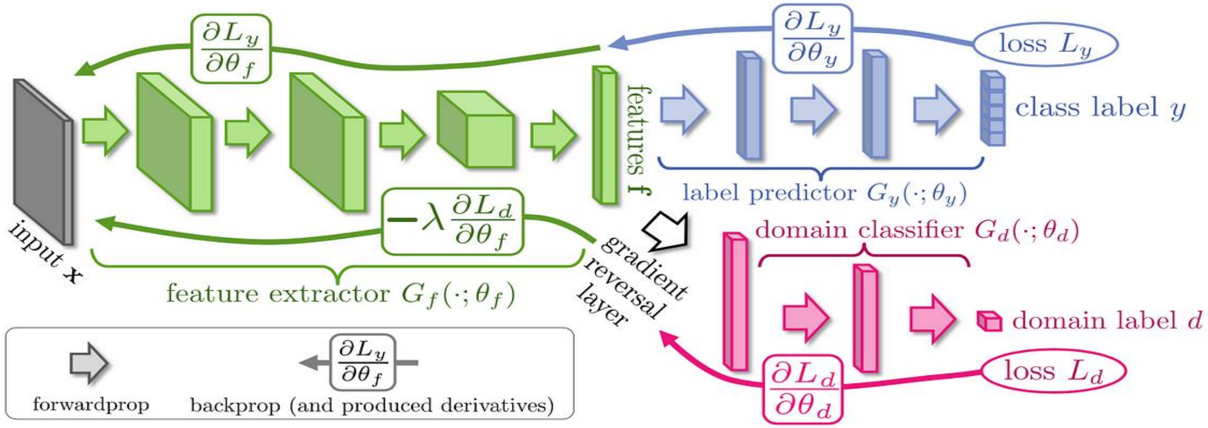
The problem:

- MC deviates from reality
  → Hard to rely on NN predictions

Solutions:

- To validate models using well-reconstructed events
- To compare distributions of model scores in MC and Experimental domains
- **Use Domain Adaptation (DA) techniques**

DA scheme:
domain-adversarial training with a gradient reversal layer



2. Solving real tasks

“pre-filter” model:

- Integrate the model into the Baikal collaboration’s data-analysis framework (BARS)
- Significantly reduce the time required for subsequent event reconstruction

“precise” model:

- Produce catalogues of neutrino events with a quantified contamination rate from EAS events
- Estimate the total neutrino flux

IV. Conclusions

1) Neural networks helps (evaluated on MC simulation):

- Suppress the EAS background from **5** (≥5 signal hits events) up to **@100** (≥16 hits) **times** using raw signals

- Preserve **60%** of nu-events while suppressing **EAS background by** @**10^6 times** using reconstructed signal hits

2) Much of work with real-experiment data to do.

Thanks!

Contacts:

matseiko.av@phystech.su
t.me/AlbertMac280

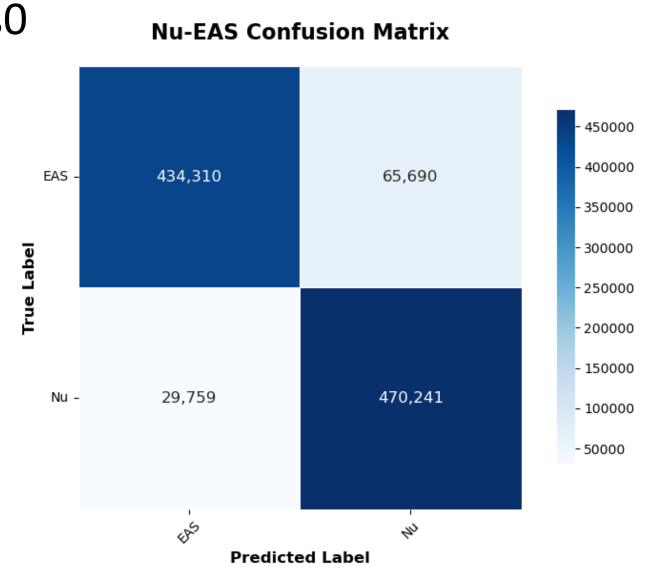
Code and models:

