

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

Национальный исследовательский ядерный университет «МИФИ»

Преддипломная практика  
на тему:

Применение свёрточных 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.

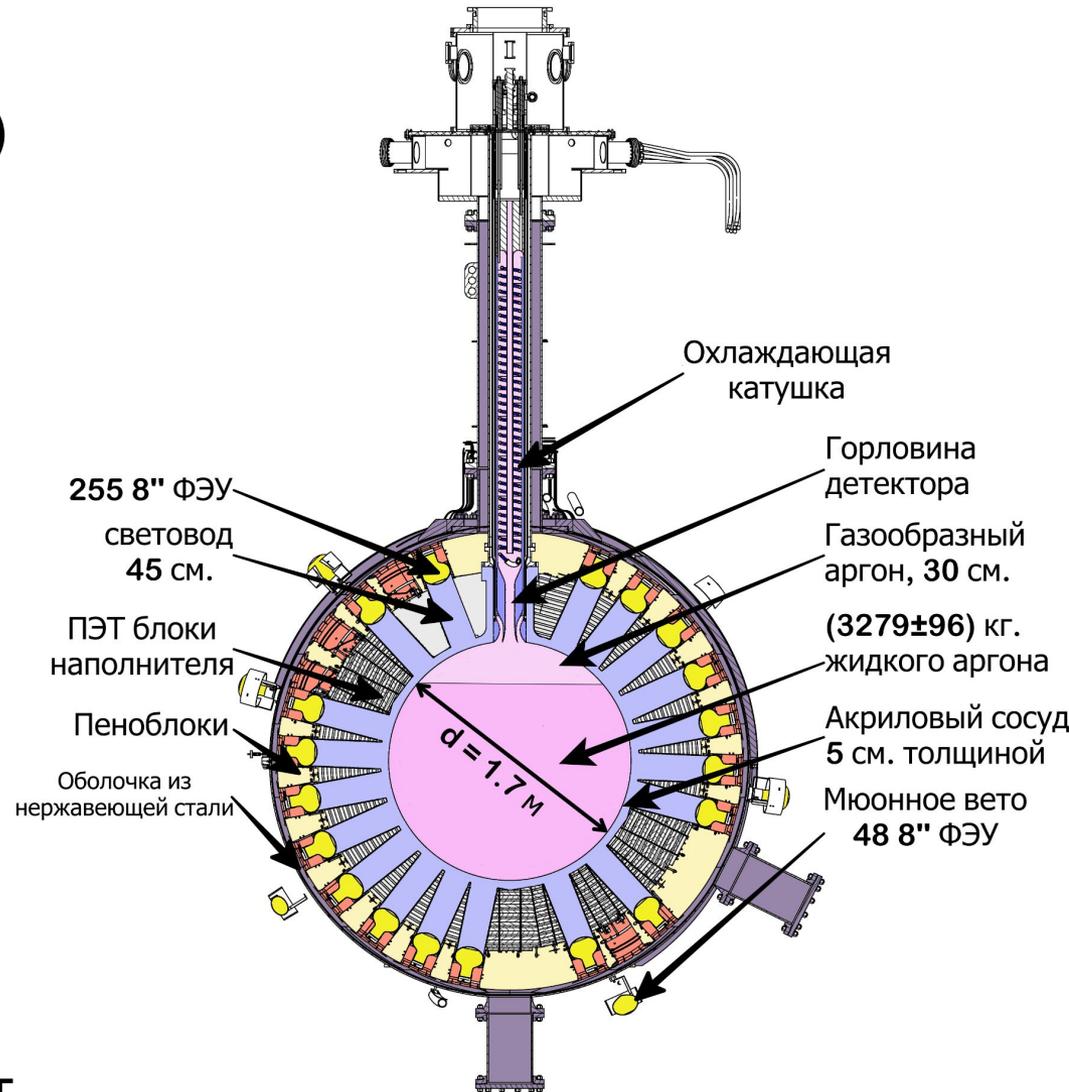
Москва, 2021



# Детектор DEAP-3600

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

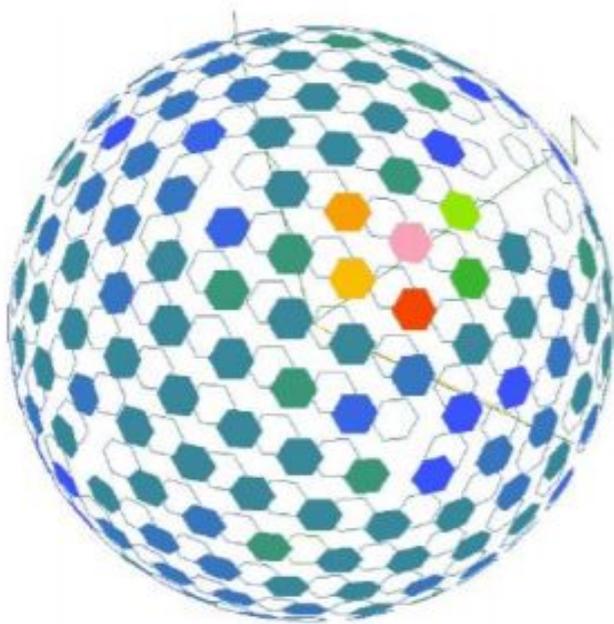
- 2 км. под землей;
- $(3279 \pm 96)$  кг. LAr (84 K) в UVA акриловом сосуде толщиной 5 см. и диаметром 1.7 м.;
- 255 + 48 ФЭУ;
- **Нейтронная и черенковская мюонная защита;**
- Детектор заключен в оболочку из нержавеющей стали и погружен в бассейн 300 т. воды.



