

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего
образования

Национальный исследовательский ядерный университет «МИФИ»

Отчет о научно-исследовательской работе
на тему:

Применение нейронных сетей для реконструкции
положения события в эксперименте DEAP-3600

Выполнил:

Студент гр. М19-115

Ильясов А.И.

Научный руководитель:

к.ф.-м.н.

Гробов А.В.

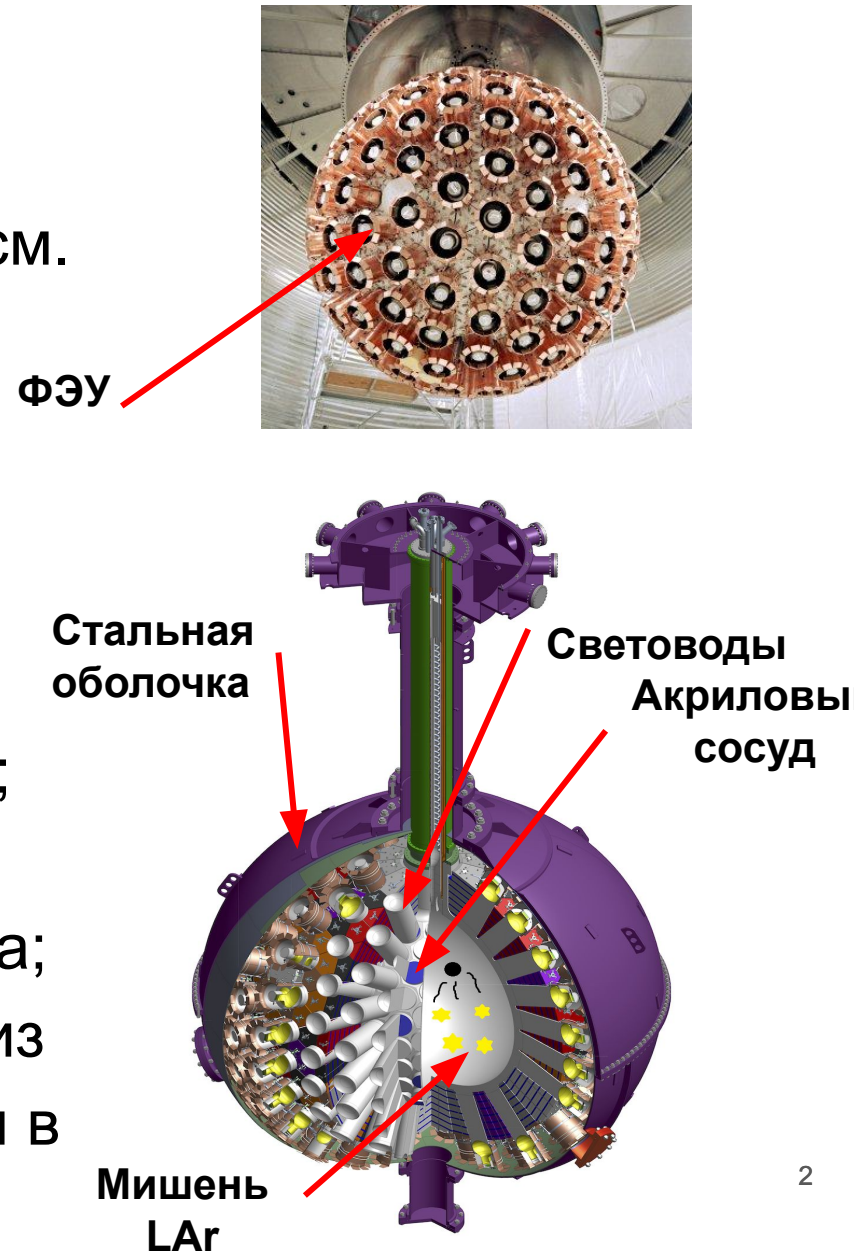
Москва, 2020



Детектор DEAP-3600

The DEAP Collaboration, Design and Construction of the DEAP-3600 Dark Matter Detector, *Astropart. Phys.* 108, 1 (2019).

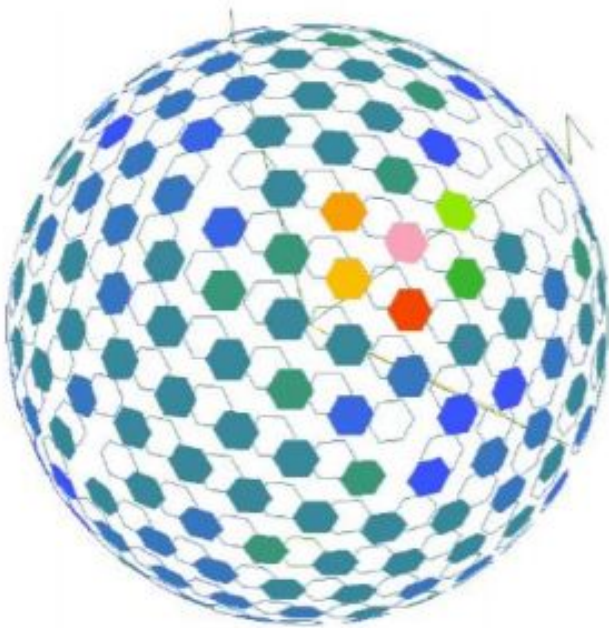
- Расположен в 2 км. под землей;
- (3279 ± 96) кг. LAr (84 K) в UVA акриловом сосуде толщиной 5 см. и диаметром 1.7 м.;
- События регистрируются 255 ФЭУ;
- **Горловина** детектора, который используется для поддержания нужной температуры детектора;
- Имеется **нейтронная и черенковская мюонная защита**;
- Детектор заключен в оболочку из нержавеющей стали и погружен в бассейн 300 т. воды.



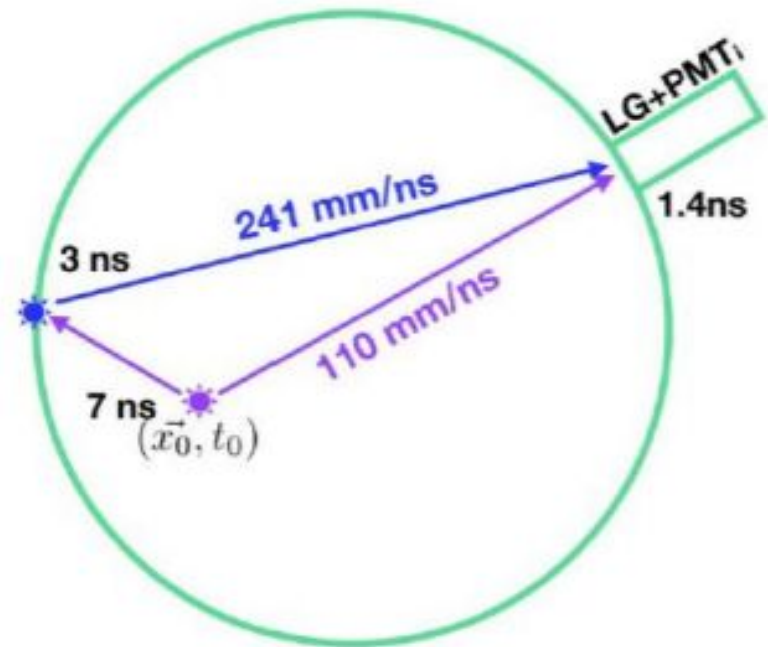
Имеющиеся алгоритмы

The DEAP Collaboration, Search for dark matter with a 231-day exposure of liquid argon using DEAP-3600 at SNOLAB, Physical Review D 100.2 (2019)