github.com/ml-inr/Baikal-ML

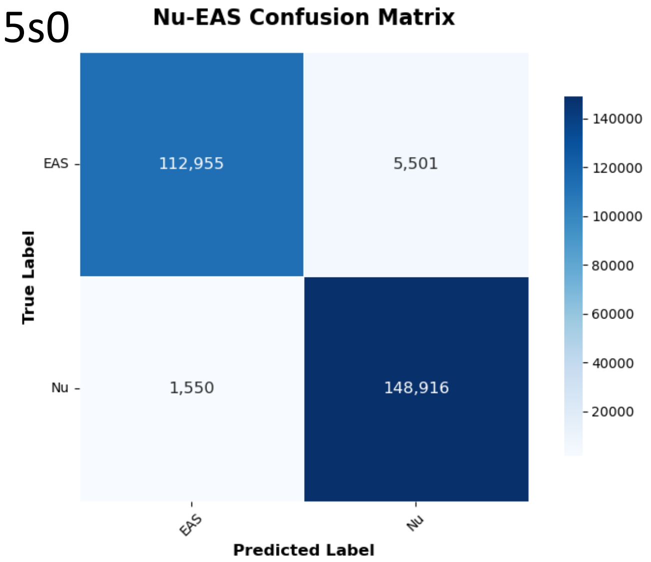
Backup

More “pre-filter” graphs

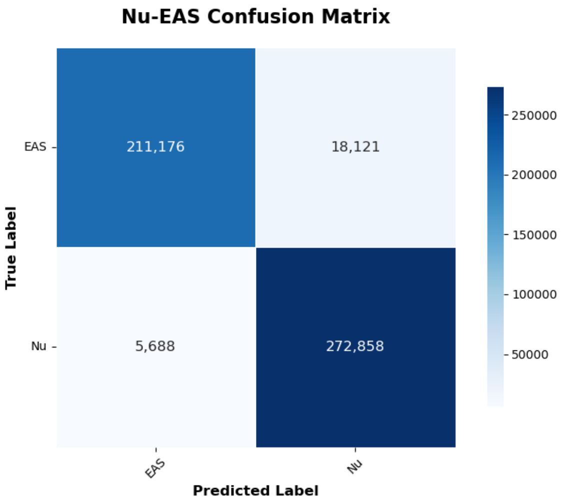
h0s0



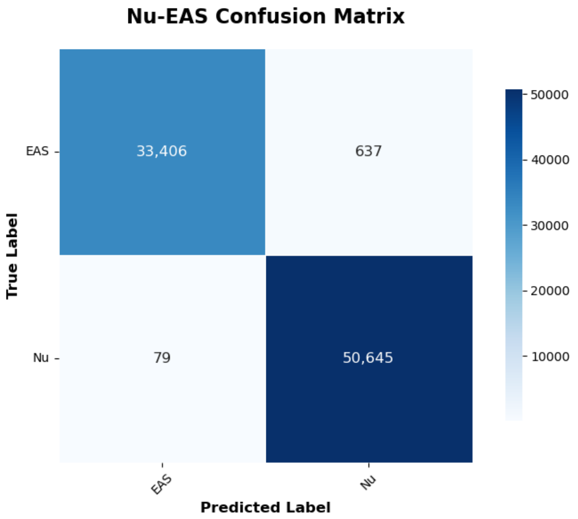
h5s0



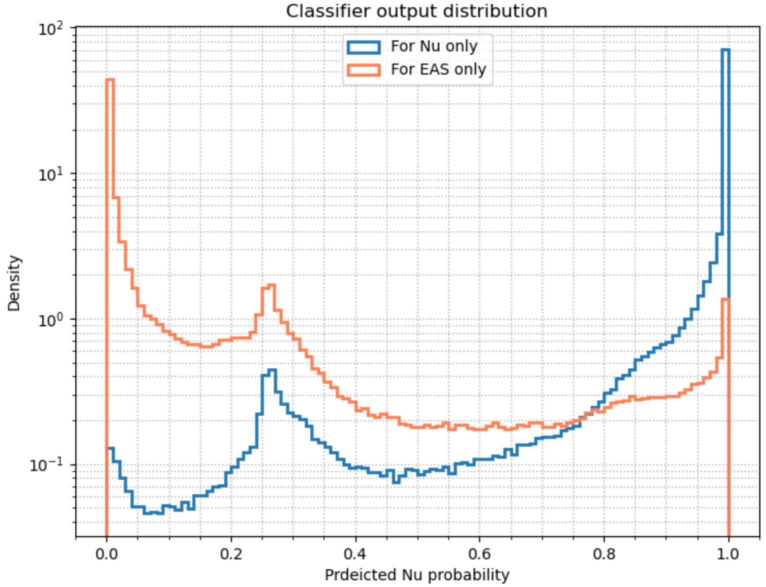
h8s0



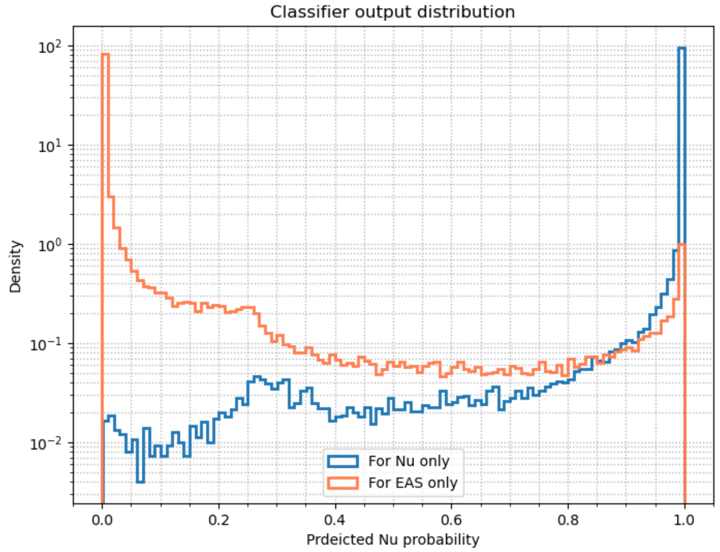
h16s0



More “pre-filter” graphs

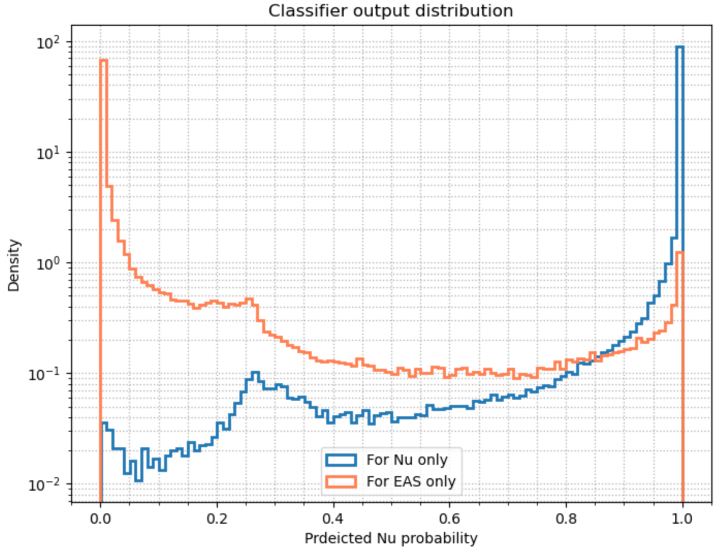


h0s0

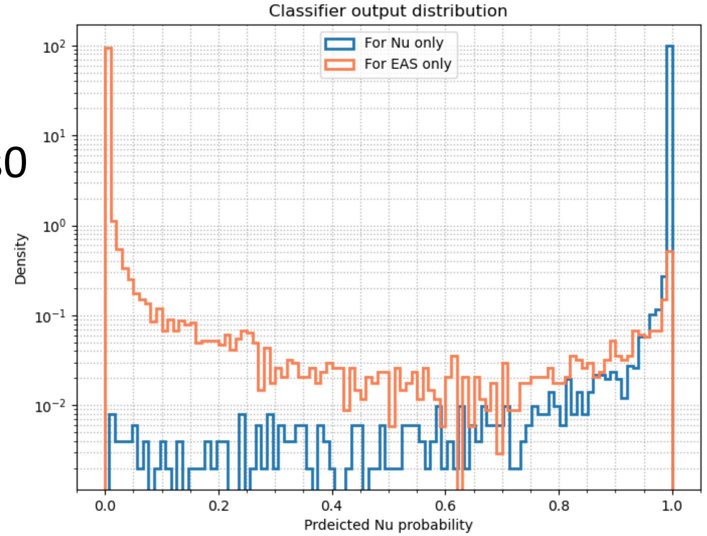


h8s0

h5s0



h16s0



Общий план применения нейронных сетей

Считываемые данные

↓

Подавление шумов

↓

Выделение нейтринных событий

↓

Реконструкция параметров событий
(энергия, направление)

Общий план применения нейронных сетей

Считываемые данные

↓

Подавление шумов

↓ → стандартная реконструкция

Выделение нейтринных событий

↓ → оценка потока нейтрино

Реконструкция параметров событий
(энергия, направление)

Общий план применения нейронных сетей

Считываемые данные

↓

Подавление шумов

↓ → стандартная реконструкция

Выделение нейтринных событий

↓ → оценка потока нейтрино

Предварительная реконструкция ⊕ → Реконструкция параметров событий (энергия, направление)

Технические данные

$\lambda^{eff}_{scatterina} \approx 480\text{м при } 475\text{ nm}$ $\lambda^{max}_{absorption} \approx 24\text{м}$

Трековые события :точность угла прилета ≈ 0, 25°
Каскадные события: разрешение ≈ 2°

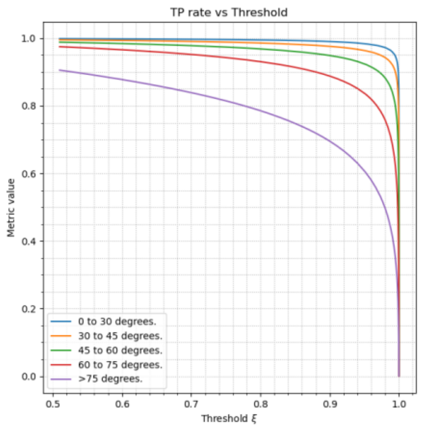
Монте - Карло:

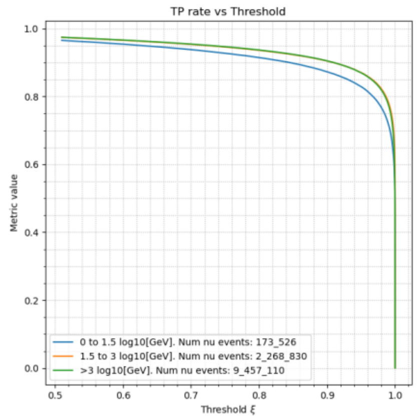
Взаимодействие нейтрино с ядрами : CTEQ4M
(нейтрино с энергиями 10 ГэВ – 100ТэВ
Прилет мюонов: программа CORSIKA 5.7 на модели адронных вз-ий QGSJET
Распространение мюонов до Байкала : MUM v1.3u
Космические лучи: модель на базе KASCADE (240 ГэВ – 20 ПэВ)
Ошибка по времени 5 нс ; 30% по заряду

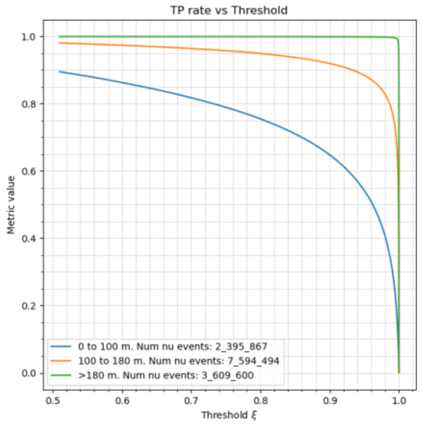
EAS:Nu = 1:1
train: 5556146 events,
test: 465253 events,
val: 22345821 events