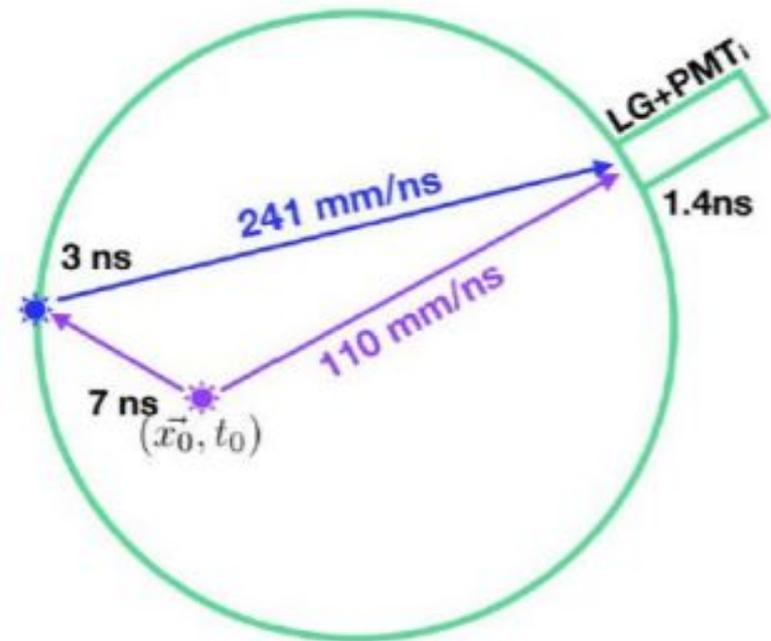
# Имеющиеся алгоритмы

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



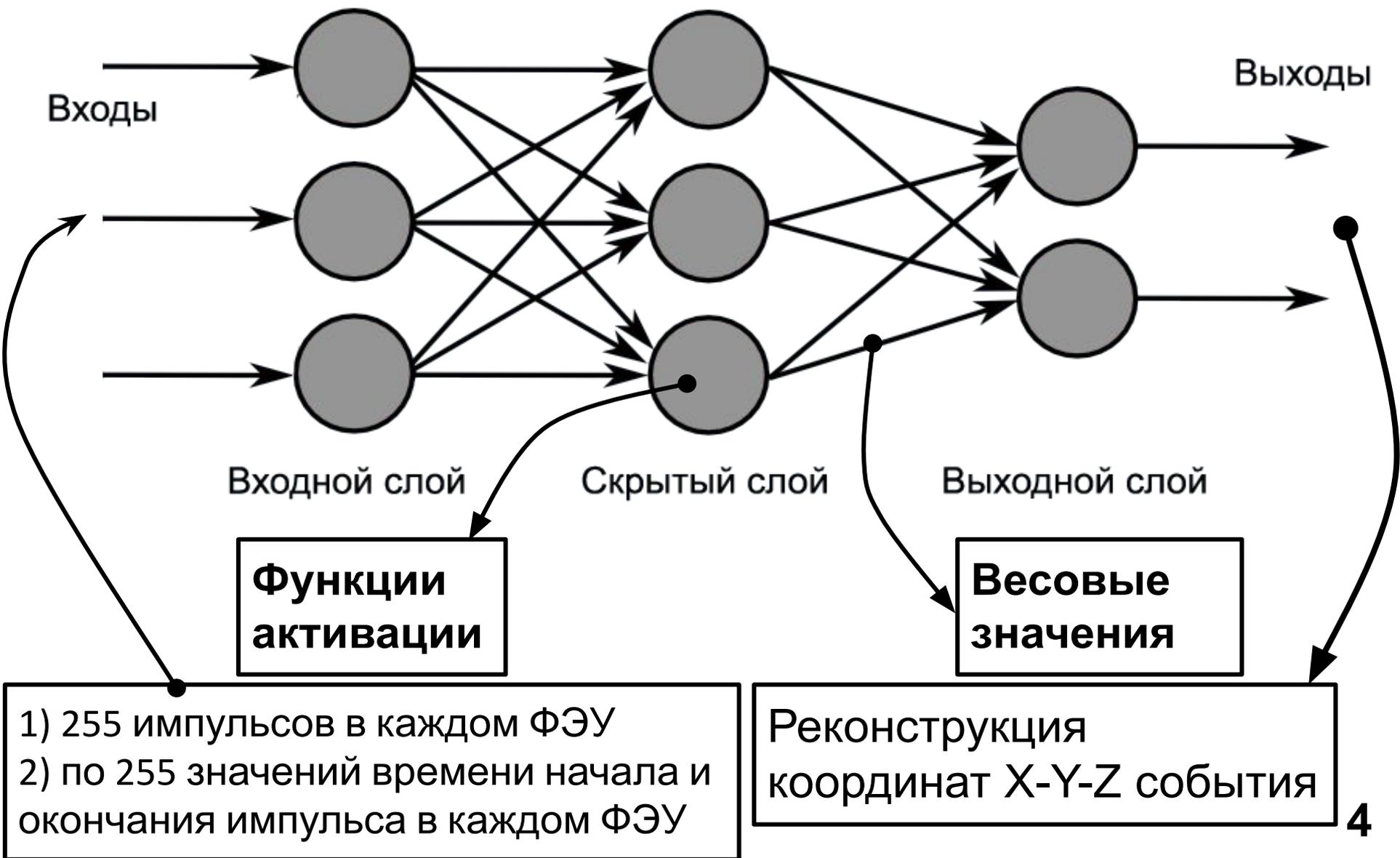
I. После **фидуциализации** ошибка второго алгоритма около **13%**;

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

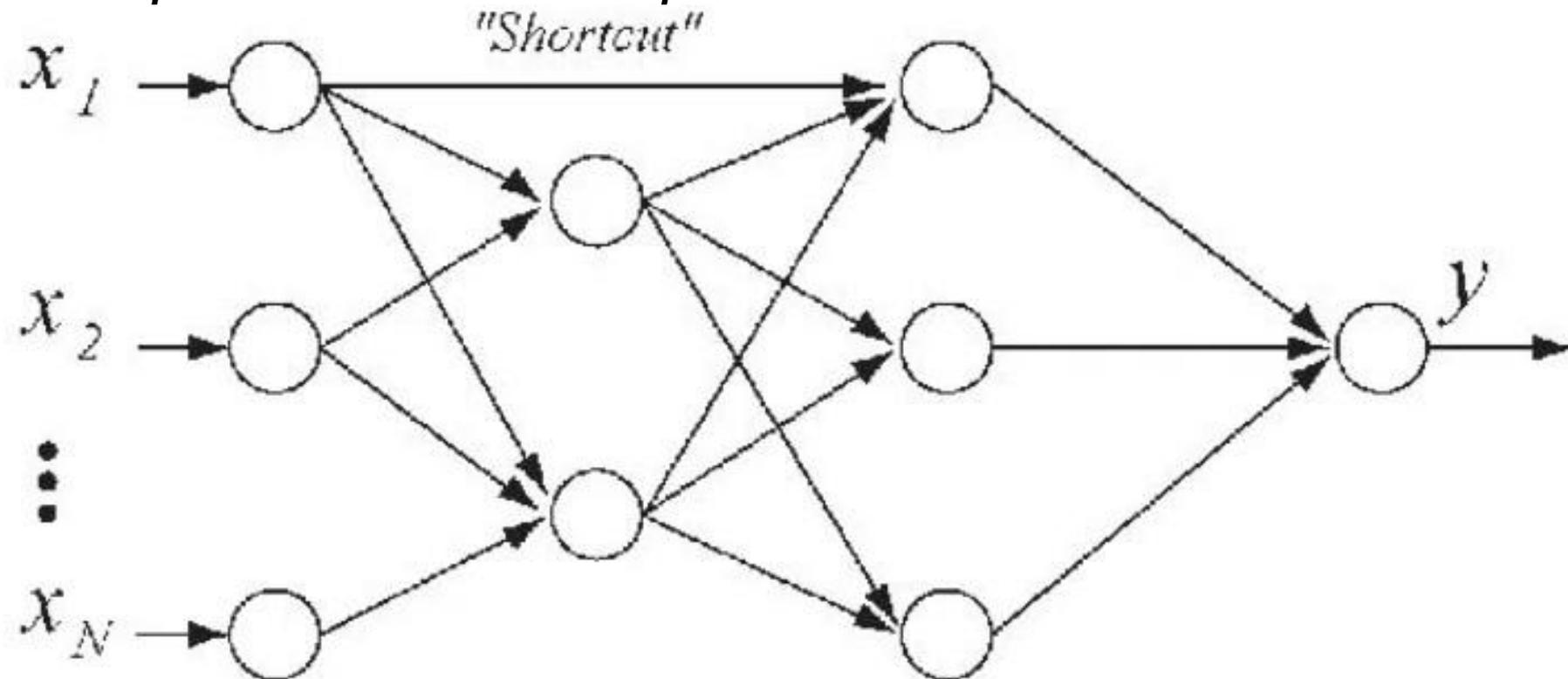
# Используемые алгоритмы

## 1. Полносвязная нейронная сеть



# Используемые алгоритмы

## 2. Нейронная сеть с короткими связями



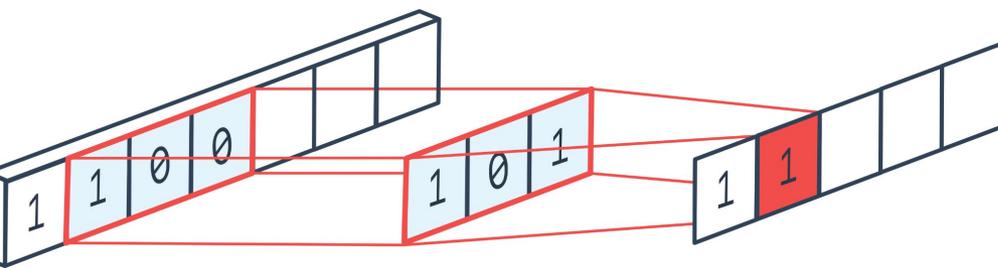
Преимущество:  
Меньше теряется  
первоначальная  
информация о данных

Недостатки:  
1) Необходимость соответствия  
длин векторов начала и конца  
shortcut-связи  
2) Длительность тренировки