1. Метод, основанный на количестве выделившихся фотоэлектронов



2. Метод, основанный на временном распределении регистрации фотоэлектронов

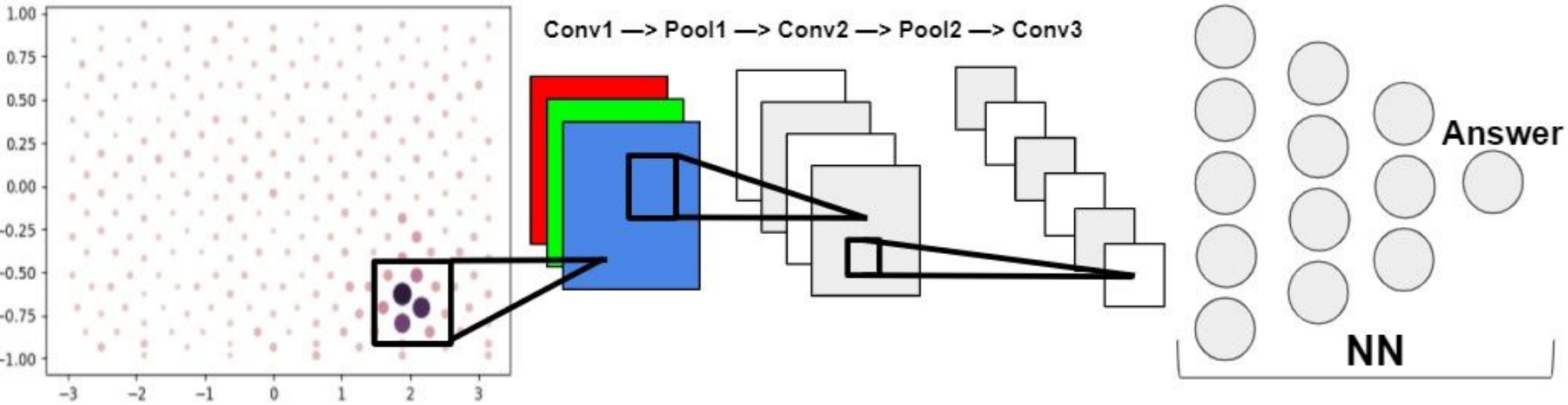


I. После **фидуциализации** ошибка второго алгоритма около **13%**;

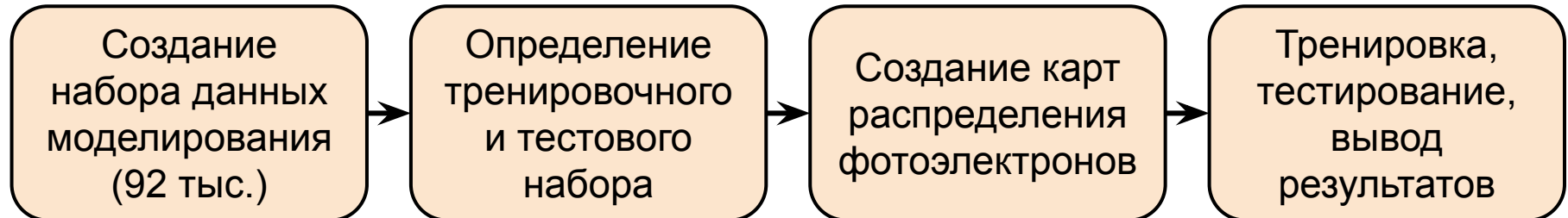
II. Алгоритмы согласуются по Z-координате с **точностью** до 35 мм. для 50% событий;

III. **Второй** алгоритм для **верхней** части детектора, **первый** алгоритм - для **нижней** части детектора

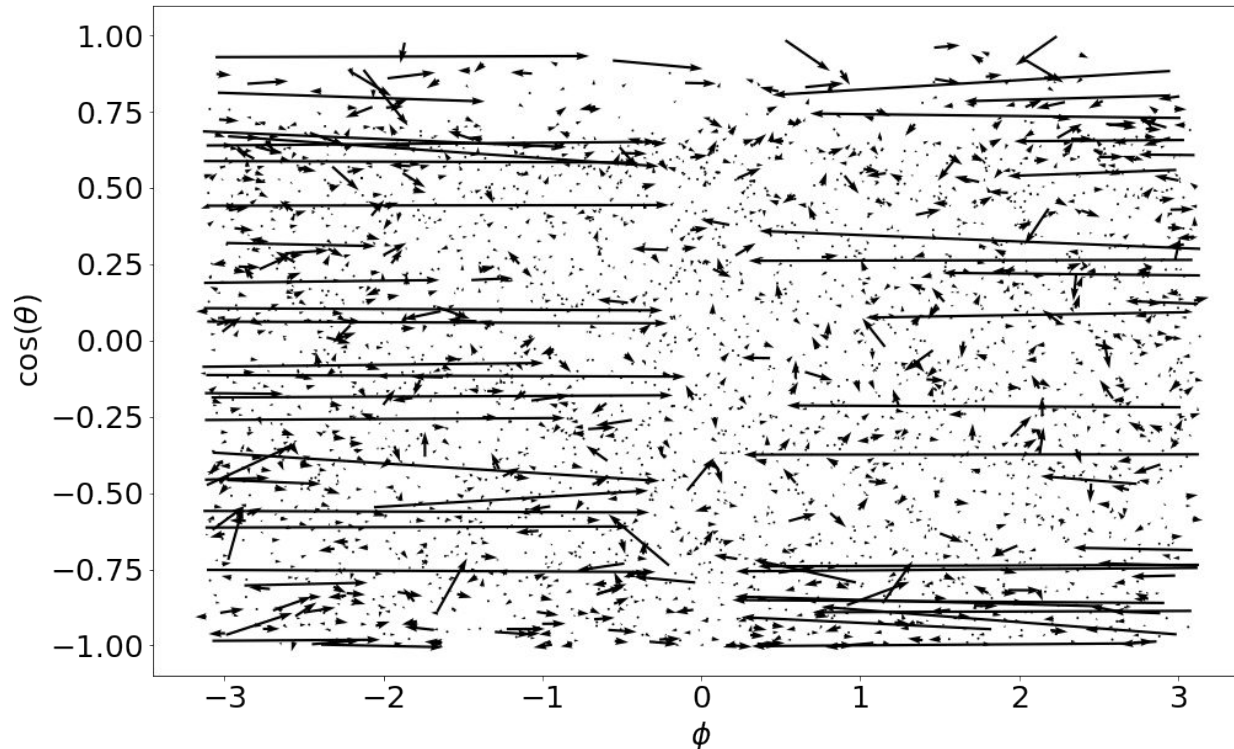
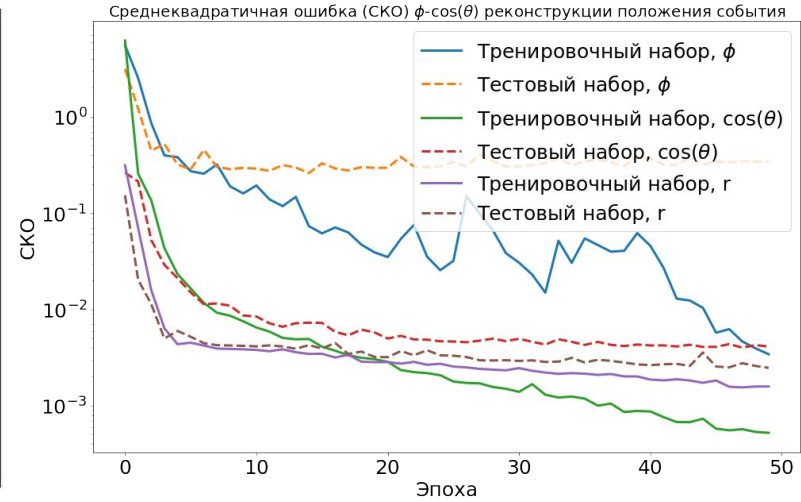
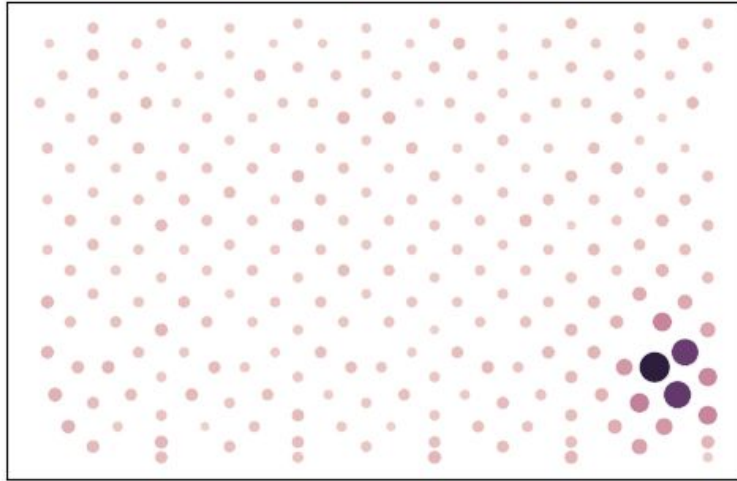
Свёрточная нейронная сеть



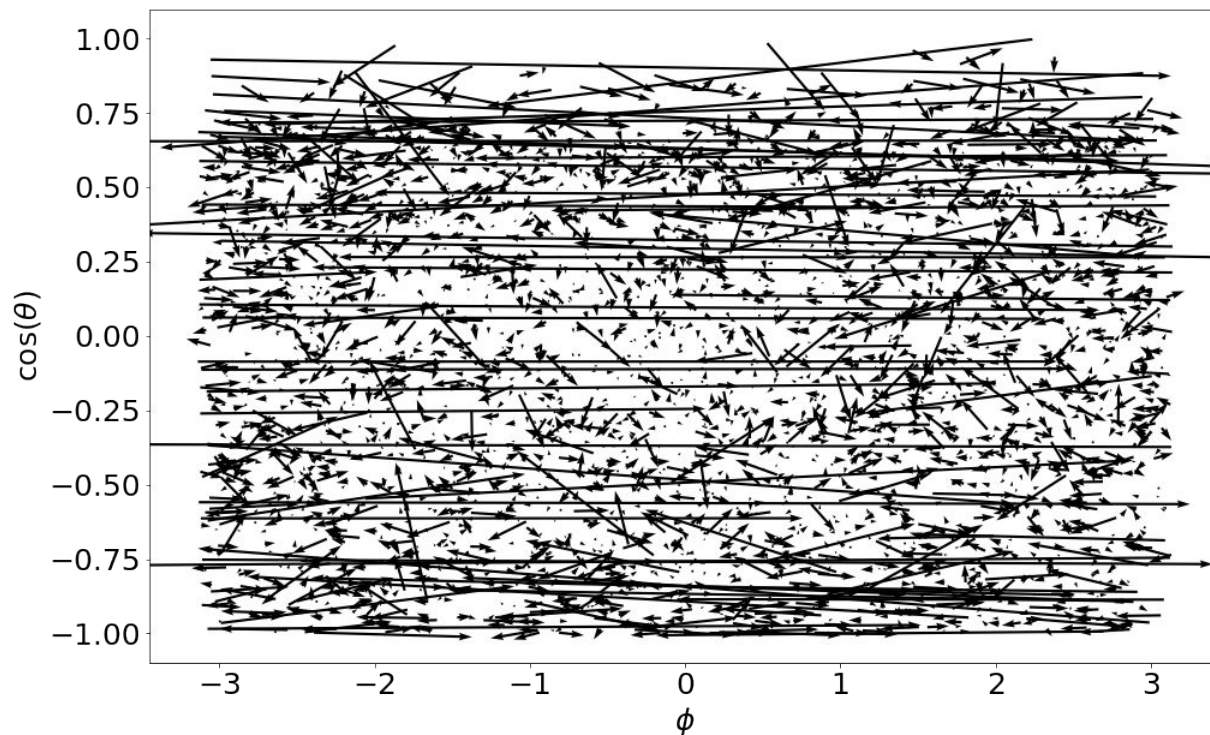
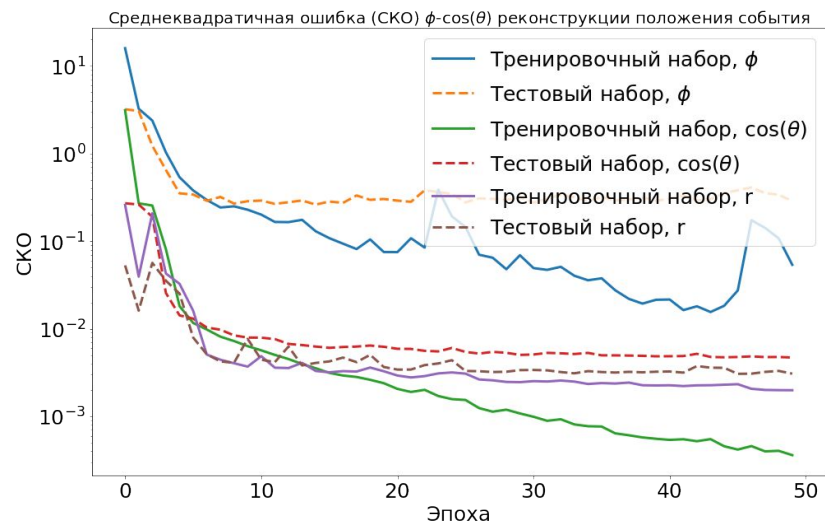
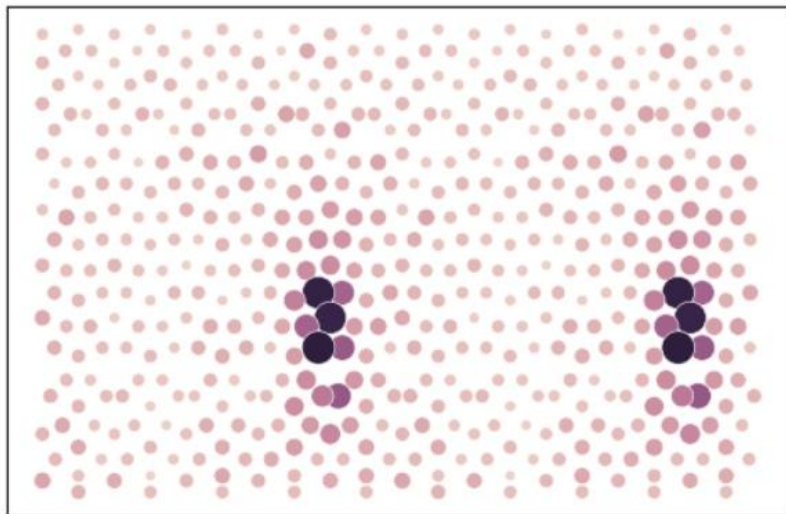
Последовательный стек слоев