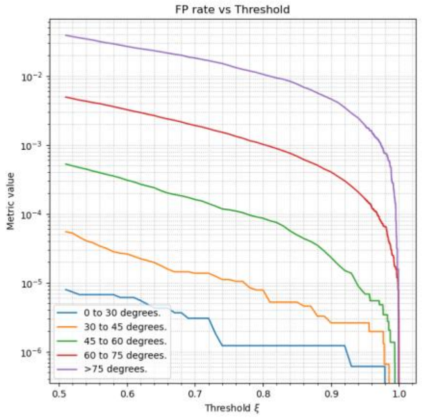
Разные графики

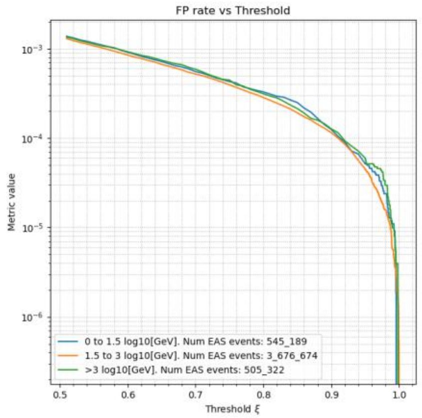
Больше разделения

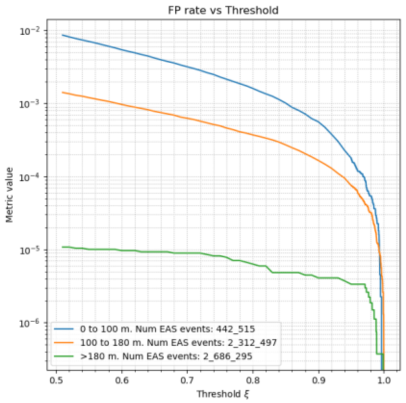


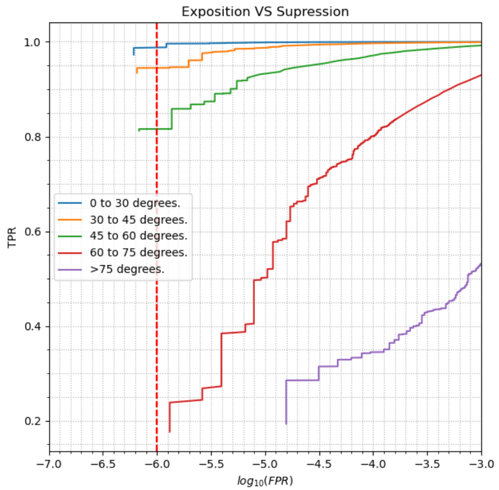


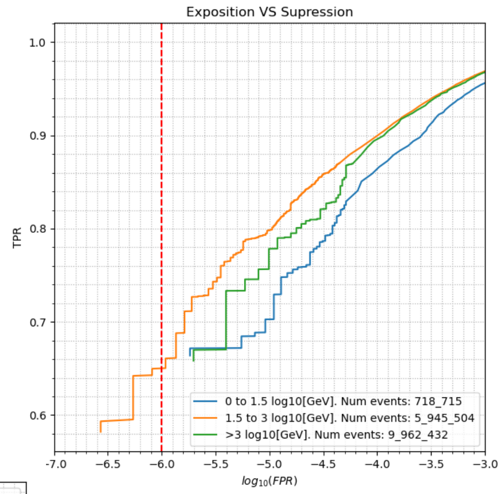


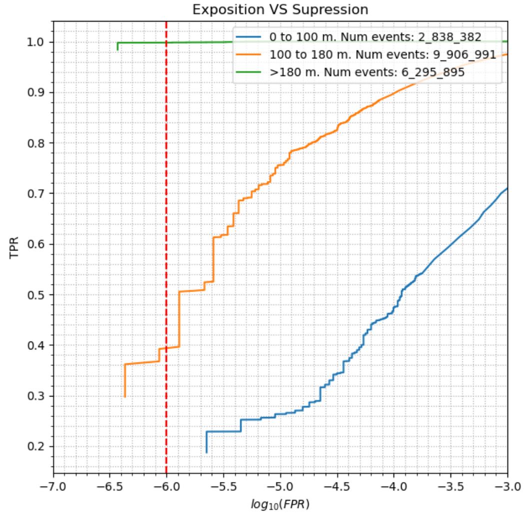






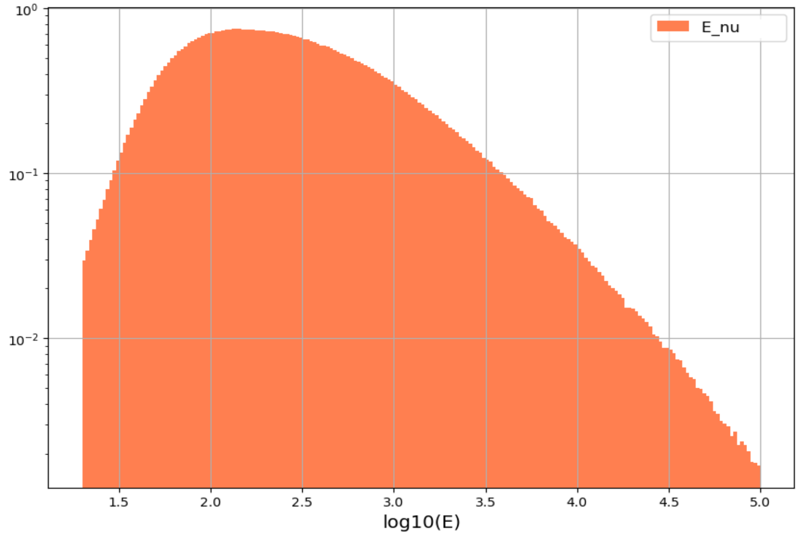


