x-jet-vis

March 18, 2024

[ ]:
```
!pip install pyarrow dask torch-geometric
```

```
Requirement already satisfied: pyarrow in /usr/local/lib/python3.10/dist-
packages (14.0.2)
Requirement already satisfied: dask in /usr/local/lib/python3.10/dist-packages
(2023.8.1)
Requirement already satisfied: torch-geometric in
/usr/local/lib/python3.10/dist-packages (2.5.0)
Requirement already satisfied: numpy>=1.16.6 in /usr/local/lib/python3.10/dist-
packages (from pyarrow) (1.25.2)
Requirement already satisfied: click>=8.0 in /usr/local/lib/python3.10/dist-
packages (from dask) (8.1.7)
Requirement already satisfied: cloudpickle>=1.5.0 in
/usr/local/lib/python3.10/dist-packages (from dask) (2.2.1)
Requirement already satisfied: fsspec>=2021.09.0 in
/usr/local/lib/python3.10/dist-packages (from dask) (2023.6.0)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.10/dist-packages (from dask) (23.2)
Requirement already satisfied: partd>=1.2.0 in /usr/local/lib/python3.10/dist-
packages (from dask) (1.4.1)
Requirement already satisfied: pyyaml>=5.3.1 in /usr/local/lib/python3.10/dist-
packages (from dask) (6.0.1)
Requirement already satisfied: toolz>=0.10.0 in /usr/local/lib/python3.10/dist-
packages (from dask) (0.12.1)
Requirement already satisfied: importlib-metadata>=4.13.0 in
/usr/local/lib/python3.10/dist-packages (from dask) (7.0.1)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages
(from torch-geometric) (4.66.2)
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages
(from torch-geometric) (1.11.4)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages
(from torch-geometric) (3.1.3)
Requirement already satisfied: aiohttp in /usr/local/lib/python3.10/dist-
packages (from torch-geometric) (3.9.3)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-
packages (from torch-geometric) (2.31.0)
Requirement already satisfied: pyparsing in /usr/local/lib/python3.10/dist-
packages (from torch-geometric) (3.1.1)
```

```
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-
packages (from torch-geometric) (1.2.2)
Requirement already satisfied: psutil>=5.8.0 in /usr/local/lib/python3.10/dist-
packages (from torch-geometric) (5.9.5)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.10/dist-
packages (from importlib-metadata>=4.13.0->dask) (3.17.0)
Requirement already satisfied: locket in /usr/local/lib/python3.10/dist-packages
(from partd>=1.2.0->dask) (1.0.0)
Requirement already satisfied: aiosignal>=1.1.2 in
/usr/local/lib/python3.10/dist-packages (from aiohttp->torch-geometric) (1.3.1)
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.10/dist-
packages (from aiohttp->torch-geometric) (23.2.0)
Requirement already satisfied: frozenlist>=1.1.1 in
/usr/local/lib/python3.10/dist-packages (from aiohttp->torch-geometric) (1.4.1)
Requirement already satisfied: multidict<7.0,>=4.5 in
/usr/local/lib/python3.10/dist-packages (from aiohttp->torch-geometric) (6.0.5)
Requirement already satisfied: yarl<2.0,>=1.0 in /usr/local/lib/python3.10/dist-
packages (from aiohttp->torch-geometric) (1.9.4)
Requirement already satisfied: async-timeout<5.0,>=4.0 in
/usr/local/lib/python3.10/dist-packages (from aiohttp->torch-geometric) (4.0.3)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.10/dist-packages (from jinja2->torch-geometric) (2.1.5)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from requests->torch-geometric) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-
packages (from requests->torch-geometric) (3.6)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from requests->torch-geometric) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from requests->torch-geometric)
(2024.2.2)
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-
packages (from scikit-learn->torch-geometric) (1.3.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in
/usr/local/lib/python3.10/dist-packages (from scikit-learn->torch-geometric)
(3.3.0)
```

[ ]:
```python
import pandas as pd
import pyarrow.parquet as parquet
first = 'QCDToGGQQ_IMGjet_RH1all_jet0_run0_n36272.test.snappy.parquet'
second = 'QCDToGGQQ_IMGjet_RH1all_jet0_run1_n47540.test.snappy.parquet'

first_file = parquet.ParquetFile('/content/drive/MyDrive/Sci_data/'+first)
```

0.0.1 Schema of the dataset

[ ]:
```
first_file.schema
```

[ ]:
```
<pyarrow._parquet.ParquetSchema object at 0x7fb95f035b00>
required group field_id=-1 schema {
  optional group field_id=-1 X_jets (List) {
    repeated group field_id=-1 list {
      optional group field_id=-1 item (List) {
        repeated group field_id=-1 list {
          optional group field_id=-1 item (List) {
            repeated group field_id=-1 list {
              optional double field_id=-1 item;
            }
          }
        }
      }
    }
  }
  optional double field_id=-1 pt;
  optional double field_id=-1 m0;
  optional double field_id=-1 y;
}
```

[ ]:
```
num_row_groups = first_file.num_row_groups

print("\nNumber of Rows:", num_row_groups)
```

```
Number of Rows: 36272
```