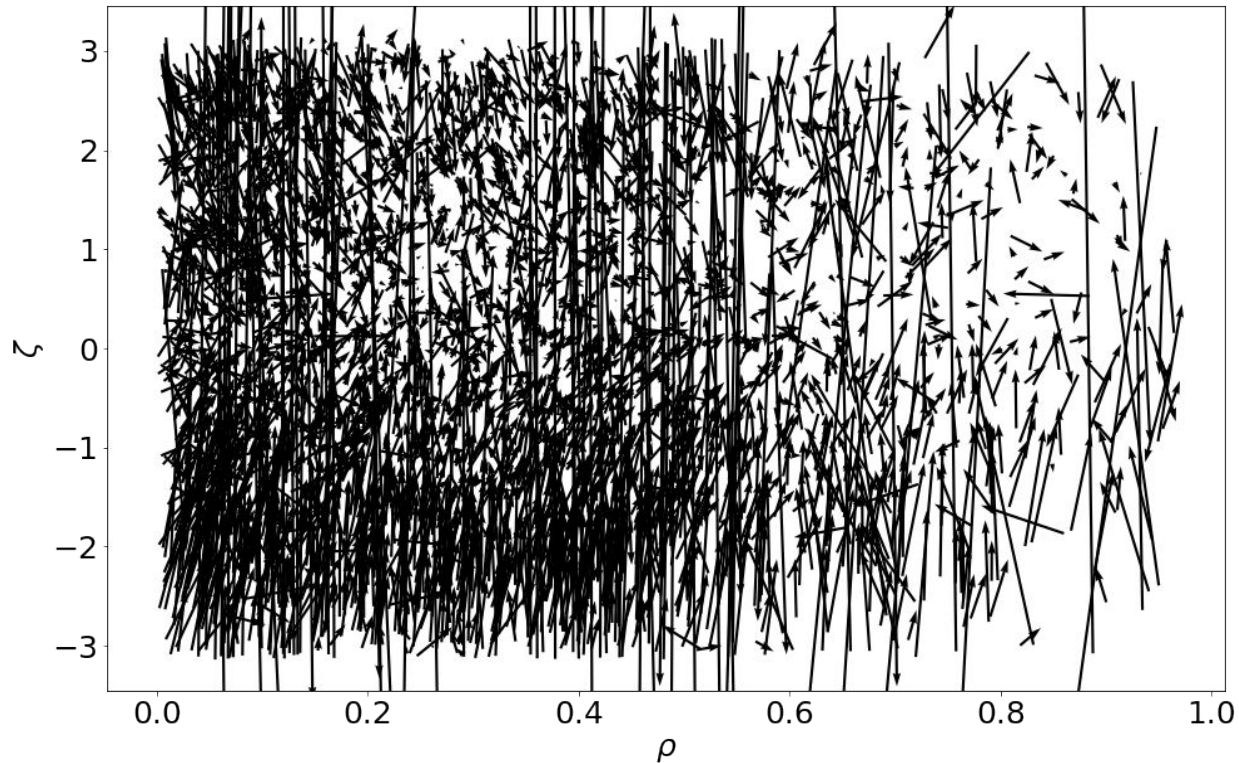
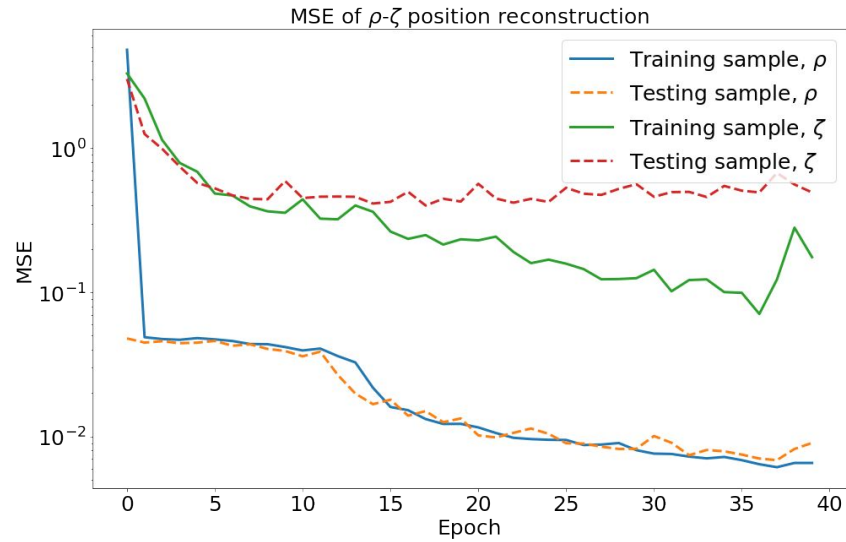
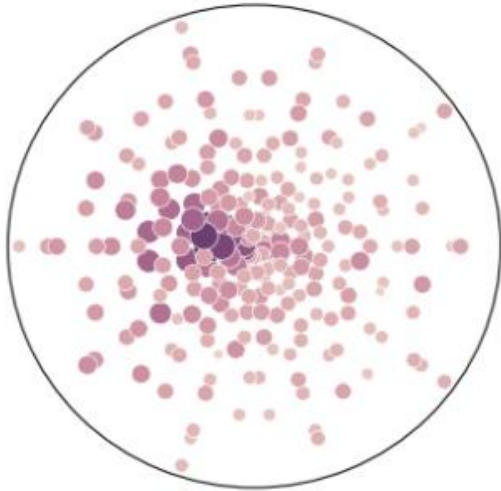
Результаты: Модель 1. $\phi \in [-\pi; \pi] - \cos(\theta) \in [-1; 1] - r \in [0, 1]$



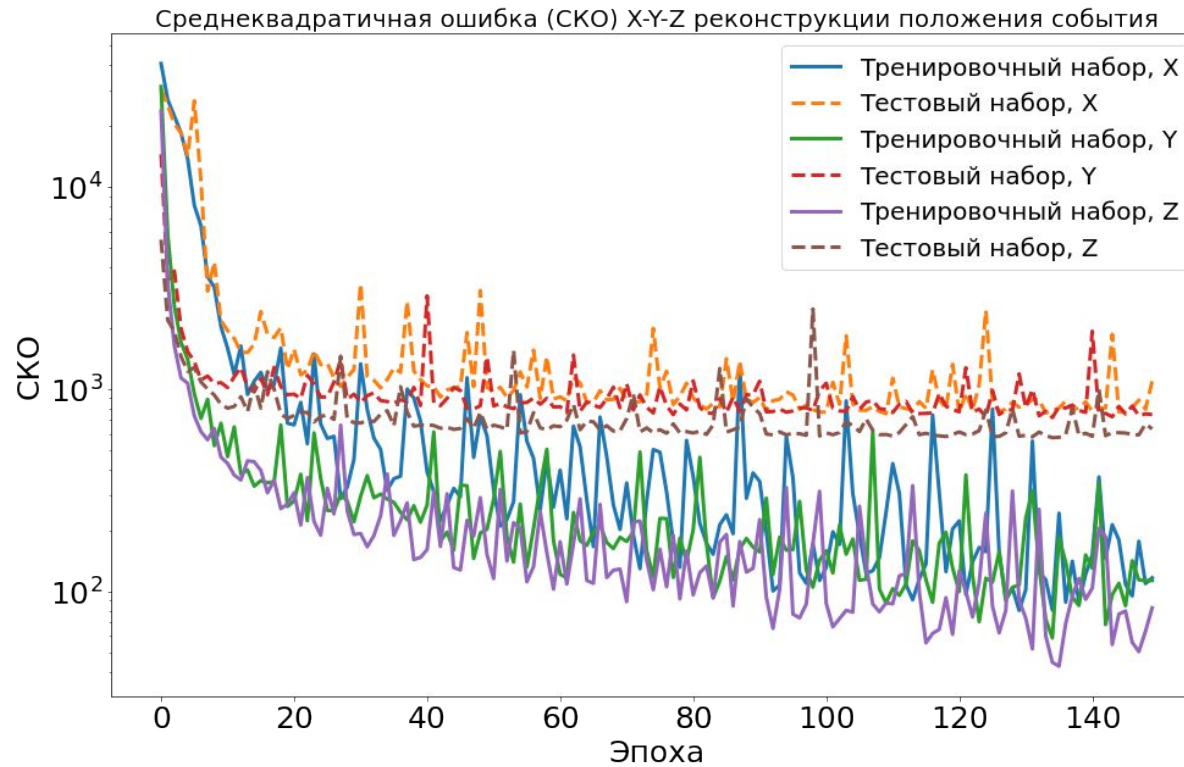
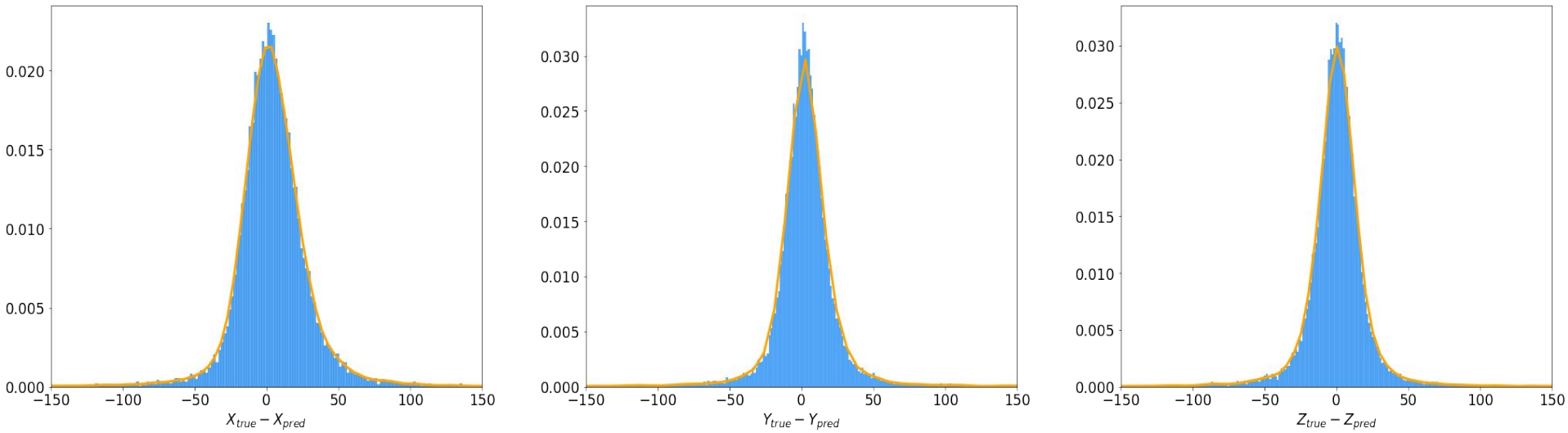
Результаты: Модель $2.\phi \in [-2\pi; 2\pi] - \cos(\theta) \in [-1; 1] - r \in [0, 1]$