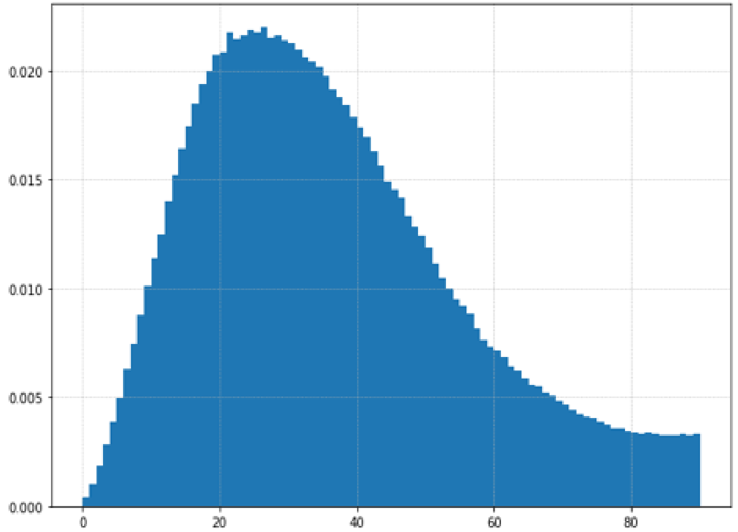




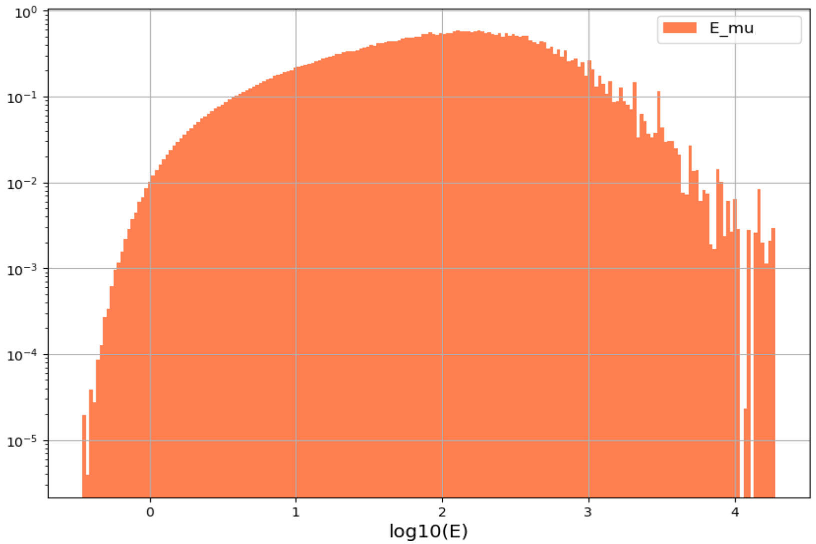
Спектры

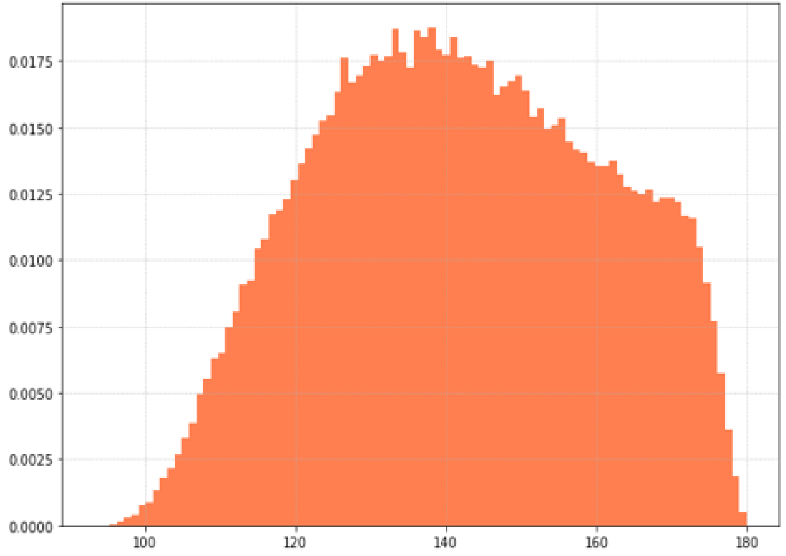
Спектры ν





Спектры μ

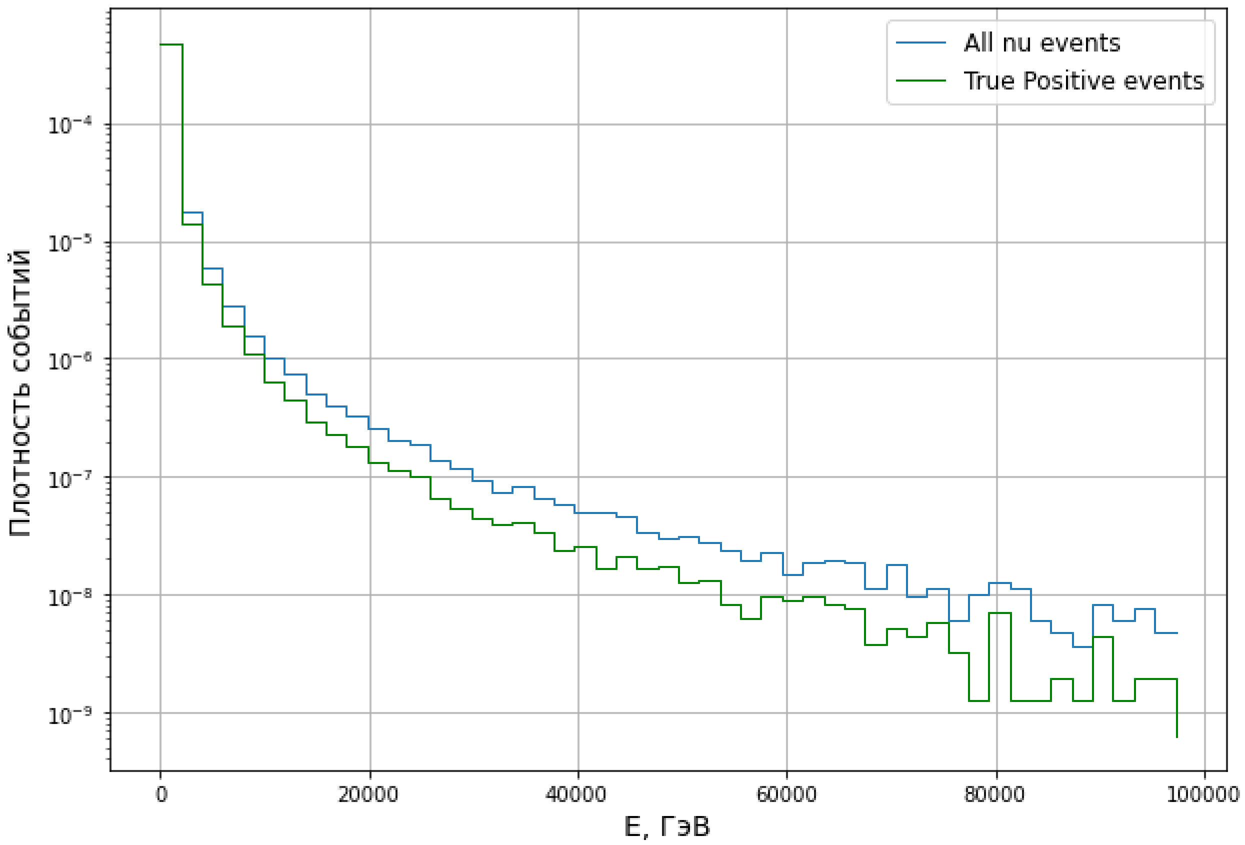




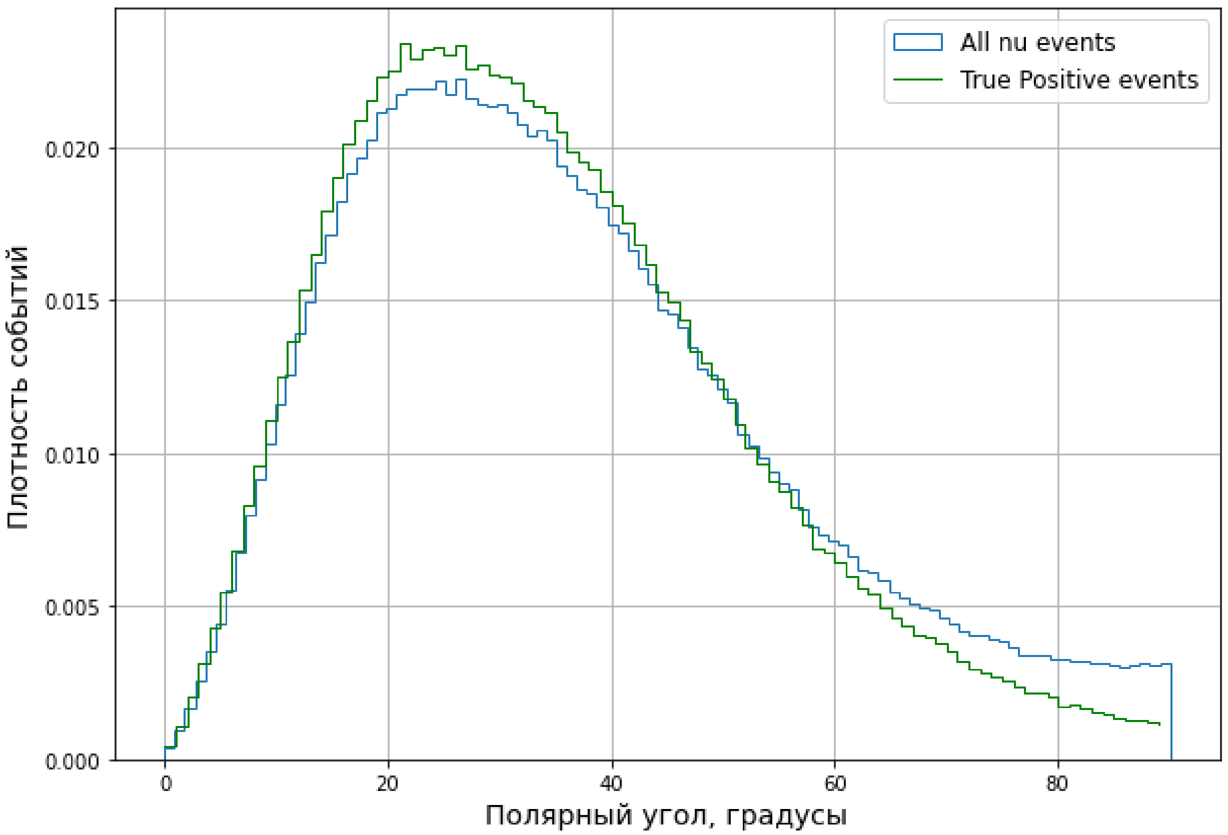
Polar angle

Спектры частиц

Энергия, нейтрино

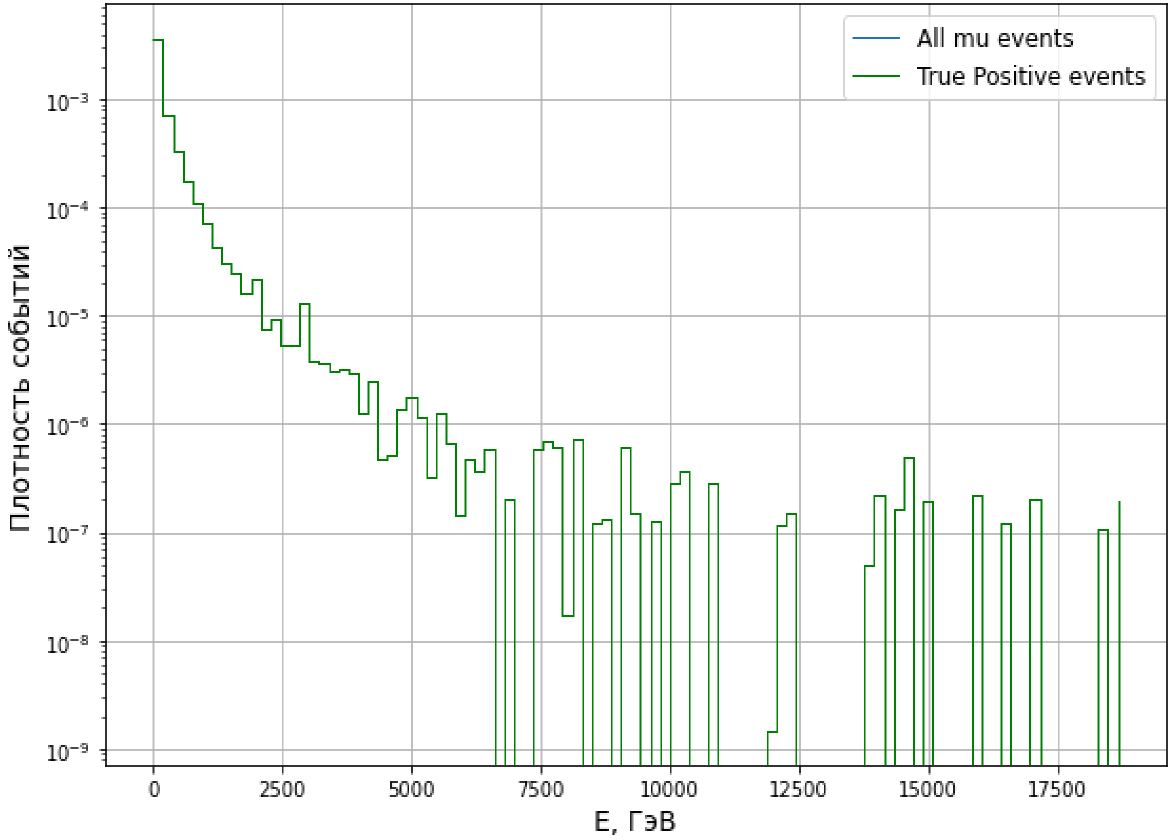