# Используемые алгоритмы

## 3. Свёрточная нейронная сеть

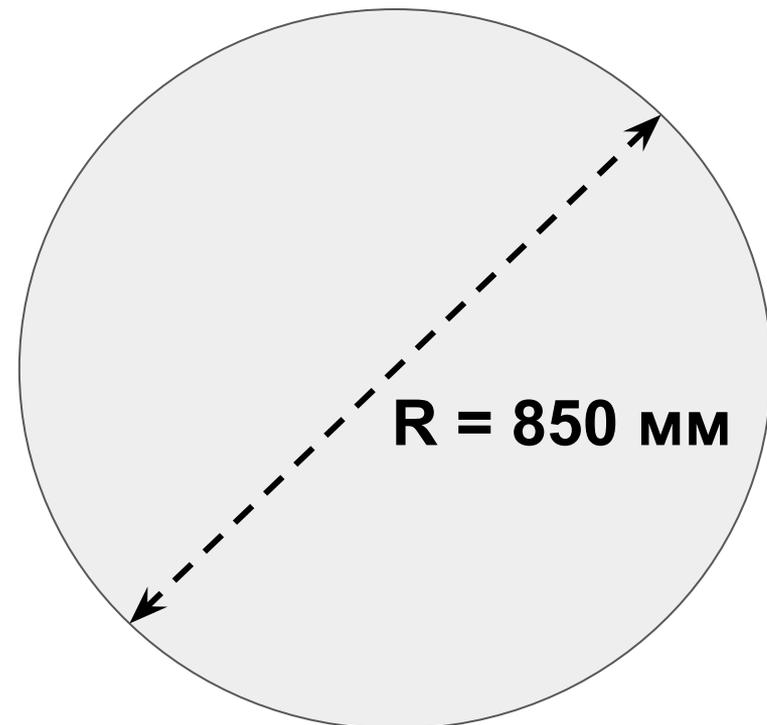
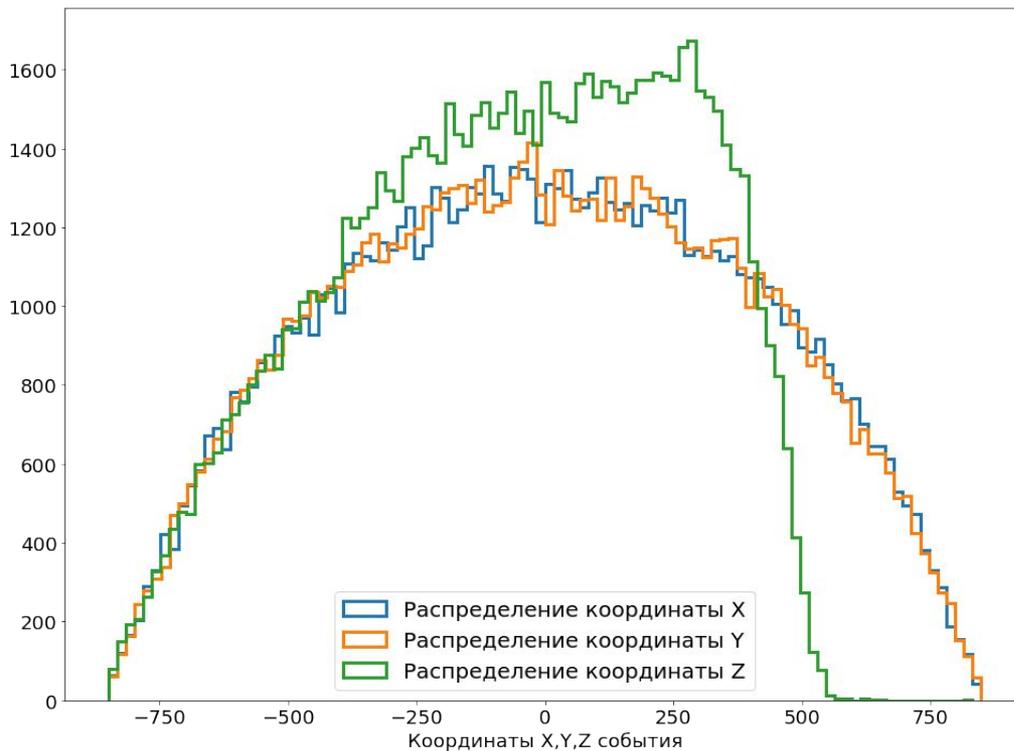
Layer (type)	Output Shape
input_11 (InputLayer)	[(None, 765, 1)]
conv1d (Conv1D)	(None, 383, 16)
conv1d_1 (Conv1D)	(None, 192, 32)
conv1d_2 (Conv1D)	(None, 96, 64)
conv1d_3 (Conv1D)	(None, 96, 128)
conv1d_4 (Conv1D)	(None, 96, 256)
flatten (Flatten)	(None, 24576)
dense_55 (Dense)	(None, 255)
dense_56 (Dense)	(None, 3)



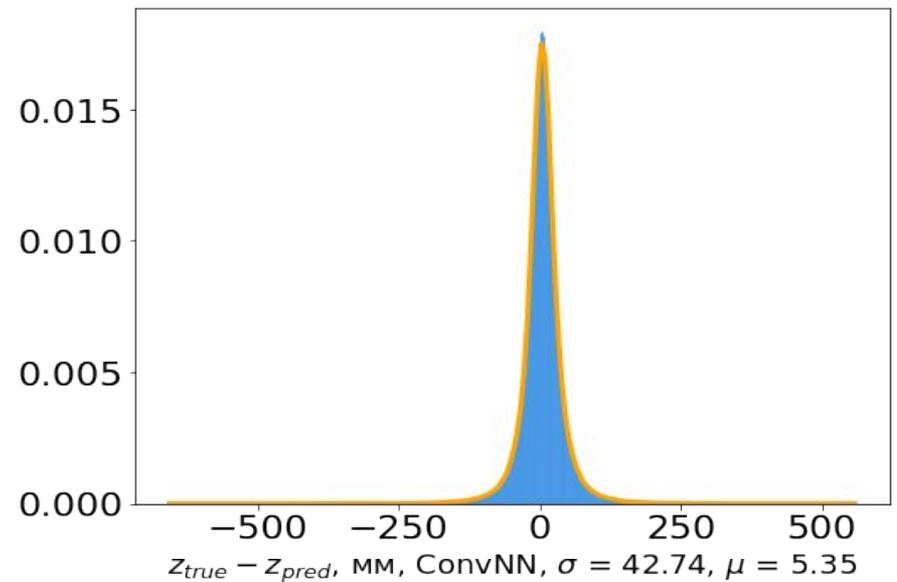
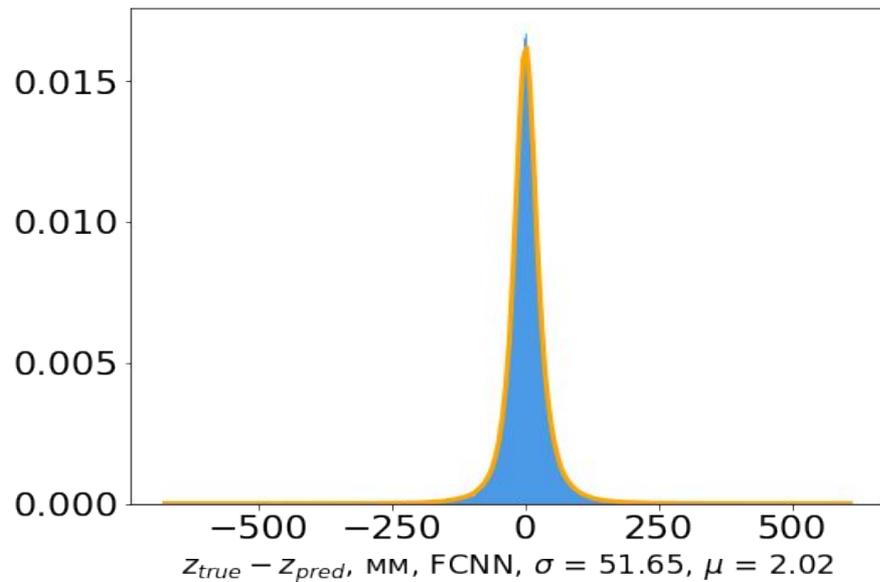
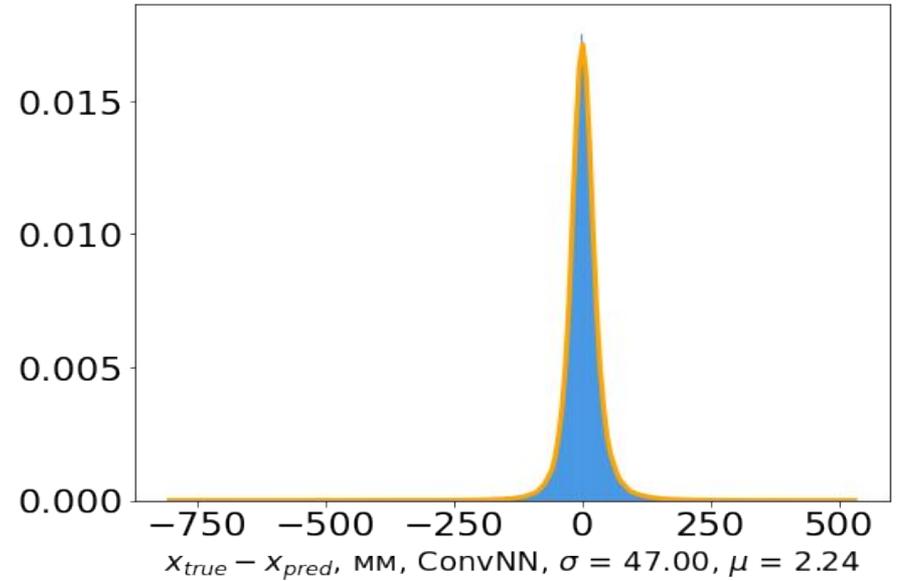
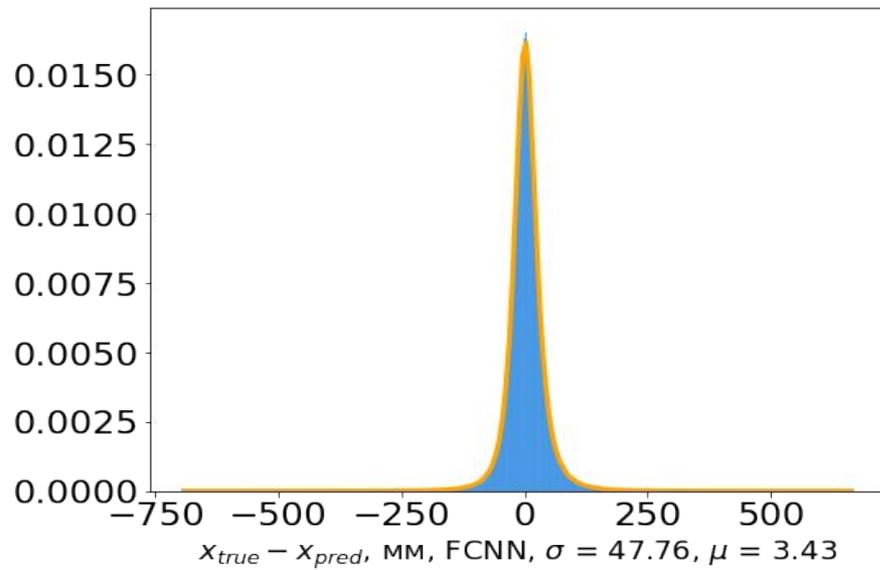
- 1) Слой **Conv1D** - свёртка входного вектора с “ядром”
- 2) **flatten** - разглаживание многомерной матрицы в одномерный вектор
- 3) **Dense** - полносвязная нейронная сеть для окончательного определения координаты