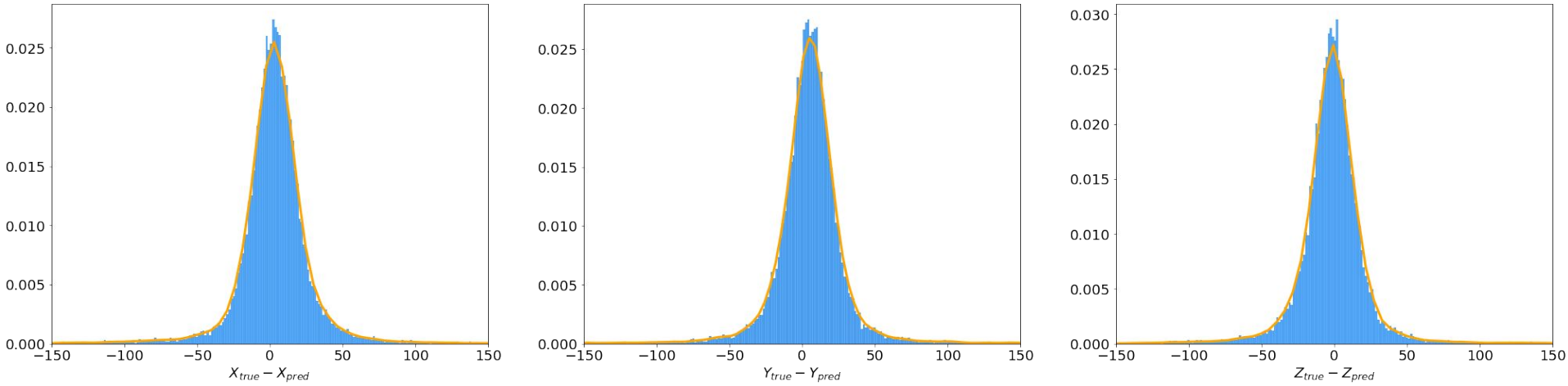
Результаты: Модель 3. $\zeta \in [-\pi; \pi] - \rho \in [0, 1]$



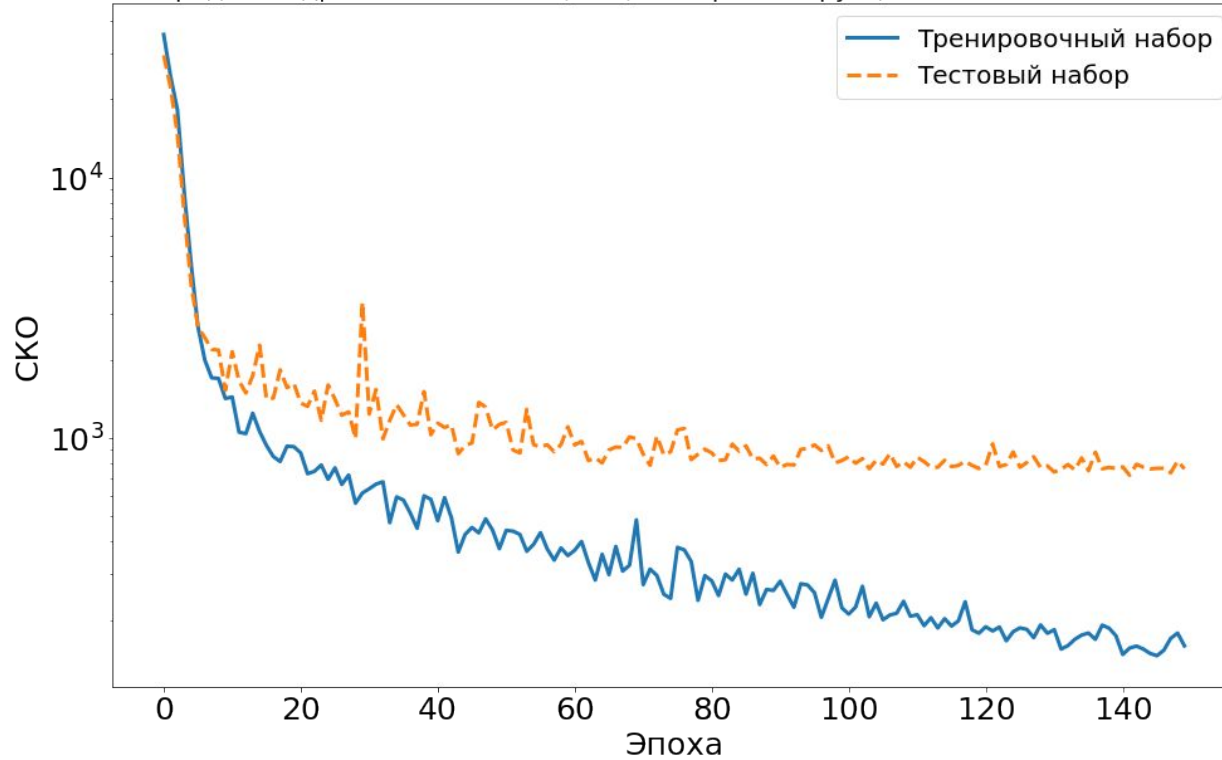
Результаты: Модель 4. Нейронная сеть (X,Y,Z)[-850;850]



Результаты: Модель 5. Нейронная сеть (X-Y-Z)[-850;850]



Среднеквадратичная ошибка (СКО) X-Y-Z реконструкции положения события



Заключение

- (Свёрточные) нейронные сети можно использовать для реконструкции положения события внутри детектора;
- На данном этапе точность разработанных моделей сопоставимы с имеющимися алгоритмами;
- Наилучшие алгоритмы:
 - Сферическая развёртка детектора $\phi [-\pi; \pi] - \cos(\theta) [-1; 1] - r [0, 1]$
 - Нейронная сеть с одновременным определением координат (X-Y-Z)[-850;850]

	$\phi, \cos(\theta), r$	$2\phi, \cos(\theta), r$	ζ, r	X-Y-Z	X, Y, Z
$\sigma(x, \text{мм})$	95.7	1162.8	122.5	32.4	39.5
$\mu(x, \text{мм})$	-7.7	-6.3	-15.7	2.9	3.8
$\sigma(y, \text{мм})$	103.1	1136.5	124.9	30.9	28.9
$\mu(y, \text{мм})$	-4.3	4.8	-21.1	5.1	2.6
$\sigma(z, \text{мм})$	57.6	922.8	—	27.9	25.1
$\mu(z, \text{мм})$	-2.6	67.3	—	-1.9	0.4

Back-up

Имеющиеся алгоритмы

The DEAP Collaboration, Search for dark matter with a 231-day exposure of liquid argon using DEAP-3600 at SNOLAB, Physical Review D 100.2 (2019)

**Метод, основанный на временном распределении
регистрации фотоэлектронов**

$$\ln \mathcal{L}(t_0, \vec{x}_0) = \sum_{i=1}^{N_{\text{PE}}} \ln \mathcal{L}^{\text{res}}(t_i - t_0; \vec{x}_0, \text{PMT}_i)$$

**Метод, основанный на количестве выделившихся
фотоэлектронов**

$$\ln \mathcal{L}(\vec{x}) = \sum_{i=1}^{N_{\text{PMTs}}} \ln \text{Poisson}(q_i, \lambda_i)$$

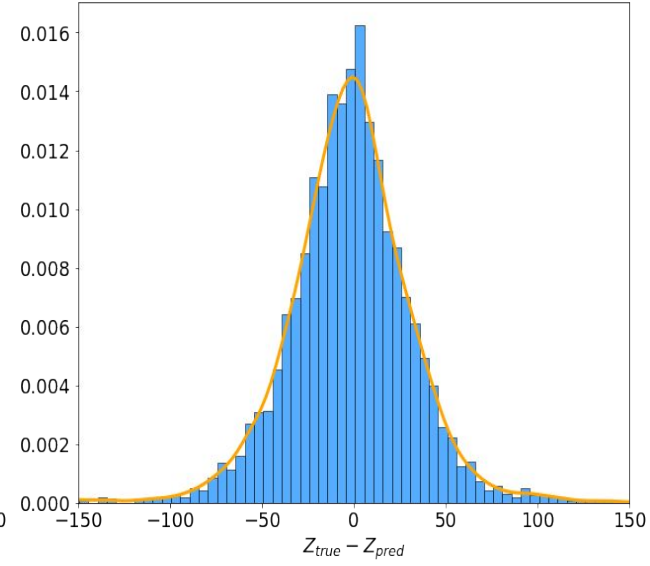
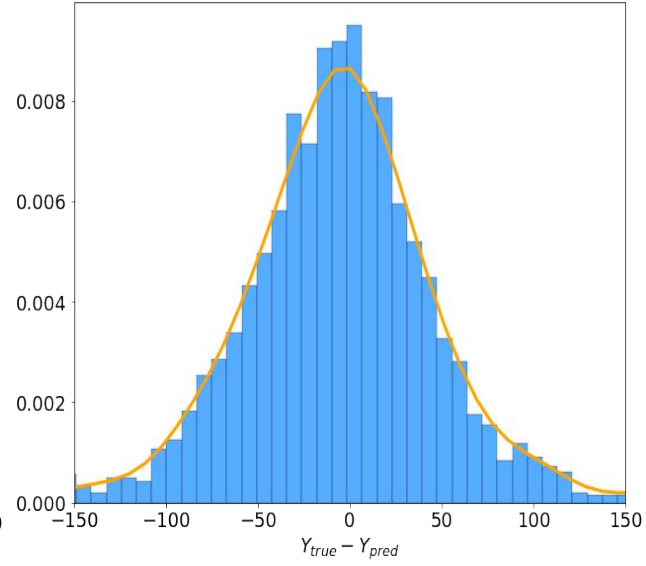
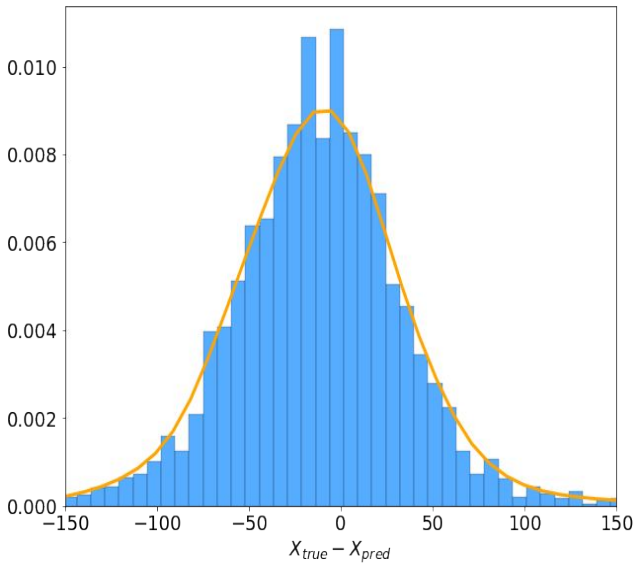
$$\lambda_i = \lambda_i \left(|\vec{x}|, \frac{\vec{x} \cdot \vec{r}_i}{|\vec{x}| |\vec{r}_i|}, q_{\text{total}} \right)$$

I. После **фидуциализации** ошибка второго алгоритма около **13%**;

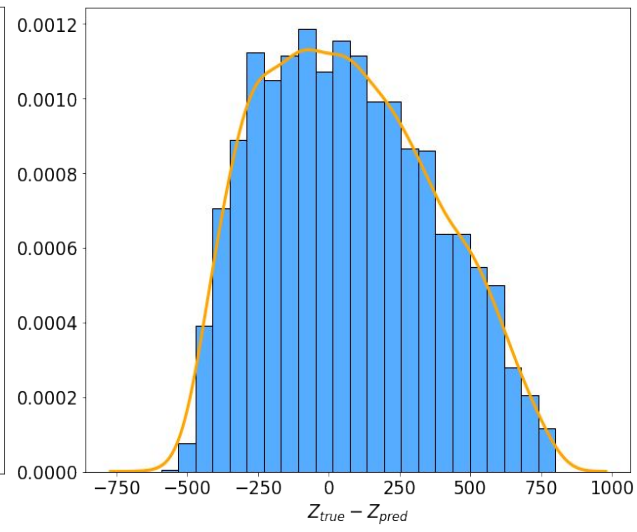
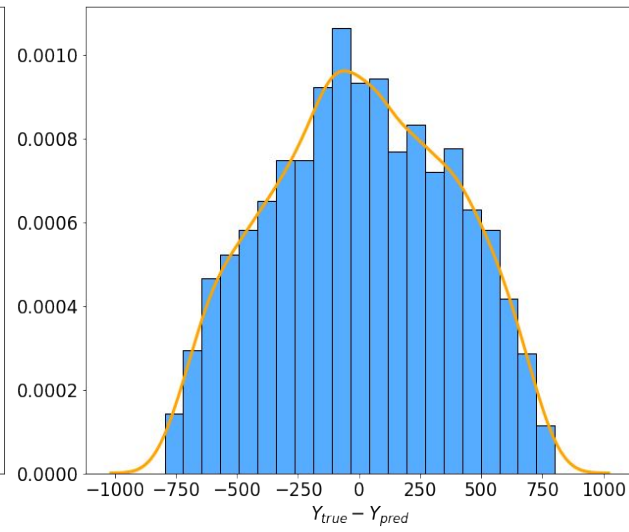
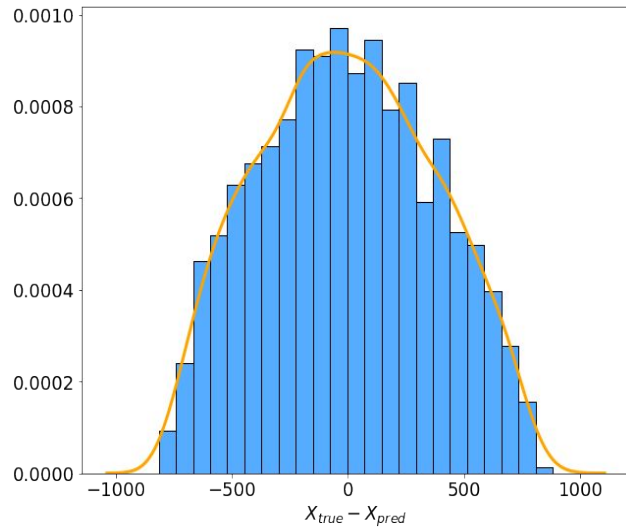
II. Алгоритмы согласуются по Z-координате с **точностью** до 35 мм. для 50% событий;

III. **Первый** алгоритм для **верхней** части детектора, **второй** алгоритм - для **нижней** части детектора

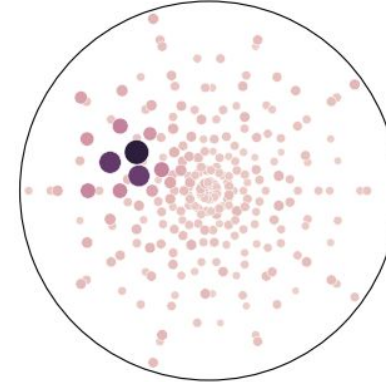
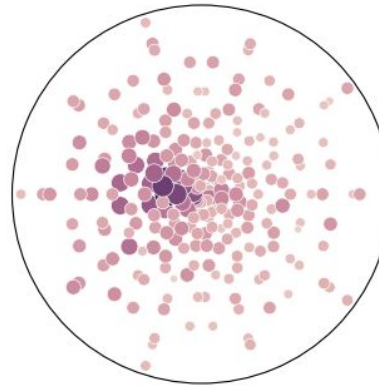
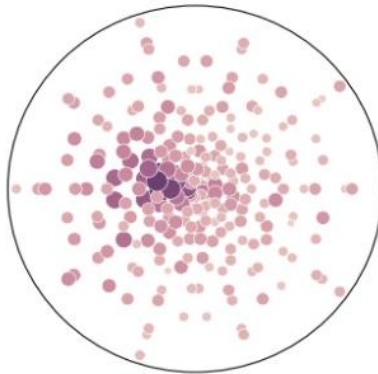
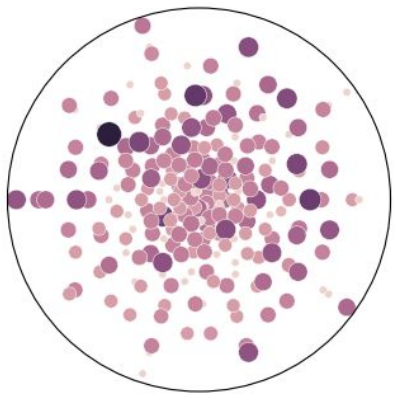
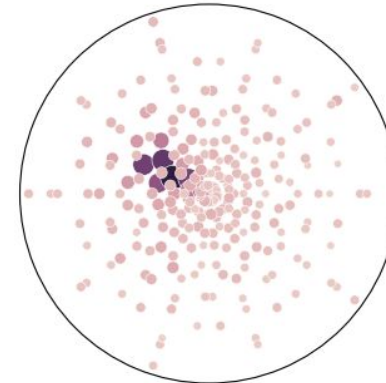
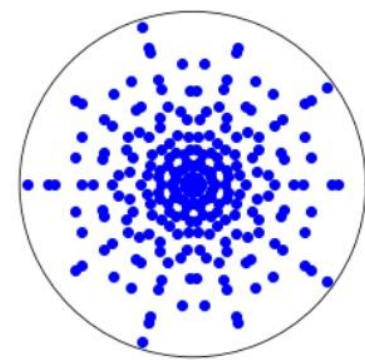
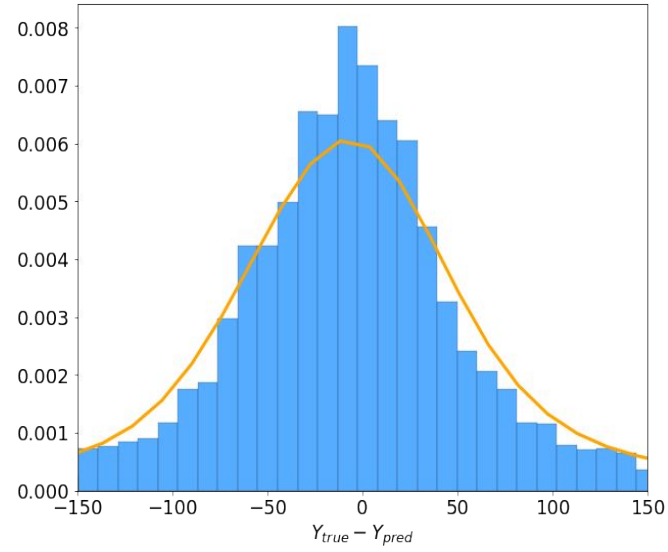
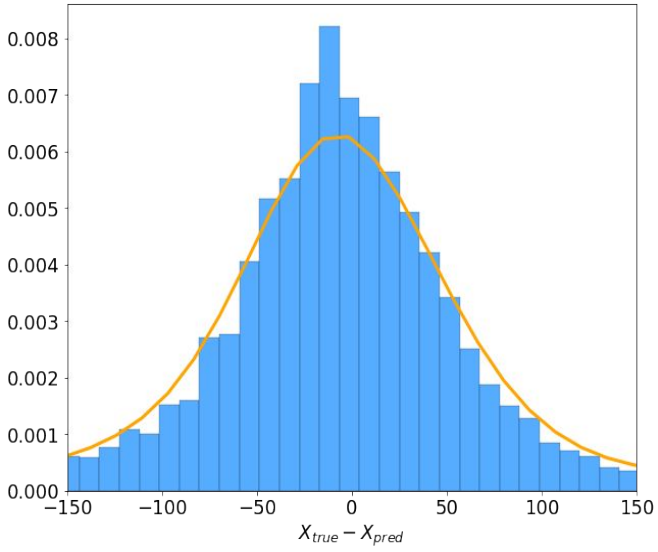
Модель 1