Полярный угол, нейтрино

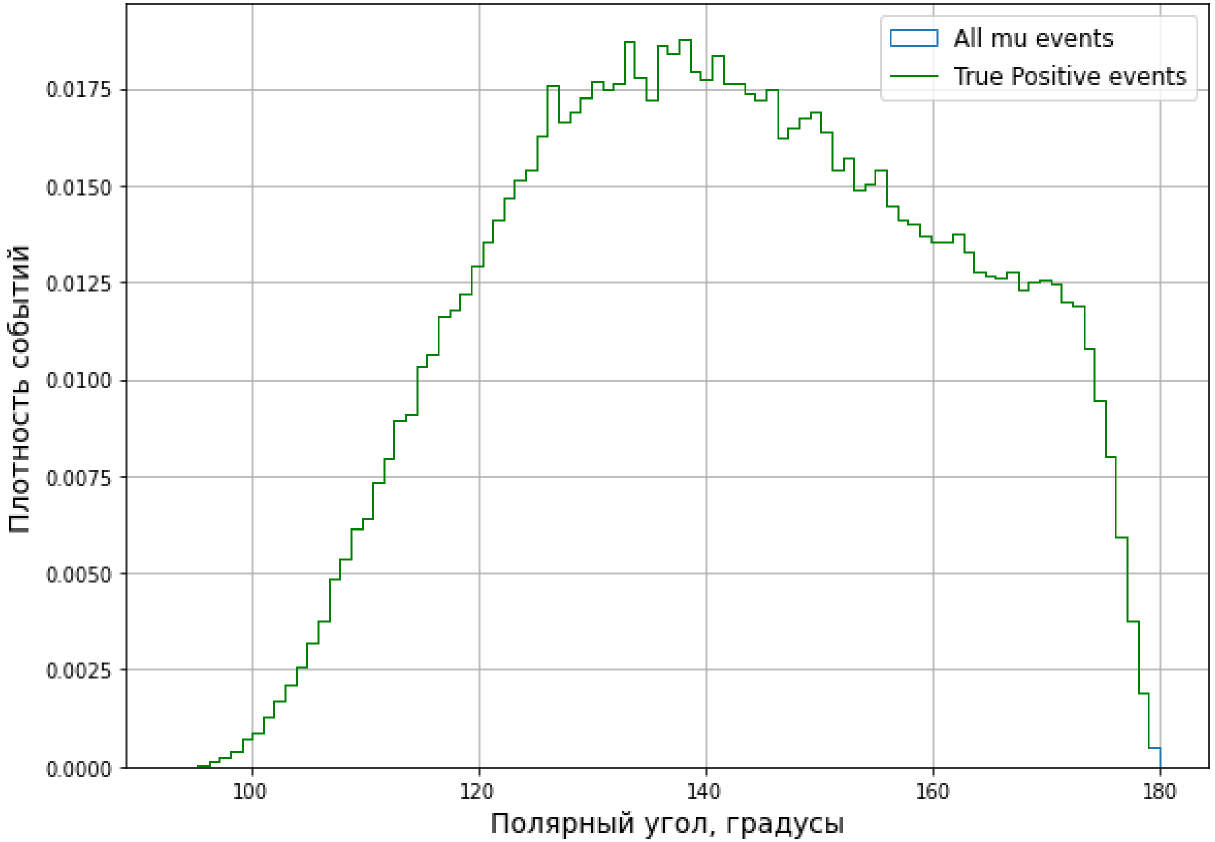


Спектры частиц

Энергия, мюоны

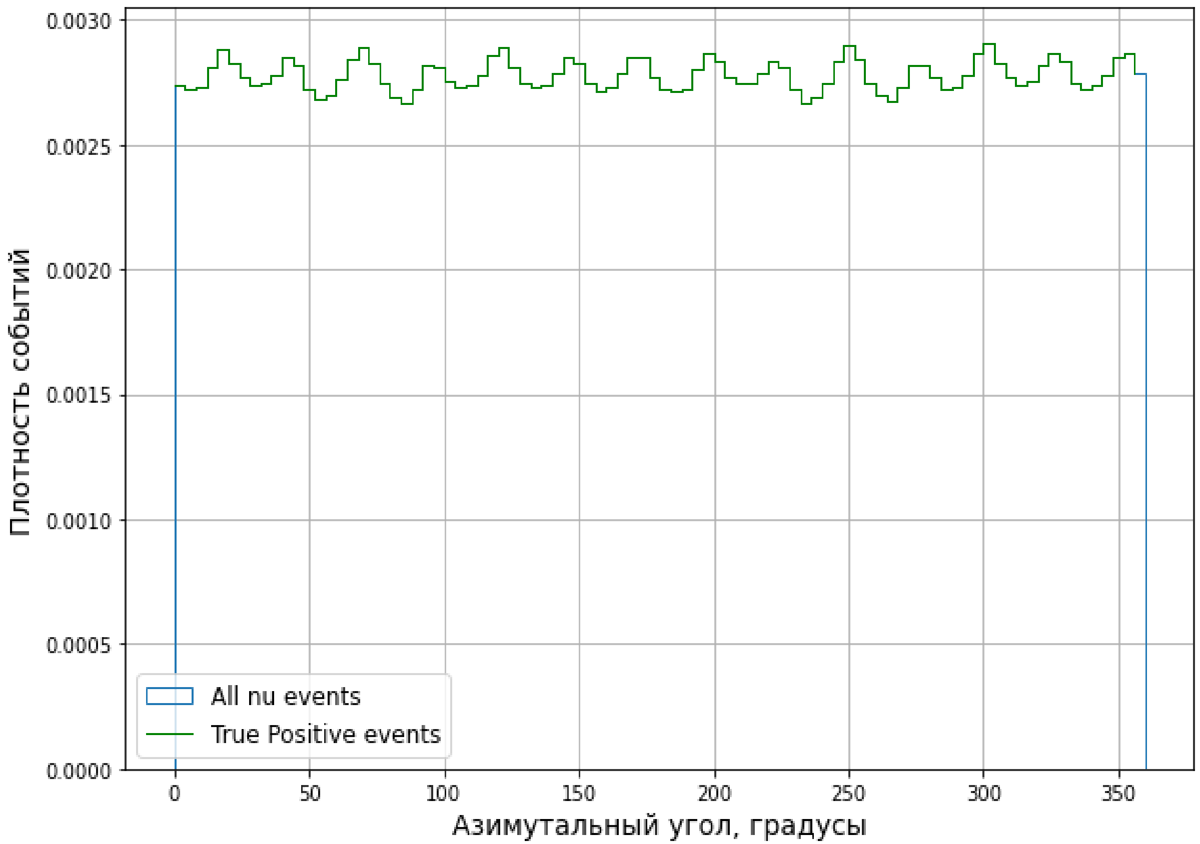


Полярный угол, мюоны

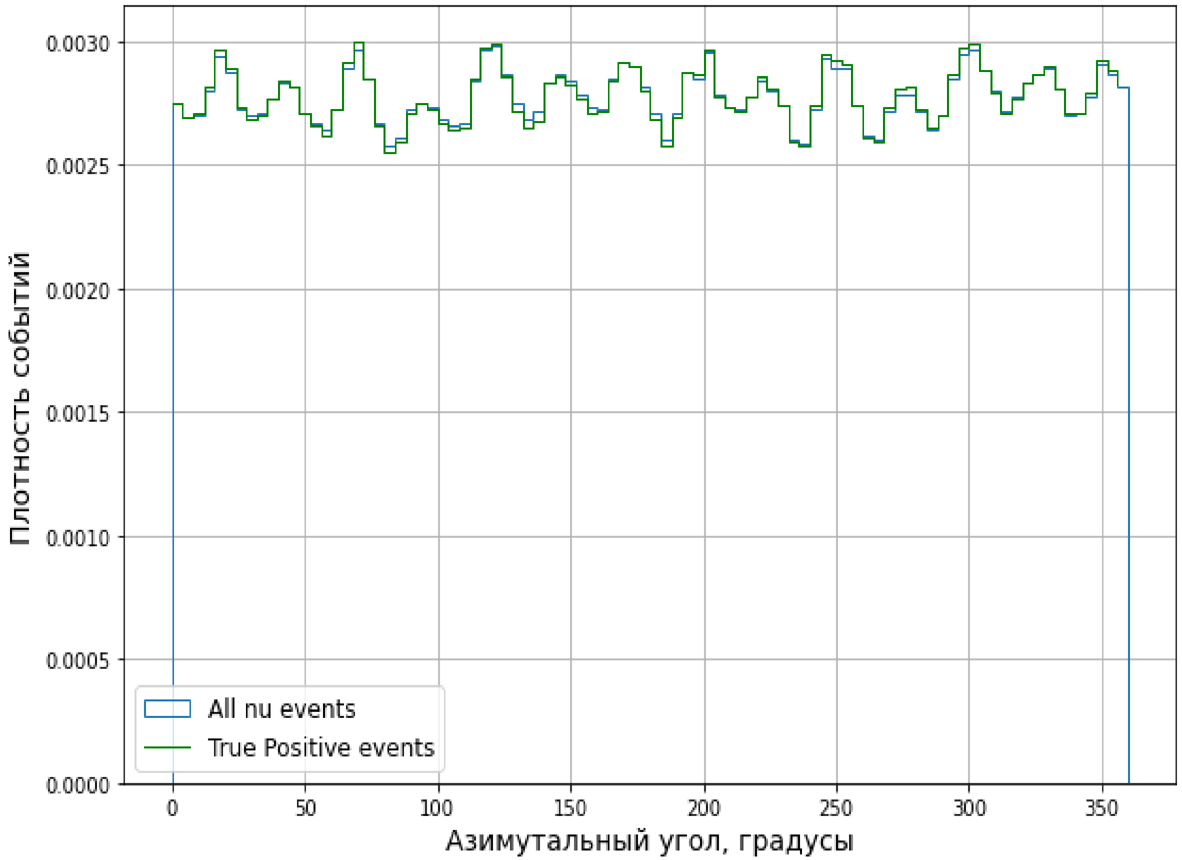


Спектры частиц

Азимутальный угол, мюоны



Азимутальный угол, нейтрино



Разные формулы

Focal loss

