# Cluster Simulation via Neural Network

YiZhou Zhang, Tao Lin, WeiDong Li
zhangyz@ihep.ac.cn

30th Aug 2023

# Outline

# 1 Background

Cluster Reconstruction via Neural Network

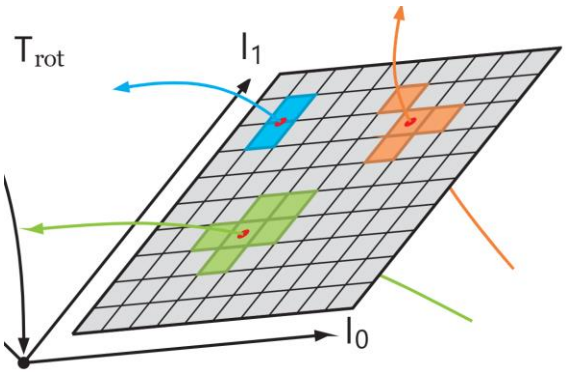
➢ **Requirements:**

- 现有的模拟算法并不能模拟出粒子穿过感应器产生的感应簇。
- Geant4模拟出的结果，只有模拟粒子穿过感应器的一个感应点。

➢ **Purpose：**

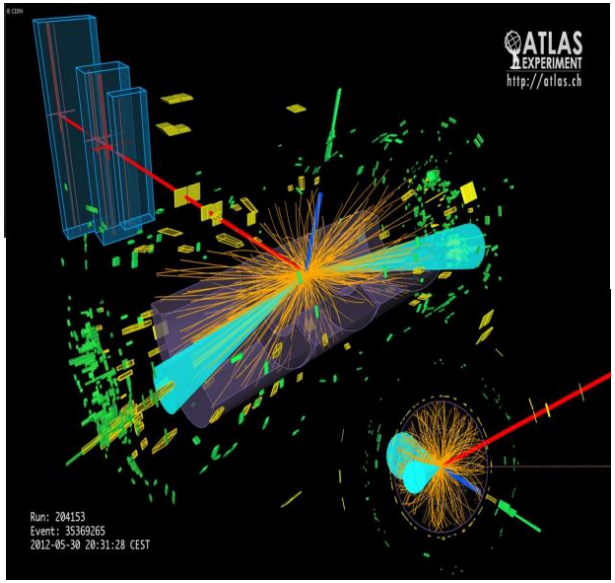
- 根据粒子穿过感应器的真实响应感应簇，通过机器学习方法，训练出能够模拟产生感应簇的模型。
- 使用TestBeam的数据训练模型。
- 使用生成对抗网络。

感应簇




Reference: Acts Common Tracking Software
https://acts.readthedocs.io/

Geant4模拟




Reference: R.M. Bianchi, “Event display of a 2-tau candidate in the ATLAS detector”

# Outline

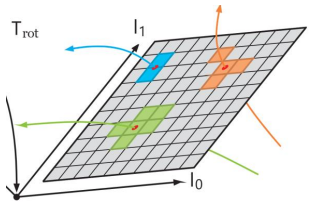
# Overview



EDM4hep::
TrackerHit

CEPCSW

EDM4hep::
RawTimeSeries

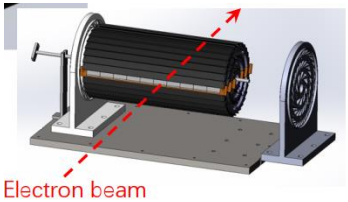



Network Model

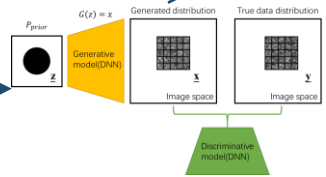
.pt

.so

Wrapped Functions




TestBeam Data




Neural Network

PyTorch

Wrapper

C++

Load_model()

predict()