# Рассматриваемые области детектора

Ограничение	[0; 430] мм	[0; 630] мм	[0; 720] мм	[0; 850] мм
Количество событий, тыс.	107	300	426	677
Ограничение	[200; 430] мм	[430; 630] мм	[630; 720] мм	[720; 850] мм
Количество событий, тыс.	96	193	126	251

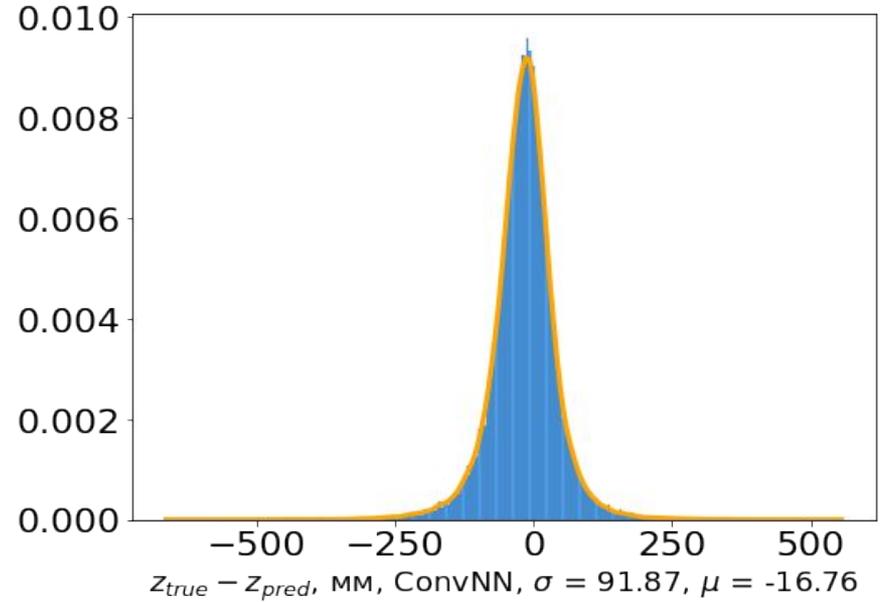
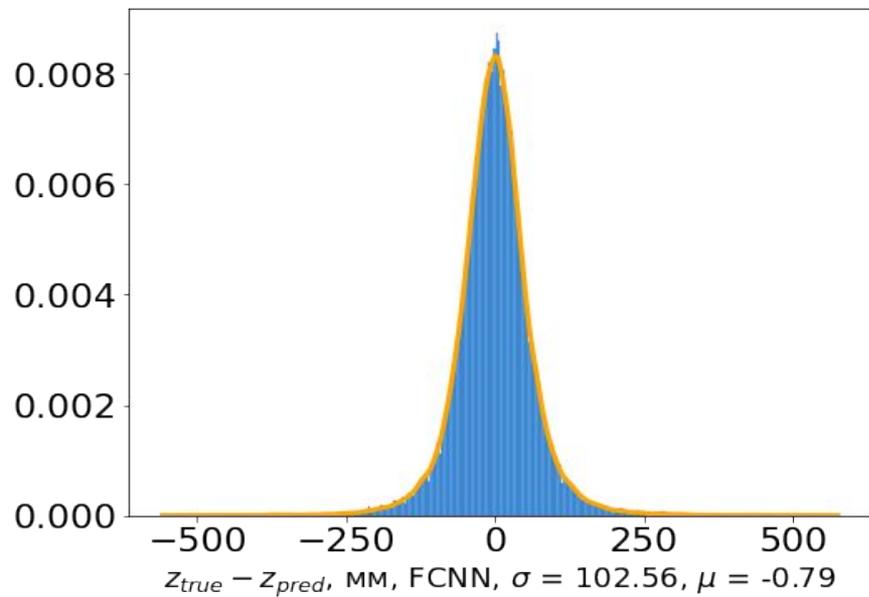
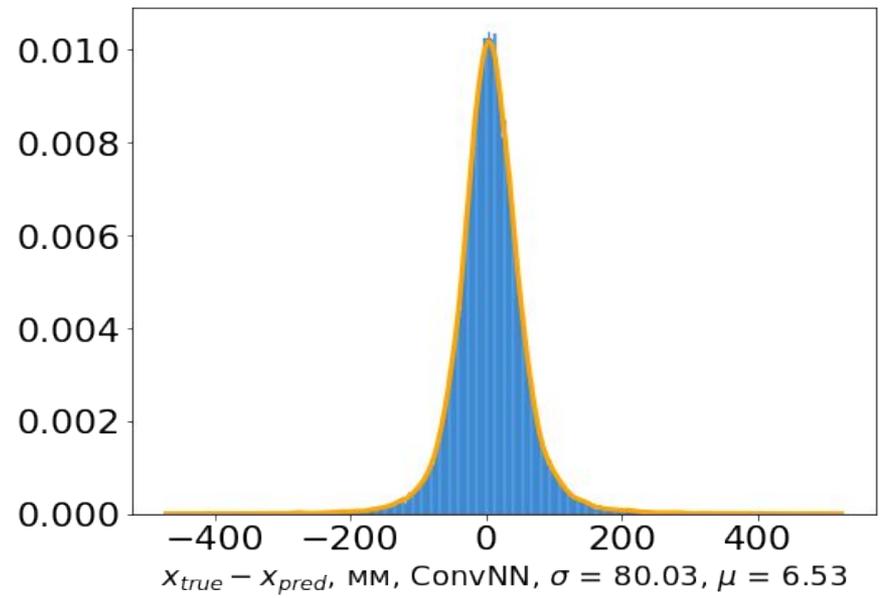
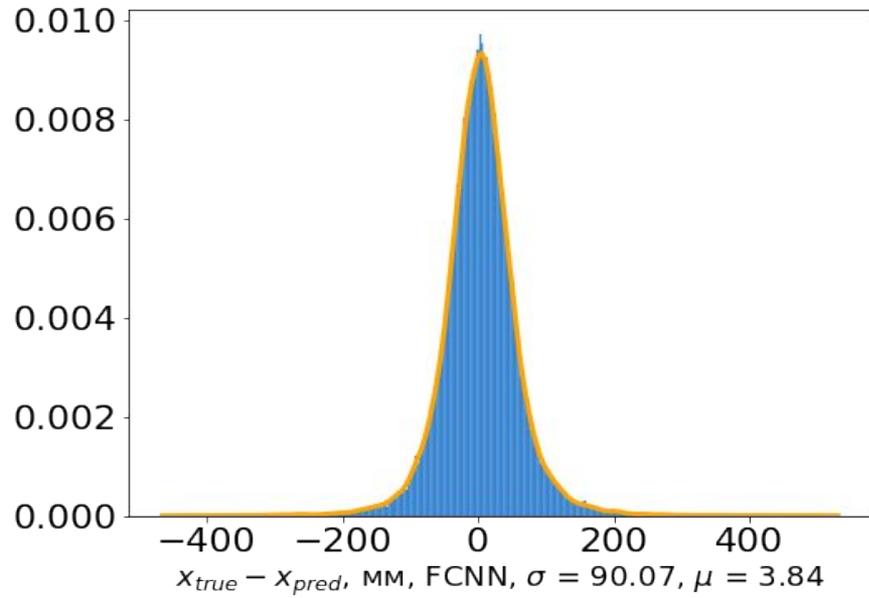