$$FL(p_t) = -\alpha_t (1-p_t)^{\gamma} \log(p_t)$$

$$p_{\rm t} = \begin{cases} p & \text{if } y = 1 \\ 1 - p & \text{otherwise,} \end{cases}$$

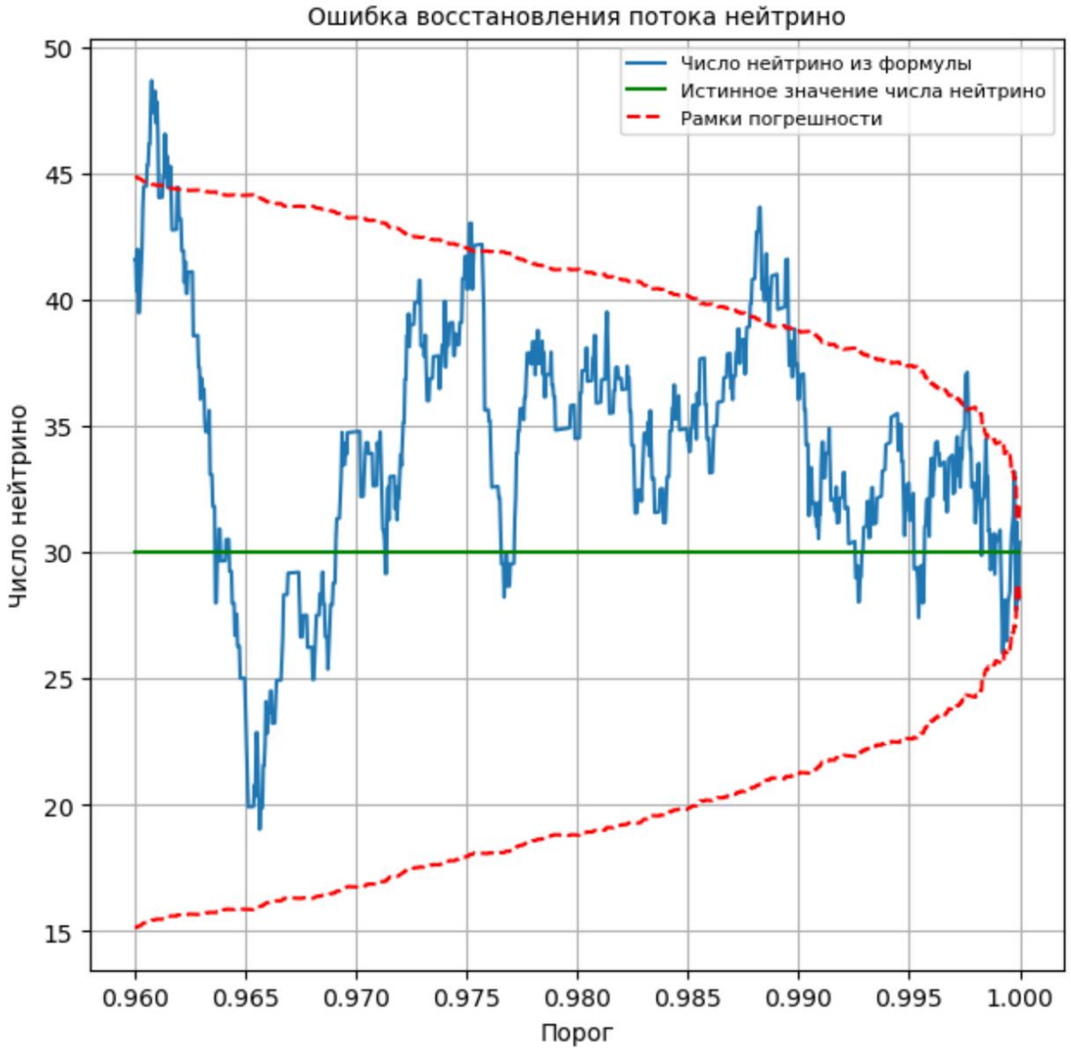
Оценка потока нейтрино (Следует из определний E и S)

$$n_\nu \approx \frac{n(\xi) - S^0(\xi)\cdot n(0)}{E^0(\xi) - S^0(\xi)}$$

ξ - порог классификации.

S^0, E^0 - оценки подавления и экспозиции на тестовом МК наборе данных.

n(ξ) - количество событий правее порога.



Можно оценить ошибку!

Отношение ШАЛ к ν : ~ 100 000

Оценка потока нейтрино

$$n_\nu \approx \frac{n(\xi) - S^0(\xi)\cdot n(0)}{E^0(\xi) - S^0(\xi)}$$

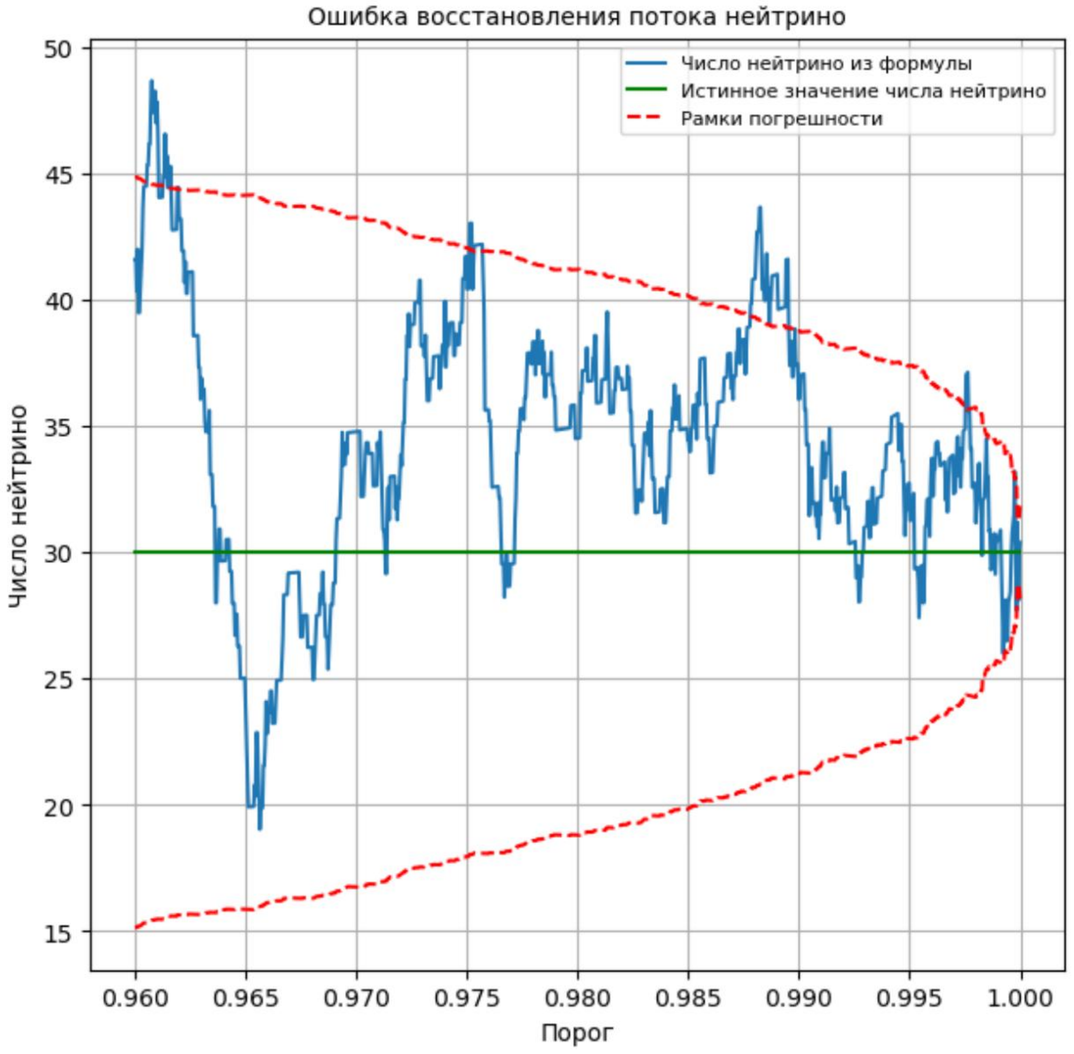
Можно оценить ошибку!

- Возьмём тестовые нейтринные события.

E - параметр биномиального распределения!
По МК оцениваем E^0 с доверительным интервалом.

- ШАЛ события и S^0 - аналогично!

- Считаем погрешность формулы потока.



Отношение ШАЛ к ν : ~ 100 000

ξ - порог классификации.
S^0, E^0 - подавление и экспозиция, оцененные на МК.
n(ξ) - количество событий, правее порога.

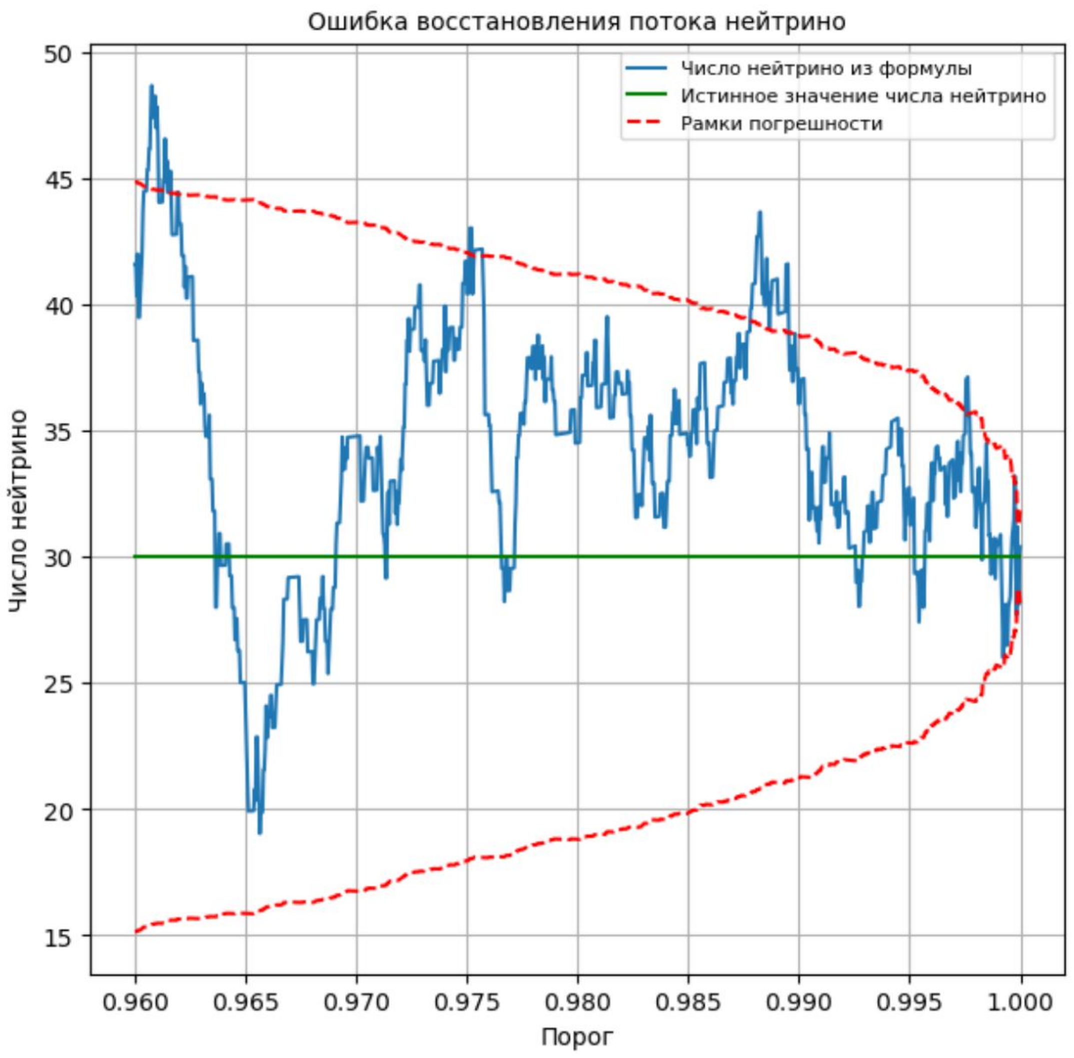
Оценка потока нейтрино

$$n_\nu \approx \frac{n(\xi) - S^0(\xi)\cdot n(0)}{E^0(\xi) - S^0(\xi)}$$

Можно оценивать в 2 режимах:

1) Оценка числа ν-событий в данных

2) Оценка параметра потока ν
(число n(ξ) - Пуассоновская случ. величина)



Отношение ШАЛ к ν : ~ 100 000

ξ - порог классификации.
S^0, E^0 - подавление и экспозиция, оцененные на МК.
n(ξ) - количество событий, правее порога.