C++

PyTorch

libtorch

2

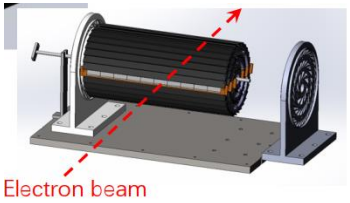
# Design

Cluster Reconstruction via Neural Network

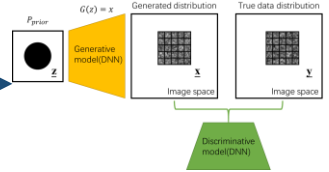
# Data Prepocessing

```
root [3] TimeInfo->Scan()
*************************************************************************************************************
*    Row   * Instance * EventNum. *  timeFPGA *  timeChip *   planeID *       row *       col * clusterID * clusterNu *
*************************************************************************************************************
*      0 *        0 *         1 *       246 *        86 *         0 *        43 *        84 *         0 *         1 *
*      0 *        1 *         1 *       246 *        86 *         0 *        42 *        84 *         0 *         1 *
*      0 *        2 *         1 *       246 *        87 *         1 *        82 *        87 *         0 *         1 *
*      0 *        3 *         1 *       246 *        87 *         1 *        82 *        88 *         0 *         1 *
*      0 *        4 *         1 *       246 *        87 *         1 *        81 *        87 *         0 *         1 *
*      0 *        5 *         1 *       246 *        87 *         1 *        81 *        88 *         0 *         1 *
*      0 *        6 *         1 *       246 *         3 *         2 *        56 *       104 *         0 *         1 *
*      0 *        7 *         1 *       246 *         5 *         3 *        92 *       115 *         0 *         1 *
*      0 *        8 *         1 *       246 *         5 *         3 *        92 *       116 *         0 *         1 *
*      0 *        9 *         1 *       246 *       191 *         4 *        87 *       150 *         0 *         1 *
*      0 *       10 *         1 *       246 *       194 *         4 *        87 *       149 *         0 *         1 *
*      0 *       11 *         1 *       246 *        68 *         5 *        65 *       179 *         0 *         1 *
*      0 *       12 *         1 *       246 *        68 *         5 *        65 *       180 *         0 *         1 *
*      0 *       13 *         1 *       246 *        68 *         5 *        66 *       179 *         0 *         1 *
*      0 *       14 *         1 *       246 *        68 *         5 *        66 *       180 *         0 *         1 *
*      0 *       15 *         1 *       246 *        67 *         6 *        97 *       176 *         0 *         1 *
*      0 *       16 *         1 *       246 *        68 *         6 *        97 *       175 *         0 *         1 *
*      0 *       17 *         1 *       246 *        75 *         6 *        98 *       176 *         0 *         1 *
*      0 *       18 *         1 *       246 *       123 *         7 *       288 *       167 *         1 *         2 *
*      0 *       19 *         1 *       246 *         3 *         7 *        65 *       199 *         0 *         2 *
*      0 *       20 *         1 *       246 *         3 *         7 *        65 *       198 *         0 *         2 *
*      0 *       21 *         1 *       246 *       125 *         8 *       502 *         4 *         2 *         3 *
*      0 *       22 *         1 *       246 *         3 *         8 *       102 *       217 *         0 *         3 *
*      0 *       23 *         1 *       246 *         3 *         8 *       102 *       216 *         0 *         3 *
*      0 *       24 *         1 *       246 *         4 *         8 *       119 *       192 *         1 *         3 *
```

Test Beam Data




TestBeam Data




Neural Network

PyTorch

Train the NN using the data

.npy

1024 pixel

512 pixel


The cluster of one hit at the plane.

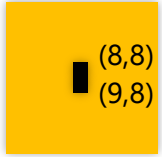
localize

(-0.5,0)
(0.5,0)