# Полученные результаты, $R < 850$ мм



$\sigma \equiv$  FWHM

# Полученные результаты, $R < 430$ мм



$\sigma \equiv$  FWHM

## Полученные результаты

	Точность	$X_{FWHM}$ , мм	$Y_{FWHM}$ , мм	$Z_{FWHM}$ , мм
$R < 850$	$0.959 \pm 0.003$	$47 \pm 4$	$49 \pm 3$	$51 \pm 7$
$R < 720$	$0.944 \pm 0.001$	$64 \pm 3$	$64 \pm 7$	$66 \pm 2$
$R < 630$	$0.933 \pm 0.002$	$75 \pm 3$	$76 \pm 6$	$78 \pm 2$
$R < 430$	$0.899 \pm 0.007$	$91 \pm 7$	$92 \pm 8$	$105 \pm 15$
$720 < R < 850$	$0.983 \pm 0.003$	$41 \pm 9$	$42 \pm 8$	$42 \pm 9$
$630 < R < 720$	$0.966 \pm 0.005$	$55 \pm 5$	$55 \pm 8$	$42 \pm 5$
$430 < R < 630$	$0.952 \pm 0.002$	$79 \pm 10$	$77 \pm 8$	$58 \pm 2$
$200 < R < 430$	$0.913 \pm 0.004$	$90 \pm 5$	$88 \pm 3$	$98 \pm 7$

### Результаты существующих алгоритмов:

- 1) При  $R < 630$  данные и моделирование 1го алгоритма соотносятся с точностью до 13%
- 2) Оба алгоритма согласуются по Z-координате с точностью до 35 мм. для 50% событий
- 3) Плохая работа алгоритмов во всем объеме детектора

$$*FWHM = 2.355\sigma$$
$$S(FWHM) = 76\%$$

## *Заключение*

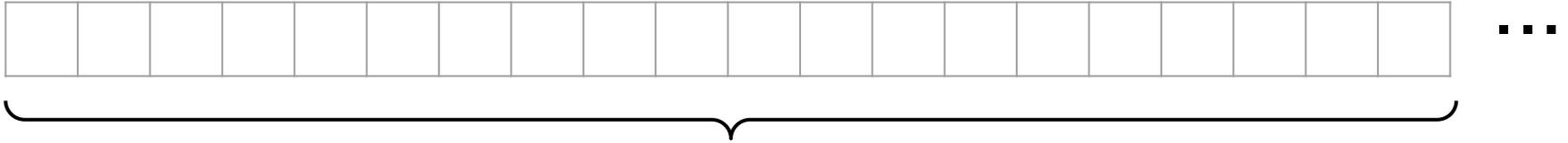
- 1) **Хуже всего** реконструируются события в центральной области детектора ( $R < 430$ ) в связи с тем, что значения импульсов, пришедших в каждый ФЭУ практически не отличается от события к событию;
- 2) **Лучше всего** алгоритм работает на внешнем слое детектора ( $720 < R < 850$ ) выдавая ошибку около 40 мм. и во всём объеме детектора ( $R < 850$ ) со средней ошибкой около 50 мм.
- 3) **Новый алгоритм по эффективности не уступает существующим;**

## *Планы*

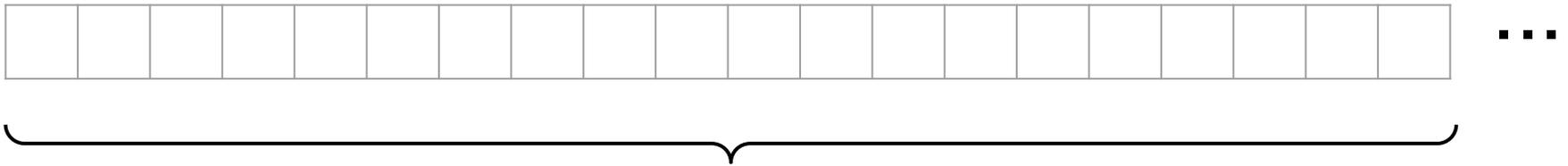
- 1) **Увеличить** статистику / Использовать **новую версию** Монте-Карло;
- 2) Сконструировать новые переменные (prompt, late signals);
- 3) Улучшать сеть далее, использовать новые методики (stacking).