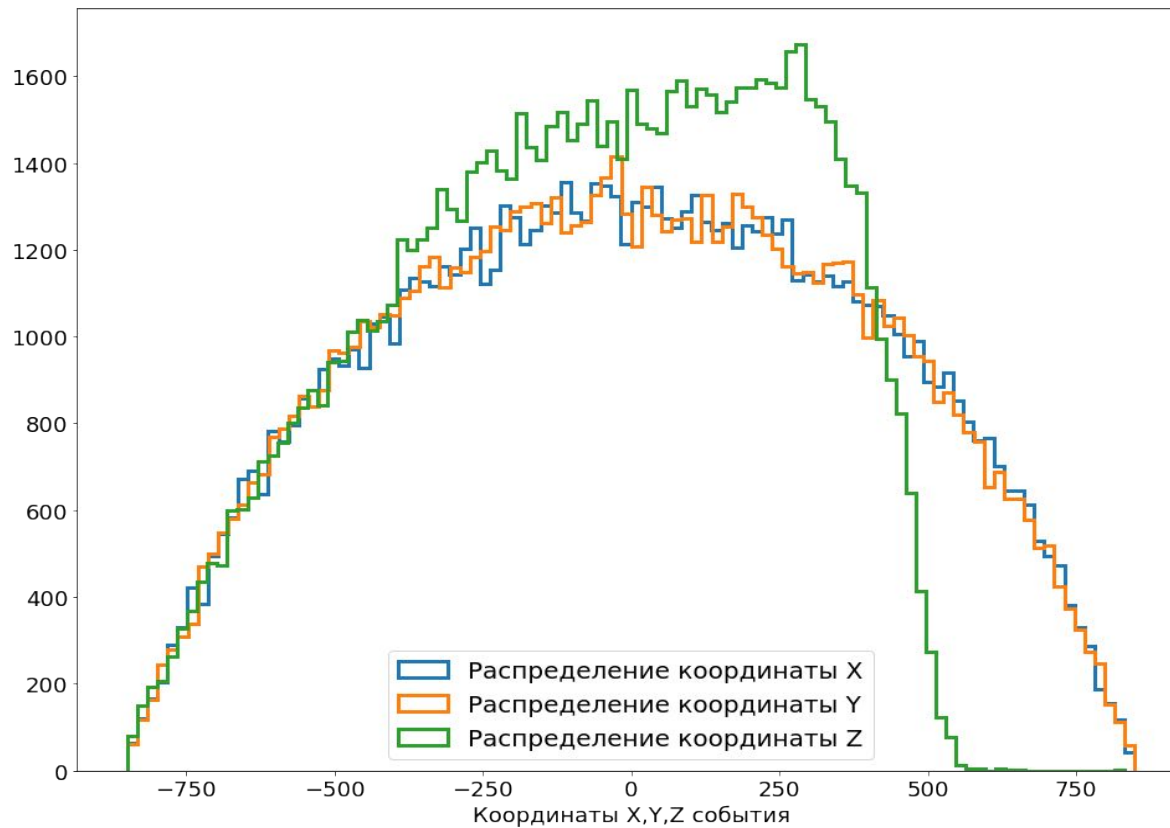
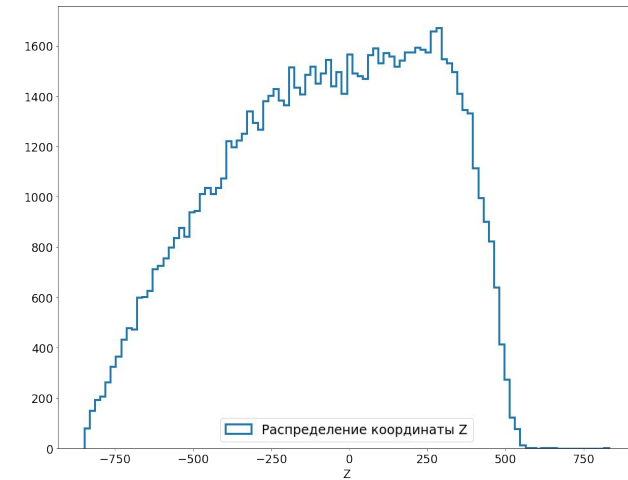
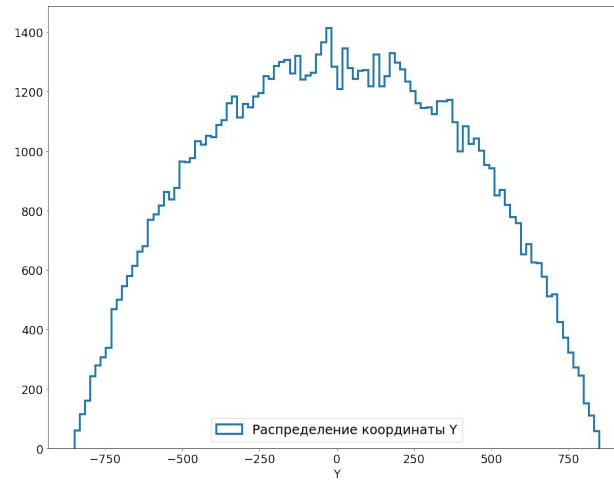
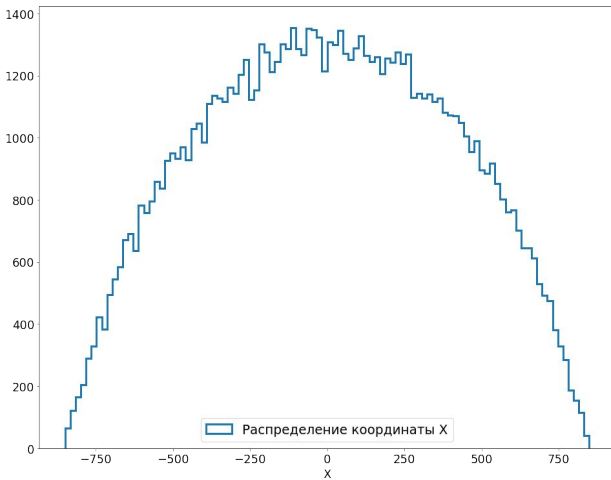
Модель 2



Модель 3

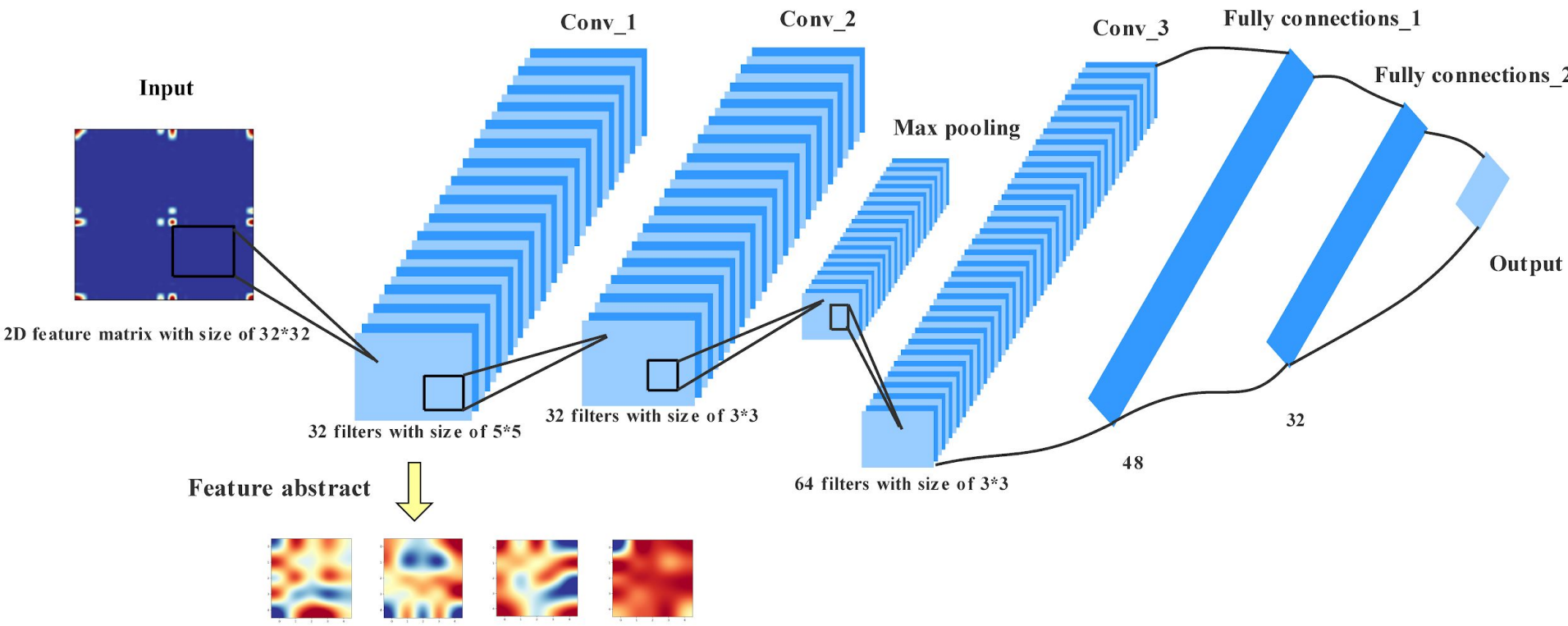


Распределение координат



Более подробное объяснение CNN

Convolutional Neural Network Framework



Как строится модель

```
def convolutional_model():
    # create model
    model = Sequential()
    model.add(Conv2D(4, (5, 5), strides=(1, 1), padding="same", activation='relu', input_shape=(128, 128, 3)))
    # 4 x 128 x 128
    model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2), padding="same"))
    # 4 x 64 x 64

    model.add(Conv2D(8, (2, 2), activation='relu', padding="same"))
    # 8 x 64 x 64
    model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2), padding="same"))
    # 8 x 32 x 32

    model.add(Conv2D(16, (2, 2), activation='relu', padding="same"))
    # 16 x 32 x 32
    model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2), padding="same"))
    # 16 x 16 x 16

    model.add(Flatten())
    model.add(Dense(4096, activation='relu'))
    model.add(Dense(512, activation='relu'))
    model.add(Dense(128, activation='relu'))
    model.add(Dense(32, activation='relu'))
    model.add(Dense(1))

    model.compile(optimizer='adam', loss='mean_squared_error', metrics=['mean_squared_error'])
    return model
```

Нейронная сеть