round

Relative Coordinates of Geometric Centers

16 pixel



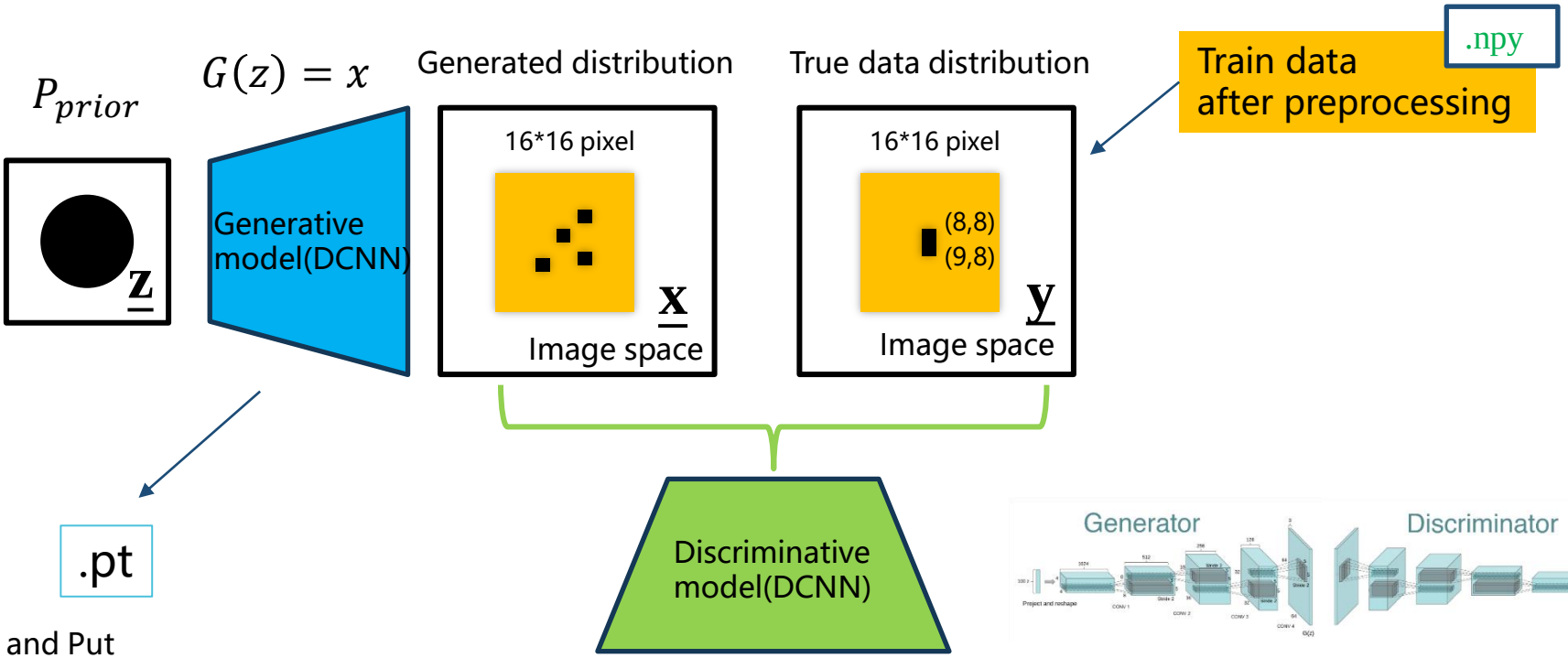

16 pixel

Put it at a 16*16 pic as the train data

# 2 Design

Cluster Reconstruction via Neural Network

## GAN



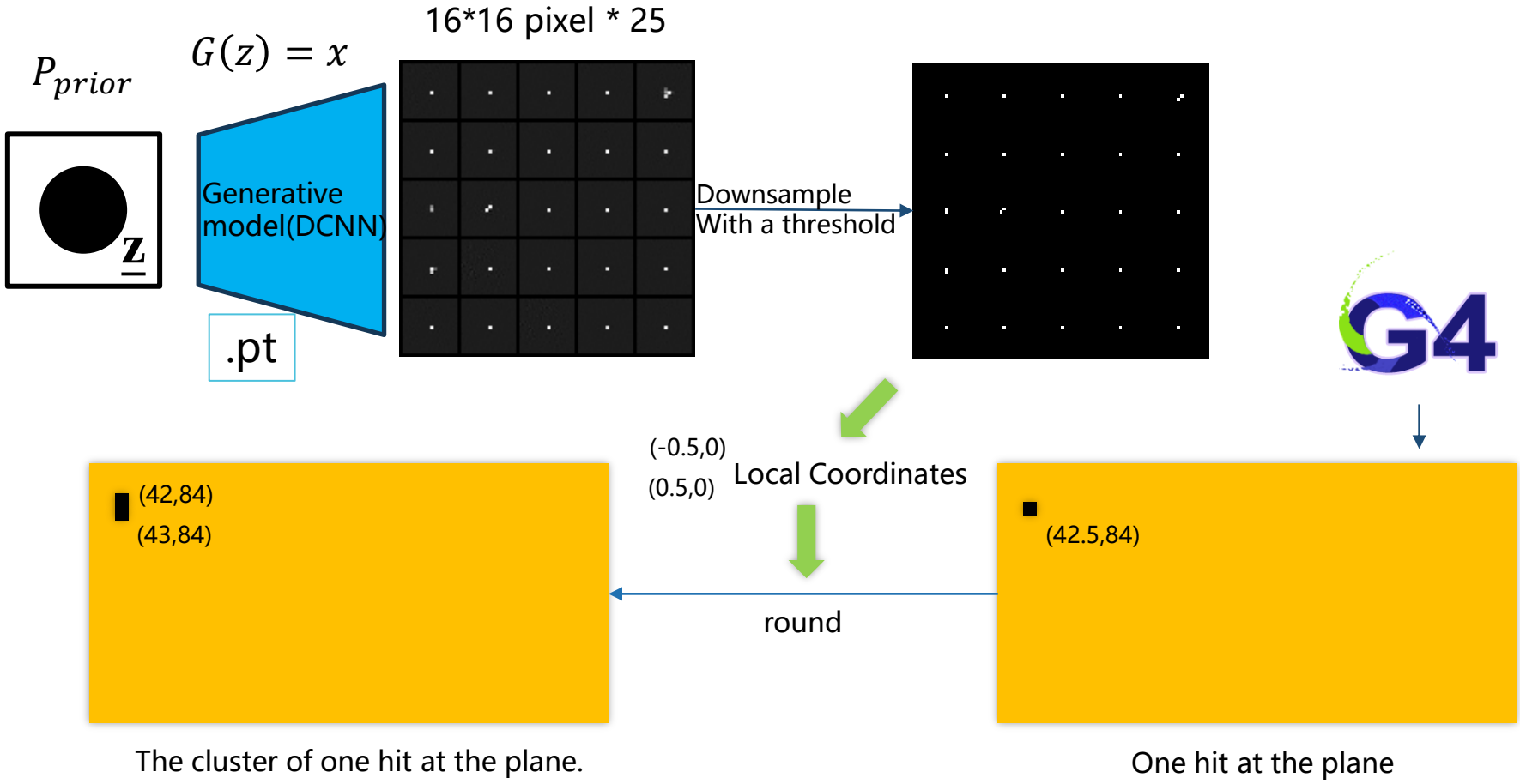

Save and Put
Generative Model into CEPCSW

We use a standard DCGAN model

# 2 Design

Cluster Reconstruction via Neural Network

GAN



# Outline

# 3 Working progress

Cluster Reconstruction via Neural Network

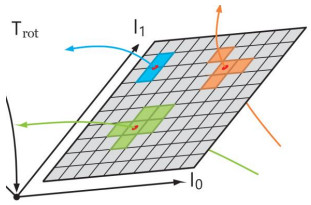
## Overview

EDM4hep:: TrackerHit

TBD

CEPCSW

EDM4hep:: RawTimeSeries

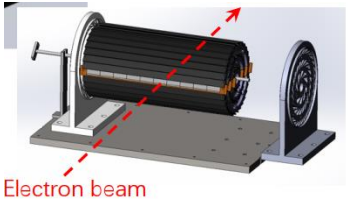



Network Model

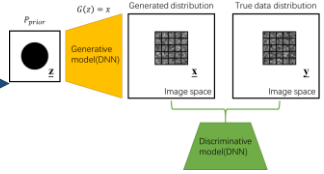
.pt

.so

Wrapped Functions




TestBeam Data




Neural Network

PyTorch

Wrapper

C++

Load_model()

predict()

C++ PyTorch

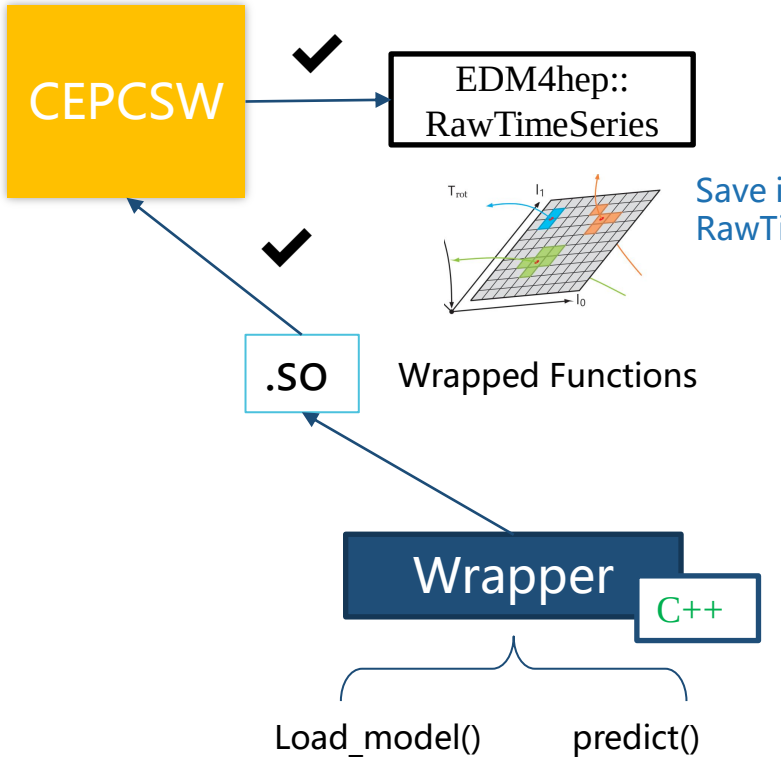
libtorch

Need to be improved!

3

# Working progress

Cluster Reconstruction via Neural Network




## Predict in CEPCSW

Run prediction

```cpp
StatusCode VXDPredict::execute()
{
    std::vector<long> nodes = predictor->run();

    auto hits = m_hitCol.createAndPut();
    for (int i = 0; i < nodes.size(); i += 3){
        auto hit = hits->create();
        // planeID:8,row:12,col:12
        dd4hep::rec::CellID ncellid;
        m_decoder.set(ncellid, "row", nodes[i+1]);
        m_decoder.set(ncellid, "col", nodes[i+2]);
        hit.setCellID(ncellid);
    }

    return StatusCode::SUCCESS;
}
```

```
root [3] events->Show()
======> EVENT:5
 RawTimeSeries   = (vector<edm4hep::RawTimeSeriesData>*)0xde2df90
 RawTimeSeries.cellID = 140724617085792, 140724611843936, 140724607650144
 RawTimeSeries.quality = 0, 0, 0
 RawTimeSeries.time = 0.000000, 0.000000, 0.000000
 RawTimeSeries.charge = 0.000000, 0.000000, 0.000000
 RawTimeSeries.interval = 0.000000, 0.000000, 0.000000
 RawTimeSeries.adcCounts_begin = 0, 0, 0
 RawTimeSeries.adcCounts_end = 0, 0, 0
 RawTimeSeries_0 = NULL
```