# Дополнительные слайды

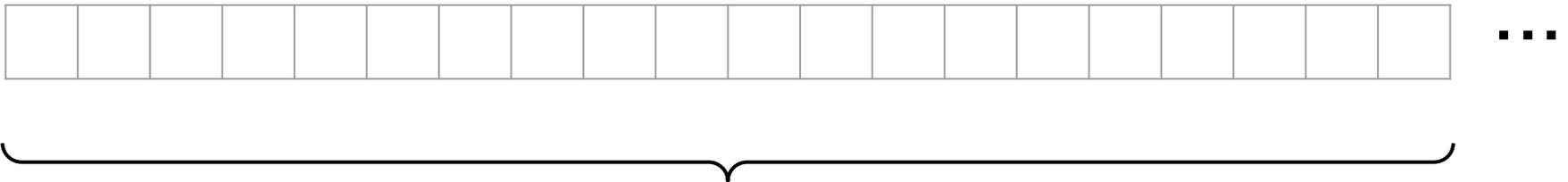
# Данные



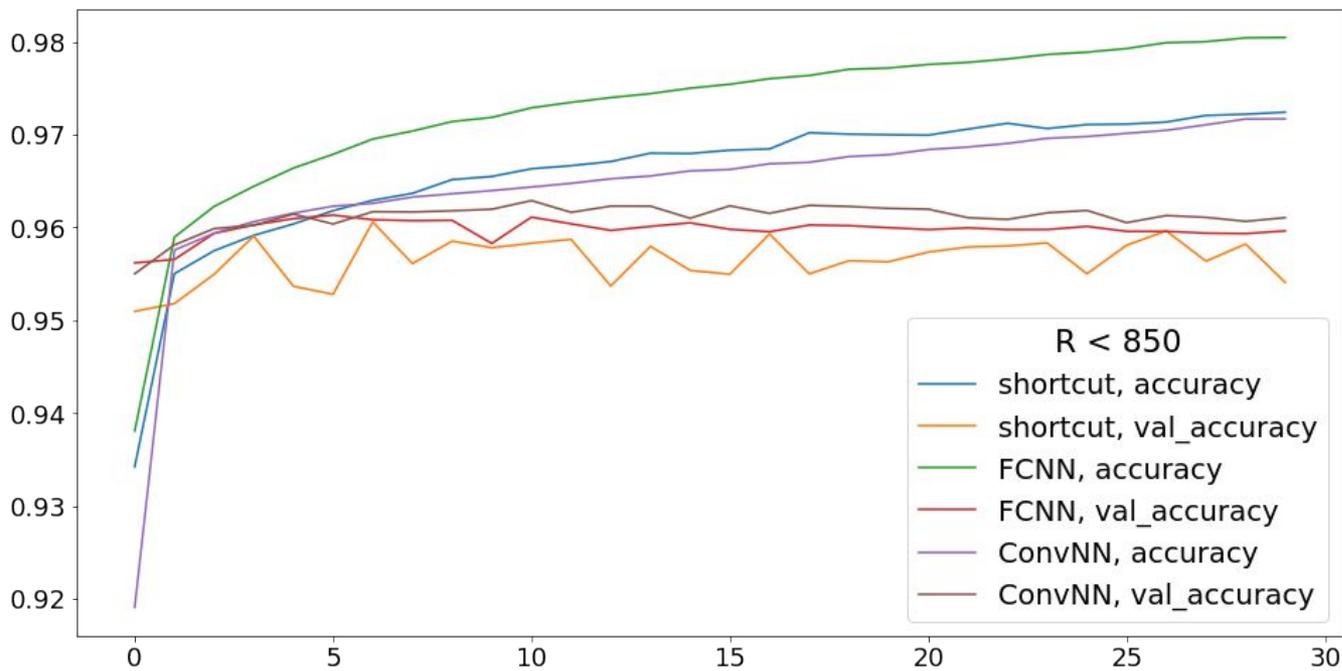
**255 значений импульсов в каждом ФЭУ**



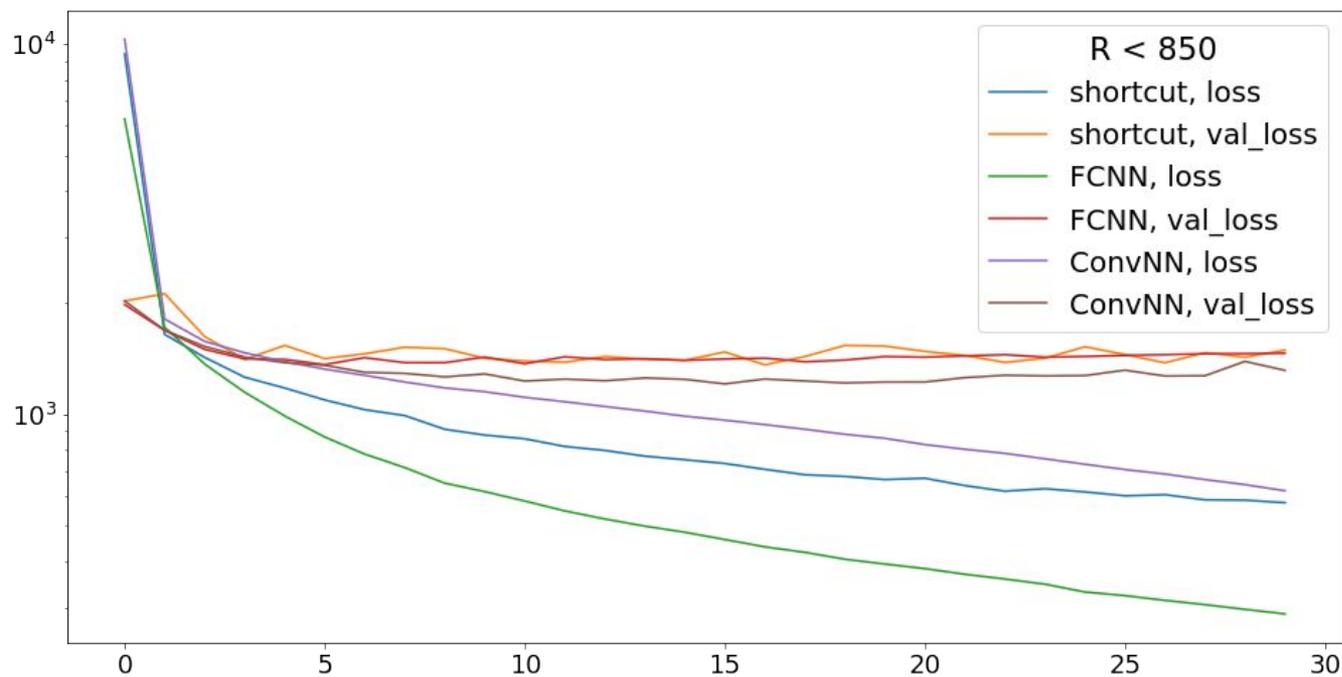
**255 значений времени начала принятия сигнала в ФЭУ**

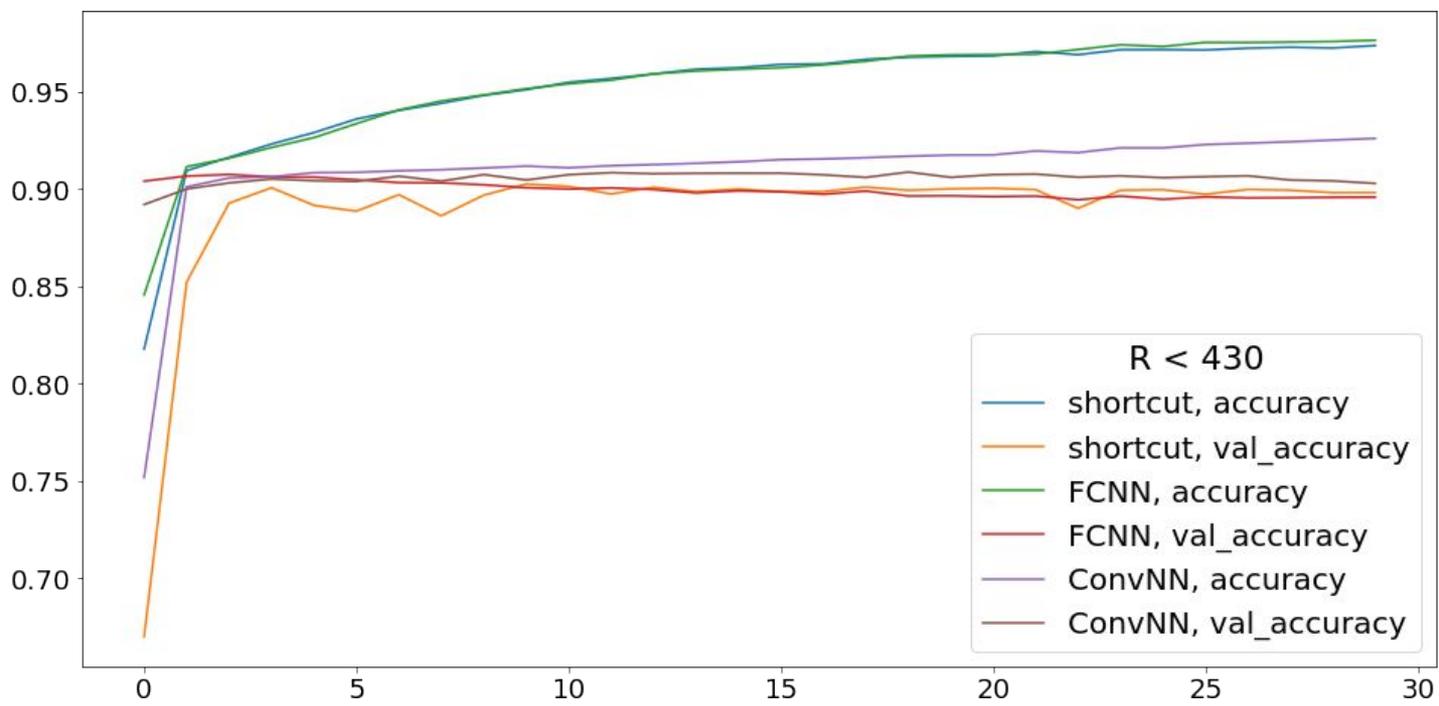


**255 значений времени окончания принятия сигнала в ФЭУ**

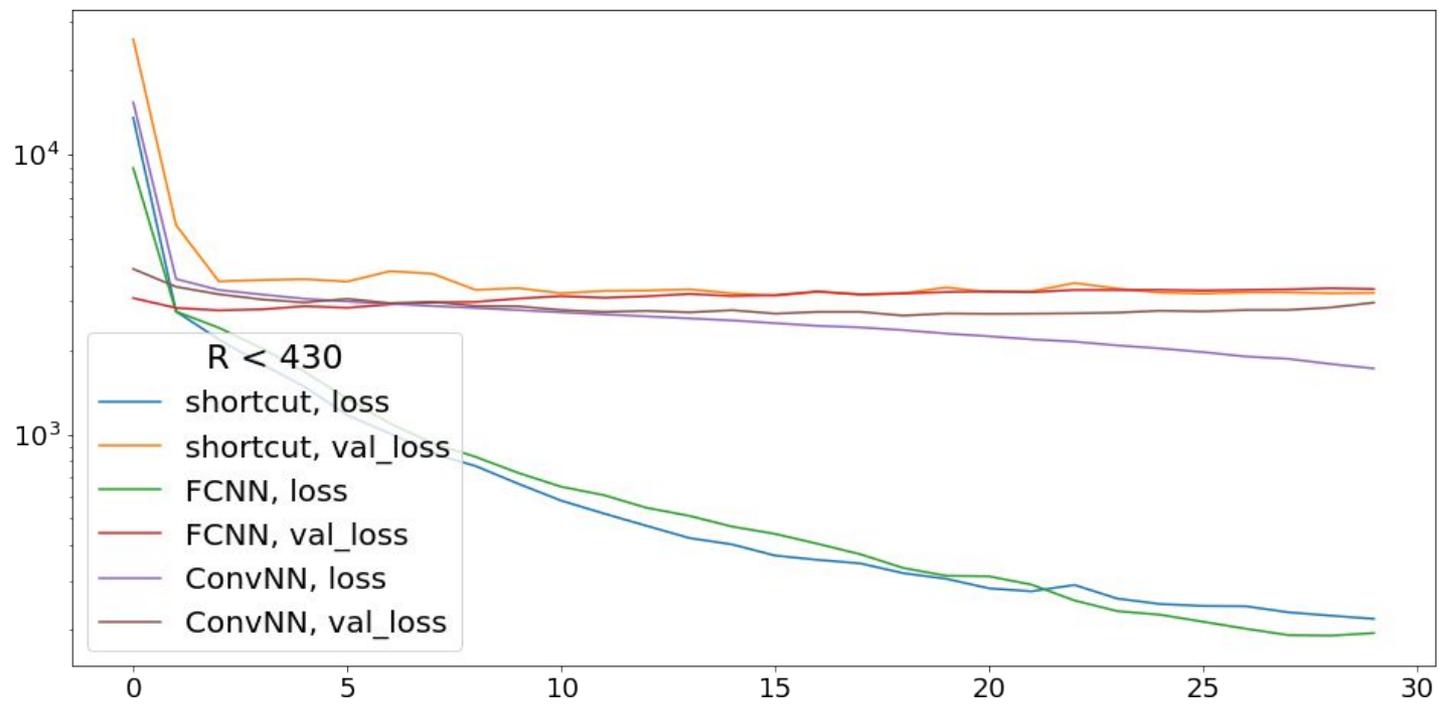


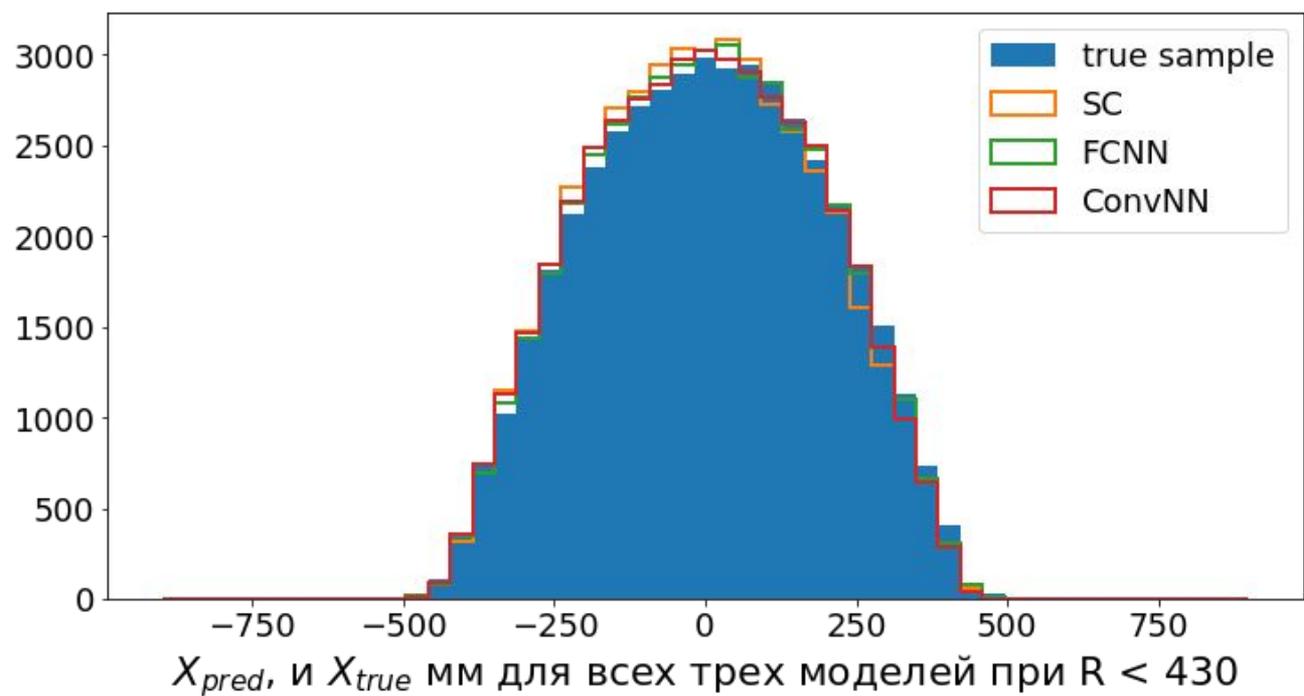
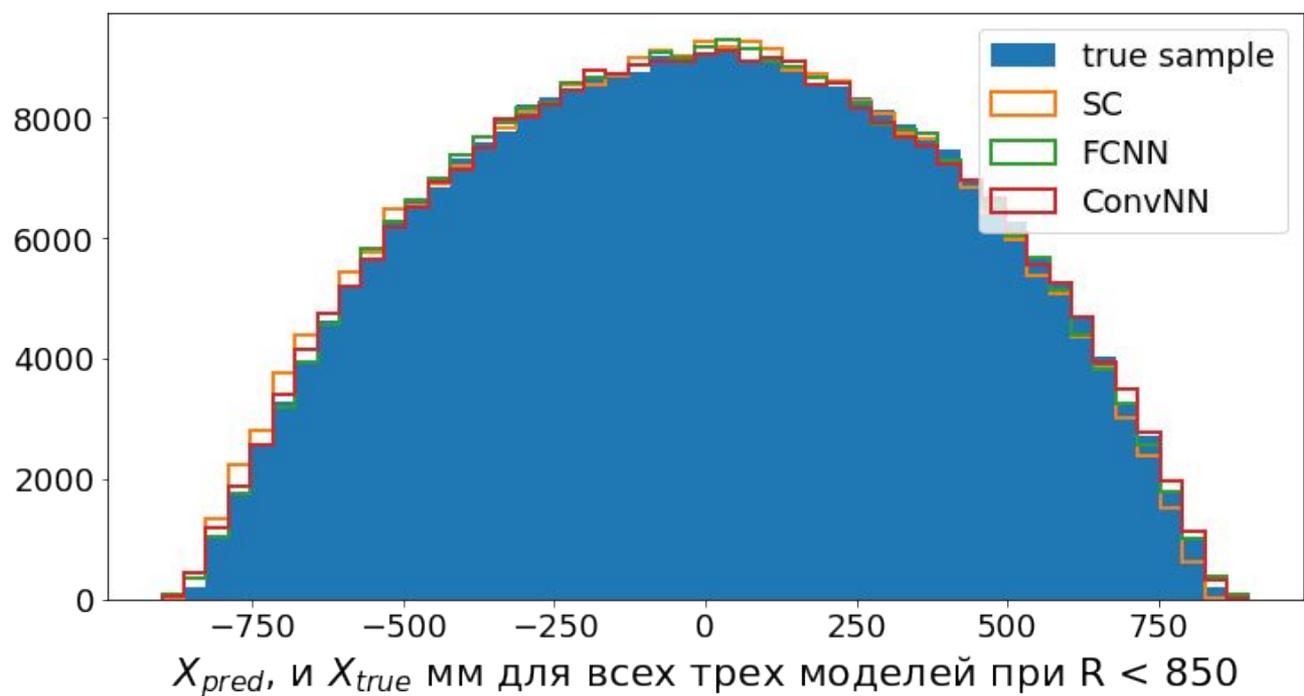
**R < 850**



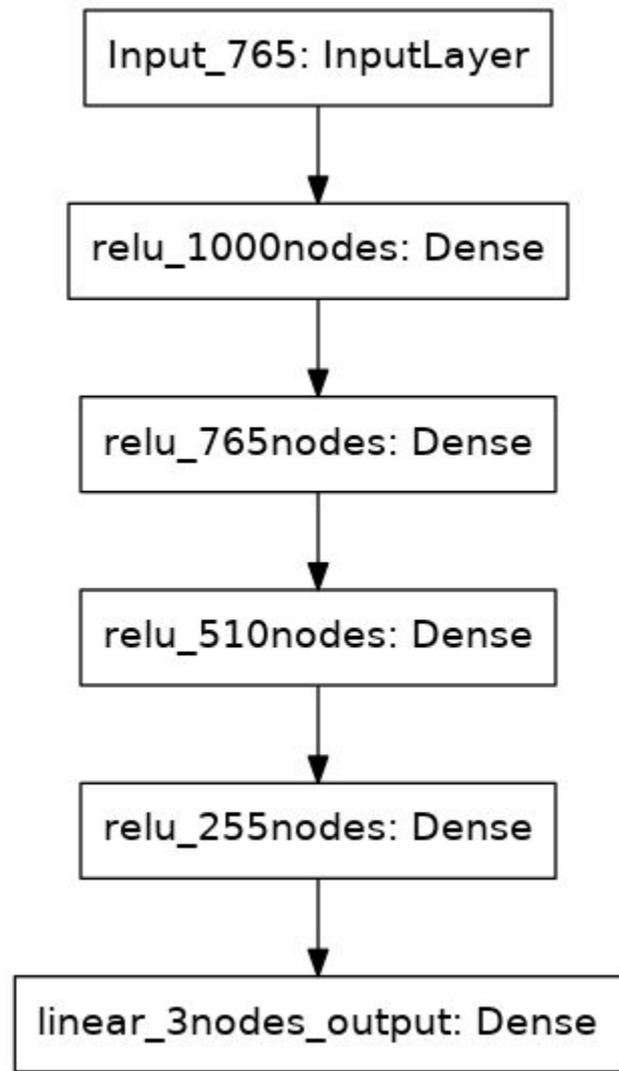


**R < 430**





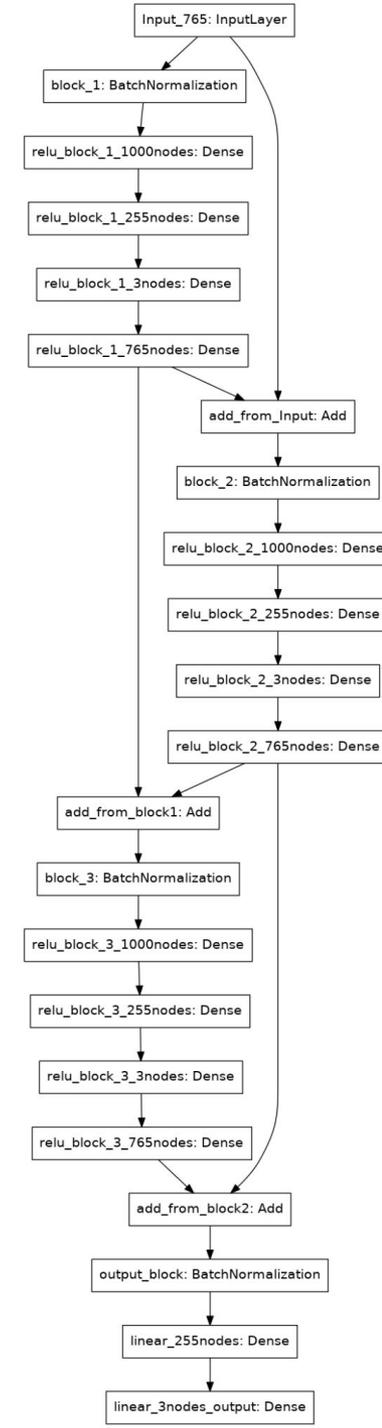
# FCNN

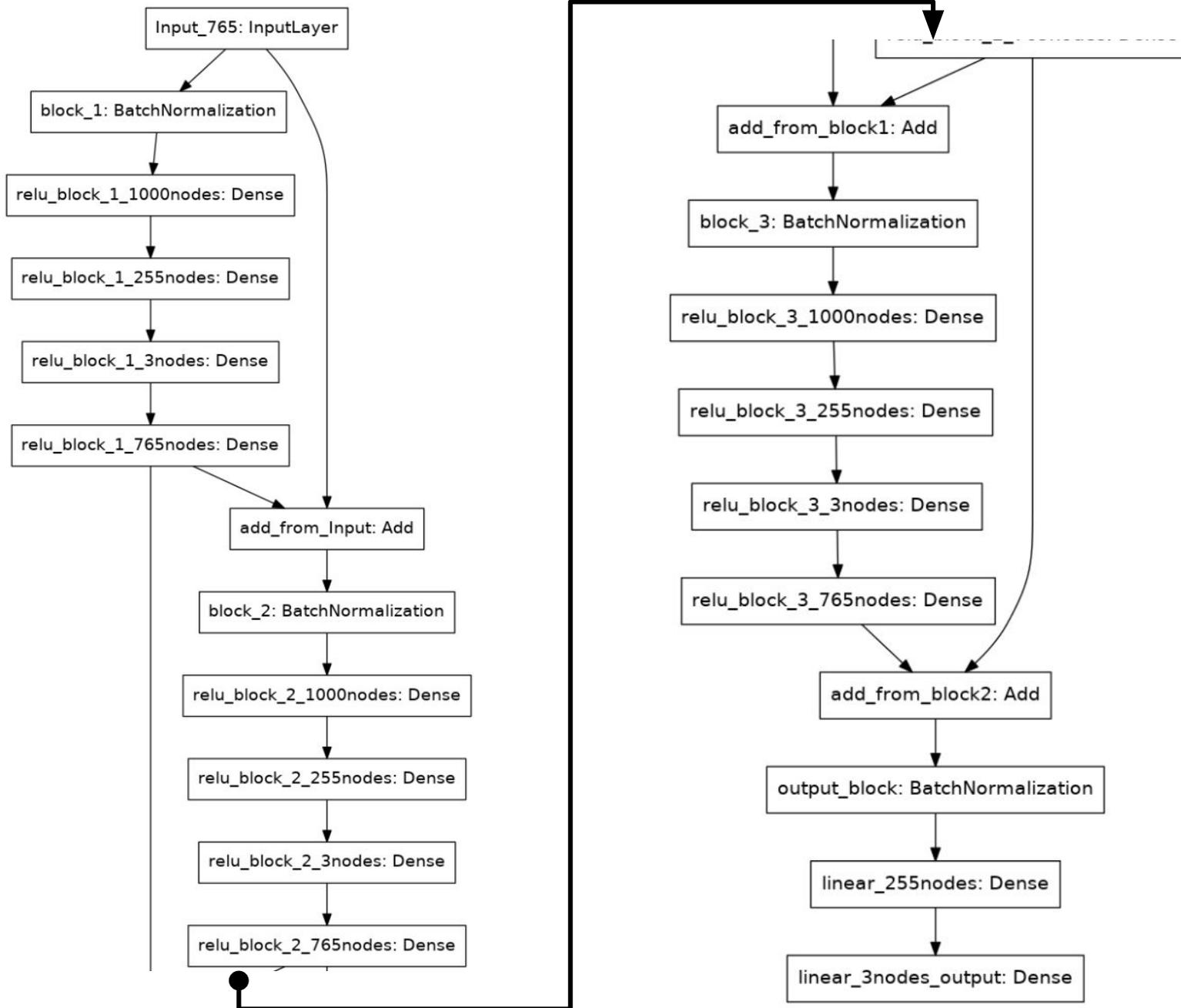


Layer (type)	Output Shape
Input_765 (InputLayer)	[(None, 765)]
1000nodes (Dense)	(None, 1000)
765nodes (Dense)	(None, 765)
510nodes (Dense)	(None, 510)
255nodes (Dense)	(None, 255)
3nodes_output (Dense)	(None, 3)

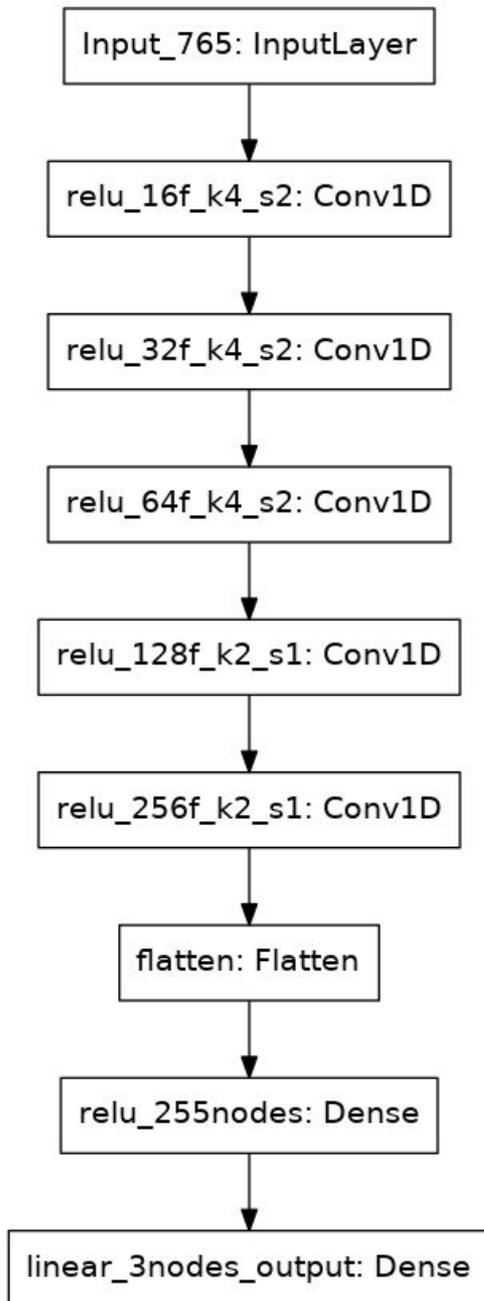
# Shortcut

Layer (type)	Output Shape
Input_765 (InputLayer)	[(None, 765)]
block_1 (BatchNormalization)	(None, 765)
block_1_1000nodes (Dense)	(None, 1000)
block_1_255nodes (Dense)	(None, 255)
block_1_3nodes (Dense)	(None, 3)
block_1_765nodes (Dense)	(None, 765)
add_from_Input (Add)	(None, 765)
block_2 (BatchNormalization)	(None, 765)
block_2_1000nodes (Dense)	(None, 1000)
block_2_255nodes (Dense)	(None, 255)
block_2_3nodes (Dense)	(None, 3)
block_2_765nodes (Dense)	(None, 765)
add_from_block1 (Add)	(None, 765)
block_3 (BatchNormalization)	(None, 765)
block_3_1000nodes (Dense)	(None, 1000)
block_3_255nodes (Dense)	(None, 255)
block_3_3nodes (Dense)	(None, 3)
block_3_765nodes (Dense)	(None, 765)
add_from_block2 (Add)	(None, 765)
output_block (BatchNormalizatio	(None, 765)
255nodes (Dense)	(None, 255)
3nodes_output (Dense)	(None, 3)





# ConvNN



Layer (type)	Output Shape
input_11 (InputLayer)	[(None, 765, 1)]
conv1d (Conv1D)	(None, 383, 16)
conv1d_1 (Conv1D)	(None, 192, 32)
conv1d_2 (Conv1D)	(None, 96, 64)
conv1d_3 (Conv1D)	(None, 96, 128)
conv1d_4 (Conv1D)	(None, 96, 256)
flatten (Flatten)	(None, 24576)
dense_55 (Dense)	(None, 255)
dense_56 (Dense)	(None, 3)