Вывод формулы ошибки

Appendix A: Derivation of Eq. (??)

In this section we derive distributions, expected values and dispersions of random values in Eq.(??). Then, using them, we evaluate the error of estimating the number of neutrino events using Eq.(??). In this section, we use P to denote probability of some outcome for a random variable, and M and D for its expected value and dispersion, accordingly.

We start by discussing the properties of random variables in the experimental dataset. Let the latter contains n^0 events in total, n^0_ν of which are neutrino-induced and $n^0_\mu \equiv n^0 - n^0_\nu$ are EAS-induced. n^0_ν and n^0_μ are random variables distributed according to the Poisson law with parameters ν and μ respectively. Hence

$$P(n^0_\nu = k) = \frac{\nu^k e^{-\nu}}{k!}\,, \qquad \text{(A1)}$$

$$P(n^0_\mu = k) = \frac{\mu^k e^{-\mu}}{k!}\,. \qquad \text{(A2)}$$

Since n^0 is a sum of n^0_ν and n^0_μ, it also follows the Poisson distribution,

$$P(n^0 = k) = \frac{(\nu+\mu)^k e^{-(\nu+\mu)}}{k!}\,. \qquad \text{(A3)}$$

It's expected value and dispersion are:

$$M(n^0) = D(n^0) = \nu + \mu\,. \qquad \text{(A4)}$$

Let us now address the classification of events by the neural network. A trained neural network can be considered as a black box. As it was discussed in section ??, for a fixed classification threshold ξ, the network classifies a neutrino-induced event correctly with some probability E, and EAS-induced event is identified incorrectly with the probability S. Hence the number of identified true and false neutrino-induced events are independent random variables with binomial distributions:

$$P(n_\nu = k | n^0_\nu = m) = Bin(m, E)(k)\,, \qquad \text{(A5)}$$

$$P(n_\mu = k | n^0_\mu = m) = Bin(m, S)(k)\,. \qquad \text{(A6)}$$

Here $n_\nu(\xi) \equiv n_\nu$, $n_\mu(\xi) \equiv n_\mu$, and $Bin(m,p)(k)$ stands for the binomial distribution with number of experiments m and success probability p:

$$Bin(m,p)(k) = C^k_m p^k (1-p)^{m-k}\,. \qquad \text{(A7)}$$

The number of neutrino-induced events identified by the neural network on a test dataset is subject to both of the above-described random processes. Hence the full probability distributions, P_f, of n_ν and n_μ can be obtained by multiplying the corresponding Poisson and binomial distributions,

$$P_f(n_\nu = k) = \sum_{m=0}^{\infty} P(n_\nu = k | n^0_\nu = m) P(n^0_\nu = m) \qquad \text{(A8)}$$

$$P_f(n_\mu = k) = \sum_{m=0}^{\infty} P(n_\mu = k | n^0_\mu = m) P(n^0_\mu = m) \qquad \text{(A9)}$$

Using Eq. (A8), Eq. (A9), Eq. (A1) and Eq. (A2), one can evaluate the expected values and dispersions of n_ν and n_μ:

$$M(n_\nu) = D(n_\nu) = E\nu\,, \qquad \text{(A10)}$$

$$M(n_\mu) = D(n_\mu) = S\mu\,. \qquad \text{(A11)}$$

For the random variable n, which is a sum of n_ν and n_μ, one has:

$$M(n) = D(n) = E\nu + S\mu\,. \qquad \text{(A12)}$$

Now the task is to estimate binomial parameters E and S of the neural network after measurements of n_ν and n_μ on the test dataset. To do this, we use standard formulae:

$$\tilde{E} = \frac{n_\nu(\xi)}{n^0_\nu}\,, \quad \tilde{S} = \frac{n_\mu(\xi)}{n^0_\mu}\,. \qquad \text{(A13)}$$

The errors of these evaluations are calculated using the Clopper–Pearson interval. For example, setting confidence level $1-\alpha$ for $\tilde{E}$, we obtain interval $E_{\min} < E < E_{\max}$, where $E_{\min}$ and $E_{\max}$ are from equations:

$$\frac{\Gamma(n^0_\nu + 1)}{\Gamma(n_\nu)\Gamma(n^0_\nu - n_\nu + 1)} \int_0^{E_{\min}} t^{n_\nu - 1}(1-t)^{n^0_\nu - n_\nu} dt = \frac{\alpha}{2} \qquad \text{(A14)}$$

$$\frac{\Gamma(n^0_\nu + 1)}{\Gamma(n_\nu + 1)\Gamma(n^0_\nu - n_\nu)} \int_0^{E_{\max}} t^{n_\nu}(1-t)^{n^0_\nu - n_\nu - 1} dt = 1 - \frac{\alpha}{2} \qquad \text{(A15)}$$

The variation of $\tilde{E}$ with confidence level $1-\alpha = 0.68$ we will consider to be equivalent to one standard deviation and calculate as

$$\sigma_{\tilde{E}} = (E_{\max} - E_{\min})/2, \qquad \text{(A16)}$$

For case of S the reasoning is similar:

$$\sigma_{\tilde{S}} = (S_{\max} - S_{\min})/2 \qquad \text{(A17)}$$

Now we are ready to evaluate the dispersion of N_ξ estimated using Eq. (??). For this purpose, we used the standard formula for the dispersion of a function of random variables,

$$\sigma^2_{N_\xi} = \sum_v \left(\frac{\partial N_\xi}{\partial v}\right)^2 \sigma^2_v + 2\sum_{v \neq u} \left(\frac{\partial N_\xi}{\partial v}\right)\left(\frac{\partial N_\xi}{\partial u}\right) Cov_{v,u}\,. \qquad \text{(A18)}$$

Here, v and u denote arguments of the function N_ξ, which are $n(\xi) \equiv n$, $n(0) \equiv n^0$, $\tilde{E}(\xi)$ and $\tilde{S}(\xi)$; σ^2_v stands for squared variance of v, and $Cov_{v,u}$ denotes covariance between v and u.

Let us explicitly write out the estimation of the variances and covariances. According to Eq. (A4), $\sigma^2_{n^0}$ can be estimated as

$$\sigma^2_{n^0} = n^0\,. \qquad \text{(A19)}$$

Further, from Eq. (A12), one gets

$$\sigma^2_n = n\,. \qquad \text{(A20)}$$

Next, $\sigma^2_{\tilde{E}}$ and $\sigma^2_{\tilde{S}}$ can be obtained from Eq. (A16) and Eq. (A17).

Finally, note that there are only two dependent random variables in Eq. (??) — n and n^0. Their covariance can be calculated using Eq. (A3), Eq. (A8) and Eq. (A9),

$$Cov_{n^0,n} = E\nu + S\mu = M(n)\,. \qquad \text{(A21)}$$

Therefore this covariance can be estimated as n.

By calculating the partial derivatives of N_ξ in Eq. (A18) and substituting the obtained expressions for variances and covariances, we obtain the final result:

$$\sigma^2_{N_\xi} = \frac{(n - n^0\tilde{S})^2}{(\tilde{E}-\tilde{S})^4}\cdot\sigma^2_{\tilde{E}} + \frac{(n - n^0\tilde{E})^2}{(\tilde{E}-\tilde{S})^4}\cdot\sigma^2_{\tilde{S}} + \frac{n + n^0(\tilde{S})^2 - 2n\tilde{S}}{(\tilde{E}-\tilde{S})^2} \qquad \text{(A22)}$$

* E-mail: matseiko.av@phystech.edu

† E-mail: ivan.kharuk@phystech.edu