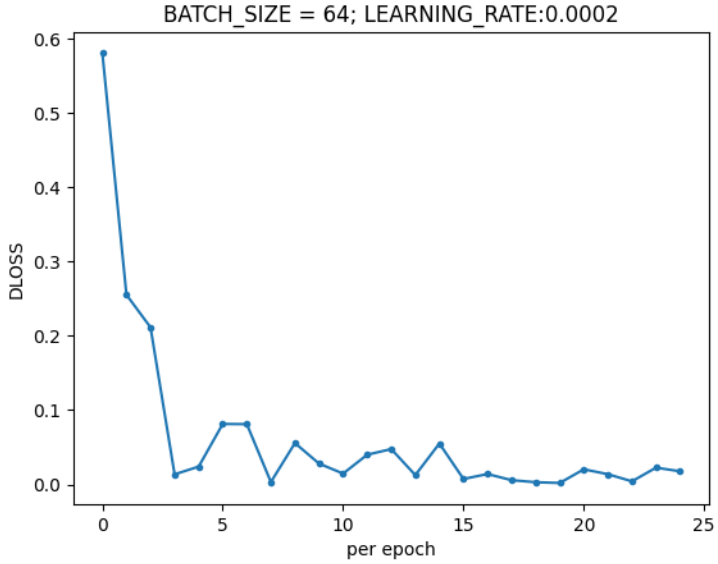
Encode in cellID

3

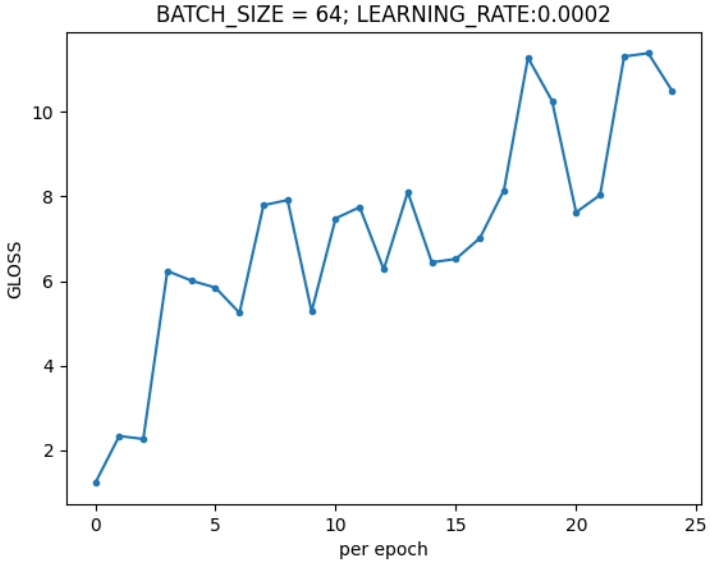
# Working progress

Cluster Reconstruction via Neural Network

## DCGAN Train




Discriminator Loss



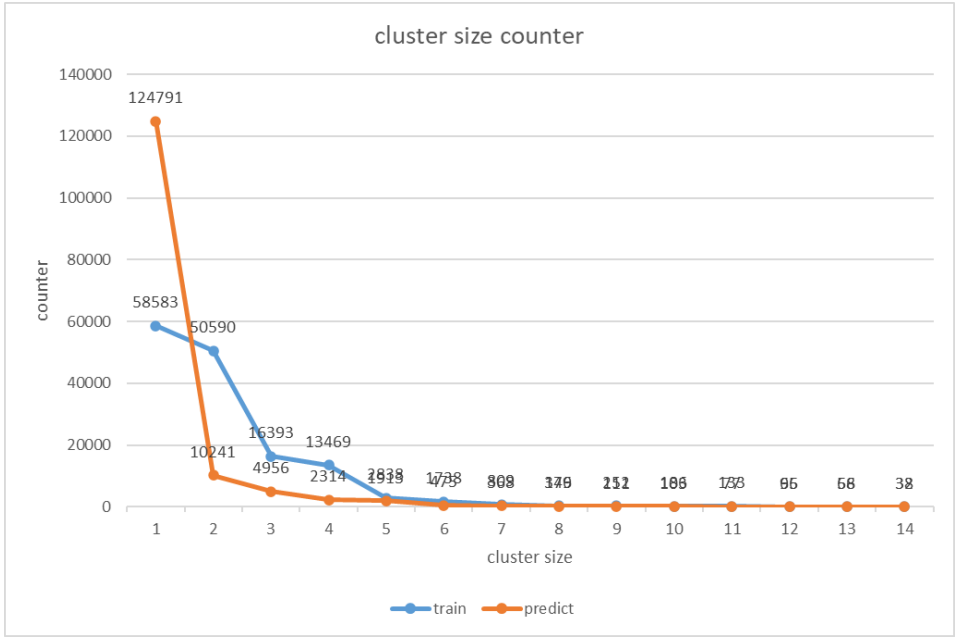

Generator Loss

基本符合预期。
Discriminator下降过快，考虑降低辨别器训练频率。

3

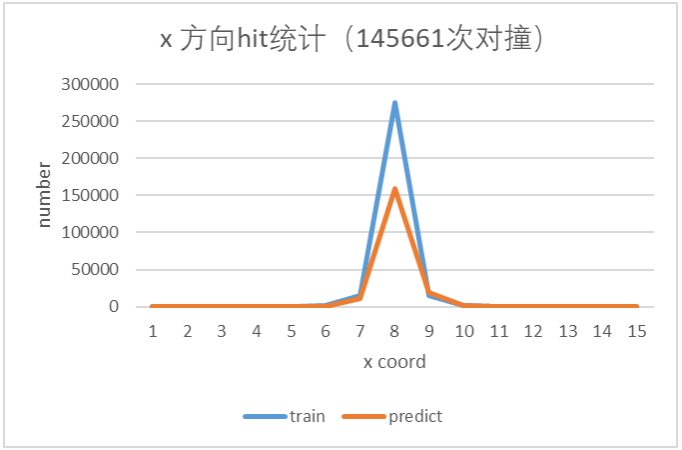
# Working progress

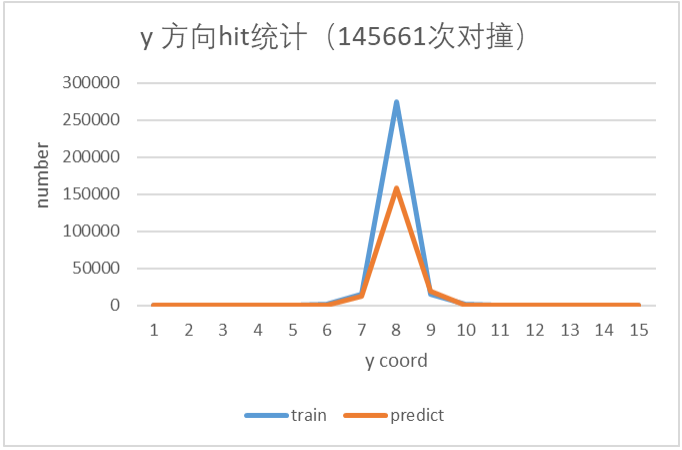
Cluster Reconstruction via Neural Network




Predictor分布分析：
训练出来的预测分布，cluster偏小，大部分预测都是只有一个点。
DCGAN没能很好的学习到训练集分布。

# DCGAN Predict

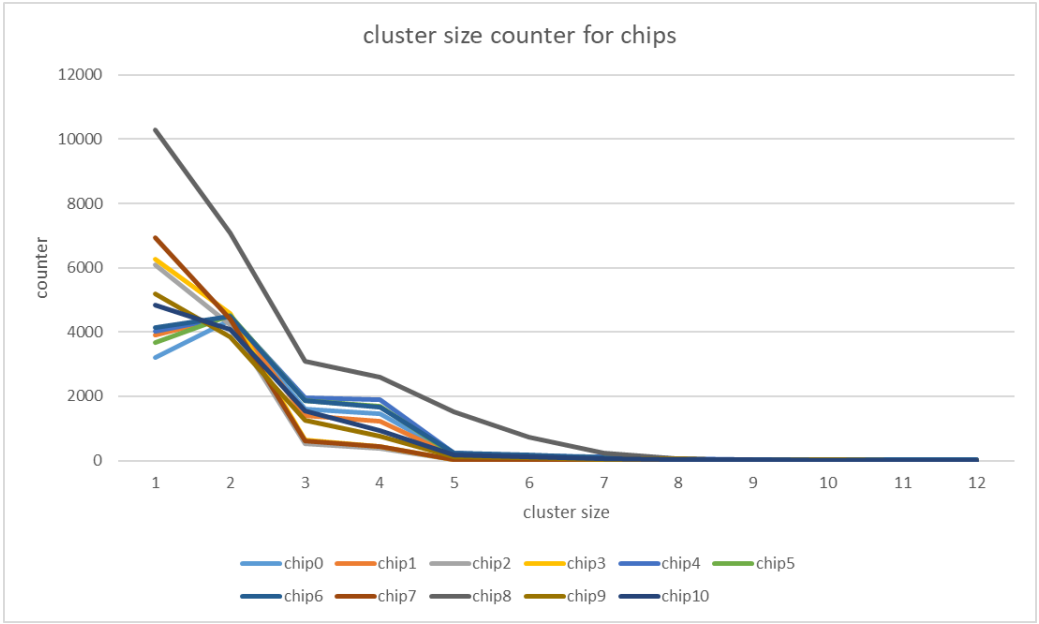


3

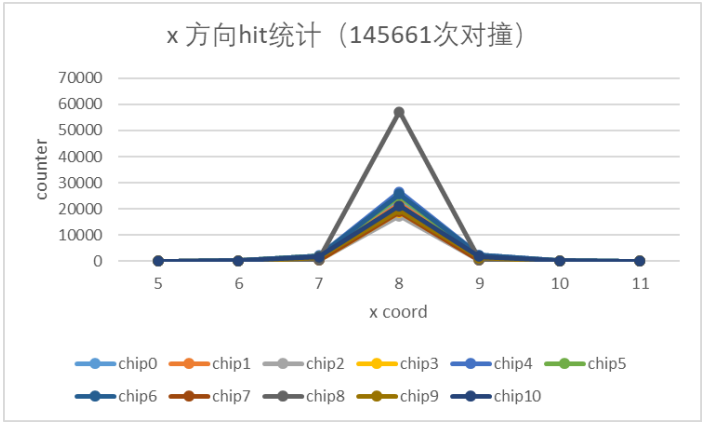
# Working progress

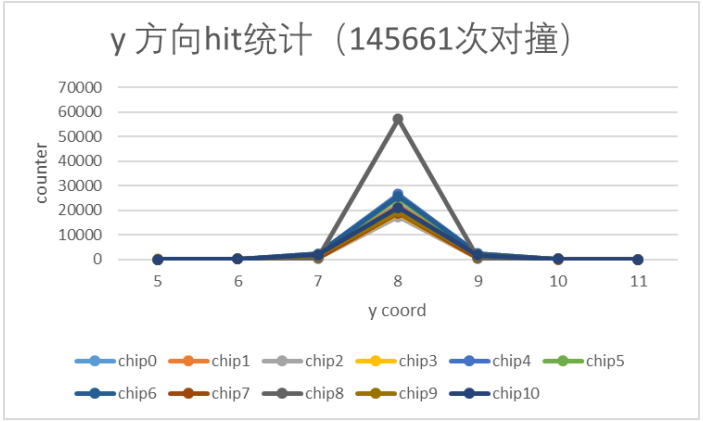
Cluster Reconstruction via Neural Network




训练数据分布统计：
从chip0-10，越靠后的plane，cluster越小

## DCGAN Predict



## Summary

- 目标：根据粒子穿过感应器的真实响应感应簇，通过机器学习方法，训练出能够模拟产生感应簇的模型。
- 使用标准的DCGAN模型，以TestBeam Data为训练集，训练了能初步预测簇形状的模型。
- 将模型加入CEPCSW，能够成功运行预测算法，并生成储存相应的RawTimeSeries Collection的root文件。

## Future Work

- 目前网络模型结果不甚理想，需要进一步优化模型（如使用FLOW模型）
- 目前模型的输入为隐空间的噪音，后续需要考虑加入其他输入（如：粒子入射角，能量，类型等）
- CEPCSW部分需要加入geant4模拟（需要TB的几何信息），使整一个预测工作流程跑通。

# Thank You

YiZhou Zhang, Tao Lin, WeiDong Li
zhangyz@ihep.ac.cn

30th Aug 2023

# Backup