0.0.2 Since the size of the raw data is very large, We will first visualise a small chunk of X_jet

[ ]:
```
chunk_size =900
batches_df = []

for batch in first_file.iter_batches(chunk_size):
    print("RecordBatch")
    batch_df = batch.to_pandas()
    batches_df.append(batch_df)
    break
    # print("batch_df:", batch_df)
```

```
RecordBatch
```

[ ]:
```
batch_df
```

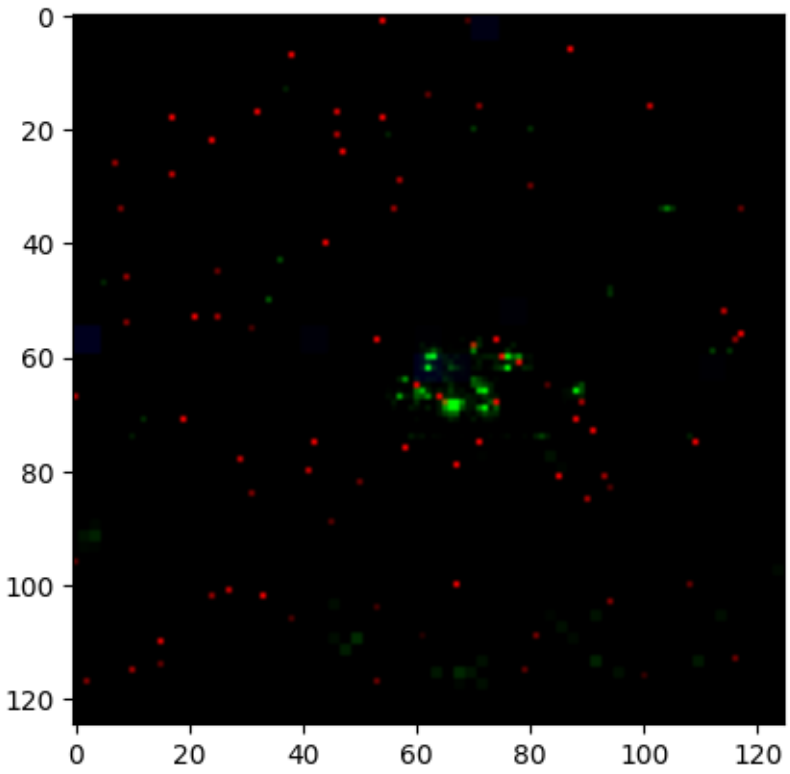
[ ]:
```
                                                  X_jets          pt         m0  \
0    [[[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0…  112.411095  21.098248
1    [[[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0…   95.220406  14.030600
2    [[[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0…   97.007317  17.728968
3    [[[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0…   82.490311  14.702741
4    [[[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.328483…  102.539238  19.456257
..                                                 …         …         …
895  [[[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0…  166.251846  23.967157
896  [[[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0…  104.868820  23.784552
897  [[[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0…   98.659180  12.243154
898  [[[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0…   84.996017  10.976908
899  [[[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0…   87.435623  19.032753

       y
0    0.0
1    1.0
2    1.0
3    0.0
4    0.0
..   …
895  1.0
896  0.0
897  1.0
898  1.0
899  1.0

[900 rows x 4 columns]
```

0.1 Visualizing `X_jets`

[ ]:
```
import numpy as np
img_list = []

limit = chunk_size
for number in range(limit):
  for idx, channels in enumerate(batch_df['X_jets'][number]):
    for i, row in enumerate(channels):
      if i==0:
        img = row
      else:
        img = np.vstack([img, row])
    if idx==0:
      final_img = img
    else:
      final_img = np.dstack([final_img, img])
  img_list.append(final_img)
```

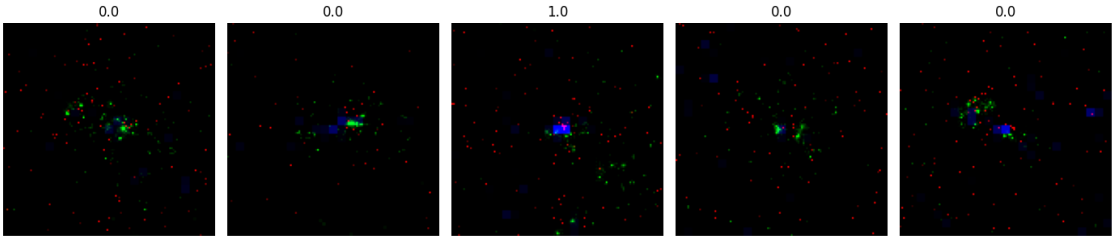
[ ]:
```
import matplotlib.pyplot as plt
plt.imshow(final_img)
plt.show()
```

```
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).
```



[ ]:
```
import random
indices = [random.randint(0,900) for _ in range(5)]

y_5 = batch_df['y'][indices].tolist()
images = [img_list[i] for i in indices]

fig, axes = plt.subplots(1, 5, figsize=(14, 12))

for i, image in enumerate(images):
    axes[i].imshow(image)
    axes[i].set_title(y_5[i]) # fix this
    axes[i].axis('off')

plt.tight_layout()
plt.show()
```

```
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).
```



[ ]: