Lab Notebook

Photonic Lantern Information Determination

Contents

Part I

Amplitude and Phase Reconstruction

16 February 2024

- Create new github branch: `AmpPhaseReconstructionRetraining`
- Download the following the files from Morgana:
  - `superK_slmcube_20230625_complsines-01sp_07`
  - `slmcube_20230625_complsines-01sp_07_PSFWFs_file00`
  - `slmcube_20230625_complsines-01sp_07_PSFWFs_file01`
  - `slmcube_20230625_complsines-01sp_07_PSFWFs_file02`
  - `slmcube_20230625_complsines-01sp_07_PSFWFs_file03`
  - `slmcube_20230625_complsines-01sp_07_PSFWFs_file04`
  - `slmcube_20230625_complsines-01sp_07_PSFWFs_file05`
  - `slmcube_20230625_complsines-01sp_07_PSFWFs_file06`
  - `slmcube_20230625_complsines-01sp_07_PSFWFs_file07`
  - `slmcube_20230625_complsines-01sp_07_PSFWFs_file08`
  - `slmcube_20230625_complsines-01sp_07_PSFWFs_file09`
- Data processing for Fully Convolutional NN training
- One fast experiment for each file: very good results, around 0.05 validation mse

---

19 February 2024

- Remove normalization from amplitude and phase for the fully connected experiments

---

EXPERIMENT NewFC10000-Processed 2

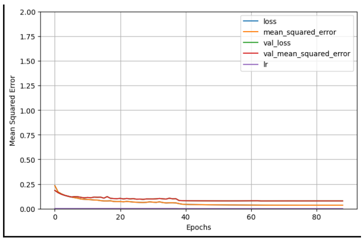
HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
  -Fully Connected
  -Input shape: 1320
  -Output shape: (2, 96, 96)
  -Hidden layers: [2000, 2000, 2000, 2000]
  -Regularizer: None
  -Hidden Layers Activation: relu
  -Output Layer Activation: linear
  -Batch Normalization: False
  -Dropout: False, 0.2

*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.001, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
  -Metric: MSE
```

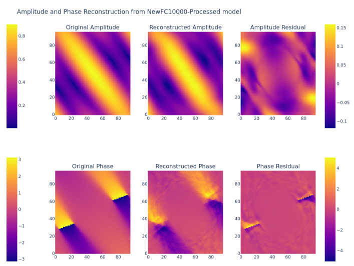
```
*TRAINING HYPERPARAMETERS:
  -Epochs: 200
  -Batch size: 64
  -Callbacks:
    -ReduceLROnPlateau: MSE 10 x0.1
    -Early Stop: MSE 25
```

VISUALIZATION:

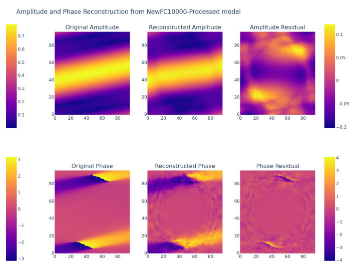
```
*RESULTS:
        -Train MSE: 0.035130929201841354
        -Validation MSE: 0.0786573588848114
```



(a) Training Evolution



(b) Validation example



(c) Train example

Figure 1: Results of training the model NewFC10000-Processed 2

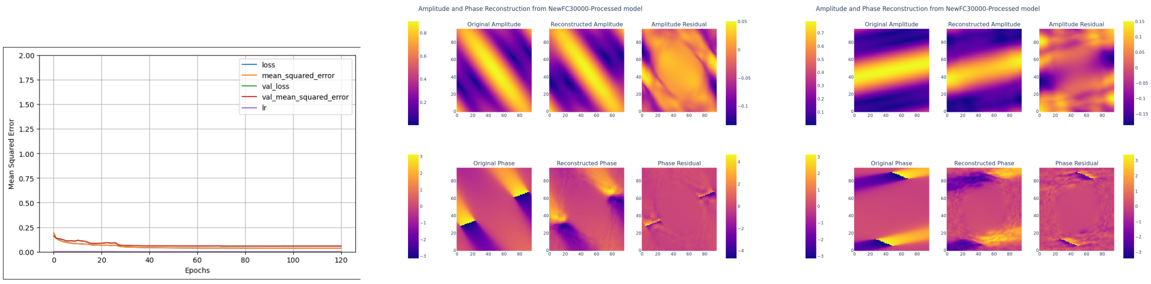
EXPERIMENT NewFC30000-Processed-1

HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
  -Fully Connected
  -Input shape: 1320
  -Output shape: (2, 96, 96)
  -Hidden layers: [2000, 2000, 2000, 2000]
  -Regularizer: None
  -Hidden Layers Activation: relu
  -Output Layer Activation: linear
  -Batch Normalization: False
  -Dropout: False, 0.2

*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.001, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
  -Metric: MSE

*TRAINING HYPERPARAMETERS:
  -Epochs: 200
  -Batch size: 64
  -Callbacks:
    -ReduceLROnPlateau: MSE 10 x0.1
    -Early Stop: MSE 25
```

VISUALIZATION:

```
*RESULTS:
      -Train MSE: 0.0357743538916111
      -Validation MSE: 0.057852283120155334
```



(a) Training Evolution (b) Validation example (c) Train example

Figure 2: Results of training the model NewFC30000-Processed-1

EXPERIMENT NewFC80000-Processed-1

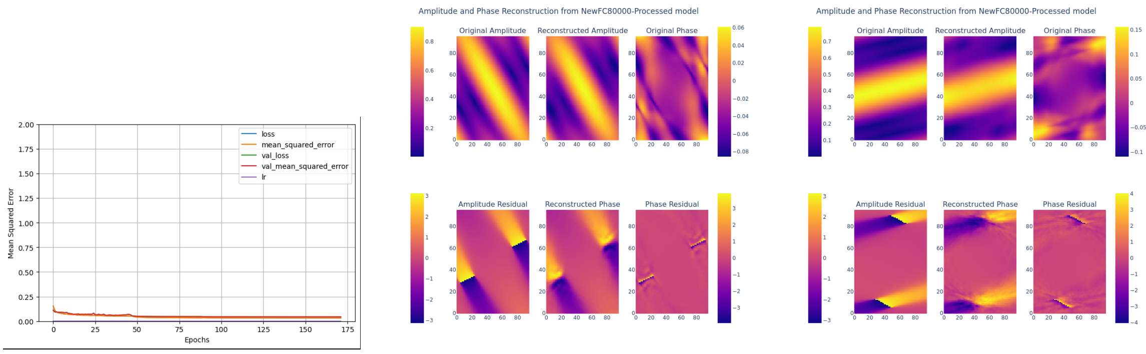
HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
   -Fully Connected
   -Input shape: 1320
   -Output shape: (2, 96, 96)
   -Hidden layers: [2000, 2000, 2000, 2000]
   -Regularizer: None
```

```
  -Hidden Layers Activation: relu
  -Output Layer Activation: linear
  -Batch Normalization: False
  -Dropout: False, 0.2

*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.001, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
  -Metric: MSE

*TRAINING HYPERPARAMETERS:
  -Epochs: 200
  -Batch size: 64
  -Callbacks:
    -ReduceLROnPlateau: MSE 10 x0.1
    -Early Stop: MSE 25
```

VISUALIZATION:

```
*RESULTS:
      -Train MSE: 0.03301804140210152
      -Validation MSE: 0.04487497732043266
```



(a) Training Evolution (b) Validation example (c) Train example

Figure 3: Results of training the model NewFC80000-Processed-1

- Normalize and split fluxes in train, validation and test files
- Stack amplitude and phase arrays and save in train, validation and test files

---

EXPERIMENT NewConv10000-1

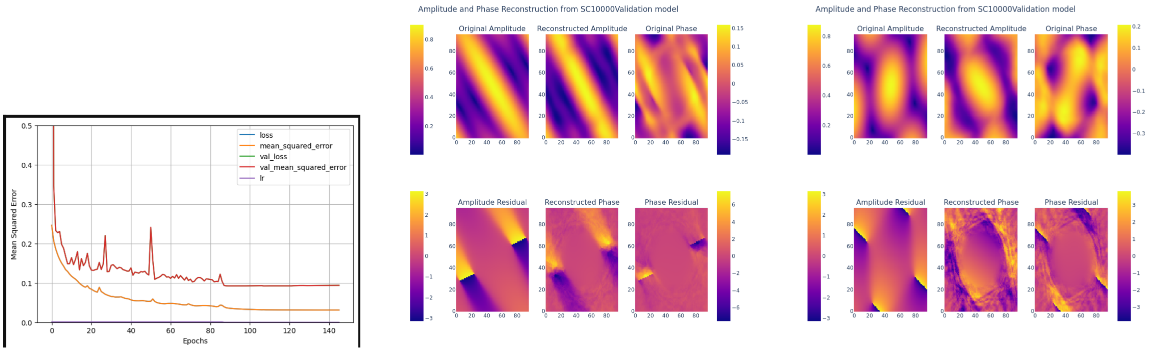
HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
  -Convolutional
  -Input shape: (55, 24, 1)
  -Output shape: (96, 96, 2)
  -Convolutional Layers: [128, 256, 512]
```

```
  -Convolutonal Kernels: [(3, 3), (3, 3), (3, 3)]
  -Fully Connected Hidden layers: [4096, 2048, 2048,
  1024, 1024, 1024]
  -Regularizer: None
  -Convolutional Activation: relu
  -Hidden Layers Activation: relu
  -Output Layer Activation: linear
  -Batch Normalization: True

*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.001, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
  -Metric: MSE

* TRAINING HYPERPARAMETERS:
  -Epochs: 200
  -Batch size: 32
  -Callbacks:
    -ReduceLROnPlateau: MSE 15 x0.1
    -Early Stop: MSE 50
```

VISUALIZATION:

```
*RESULTS:
      -Train MSE: 0.031087761744856834
      -Validation MSE: 0.09376049041748047
```



(a) Training Evolution (b) Validation example (c) Train example

Figure 4: Results of training the model NewConv10000-1

---

20 February 2024

EXPERIMENT NewConv30000-1

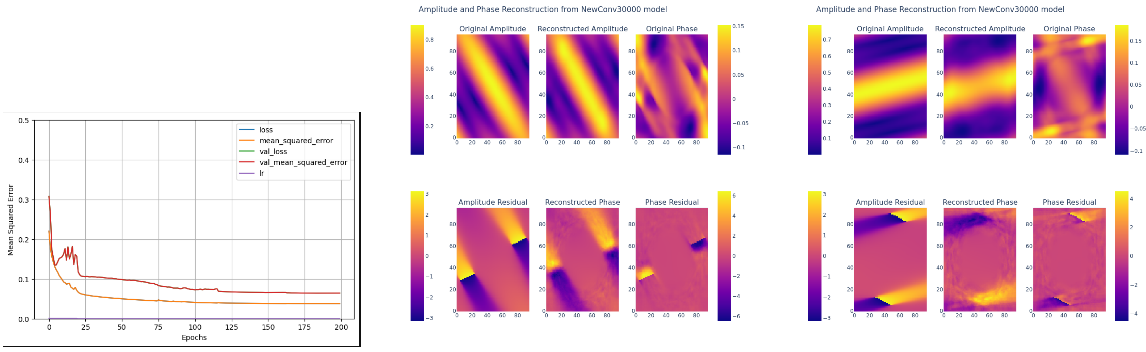
HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
  -Convolutional
  -Input shape: (55, 24, 1)
  -Output shape: (96, 96, 2)
  -Convolutional Layers: [128, 256, 512]
  -Convolutonal Kernels: [(3, 3), (3, 3), (3, 3)]
```

```
  -Fully Connected Hidden layers: [4096, 2048, 2048, 1024, 1024, 102
  -Regularizer: None
  -Convolutional Activation: relu
  -Hidden Layers Activation: relu
  -Output Layer Activation: linear
  -Batch Normalization: True

*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.001, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
  -Metric: MSE

* TRAINING HYPERPARAMETERS:
  -Epochs: 200
  -Batch size: 32
  -Callbacks:
    -ReduceLROnPlateau: MSE 15 x0.1
    -Early Stop: MSE 50
```

VISUALIZATION:

```
*RESULTS:
      -Train MSE: 0.03819489851593971
      -Validation MSE: 0.06443119794130325
```



(a) Training Evolution (b) Validation example (c) Train example

Figure 5: Results of training the model NewConv30000-1

EXPERIMENT NewConv30000-1

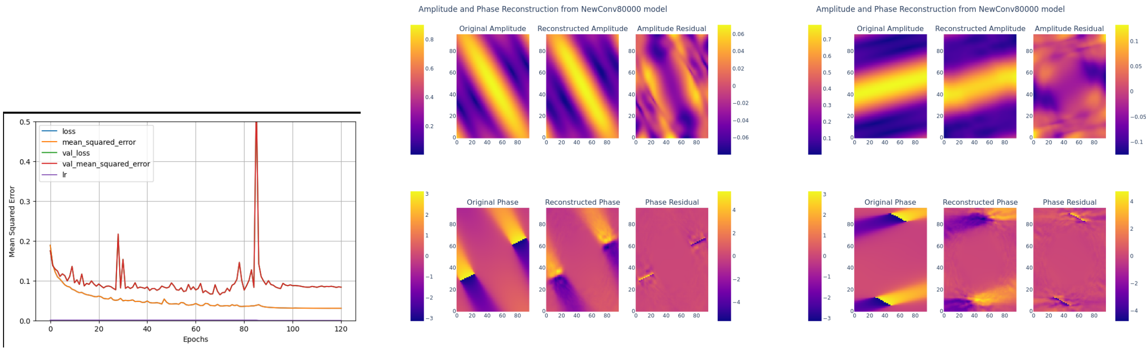
HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
  -Convolutional
  -Input shape: (55, 24, 1)
  -Output shape: (96, 96, 2)
  -Convolutional Layers: [128, 256, 512]
  -Convolutonal Kernels: [(3, 3), (3, 3), (3, 3)]
  -Fully Connected Hidden layers: [4096, 2048, 2048, 1024, 1024, 102
  -Regularizer: None
  -Convolutional Activation: relu
  -Hidden Layers Activation: relu
```

```
  -Output Layer Activation: linear
  -Batch Normalization: True

*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.001, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
  -Metric: MSE

* TRAINING HYPERPARAMETERS:
  -Epochs: 200
  -Batch size: 32
  -Callbacks:
    -ReduceLROnPlateau: MSE 15 x0.1
    -Early Stop: MSE 50
```

VISUALIZATION:

```
*RESULTS:
      -Train MSE: 0.030630357563495636
      -Validation MSE: 0.08355200290679932
```



(a) Training Evolution (b) Validation example (c) Train example

Figure 6: Results of training the model NewConv80000-1

- Normalize flux data and add padding for flux autoencoder
- Stack amplitude and phase and add padding for encoder+convolutional model

---

EXPERIMENT NewFluxAutoencoder10000-1

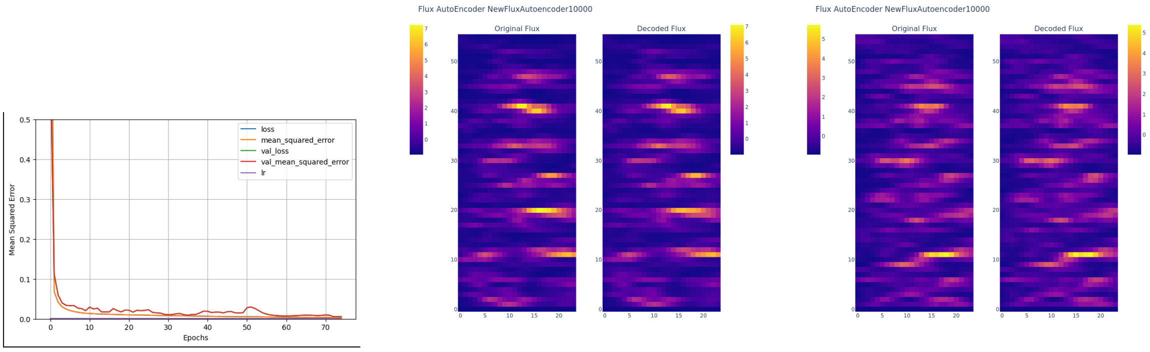
HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
  -Autoencoder
  -Input shape: (56, 24, 1)
  -Convolutional Layers: [512, 128, 64, 8]
    (Inverse in the decoder)
```

```
  -Convolutonal Kernels: [(3, 3), (3, 3), (3, 3), (3, 3)]
    (Inverse in the decoder)
  -Convolutional Activation: relu
  -Output Layer Activation: linear
  -Padding: same
  -Use Batch Normalization: True

*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.001, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
  -Metric: MSE

*TRAINING HYPERPARAMETERS:
  -Epochs: 75
  -Batch size: 32
  -Callbacks:
    -ReduceLROnPlateau: MSE 8 x0.1
    -Early Stop: MSE 15
```

VISUALIZATION:

```
*RESULTS:
      -Train MSE: 0.0018673702143132687
      -Validation MSE: 0.005482238717377186
```



(a) Training Evolution (b) Validation example (c) Train example

Figure 7: Results of training the model NewFluxAutoencoder10000-1

---

21 February 2024

EXPERIMENT NewFluxAutoencoder80000-1

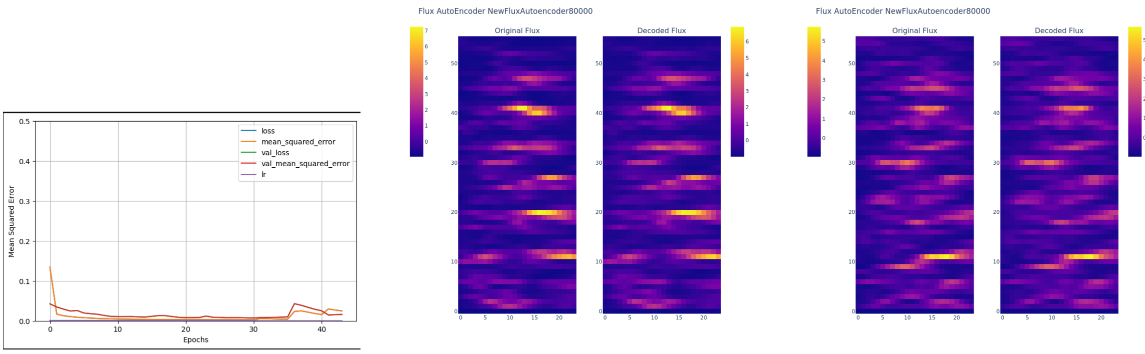
HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
  -Autoencoder
  -Input shape: (56, 24, 1)
  -Convolutional Layers: [512, 128, 64, 8]
    (Inverse in the decoder)
```

```
  -Convolutonal Kernels: [(3, 3), (3, 3), (3, 3), (3, 3)]
    (Inverse in the decoder)
  -Convolutional Activation: relu
  -Output Layer Activation: linear
  -Padding: same
  -Use Batch Normalization: True

*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.001, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
  -Metric: MSE

*TRAINING HYPERPARAMETERS:
  -Epochs: 75
  -Batch size: 32
  -Callbacks:
    -ReduceLROnPlateau: MSE 8 x0.1
    -Early Stop: MSE 15
```

VISUALIZATION:

```
      -Train MSE: 0.024667566642165184
      -Validation MSE: 0.015850609168410
```



(a) Training Evolution (b) Validation example (c) Train example

Figure 8: Results of training the model NewFluxAutoencoder10000-1

EXPERIMENT NewEncConv10000-1

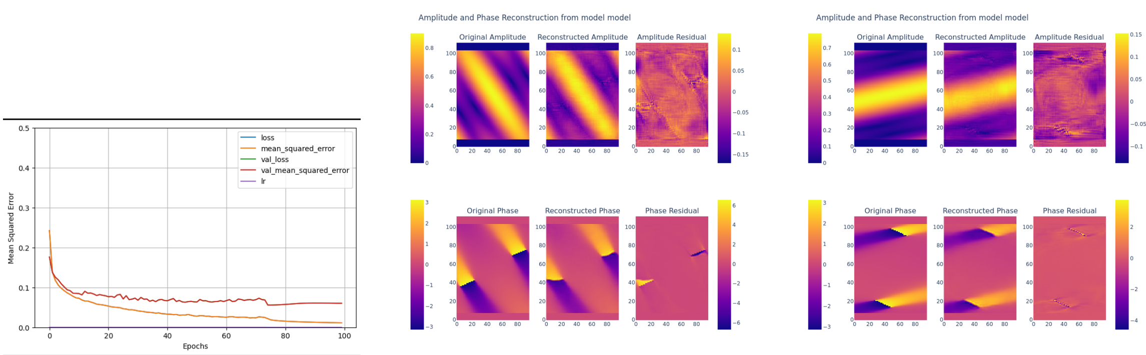
HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
  -Encoder + Convolutional
  -Convolutional Layers: [1024, 512, 256, 256]
  -Convolutonal Kernels: [(3, 3), (3, 3), (3, 3), (3, 3)]
  -Convolutional Activation: relu
  -Output Layer Activation: linear

*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.0001, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
```

```
    -Metric: MSE


  * TRAINING HYPERPARAMETERS:
    -Epochs: 100
    -Batch size: 32
    -Callbacks:
      -ReduceLROnPlateau: MSE 8 x0.1
      -Early Stop: MSE 15
```

VISUALIZATION:

```
  *RESULTS:
        -Train MSE: 0.011328332126140594
        -Validation MSE: 0.06024651601910591
```



(a) Training Evolution (b) Validation example (c) Train example

Figure 9: Results of training the model NewEncConv10000-1

EXPERIMENT NewEncConv30000-1

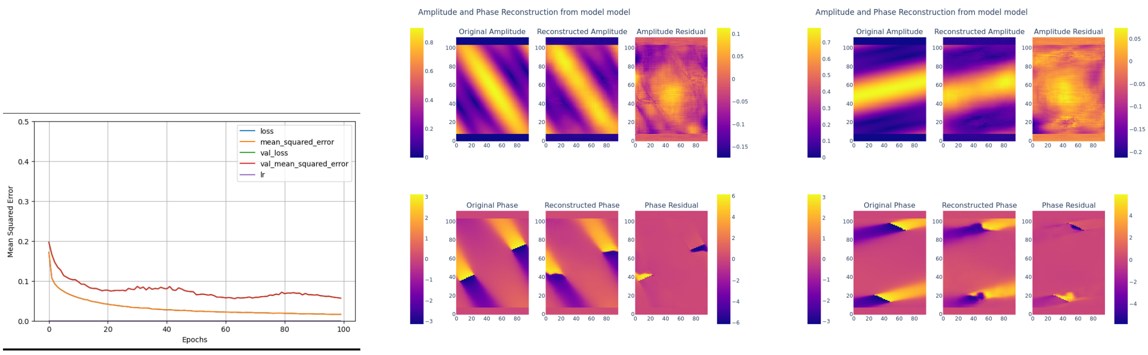
HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
  -Encoder + Convolutional
  -Convolutional Layers: [1024, 512, 256, 256]
  -Convolutonal Kernels: [(3, 3), (3, 3), (3, 3), (3, 3)]
  -Convolutional Activation: relu
  -Output Layer Activation: linear

*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.0001, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
  -Metric: MSE

* TRAINING HYPERPARAMETERS:
  -Epochs: 100
  -Batch size: 32
  -Callbacks:
    -ReduceLROnPlateau: MSE 8 x0.1
    -Early Stop: MSE 15
```

VISUALIZATION:

```
*RESULTS:
      -Train MSE: 0.016221819445490837
```

```
    -Validation MSE: 0.05662507936358452
```



(a) Training Evolution (b) Validation example (c) Train example

Figure 10: Results of training the model NewEncConv30000-1

---

22 February 2024

EXPERIMENT NewEncConv80000-1

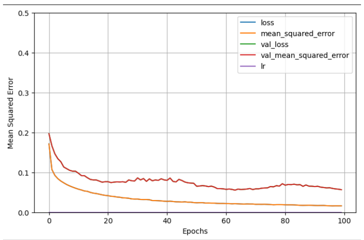
HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
  -Encoder + Convolutional
  -Convolutional Layers: [1024, 512, 256, 256]
```

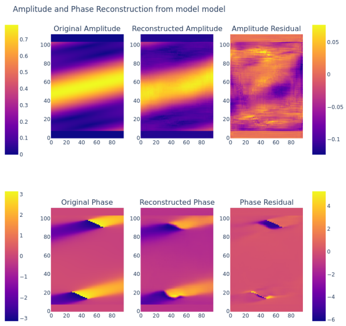
```
  -Convolutonal Kernels: [(3, 3), (3, 3), (3, 3), (3, 3)]
  -Convolutional Activation: relu
  -Output Layer Activation: linear

*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.0001, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
  -Metric: MSE

* TRAINING HYPERPARAMETERS:
  -Epochs: 100
  -Batch size: 32
  -Callbacks:
    -ReduceLROnPlateau: MSE 8 x0.1
    -Early Stop: MSE 15
```

VISUALIZATION:

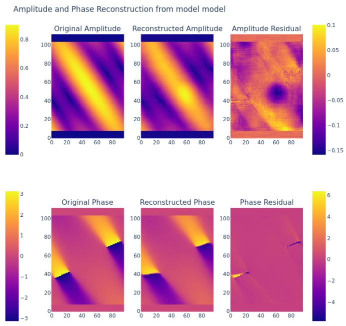
```
*RESULTS:
      -Train MSE: 0.0163
      -Validation MSE: 0.0369
```



(a) IGNORE THIS EVOLUTION FIGURE, TRAINING FINISHED BEFORE STORING THE HISTORY, THIS IS FROM THE PREVIOUS EXPERIMENT



(b) Validation example



(c) Train example

Figure 11: Results of training the model NewEncConv80000-1

Part II

PSF Reconstruction

20 February 2024

- Create custom dynamic dataloader for PL output flux and PSF complex field for PSF reconstruction

26 February 2024

- Create a dataloader to dynamically load flux and psf
- Normalize electric fields, real and imaginary parts independently
- Normalize PL output fluxes
- Train a fully connected nn with horrible results, train mse keeps stable at 1, will have to look at the data

---

27 February 2024

- Create plotting functions to show amplitude, phase and intensity from an electric field
- Normalize electric field, this time dividing the complex number matrix by a constant (50.000 in this case)
- Redo output flux calculation with the normalized electric fields through the transfer matrix of the 19 fibre PL

---

28 February 2024

EXPERIMENT PSF-FC-Reconstructor-1

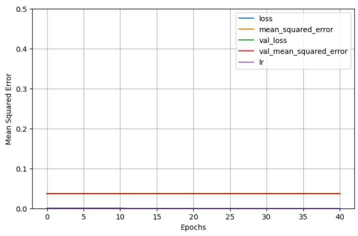
HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
  -Fully Connected
  -Input shape: 19
  -Output shape: 32768
  -Hidden layers: [2000, 2000, 2000, 2000]
  -Regularizer: None
  -Hidden Layers Activation: relu
  -Output Layer Activation: linear
  -Batch Normalization: False
  -Dropout: False, 0.2

*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.001, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
  -Metric: MSE

*TRAINING HYPERPARAMETERS:
  -Epochs: 200
```

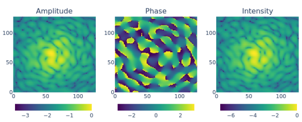
```
    -Batch size: 64
    -Callbacks:
      -ReduceLROnPlateau: MSE 10 x0.1
      -Early Stop: MSE 25
```

VISUALIZATION:

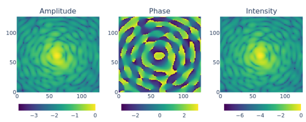
```
        -Train MSE: 0.03701553866267204
        -Validation MSE: 0.03701810911297798
```



(a) Training Evolution



(b) Validation prediction



(c) Validation original

Figure 12: Results of training the model PSF-FC-Reconstructor-1

---

29 February 2024

- Create a function to plot amplitude and phase predictions and residuals from a complex field
- Create evaluation notebook
- Since in yesterdays experiment the model does not learn at all, I design two different experiments, to check if there is a problem with the network:

1. A model that sums the output flux of the PL
2. A model that sums the scalars of the complex numbers array containing the electric fields at the pupil.

EXPERIMENT Flux-Sum-1

This model performs a sum of all the output fluxes from the PL

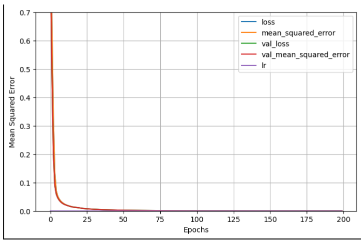
HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
  -Fully Connected
  -Input shape: 19
  -Output shape: 1
  -Hidden layers: [100, 100]
  -Regularizer: None
  -Hidden Layers Activation: relu
  -Output Layer Activation: linear
  -Batch Normalization: False
  -Dropout: False, 0.2

*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.0001, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
  -Metric: MSE
```

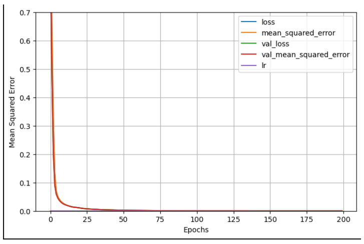
```
*TRAINING HYPERPARAMETERS:
  -Epochs: 200
  -Batch size: 64
  -Callbacks:
    -ReduceLROnPlateau: MSE 10 x0.1
    -Early Stop: MSE 25
```

VISUALIZATION:

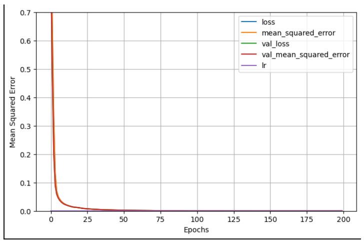
```
  *RESULTS:
      -Train MSE: 0.00018273142632097006
      -Validation MSE: 0.00018727070710156113
```



(a) Training Evolution



(b) Training Evolution



(c) Training Evolution

Figure 13: Results of training the model Flux-Sum-1

EXPERIMENT ElectricField-Sum-1

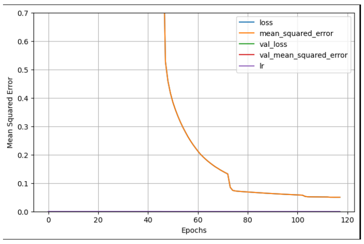
This model performs a sum of all the output fluxes from the PL

HYPERPARAMETERS:

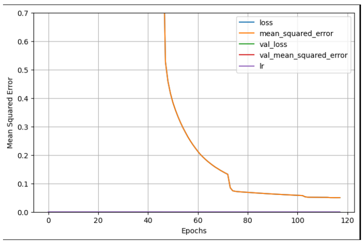
```
*ARCHITECTURE HYPERPARAMETERS:
  -Fully Connected
  -Input shape: 32768
  -Output shape: 1
  -Hidden layers: [100, 100, 100]
  -Regularizer: None
  -Hidden Layers Activation: relu
  -Output Layer Activation: linear
  -Batch Normalization: False
  -Dropout: False, 0.2

*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.0001, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
  -Metric: MSE

*TRAINING HYPERPARAMETERS:
  -Epochs: 200
  -Batch size: 64
  -Callbacks:
    -ReduceLROnPlateau: MSE 10 x0.1
    -Early Stop: MSE 25
```

VISUALIZATION:

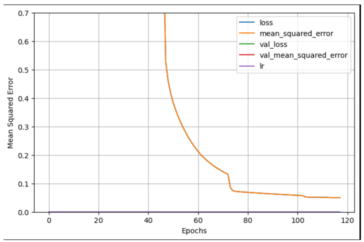
```
*RESULTS:
    -Train MSE: 0.050156451761722565
    -Validation MSE: 0.8189939856529236
```



(a) Training Evolution



(b) Training Evolution



(c) Training Evolution

Figure 14: Results of training the model ElectricField-Sum-1

---

01 March 2024

- Today I decide to train a model with just one datapoint and see if is able to learn the PSF from one flux and one electric field

---

EXPERIMENT TestWith1DataPoint-1

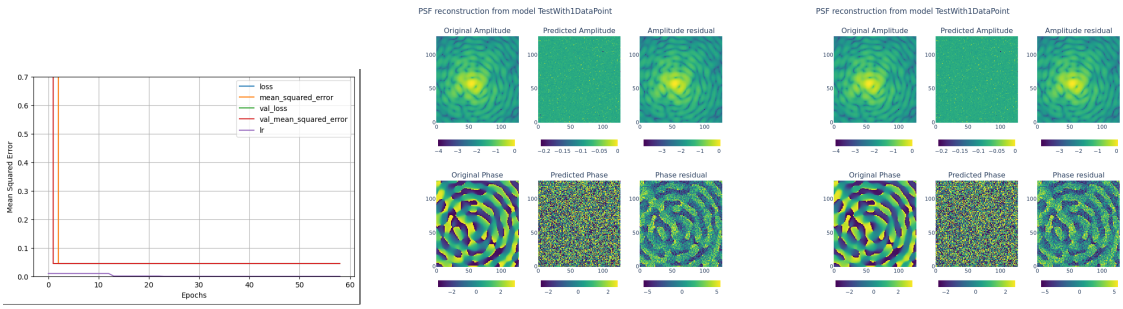
HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
```

```
  -Fully Connected
  -Input shape: 19
  -Output shape: 32768
  -Hidden layers: [1000, 1000, 1000, 1000]
  -Regularizer: None
  -Hidden Layers Activation: relu
  -Output Layer Activation: linear
  -Batch Normalization: False
  -Dropout: False, 0.2

*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.01, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
  -Metric: MSE

*TRAINING HYPERPARAMETERS:
  -Epochs: 1000
  -Batch size: 64
  -Callbacks:
    -ReduceLROnPlateau: MSE 10 x0.1
    -Early Stop: MSE 25
```

VISUALIZATION:

```
    -Train MSE: 0.03701553866267204
    -Validation MSE: 0.03701810911297798
```



(a) Training Evolution (b) Train datapoint (c) Train datapoint

Figure 15: Results of training the model TestWith1DataPoint-1

- Something is wrong, going try with a bigger NN, don't think this will change anything but I have to try just in case.

EXPERIMENT TestWith1DataPoint-2

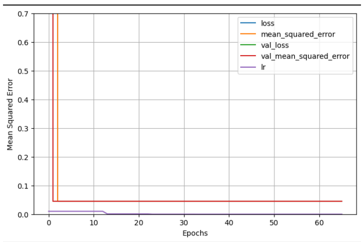
HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
  -Fully Connected
  -Input shape: 19
  -Output shape: 32768
  -Hidden layers: [128, 128, 128, 128, 256,
          256, 512, 2000, 4000]
  -Regularizer: None
```

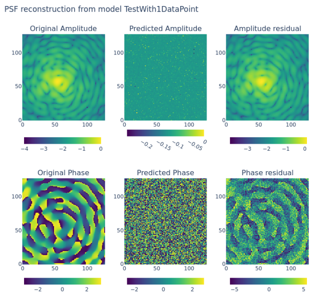
```
  -Hidden Layers Activation: relu
  -Output Layer Activation: linear
  -Batch Normalization: False
  -Dropout: False, 0.2

*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.01, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
  -Metric: MSE

*TRAINING HYPERPARAMETERS:
  -Epochs: 1000
  -Batch size: 64
  -Callbacks:
    -ReduceLROnPlateau: MSE 10 x0.1
    -Early Stop: MSE 25
```

VISUALIZATION:

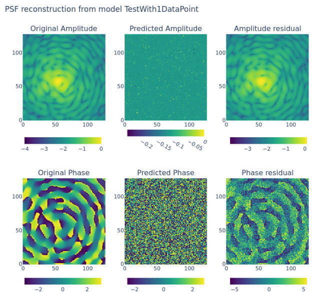
```
*RESULTS:
      -Train MSE: 0.04261719062924385
      -Validation MSE: 0.04261719062924385
```



(a) Training Evolution



(b) Train datapoint



(c) Train datapoint

Figure 16: Results of training the model TestWith1DataPoint-2

---

04 March 2024

- After seeing that with 1 datapoint the model does not learn I decide to make some simpler tests, I will train a model that from a 19 element array predicts images with 2 channels and see if the input is too small for the model to predict anything. Three experiments will be performed:
  - From a 19 elements array to a 2x128x128 image
  - From a 19 elements array to a 2x64x64 image
  - From a 19 elements array to a 2x32x32 image

EXPERIMENT FromArrayToImage128-1

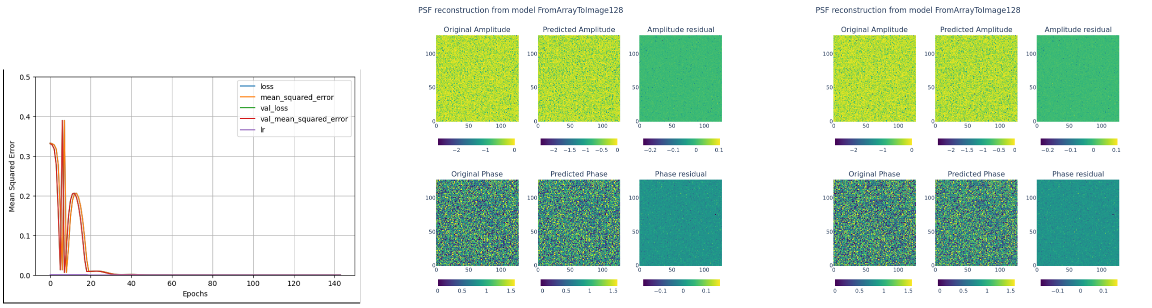
HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
  -Fully Connected
  -Input shape: 19
  -Output shape: 32768
  -Hidden layers: [128, 128, 128, 128, 256, 256, 512, 2000, 4000]
  -Regularizer: None
  -Hidden Layers Activation: relu
  -Output Layer Activation: linear
  -Batch Normalization: False
  -Dropout: False, 0.2

*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.001, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
  -Metric: MSE

*TRAINING HYPERPARAMETERS:
  -Epochs: 10000
  -Batch size: 1
  -Callbacks:
    -ReduceLROnPlateau: MSE 10 x0.1
    -Early Stop: MSE 25
```

VISUALIZATION:

```
*RESULTS:
```

```
    -Train MSE: 0.00010900569031946361
```



(a) Training Evolution (b) Train example (c) Train example

Figure 17: Results of training the model FromArrayToImage128-1

- From the previous experiment it is clear that a 19 element array as input is enough to reconstruct a random image of 2x128x128 resolution, what is happening for the psf reconstruction?.

  I will perform the other 2 stated experiments because it is quite fast.

EXPERIMENT FromArrayToImage64-1

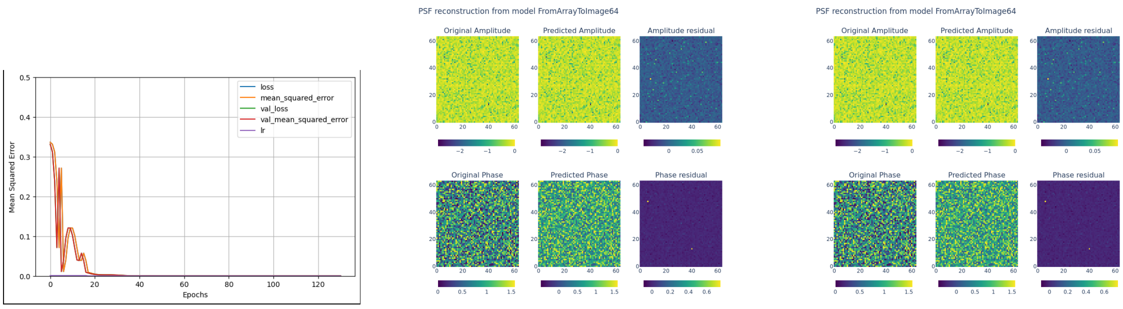
HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
  -Fully Connected
```

```
  -Input shape: 19
  -Output shape: 8192
  -Hidden layers: [128, 128, 128, 128, 256, 256, 512, 2000, 4000]
  -Regularizer: None
  -Hidden Layers Activation: relu
  -Output Layer Activation: linear
  -Batch Normalization: False
  -Dropout: False, 0.2

*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.001, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
  -Metric: MSE

*TRAINING HYPERPARAMETERS:
  -Epochs: 10000
  -Batch size: 1
  -Callbacks:
    -ReduceLROnPlateau: MSE 10 x0.1
    -Early Stop: MSE 25
```

VISUALIZATION:

```
*RESULTS:
      -Train MSE: 0.00012627203250303864
```



(a) Training Evolution (b) Train example (c) Train example

Figure 18: Results of training the model FromArrayToImage64-1

EXPERIMENT FromArrayToImage32-1

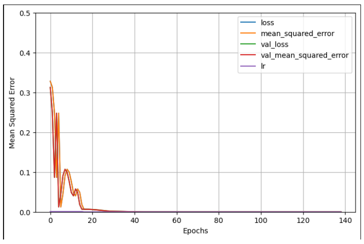
HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
  -Fully Connected
  -Input shape: 19
  -Output shape: 2048
  -Hidden layers: [128, 128, 128, 128, 256, 256, 512, 2000, 4000]
  -Regularizer: None
  -Hidden Layers Activation: relu
  -Output Layer Activation: linear
  -Batch Normalization: False
```

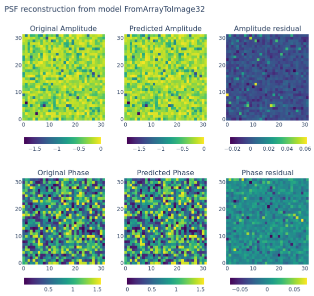
```
  -Dropout: False, 0.2


*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.001, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
  -Metric: MSE


*TRAINING HYPERPARAMETERS:
  -Epochs: 10000
  -Batch size: 1
  -Callbacks:
    -ReduceLROnPlateau: MSE 10 x0.1
    -Early Stop: MSE 25
```

VISUALIZATION:

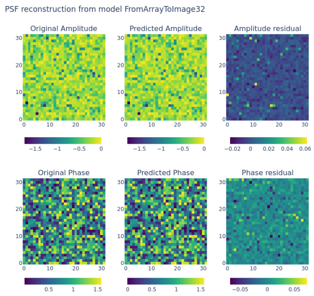
```
*RESULTS:
      -Train MSE: 5.556200267164968e-05
```



(a) Training Evolution



(b) Train example



(c) Train example

Figure 19: Results of training the model FromArrayToImage32-1

- The results are satisfactory, I am going to try the exact same configuration for one data point of the PSF dataset and see what happens

---

EXPERIMENT TestWith1DataPoint-3

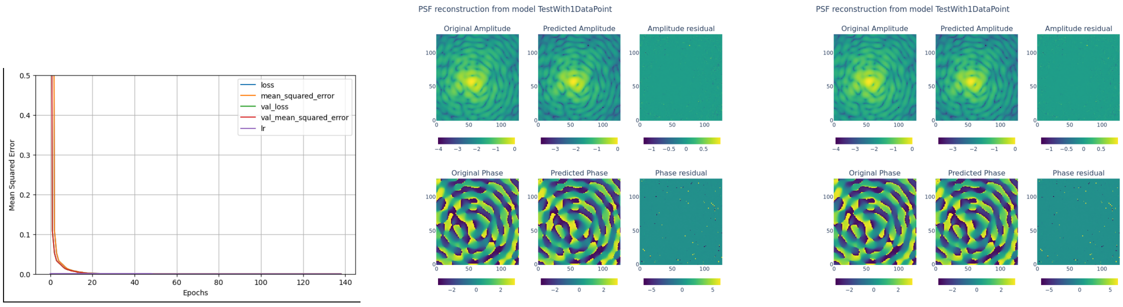
HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
  -Fully Connected
  -Input shape: 19
  -Output shape: 32768
  -Hidden layers: [128, 128, 128, 128, 256, 256, 512, 2000, 4000]
  -Regularizer: None
  -Hidden Layers Activation: relu
```

```
  -Output Layer Activation: linear
  -Batch Normalization: False
  -Dropout: False, 0.2

*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.001, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
  -Metric: MSE

*TRAINING HYPERPARAMETERS:
  -Epochs: 10000
  -Batch size: 1
  -Callbacks:
    -ReduceLROnPlateau: MSE 10 x0.1
    -Early Stop: MSE 25
```

VISUALIZATION:

```
*RESULTS:
      -Train MSE: 1.2409615010255948e-05
```



(a) Training Evolution (b) Train datapoint (c) Train datapoint

Figure 20: Results of training the model TestWith1DataPoint-3

- The result has improved incredibly compared to the TestWith1DataPoint-1 and TestWith1DataPoint-2. The only difference I can see are the learning rates and the batch size. I will perform a series of experiments varying batch size: 100 datapoints with 64, 32 and 16 batch size

---

EXPERIMENT PSFReconstructor-100-64-1

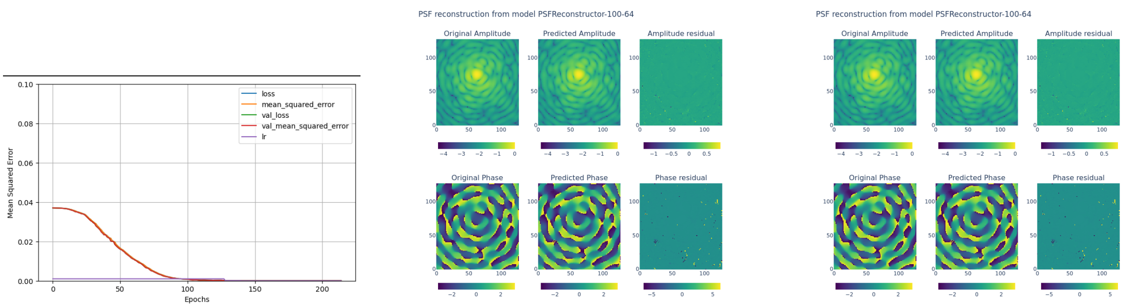
HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
  -Fully Connected
  -Input shape: 19
  -Output shape: 32768
  -Hidden layers: [128, 128, 128, 128, 256,
```

```
          256, 512, 2000, 4000]
    -Regularizer: None
    -Hidden Layers Activation: relu
    -Output Layer Activation: linear
    -Batch Normalization: False
    -Dropout: False, 0.2

  *COMPILATION HYPERPARAMETERS:
    -Optimizer: ADAM lr=0.001, beta_1=0.9, beta_2=0.999
    -Loss Function: MSE
    -Metric: MSE

  *TRAINING HYPERPARAMETERS:
    -Epochs: 10000
    -Batch size: 64
    -Callbacks:
      -ReduceLROnPlateau: MSE 10 x0.1
      -Early Stop: MSE 25
```

VISUALIZATION:

```
*RESULTS:
      -Train MSE: 1.7399888747604564e-05
```



(a) Training Evolution (b) Train datapoint (c) Train datapoint

Figure 21: Results of training the model PSFReconstructor-100-64-1

---

05 March 2024

EXPERIMENT PSFReconstructor-100-32-1

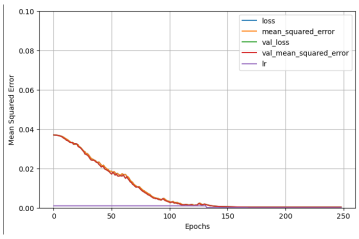
HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
  -Fully Connected
  -Input shape: 19
  -Output shape: 32768
  -Hidden layers: [128, 128, 128, 128, 256,
          256, 512, 2000, 4000]
```

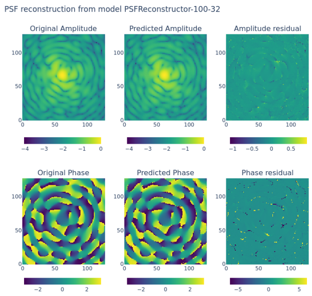
```
  -Regularizer: None
  -Hidden Layers Activation: relu
  -Output Layer Activation: linear
  -Batch Normalization: False
  -Dropout: False, 0.2

*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.001, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
  -Metric: MSE

*TRAINING HYPERPARAMETERS:
  -Epochs: 10000
  -Batch size: 32
  -Callbacks:
    -ReduceLROnPlateau: MSE 10 x0.1
    -Early Stop: MSE 25
```

VISUALIZATION:

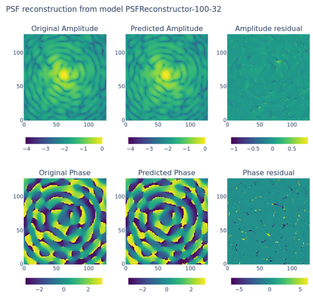
```
*RESULTS:
      -Train MSE: 0.00033666446688584983
```



(a) Training Evolution



(b) Train datapoint



(c) Train datapoint

Figure 22: Results of training the model PSFReconstructor-100-32-1

EXPERIMENT PSFReconstructor-100-16-1

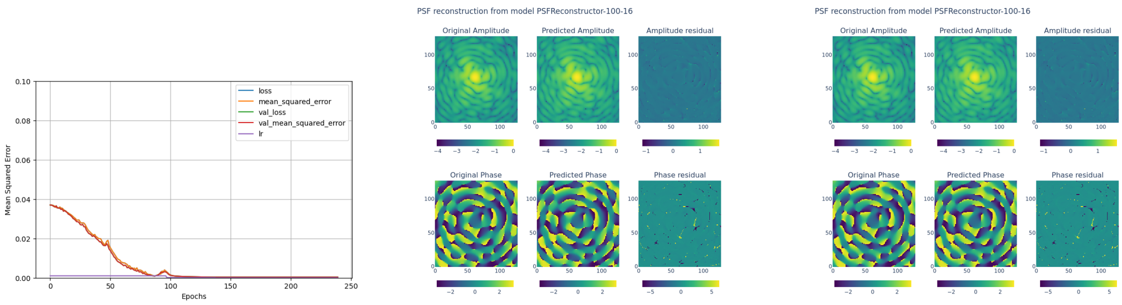
HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
  -Fully Connected
  -Input shape: 19
  -Output shape: 32768
  -Hidden layers: [128, 128, 128, 128, 256,
          256, 512, 2000, 4000]
  -Regularizer: None
  -Hidden Layers Activation: relu
  -Output Layer Activation: linear
```

```
  -Batch Normalization: False
  -Dropout: False, 0.2

*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.001, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
  -Metric: MSE

*TRAINING HYPERPARAMETERS:
  -Epochs: 10000
  -Batch size: 16
  -Callbacks:
    -ReduceLROnPlateau: MSE 10 x0.1
    -Early Stop: MSE 25
```

VISUALIZATION:

```
*RESULTS:
    -Train MSE: 0.0003098844608757645
```



(a) Training Evolution (b) Train datapoint (c) Train datapoint

Figure 23: Results of training the model PSFReconstructor-100-16-1

- Apart from the mse, no significant difference can be appreciated for different batch sizes using 100 datapoints.
- The goal for today is to study the correlation between the L2 norm of the pupil plane electric field and the L2 norm of the LP modes complex coefficients.
- First compute the L2 norm of the LP modes complex coefficients from the validation psf electric fields.
- Done!, here it is what it looks like

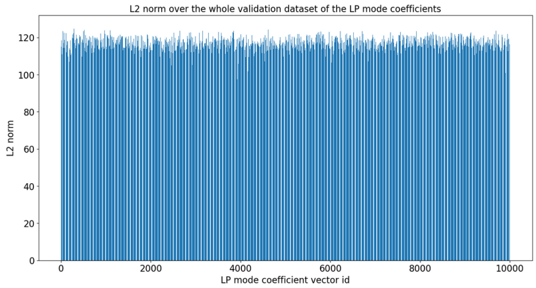


Figure 24: L2 norms for the overlap integral mode coefficients of the validation PSFs

- Let's do the same for the electric fields of the PSF
- Done!, here are the results:

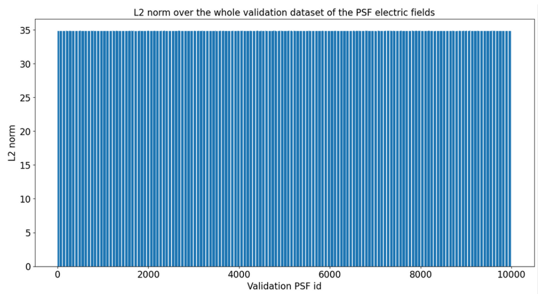


Figure 25: L2 norms of the validation PSFs electric fields

- After doing a correlation analysis the result is the following:
  `Pearson correlation coefficient:  0.1451513895198259`
  Which means that there is no correlation

---

06 March 2024

- Today I will be looking at the code from the `lantern_fiber_utils.py` file looking for the place where the LP mode coefficients are normalized

- Found it, in the function `make_fiber_modes()` used to compute de coefficients, there is a parameter called `normtosum` that is set to True by default, it looks as it normalizes the coefficients after the overlap integral is computed.

- Set `normtosum` to False, and compute 10000 output fluxes.

---

07 March 2024

- I will test a model with the new generated unnormalized output fluxes

---

EXPERIMENT UnnormalizedCoefficients-1

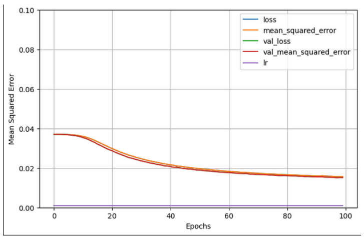
HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
  -Fully Connected
  -Input shape: 19
  -Output shape: 32768
  -Hidden layers: [256, 256, 256]
  -Regularizer: None
  -Hidden Layers Activation: relu
  -Output Layer Activation: linear
  -Batch Normalization: False
```

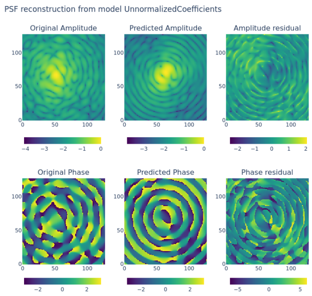
```
  -Dropout: False, 0.2


*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.001, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
  -Metric: MSE


*TRAINING HYPERPARAMETERS:
  -Epochs: 100
  -Batch size: 64
  -Callbacks:
    -ReduceLROnPlateau: MSE 10 x0.1
    -Early Stop: MSE 25
```

VISUALIZATION:

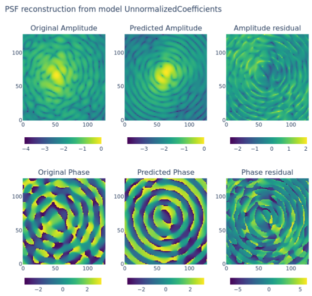
```
  *RESULTS:
      -Train MSE: 0.01565057598054409
```



(a) Training Evolution



(b) Train datapoint



(c) Train datapoint

Figure 26: Results of training the model UnnormalizedCoefficientsS-1

- Looks like the problem with the data was in the normalization of the coefficients, I will reprocess the output fluxes again and then test with a bigger dataset.

- While, the processing is running I create two new functions to plot, amplitude-phase-intensity and output flux of the PL.

---

08 March 2024

- FUCK, When processing the data I forgot to disable the `normtosum` variable. Have to repeat the process again

---

11 March 2024

- Finally got the output fluxes from the unnormalized mode coefficients

---

EXPERIMENT PSFRecontructorFC10000-1

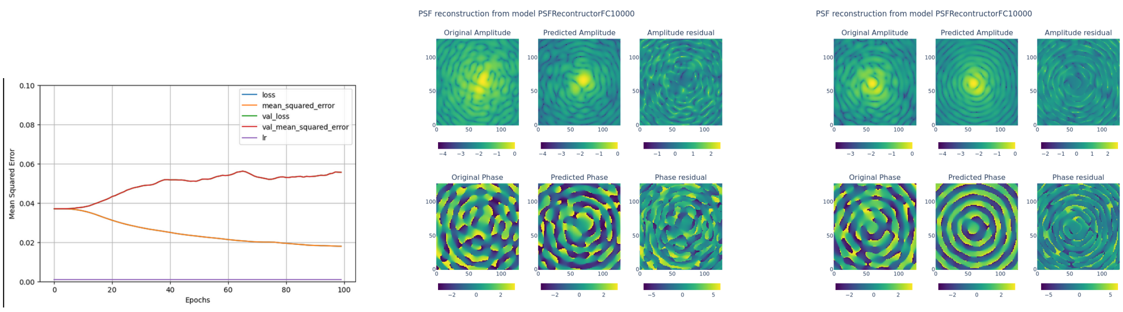
HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
  -Fully Connected
  -Input shape: 19
  -Output shape: 32768
  -Hidden layers: [256, 256, 256]
  -Regularizer: None
  -Hidden Layers Activation: relu
  -Output Layer Activation: linear
  -Batch Normalization: False
  -Dropout: False, 0.2

*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.001, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
  -Metric: MSE

*TRAINING HYPERPARAMETERS:
```

```
-Epochs: 100
-Batch size: 64
-Callbacks:
  -ReduceLROnPlateau: MSE 10 x0.1
  -Early Stop: MSE 25
```

VISUALIZATION:

```
*RESULTS:
    -Train MSE: 0.017938978970050812
    -Validation MSE: 0.055591899901628494
```



(a) Training Evolution (b) Validation datapoint (c) Train datapoint

Figure 27: Results of training the model PSFRecontructorFC10000-1

EXPERIMENT PSFRecontructorFC30000-1

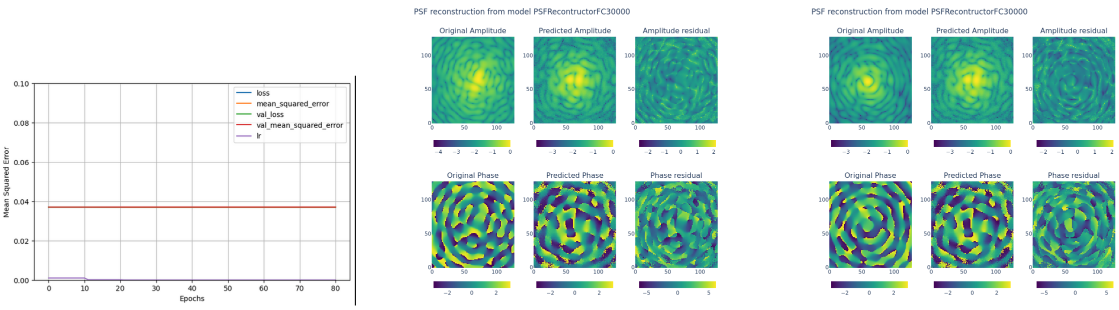
HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
  -Fully Connected
  -Input shape: 19
  -Output shape: 32768
  -Hidden layers: [256, 256, 256]
  -Regularizer: None
  -Hidden Layers Activation: relu
  -Output Layer Activation: linear
  -Batch Normalization: False
  -Dropout: False, 0.2

*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.001, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
  -Metric: MSE

*TRAINING HYPERPARAMETERS:
  -Epochs: 100
  -Batch size: 64
  -Callbacks:
    -ReduceLROnPlateau: MSE 10 x0.1
    -Early Stop: MSE 25
```

VISUALIZATION:

```
  *RESULTS:
```

```
    -Train MSE: 0.03696389123797417
    -Validation MSE: 0.03704820200800896
```



(a) Training Evolution (b) Validation datapoint (c) Train datapoint

Figure 28: Results of training the model PSFRecontructorFC30000-1

- Again, flatline, and same output for any input. I will try with all the train dataset and then start using different batch sizes, see how this works

---

EXPERIMENT PSFRecontructorFC70000-1

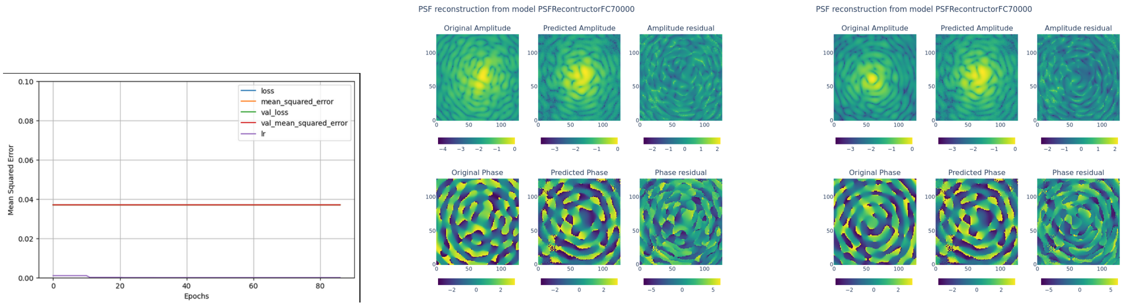
HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
  -Fully Connected
  -Input shape: 19
  -Output shape: 32768
```

```
  -Hidden layers: [256, 256, 256]
  -Regularizer: None
  -Hidden Layers Activation: relu
  -Output Layer Activation: linear
  -Batch Normalization: False
  -Dropout: False, 0.2

*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.001, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
  -Metric: MSE

*TRAINING HYPERPARAMETERS:
  -Epochs: 100
  -Batch size: 64
  -Callbacks:
    -ReduceLROnPlateau: MSE 10 x0.1
    -Early Stop: MSE 25
```

VISUALIZATION:

```
  *RESULTS:
      -Train MSE: 0.03700976073741913
      -Validation MSE: 0.0370212197303772
```



(a) Training Evolution (b) Validation datapoint (c) Train datapoint

Figure 29: Results of training the model PSFRecontructorFC70000-1

- Batch size for these experiments is 64, maybe it is too big , let's try with 32

EXPERIMENT PSFRecontructorFC10000-32-1

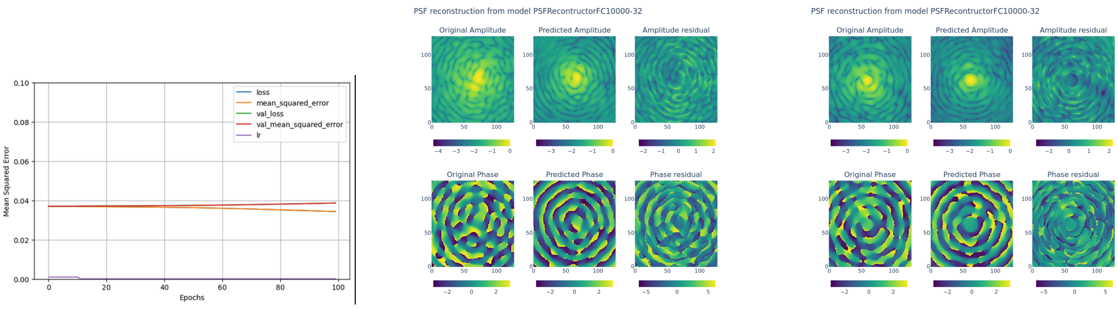
HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
  -Fully Connected
  -Input shape: 19
  -Output shape: 32768
  -Hidden layers: [256, 256, 256]
  -Regularizer: None
  -Hidden Layers Activation: relu
  -Output Layer Activation: linear
```

```
  -Batch Normalization: False
  -Dropout: False, 0.2

*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.001, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
  -Metric: MSE

*TRAINING HYPERPARAMETERS:
  -Epochs: 100
  -Batch size: 32
  -Callbacks:
    -ReduceLROnPlateau: MSE 10 x0.1
    -Early Stop: MSE 25
```

VISUALIZATION:

```
  *RESULTS:
      -Train MSE: 0.03432031720876694
      -Validation MSE: 0.038697339594364166
```



(a) Training Evolution (b) Validation datapoint (c) Train datapoint

Figure 30: Results of training the model PSFRecontructorFC10000-32-1

- Interesting, slower convergence but less overfitting

EXPERIMENT PSFRecontructorFC30000-32-1

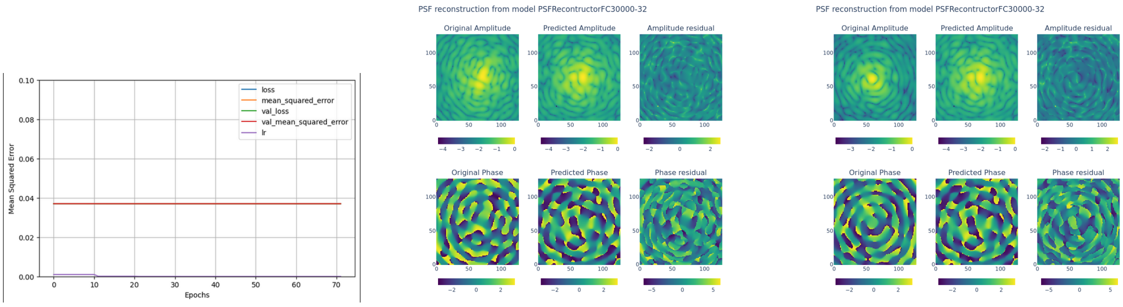
HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
  -Fully Connected
  -Input shape: 19
  -Output shape: 32768
  -Hidden layers: [256, 256, 256]
  -Regularizer: None
  -Hidden Layers Activation: relu
  -Output Layer Activation: linear
```

```
  -Batch Normalization: False
  -Dropout: False, 0.2

*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.001, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
  -Metric: MSE

*TRAINING HYPERPARAMETERS:
  -Epochs: 100
  -Batch size: 32
  -Callbacks:
    -ReduceLROnPlateau: MSE 10 x0.1
    -Early Stop: MSE 25
```

VISUALIZATION:

```
  *RESULTS:
      -Train MSE: 0.03701305016875267
      -Validation MSE: 0.03701600059866905
```



(a) Training Evolution (b) Validation datapoint (c) Train datapoint

Figure 31: Results of training the model PSFRecontructorFC30000-32-1

EXPERIMENT PSFRecontructorFC70000-32-1

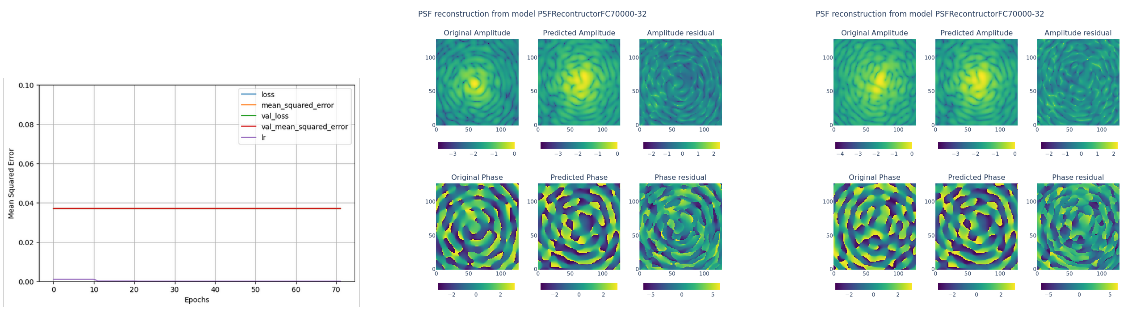
HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
  -Fully Connected
  -Input shape: 19
  -Output shape: 32768
  -Hidden layers: [256, 256, 256]
  -Regularizer: None
  -Hidden Layers Activation: relu
  -Output Layer Activation: linear
  -Batch Normalization: False
  -Dropout: False, 0.2
```

```
*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.001, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
  -Metric: MSE

*TRAINING HYPERPARAMETERS:
  -Epochs: 100
  -Batch size: 32
  -Callbacks:
    -ReduceLROnPlateau: MSE 10 x0.1
    -Early Stop: MSE 25
```

VISUALIZATION:

```
  *RESULTS:
      -Train MSE: 0.03701675683259964
      -Validation MSE: 0.037016723304986954
```



(a) Training Evolution (b) Validation datapoint (c) Train datapoint

Figure 32: Results of training the model PSFRecontructorFC70000-32-1

- So no good results, what I saw is that the lower the batch size, the longer it takes to coverge. Options now are:
  - Bigger NN
  - Bigger Batch size
  - Longer training

  I will start with 10000 datapoints and a bigger NN

EXPERIMENT PSFRecontructorBigFC10000-32-1

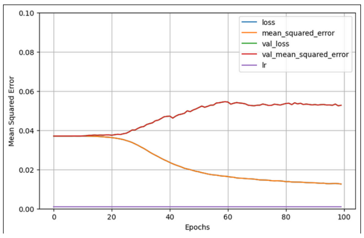
HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
  -Fully Connected
  -Input shape: 19
  -Output shape: 32768
  -Hidden layers: [256, 256, 256, 256, 256, 256]
  -Regularizer: None
  -Hidden Layers Activation: relu
  -Output Layer Activation: linear
  -Batch Normalization: False
  -Dropout: False, 0.2
```

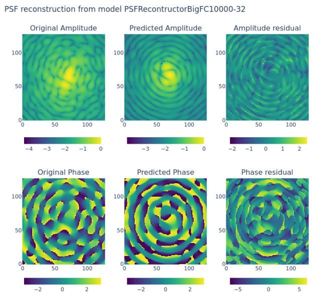
```
*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.001, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
  -Metric: MSE

*TRAINING HYPERPARAMETERS:
  -Epochs: 100
  -Batch size: 32
  -Callbacks:
    -ReduceLROnPlateau: MSE 20 x0.1
    -Early Stop: MSE 50
```

VISUALIZATION:

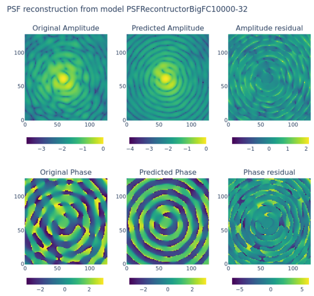
```
*RESULTS:
      -Train MSE: 0.012548761442303658
      -Validation MSE: 0.0527174137532711
```



(a) Training Evolution



(b) Validation example



(c) Train example

Figure 33: Results of training the model psf-PSFRecontructorBigFC10000-32-1

EXPERIMENT PSFRecontructorBigFC30000-32-1

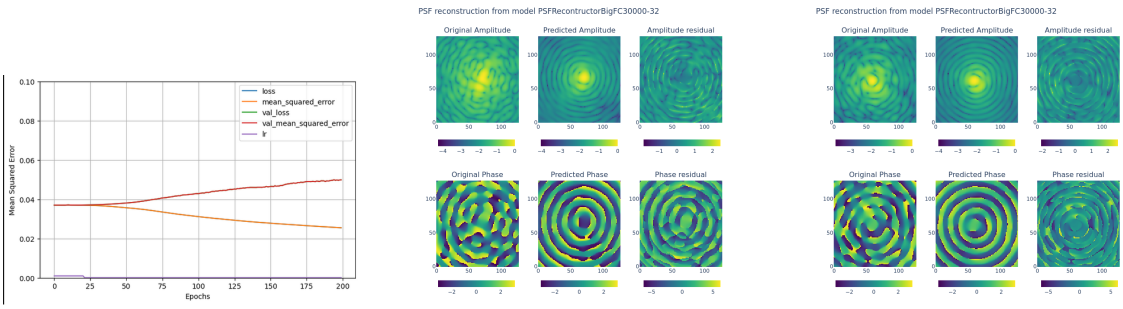
HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
  -Fully Connected
  -Input shape: 19
  -Output shape: 32768
  -Hidden layers: [256, 256, 256, 256, 256, 256]
  -Regularizer: None
  -Hidden Layers Activation: relu
  -Output Layer Activation: linear
  -Batch Normalization: False
```

```
  -Dropout: False, 0.2


*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.001, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
  -Metric: MSE


*TRAINING HYPERPARAMETERS:
  -Epochs: 200
  -Batch size: 32
  -Callbacks:
    -ReduceLROnPlateau: MSE 20 x0.1
    -Early Stop: MSE 50
```

VISUALIZATION:

```
*RESULTS:
      -Train MSE: 0.025507930666208267
      -Validation MSE: 0.04988955706357956
```



(a) Training Evolution (b) Validation example (c) Train example

Figure 34: Results of training the model PSFRecontructorBigFC30000-32-1

- With respect to the last to experiments, a bigger neural network seems to work better, with more datapoints the learning is delayed a bit, and the overfitting slightly improves, let's check with 70000 datapoints.

---

12 March 2024

EXPERIMENT PSFRecontructorBigFC70000-32-1

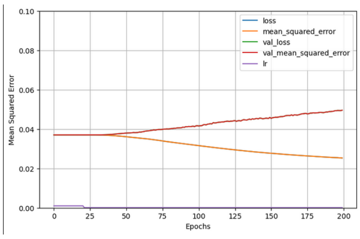
HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
  -Fully Connected
```

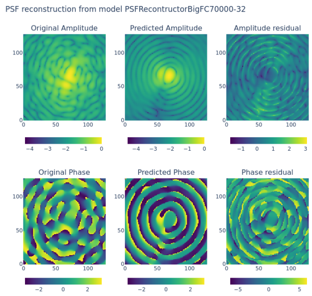
```
  -Input shape: 19
  -Output shape: 32768
  -Hidden layers: [256, 256, 256, 256, 256, 256]
  -Regularizer: None
  -Hidden Layers Activation: relu
  -Output Layer Activation: linear
  -Batch Normalization: False
  -Dropout: False, 0.2

*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.001, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
  -Metric: MSE

*TRAINING HYPERPARAMETERS:
  -Epochs: 200
  -Batch size: 32
  -Callbacks:
    -ReduceLROnPlateau: MSE 20 x0.1
    -Early Stop: MSE 50
```

VISUALIZATION:

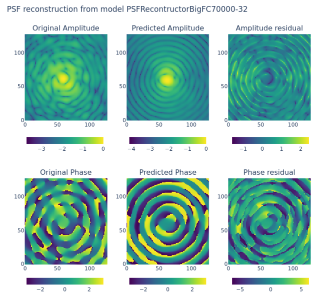
```
*RESULTS:
      -Train MSE: 0.025293584913015366
      -Validation MSE: 0.04955973103642464
```



(a) Training Evolution



(b) Validation example



(c) Train example

Figure 35: Results of training the model PSFRecontructorBigFC70000-32-1

- There is barely any difference in training with 30000 and 70000 datapoints. I think I will make a final set of experiments with an even bigger neural network before addressing the overfitting.

EXPERIMENT PSFRecontructorSuperBigFC10000-1

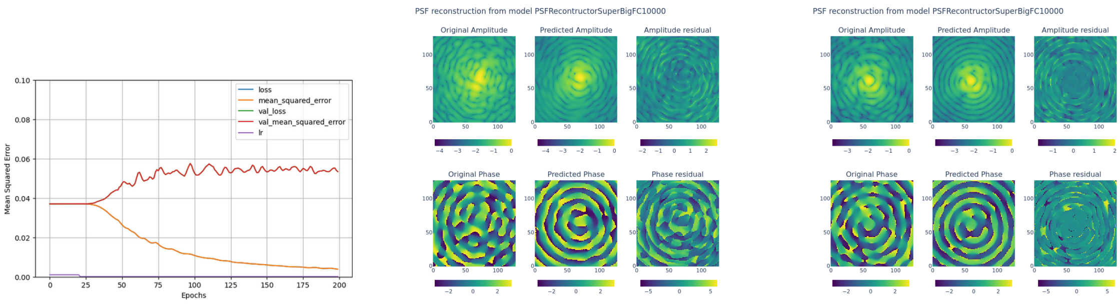
HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
  -Fully Connected
  -Input shape: 19
  -Output shape: 32768
  -Hidden layers: [1024, 1024, 1024, 1024, 1024, 1024]
  -Regularizer: None
```

```
  -Hidden Layers Activation: relu
  -Output Layer Activation: linear
  -Batch Normalization: False
  -Dropout: False, 0.2

*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.001, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
  -Metric: MSE

*TRAINING HYPERPARAMETERS:
  -Epochs: 200
  -Batch size: 32
  -Callbacks:
    -ReduceLROnPlateau: MSE 20 x0.1
    -Early Stop: MSE 50
```

VISUALIZATION:

```
*RESULTS:
      -Train MSE: 0.00384897249750793
      -Validation MSE: 0.05339088663458824
```



(a) Training Evolution (b) Validation example (c) Train example

Figure 36: Results of training the model PSFRecontructorSuperBigFC10000-1

EXPERIMENT PSFRecontructorSuperBigFC30000-1

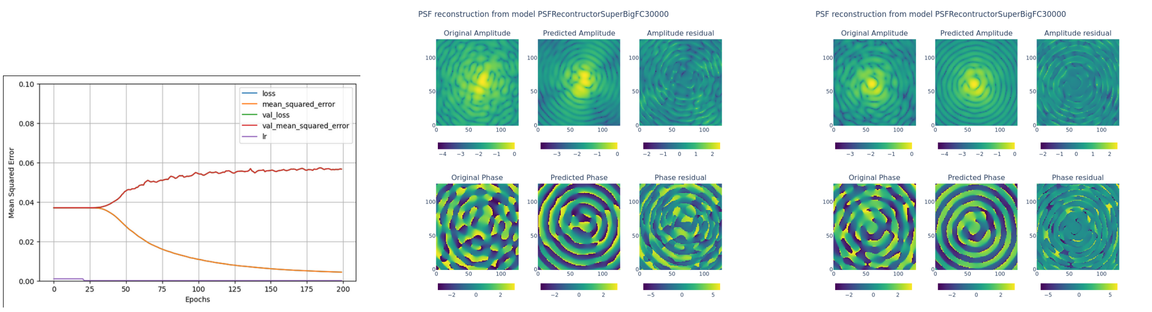
HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
  -Fully Connected
  -Input shape: 19
  -Output shape: 32768
  -Hidden layers: [1024, 1024, 1024, 1024, 1024, 1024]
  -Regularizer: None
  -Hidden Layers Activation: relu
  -Output Layer Activation: linear
  -Batch Normalization: False
  -Dropout: False, 0.2
```

```
*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.001, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
  -Metric: MSE

*TRAINING HYPERPARAMETERS:
  -Epochs: 200
  -Batch size: 32
  -Callbacks:
    -ReduceLROnPlateau: MSE 20 x0.1
    -Early Stop: MSE 50
```

VISUALIZATION:

```
*RESULTS:
      -Train MSE: 0.004385718610137701
      -Validation MSE: 0.05662728846073150б
```



(a) Training Evolution (b) Validation example (c) Train example

Figure 37: Results of training the model PSFRecontructorSuperBigFC30000-1

---

21 March 2024

EXPERIMENT PSFRecontructorSuperBigFC70000-1

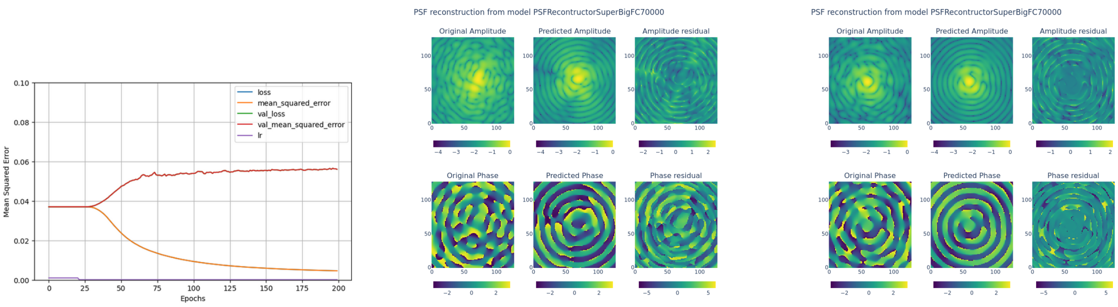
HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
  -Fully Connected
  -Input shape: 19
  -Output shape: 32768
  -Hidden layers: [1024, 1024, 1024, 1024, 1024, 1024]
  -Regularizer: None
  -Hidden Layers Activation: relu
  -Output Layer Activation: linear
  -Batch Normalization: False
  -Dropout: False, 0.2

*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.001, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
```

```
  -Metric: MSE


*TRAINING HYPERPARAMETERS:
  -Epochs: 200
  -Batch size: 32
  -Callbacks:
    -ReduceLROnPlateau: MSE 20 x0.1
    -Early Stop: MSE 50
```

VISUALIZATION:

```
*RESULTS:
      -Train MSE: 0.004607476759701967
      -Validation MSE: 0.056021399796009064
```



(a) Training Evolution (b) Validation example (c) Train example

Figure 38: Results of training the model PSFRecontructorSuperBigFC70000-1

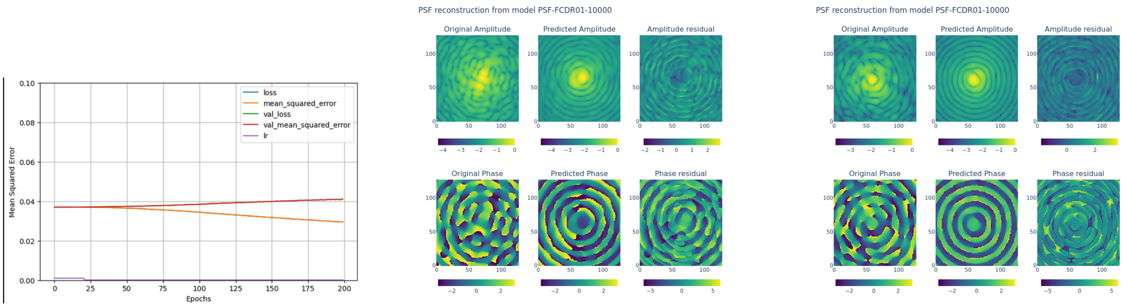
EXPERIMENT PSF-FCDR01-10000-1

HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
  -Fully Connected
  -Input shape: 19
  -Output shape: 32768
  -Hidden layers: [256, 256, 256, 256, 256, 256]
  -Regularizer: None
  -Hidden Layers Activation: relu
  -Output Layer Activation: linear
  -Batch Normalization: False
  -Dropout: True, 0.1

*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.001, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
  -Metric: MSE

*TRAINING HYPERPARAMETERS:
  -Epochs: 200
  -Batch size: 32
  -Callbacks:
    -ReduceLROnPlateau: MSE 20 x0.1
    -Early Stop: MSE 50
```

VISUALIZATION:

```
*RESULTS:
      -Train MSE: 0.029477272182703018
      -Validation MSE: 0.04117809608578682
```



(a) Training Evolution (b) Validation example (c) Train example

Figure 39: Results of training the model PSF-FCDR01-10000-1

EXPERIMENT PSF-FCBN-10000-1

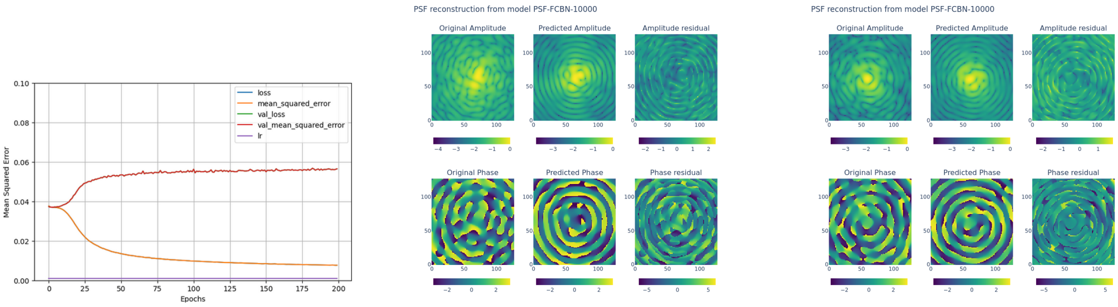
HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
    -Fully Connected
    -Input shape: 19
    -Output shape: 32768
    -Hidden layers: [256, 256, 256, 256, 256, 256]
```

```
  -Regularizer: None
  -Hidden Layers Activation: relu
  -Output Layer Activation: linear
  -Batch Normalization: True
  -Dropout: False, 0.1

*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.001, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
  -Metric: MSE

*TRAINING HYPERPARAMETERS:
  -Epochs: 200
  -Batch size: 32
  -Callbacks:
    -ReduceLROnPlateau: MSE 20 x0.1
    -Early Stop: MSE 50
```

VISUALIZATION:

```
*RESULTS:
      -Train MSE: 0.007657112553715706
      -Validation MSE: 0.05646741762757301
```



(a) Training Evolution (b) Validation example (c) Train example

Figure 40: Results of training the model PSF-FCBN-10000-1

EXPERIMENT PSF-FCL1001-10000-1

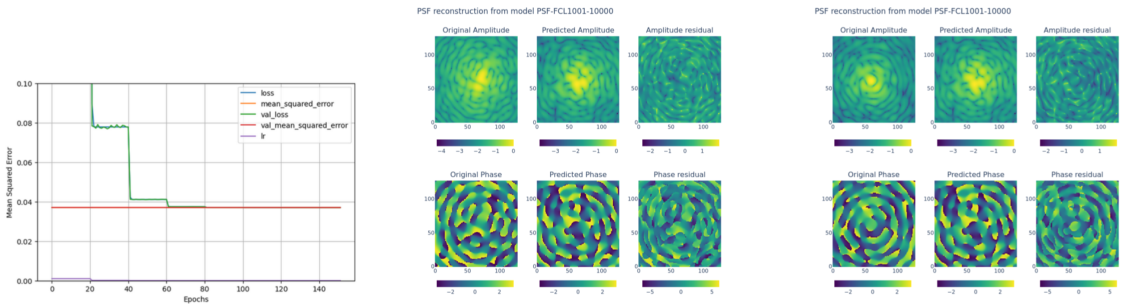
HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
  -Fully Connected
  -Input shape: 19
  -Output shape: 32768
  -Hidden layers: [256, 256, 256, 256, 256, 256]
  -Regularizer: L1 0.01
  -Hidden Layers Activation: relu
  -Output Layer Activation: linear
  -Batch Normalization: False
```

```
  -Dropout: False, 0.1


*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.001, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
  -Metric: MSE


*TRAINING HYPERPARAMETERS:
  -Epochs: 200
  -Batch size: 32
  -Callbacks:
    -ReduceLROnPlateau: MSE 20 x0.1
    -Early Stop: MSE 50
```

VISUALIZATION:

```
*RESULTS:
      -Train MSE: 0.03701375797390938
      -Validation MSE: 0.037019241601228714
```



(a) Training Evolution (b) Validation example (c) Train example

Figure 41: Results of training the model PSF-FCL1001-10000-1

EXPERIMENT PSF-FCL10001-10000-1

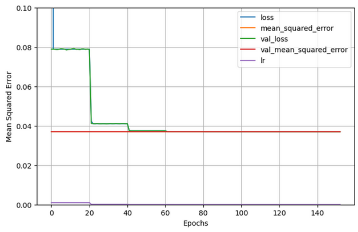
HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
  -Fully Connected
  -Input shape: 19
  -Output shape: 32768
  -Hidden layers: [256, 256, 256, 256, 256, 256]
  -Regularizer: L1 0.001
  -Hidden Layers Activation: relu
  -Output Layer Activation: linear
  -Batch Normalization: False
```

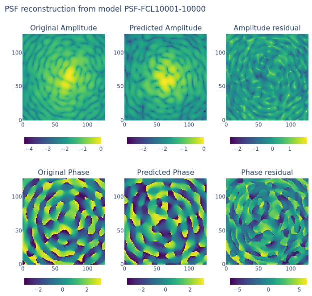
```
  -Dropout: False, 0.1

*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.001, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
  -Metric: MSE

*TRAINING HYPERPARAMETERS:
  -Epochs: 200
  -Batch size: 32
  -Callbacks:
    -ReduceLROnPlateau: MSE 20 x0.1
    -Early Stop: MSE 50
```

VISUALIZATION:

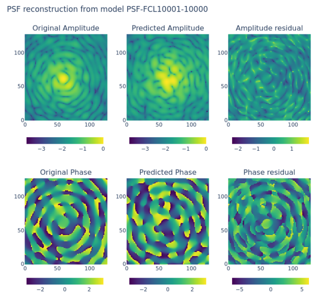
```
*RESULTS:
      -Train MSE: 0.03701375797390938
      -Validation MSE: 0.037019241601228714
```



(a) Training Evolution



(b) Validation example



(c) Train example

Figure 42: Results of training the model PSF-FCL10001-10000-1

---

22 March 2024

EXPERIMENT PSF-FCDR02-10000-2

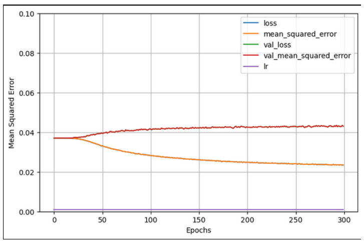
HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
  -Fully Connected
  -Input shape: 19
  -Output shape: 32768
  -Hidden layers: [256, 256, 256, 256, 256, 256]
```

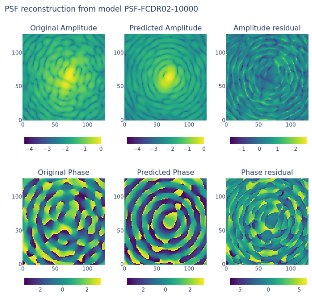
```
  -Regularizer: None
  -Hidden Layers Activation: relu
  -Output Layer Activation: linear
  -Batch Normalization: False
  -Dropout: True, 0.2

*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.001, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
  -Metric: MSE

*TRAINING HYPERPARAMETERS:
  -Epochs: 300
  -Batch size: 32
  -Callbacks:
    -ReduceLROnPlateau: MSE 50 x0.1
    -Early Stop: MSE 70
```

VISUALIZATION:

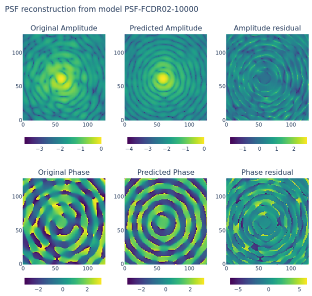
```
*RESULTS:
      -Train MSE: 0.0234194565564394
      -Validation MSE: 0.043008171021938324
```



(a) Training Evolution



(b) Validation example



(c) Train example

Figure 43: Results of training the model PSF-FCDR02-10000-2

EXPERIMENT PSF-FCDR02-30000-1

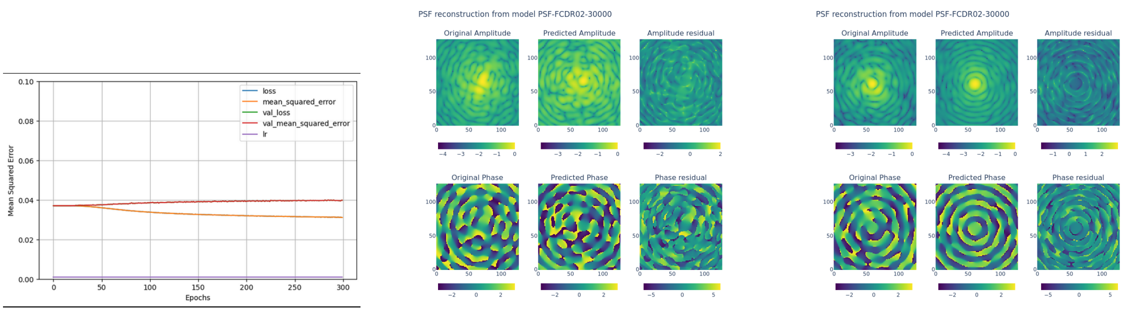
HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
  -Fully Connected
  -Input shape: 19
  -Output shape: 32768
  -Hidden layers: [256, 256, 256, 256, 256, 256]
  -Regularizer: None
  -Hidden Layers Activation: relu
  -Output Layer Activation: linear
  -Batch Normalization: False
  -Dropout: True, 0.2
```

```
*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.001, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
  -Metric: MSE

*TRAINING HYPERPARAMETERS:
  -Epochs: 300
  -Batch size: 32
  -Callbacks:
    -ReduceLROnPlateau: MSE 20 x0.1
    -Early Stop: MSE 50
```

VISUALIZATION:

```
*RESULTS:
      -Train MSE: 0.03109411522746086
      -Validation MSE: 0.03979235887527466
```



(a) Training Evolution (b) Validation example (c) Train example

Figure 44: Results of training the model PSF-FCDR02-30000-1

EXPERIMENT PSF-FCDR02-70000-2

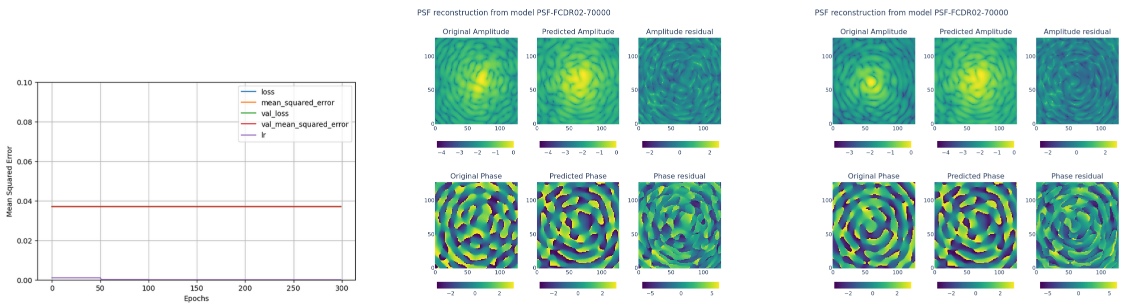
HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
  -Fully Connected
  -Input shape: 19
  -Output shape: 32768
  -Hidden layers: [256, 256, 256, 256, 256, 256]
  -Regularizer: None
  -Hidden Layers Activation: relu
  -Output Layer Activation: linear
  -Batch Normalization: False
  -Dropout: True, 0.2

*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.001, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
  -Metric: MSE

*TRAINING HYPERPARAMETERS:
  -Epochs: 300
  -Batch size: 32
```

```
-Callbacks:
  -ReduceLROnPlateau: MSE 20 x0.1
  -Early Stop: MSE 50
```

VISUALIZATION:

```
*RESULTS:
    -Train MSE: 0.037016715854406 36
    -Validation MSE: 0.037016704678535 46
```



(a) Training Evolution (b) Validation example (c) Train example

Figure 45: Results of training the model PSF-FCDR02-30000-1

---

25 March 2024

- Regularizations:
  - Dropout, the best results altough not enough

- Batch normalization makes the convergence faster but makes overfitting worse
- Kernel regularizers, they don't help, the model does not learn.

- Apart from the overfitting, the predictions made from the training samples are accurate but just in the center of the pupil, as we go away from the center the ripples become more regular

- My guess is either we need more fibers on the PL or a lower resolution image

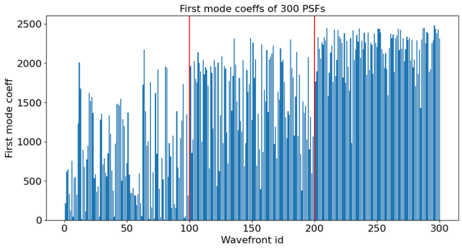
---

26 March 2024

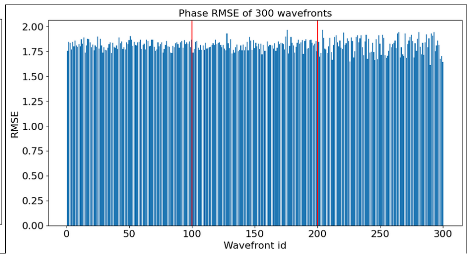
- Perform a correlation analysis: In the figures the red line indicate a change in the fried parameter, from left to right it incresases from 0.1 to 0.2 and then 0.4. There are 100 wavefronts per fried parameter, it looks like the data correlates in all the graphs except for the phase RMSE which does not vary too much

---

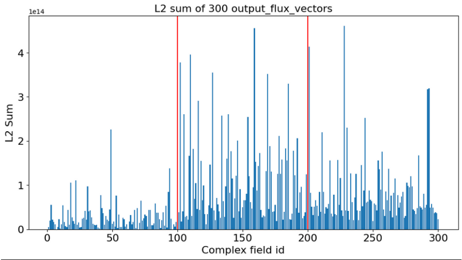
CORRELATION ANALYSIS



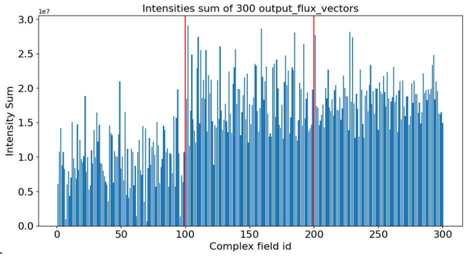
(a) First mode coefficients



(b) Wavefront RMSE



(c) PSF L2 sum



(d) PSF Intensity sum

Figure 46: Correlation analysis

---

02 April 2024

EXPERIMENT CroppedSimpleFC10000-1

HYPERPARAMETERS:

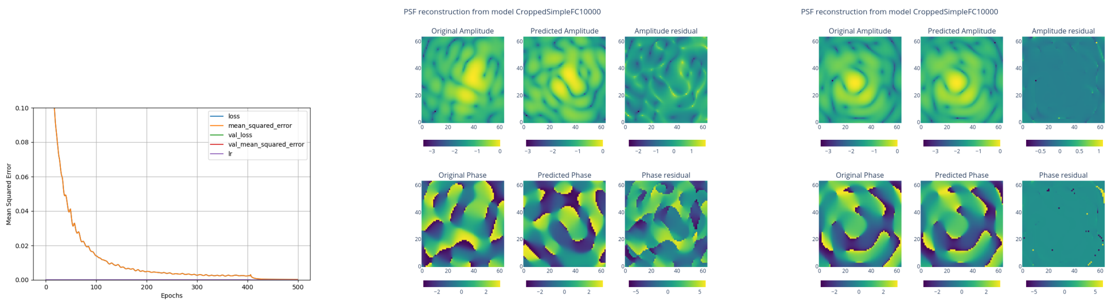
```
*ARCHITECTURE HYPERPARAMETERS:
```

```
  -Fully Connected
  -Input shape: 19
  -Output shape: 8192
  -Hidden layers: [1024, 1024, 1024, 1024, 1024, 1024]
  -Regularizer: None
  -Hidden Layers Activation: relu
  -Output Layer Activation: linear
  -Batch Normalization: False
  -Dropout: False, 0.2

*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.0001, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
  -Metric: MSE

*TRAINING HYPERPARAMETERS:
  -Epochs: 500
  -Samples: 10000
  -Batch size: 32
  -Callbacks:
    -ReduceLROnPlateau: MSE 50 x0.1
    -Early Stop: MSE 70
```

VISUALIZATION:

```
*RESULTS:
```

```
    -Train MSE: 0.00028163602109998465
    -Validation MSE: 0.1800803393125534
```



(a) Training Evolution (b) Validation example (c) Train example

Figure 47: Results of training the model CroppedSimpleFC10000-1

EXPERIMENT CroppedSimpleFC30000-1

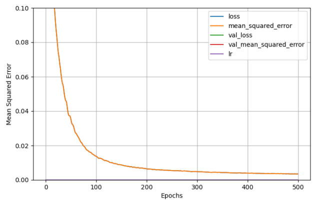
HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
  -Fully Connected
  -Input shape: 19
  -Output shape: 8192
  -Hidden layers: [1024, 1024, 1024, 1024, 1024, 1024]
  -Regularizer: None
  -Hidden Layers Activation: relu
```

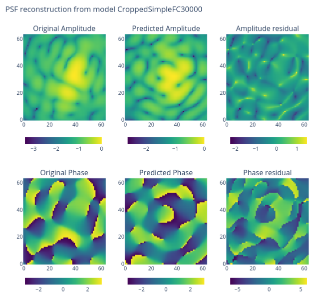
```
  -Output Layer Activation: linear
  -Batch Normalization: False
  -Dropout: False, 0.2

*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.0001, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
  -Metric: MSE

*TRAINING HYPERPARAMETERS:
  -Epochs: 500
  -Samples: 30000
  -Batch size: 32
  -Callbacks:
    -ReduceLROnPlateau: MSE 50 x0.1
    -Early Stop: MSE 70
```

VISUALIZATION:

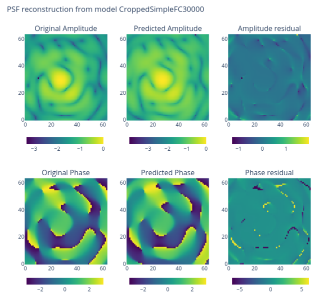
```
*RESULTS:
      -Train MSE: 0.0034643723629415035
      -Validation MSE: 0.1996324509382248
```



(a) Training Evolution



(b) Validation example



(c) Train example

Figure 48: Results of training the model CroppedSimpleFC30000-1

EXPERIMENT CroppedSimpleFC70000-1

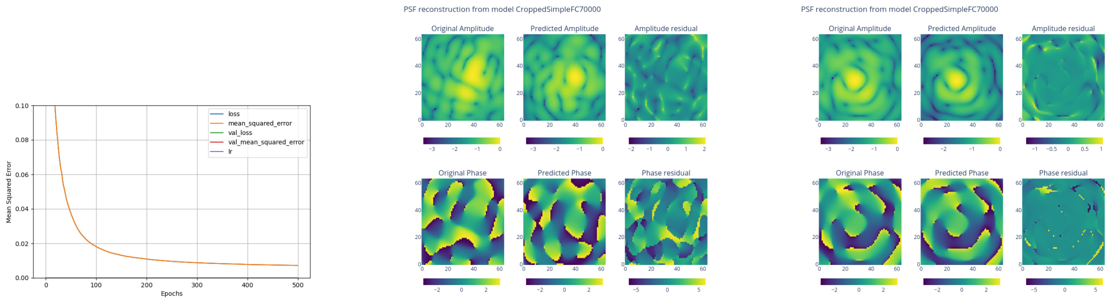
HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
  -Fully Connected
  -Input shape: 19
  -Output shape: 8192
  -Hidden layers: [1024, 1024, 1024, 1024, 1024, 1024]
  -Regularizer: None
  -Hidden Layers Activation: relu
  -Output Layer Activation: linear
  -Batch Normalization: False
```

```
  -Dropout: False, 0.2


*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.0001, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
  -Metric: MSE


*TRAINING HYPERPARAMETERS:
  -Epochs: 500
  -Samples: 30000
  -Batch size: 32
  -Callbacks:
    -ReduceLROnPlateau: MSE 50 x0.1
    -Early Stop: MSE 70
```

VISUALIZATION:

```
*RESULTS:
        -Train MSE: 0.007220898289233446
        -Validation MSE: 0.20809102058410645
```



(a) Training Evolution (b) Validation example (c) Train example

Figure 49: Results of training the model CroppedSimpleFC70000-1

EXPERIMENT CroppedDR01FC10000-1

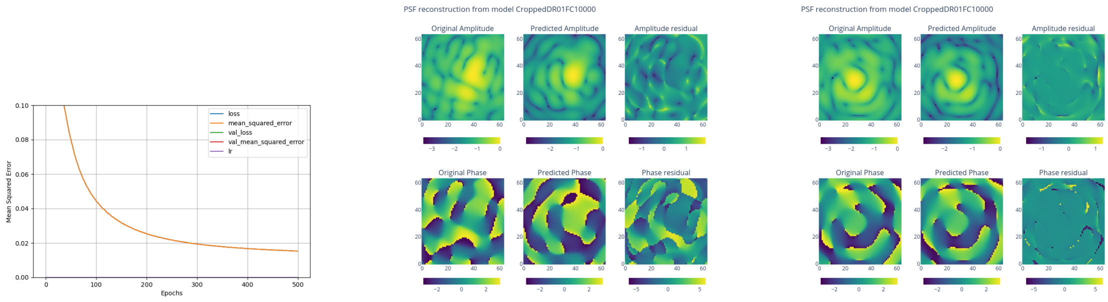
HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
  -Fully Connected
  -Input shape: 19
  -Output shape: 8192
  -Hidden layers: [1024, 1024, 1024, 1024, 1024, 1024]
  -Regularizer: None
  -Hidden Layers Activation: relu
  -Output Layer Activation: linear
  -Batch Normalization: False
```

```
  -Dropout: True, 0.1


*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.0001, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
  -Metric: MSE


*TRAINING HYPERPARAMETERS:
  -Epochs: 500
  -Samples: 10000
  -Batch size: 32
  -Callbacks:
    -ReduceLROnPlateau: MSE 50 x0.1
    -Early Stop: MSE 70
```

VISUALIZATION:

```
*RESULTS:
      -Train MSE: 0.0152916694059968
      -Validation MSE: 0.18240582942962646
```



(a) Training Evolution (b) Validation example (c) Train example

Figure 50: Results of training the model CroppedDR01FC10000-1

EXPERIMENT CroppedDR01FC30000-1

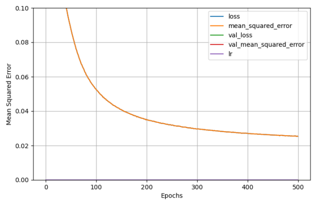
HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
  -Fully Connected
  -Input shape: 19
  -Output shape: 8192
  -Hidden layers: [1024, 1024, 1024, 1024, 1024, 1024]
  -Regularizer: None
  -Hidden Layers Activation: relu
  -Output Layer Activation: linear
  -Batch Normalization: False
```

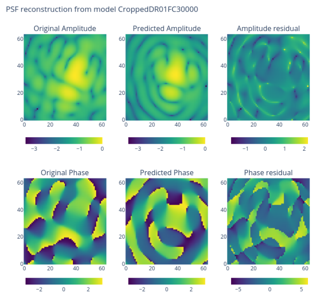
```
  -Dropout: True, 0.1

*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.0001, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
  -Metric: MSE

*TRAINING HYPERPARAMETERS:
  -Epochs: 500
  -Samples: 30000
  -Batch size: 32
  -Callbacks:
    -ReduceLROnPlateau: MSE 50 x0.1
    -Early Stop: MSE 70
```

VISUALIZATION:

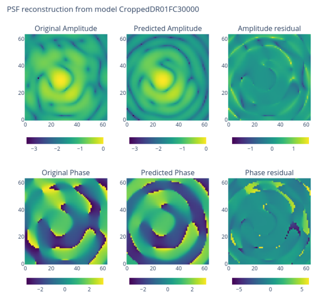
```
*RESULTS:
      -Train MSE: 0.025341492146253586
      -Validation MSE: 0.18506889045238495
```



(a) Training Evolution



(b) Validation example



(c) Train example

Figure 51: Results of training the model CroppedDR01FC30000-1

EXPERIMENT CroppedDR01FC70000-1

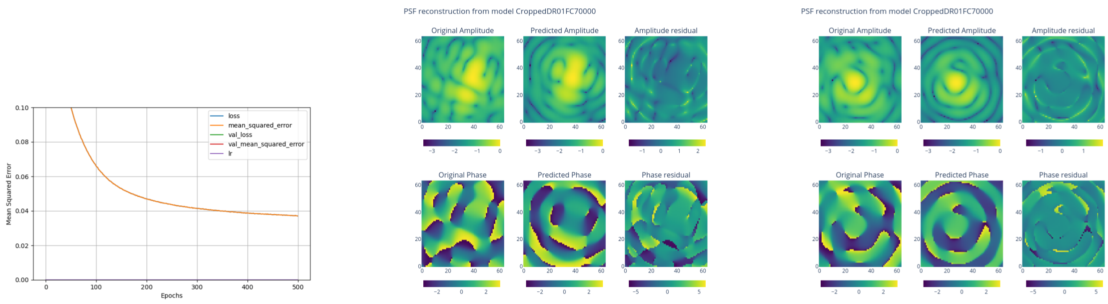
HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
  -Fully Connected
  -Input shape: 19
  -Output shape: 8192
  -Hidden layers: [1024, 1024, 1024, 1024, 1024, 1024]
  -Regularizer: None
  -Hidden Layers Activation: relu
  -Output Layer Activation: linear
  -Batch Normalization: False
```

```
  -Dropout: True, 0.1


*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.0001, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
  -Metric: MSE


*TRAINING HYPERPARAMETERS:
  -Epochs: 500
  -Samples: 70000
  -Batch size: 32
  -Callbacks:
    -ReduceLROnPlateau: MSE 50 x0.1
    -Early Stop: MSE 70
```

VISUALIZATION:

```
*RESULTS:
      -Train MSE: 0.03703419491648674
      -Validation MSE: 0.18717476725578308
```



(a) Training Evolution (b) Validation example (c) Train example

Figure 52: Results of training the model CroppedDR01FC70000-1

EXPERIMENT CroppedDR02FC10000-1

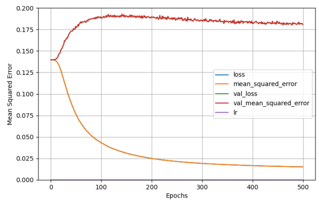
HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
  -Fully Connected
  -Input shape: 19
  -Output shape: 8192
  -Hidden layers: [1024, 1024, 1024, 1024, 1024, 1024]
  -Regularizer: None
  -Hidden Layers Activation: relu
  -Output Layer Activation: linear
  -Batch Normalization: False
```

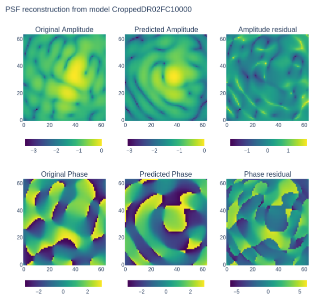
```
  -Dropout: True, 0.2

*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.0001, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
  -Metric: MSE

*TRAINING HYPERPARAMETERS:
  -Epochs: 500
  -Samples: 30000
  -Batch size: 32
  -Callbacks:
    -ReduceLROnPlateau: MSE 50 x0.1
    -Early Stop: MSE 70
```

VISUALIZATION:

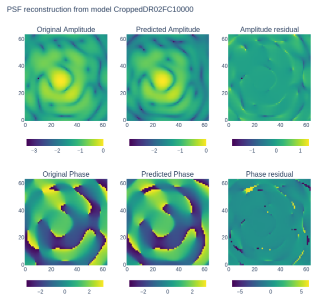
```
*RESULTS:
      -Train MSE: 0.015202851034700870
      -Validation MSE: 0.18062736093997955
```



(a) Training Evolution



(b) Validation example



(c) Train example

Figure 53: Results of training the model CroppedDR02FC10000-1

EXPERIMENT CroppedDR02FC30000-1

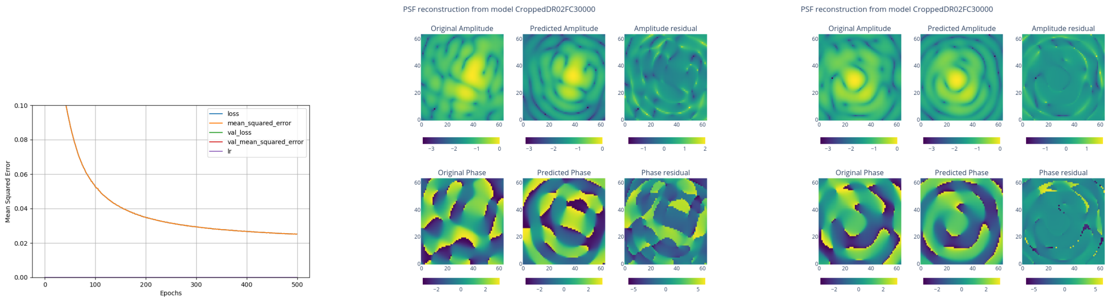
HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
  -Fully Connected
  -Input shape: 19
  -Output shape: 8192
  -Hidden layers: [1024, 1024, 1024, 1024, 1024, 1024]
  -Regularizer: None
  -Hidden Layers Activation: relu
  -Output Layer Activation: linear
  -Batch Normalization: False
```

```
  -Dropout: True, 0.2


*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.0001, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
  -Metric: MSE


*TRAINING HYPERPARAMETERS:
  -Epochs: 500
  -Samples: 30000
  -Batch size: 32
  -Callbacks:
    -ReduceLROnPlateau: MSE 50 x0.1
    -Early Stop: MSE 70
```

VISUALIZATION:

```
*RESULTS:
      -Train MSE: 0.025196749716997147
      -Validation MSE: 0.1864871084690094
```



(a) Training Evolution (b) Validation example (c) Train example

Figure 54: Results of training the model CroppedDR02FC30000-1

EXPERIMENT CroppedDR02FC70000-1

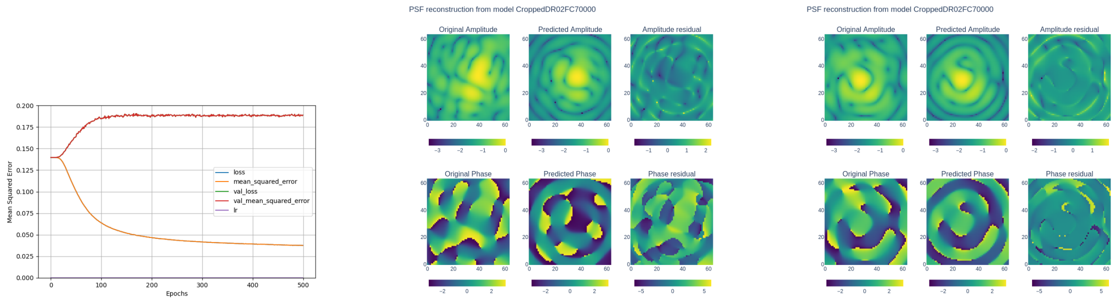
HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
  -Fully Connected
  -Input shape: 19
  -Output shape: 8192
  -Hidden layers: [1024, 1024, 1024, 1024, 1024, 1024]
  -Regularizer: None
  -Hidden Layers Activation: relu
  -Output Layer Activation: linear
  -Batch Normalization: False
```

```
  -Dropout: True, 0.2


*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.0001, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
  -Metric: MSE


*TRAINING HYPERPARAMETERS:
  -Epochs: 500
  -Samples: 30000
  -Batch size: 32
  -Callbacks:
    -ReduceLROnPlateau: MSE 50 x0.1
    -Early Stop: MSE 70
```

VISUALIZATION:

```
*RESULTS:
      -Train MSE: 0.03758401796221733
      -Validation MSE: 0.1887626349925995
```



(a) Training Evolution (b) Validation example (c) Train example

Figure 55: Results of training the model CroppedDR02FC70000-1

EXPERIMENT CroppedBNFC10000-1

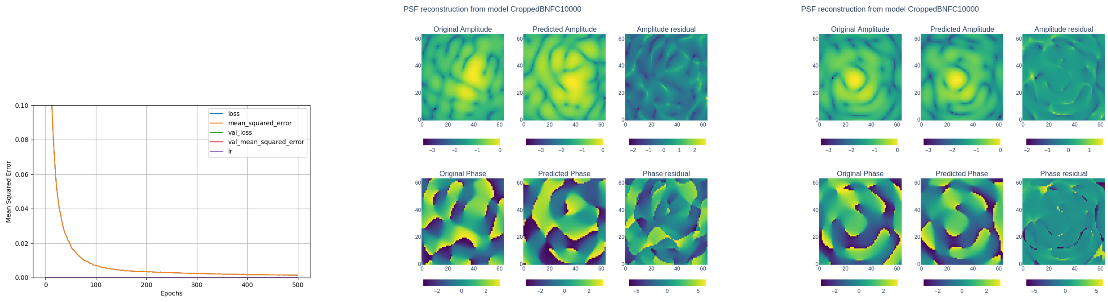
HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
  -Fully Connected
  -Input shape: 19
  -Output shape: 8192
  -Hidden layers: [1024, 1024, 1024, 1024, 1024, 1024]
  -Regularizer: None
  -Hidden Layers Activation: relu
  -Output Layer Activation: linear
  -Batch Normalization: True
```

```
  -Dropout: False, 0.2


*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.0001, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
  -Metric: MSE


*TRAINING HYPERPARAMETERS:
  -Epochs: 500
  -Samples: 10000
  -Batch size: 32
  -Callbacks:
    -ReduceLROnPlateau: MSE 50 x0.1
    -Early Stop: MSE 70
```

VISUALIZATION:

```
*RESULTS:
      -Train MSE: 0.001437550876289606
      -Validation MSE: 0.18398649990558624
```



(a) Training Evolution (b) Validation example (c) Train example

Figure 56: Results of training the model CroppedBNFC10000-1

EXPERIMENT CroppedBNFC30000-1

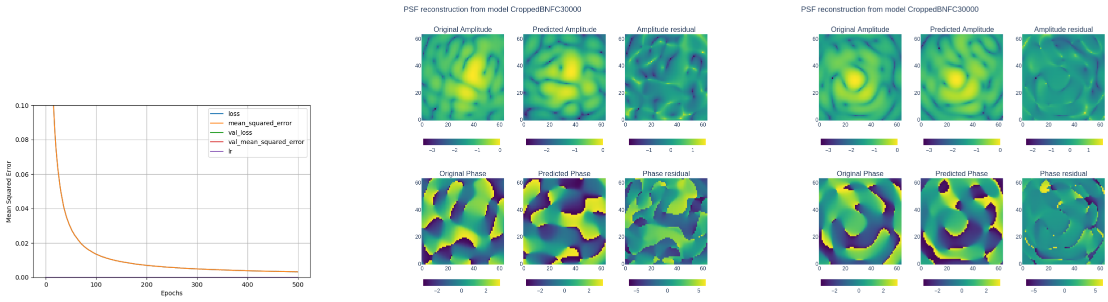
HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
  -Fully Connected
  -Input shape: 19
  -Output shape: 8192
  -Hidden layers: [1024, 1024, 1024, 1024, 1024, 1024]
  -Regularizer: None
  -Hidden Layers Activation: relu
  -Output Layer Activation: linear
  -Batch Normalization: True
```

```
  -Dropout: False, 0.2


*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.0001, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
  -Metric: MSE


*TRAINING HYPERPARAMETERS:
  -Epochs: 500
  -Samples: 30000
  -Batch size: 32
  -Callbacks:
    -ReduceLROnPlateau: MSE 50 x0.1
    -Early Stop: MSE 70
```

VISUALIZATION:

```
*RESULTS:
      -Train MSE: 0.0032125315628945827
      -Validation MSE: 0.1955052763223648
```



(a) Training Evolution (b) Validation example (c) Train example

Figure 57: Results of training the model CroppedBNFC30000-1

EXPERIMENT CroppedBNFC70000-1

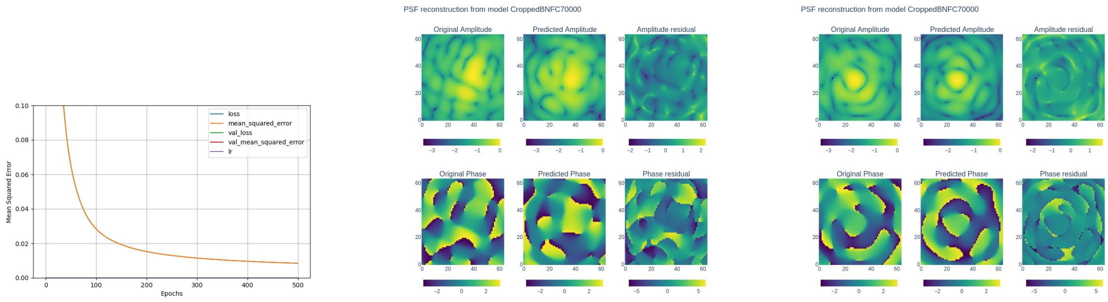
HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
  -Fully Connected
  -Input shape: 19
  -Output shape: 8192
  -Hidden layers: [1024, 1024, 1024, 1024, 1024, 1024]
  -Regularizer: None
  -Hidden Layers Activation: relu
  -Output Layer Activation: linear
  -Batch Normalization: True
```

```
  -Dropout: False, 0.2


*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.0001, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
  -Metric: MSE


*TRAINING HYPERPARAMETERS:
  -Epochs: 500
  -Samples: 70000
  -Batch size: 32
  -Callbacks:
    -ReduceLROnPlateau: MSE 50 x0.1
    -Early Stop: MSE 70
```

VISUALIZATION:

```
*RESULTS:
      -Train MSE: 0.008466990664601326
      -Validation MSE: 0.20970138907432556
```



(a) Training Evolution (b) Validation example (c) Train example

Figure 58: Results of training the model CroppedBNFC70000-1

---

09 April 2024

- PL information determination

- Given some set of data containing the inputs (eg mode coeffs or images) X and outputs (eg PL fluxes) y of a photonic lantern, quantify the amount of information preserved by the encoding, independent of any assumptions about the transfer function or reconstruction algorithm.

- First create files with the predictions from the models `PSFRecontructorSuperBigFC70000-1` (for the original sized complex fields predictions) and `CroppedBNFC70000-1` (for the cropped complex fields predictions). The predictions are made for the whole train dataset.

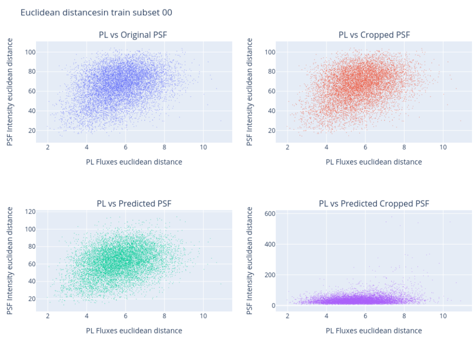
- **BRUTE FORCE ANALYSIS**:

  - Pick 70000 random pair of frames
  - Save euclidean distances for each of them
  - Plot euclidean distances
  - See if PSF similarity implies PL similarity
  - Metric will be the ratio between PL output distances and PSF distances
  - Perform ANOVA test for Uncropped, Cropped, PredictedUncropped, PredictedCropped pairs.

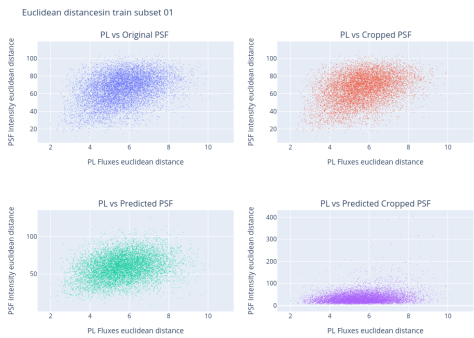
- I compute the euclidean distances between pairs and store the pairs in `pairsxx.npy` and `euclidean_distancesxx.npy`. The `euclidean_distancesxx.npy` array has shape of 1000x5, each column being the distance between pairs of psf fluxes, original complex fields, cropped complex fields, predicted complex fields and predicted cropped complex fields.

- The complex fields are 2x128x128 and the cropped complex fields are 2x64x64, when I say predicted I mean the complex generated by the models.

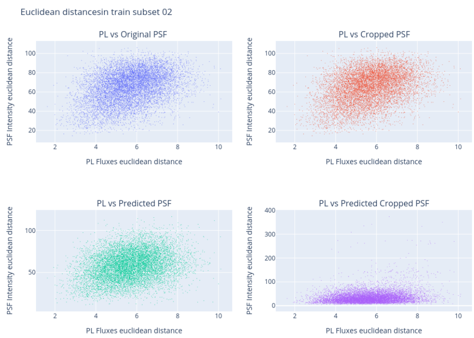
EXPERIMENT Euclidean distances



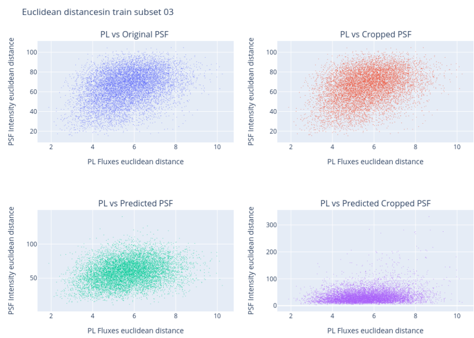
(a) Train subset 0



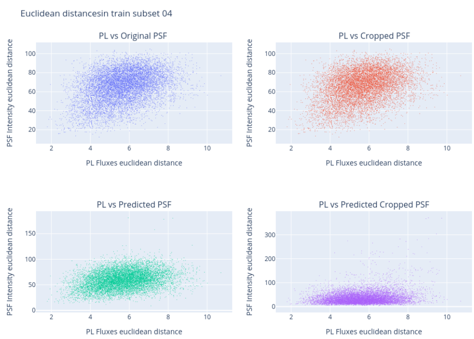
(b) Train subset 1



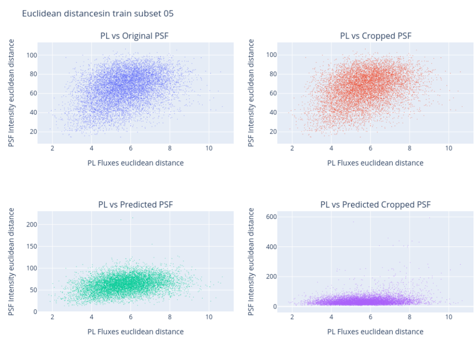
(c) Train subset 2



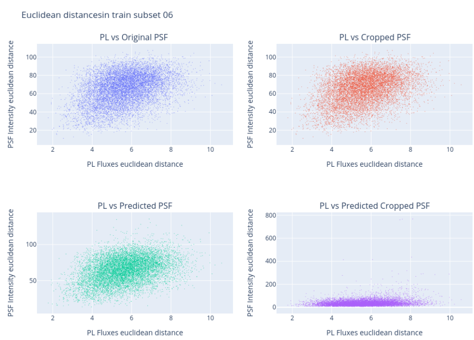
(d) Train subset 3



(e) Train subset 4



(f) Train subset 5



(g) Train subset 6

Figure 59: Euclidean distances between PL and PSF pairs

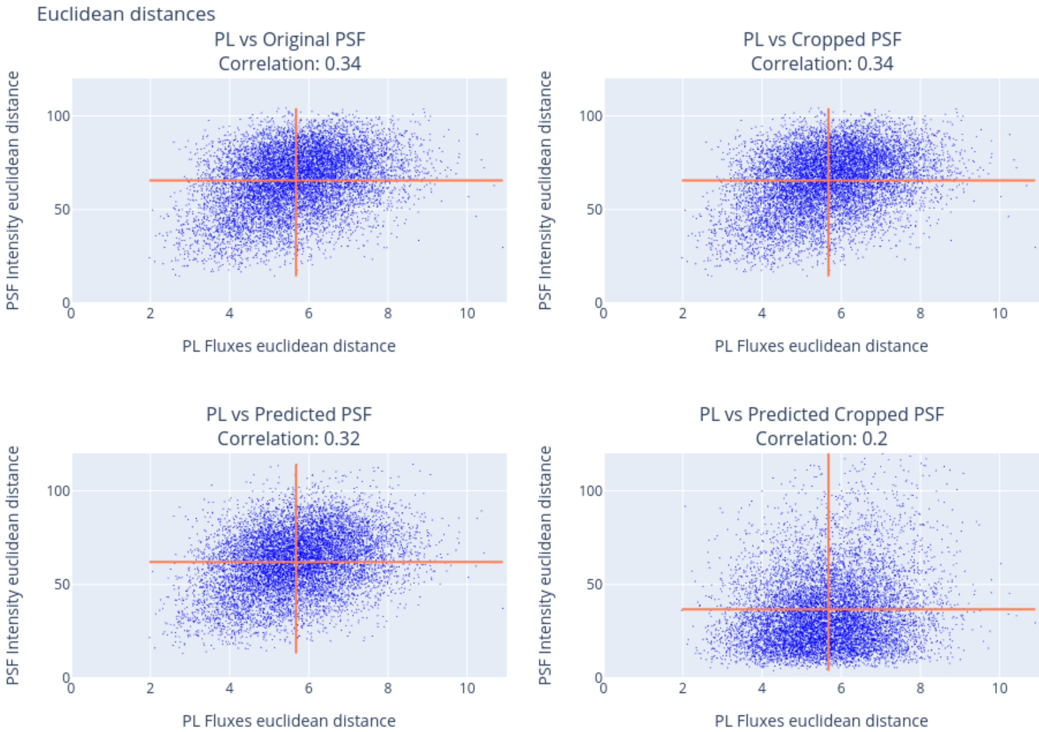
10 April 2024

- Repeat the same analysis but this time for the whole train dataset

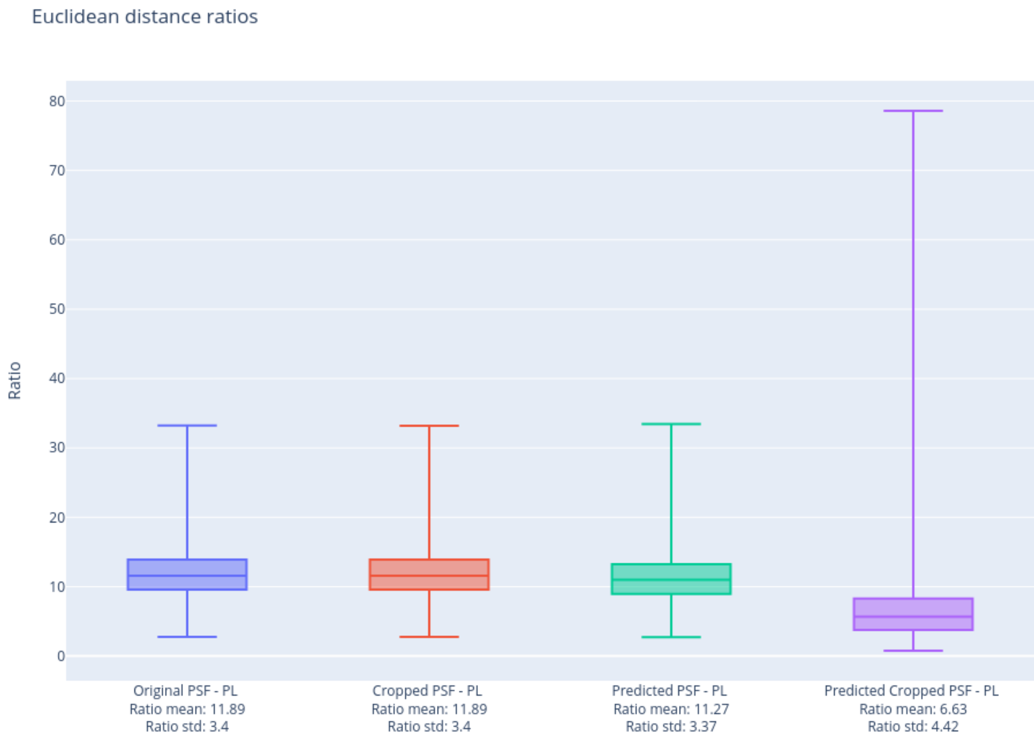
- The correlation is 0.3 which indicates a slightly positive linear relationship between the PL flux and PSF in all caes except for the cropped predictions which has a 0.2 correlation ratex.

  The cropped predictions seem to be closer between them than in the rest of the cases, the model that predicts the original sized PSFs has a very similar cloud structure to the one composed of the PL and original dataset.

EXPERIMENT Euclidean distances for the whole dataset



(a) Euclidean distance cloud



(b) Euclidean distance ratios

Figure 60: Euclidean distances ratios between PL and PSF pairs

---

---

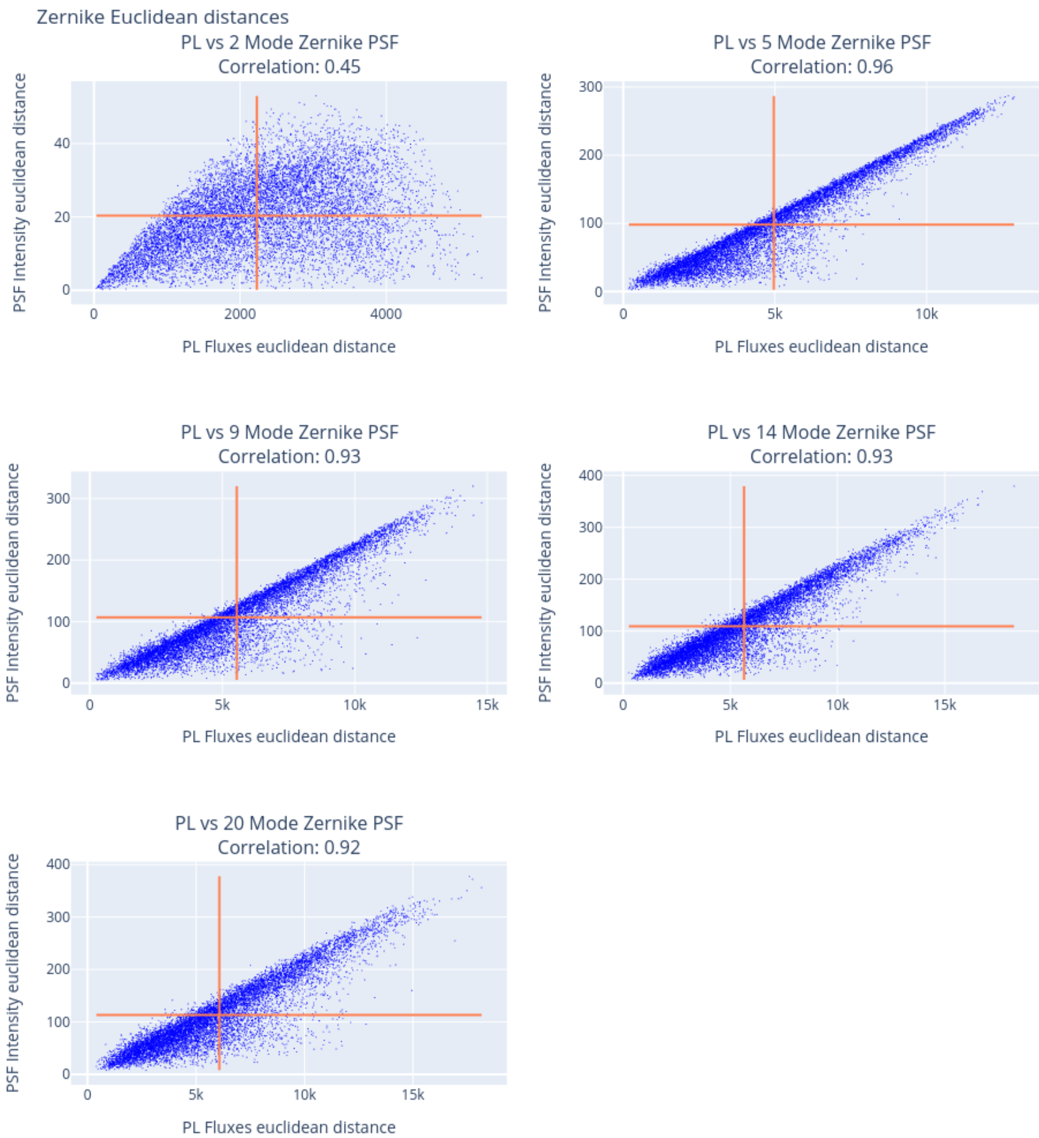
19 April 2024

- Create low order zernike psf aberration and repeat the same analysis

---

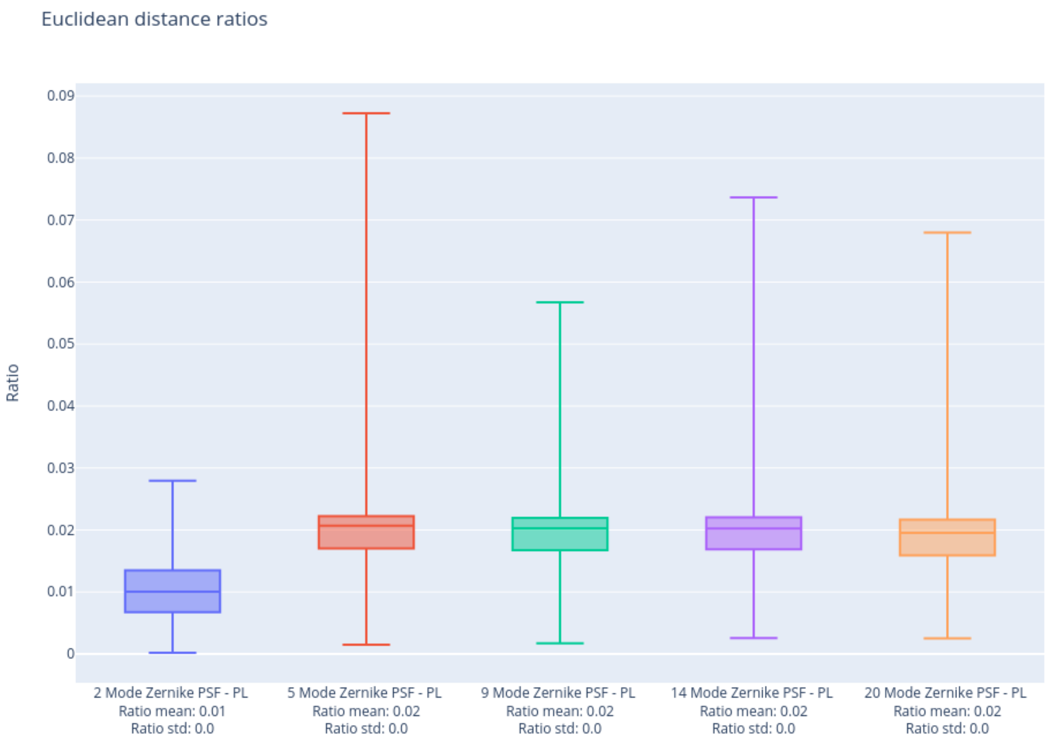
EXPERIMENT Euclidean distances for the whole dataset

Anova Test p value is 0.0



(a) Euclidean distance cloud

Figure 61: Euclidean distances ratios between PL and Zernike PSF pairs



(a) Euclidean distance ratios

Figure 62: Euclidean distances ratios between PL and Zernike PSF pairs

Part III

PL Information Determination

0.1 The data

There are two groups of datasets.

0.1.1 Atmospheric aberration related

There are 4 datasets composed by PSFs and their corresponding PL intensities.

PSFs The PSFs' electric fields are stored in a 3d matrix of depth 2: depth 1 and 2 represent the real and imaginary value of the electric field in a point.

- **Original sized PSFs**: Two datasets of 70000 electric fields and corresponding intensities stored in 128x128x2 and 128x128 matrices respectively.

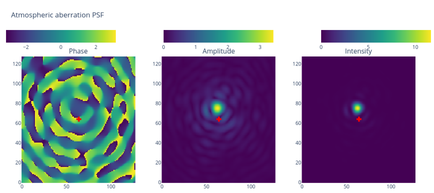


Figure 63: Example original sized PSF

- **Cropped sized PSFs**: Two datasets of 70000 electric fields and corresponding intensities stored in 64x64x2 and 64x64 matrices respectively. These cropped PSFs correspond to the central pixels from the Original sized PSFs.

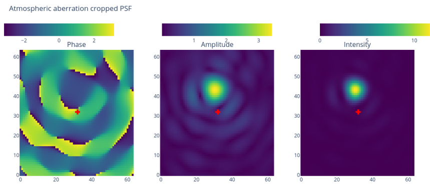


Figure 64: Example Cropped sized PSF

- **Original sized predicted PSFs**: Two datasets of 70000 predicted electric fields and predicted intensities stored in 128x128x2 and 128x128 matrices respectively. These predicted PSFs are the outputs of a model trained with the Original PSFs dataset and their corresponding PL intensities.

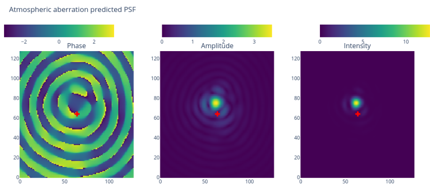


Figure 65: Example original sized predicted PSF

- **Cropped sized predicted PSF**: Two datasets of 70000 predicted electric fields and predicted intensities stored in 64x64x2 and 64x64 matrices respectively. These cropped predicted PSFs are the outputs of a model trained with the Cropped sized PSFs dataset and their corresponding PL intensities (which are the same ouput intensities from the Original sized PSFs dataset).

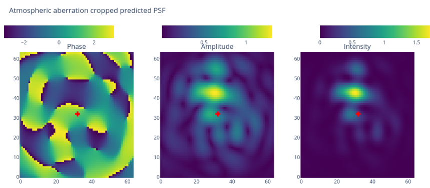


Figure 66: Example cropped sized predicted PSF

PL intensities The same dataset of PL output intensities are used for every PSF dataset. The intensities are computed multiplying the LP coefficients by the transfer matrix of the **19 mode PL**. This dataset has 70000 datapoints, each datapoint being a vector of 19 elements.

0.1.2 Zernike modes related

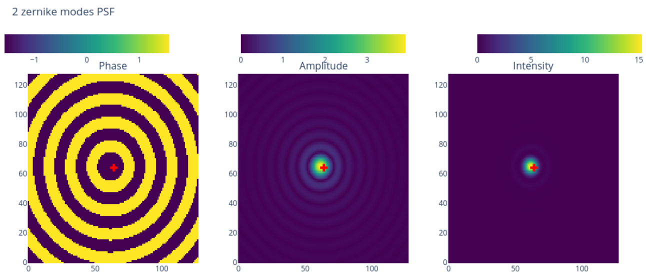
There are 5 subgroups of datasets: PSFs generated with 2, 5, 9, 14 and 20 zernike modes. Each subgroup is divided in original sized, cropped sized, predicted and cropped predicted as in the case of the atmospheric aberration PSFs.

2 Zernike modes PSFs

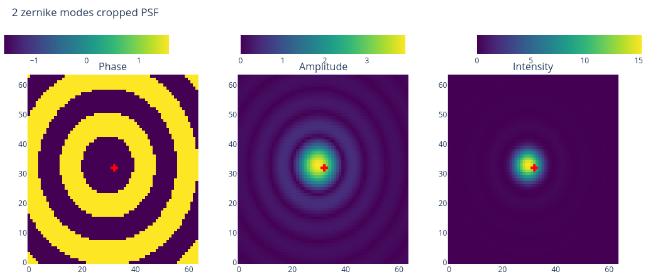
- **Original sized 2 modes PSFs**: Two datasets of 70000 electric fields and corresponding intensities stored in 128x128x2 and 128x128 matrices respectively. The aberration by a 2 modes zernike basis.
- **Cropped sized 2 modes PSFs**: Two datasets of 70000 electric fields and corresponding intensities stored in 64x64x2 and 64x64 matrices respectively. These cropped PSFs correspond to the central pixels from the Original sized 2 modes PSFs.
- **Original sized predicted 2 modes PSFs**: Two datasets of 70000 predicted electric fields and predicted intensities stored in 128x128x2 and 128x128 matri-

ces respectively. These predicted PSFs are the outputs of a model trained with the Original sized 2 modes PSFs dataset and their corresponding PL intensities.

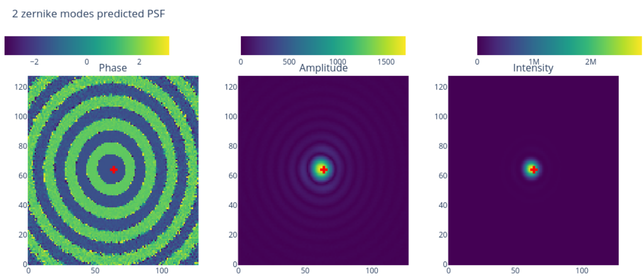
- **Cropped sized predicted 2 modes PSF**: Two datasets of 70000 predicted electric fields and predicted intensities stored in 64x64x2 and 64x64 matrices respectively. These cropped predicted PSFs are the outputs of a model trained with the Cropped sized 2 modes PSFs dataset and their corresponding PL intensities (which are the same ouput intensities from the Original sized 2 modes PSFs dataset).



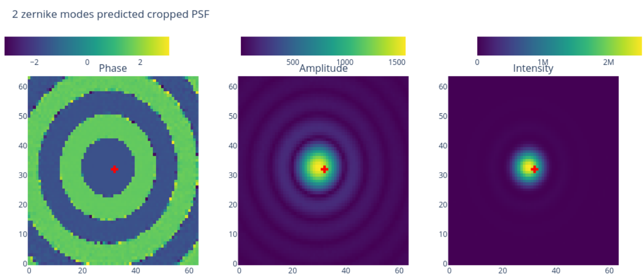
(a) Original sized 2 modes PSF example



(b) Cropped sized 2 modes PSF example



(c) Original sized predicted 2 modes PSF example



(d) cropped sized predicted 2 modes PSF example

Figure 67: 2 Zernike modes PSF datasets examples

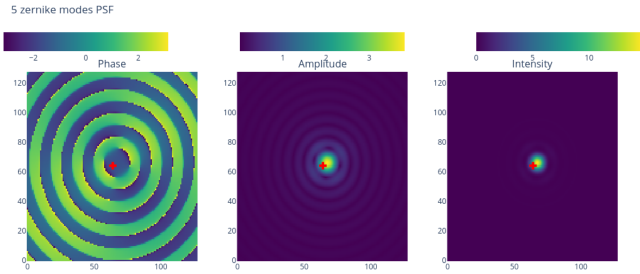
5 Zernike modes PSFs

- **Original sized 5 modes PSFs**: Two datasets of 70000 electric fields and corresponding intensities stored in 128x128x2 and 128x128 matrices respectively. The aberration by a 5 modes zernike basis.

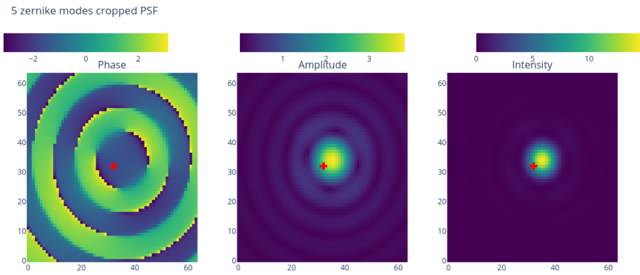
- **Cropped sized 5 modes PSFs**: Two datasets of 70000 electric fields and corresponding intensities stored in 64x64x2 and 64x64 matrices respectively. These cropped PSFs correspond to the central pixels from the Original sized 5 modes PSFs.

- **Original sized predicted 5 modes PSFs**: Two datasets of 70000 predicted electric fields and predicted intensities stored in 128x128x2 and 128x128 matrices respectively. These predicted PSFs are the outputs of a model trained with the Original sized 5 modes PSFs dataset and their corresponding PL intensities.

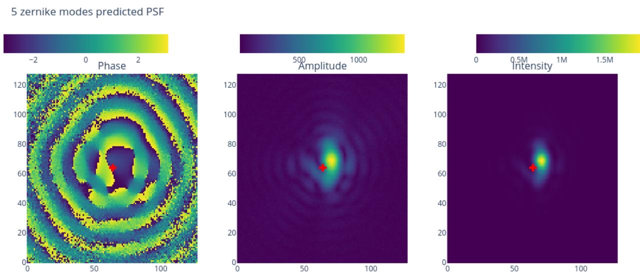
- **Cropped sized predicted 5 modes PSF**: Two datasets of 70000 predicted electric fields and predicted intensities stored in 64x64x2 and 64x64 matrices respectively. These cropped predicted PSFs are the outputs of a model trained with the Cropped sized 5 modes PSFs dataset and their corresponding PL intensities (which are the same ouput intensities from the Original sized 5 modes PSFs dataset).



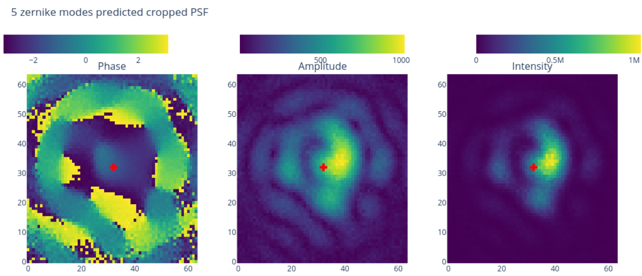
(a) Original sized 5 modes PSF example



(b) Cropped sized 5 modes PSF example



(c) Original sized predicted 5 modes PSF example



(d) cropped sized predicted 5 modes PSF example

Figure 68: 5 Zernike modes PSF datasets examples

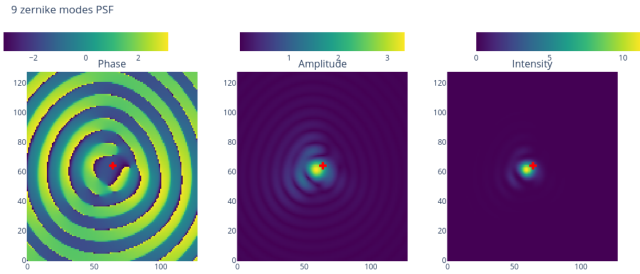
9 Zernike modes PSFs

- **Original sized 9 modes PSFs**: Two datasets of 70000 electric fields and corresponding intensities stored in 128x128x2 and 128x128 matrices respectively. The aberration by a 9 modes zernike basis.

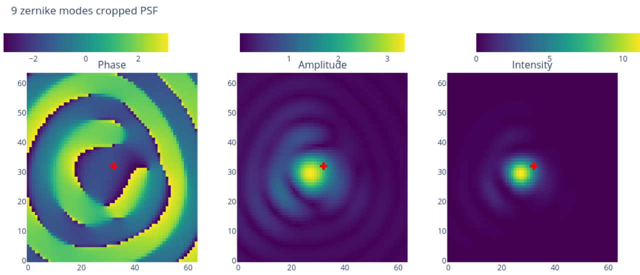
- **Cropped sized 9 modes PSFs**: Two datasets of 70000 electric fields and corresponding intensities stored in 64x64x2 and 64x64 matrices respectively. These cropped PSFs correspond to the central pixels from the Original sized 9 modes PSFs.

- **Original sized predicted 9 modes PSFs**: Two datasets of 70000 predicted electric fields and predicted intensities stored in 128x128x2 and 128x128 matrices respectively. These predicted PSFs are the outputs of a model trained with the Original sized 9 modes PSFs dataset and their corresponding PL intensities.

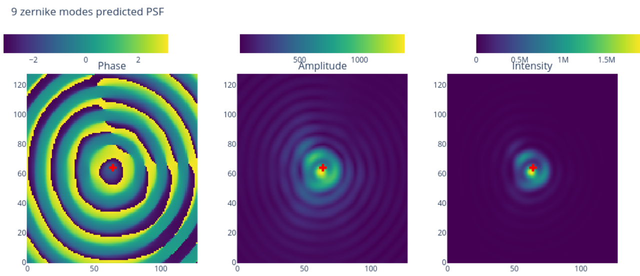
- **Cropped sized predicted 9 modes PSF**: Two datasets of 70000 predicted electric fields and predicted intensities stored in 64x64x2 and 64x64 matrices respectively. These cropped predicted PSFs are the outputs of a model trained with the Cropped sized 9 modes PSFs dataset and their corresponding PL intensities (which are the same ouput intensities from the Original sized 9 modes PSFs dataset).



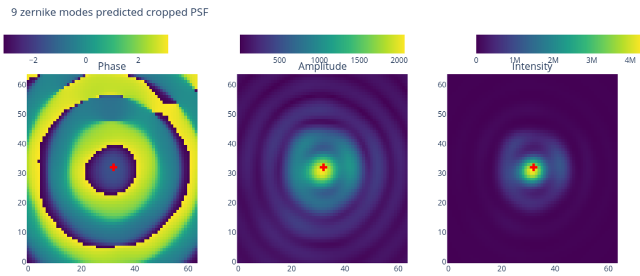
(a) Original sized 9 modes PSF example



(b) Cropped sized 9 modes PSF example



(c) Original sized predicted 9 modes PSF example



(d) cropped sized predicted 9 modes PSF example

Figure 69: 9 Zernike modes PSF datasets examples

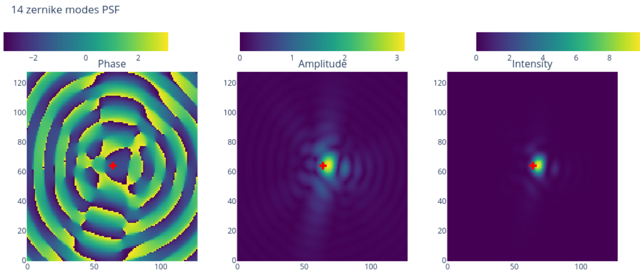
14 Zernike modes PSFs

- **Original sized 14 modes PSFs**: Two datasets of 70000 electric fields and corresponding intensities stored in 128x128x2 and 128x128 matrices respectively. The aberration by a 14 modes zernike basis.

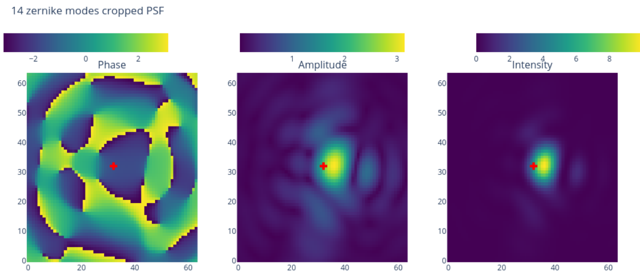
- **Cropped sized 14 modes PSFs**: Two datasets of 70000 electric fields and corresponding intensities stored in 64x64x2 and 64x64 matrices respectively. These cropped PSFs correspond to the central pixels from the Original sized 14 modes PSFs.

- **Original sized predicted 14 modes PSFs**: Two datasets of 70000 predicted electric fields and predicted intensities stored in 128x128x2 and 128x128 matrices respectively. These predicted PSFs are the outputs of a model trained with the Original sized 14 modes PSFs dataset and their corresponding PL intensities.

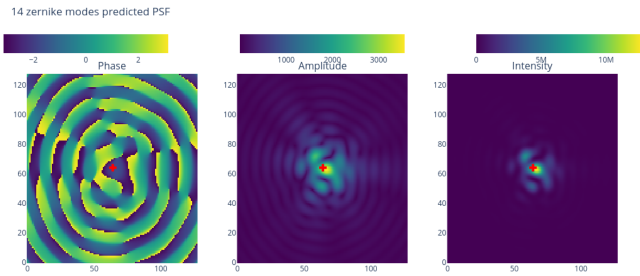
- **Cropped sized predicted 14 modes PSF**: Two datasets of 70000 predicted electric fields and predicted intensities stored in 64x64x2 and 64x64 matrices respectively. These cropped predicted PSFs are the outputs of a model trained with the Cropped sized 14 modes PSFs dataset and their corresponding PL intensities (which are the same ouput intensities from the Original sized 14 modes PSFs dataset).



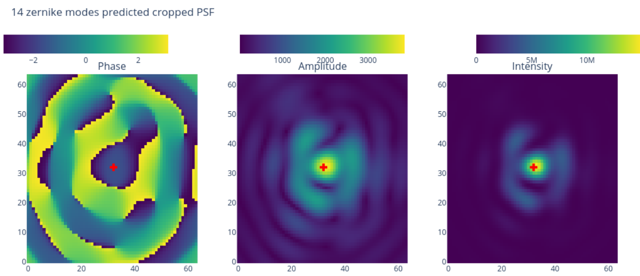
(a) Original sized 14 modes PSF example



(b) Cropped sized 14 modes PSF example



(c) Original sized predicted 14 modes PSF example



(d) cropped sized predicted 14 modes PSF example

Figure 70: 14 Zernike modes PSF datasets examples

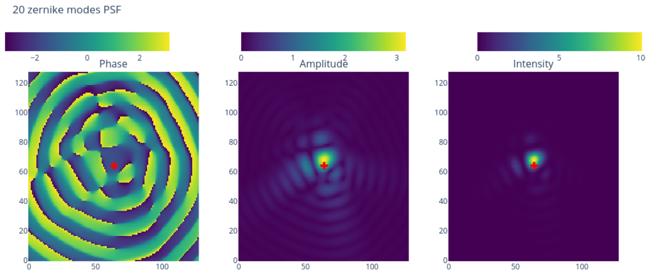
20 Zernike modes PSFs

- **Original sized 20 modes PSFs**: Two datasets of 70000 electric fields and corresponding intensities stored in 128x128x2 and 128x128 matrices respectively. The aberration by a 20 modes zernike basis.

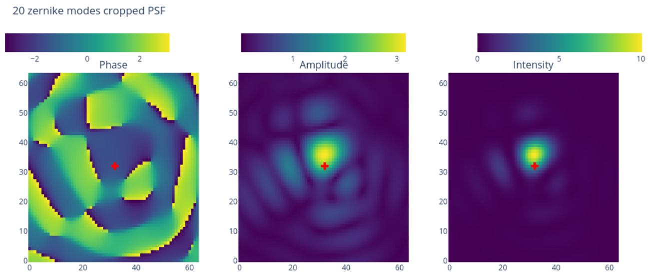
- **Cropped sized 20 modes PSFs**: Two datasets of 70000 electric fields and corresponding intensities stored in 64x64x2 and 64x64 matrices respectively. These cropped PSFs correspond to the central pixels from the Original sized 20 modes PSFs.

- **Original sized predicted 20 modes PSFs**: Two datasets of 70000 predicted electric fields and predicted intensities stored in 128x128x2 and 128x128 matrices respectively. These predicted PSFs are the outputs of a model trained with the Original sized 20 modes PSFs dataset and their corresponding PL intensities.

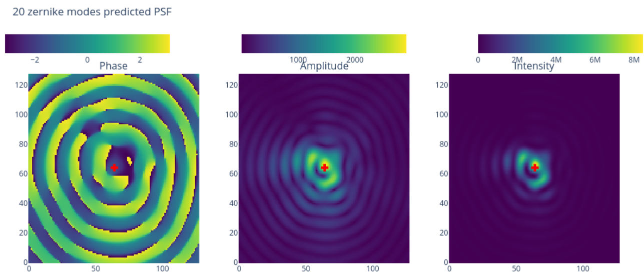
- **Cropped sized predicted 20 modes PSF**: Two datasets of 70000 predicted electric fields and predicted intensities stored in 64x64x2 and 64x64 matrices respectively. These cropped predicted PSFs are the outputs of a model trained with the Cropped sized 20 modes PSFs dataset and their corresponding PL intensities (which are the same ouput intensities from the Original sized 20 modes PSFs dataset).



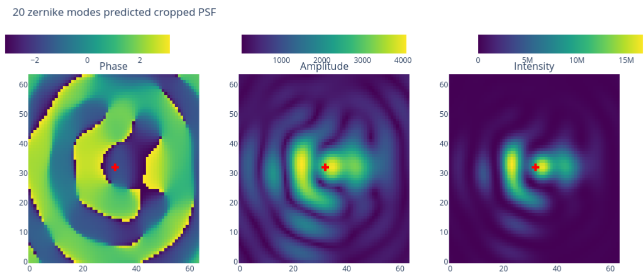
(a) Original sized 20 modes PSF example



(b) Cropped sized 20 modes PSF example



(c) Original sized predicted 20 modes PSF example



(d) cropped sized predicted 20 modes PSF example

Figure 71: 20 Zernike modes PSF datasets examples

LP mode coefficients There are two PL intensities dataset per Zernike aberration PSF subgroup: LP modes coefficients for 2, 5, 9, 14, 20 modes PSFs. Each of the dataset has 70000 datapoints each datapoint being the complex coefficients stored in a 19x2 matrix that separates the real and imaginary part of the coefficients.

The two datasets correspond to the LP coefficients that are computed in the multi-mode end of Photonic Lanterns. The PLs are:

- 19 mode supporting multimode end with 19 waveguides in the single mode end.
- 42 mode supporting multimode end with 42 waveguides in the single mode end.

PL intensities There is one PL intensities dataset per Zernike aberration PSF subgroup: PL intensities for 2, 5, 9, 14, 20 modes PSFs. Each of the dataset has 70000 datapoints each datapoint being the 19 intensities corresponding to the PSF

The two datasets correspond to the single mode end intensities of Photonic Lanterns. The PLs are:

- 19 mode supporting multimode end with 19 waveguides in the single mode end.
- 42 mode supporting multimode end with 42 waveguides in the single mode end.

0.2 The models

For all the datasets a model with the following configuration has been trained. The inputs of the model are the PL intensities and the outputs are the flattened matrices that represent the PSFs' complex fields.

HYPERPARAMETERS:

```
*ARCHITECTURE HYPERPARAMETERS:
  -Fully Connected
  -Input shape: 19
  -Output shape:
    - 32768 for original sized PSF electric field
    - 16384 for original sized PSF intensity
    - 8192 for cropped sized PSF electric field
    - 4096 for cropped sized PSF intensity
  -Hidden layers: [1024, 1024, 1024, 1024, 1024, 1024]
  -Regularizer: None
  -Hidden Layers Activation: relu
  -Output Layer Activation: linear
  -Batch Normalization: False
  -Dropout: False, 0.2

*COMPILATION HYPERPARAMETERS:
  -Optimizer: ADAM lr=0.001, beta_1=0.9, beta_2=0.999
  -Loss Function: MSE
  -Metric: MSE

*TRAINING HYPERPARAMETERS:
  -Epochs: 100
  -Batch size: 32
  -Callbacks:
```

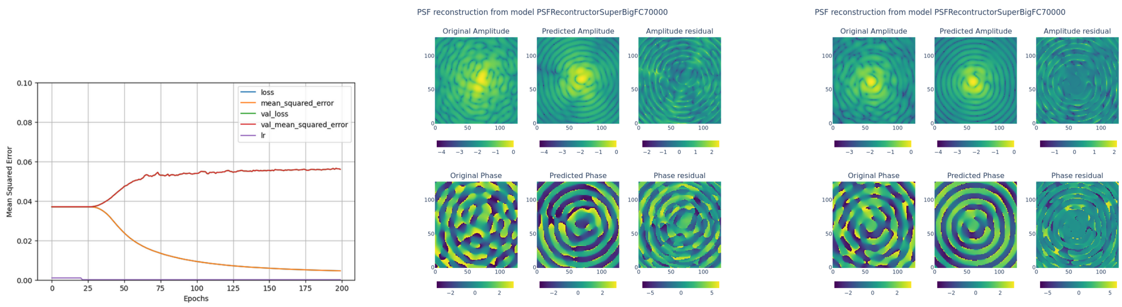
```
-ReduceLROnPlateau: MSE 20 x0.1
-Early Stop: MSE 50
```

The exception is the model trained for the Atmospheric Aberration Cropped PSF which has Batch Normalization activated.

0.2.1 Atmospheric aberration related models

Original sized PSF :

```
-Train MSE: 0.004607476759701967
-Validation MSE: 0.056021399796009064
```

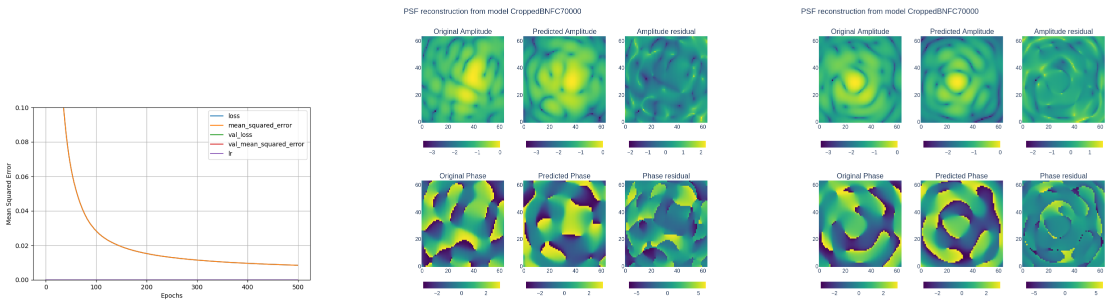


(a) Training Evolution (b) Validation example (c) Train example

Figure 72: Results of training the model PSFRecontructorSuperBigFC70000-1

Cropped sized PSF :

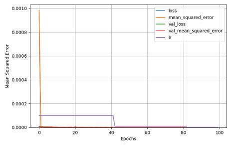
```
-Train MSE: 0.008466990664601326
-Validation MSE: 0.20970138907432556
```

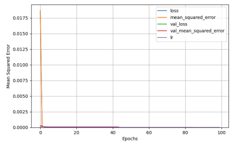


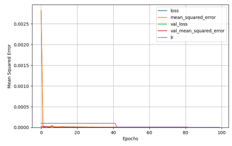
(a) Training Evolution (b) Validation example (c) Train example

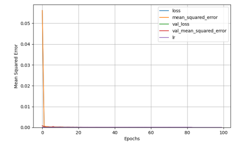
Figure 73: Results of training the model PSFRecontructorSuperBigFC70000-1

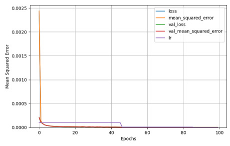
0.2.2 Zernike modes related models

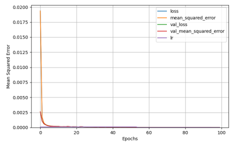

(a) 2 zernike mode PSF electric field model training

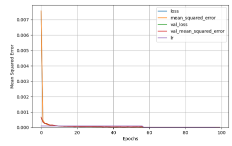

(b) 2 zernike mode PSF intensity model training

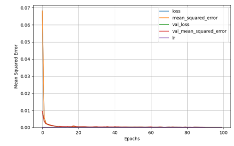

(c) 2 zernike mode cropped PSF electric field model training

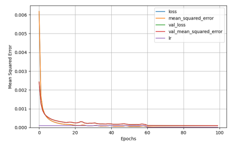

(d) 2 zernike mode cropped PSF intensity model training

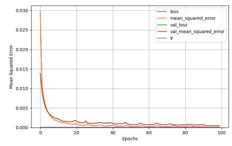

(e) 5 zernike mode PSF electric field model training

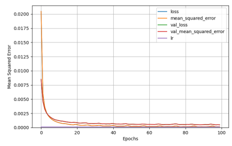

(f) 5 zernike mode PSF intensity model training

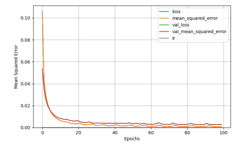

(g) 5 zernike mode cropped PSF electric field model training

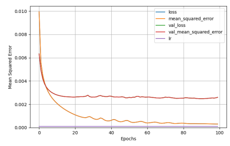

(h) 5 zernike mode cropped PSF intensity model training

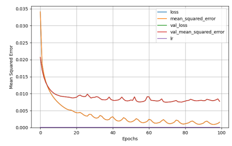

(i) 9 zernike mode PSF electric field model training

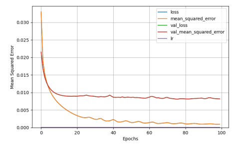

(j) 9 zernike mode PSF intensity model training

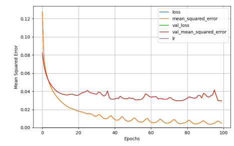

(k) 9 zernike mode cropped PSF electric field model training

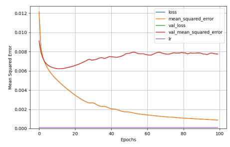

(l) 9 zernike mode cropped PSF intensity model training

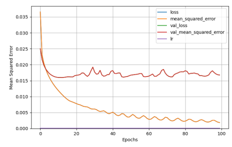

(m) 14 zernike mode PSF electric field model training

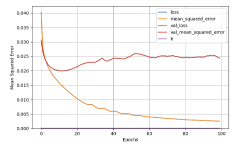

(n) 14 zernike mode PSF intensity model training


(o) 14 zernike mode cropped PSF electric field model training


(p) 14 zernike mode cropped PSF intensity model training


(q) 20 zernike mode PSF electric field model training


(r) 20 zernike mode PSF intensity model training


(s) 20 zernike mode cropped PSF electric field model training

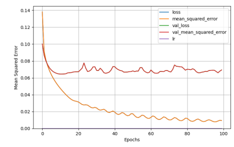

(t) 20 zernike mode cropped PSF intensity model training

Figure 74: Training evolution comparison for the different Zernike datasets

2 modes PSF models MSE :

	Electric field	Cropped electric field	Intensity	Cropped intensity
Train MSE	6.823e-8	1.770e-7	6.212e-7	2.646e-6
Val MSE	6.053e-8	1.450e-7	5.212e-7	1.880e-6

Table 1: 2 Zernike modes related models MSE

5 modes PSF models MSE :

	Electric field	Cropped electric field	Intensity	Cropped intensity
Train MSE	1.753e-6	4.443e-6	6.019e-6	3.044e-5
Val MSE	2.529e-6	7.328e-6	1.142e-6	4.700e-5

Table 2: 5 Zernike modes related models MSE

9 modes PSF models MSE :

	Electric field	Cropped electric field	Intensity	Cropped intensity
Train MSE	1.825e-5	1.599e-4	1.025e-4	8.590e-4
Val MSE	1.025e-4	4.883e-4	4.667e-4	2.770e-3

Table 3: 9 Zernike modes related models MSE

14 modes PSF models MSE :

	Electric field	Cropped electric field	Intensity	Cropped intensity
Train MSE	3.085e-4	9.827e-4	1.597e-3	4.715e-3
Val MSE	2.602e-3	8.197e-3	7.773e-3	0.0294

Table 4: 14 Zernike modes related models MSE

20 modes PSF models MSE :

	Electric field	Cropped electric field	Intensity	Cropped intensity
Train MSE	8.804e-4	2.546e-3	1.872e-3	9.445e-3
Val MSE	7.74e-3	0.024	0.0167	0.069

Table 5: 20 Zernike modes related models MSE

A summary of the MSE evolution over the Zernike PSFs datasets is shown below. The fact that the validation MSE for 2 modes is the worse may be because the neural network is not able to understand traslations.

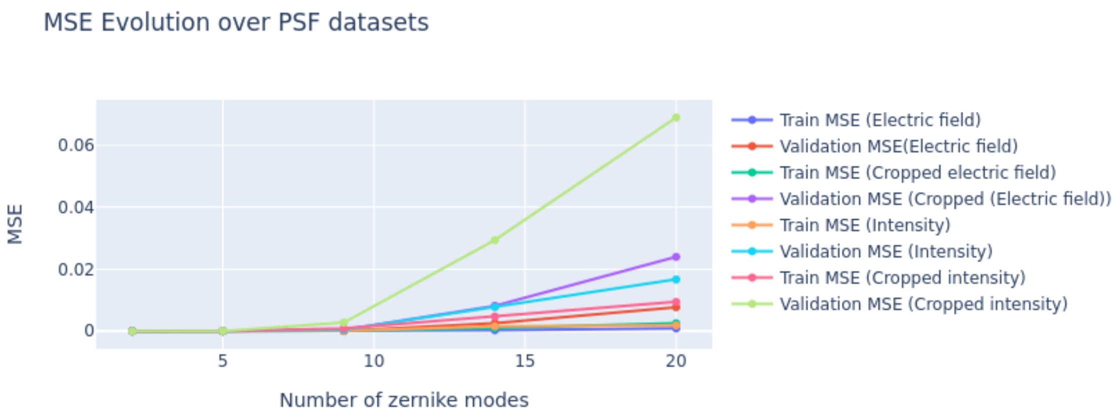


Figure 75: MSE evolution over the Zernike PSFs datasets

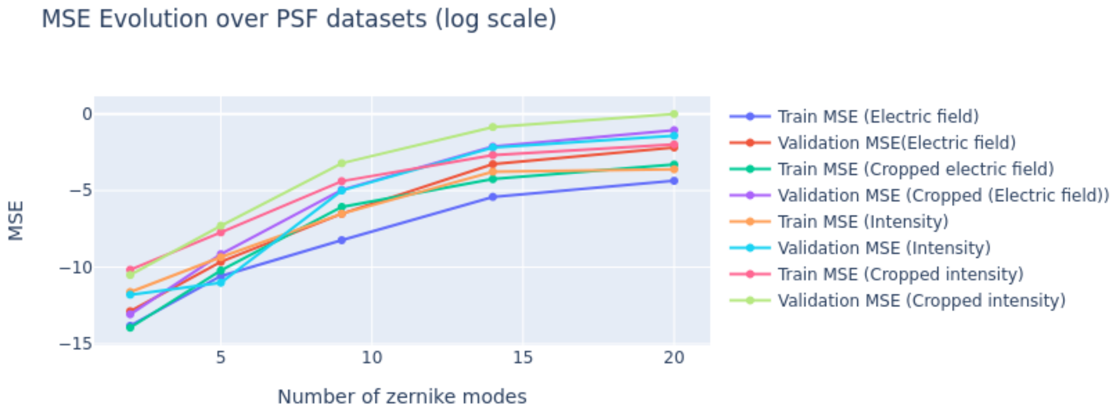
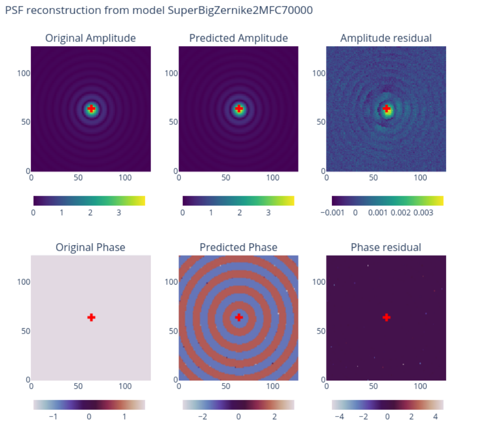
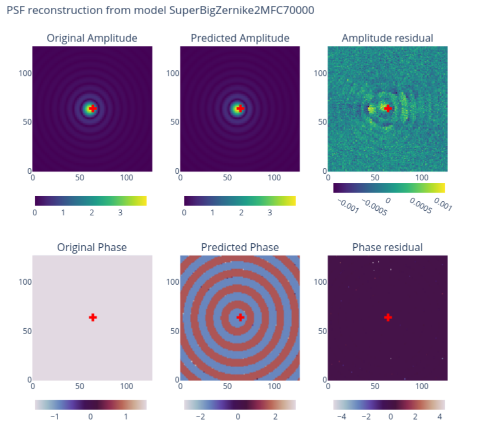


Figure 76: MSE evolution over the Zernike PSFs datasets in logarithmic scale

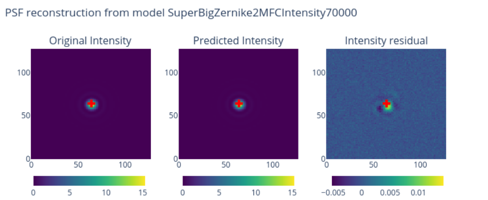
Model output examples :



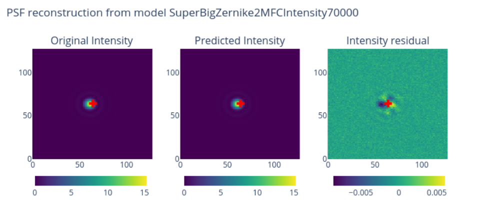
(a) Train example from electric field model



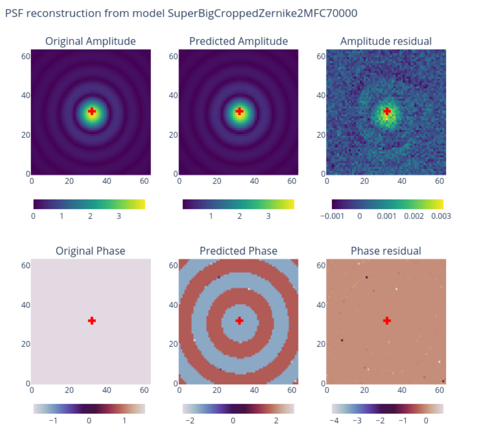
(b) Validation example from electric field model



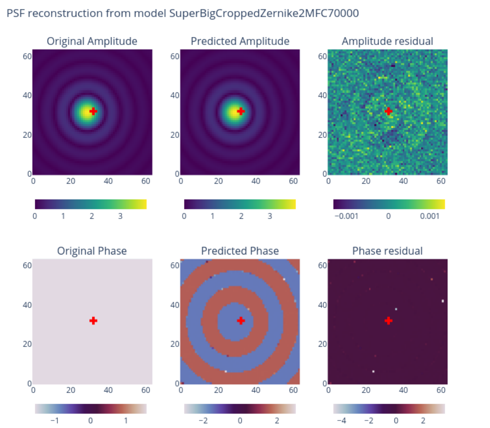
(c) Train example from intensity model



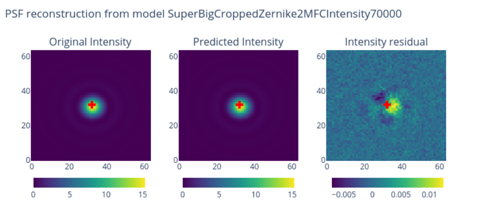
(d) Validation example from intensity model



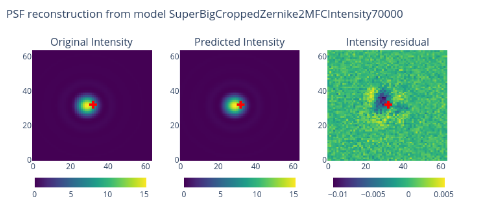
(e) Train example from cropped electric field model



(f) Validation example cropped from electric field model

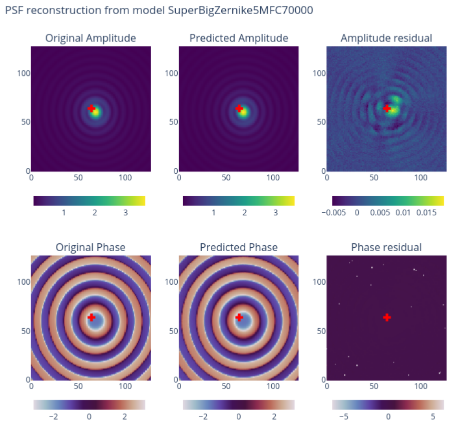


(g) Train example from cropped intensity model

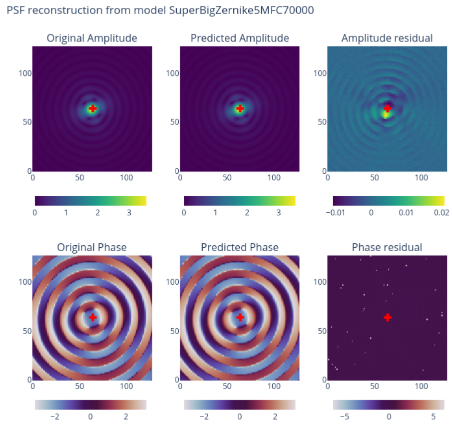


(h) Validation example from cropped model

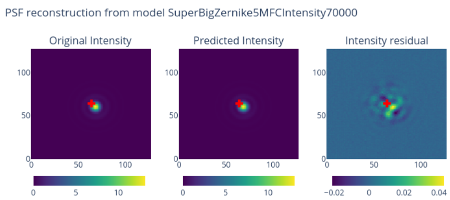
Figure 77: Model outputs for 2 mode PSF datasets



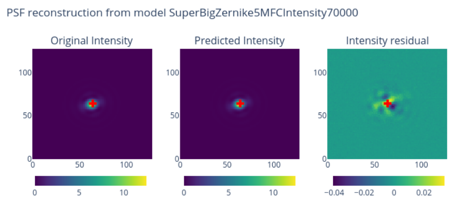
(a) Train example from electric field model



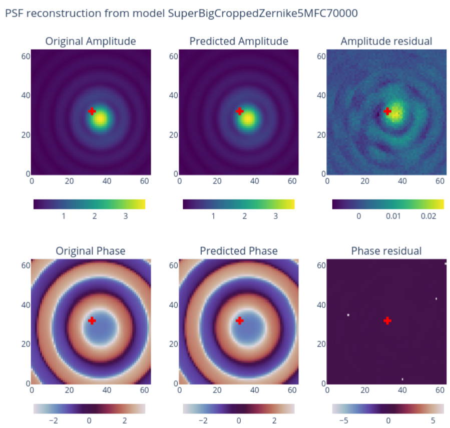
(b) Validation example from electric field model



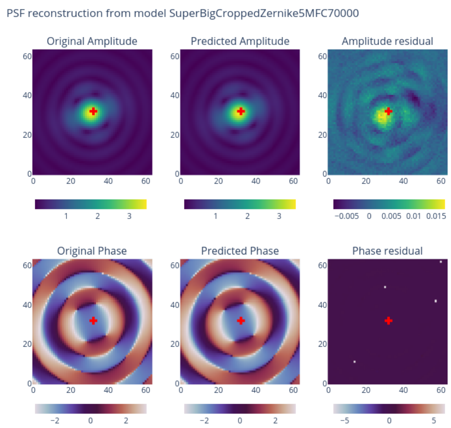
(c) Train example from intensity model



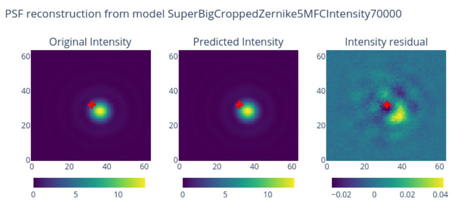
(d) Validation example from intensity model



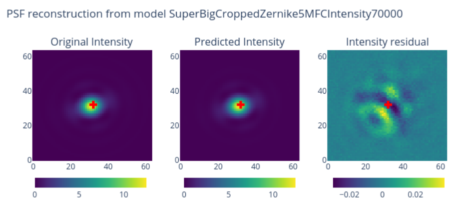
(e) Train example from cropped electric field model



(f) Validation example cropped from electric field model

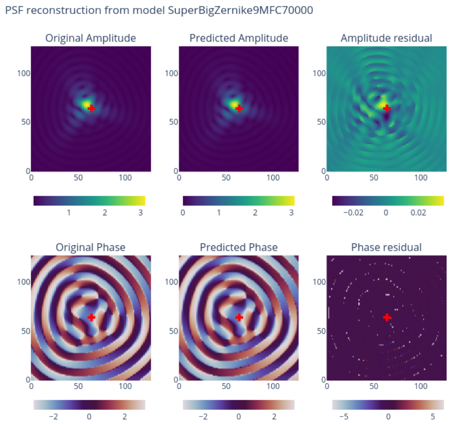


(g) Train example from cropped intensity model

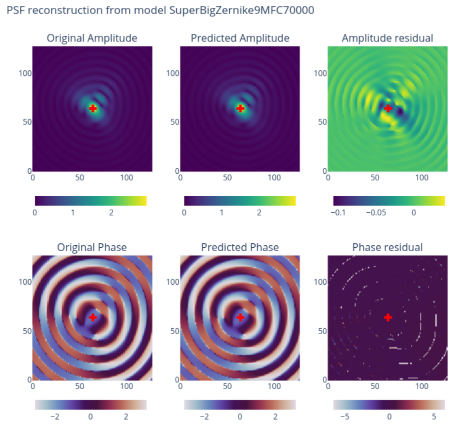


(h) Validation example from cropped model

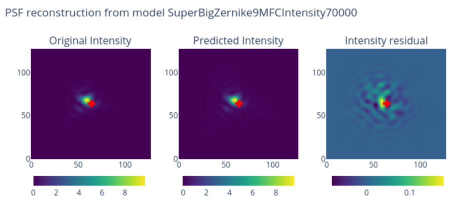
Figure 78: Model outputs for 5 mode PSF datasets



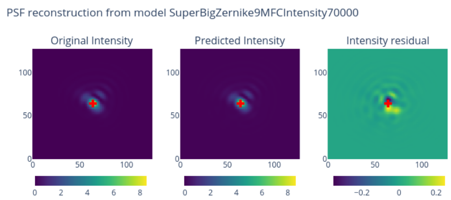
(a) Train example from electric field model



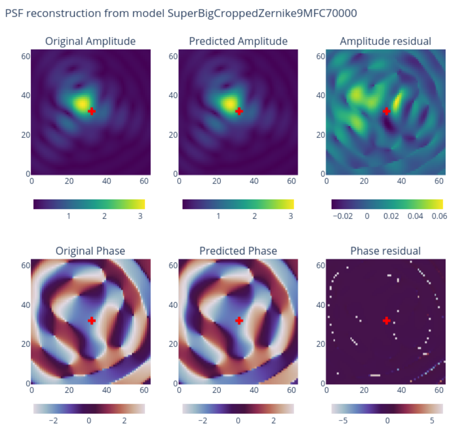
(b) Validation example from electric field model



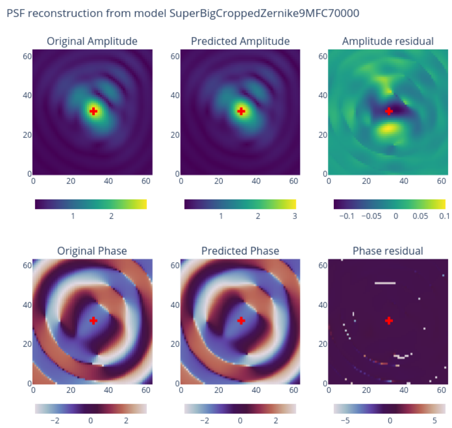
(c) Train example from intensity model



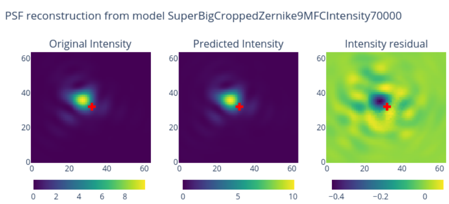
(d) Validation example from intensity model



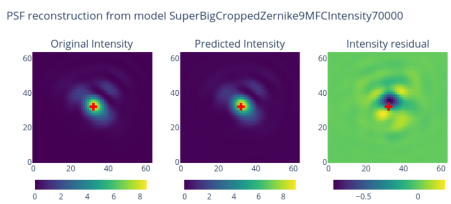
(e) Train example from cropped electric field model



(f) Validation example cropped from electric field model

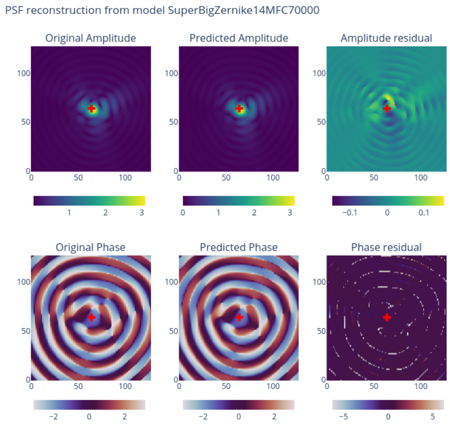


(g) Train example from cropped intensity model

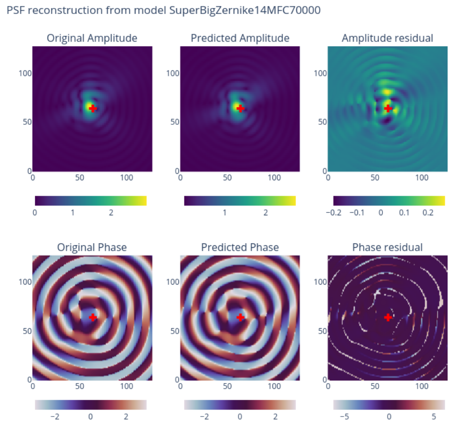


(h) Validation example from cropped model

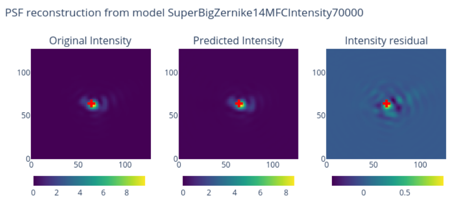
Figure 79: Model outputs for 9 mode PSF datasets



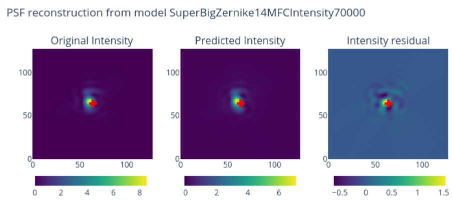
(a) Train example from electric field model



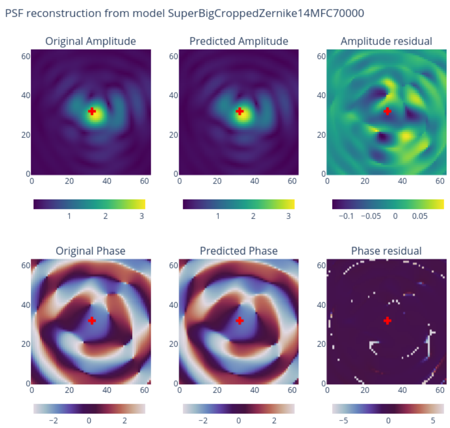
(b) Validation example from electric field model



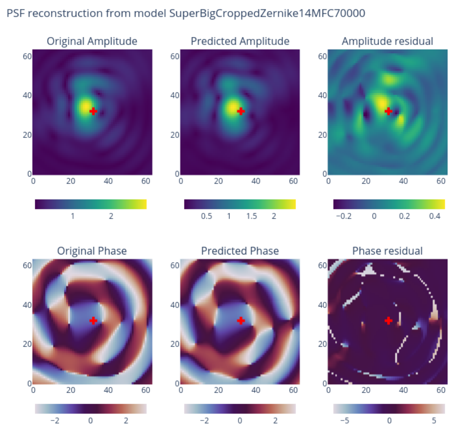
(c) Train example from intensity model



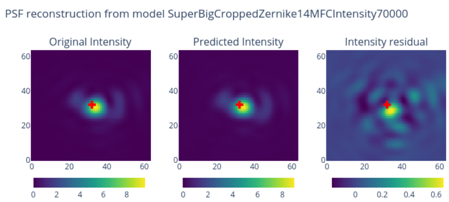
(d) Validation example from intensity model



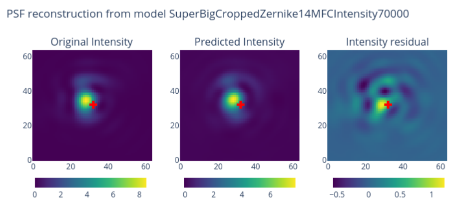
(e) Train example from cropped electric field model



(f) Validation example cropped from electric field model

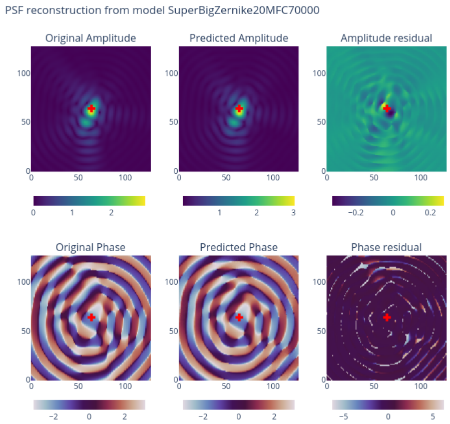


(g) Train example from cropped intensity model

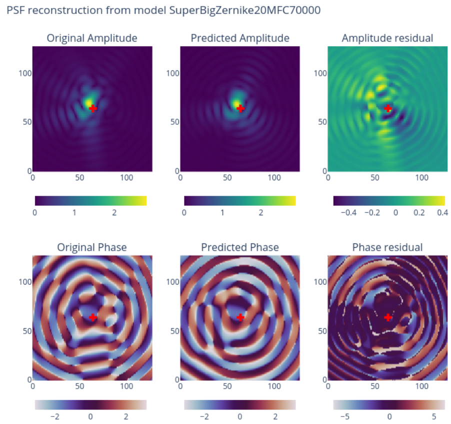


(h) Validation example from cropped model

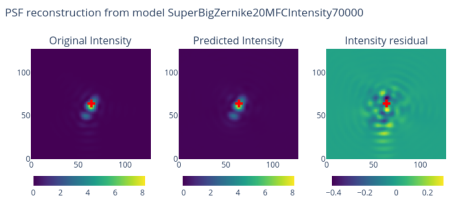
Figure 80: Model outputs for 14 mode PSF datasets



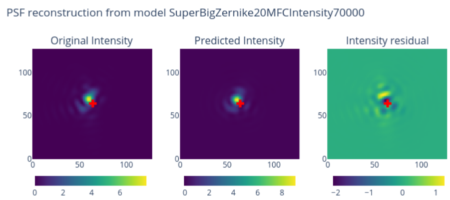
(a) Train example from electric field model



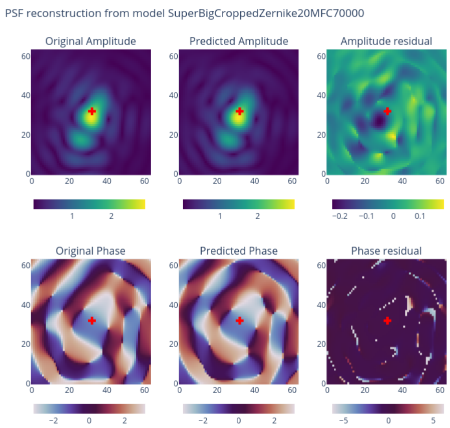
(b) Validation example from electric field model



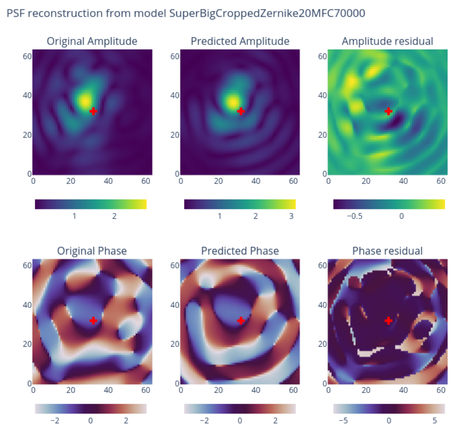
(c) Train example from intensity model



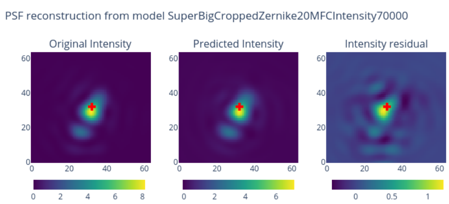
(d) Validation example from intensity model



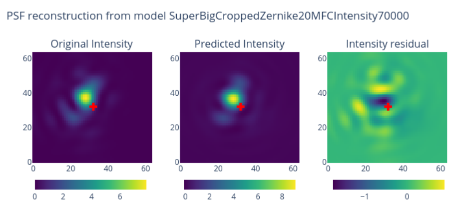
(e) Train example from cropped electric field model



(f) Validation example cropped from electric field model



(g) Train example from cropped intensity model



(h) Validation example from cropped model

Figure 81: Model outputs for 20 mode PSF datasets

0.3 Euclidean distances analysis for atmospheric aberration PSFs

0.3.1 Preprocessing

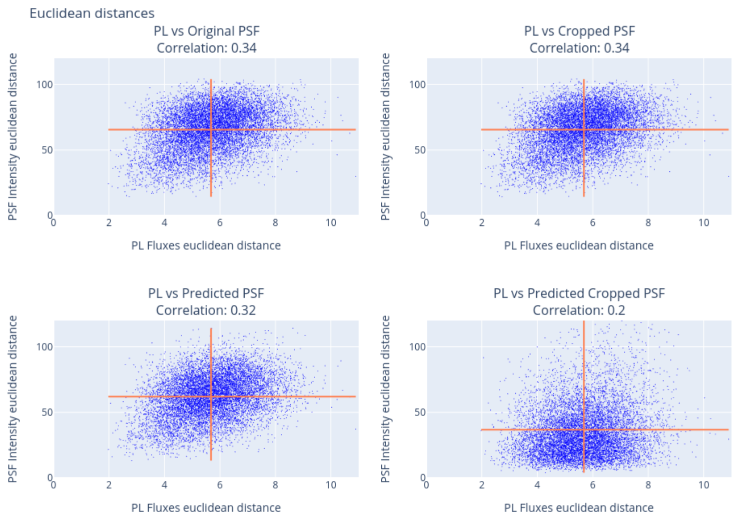
- The PSF electric fields are converted to a matrix of intensities of 128x128 size and the flattened to calculate the euclidean distances between them.
- 70000 datapoint pairs are defined for which the euclidean distances will be calculated.

0.3.2 Results

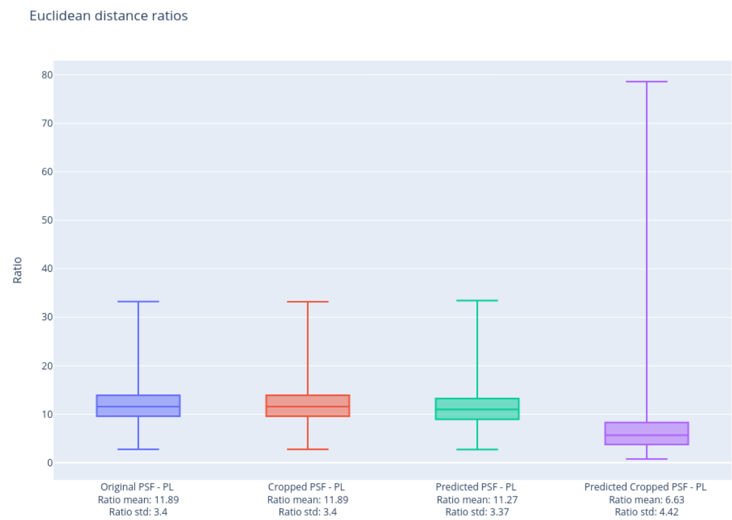
After performing an ANOVA test on the euclidean distances from the selected pairs of the 4 datasets obtaining a p-value of 0 and F-statistic of 4789.1531.

0.3.3 Analysis

The correlation is 0.3 which indicates a slightly positive linear relationship between the PL flux and PSF in all cases except for the cropped predictions which has a 0.2 correlation rate. This makes sense as the model that predicted those PSFs is more overfitted than the model that predicts the original sized PSFs.The clouds are dispersed almost equally from the center of mass which may indicate that a 19 mode PL may not be enough to encode all PSF information.



(a) Euclidean distance cloud



(b) Euclidean distance ratios

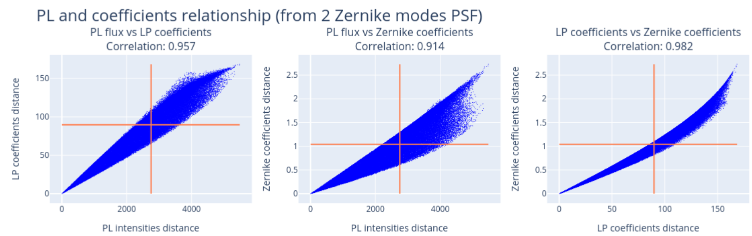
Figure 82: Euclidean distances ratios between PL and PSF pairs

0.4 Euclidean distances analysis for Zernike modes PSFs

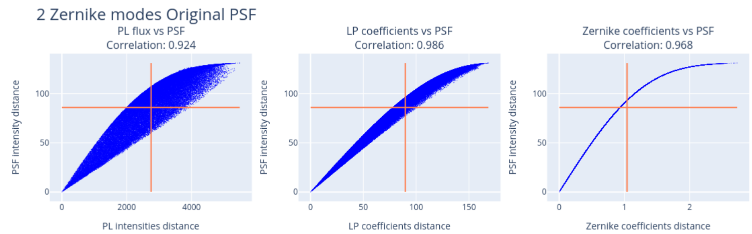
0.4.1 Preprocessing

- The PSF electric fields matrices are flattened to compute the euclidean distances between 1d vectors.
- 70000 datapoint pairs for each zernike datasets are randomly defined. The euclidean distances will be calculated for these selected pairs.
- In this case, LP coefficients are also analysed.

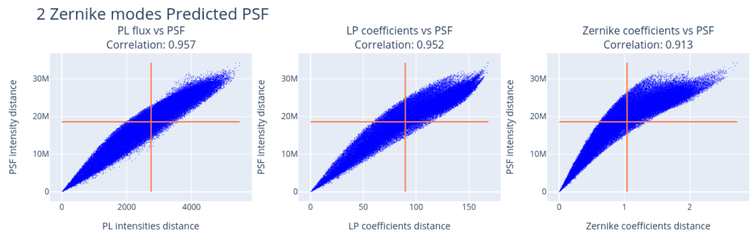
0.4.2 Euclidean distances comparison per number of zernike modes



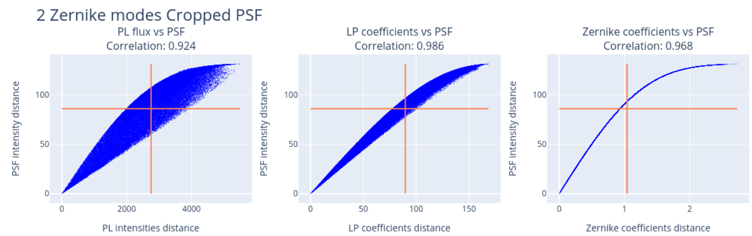
(a) Euclidean distance comparison between coefficients and PL flux



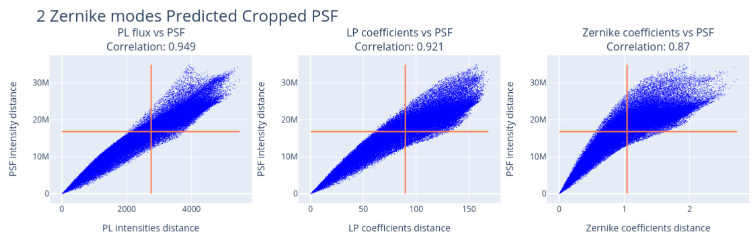
(b) Euclidean distances for PSF intensity



(c) Euclidean distances for predicted PSF intensity

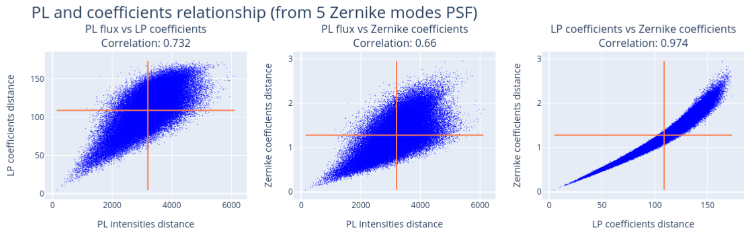


(d) Euclidean distances for cropped PSF intensity

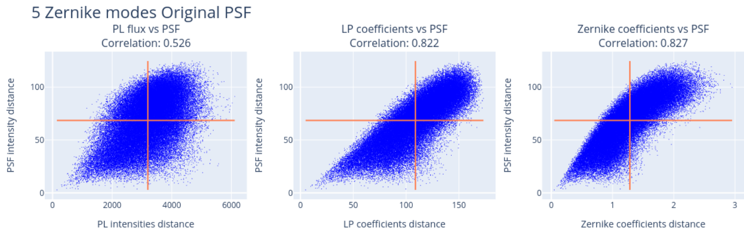


(e) Euclidean distances for predicted PSF intensity

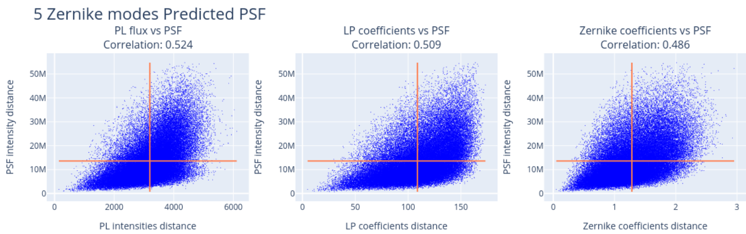
Figure 83: Euclidean distances comparison for 2 zernike modes related datasets



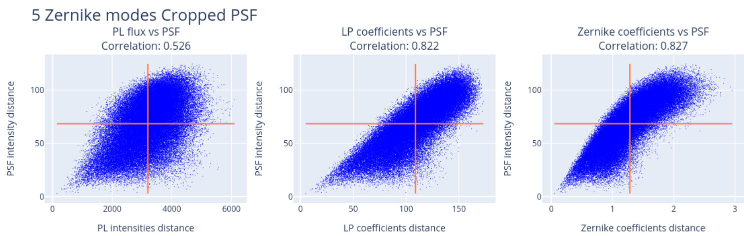
(a) Euclidean distance comparison between coefficients and PL flux



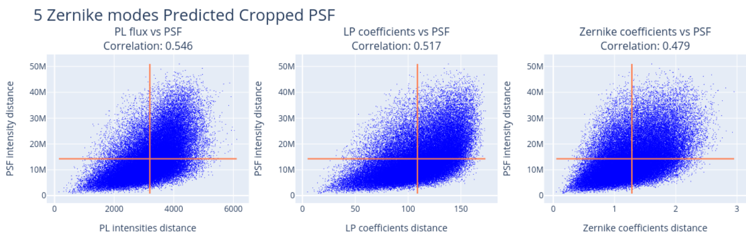
(b) Euclidean distances for PSF intensity



(c) Euclidean distances for predicted PSF intensity

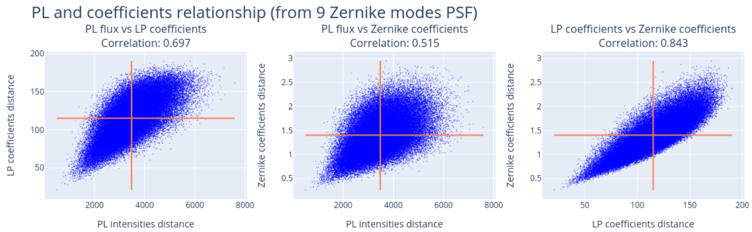


(d) Euclidean distances for cropped PSF intensity

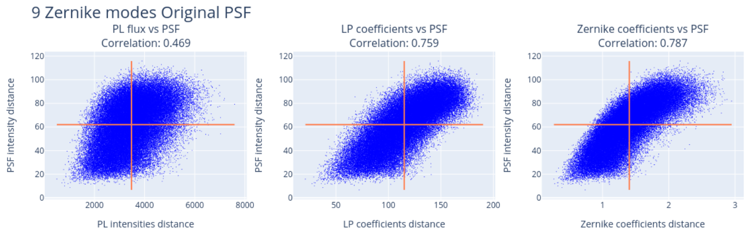


(e) Euclidean distances for predicted PSF intensity

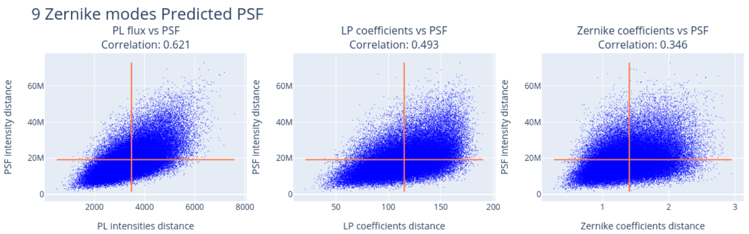
Figure 84: Euclidean distances comparison for 5 zernike modes related datasets



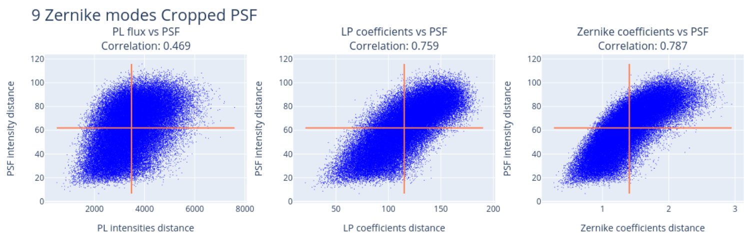
(a) Euclidean distance comparison between coefficients and PL flux



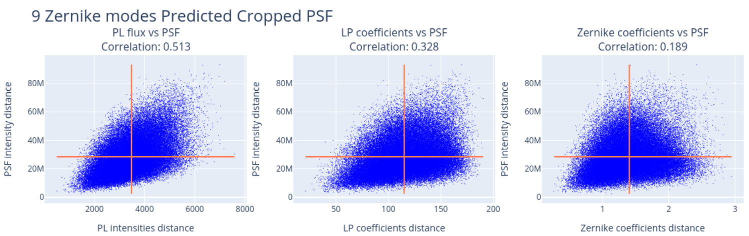
(b) Euclidean distances for PSF intensity



(c) Euclidean distances for predicted PSF intensity

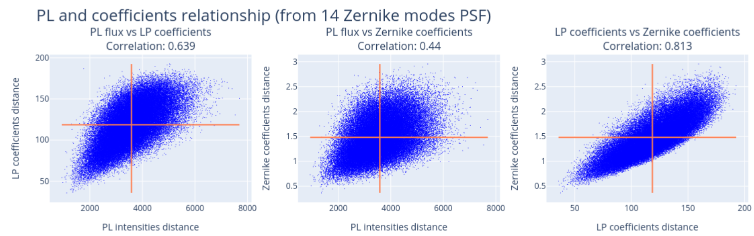


(d) Euclidean distances for cropped PSF intensity

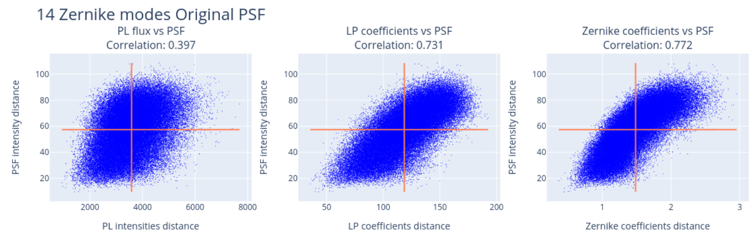


(e) Euclidean distances for predicted PSF intensity

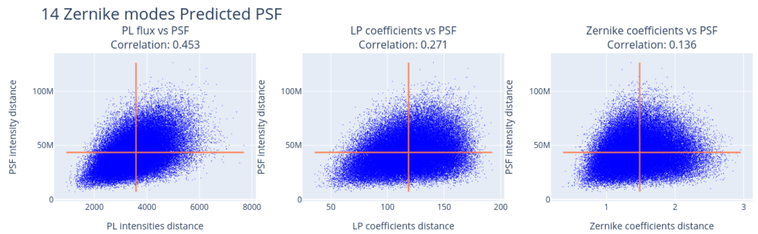
Figure 85: Euclidean distances comparison for 9 zernike modes related datasets



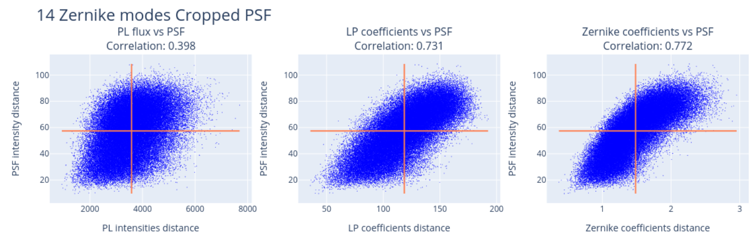
(a) Euclidean distance comparison between coefficients and PL flux



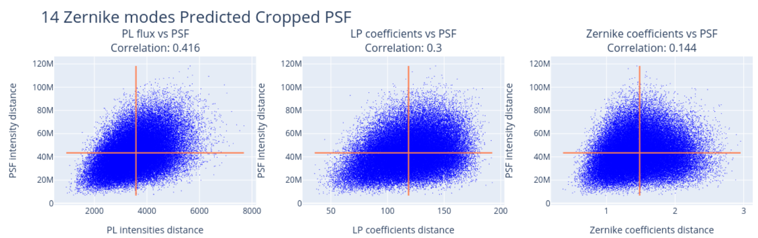
(b) Euclidean distances for PSF intensity



(c) Euclidean distances for predicted PSF intensity

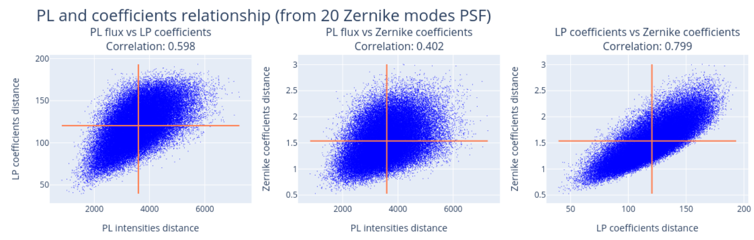


(d) Euclidean distances for cropped PSF intensity

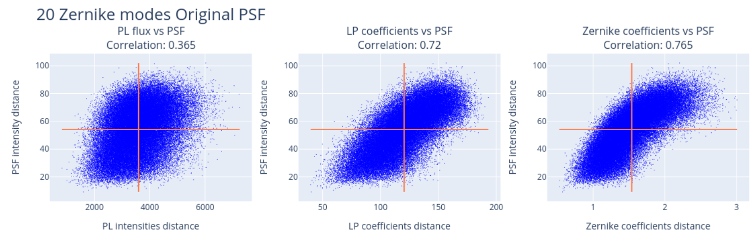


(e) Euclidean distances for predicted PSF intensity

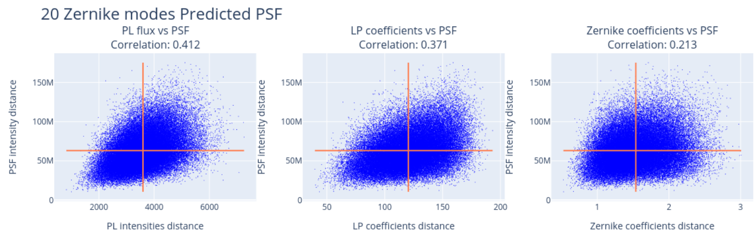
Figure 86: Euclidean distances comparison for 14 zernike modes related datasets



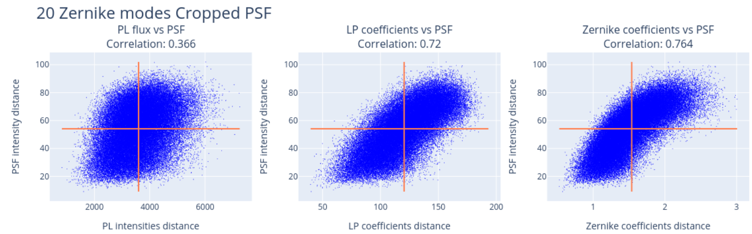
(a) Euclidean distance comparison between coefficients and PL flux



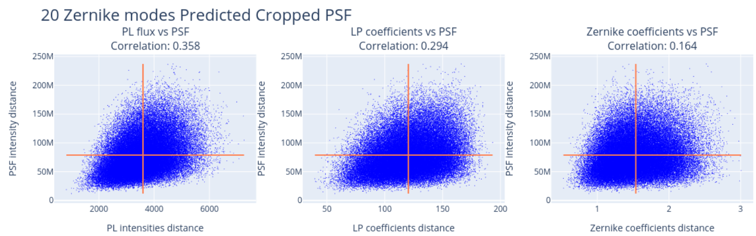
(b) Euclidean distances for PSF intensity



(c) Euclidean distances for predicted PSF intensity



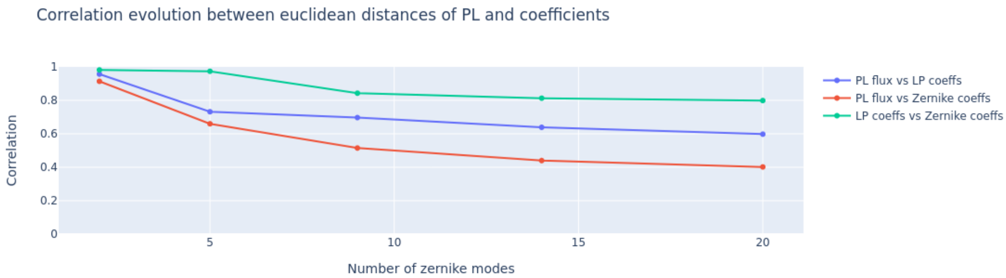
(d) Euclidean distances for cropped PSF intensity



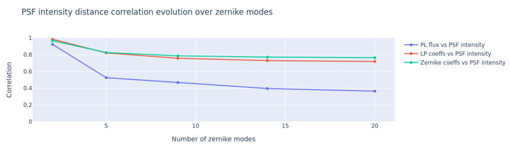
(e) Euclidean distances for predicted PSF intensity

Figure 87: Euclidean distances comparison for 20 zernike modes related datasets

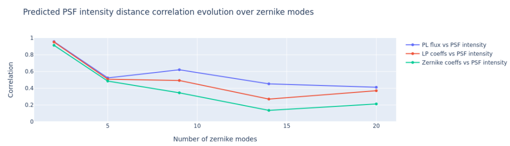
0.4.3 Euclidean distances comparison evolution over number of zernike modes



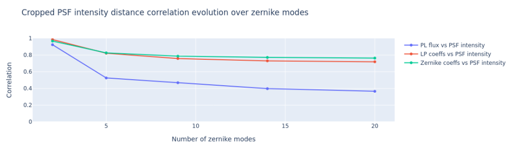
(a) Coefficients distances correlation evolution



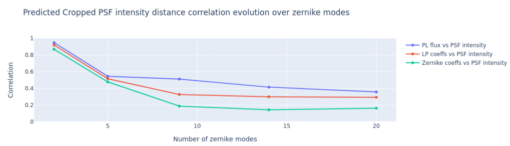
(b) PSF intensity distance correlation evolution



(c) Predicted PSF intensity distance correlation evolution

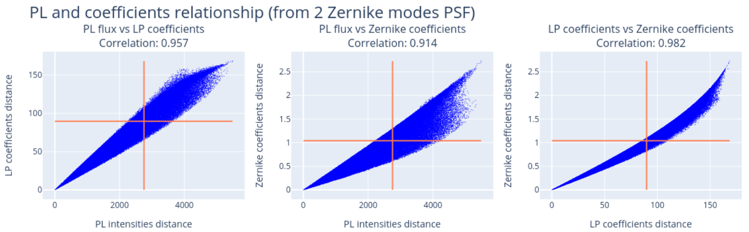


(d) Coefficients vs PL flux for 9 zernike modes

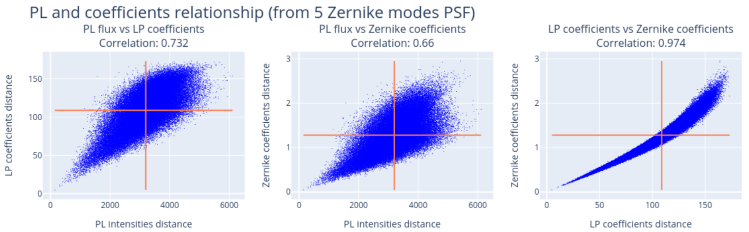


(e) Coefficients vs PL flux for 2 zernike modes

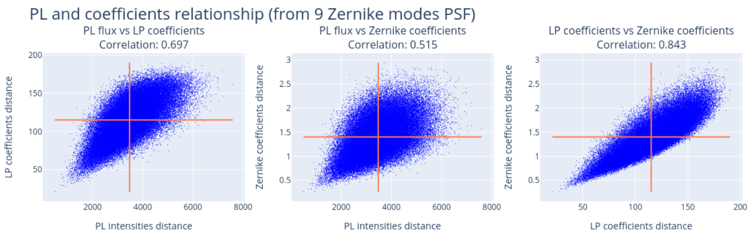
Figure 88: Euclidean distances correlation evolution over number of Zernike modes



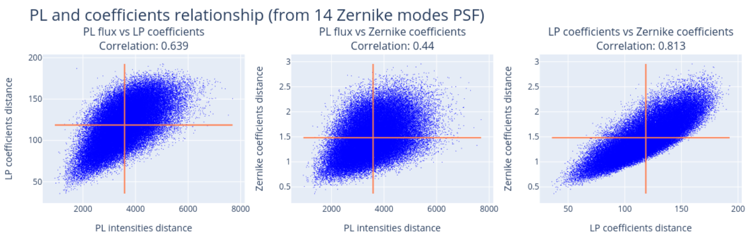
(a) Coefficients vs PL flux for 2 zernike modes



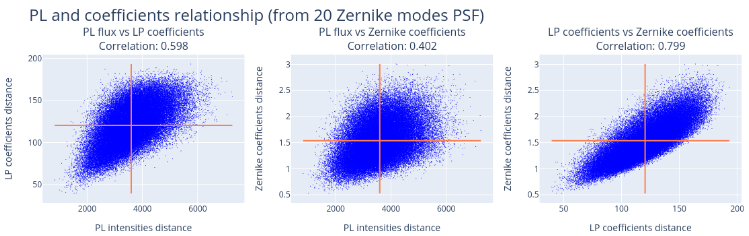
(b) Coefficients vs PL flux for 5 zernike modes



(c) Coefficients vs PL flux for 9 zernike modes

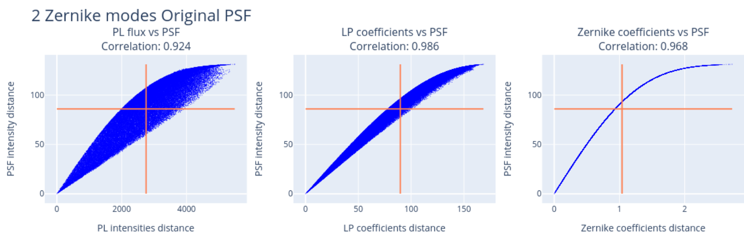


(d) Coefficients vs PL flux for 14 zernike modes

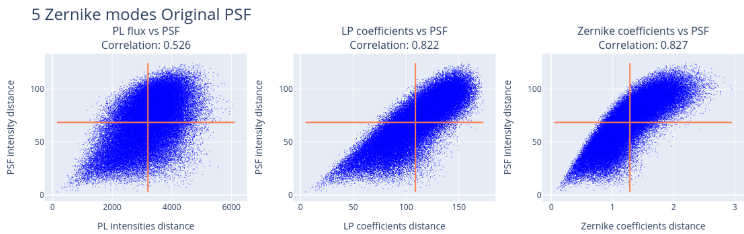


(e) Coefficients vs PL flux for 20 zernike modes

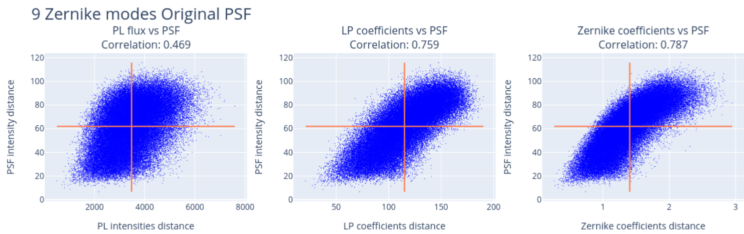
Figure 89: Euclidean distance comparison between coefficients and PL flux for different number of modes



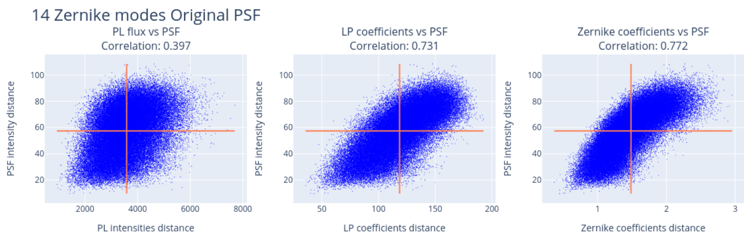
(a) Euclidean distances for 2 zernike modes PSF intensity



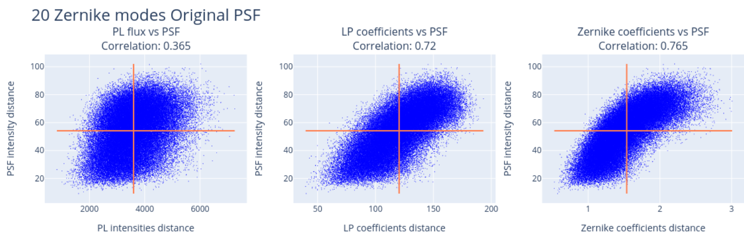
(b) Euclidean distances for 5 zernike modes PSF intensity



(c) Euclidean distances for 9 zernike modes PSF intensity

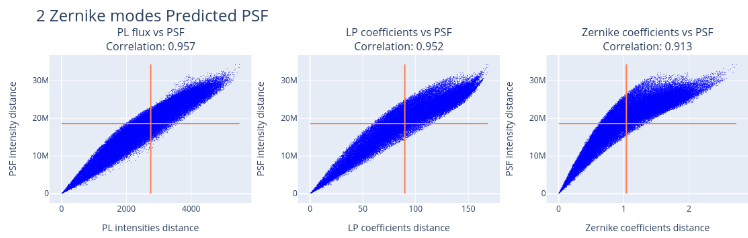


(d) Euclidean distances for 14 zernike modes PSF intensity

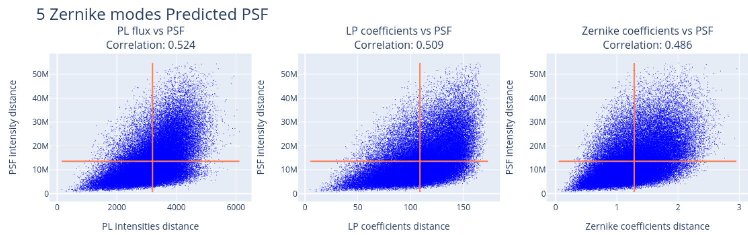


(e) Euclidean distances for 20 zernike modes PSF intensity

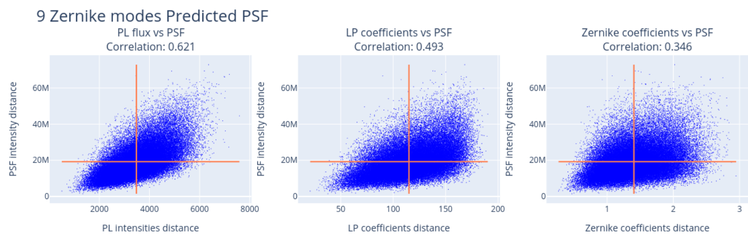
Figure 90: Euclidean distance comparison for PSF intensity for different number of modes



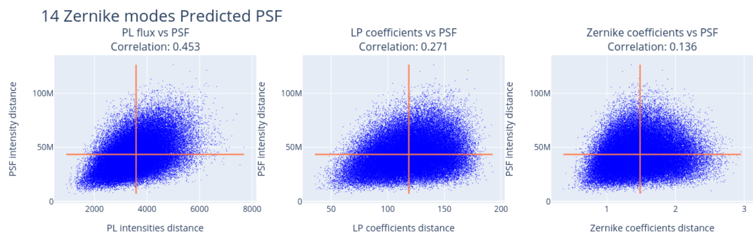
(a) Euclidean distances for 2 zernike modes predicted PSF intensity



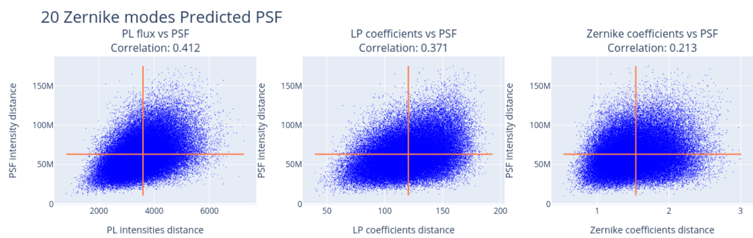
(b) Euclidean distances for 5 zernike modes predicted PSF intensity



(c) Euclidean distances for 9 zernike modes predicted PSF intensity

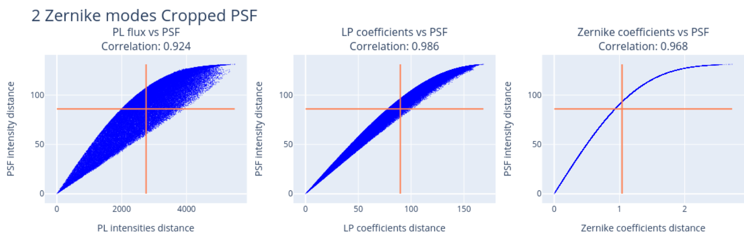


(d) Euclidean distances for 14 zernike modes predicted PSF intensity

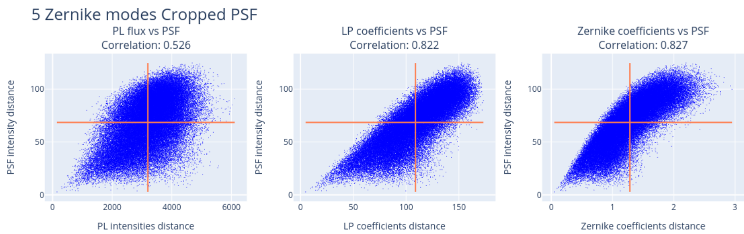


(e) Euclidean distances for 20 zernike modes predicted PSF intensity

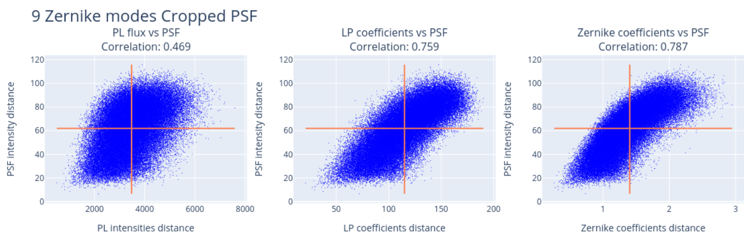
Figure 91: Euclidean distance comparison for predicted PSF intensity for different number of modes



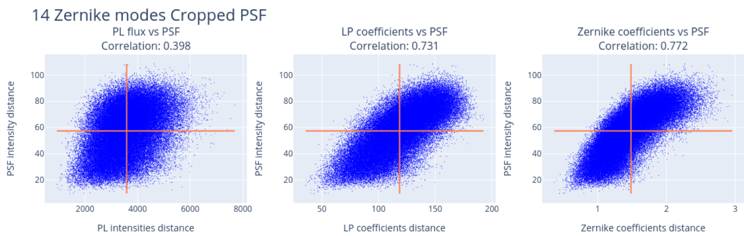
(a) Euclidean distances for 2 zernike modes cropped PSF intensity



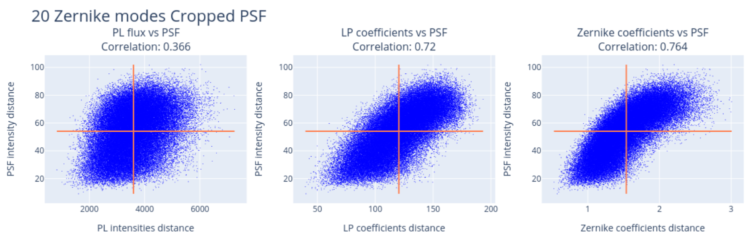
(b) Euclidean distances for 5 zernike modes cropped PSF intensity



(c) Euclidean distances for 9 zernike modes cropped PSF intensity

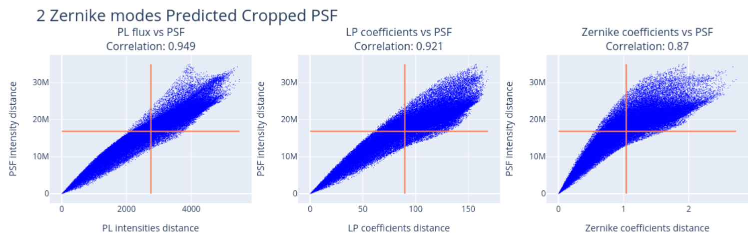


(d) Euclidean distances for 14 zernike modes cropped PSF intensity

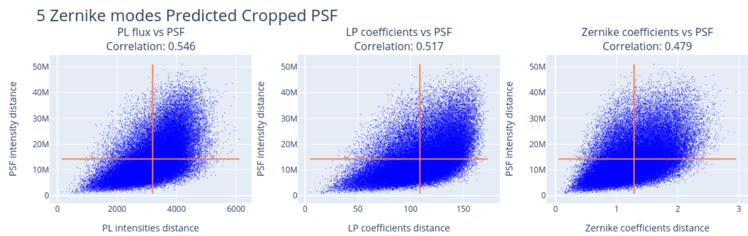


(e) Euclidean distances for 20 zernike modes cropped PSF intensity

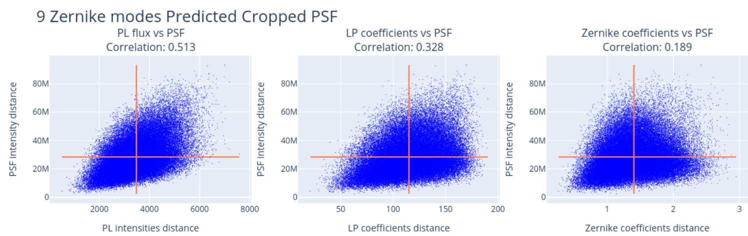
Figure 92: Euclidean distance comparison for cropped PSF intensity for different number of modes



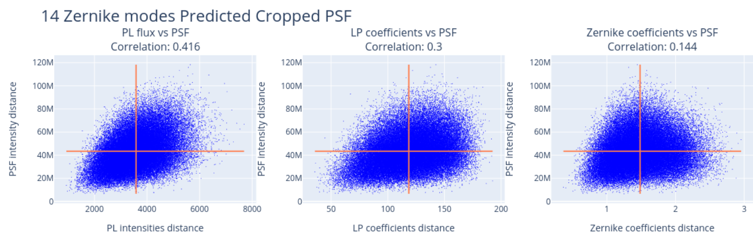
(a) Euclidean distances for 2 zernike modes predicted cropped PSF intensity



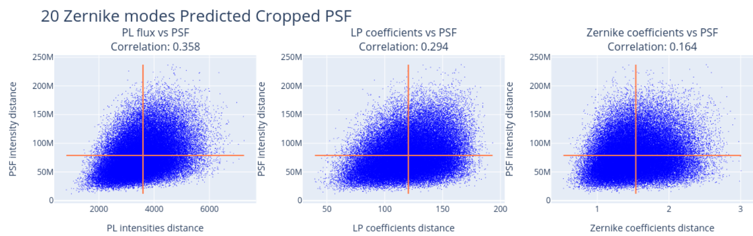
(b) Euclidean distances for 5 zernike modes predicted cropped PSF intensity



(c) Euclidean distances for 9 zernike modes predicted cropped PSF intensity



(d) Euclidean distances for 14 zernike modes predicted cropped PSF intensity



(e) Euclidean distances for 20 zernike modes predicted cropped PSF intensity

Figure 93: Euclidean distance comparison for predicted cropped PSF intensity for different number of modes

0.4.4 Analysis

0.5 Zernike modes PSFs Clustering

0.5.1 UMAPS

Before clustering, UMAPS for flattened PSF matrices, flattend LP coefficients matrices and PL intensities are processed. The same configuration is used for the different number of modes.

Dataset type	**Number of neighbors**	**Min distance**	**Number of components**
PSF electric field	500	0.5	1000
PSF intensity	500	0.5	1000
Pred PSF electric field	500	0.5	1000
Pred PSF intensity	500	0.5	1000

Table 6: UMAP parameter configurations for PSFs

0.5.2 Clustering

Using DBSCAN I create clusters for the UMAP representations. In the tables CDM and CDV are Cluster Density Mean and Cluster Density Variance respectively.

2 Zernike modes :

	ϵ	**neighbors**	**Clusters**	**CDM**	**CDV**	**Non noise points**
Zernike coeffs	0.01	6	1535	41.12	92.65	63130
LP coeffs	0.99	5	1470	44.00	631.91	64686
PL fluxes	34.7	5	1431	45.46	739.78	65065
PSF electric field	0.154	5	1563	40.16	110.41	62774
PSF intensity	0.155	5	1517	41.61	124.37	63125
Pred PSF ef	0.154	5	1339	41.01	159.22	63202
Pred PSF intensity	0.154	4	1497	42.40	206.78	63480

Table 7: DBSCAN clustering for 2 Zernike modes datasets

5 Zernike modes :

	ϵ	neighbors	Clusters	CDM	CDV	Non noise points
Zernike coeffs	0.128	4	1448	38.10	1204.20	55183
LP coeffs	12.8	4	1551	34.98	1109.08	54265
PL fluxes	410	4	1500	35.54	1131.51	53324
PSF electric field	0.31	4	1587	29.02	862.14	46065
PSF intensity	0.25	4	1533	34.41	1033.01	52753
Pred PSF ef	0.15	4	1364	48.16	210.10	65691
Pred PSF intensity	0.27	3	1645	24.95	1182.07	57491

Table 8: DBSCAN Clustering for 5 Zernike modes datasets

9 Zernike modes :

	ϵ	neighbors	Clusters	CDM	CDV	Non noise points
Zernike coeffs	0.29	4	1428	30.30	953.86	43275
LP coeffs	26	4	1485	28.01	889.60	43397
PL fluxes	770	3	1616	31.85	1099.66	51474
PSF electric field	0.26	4	1459	30.45	894.56	44429
PSF intensity	0.22	4	1402	39.29	1205.03	55096
Pred PSF ef	0.235	4	1460	22.28	512.15	32530
Pred PSF intensity	0.235	4	1385	30.49	851.07	42240

Table 9: DBSCAN Clustering for 9 Zernike modes datasets

14 Zernike modes :

	ϵ	**neighbors**	**Clusters**	**CDM**	**CDV**	**Non noise points**
Zernike coeffs	0.424	3	1639	25.89	880.85	42439
LP coeffs	35.5	3	1445	32.26	1064.67	46626
PL fluxes	950	3	1669	24.31	814.44	40577
PSF electric field	0.25	4	1607	27.05	831.31	44202
PSF intensity	0.21	4	1436	36.52	1129.68	42830
Pred PSF ef	0.35	3	1486	36.43	1214.19	54135
Pred PSF intensity	0.32	3	1567	36.16	1221.05	56673

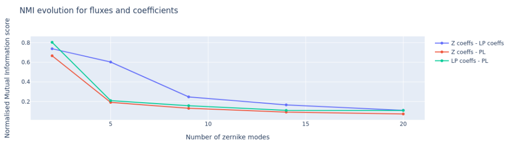
Table 10: DBSCAN Clustering for 14 Zernike modes datasets

20 Zernike modes :

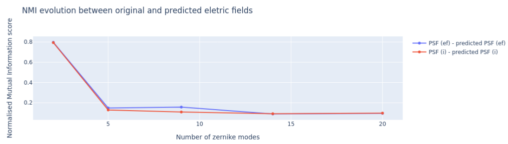
	ϵ	**neighbors**	**Clusters**	**CDM**	**CDV**	**Non noise points**
Zernike coeffs	0.53	3	1431	26.32	841.71	37668
LP coeffs	40.05	3	1558	24.44	801.25	38079
PL fluxes	1010	3	1647	18.25	566.01	30067
PSF electric field	0.25	4	1621	27.86	852.32	34332
PSF intensity	0.20	4	1559	31.47	986.86	49070
Pred PSF ef	0.33	3	1524	36.46	1222.67	55567
Pred PSF intensity	0.32	3	1643	35.18	1191.70	57810

Table 11: DBSCAN Clustering for 20 Zernike modes datasets

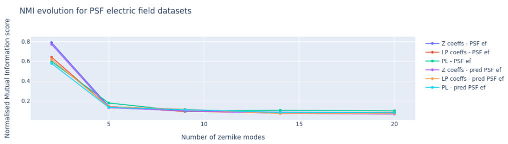
0.5.3 Results



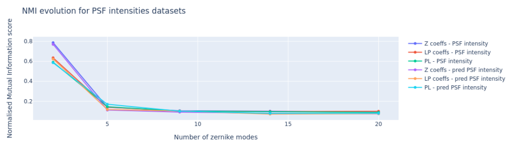
(a) Coefficients NMI evolution



(b) PSF to predicted PSF



(c) PSF electric fields to coeffs NMI evolution



(d) PSF intenities to coeffs NMI evolution

Figure 94: NMI evolution over number of Zernike modes

Figure 95: Coefficients NMI

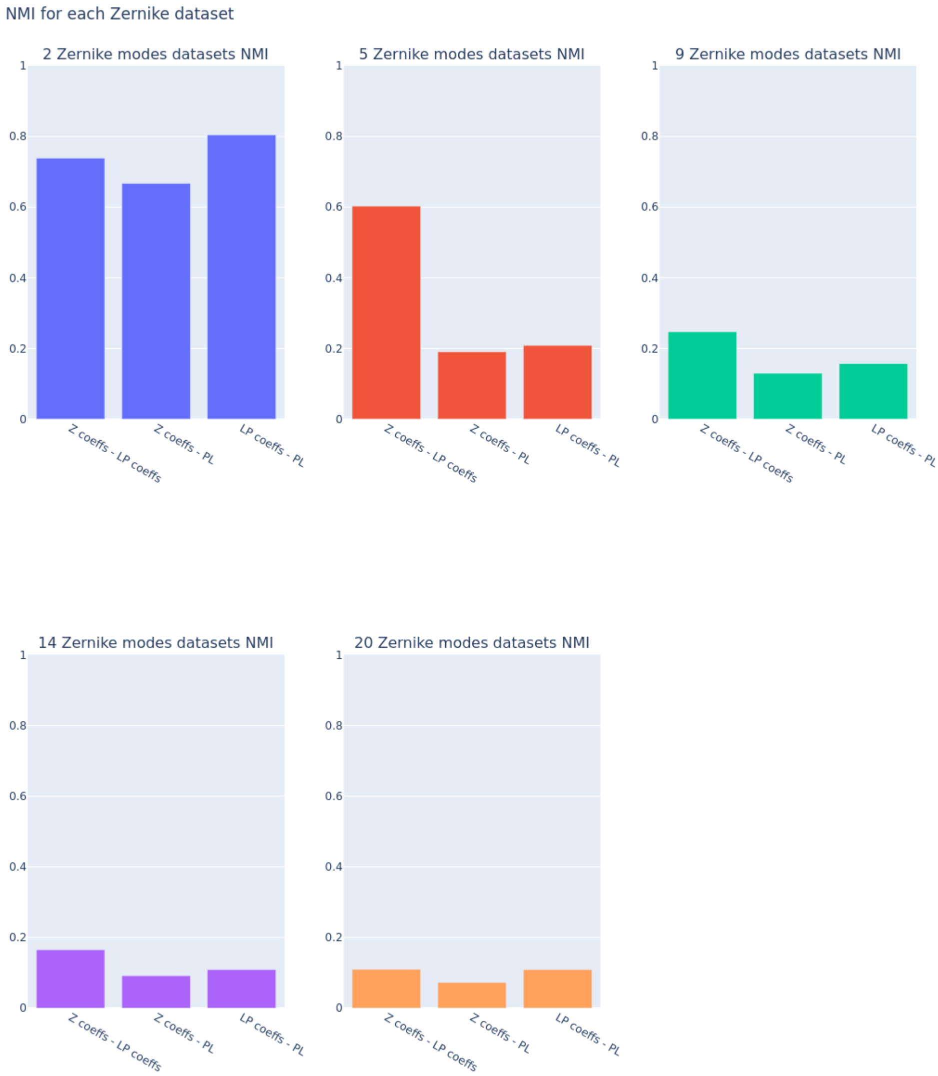
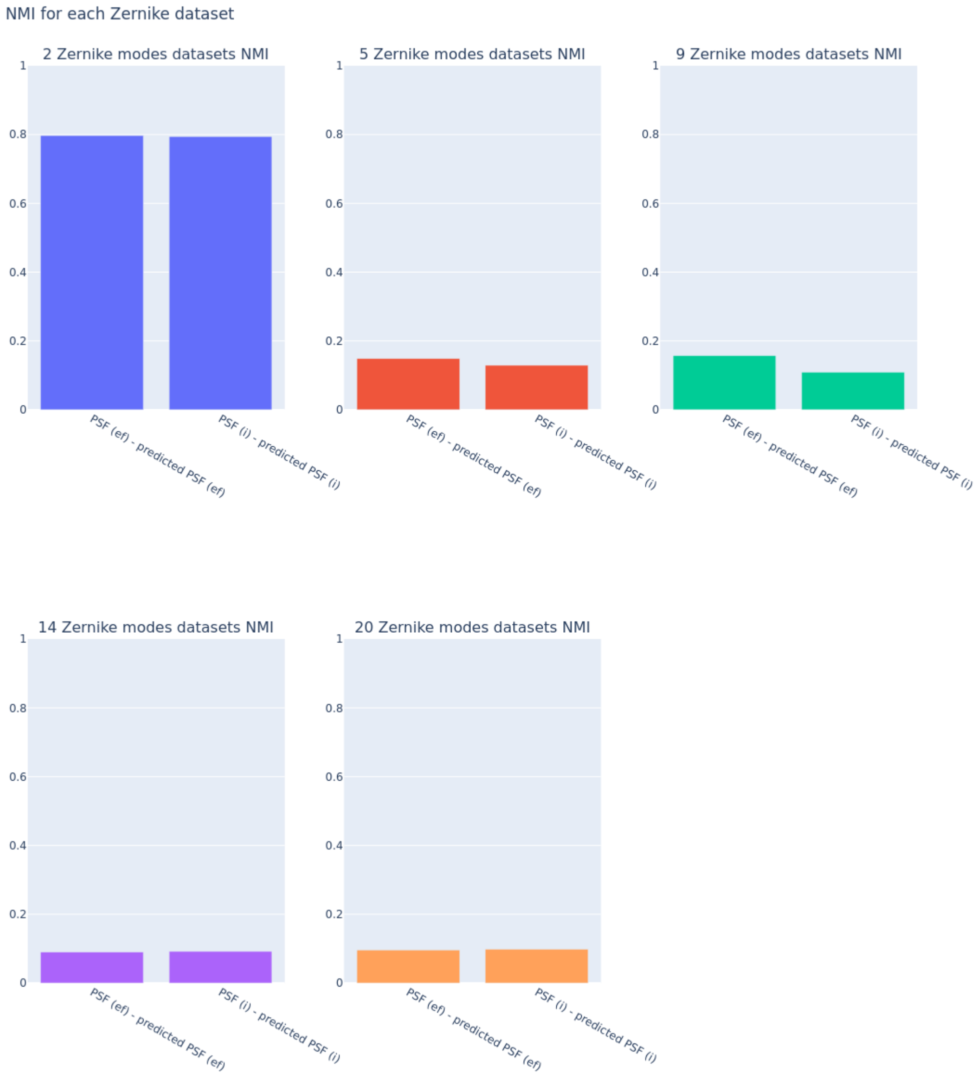


Figure 96: PSF electric fields to coeffs NMI evolution

Figure 97: PSF intensities to coeffs NMI evolution

Figure 98: PSF to predicted PSF NMI



Part IV

Mini Dataset 2 Zernike modes PL Information Determination

1 The data

1.1 Zernike coefficients dataset

A dataset of 1000 zernike coefficients is created for this report.In particular, each datapoint represent the coefficients of the first 2 Zernike modes, their values ranging between [-2, -1.8] and [1.8, 2].

1.2 PSFs coefficients dataset

A dataset of 1000 PSFs is created using the Zernike coefficients dataset.

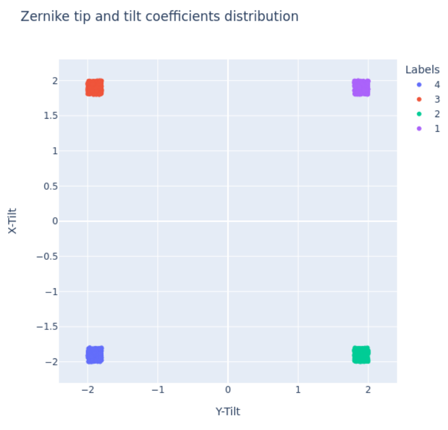
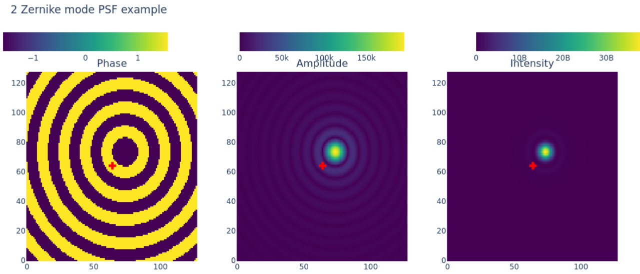
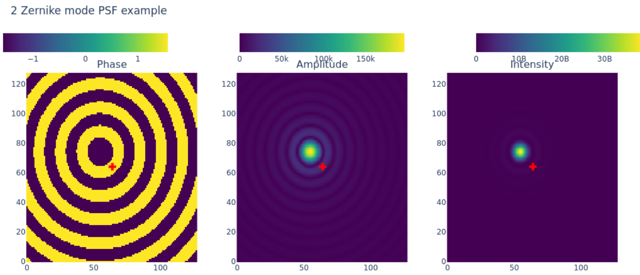


Figure 99: Example original sized PSF

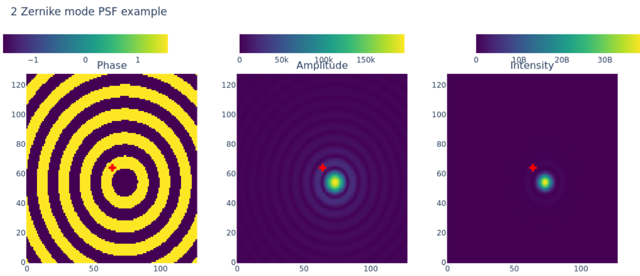
These ranges create 4 original clusters that will be used as reference.



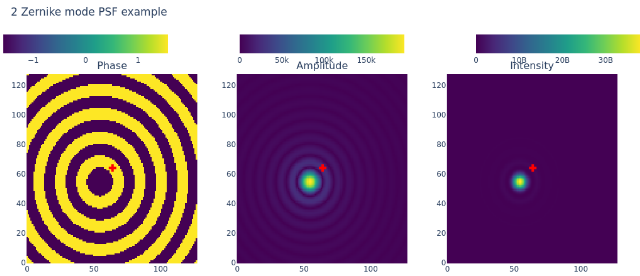
(a) Positive X-tilt and Positive Y-tilt PSF example



(b) Negative X-tilt and Positive Y-tilt PSF example



(c) Positive X-tilt and Negative Y-tilt PSF example



(d) Negative X-tilt and Negative Y-tilt PSF example

Figure 100: 2 Zernike modes PSF examples

1.3 LP mode coefficients dataset

A dataset of 1000 LP mode coefficients obtained from computing the overlap integral of the first 19 LP modes with the PSF dataset.

1.4 LP mode coefficients dataset

A dataset of 1000 PL output fluxes obtained from the PL transfer matrix and LP coefficients.

2 Preprocessing

2.1 PSF Intensities

The 1000x128x128 array is dimensionally reduced using PCA and UMAP both giving an array of 1000x19 projections of the PSF Intensities.

2.2 LP Coefficients

The 1000x19x2 array is dimensionally reduced using PCA and UMAP both giving an array of 1000x2 projections of the original LP coefficients.

2.3 Output fluxes

The 1000x19 array is dimensionally reduced using PCA and UMAP both giving an array of 1000x2 projections of the original LP coefficients.

3 Clustering

A series of different clustering algorithms are used:

- K-Means
- DBSCAN
- HDBSCAN
- Agglomerative clustering

The clusters obtained will be compared the original clusters using NMI

3.1 Zernike coefficients clustering

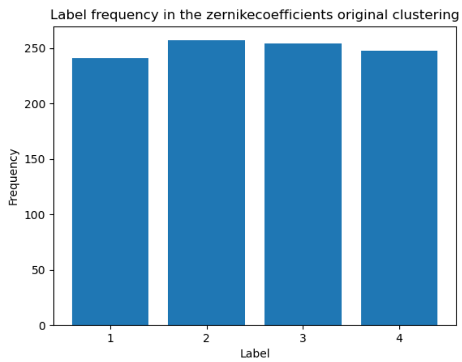
3.1.1 K-Means

As K-Means allows for the number of clusters to define, and we know that there are 4 in the original dataset, K-Means is used to find 4 clusters.

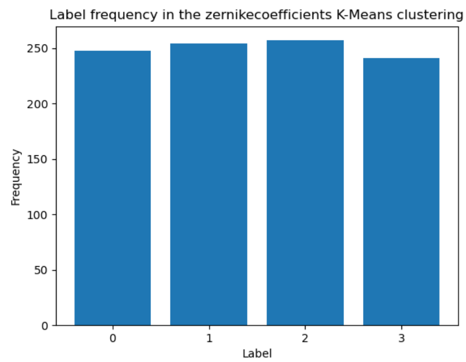
Number of clusters	Number of initializations
4	10

Table 12: K-Means hyperparameter configuration for Zernike coefficients clustering

The results are the following:

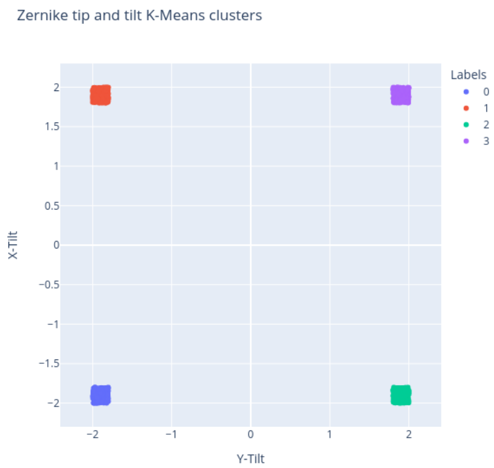


(a) Original cluster densities

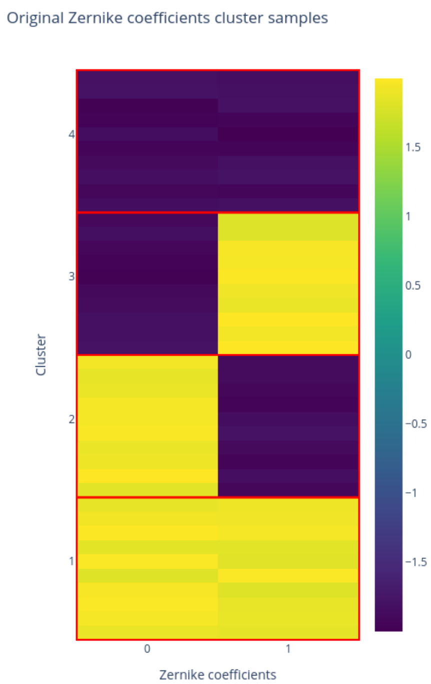


(b) K-Means clusters densities

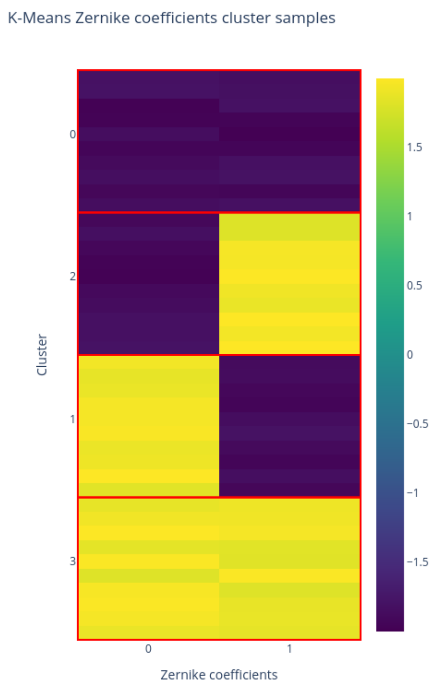
(c) Original clusters



(d) K-Means clusters



(e) Original cluster samples



(f) K-Means cluster samples

Figure 101: Comparison between original clustering and K-Means clustering

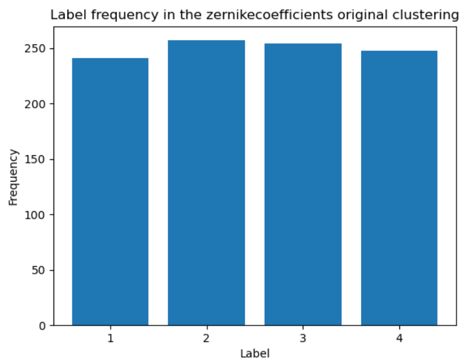
3.1.2 DBSCAN

A configuration that outputs 4 clusters is searched

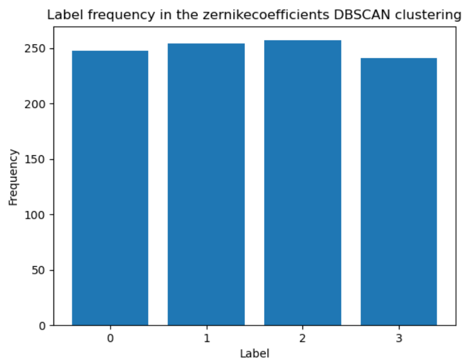
Number of neighbours	**Epsilon**
5	0.3

Table 13: DBSCAN hyperparameter configuration for Zernike coefficients clustering

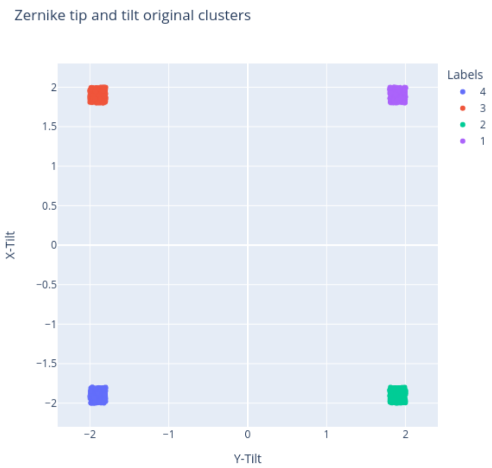
The results are the following:



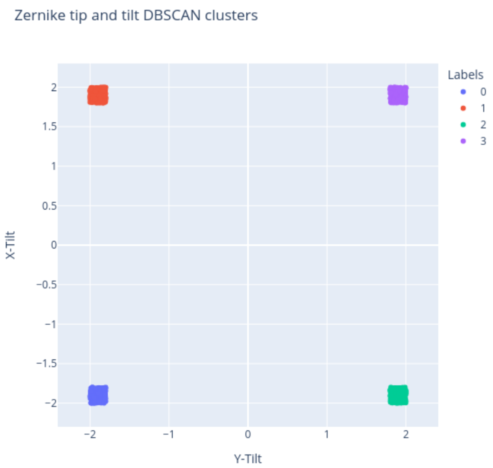
(a) Original cluster densities



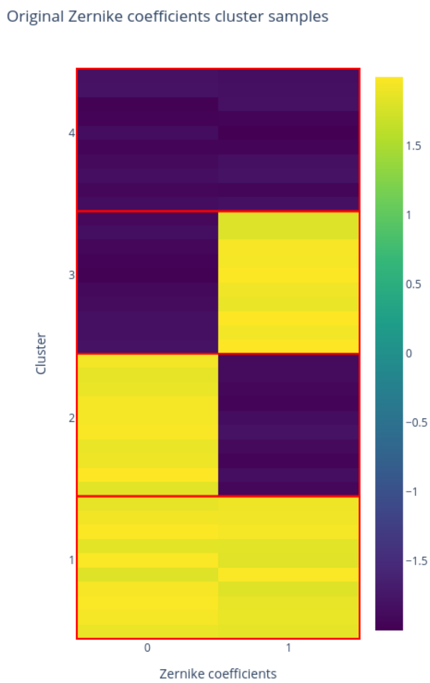
(b) DBSCAN clusters densities



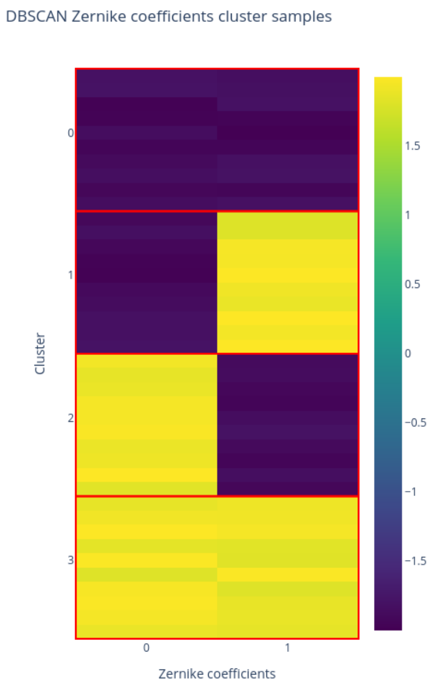
(c) Original clusters



(d) DBSCAN clusters



(e) Original cluster samples



(f) DBSCAN cluster samples

Figure 102: Comparison between original clustering and DBSCAN clustering

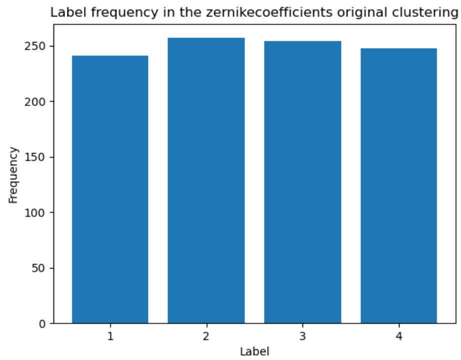
3.1.3 HDBSCAN

A configuration that outputs 4 clusters is searched.

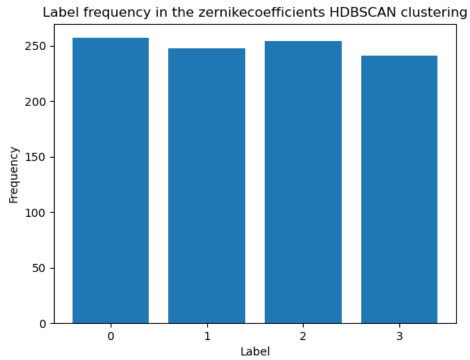
Minimum cluster size
5

Table 14: HDBSCAN hyperparameter configuration for Zernike coefficients clustering

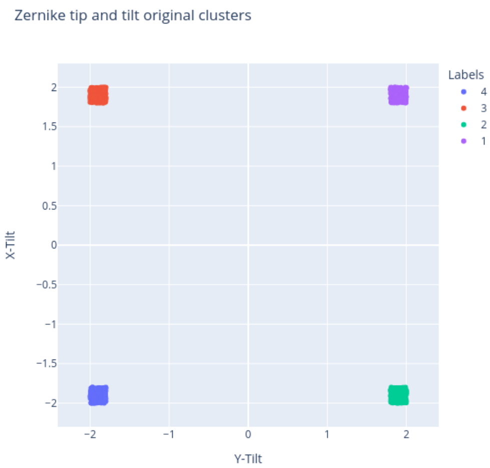
The results are the following:



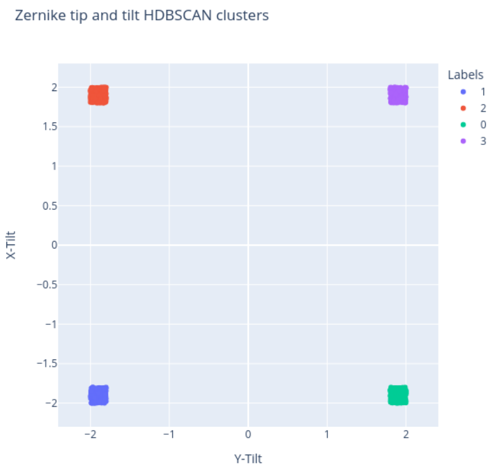
(a) Original cluster densities



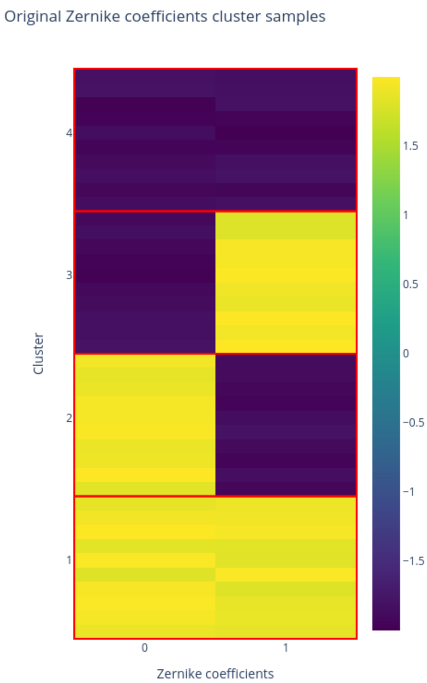
(b) HDBSCAN clusters densities



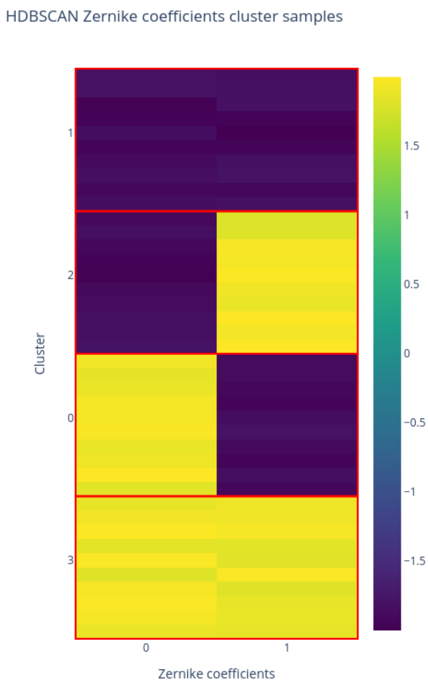
(c) Original clusters



(d) HDBSCAN clusters



(e) Original cluster samples



(f) HDBSCAN cluster samples

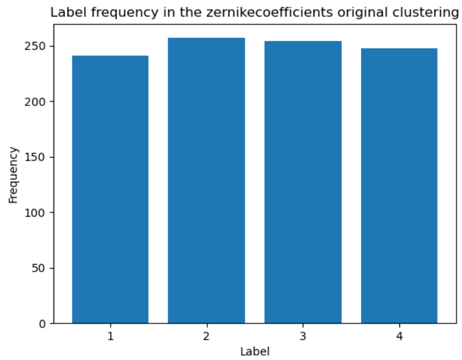
Figure 103: Comparison between original clustering and HDBSCAN clustering

3.1.4 Agglomerative clustering

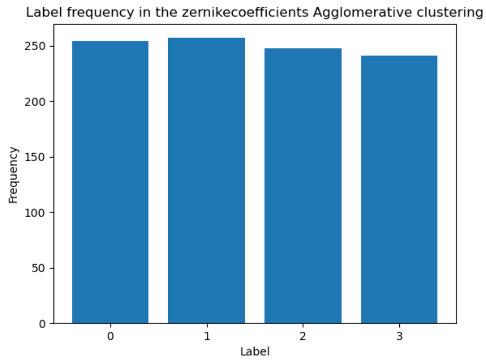
Number of clusters 4
5

Table 15: Agglomerative hyperparameter configuration for Zernike coefficients clustering

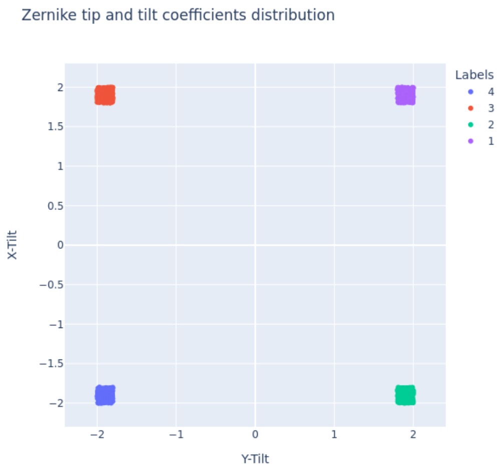
The results are the following:



(a) Original cluster densities

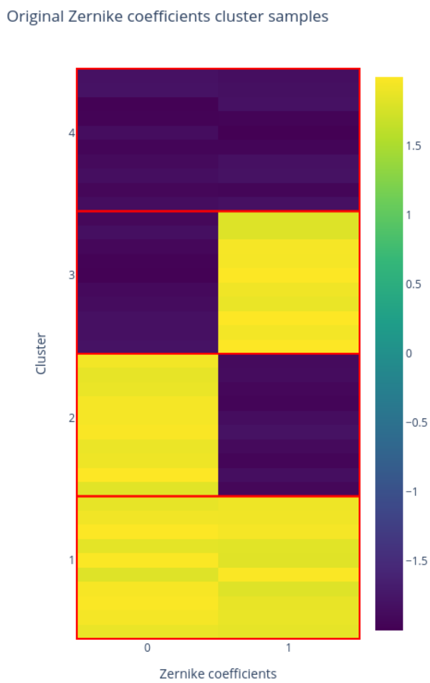


(b) Agglomerative clusters densities

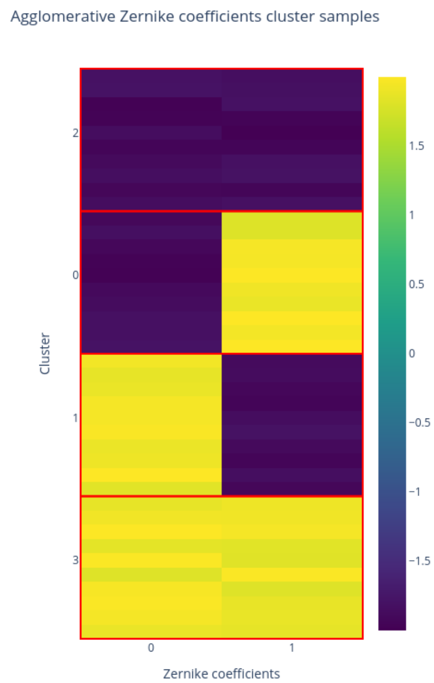


(c) Original clusters

(d) Agglomerative clusters



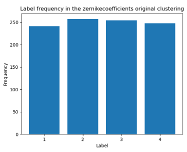
(e) Original cluster samples



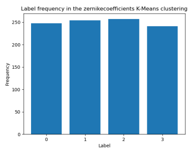
(f) Agglomerative cluster samples

Figure 104: Comparison between original clustering and Agglomerative clustering

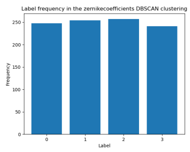
3.1.5 Summary



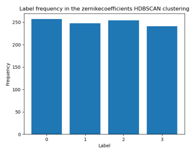
(a) Original cluster densities



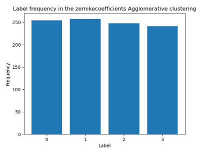
(b) K-means cluster densities



(c) DBSCAN cluster densities



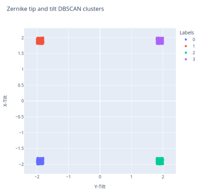
(d) HDBSCAN cluster densities



(e) Agglomerative cluster densities

(f) Original cluster

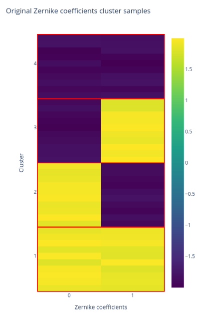
(g) K-means clusters



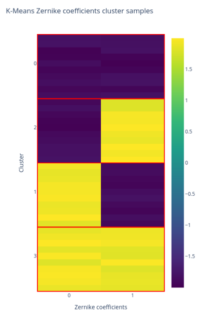
(h) DBSCAN cluster clusters

(i) HDBSCAN clusters

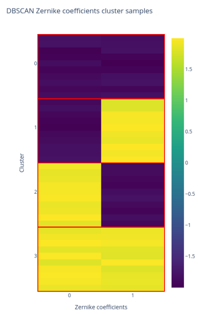
(j) Agglomerative clusters



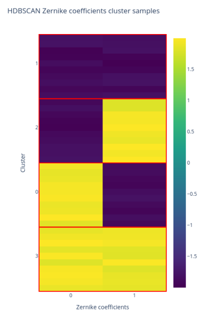
(k) Original cluster samples



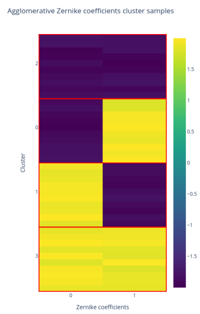
(l) K-means cluster samples



(m) DBSCAN cluster samples



(n) HDBSCAN cluster samples



(o) Agglomerative cluster samples

Figure 105: Comparison between clustering algorithms

	Original	**K-Means**	**DBSCAN**	**HDBSCAN**	**Agglomerative**
Original	\	1	1	1	1
K-Means		\	1	1	1
DBSCAN			\	1	1
HDBSCAN				\	1

Table 16: Normalized Mutual Information between clusters

3.2 LP coefficients clustering

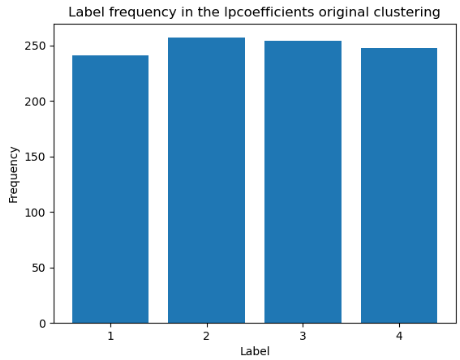
3.2.1 K-Means

As K-Means allows for the number of clusters to be defined, and we know that there are 4 in the original dataset, K-Means is used to find 4 clusters.

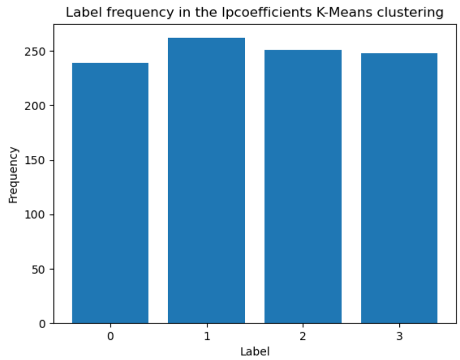
	Number of clusters	**Number of initializations**
Original LP coefficients	4	10
PCA LP coefficients	4	10
UMAP LP coefficients	4	10

Table 17: K-Means hyperparameter configuration for c coefficients clustering

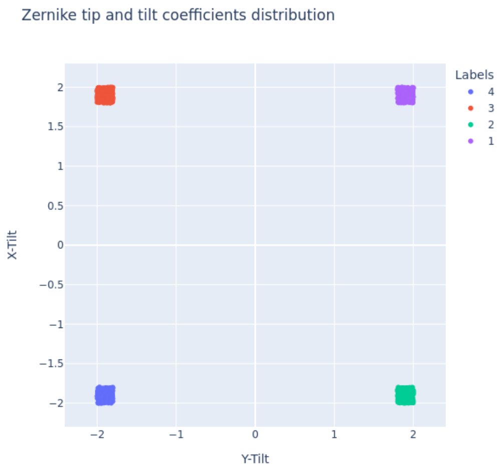
The results are the following:



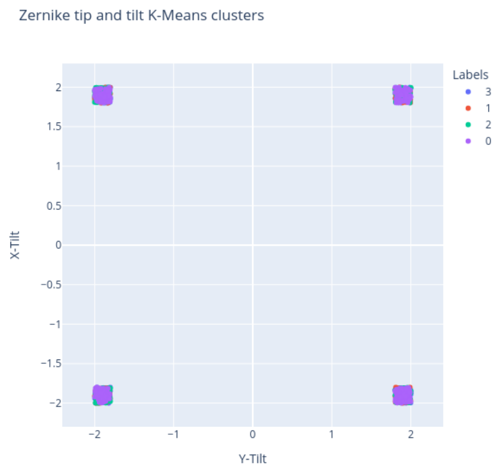
(a) Original cluster densities



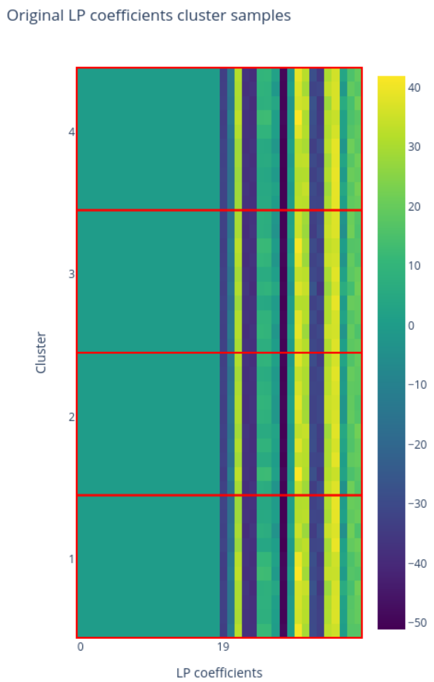
(b) K-Means clusters densities



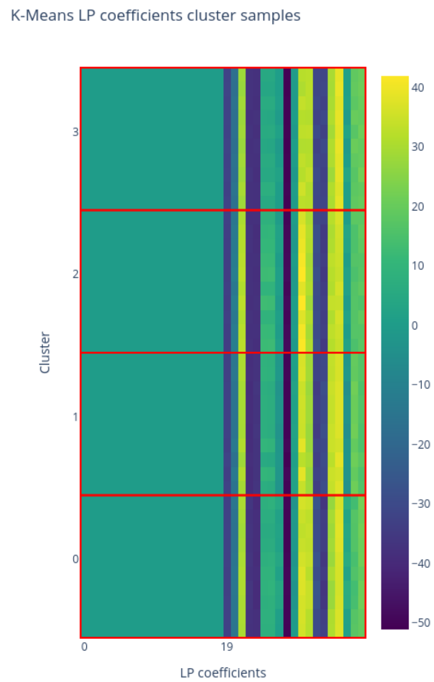
(c) Original clusters



(d) K-Means clusters

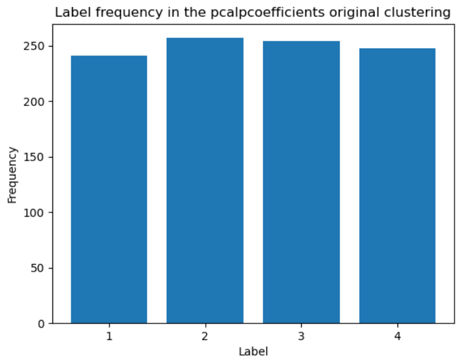


(e) Original cluster samples

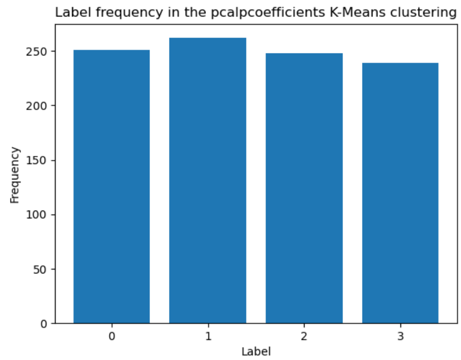


(f) K-Means cluster samples

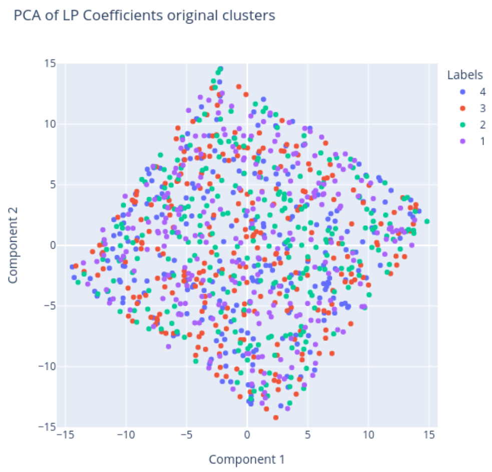
Figure 106: Comparison between original clustering and K-Means clustering from



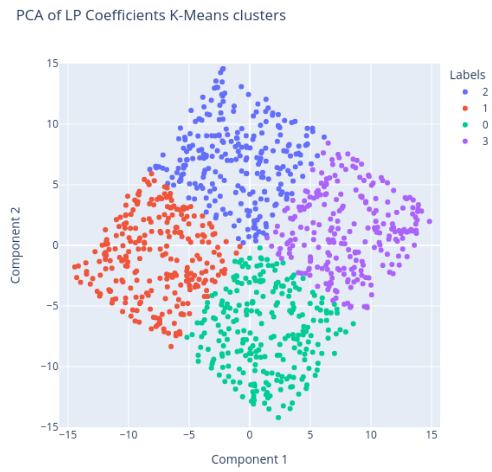
(a) Original cluster densities from PCA



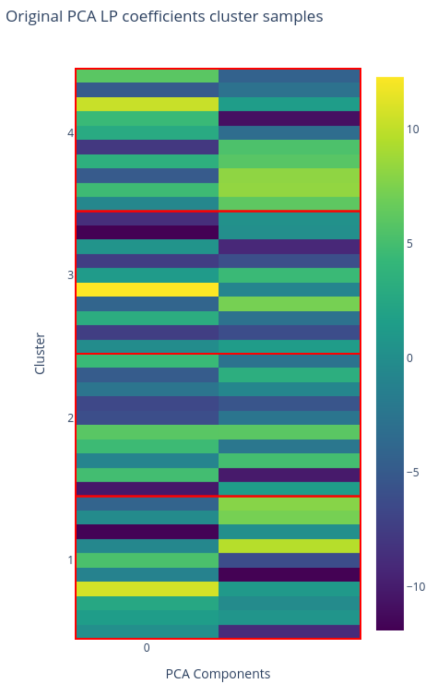
(b) K-Means clusters densities from PCA



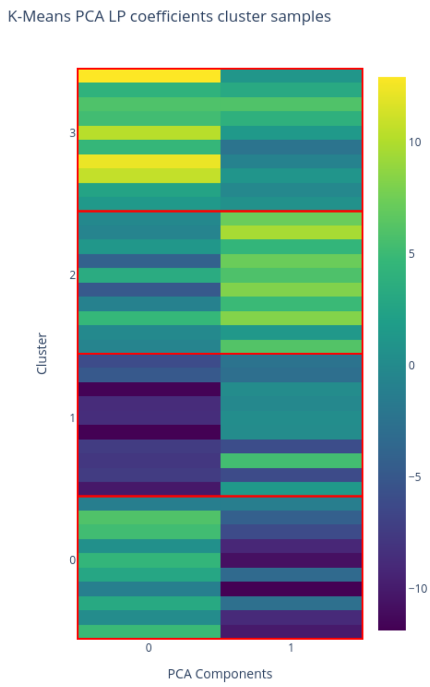
(c) Original clusters



(d) K-Means clusters from PCA

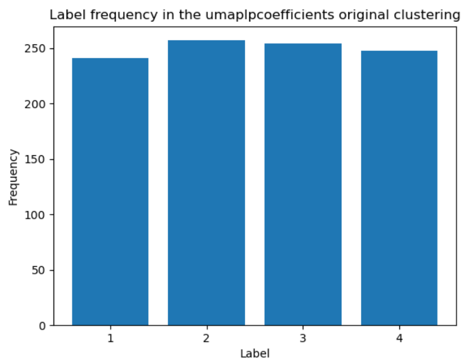


(e) Original cluster samples from PCA

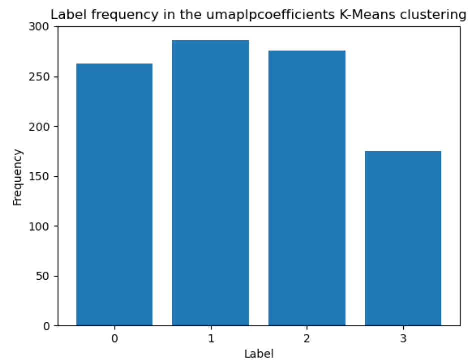


(f) K-Means cluster samples from PCA

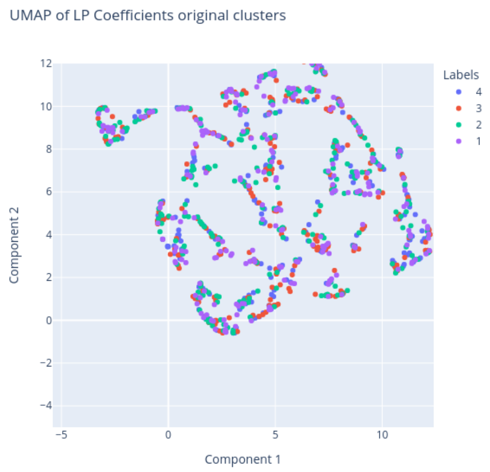
Figure 107: Comparison between original clustering and K-Means clustering from



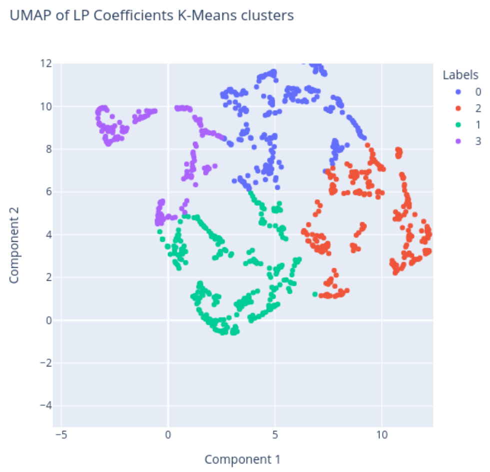
(a) Original cluster densities from UMAP



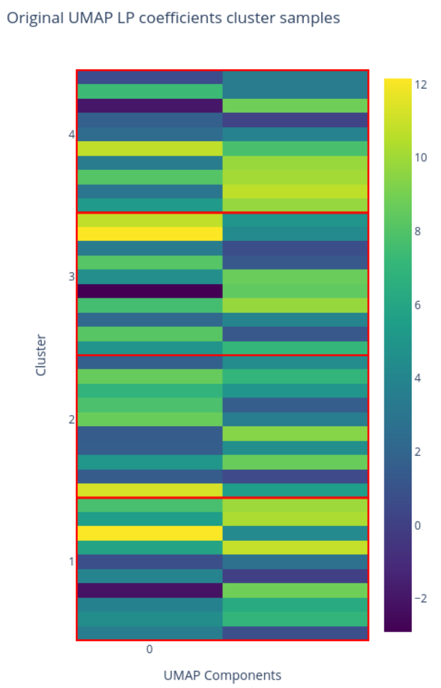
(b) K-Means clusters densities from UMAP



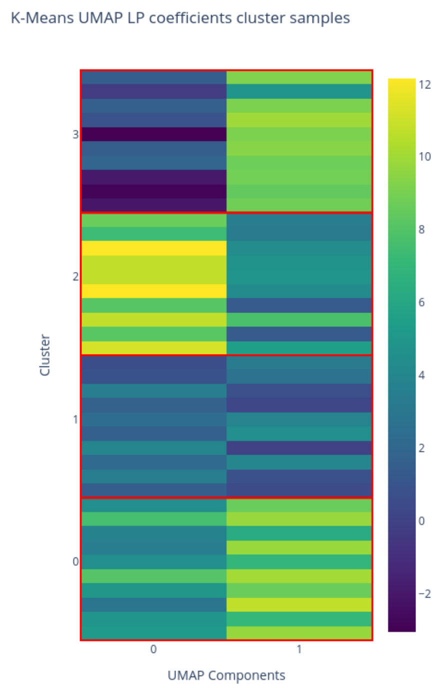
(c) Original clusters from UMAP



(d) K-Means clusters from UMAP



(e) Original cluster samples from UMAP



(f) K-Means cluster samples from UMAP

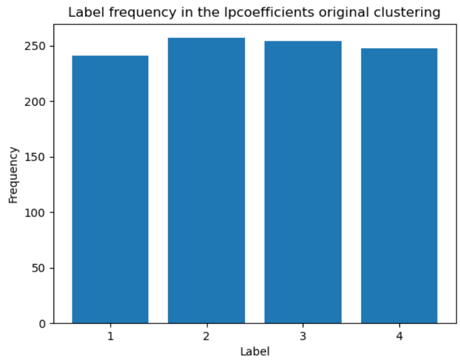
3.2.2 DBSCAN

A configuration that outputs 4 clusters is searched

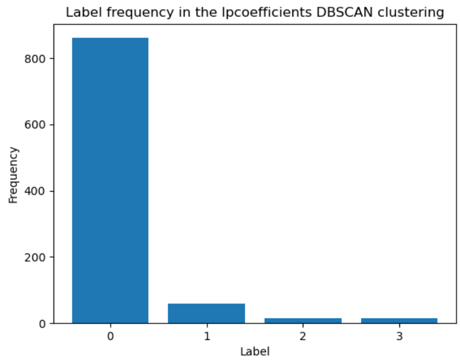
	Number of neighbours	**Epsilon**
Original LP coefficients	15	1.52
PCA LP coefficients	15	1.52
UMAP LP coefficients	10	0.85

Table 18: DBSCAN hyperparameter configuration for LP coefficients clustering

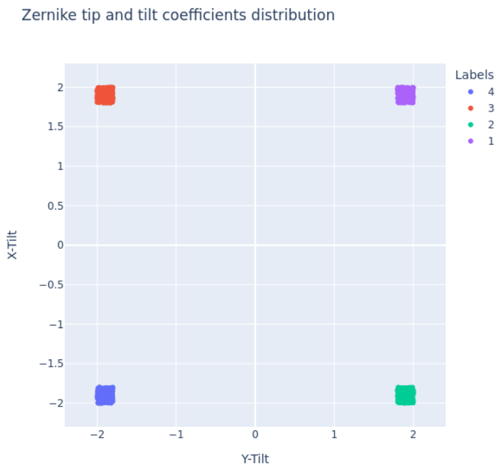
The results are the following:



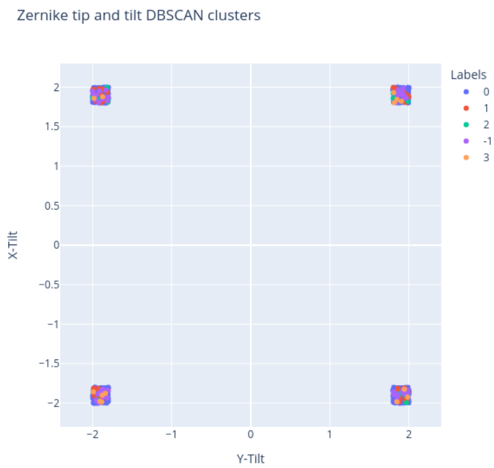
(a) Original cluster densities



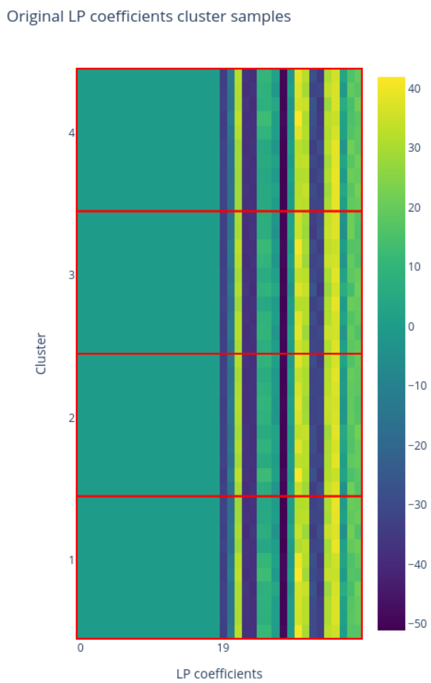
(b) DBSCAN clusters densities



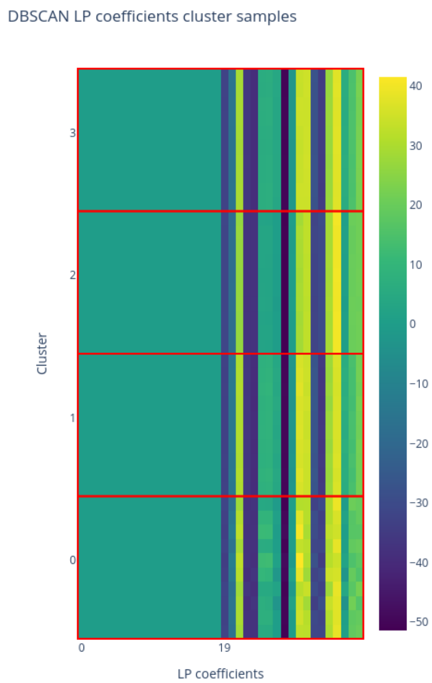
(c) Original clusters



(d) DBSCAN clusters

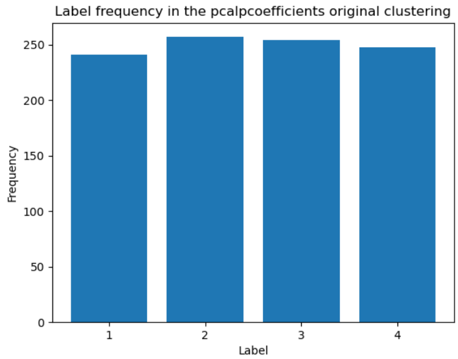


(e) Original cluster samples

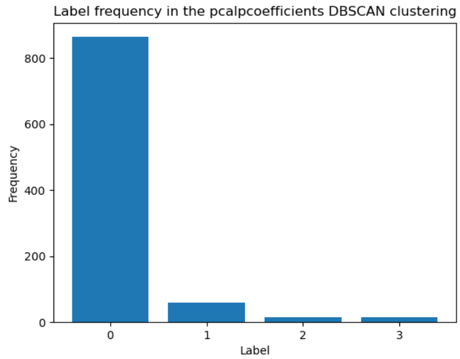


(f) DBSCAN cluster samples

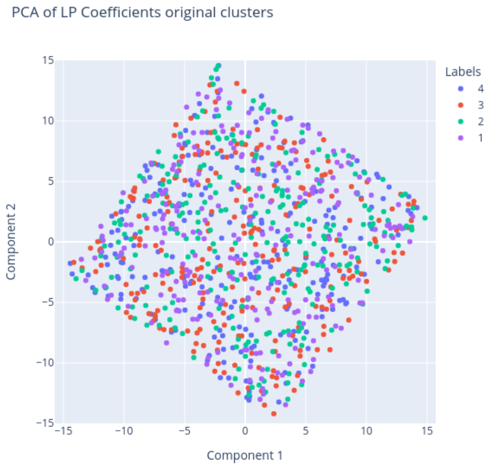
Figure 109: Comparison between original clustering and DBSCAN clustering



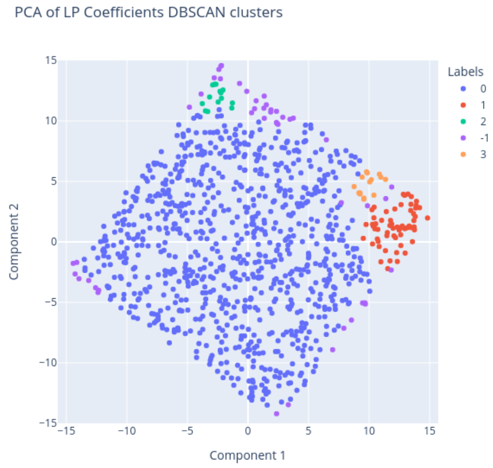
(a) Original cluster densities from PCA



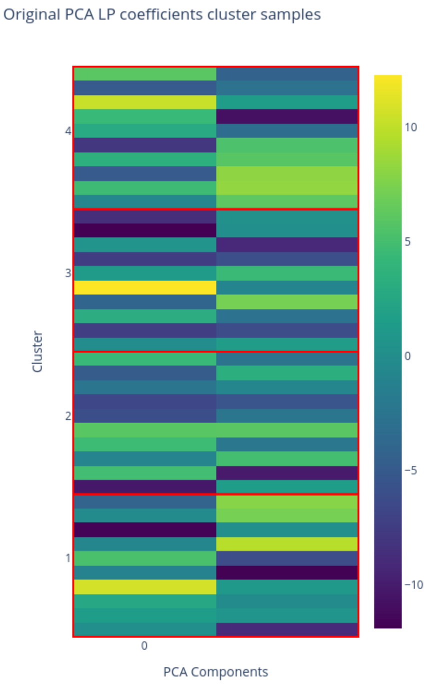
(b) DBSCAN clusters densities from PCA



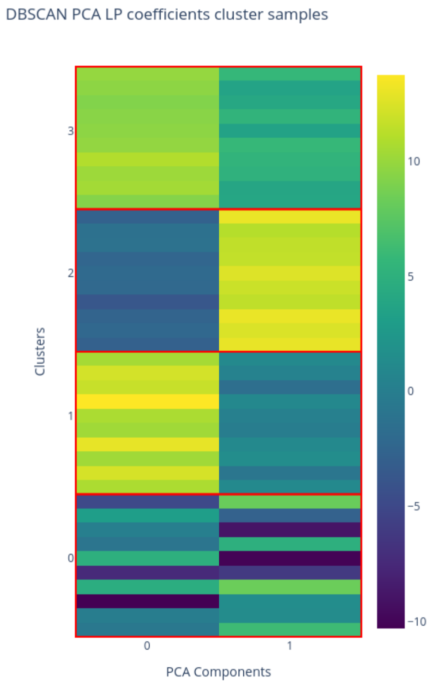
(c) Original clusters from PCA



(d) DBSCAN clusters from PCA

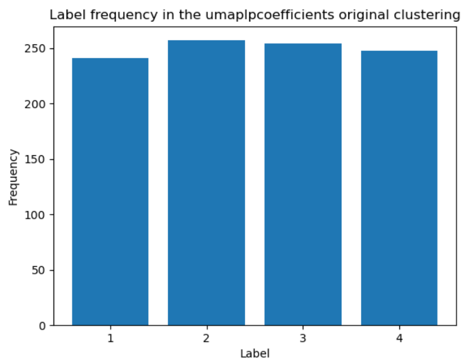


(e) Original cluster samples from PCA

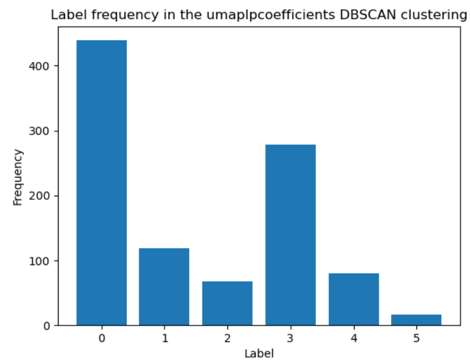


(f) DBSCAN cluster samples from PCA

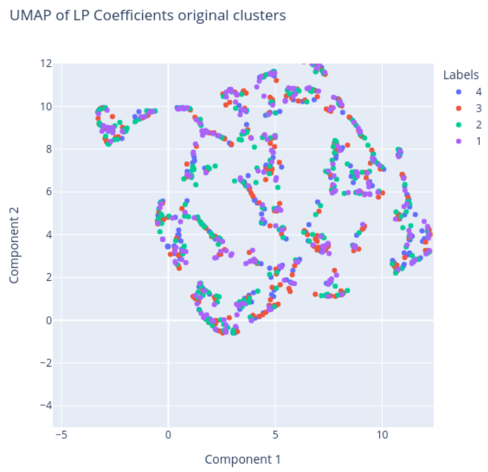
Figure 110: Comparison between original clustering and DBSCAN clustering from



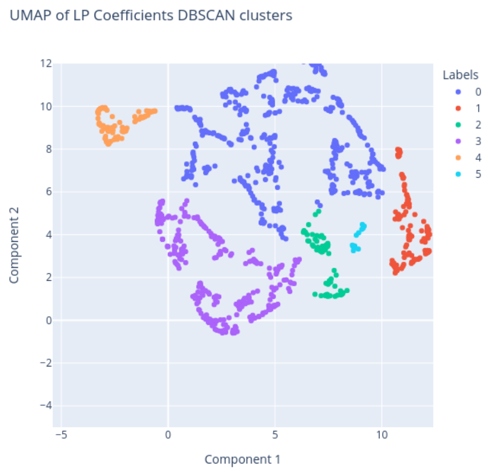
(a) Original cluster densities from UMAP



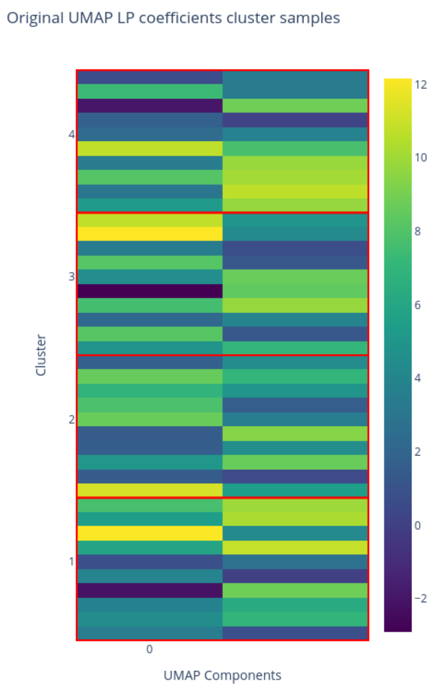
(b) DBSCAN clusters densities from UMAP



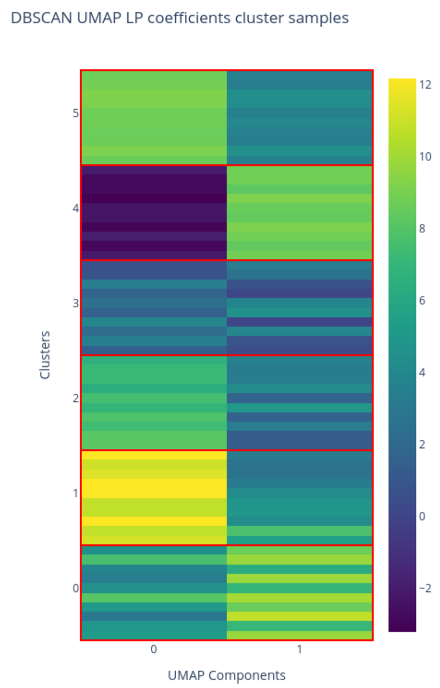
(c) Original clusters from UMAP



(d) DBSCAN clusters from UMAP



(e) Original cluster samples from UMAP



(f) DBSCAN cluster samples from UMAP

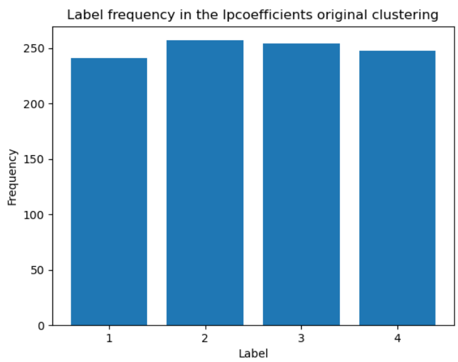
3.2.3 HDBSCAN

A configuration that outputs 4 clusters is searched.

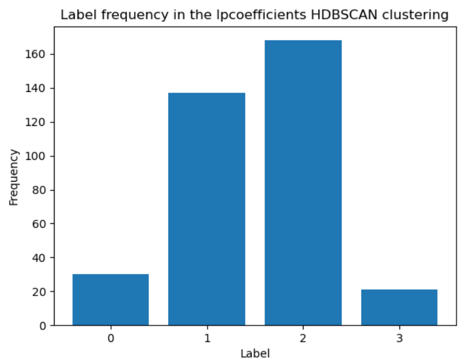
	Minimum cluster size
Original LP coefficients	21
PCA LP coefficients	21
UMAP LP coefficients	25

Table 19: HDBSCAN hyperparameter configuration for LP coefficients clustering

The results are the following:

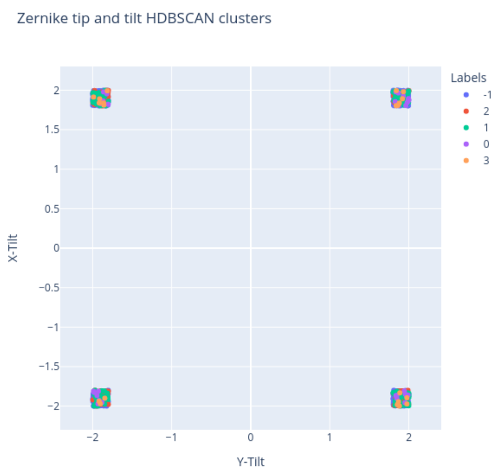


(a) Original cluster densities

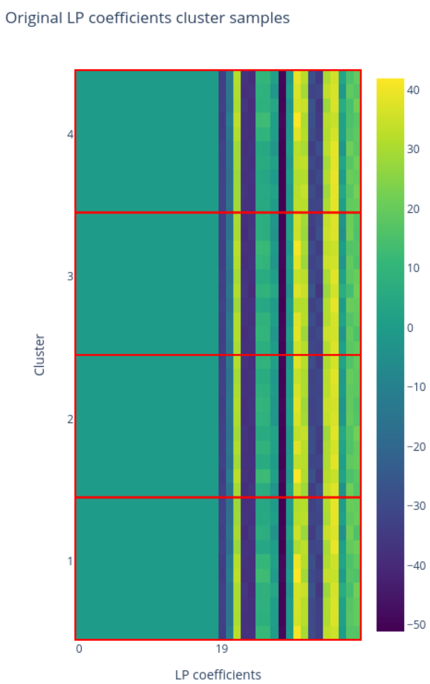


(b) HDBSCAN clusters densities

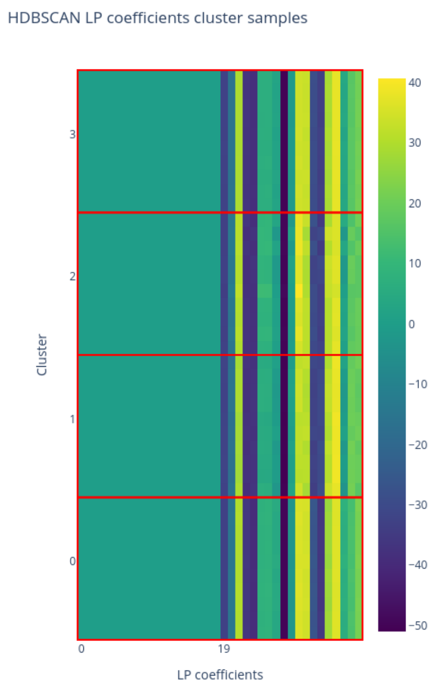
(c) Original clusters



(d) HDBSCAN clusters

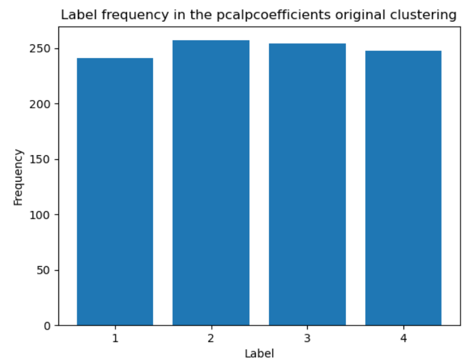


(e) Original cluster samples

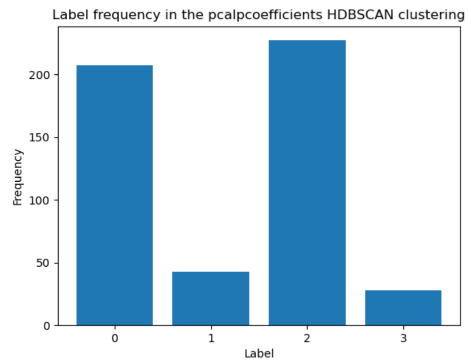


(f) HDBSCAN cluster samples

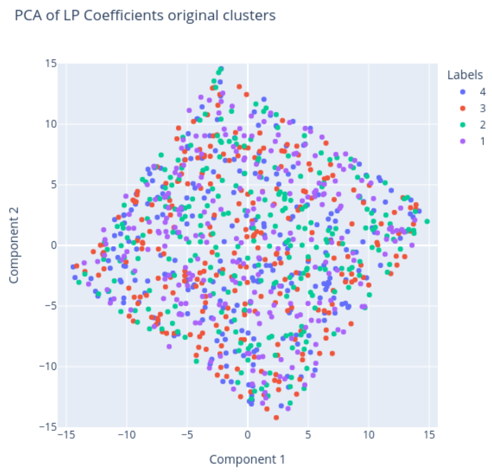
Figure 112: Comparison between original clustering and HDBSCAN clustering



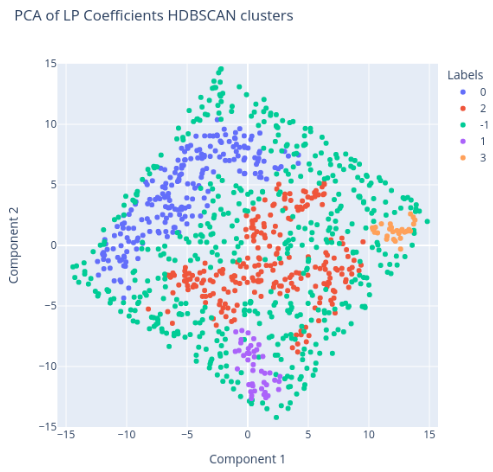
(a) Original cluster densities from PCA



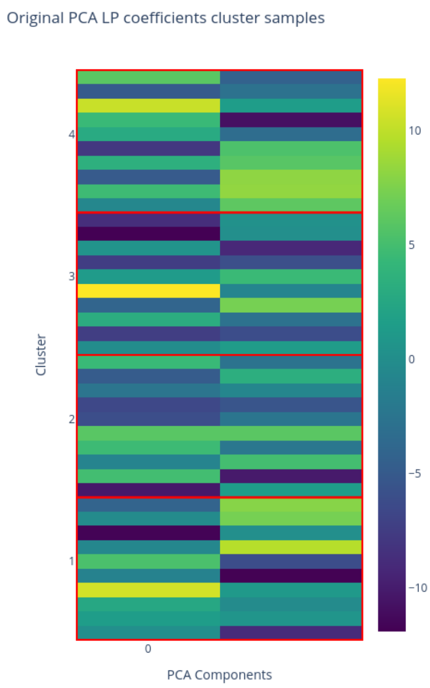
(b) HDBSCAN clusters densities from PCA



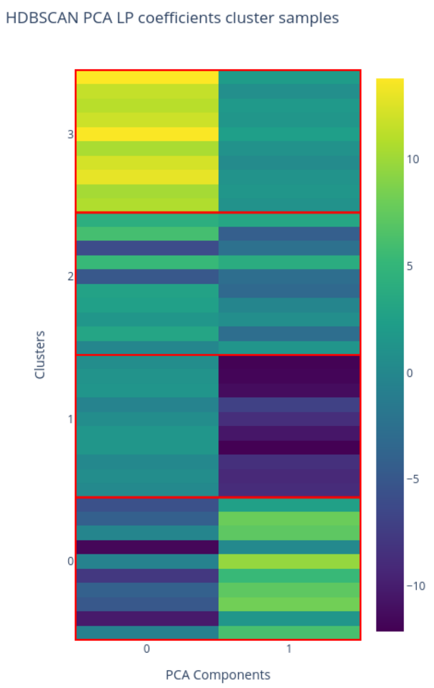
(c) Original clusters from PCA



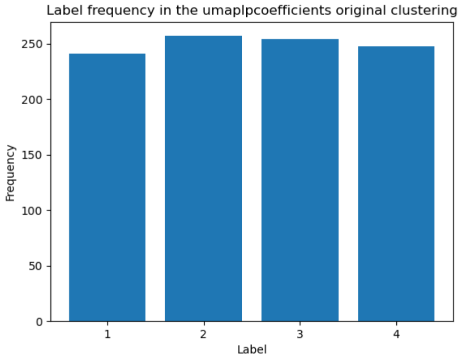
(d) HDBSCAN clusters from PCA



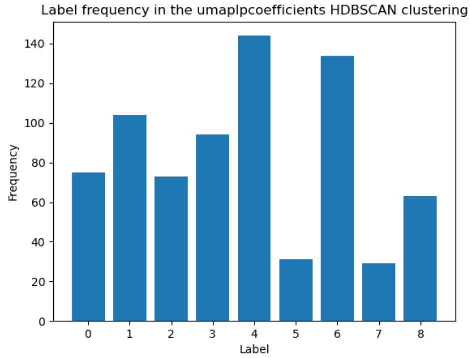
(e) Original cluster samples from PCA



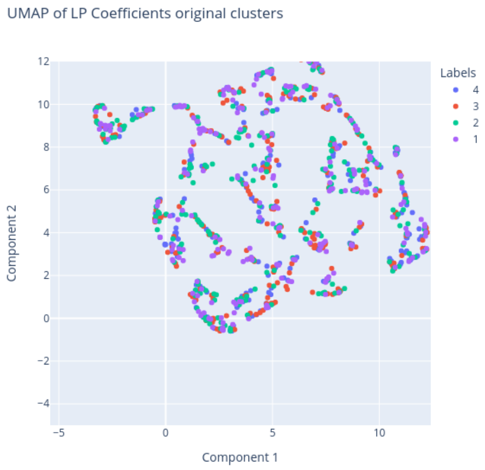
(f) HDBSCAN cluster samples from PCA



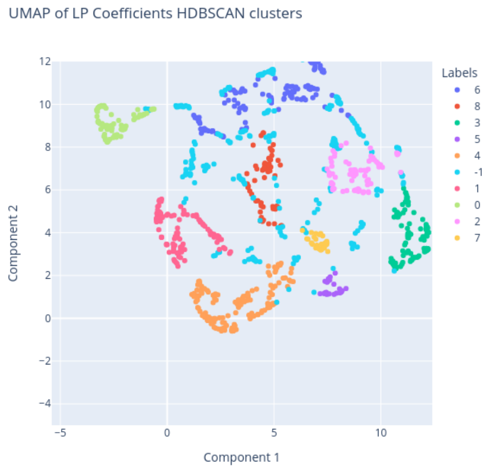
(a) Original cluster densities from UMAP



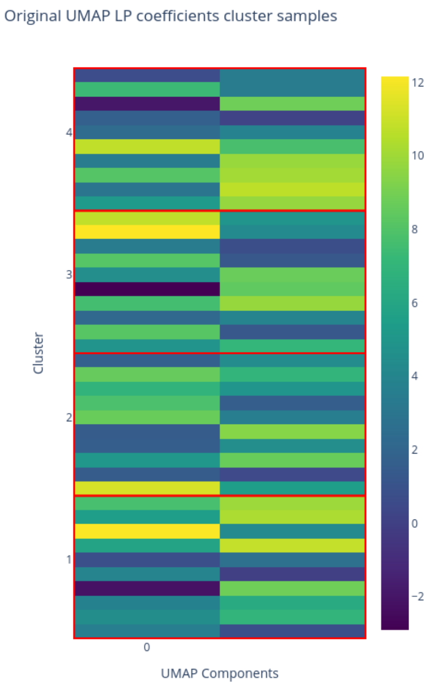
(b) HDBSCAN clusters densities from UMAP



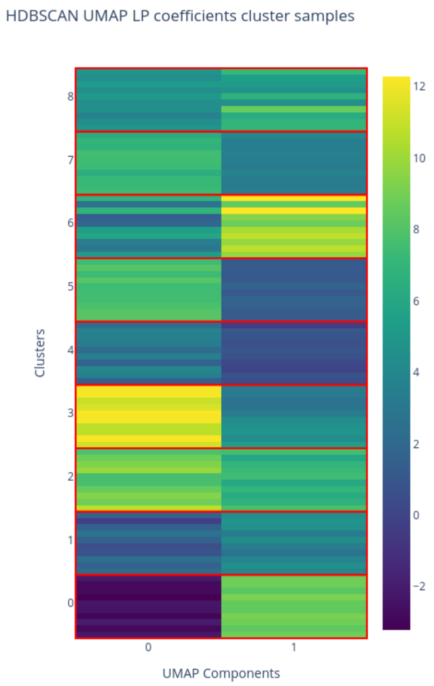
(c) Original clusters from UMAP



(d) HDBSCAN clusters from UMAP



(e) Original cluster samples from UMAP



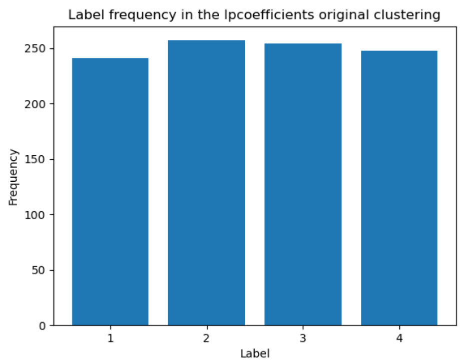
(f) HDBSCAN cluster samples from UMAP

3.2.4 Agglomerative clustering

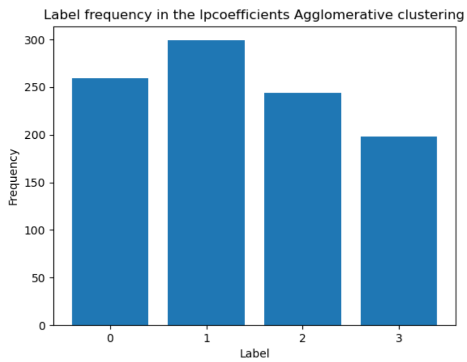
	Number of clusters
Original LP coefficients	4
PCA LP coefficients	4
UMAP LP coefficients	4

Table 20: Agglomerative hyperparameter configuration for LP coefficients clustering

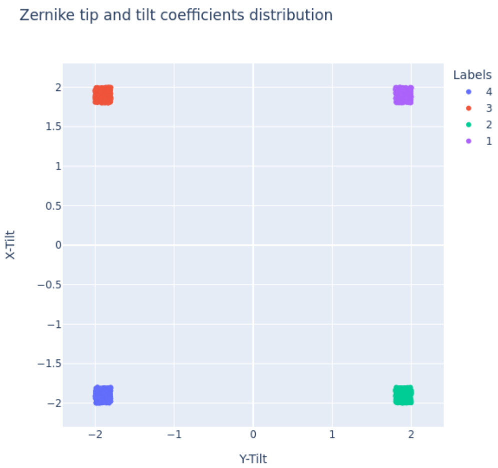
The results are the following:



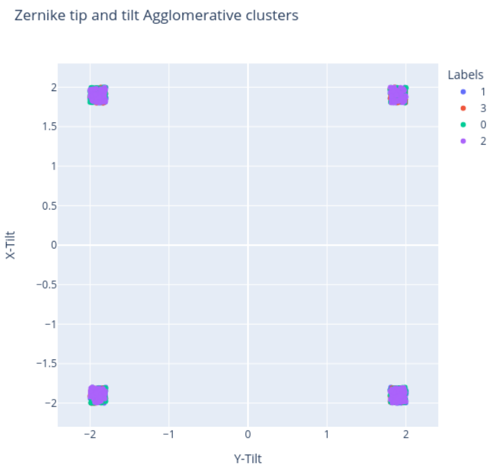
(a) Original cluster densities



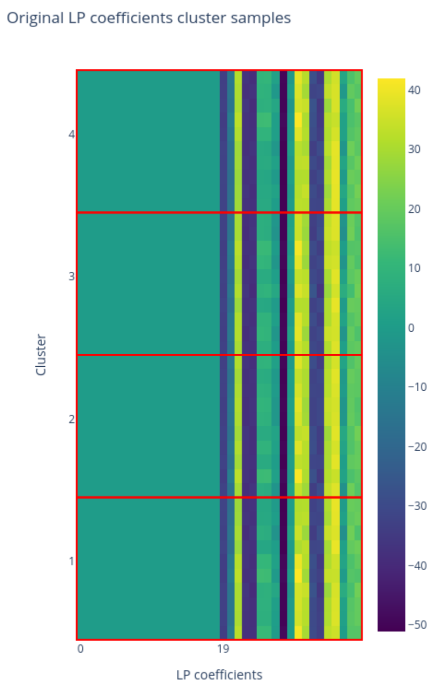
(b) Agglomerative clusters densities



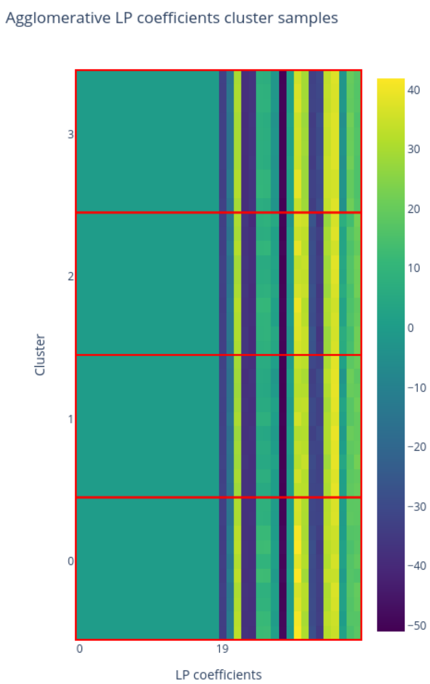
(c) Original clusters



(d) Agglomerative clusters

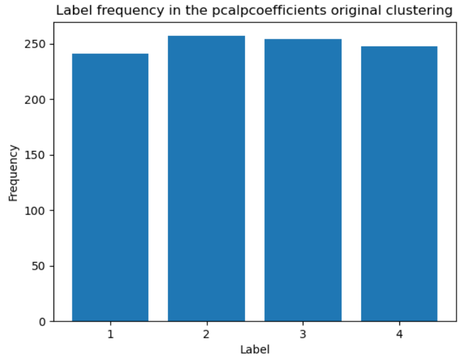


(e) Original cluster samples

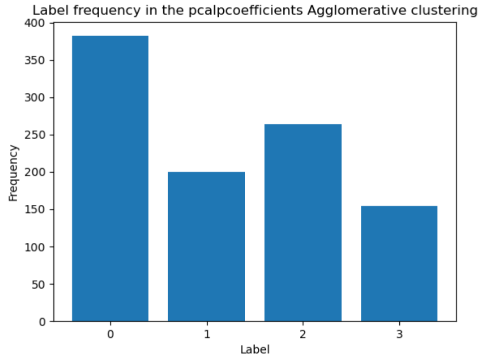


(f) Agglomerative cluster samples

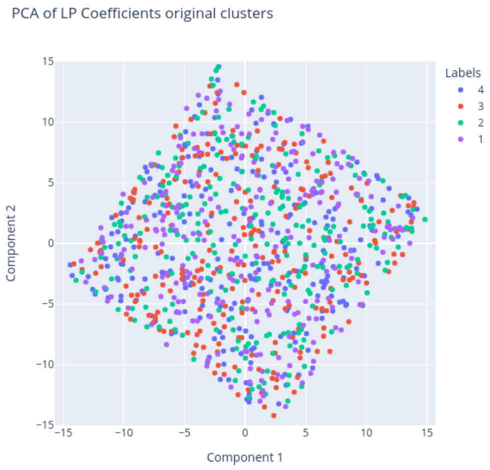
Figure 115: Comparison between original clustering and Agglomerative clustering



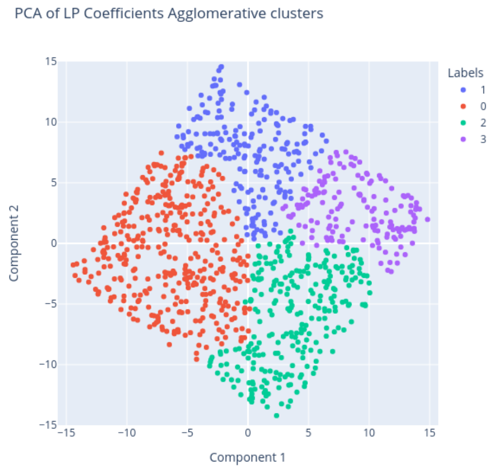
(a) Original cluster densities from PCA



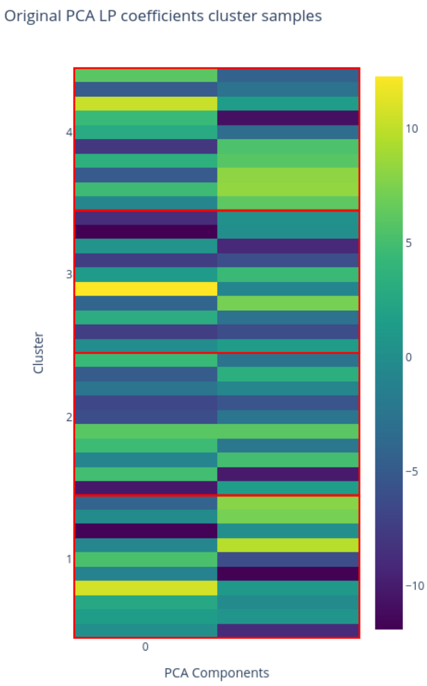
(b) Agglomerative clusters densities from PCA



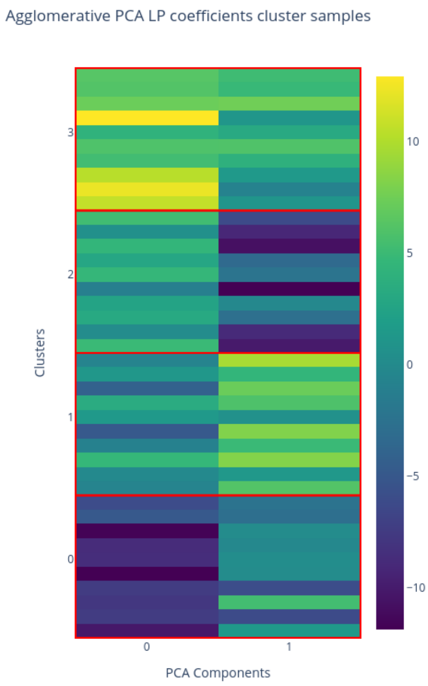
(c) Original clusters from PCA



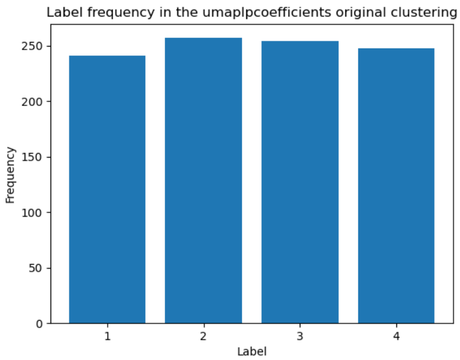
(d) Agglomerative clusters from PCA



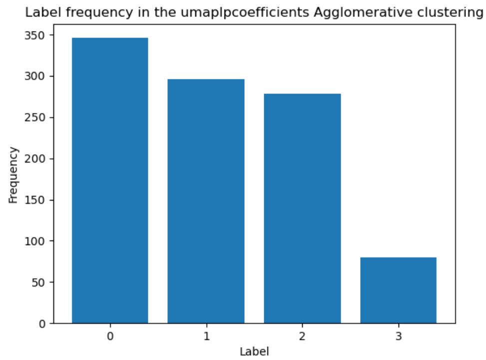
(e) Original cluster samples from PCA



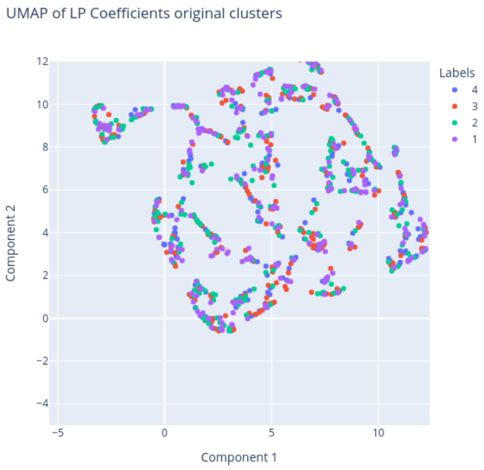
(f) Agglomerative cluster samples from PCA



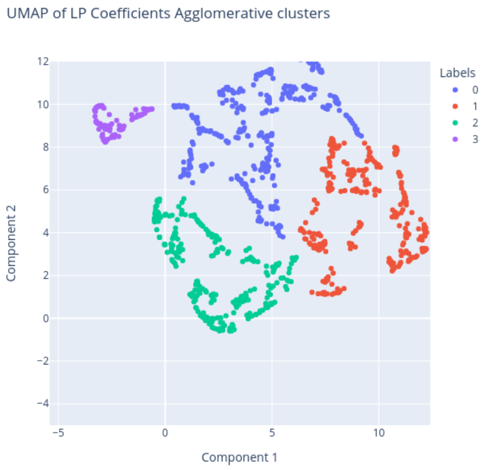
(a) Original cluster densities from UMAP



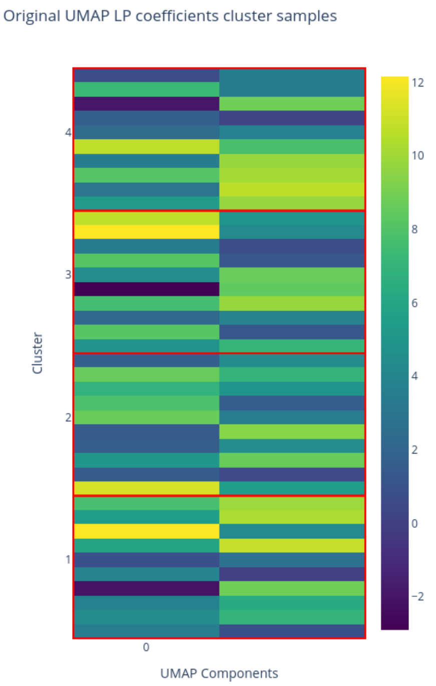
(b) Agglomerative clusters densities from UMAP



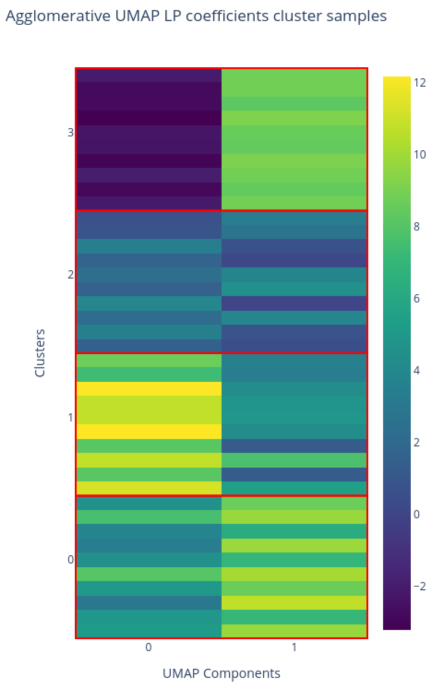
(c) Original clusters from UMAP



(d) Agglomerative clusters from UMAP

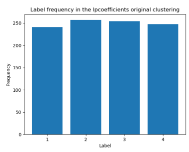


(e) Original cluster samples from UMAP

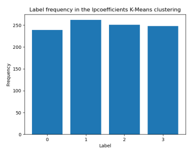


(f) Agglomerative cluster samples from UMAP

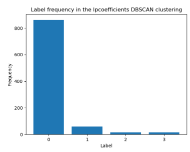
3.2.5 Summary



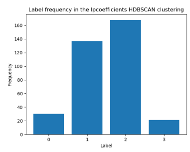
(a) Original cluster densities



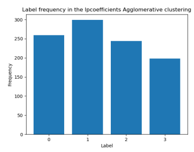
(b) K-means cluster densities



(c) DBSCAN cluster densities

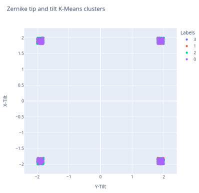


(d) HDBSCAN cluster densities

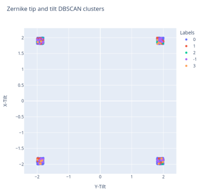


(e) Agglomerative cluster densities

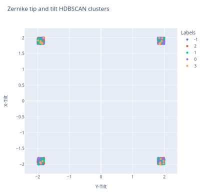
(f) Original cluster



(g) K-means clusters

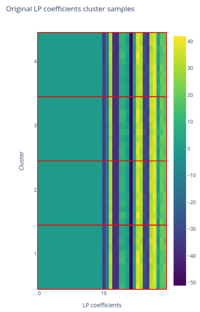


(h) DBSCAN cluster clusters

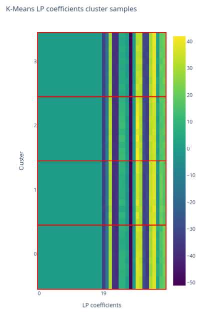


(i) HDBSCAN clusters

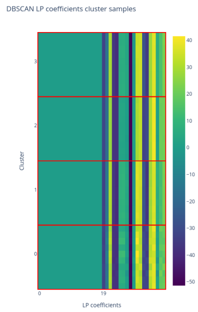
(j) Agglomerative clusters



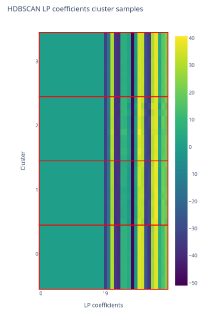
(k) Original cluster samples



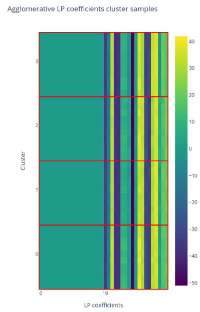
(l) K-means cluster samples



(m) DBSCAN cluster samples



(n) HDBSCAN cluster samples

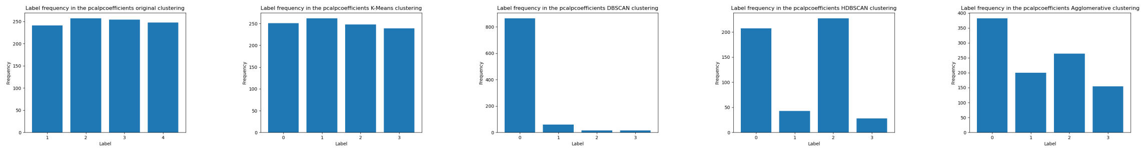


(o) Agglomerative cluster samples

Figure 118: Comparison between clustering LP coefficients algorithms

	Original	**K-Means**	**DBSCAN**	**HDBSCAN**	**Agglomerative**
Original		0.002	0.008	0.004	0.002
K-Means			0.154	0.143	0.695
DBSCAN				0.175	0.142
HDBSCAN					0.002

Table 21: Normalized Mutual Information between original LP coefficients clusters



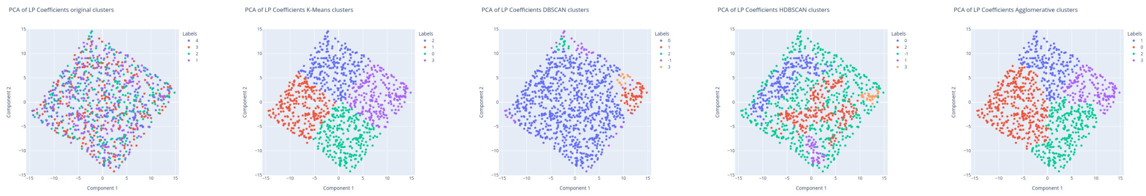
(a) Original cluster densities from PCA

(b) K-means cluster densities from PCA

(c) DBSCAN cluster densities from PCA

(d) HDBSCAN cluster densities from PCA

(e) Agglomerative cluster densities from PCA



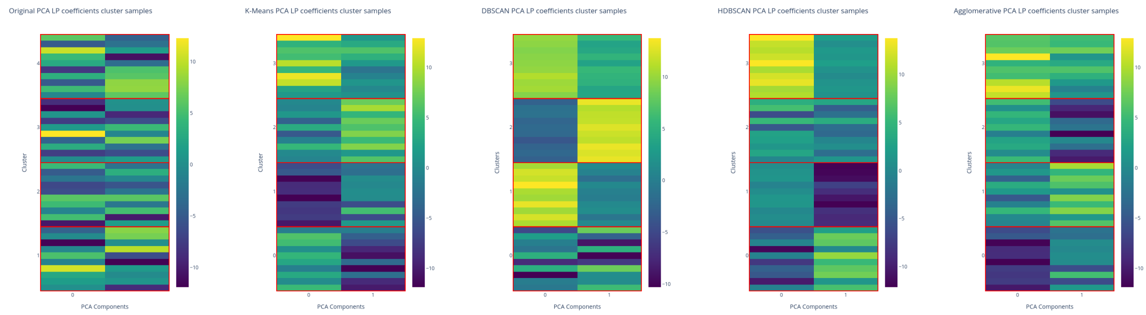
(f) Original cluster from PCA

(g) K-means clusters from PCA

(h) DBSCAN cluster clusters from PCA

(i) HDBSCAN clusters from PCA

(j) Agglomerative clusters from PCA



(k) Original cluster samples from PCA

(l) K-means cluster samples from PCA

(m) DBSCAN cluster samples-from PCA

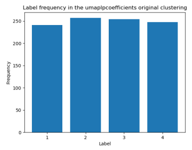
(n) HDBSCAN cluster samples from PCA

(o) Agglomerative cluster samples from PCA

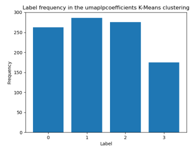
Figure 119: Comparison between clustering PCA LP coefficients algorithms

	Original	**K-Means**	**DBSCAN**	**HDBSCAN**	**Agglomerative**
Original		0.002	0.003	0.004	0.003
K-Means			0.150	0.143	0.642
DBSCAN				0.174	0.209
HDBSCAN					0.003

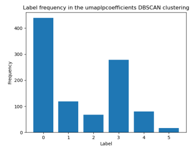
Table 22: Normalized Mutual Information between PCA LP coefficients clusters



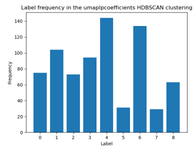
(a) Original cluster densities from UMAP



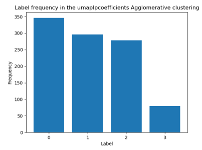
(b) K-means cluster densities from UMAP



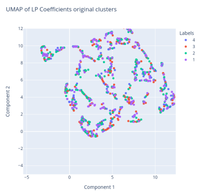
(c) DBSCAN cluster densities from UMAP



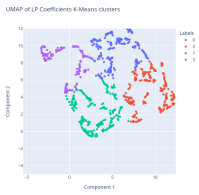
(d) HDBSCAN cluster densities from UMAP



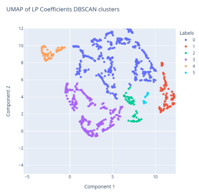
(e) Agglomerative cluster densities from UMAP



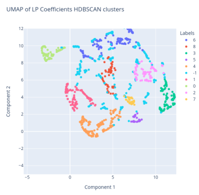
(f) Original cluster from UMAP



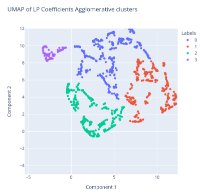
(g) K-means clusters from UMAP



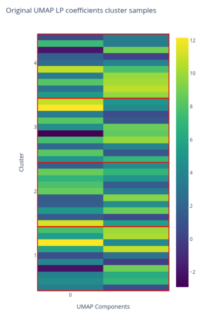
(h) DBSCAN cluster clusters from UMAP



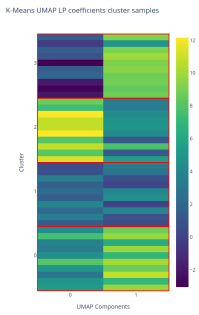
(i) HDBSCAN clusters from UMAP



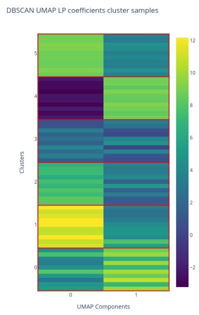
(j) Agglomerative clusters from UMAP



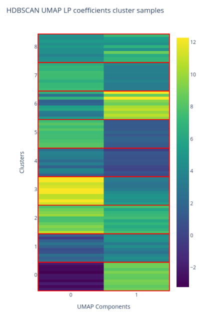
(k) Original cluster samples from UMAP



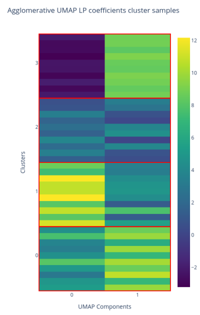
(l) K-means cluster samples from UMAP



(m) DBSCAN cluster samples-from UMAP



(n) HDBSCAN cluster samples from UMAP



(o) Agglomerative cluster samples from UMAP

Figure 120: Comparison between clustering UMAP LP coefficients algorithms

	Original	K-Means	DBSCAN	HDBSCAN	Agglomerative
Original	\	0.003	0.001	0.009	0.001
K-Means		\	0.438	0.480	0.667
DBSCAN			\	0.487	0.529
HDBSCAN				\	0.001

Table 23: Normalized Mutual Information between UMAP LP coefficients clusters

3.3 Output fluxes clustering

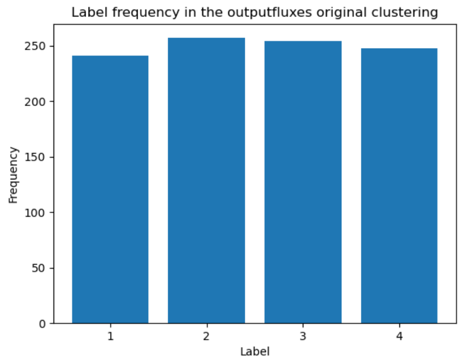
3.3.1 K-Means

As K-Means allows for the number of clusters to be defined, and we know that there are 4 in the original dataset, K-Means is used to find 4 clusters.

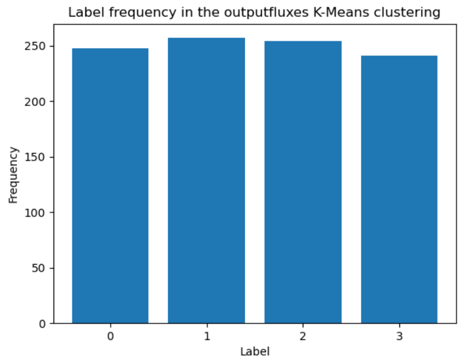
	Number of clusters	Number of initializations
Original Output fluxes	4	10
PCA Output fluxes	4	10
UMAP Output fluxes	4	10

Table 24: K-Means hyperparameter configuration for c coefficients clustering

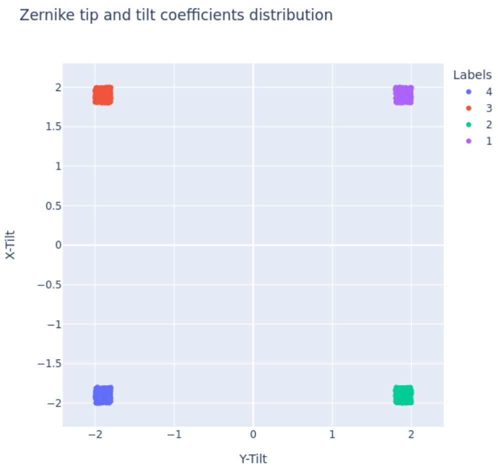
The results are the following:



(a) Original cluster densities

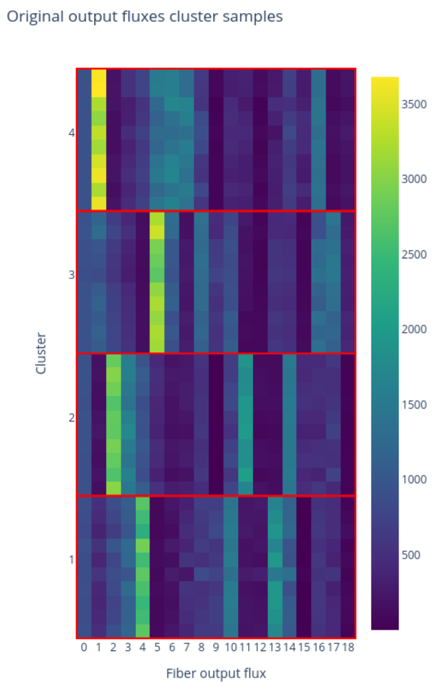


(b) K-Means clusters densities

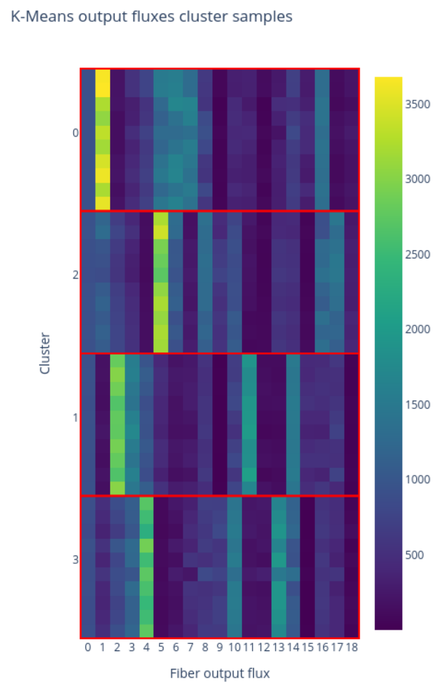


(c) Original clusters

(d) K-Means clusters

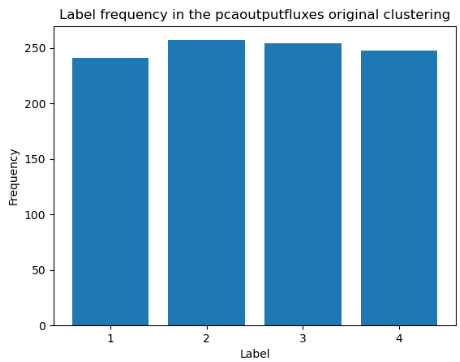


(e) Original cluster samples

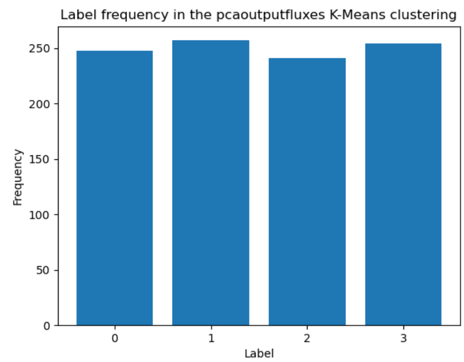


(f) K-Means cluster samples

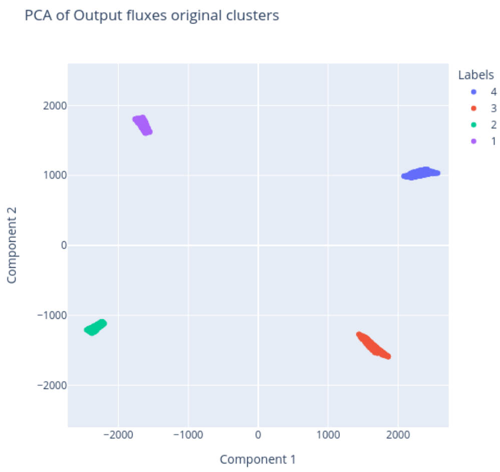
Figure 121: Comparison between original clustering and K-Means clustering from



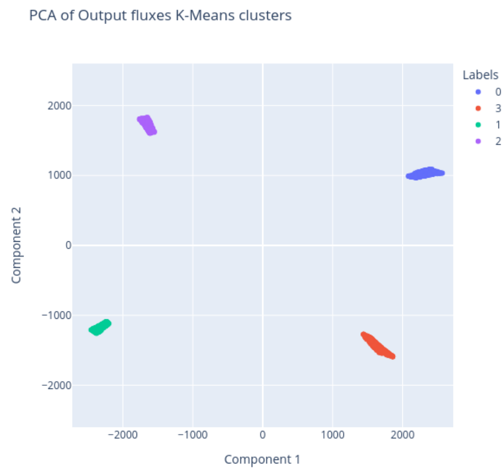
(a) Original cluster densities from PCA



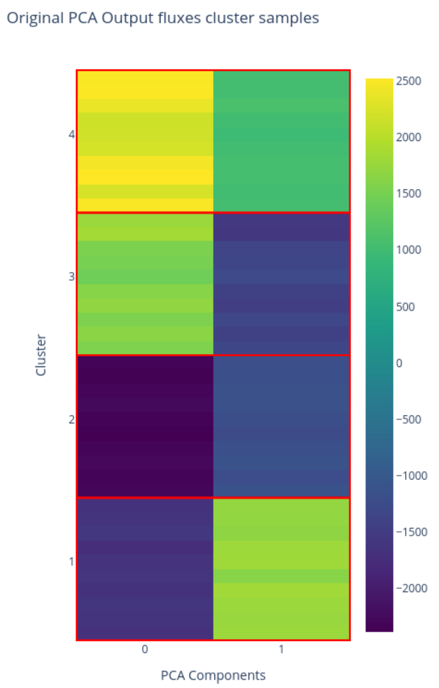
(b) K-Means clusters densities from PCA



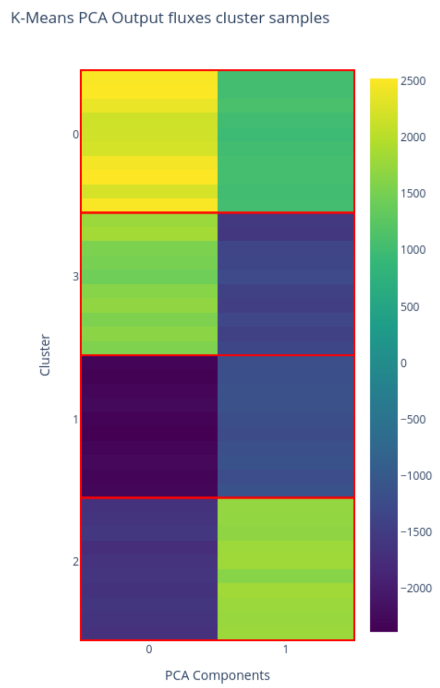
(c) Original clusters



(d) K-Means clusters from PCA

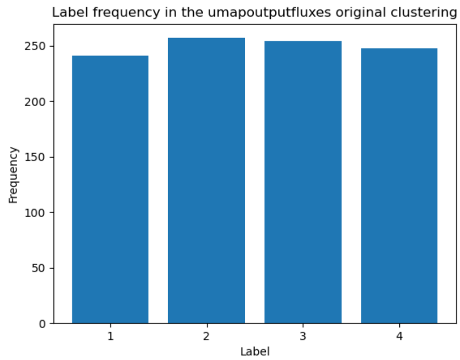


(e) Original cluster samples from PCA

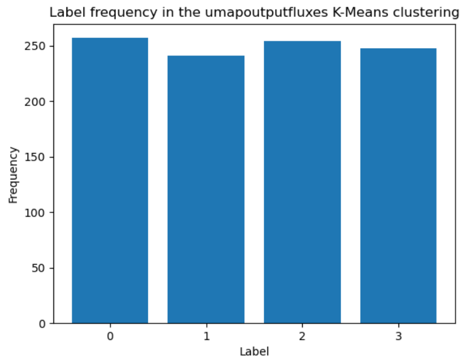


(f) K-Means cluster samples from PCA

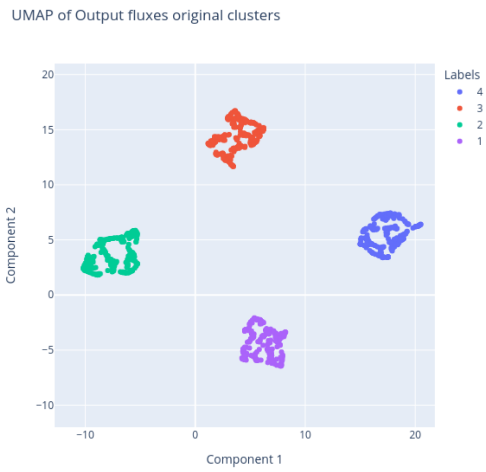
Figure 122: Comparison between original clustering and K-Means clustering from



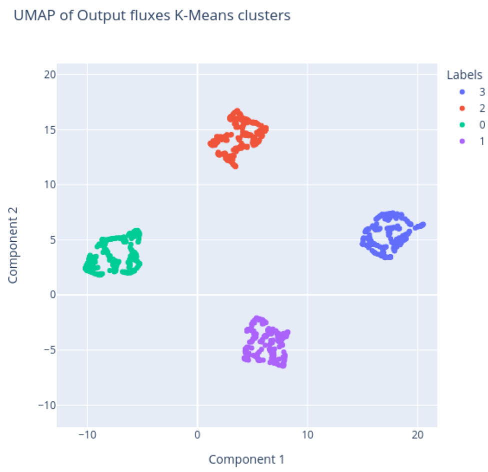
(a) Original cluster densities from UMAP



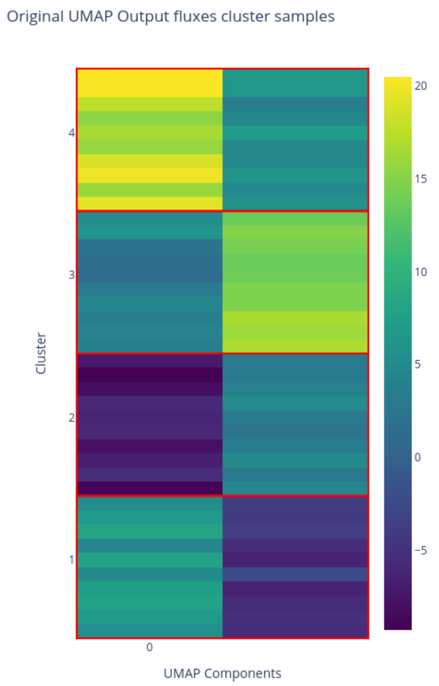
(b) K-Means clusters densities from UMAP



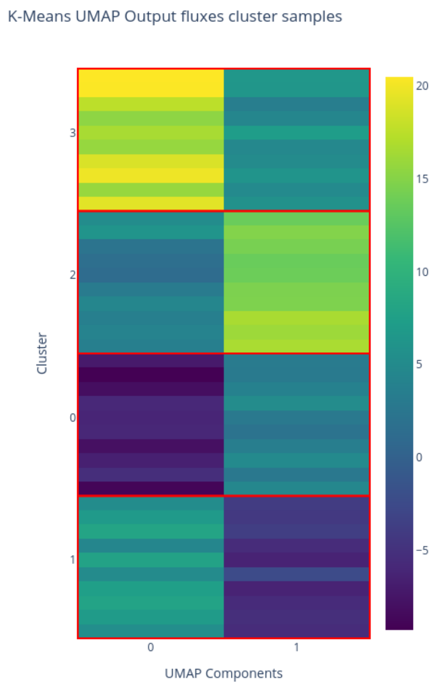
(c) Original clusters from UMAP



(d) K-Means clusters from UMAP



(e) Original cluster samples from UMAP



(f) K-Means cluster samples from UMAP

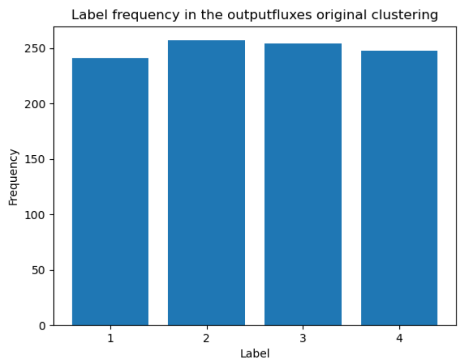
3.3.2 DBSCAN

A configuration that outputs 4 clusters is searched

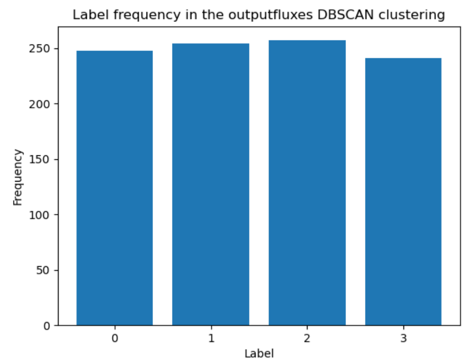
	Number of neighbours	**Epsilon**
Original Output fluxes	7	120
PCA Output fluxes	15	40
UMAP Output fluxes	10	0.85

Table 25: DBSCAN hyperparameter configuration for Output fluxes clustering

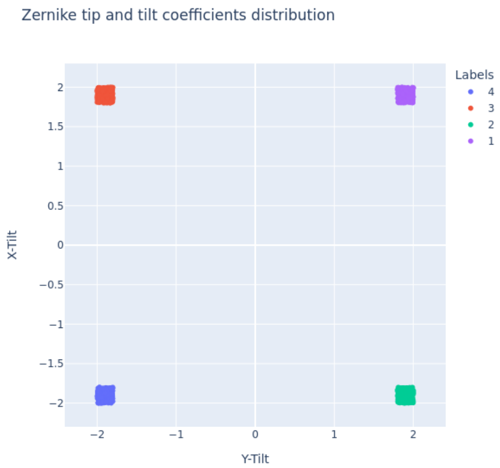
The results are the following:



(a) Original cluster densities

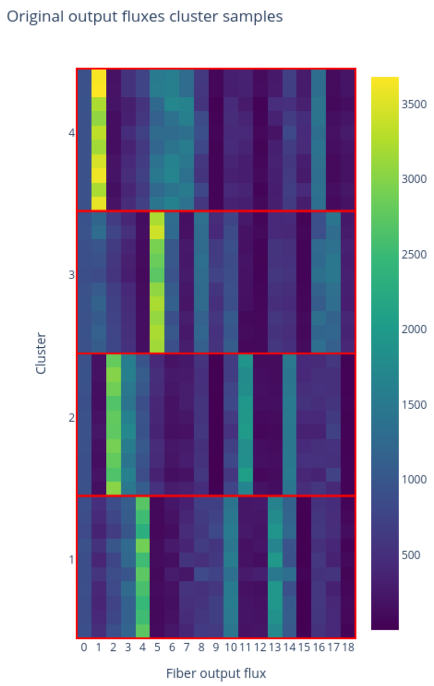


(b) DBSCAN clusters densities

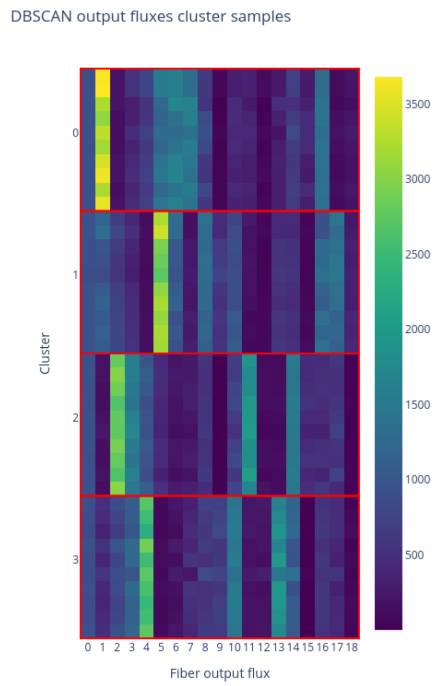


(c) Original clusters

(d) DBSCAN clusters

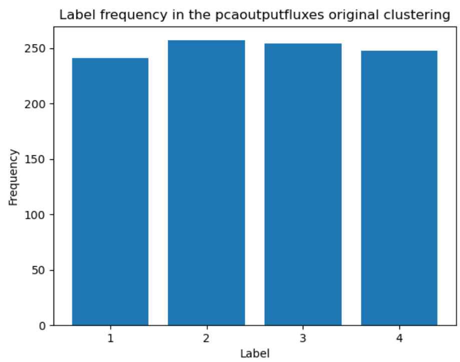


(e) Original cluster samples

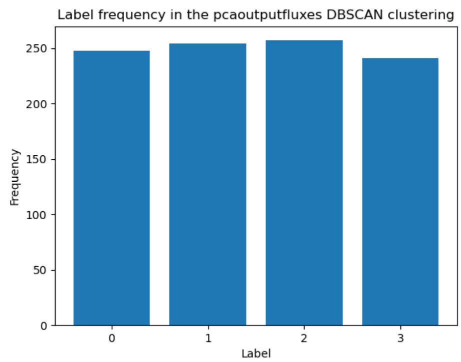


(f) DBSCAN cluster samples

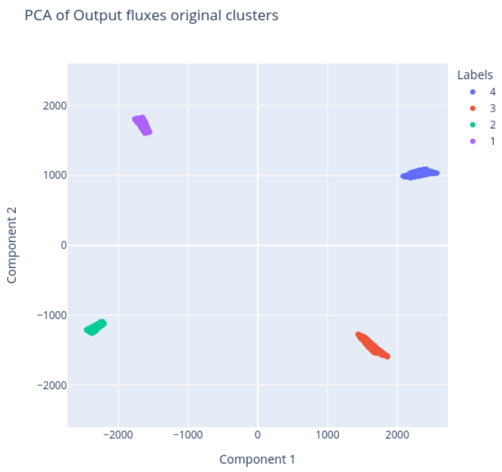
Figure 124: Comparison between original clustering and DBSCAN clustering



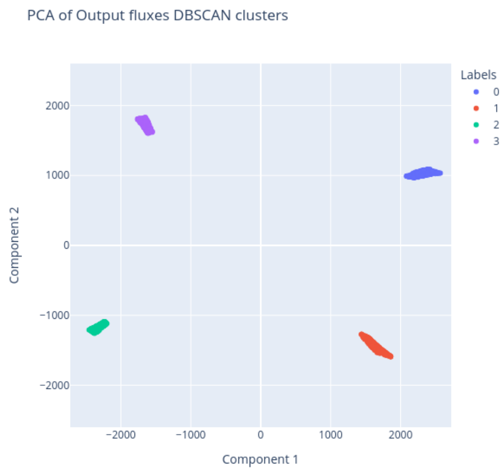
(a) Original cluster densities from PCA



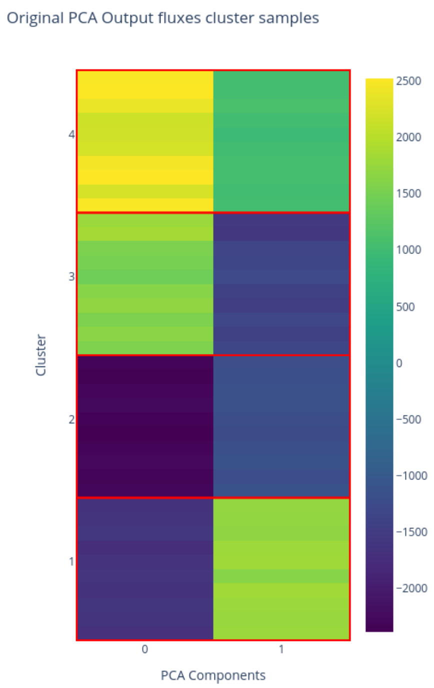
(b) DBSCAN clusters densities from PCA



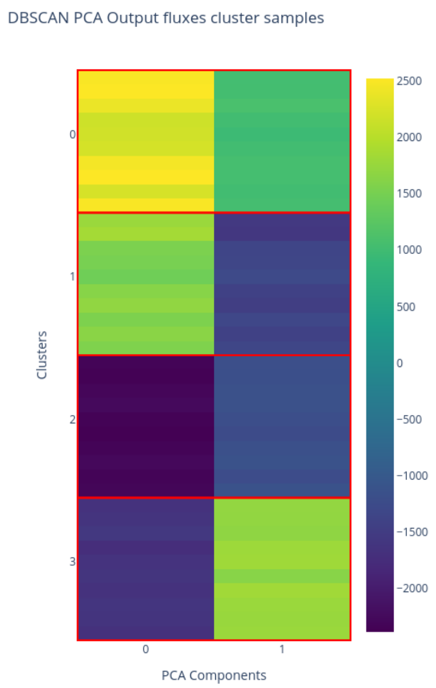
(c) Original clusters from PCA



(d) DBSCAN clusters from PCA

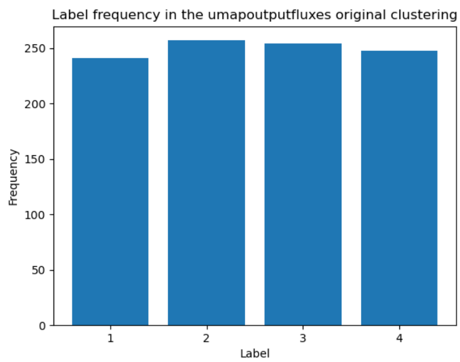


(e) Original cluster samples from PCA

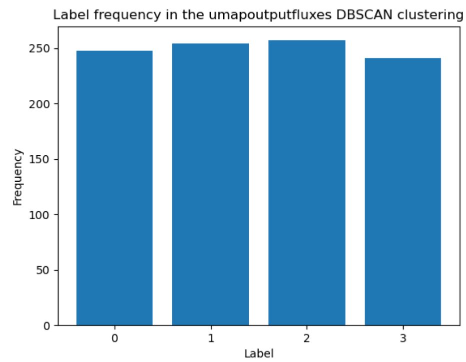


(f) DBSCAN cluster samples from PCA

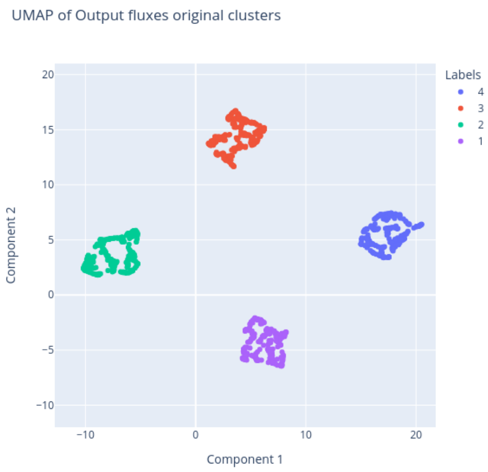
Figure 125: Comparison between original clustering and DBSCAN clustering from



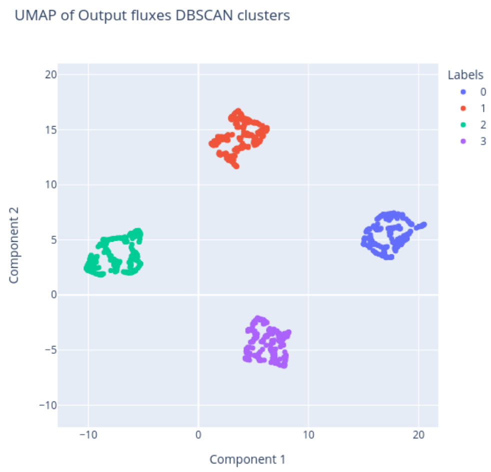
(a) Original cluster densities from UMAP



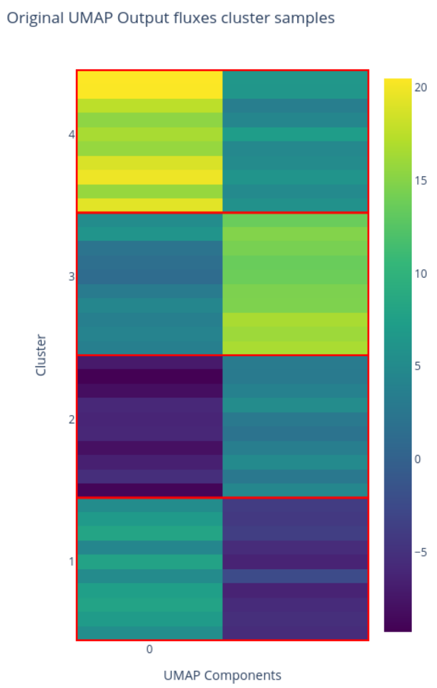
(b) DBSCAN clusters densities from UMAP



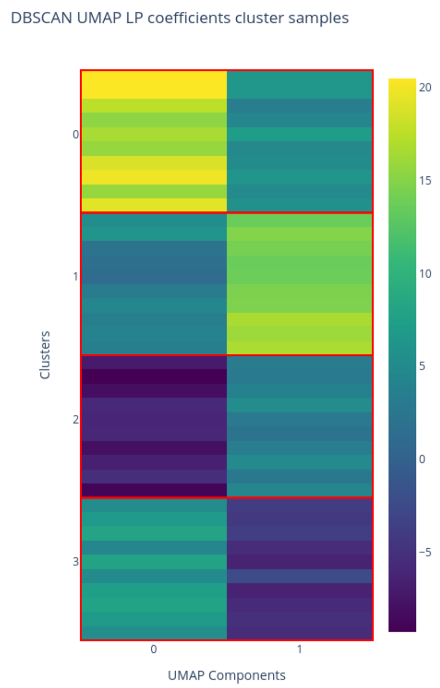
(c) Original clusters from UMAP



(d) DBSCAN clusters from UMAP



(e) Original cluster samples from UMAP



(f) DBSCAN cluster samples from UMAP

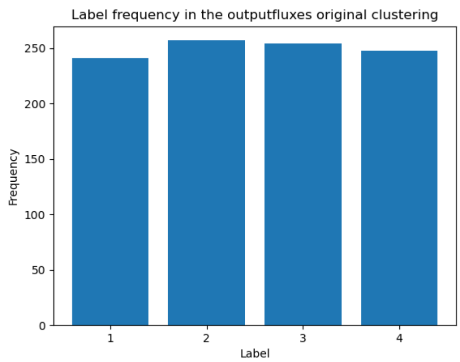
3.3.3 HDBSCAN

A configuration that outputs 4 clusters is searched.

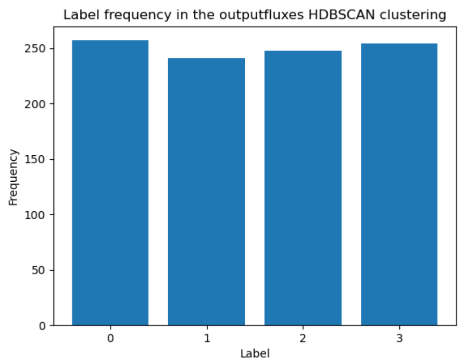
	Minimum cluster size
Original Output fluxes	21
PCA Output fluxes	21
HDBSCAN Output fluxes	25

Table 26: HDBSCAN hyperparameter configuration for Output fluxes clustering

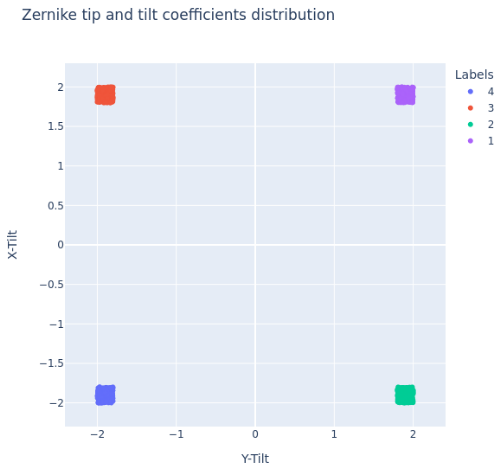
The results are the following:



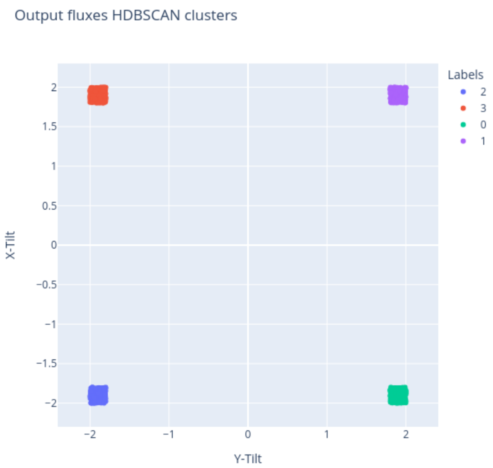
(a) Original cluster densities



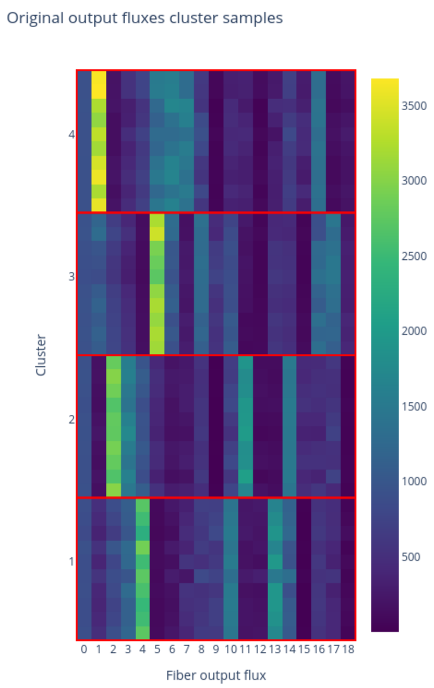
(b) HDBSCAN clusters densities



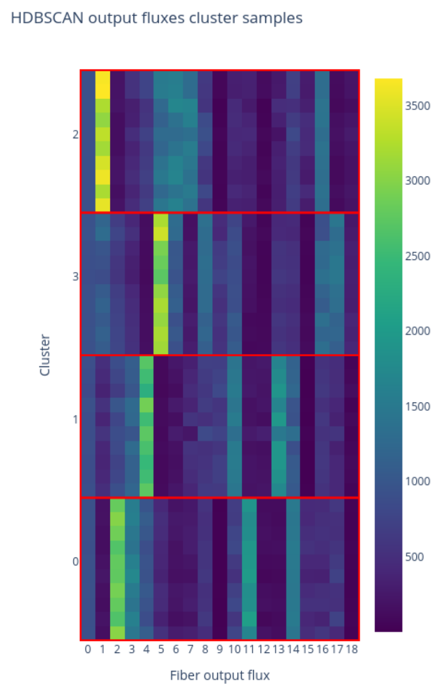
(c) Original clusters



(d) HDBSCAN clusters

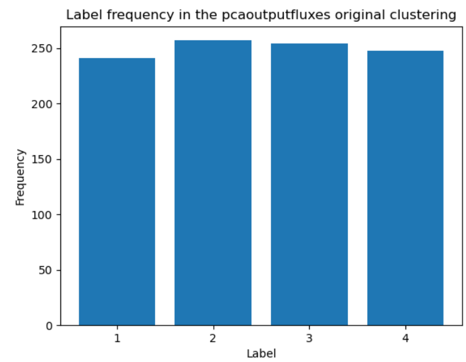


(e) Original cluster samples

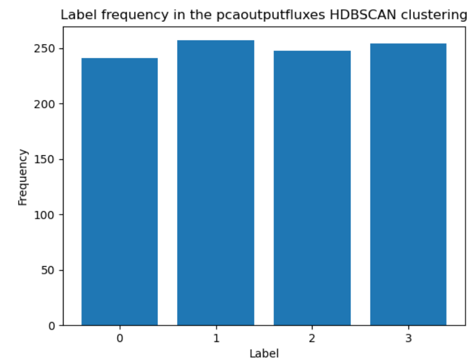


(f) HDBSCAN cluster samples

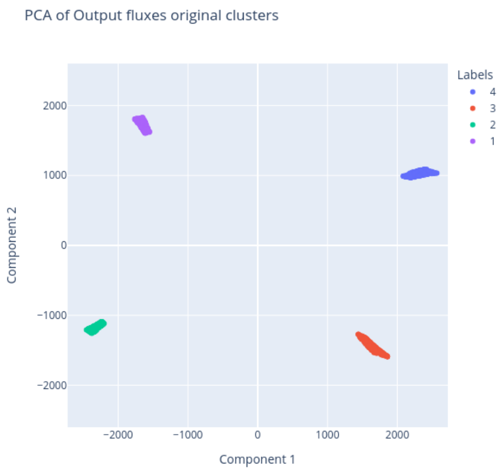
Figure 127: Comparison between original clustering and HDBSCAN clustering



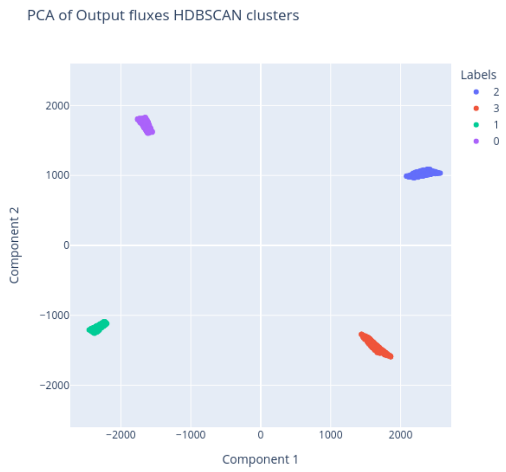
(a) Original cluster densities from PCA



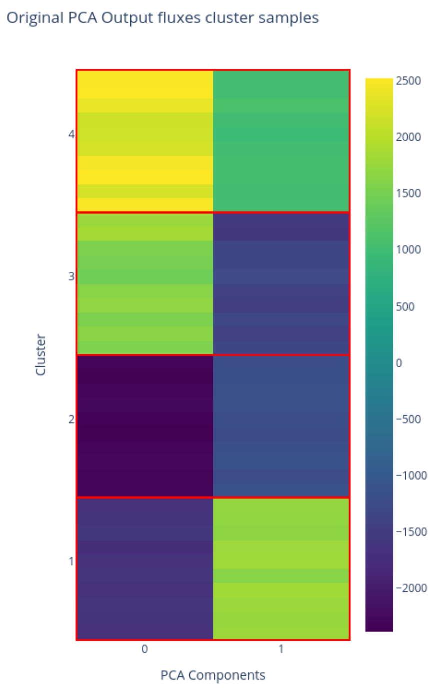
(b) HDBSCAN clusters densities from PCA



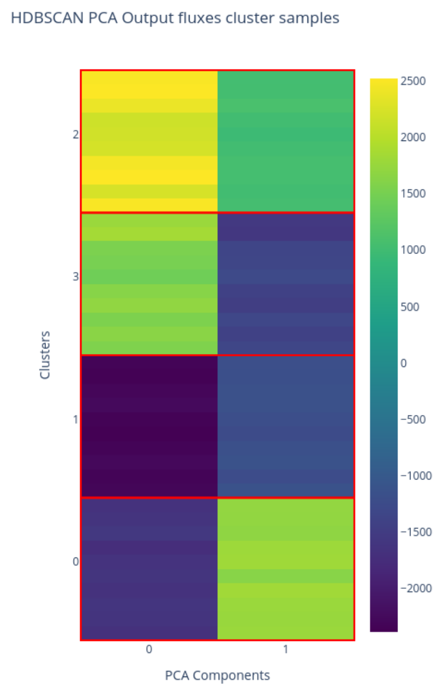
(c) Original clusters from PCA



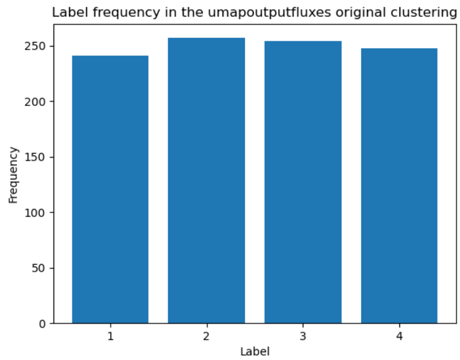
(d) HDBSCAN clusters from PCA



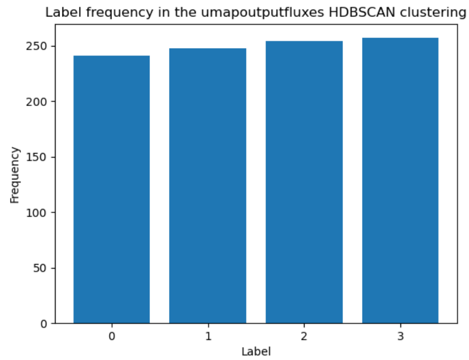
(e) Original cluster samples from PCA



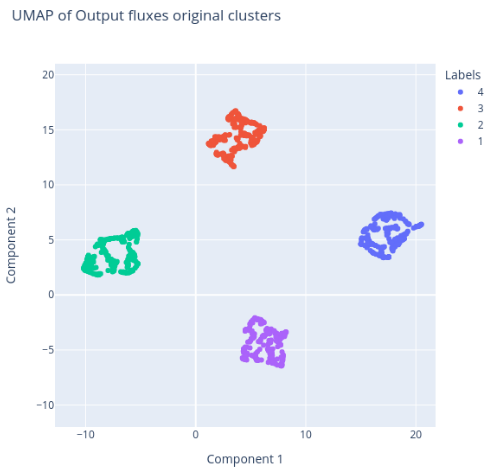
(f) HDBSCAN cluster samples from PCA



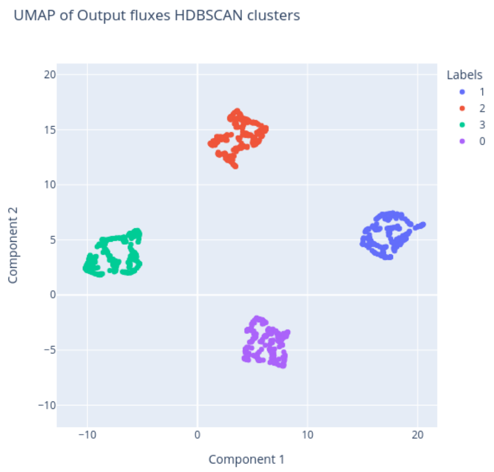
(a) Original cluster densities from UMAP



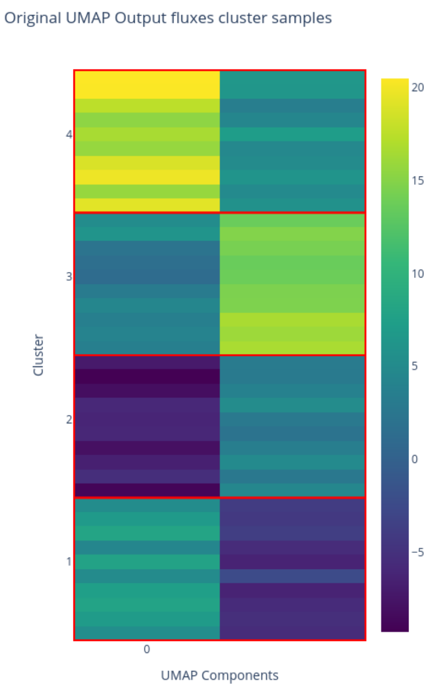
(b) HDBSCAN clusters densities from UMAP



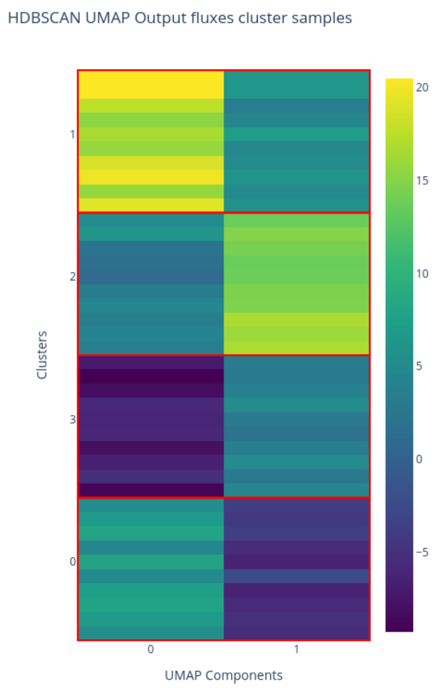
(c) Original clusters from UMAP



(d) HDBSCAN clusters from UMAP



(e) Original cluster samples from UMAP



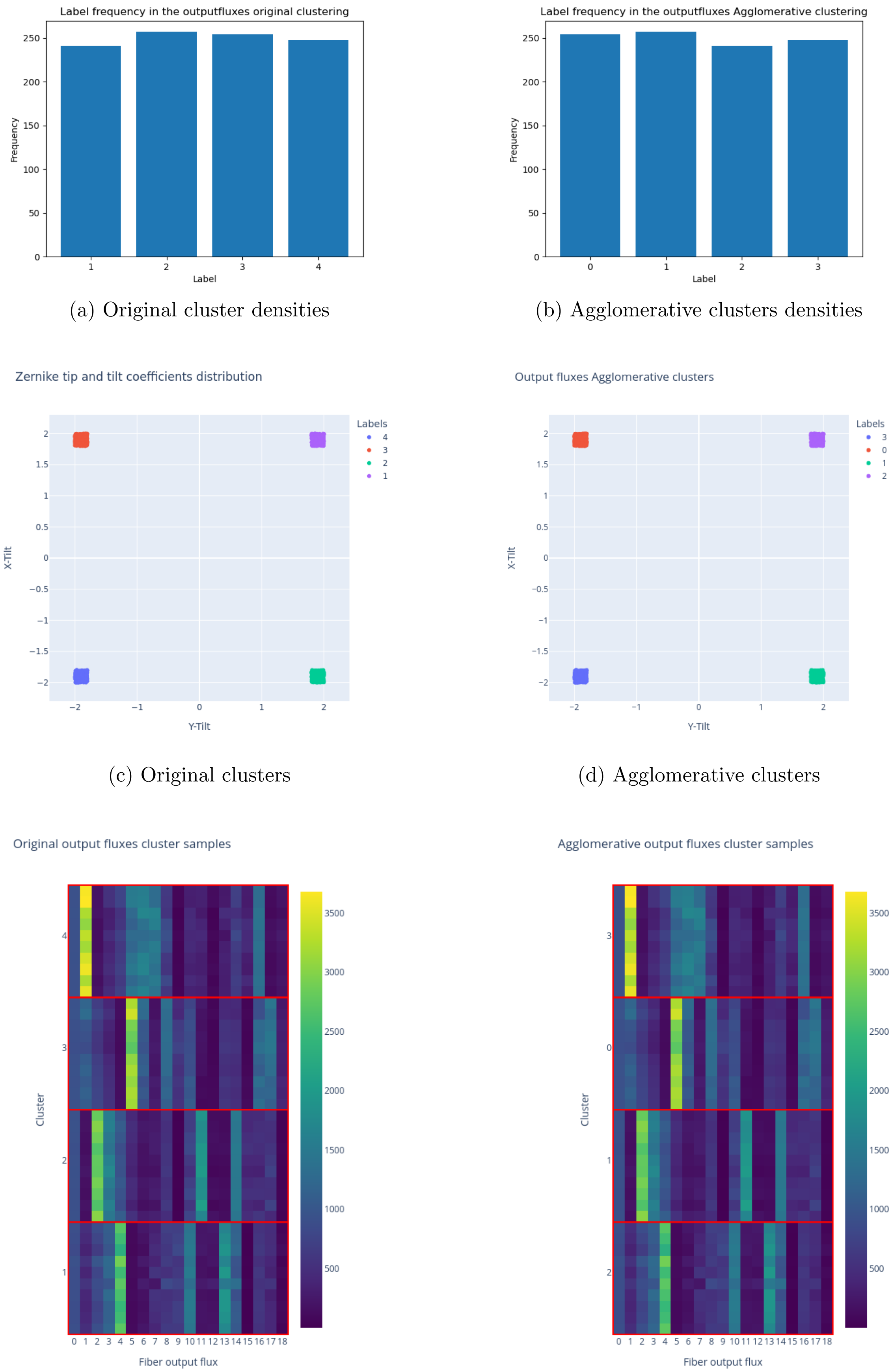
(f) HDBSCAN cluster samples from UMAP

3.3.4 Agglomerative clustering

	Number of clusters
Original Output fluxes	4
PCA Output fluxes	4
UMAP Output fluxes	4

Table 27: Agglomerative hyperparameter configuration for Output fluxes clustering

The results are the following:



(a) Original cluster densities

(b) Agglomerative clusters densities

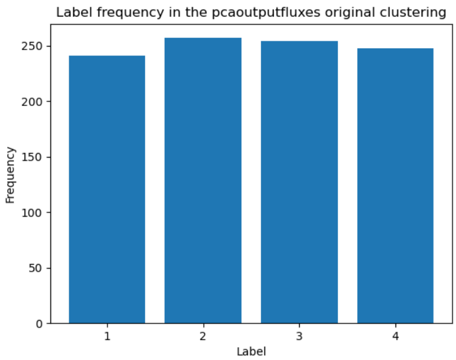
(c) Original clusters

(d) Agglomerative clusters

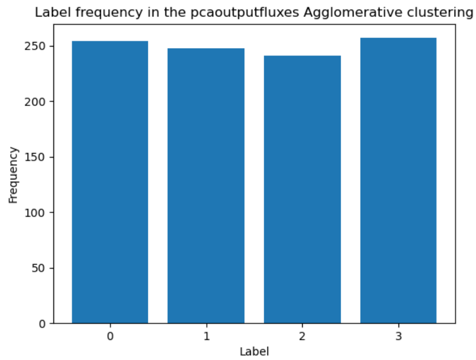
(e) Original cluster samples

(f) Agglomerative cluster samples

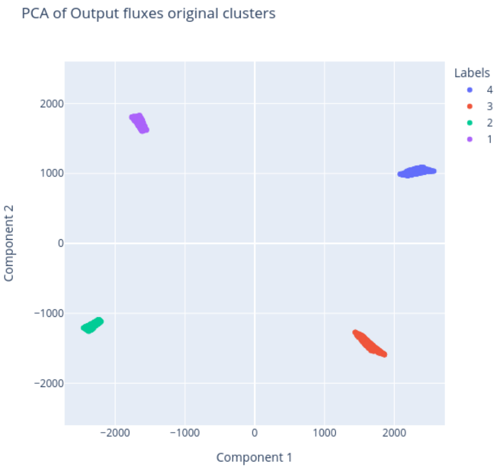
Figure 130: Comparison between original clustering and Agglomerative clustering



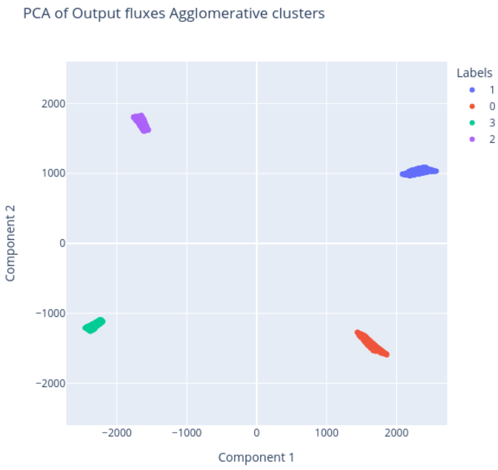
(a) Original cluster densities from PCA



(b) Agglomerative clusters densities from PCA

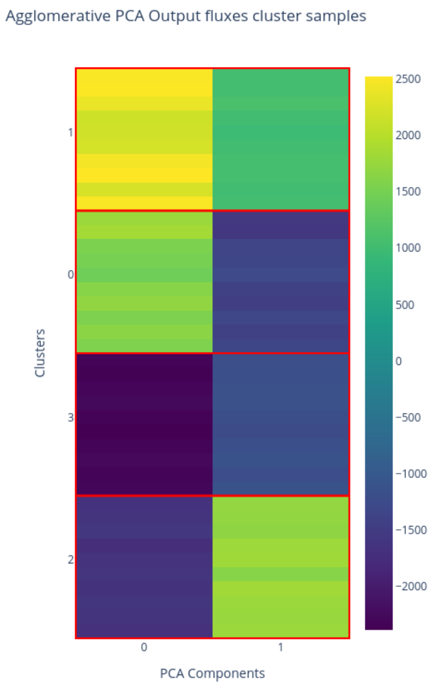


(c) Original clusters from PCA

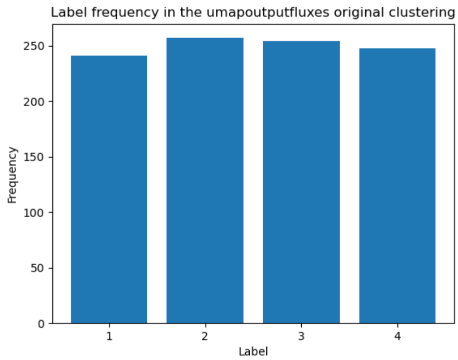


(d) Agglomerative clusters from PCA

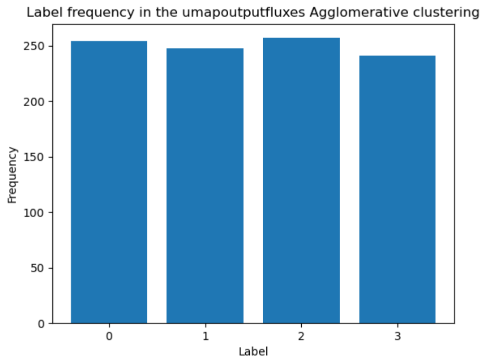
(e) Original cluster samples from PCA



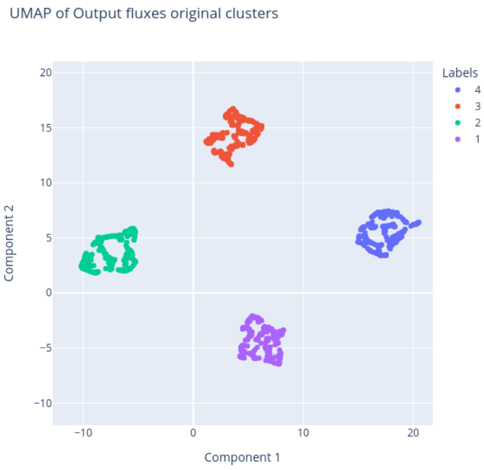
(f) Agglomerative cluster samples from PCA



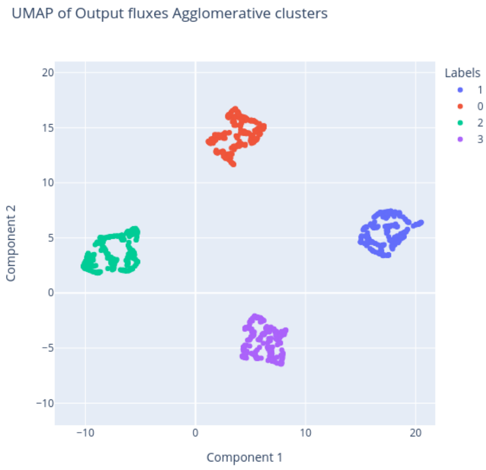
(a) Original cluster densities from UMAP



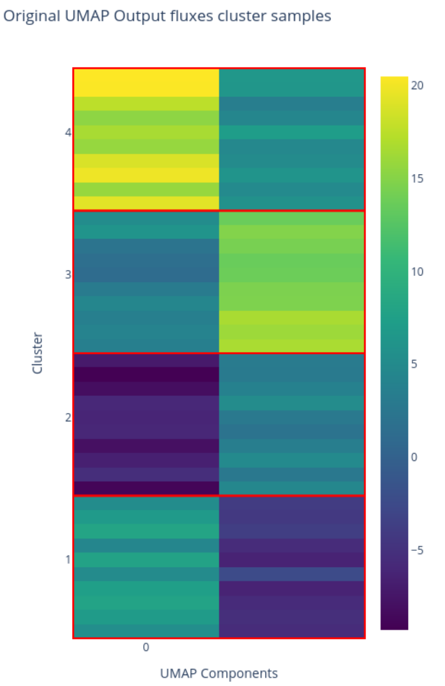
(b) Agglomerative clusters densities from UMAP



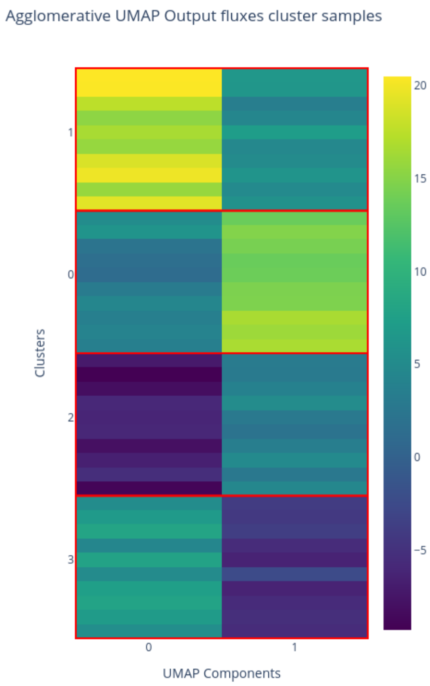
(c) Original clusters from UMAP



(d) Agglomerative clusters from UMAP

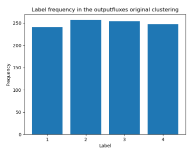


(e) Original cluster samples from UMAP

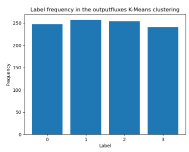


(f) Agglomerative cluster samples from UMAP

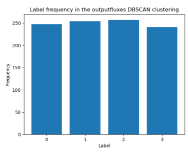
3.3.5 Summary



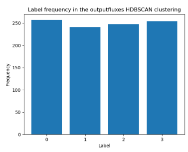
(a) Original cluster densities



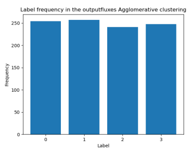
(b) K-means cluster densities



(c) DBSCAN cluster densities



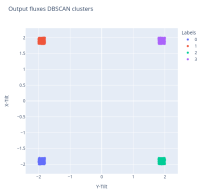
(d) HDBSCAN cluster densities



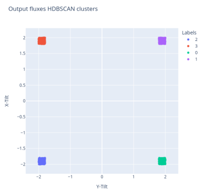
(e) Agglomerative cluster densities

(f) Original cluster

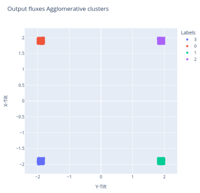
(g) K-means clusters



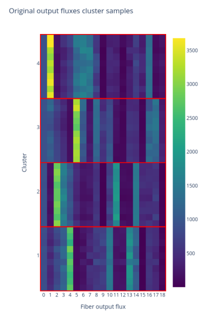
(h) DBSCAN cluster clusters



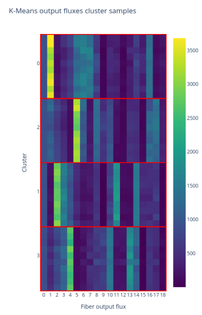
(i) HDBSCAN clusters



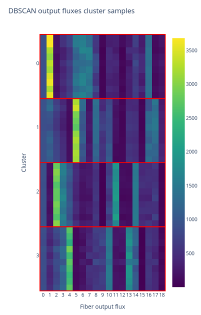
(j) Agglomerative clusters



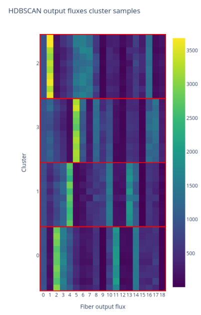
(k) Original cluster samples



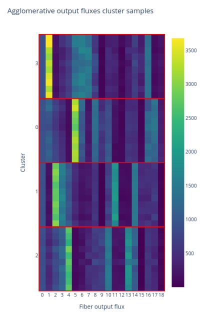
(l) K-means cluster samples



(m) DBSCAN cluster samples



(n) HDBSCAN cluster samples

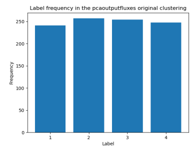


(o) Agglomerative cluster samples

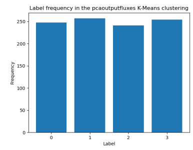
Figure 133: Comparison between clustering Output fluxes algorithms

	Original	**K-Means**	**DBSCAN**	**HDBSCAN**	**Agglomerative**
Original		1	1	1	1
K-Means			1	1	1
DBSCAN				1	1
HDBSCAN					1

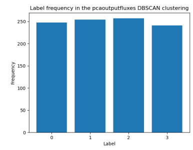
Table 28: Normalized Mutual Information between original Output fluxes clusters



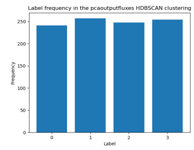
(a) Original cluster densities from PCA



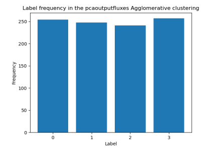
(b) K-means cluster densities from PCA



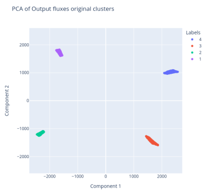
(c) DBSCAN cluster densities from PCA



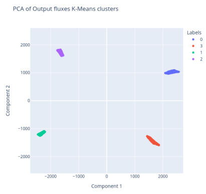
(d) HDBSCAN cluster densities from PCA



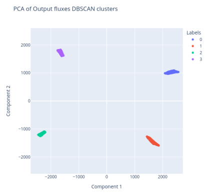
(e) Agglomerative cluster densities from PCA



(f) Original cluster from PCA



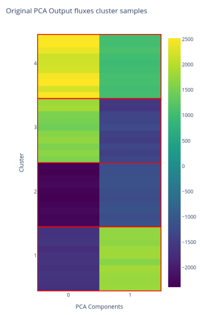
(g) K-means clusters from PCA



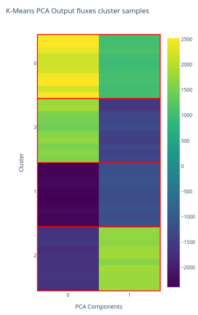
(h) DBSCAN cluster clusters from PCA

(i) HDBSCAN clusters from PCA

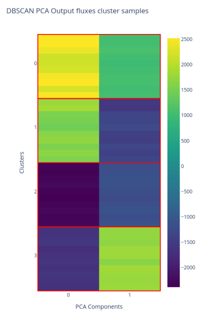
(j) Agglomerative clusters from PCA



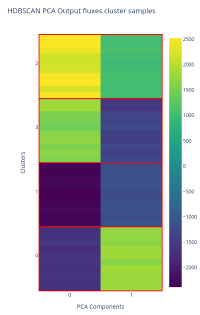
(k) Original cluster samples from PCA



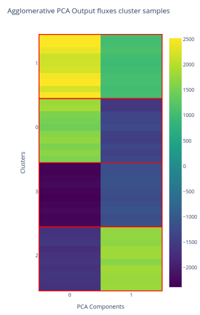
(l) K-means cluster samples from PCA



(m) DBSCAN cluster samples from PCA



(n) HDBSCAN cluster samples from PCA

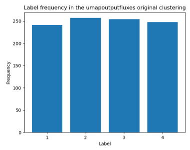


(o) Agglomerative cluster samples from PCA

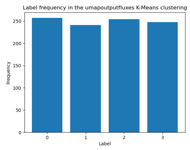
Figure 134: Comparison between clustering PCA Output fluxes algorithms

	Original	**K-Means**	**DBSCAN**	**HDBSCAN**	**Agglomerative**
Original		1	1	1	1
K-Means			1	1	1
DBSCAN				1	1
HDBSCAN					1

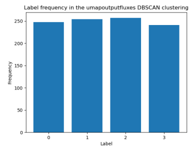
Table 29: Normalized Mutual Information between PCA Output fluxes clusters



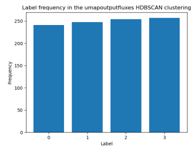
(a) Original cluster densities from UMAP



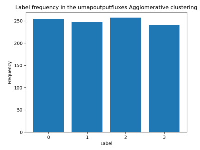
(b) K-means cluster densities from UMAP



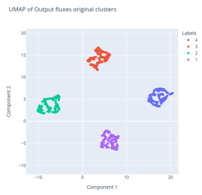
(c) DBSCAN cluster densities from UMAP



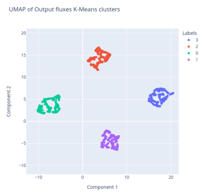
(d) HDBSCAN cluster densities from UMAP



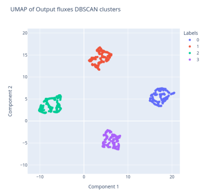
(e) Agglomerative cluster densities from UMAP



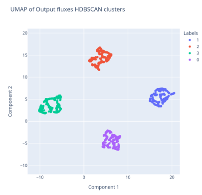
(f) Original cluster from UMAP



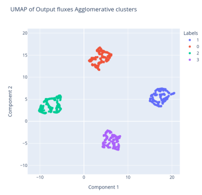
(g) K-means clusters from UMAP



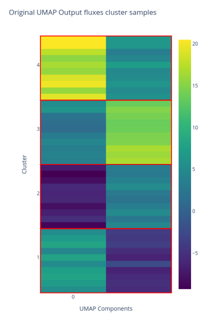
(h) DBSCAN cluster clusters from UMAP



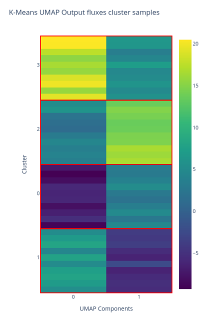
(i) HDBSCAN clusters from UMAP



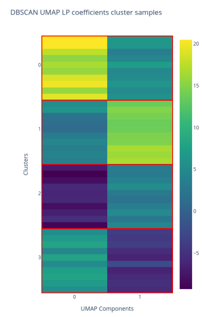
(j) Agglomerative clusters from UMAP



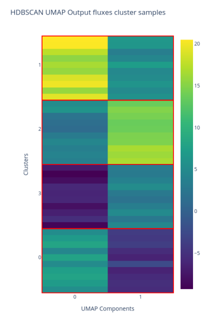
(k) Original cluster samples from UMAP



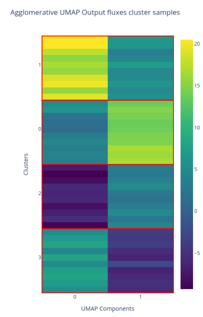
(l) K-means cluster samples from UMAP



(m) DBSCAN cluster samples-from UMAP



(n) HDBSCAN cluster samples from UMAP



(o) Agglomerative cluster samples from UMAP

Figure 135: Comparison between clustering UMAP Output fluxes algorithms

	Original	**K-Means**	**DBSCAN**	**HDBSCAN**	**Agglomerative**
Original	$\diagdown$	1	1	1	1
K-Means		$\diagdown$	1	1	1
DBSCAN			$\diagdown$	1	1
HDBSCAN				$\diagdown$	1

Table 30: Normalized Mutual Information between UMAP Output fluxes clusters

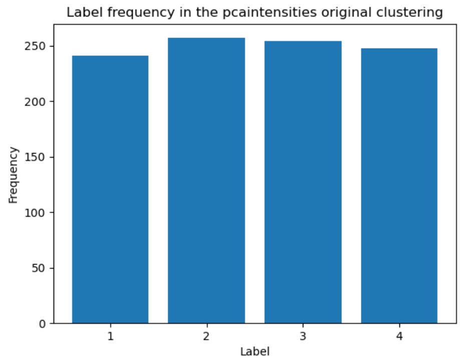
3.4 PSF Intensities clustering

3.4.1 K-Means

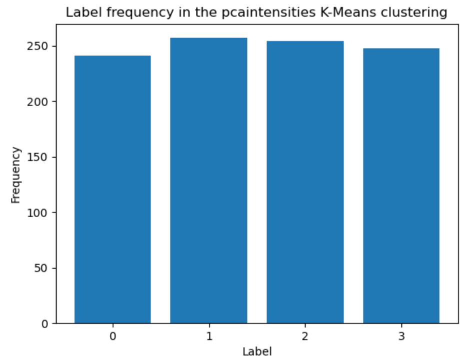
As K-Means allows for the number of clusters to be defined, and we know that there are 4 in the original dataset, K-Means is used to find 4 clusters.

	Number of clusters	**Number of initializations**
PCA PSF Intensities	4	10
UMAP PSF Intensities	4	10

Table 31: K-Means hyperparameter configuration for c coefficients clustering

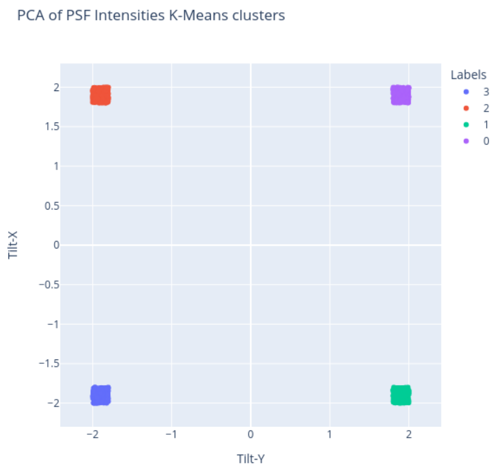


(a) Original cluster densities from PCA

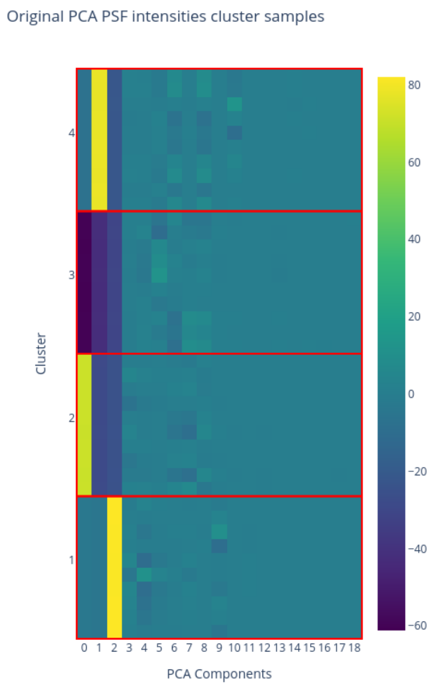


(b) K-Means clusters densities from PCA

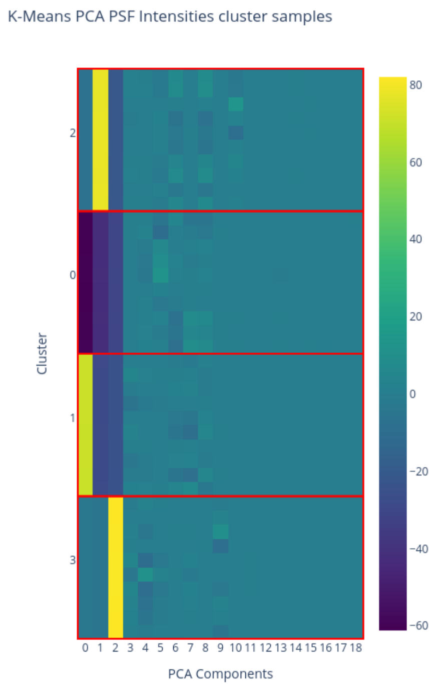
(c) Original clusters



(d) K-Means clusters from PCA

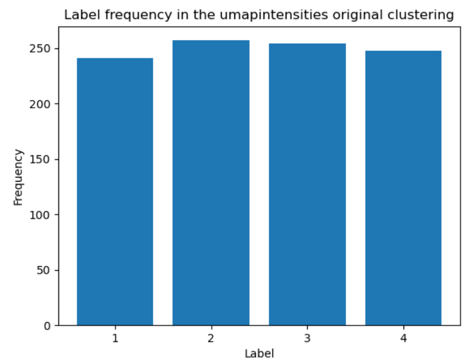


(e) Original cluster samples from PCA

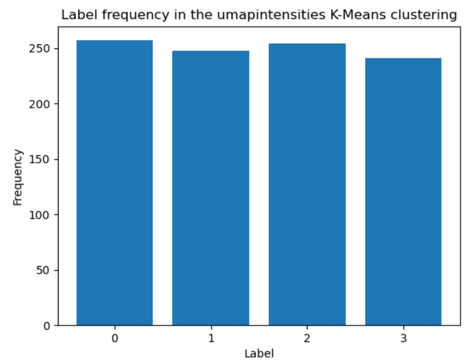


(f) K-Means cluster samples from PCA

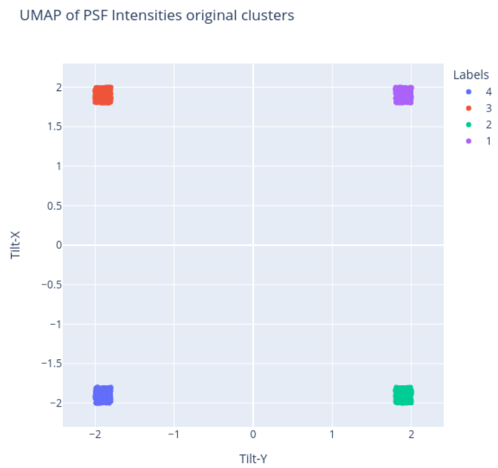
Figure 136: Comparison between original clustering and K-Means clustering from



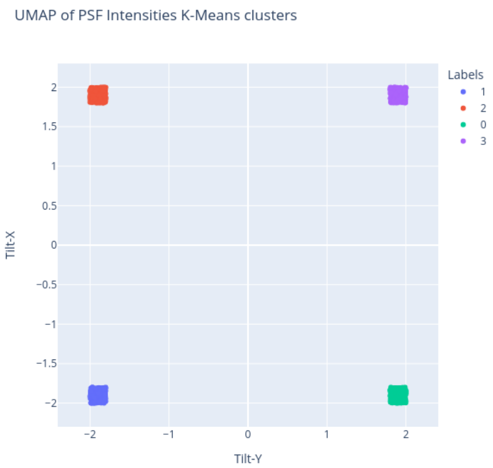
(a) Original cluster densities from UMAP



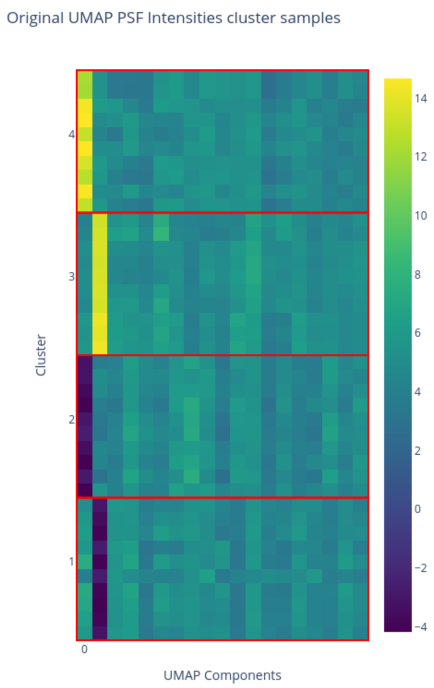
(b) K-Means clusters densities from UMAP



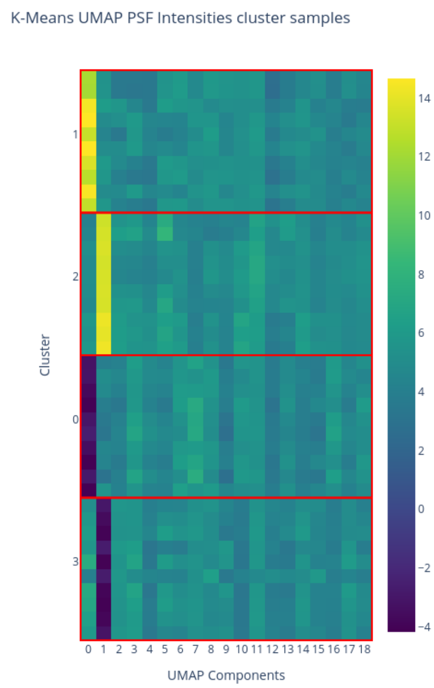
(c) Original clusters from UMAP



(d) K-Means clusters from UMAP



(e) Original cluster samples from UMAP



(f) K-Means cluster samples from UMAP

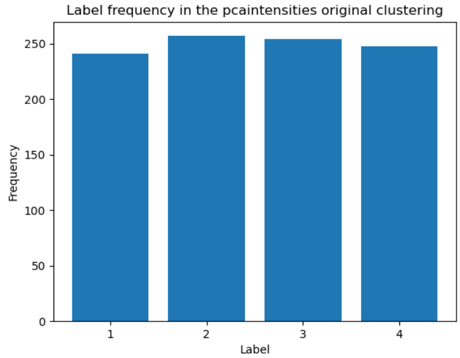
3.4.2 DBSCAN

A configuration that outputs 4 clusters is searched

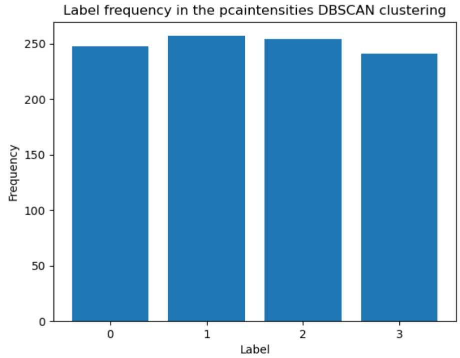
	Number of neighbours	**Epsilon**
PCA PSF Intensities	15	4.5
UMAP PSF Intensities	10	0.85

Table 32: DBSCAN hyperparameter configuration for PSF Intensities clustering

The results are the following:

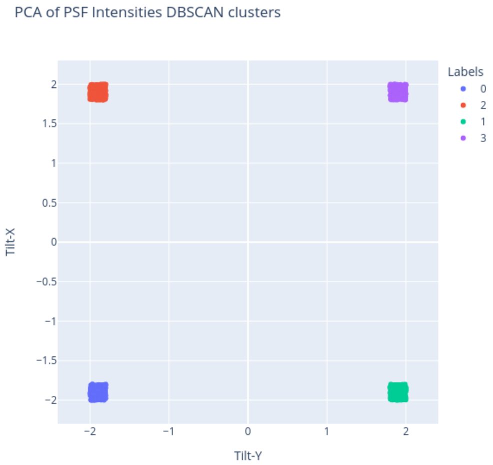


(a) Original cluster densities from PCA

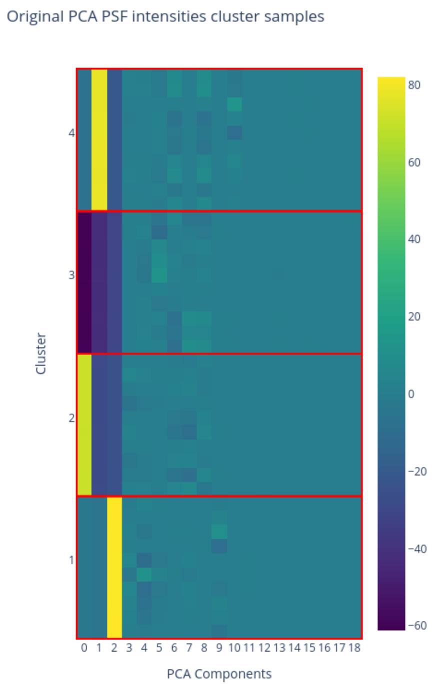


(b) DBSCAN clusters densities from PCA

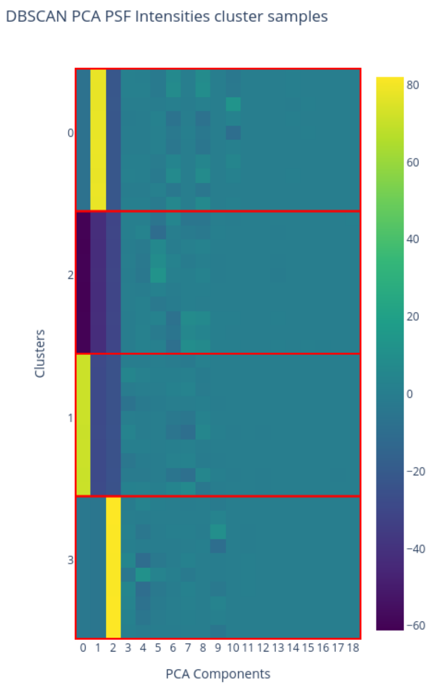
(c) Original clusters from PCA



(d) DBSCAN clusters from PCA

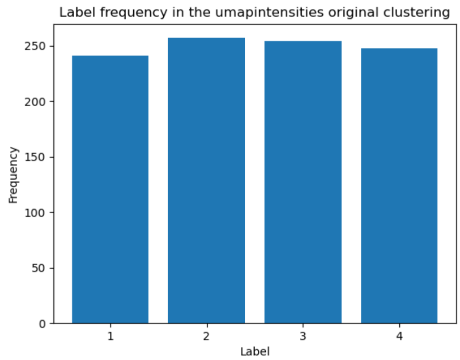


(e) Original cluster samples from PCA

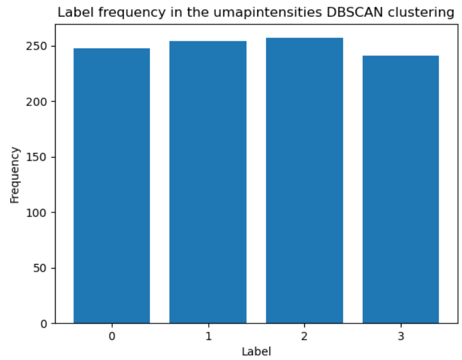


(f) DBSCAN cluster samples from PCA

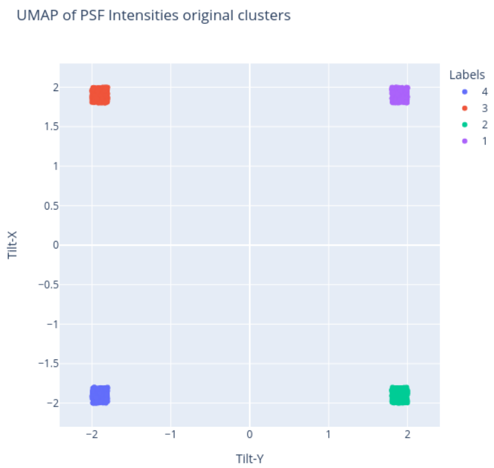
Figure 138: Comparison between original clustering and DBSCAN clustering from



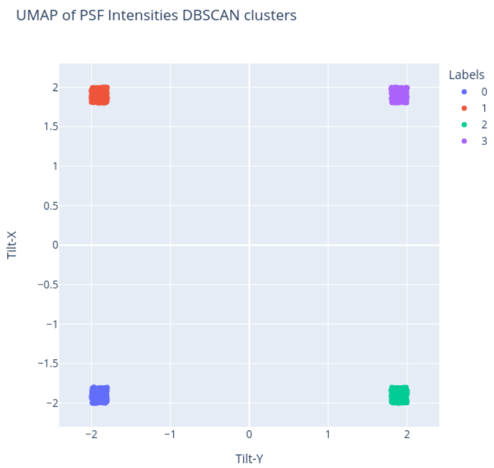
(a) Original cluster densities from UMAP



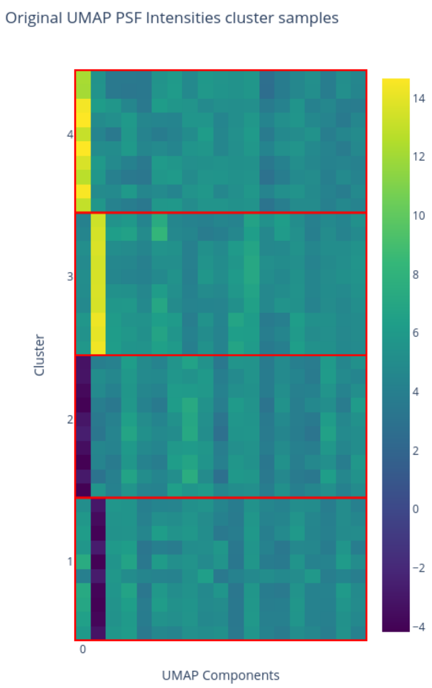
(b) DBSCAN clusters densities from UMAP



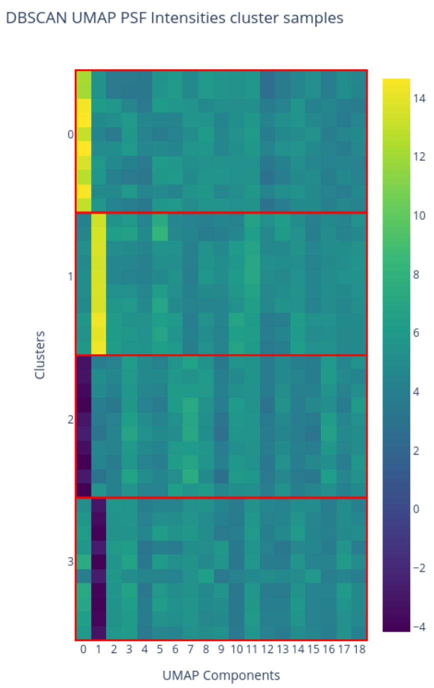
(c) Original clusters from UMAP



(d) DBSCAN clusters from UMAP



(e) Original cluster samples from UMAP



(f) DBSCAN cluster samples from UMAP

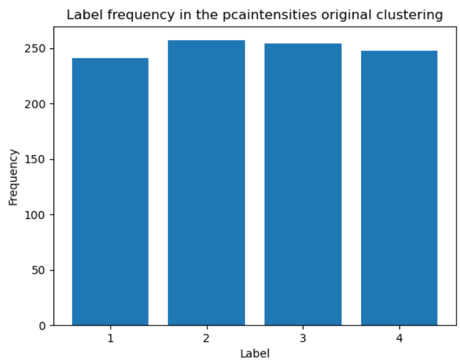
3.4.3 HDBSCAN

A configuration that outputs 4 clusters is searched.

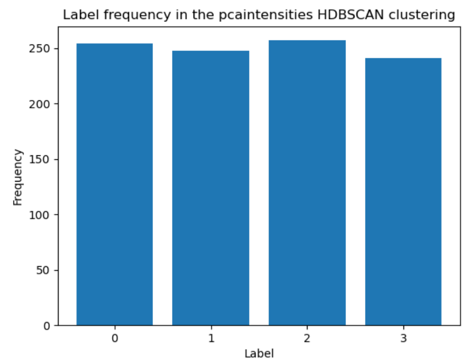
	Minimum cluster size
PCA PSF Intensities	21
UMAP PSF Intensities	25

Table 33: HDBSCAN hyperparameter configuration for PSF Intensities clustering

The results are the following:

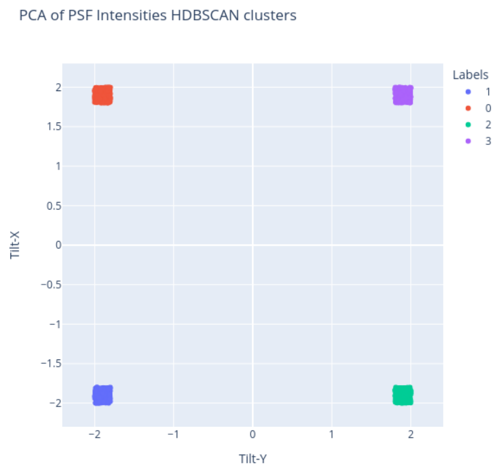


(a) Original cluster densities from PCA

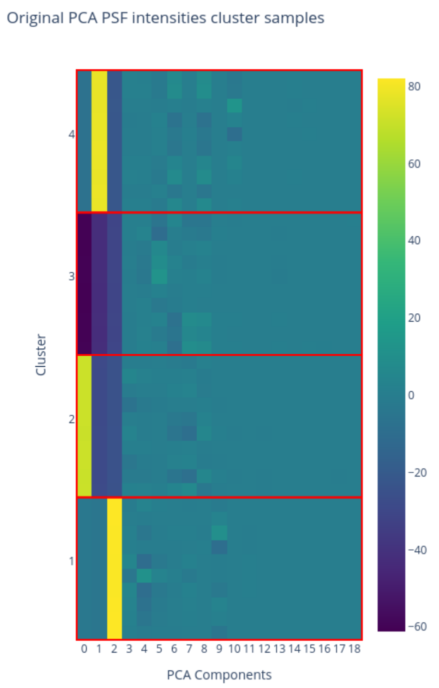


(b) HDBSCAN clusters densities from PCA

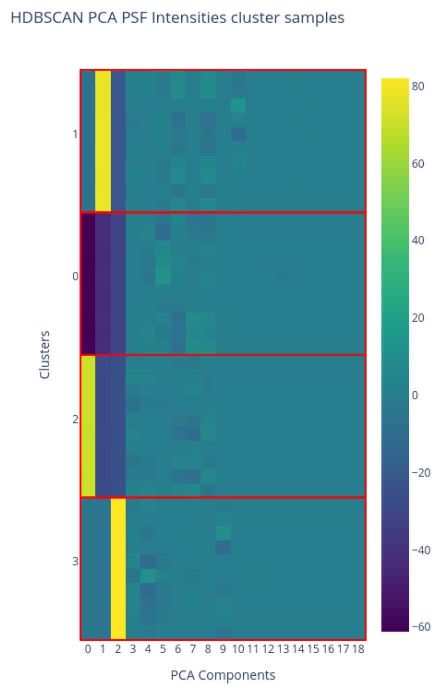
(c) Original clusters from PCA



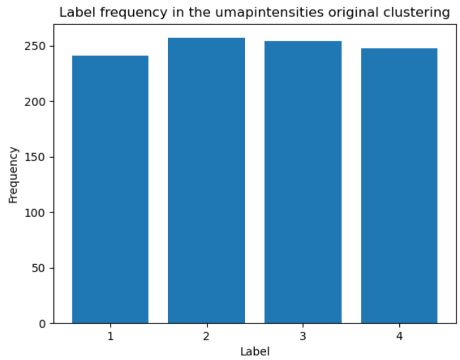
(d) HDBSCAN clusters from PCA



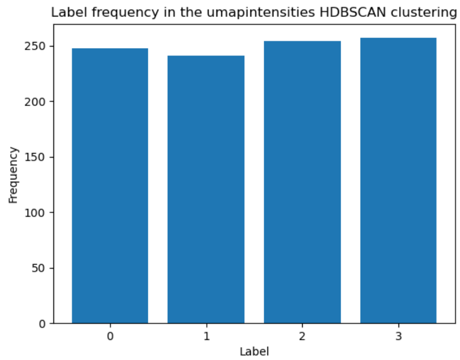
(e) Original cluster samples from PCA



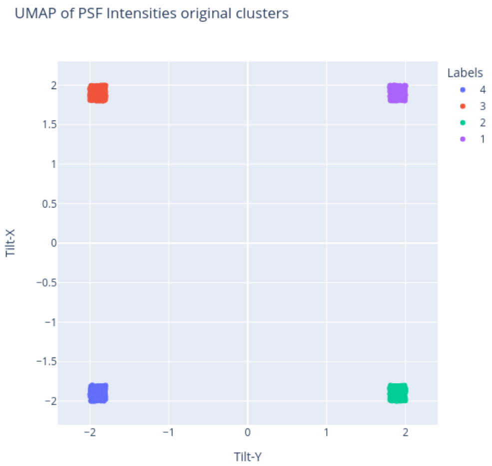
(f) HDBSCAN cluster samples from PCA



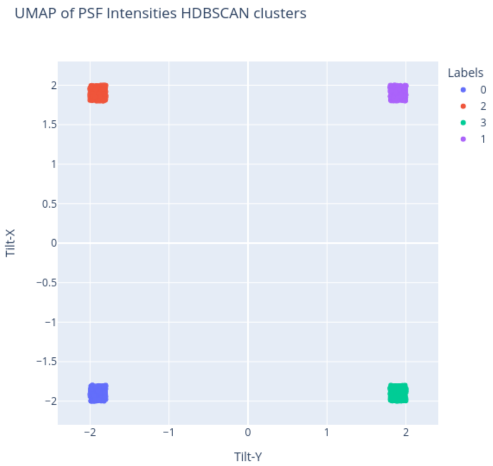
(a) Original cluster densities from UMAP



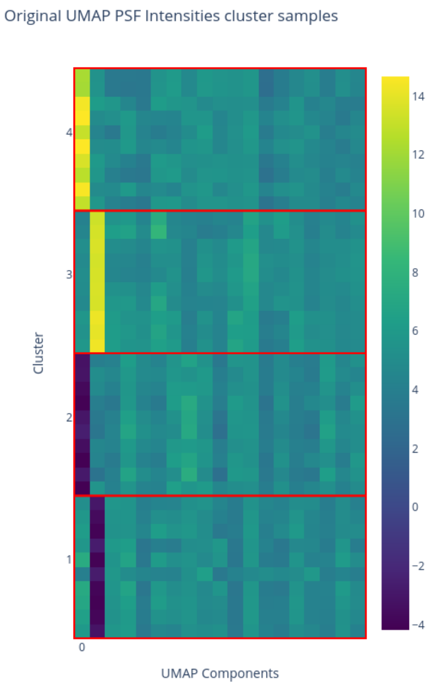
(b) HDBSCAN clusters densities from UMAP



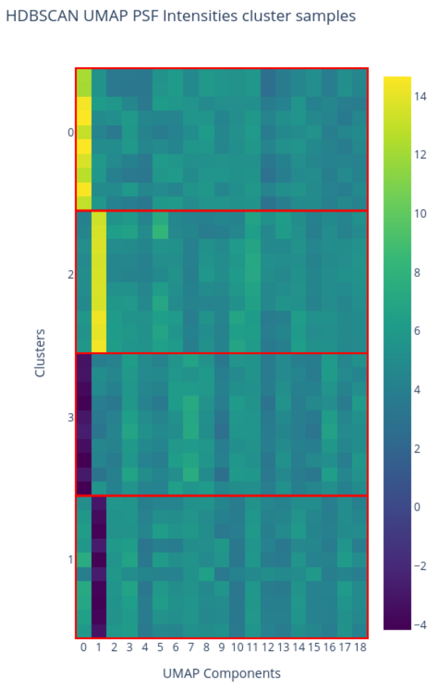
(c) Original clusters from UMAP



(d) HDBSCAN clusters from UMAP



(e) Original cluster samples from UMAP



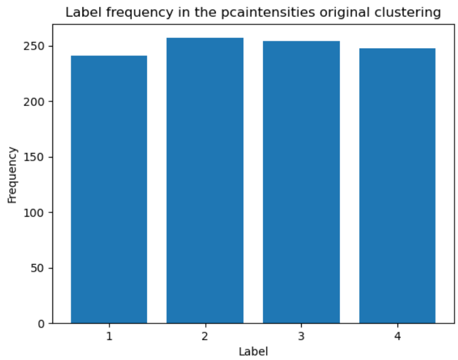
(f) HDBSCAN cluster samples from UMAP

3.4.4 Agglomerative clustering

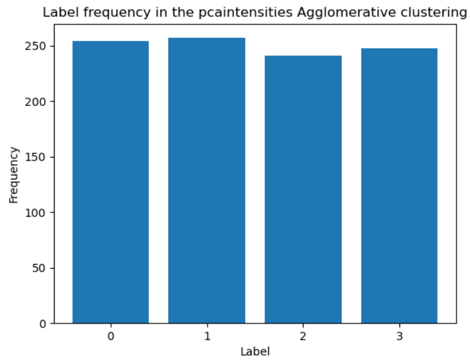
	Number of clusters
PCA PSF Intensities	4
UMAP PSF Intensities	4

Table 34: Agglomerative hyperparameter configuration for PSF Intensities clustering

The results are the following:



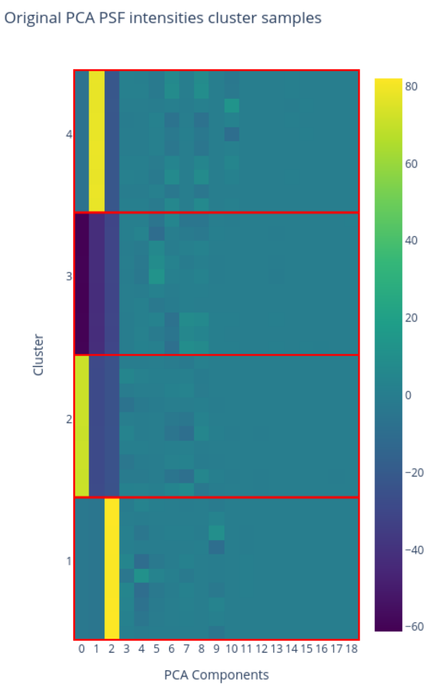
(a) Original cluster densities from PCA



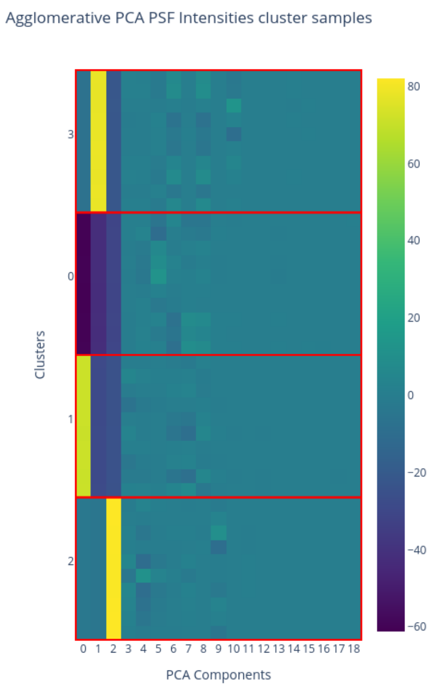
(b) Agglomerative clusters densities from PCA

(c) Original clusters from PCA

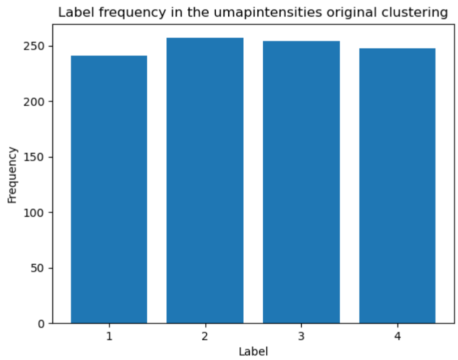
(d) Agglomerative clusters from PCA



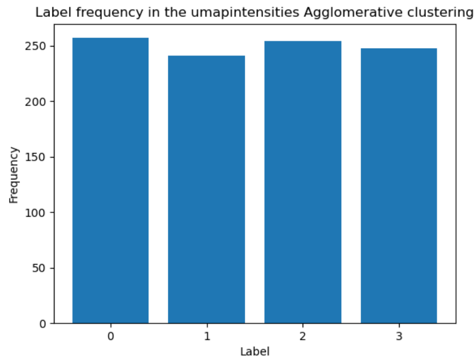
(e) Original cluster samples from PCA



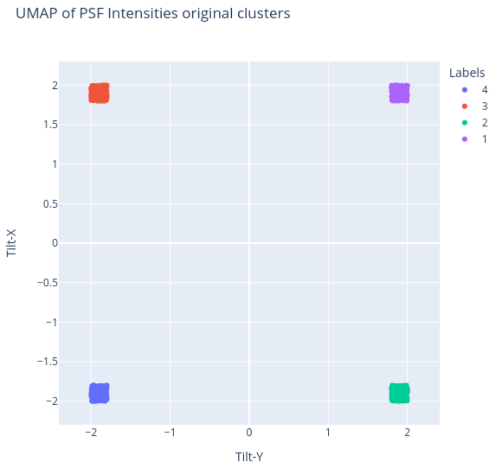
(f) Agglomerative cluster samples from PCA



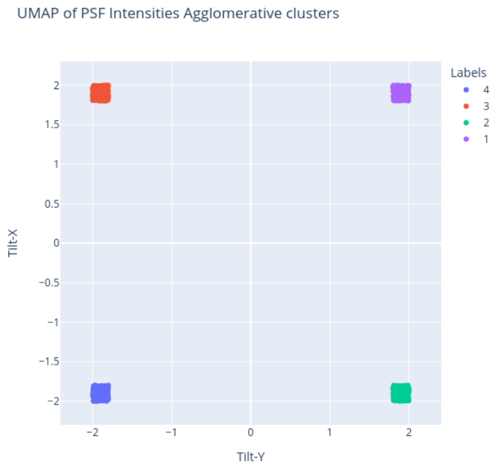
(a) Original cluster densities from UMAP



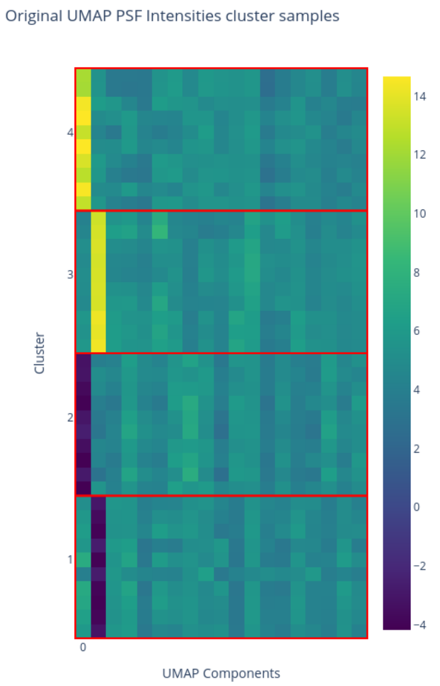
(b) Agglomerative clusters densities from UMAP



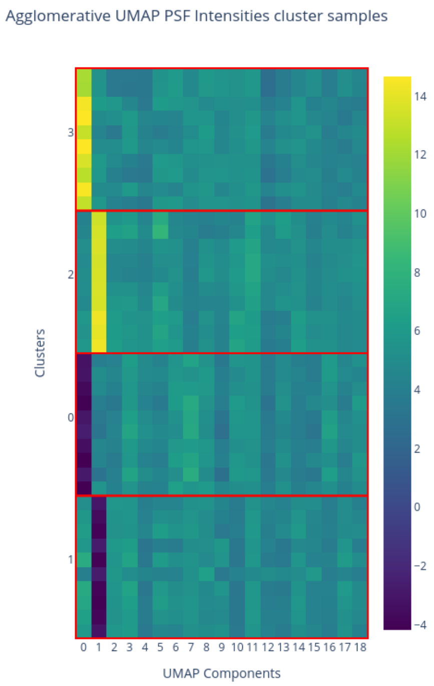
(c) Original clusters from UMAP



(d) Agglomerative clusters from UMAP

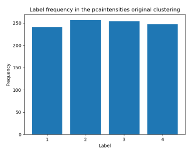


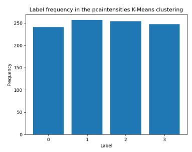
(e) Original cluster samples from UMAP

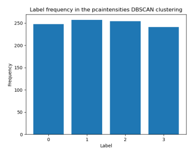


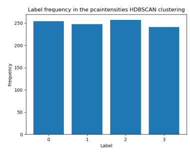
(f) Agglomerative cluster samples from UMAP

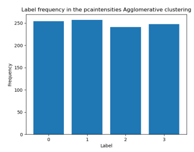
3.4.5 Summary

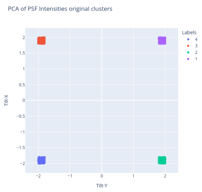

(a) Original cluster densities from PCA

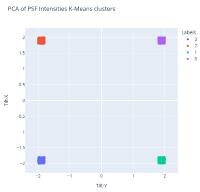

(b) K-means cluster densities from PCA

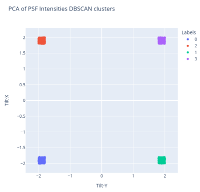

(c) DBSCAN cluster densities from PCA


(d) HDBSCAN cluster densities from PCA


(e) Agglomerative cluster densities from PCA

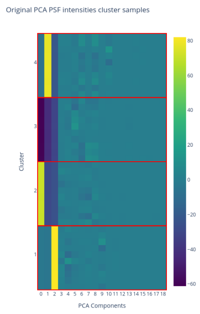

(f) Original cluster from PCA

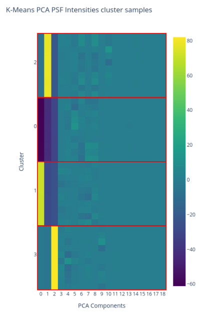

(g) K-means clusters from PCA

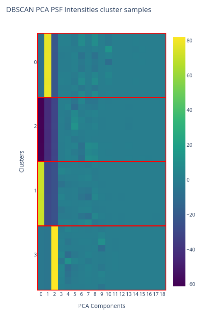

(h) DBSCAN cluster clusters from PCA

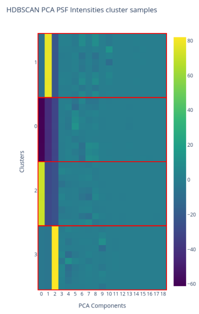

(i) HDBSCAN clusters from PCA


(j) Agglomerative clusters from PCA


(k) Original cluster samples from PCA


(l) K-means cluster samples from PCA


(m) DBSCAN cluster samples- from PCA


(n) HDBSCAN cluster samples from PCA

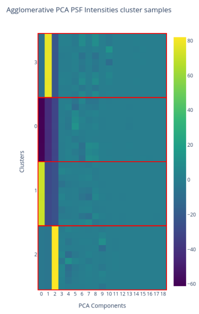
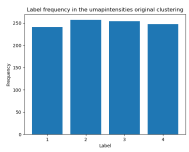

(o) Agglomerative cluster samples from PCA

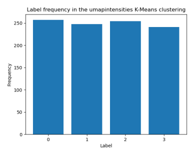
Figure 144: Comparison between clustering PCA PSF Intensities algorithms

	Original	**K-Means**	**DBSCAN**	**HDBSCAN**	**Agglomerative**
Original		1	1	1	1
K-Means			1	1	1
DBSCAN				1	1
HDBSCAN					1

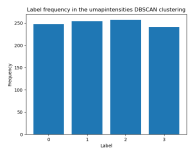
Table 35: Normalized Mutual Information between PCA PSF Intensities clusters



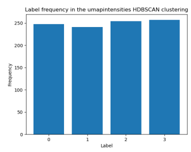
(a) Original cluster densities from UMAP



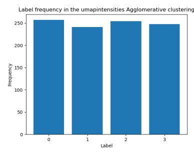
(b) K-means cluster densities from UMAP



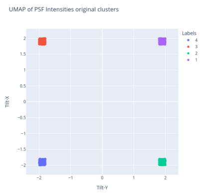
(c) DBSCAN cluster densities from UMAP



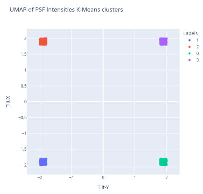
(d) HDBSCAN cluster densities from UMAP



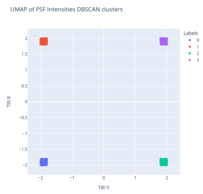
(e) Agglomerative cluster densities from UMAP



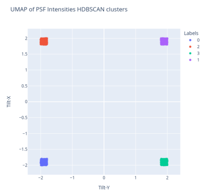
(f) Original cluster from UMAP



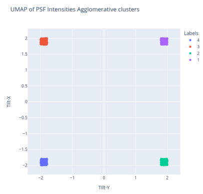
(g) K-means clusters from UMAP



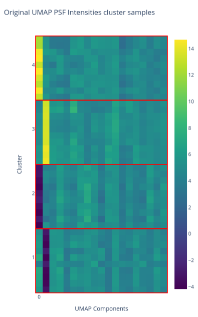
(h) DBSCAN cluster clusters from UMAP



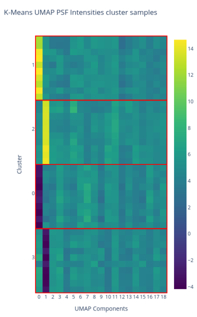
(i) HDBSCAN clusters from UMAP



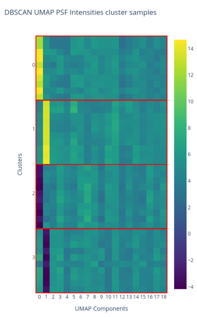
(j) Agglomerative clusters from UMAP



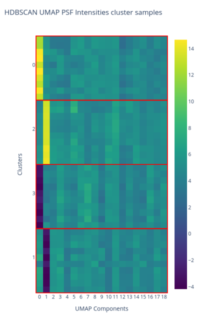
(k) Original cluster samples from UMAP



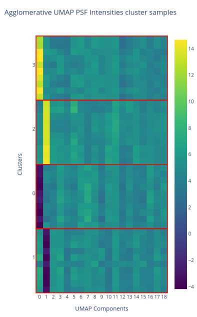
(l) K-means cluster samples from UMAP



(m) DBSCAN cluster samples-from UMAP



(n) HDBSCAN cluster samples from UMAP



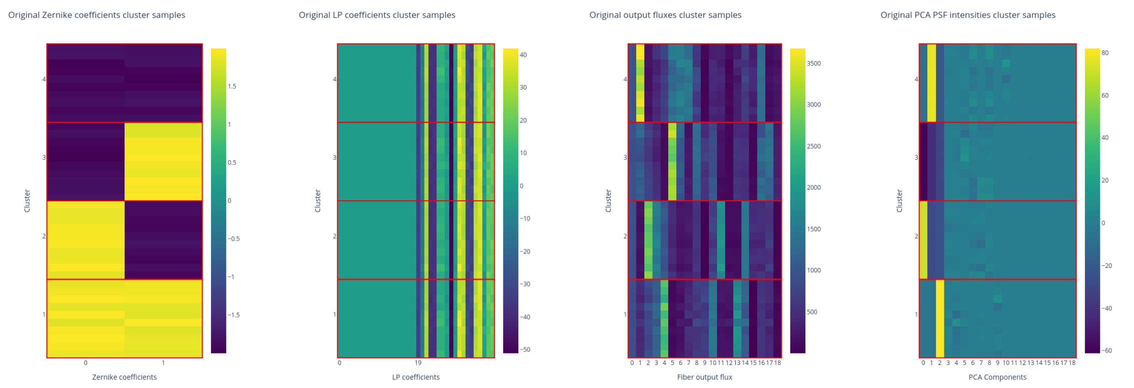
(o) Agglomerative cluster samples from UMAP

Figure 145: Comparison between clustering UMAP PSF Intensities algorithms

	Original	**K-Means**	**DBSCAN**	**HDBSCAN**	**Agglomerative**
Original		1	1	1	1
K-Means			1	1	1
DBSCAN				1	1
HDBSCAN					1

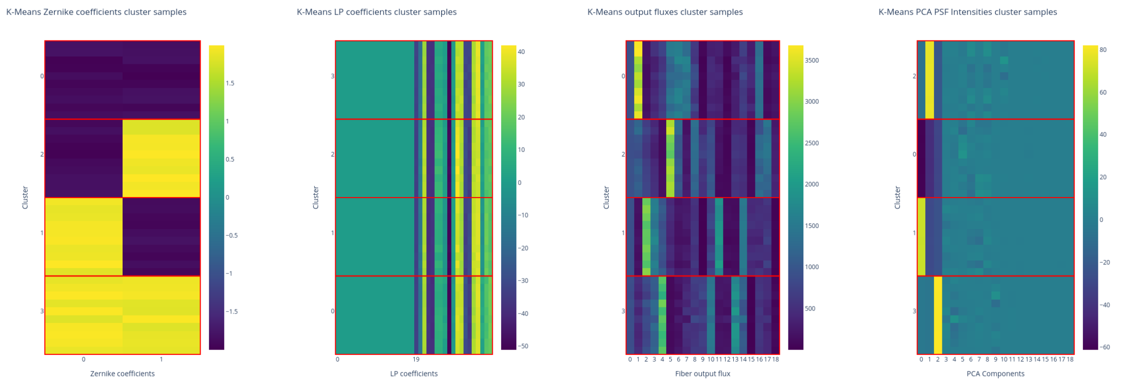
Table 36: Normalized Mutual Information between UMAP PSF Intensities clusters

4 Dataset clusters comparison



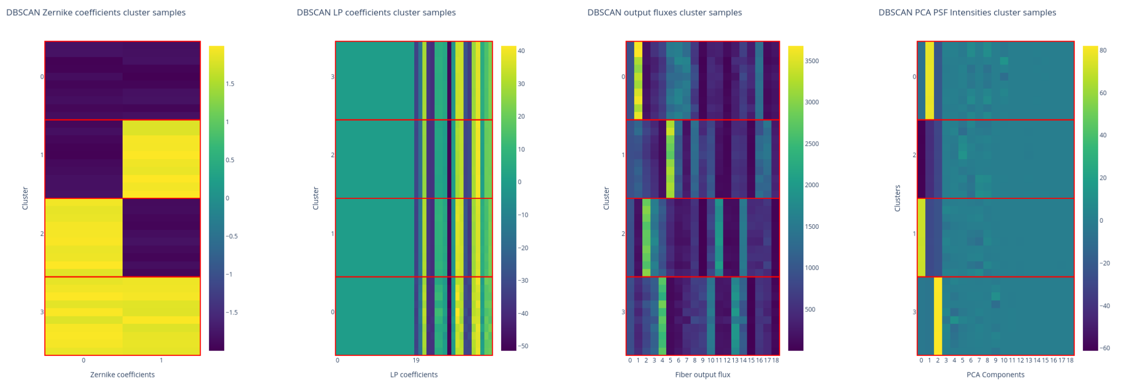
(a) Zernike coefficients (b) LP coefficients (c) Output fluxes (d) PCA PSF Intensities

Figure 146: Original clusters from the datasets



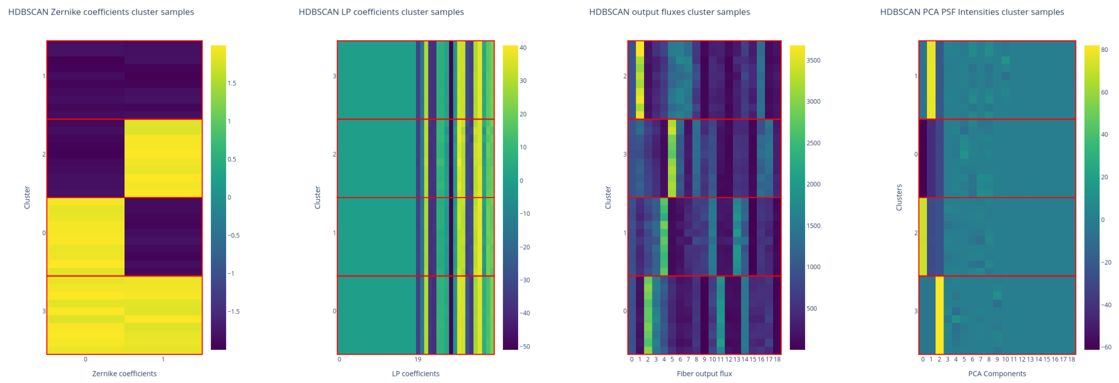
(a) Zernike coefficients (b) LP coefficients (c) Output fluxes (d) PCA PSF Intensities

Figure 147: K-Means clusters from the datasets



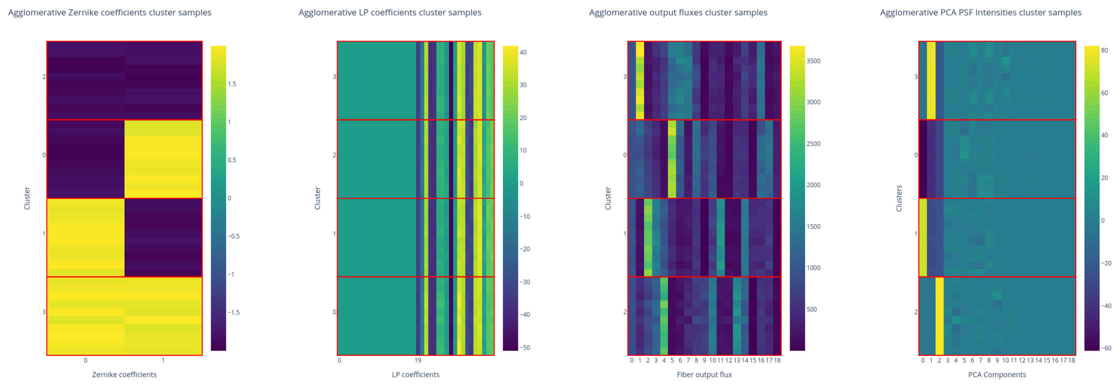
(a) Zernike coefficients (b) LP coefficients (c) Output fluxes (d) PCA PSF Intensities

Figure 148: DBSCAN clusters from the datasets



(a) Zernike coefficients (b) LP coefficients (c) Output fluxes (d) PCA PSF Intensities

Figure 149: HDBSCAN clusters from the datasets



(a) Zernike coefficients (b) LP coefficients (c) Output fluxes (d) PCA PSF Intensities

Figure 150: Agglomerative clusters from the datasets

Part V

Mini Dataset 5 Zernike modes PL Information Determination

5 The data

5.1 Zernike coefficients dataset

A dataset of 3200 zernike coefficients is created for this report.In particular, each datapoint represent the coefficients of the first 5 Zernike modes, their values ranging between:

- The first 2 modes between[-2, -1.8] and [1.8, 2].
- Modes 4, 5 and 6 between [-1, -0.8] and [0.8, 1]

These ranges create 32 original clusters that will be used as reference.

5.2 PSFs intensities dataset

A dataset of 3200 PSFs is created using the Zernike coefficients dataset.

5.3 LP mode coefficients dataset

A dataset of 3200 LP mode coefficients obtained from computing the overlap integral of the first 19 LP modes with the PSF dataset.

5.4 LP mode coefficients dataset

A dataset of 3200 PL output fluxes obtained from the PL transfer matrix and LP coefficients.

6 Preprocessing

6.1 PSF Intensities

The 3200x128x128 array is dimensionally reduced using PCA and UMAP both giving an array of 3200x19 projections of the PSF Intensities.

7 Clustering

A series of different clustering algorithms are used:

- K-Means
- DBSCAN
- HDBSCAN
- Agglomerative clustering

The clusters obtained will be compared the original clusters using NMI

7.1 Zernike coefficients clustering

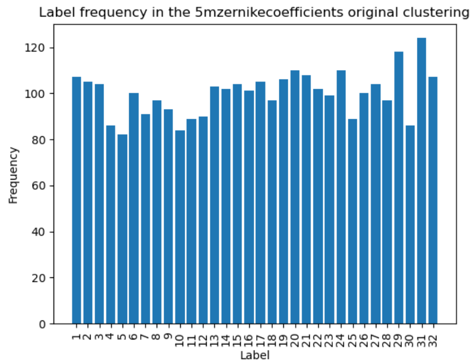
7.1.1 K-Means

As K-Means allows for the number of clusters to define, and we know that there are 4 in the original dataset, K-Means is used to find 4 clusters.

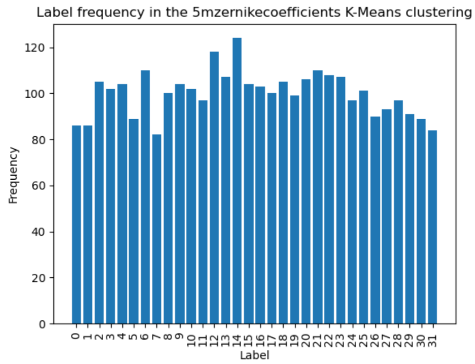
Number of clusters	**Number of initializations**
32	100

Table 37: K-Means hyperparameter configuration for Zernike coefficients clustering

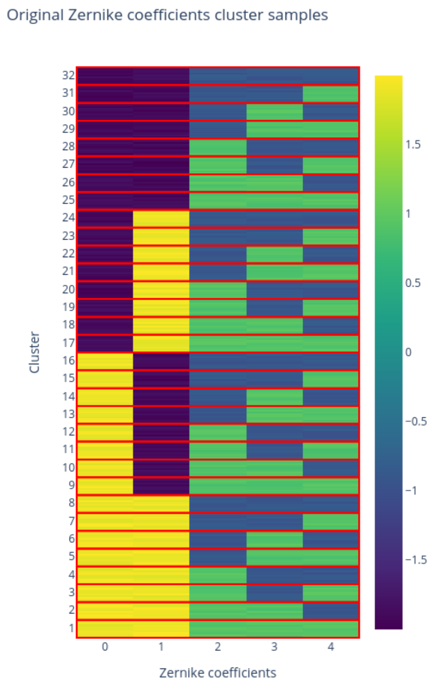
The results are the following:



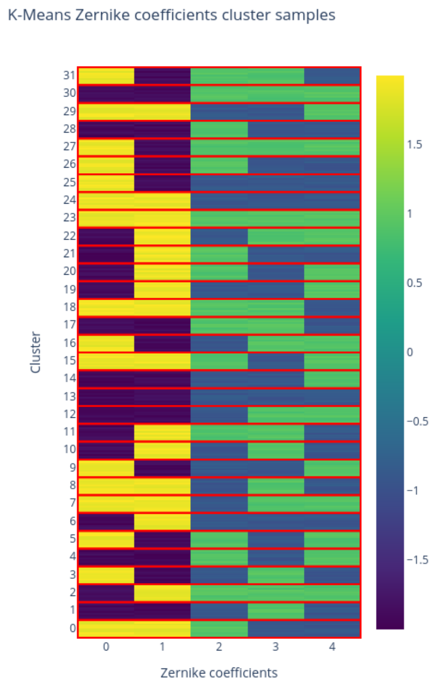
(a) Original cluster densities



(b) K-Means clusters densities



(c) Original cluster samples



(d) K-Means cluster samples

Figure 151: Comparison between original clustering and K-Means clustering

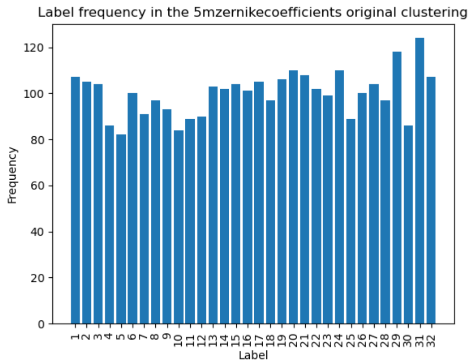
7.1.2 DBSCAN

A configuration that outputs 4 clusters is searched

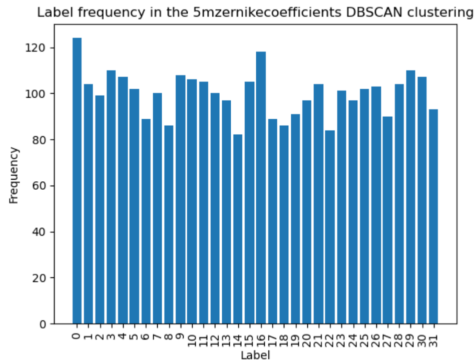
Number of neighbours	**Epsilon**
5	0.14

Table 38: DBSCAN hyperparameter configuration for Zernike coefficients clustering

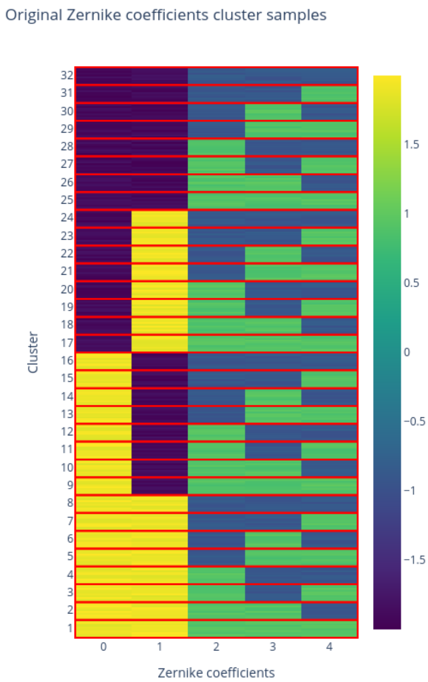
The results are the following:



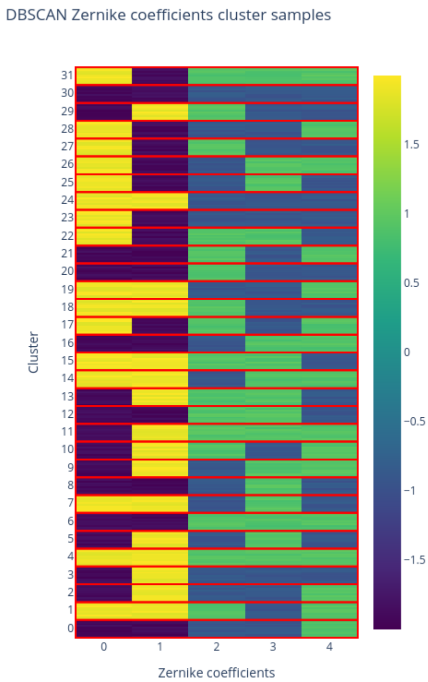
(a) Original cluster densities



(b) DBSCAN clusters densities



(c) Original cluster samples



(d) DBSCAN cluster samples

Figure 152: Comparison between original clustering and DBSCAN clustering

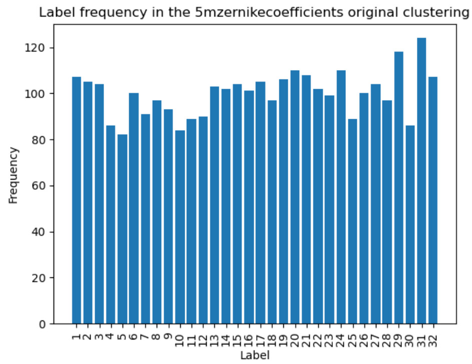
7.1.3 HDBSCAN

A configuration that outputs 4 clusters is searched.

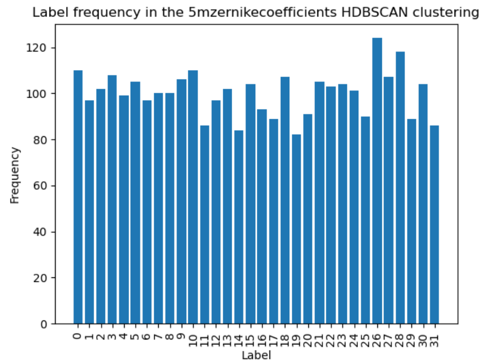
Minimum cluster size
50

Table 39: HDBSCAN hyperparameter configuration for Zernike coefficients clustering

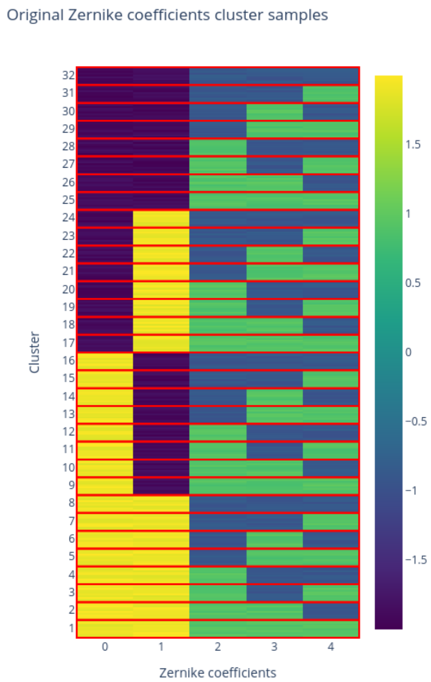
The results are the following:



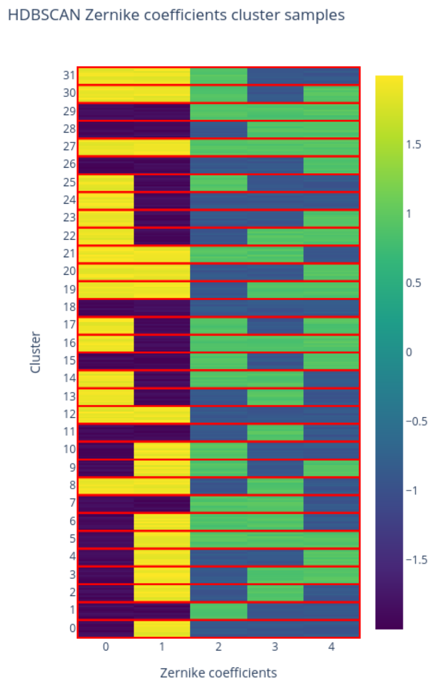
(a) Original cluster densities



(b) HDBSCAN clusters densities



(c) Original cluster samples



(d) HDBSCAN cluster samples

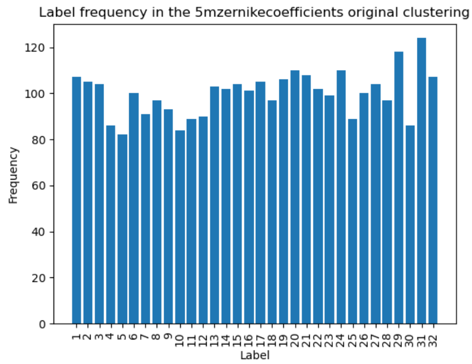
Figure 153: Comparison between original clustering and HDBSCAN clustering

7.1.4 Agglomerative clustering

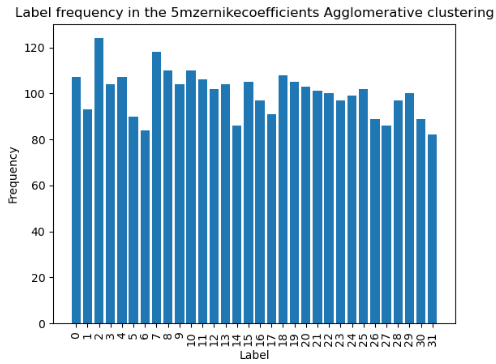
Number of clusters
32

Table 40: Agglomerative hyperparameter configuration for Zernike coefficients clustering

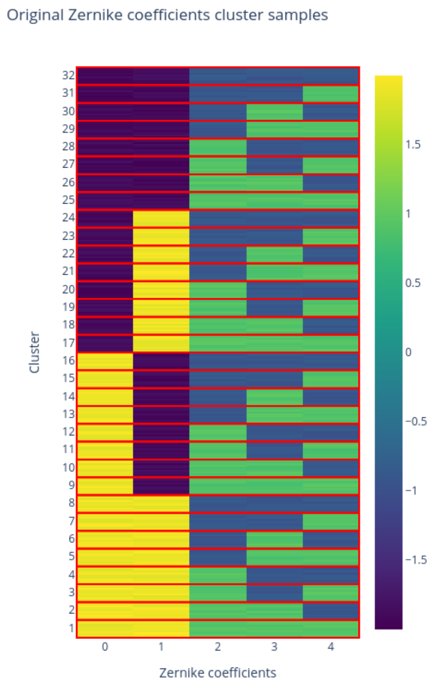
The results are the following:



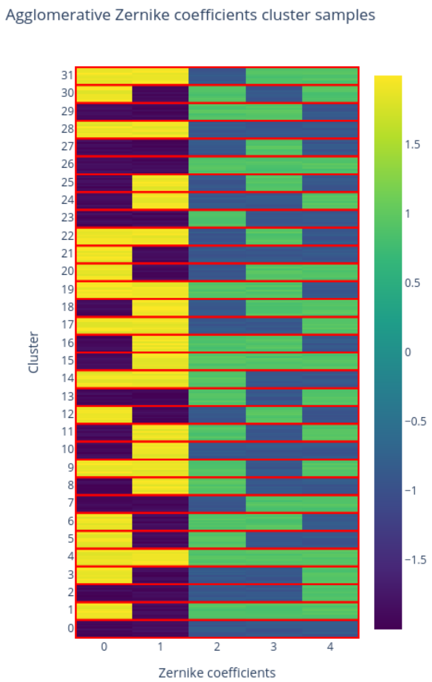
(a) Original cluster densities



(b) Agglomerative clusters densities



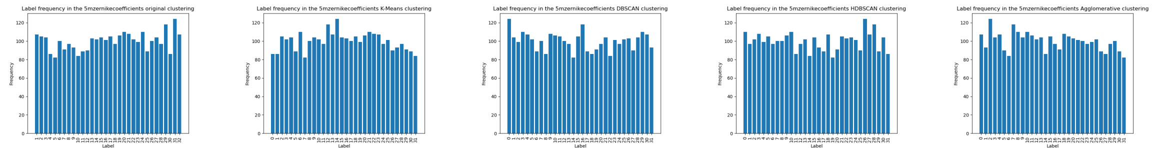
(c) Original cluster samples



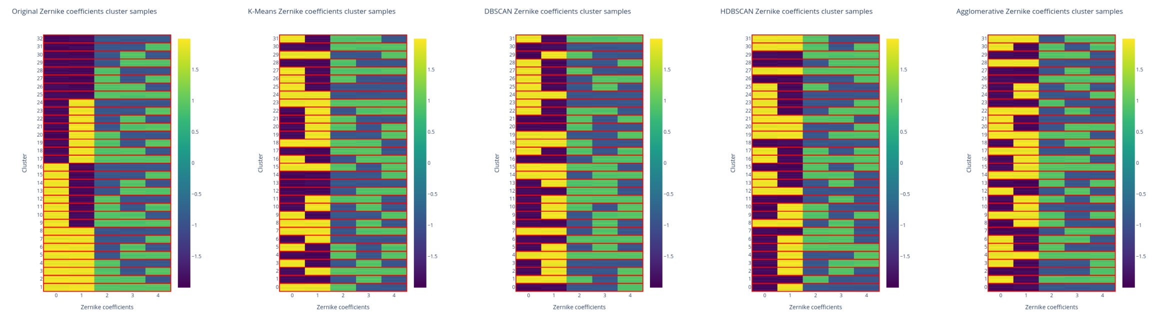
(d) Agglomerative cluster samples

Figure 154: Comparison between original clustering and Agglomerative clustering

7.1.5 Summary



(a) Original cluster densities (b) K-means cluster densities (c) DBSCAN cluster densities (d) HDBSCAN cluster densities (e) Agglomerative cluster densities



(f) Original cluster samples (g) K-means cluster samples (h) DBSCAN cluster samples (i) HDBSCAN cluster samples (j) Agglomerative cluster samples

Figure 155: Comparison between clustering algorithms

	Original	**K-Means**	**DBSCAN**	**HDBSCAN**	**Agglomerative**
Original	\	1	1	1	1
K-Means		\	1	1	1
DBSCAN			\	1	1
HDBSCAN				\	1

Table 41: Normalized Mutual Information between clusters

7.2 LP coefficients clustering

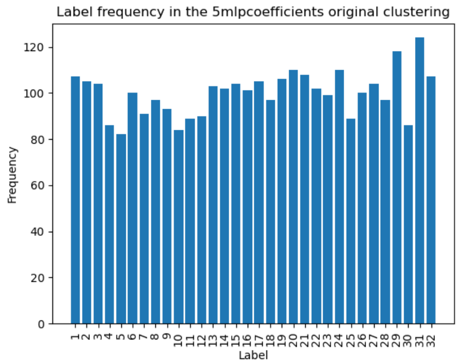
7.2.1 K-Means

As K-Means allows for the number of clusters to be defined, and we know that there are 4 in the original dataset, K-Means is used to find 4 clusters.

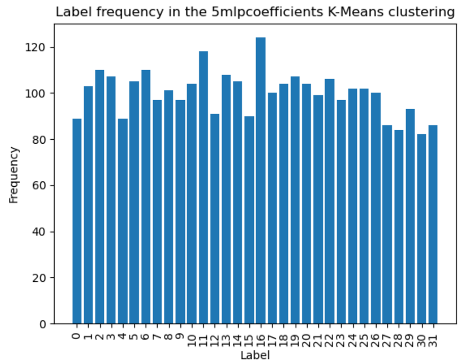
The results are the following:

	Number of clusters	**Number of initializations**
Original LP coefficients	32	100

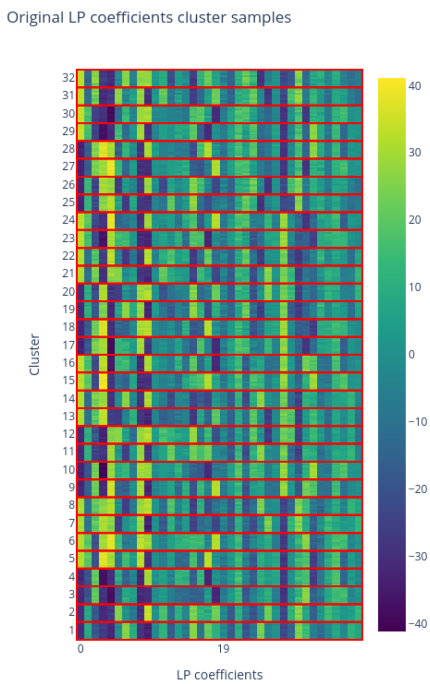
Table 42: K-Means hyperparameter configuration for c coefficients clustering



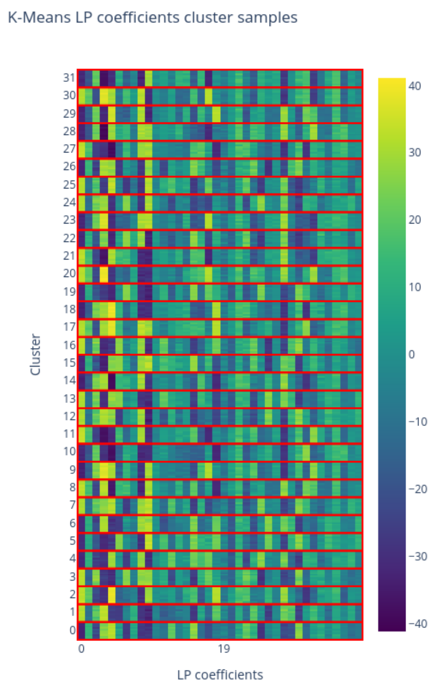
(a) Original cluster densities



(b) K-Means clusters densities



(c) Original cluster samples



(d) K-Means cluster samples

Figure 156: Comparison between original clustering and K-Means clustering from original LP coefficients

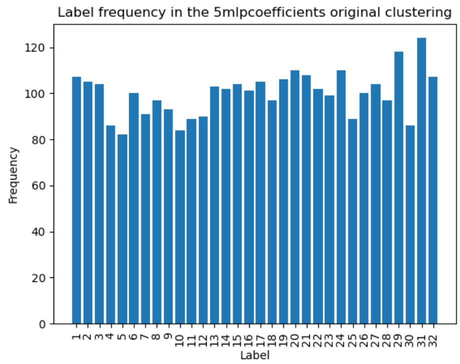
7.2.2 DBSCAN

A configuration that outputs 4 clusters is searched

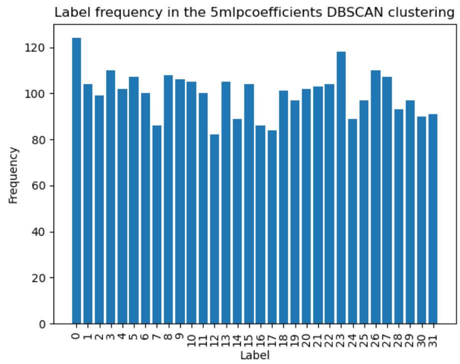
	Number of neighbours	**Epsilon**
Original LP coefficients	15	11

Table 43: DBSCAN hyperparameter configuration for LP coefficients clustering

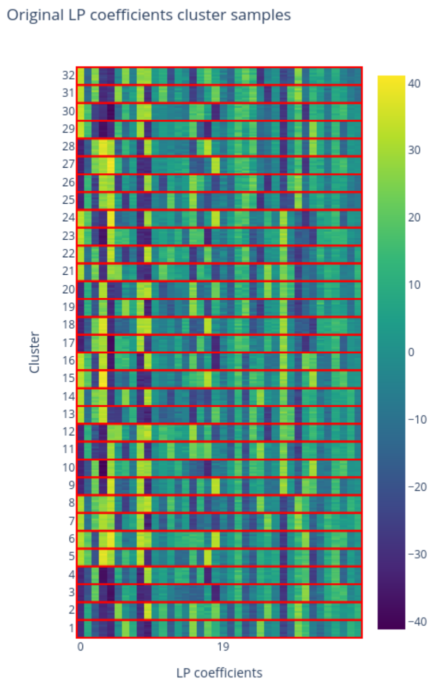
The results are the following:



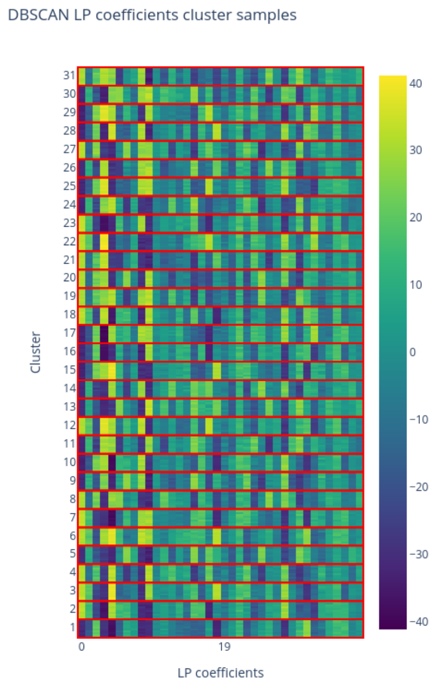
(a) Original cluster densities



(b) DBSCAN clusters densities



(c) Original cluster samples



(d) DBSCAN cluster samples

Figure 157: Comparison between original clustering and DBSCAN clustering

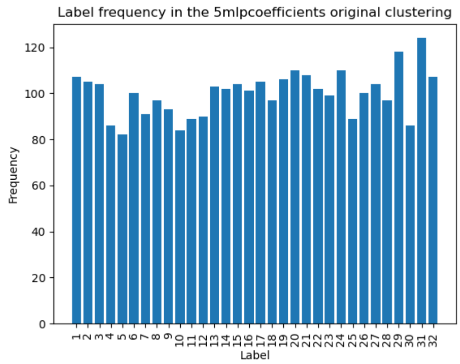
7.2.3 HDBSCAN

A configuration that outputs 4 clusters is searched.

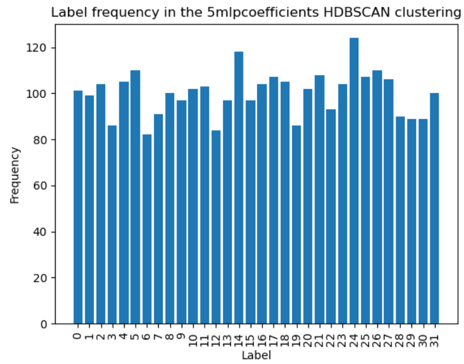
	Minimum cluster size
Original LP coefficients	21

Table 44: HDBSCAN hyperparameter configuration for LP coefficients clustering

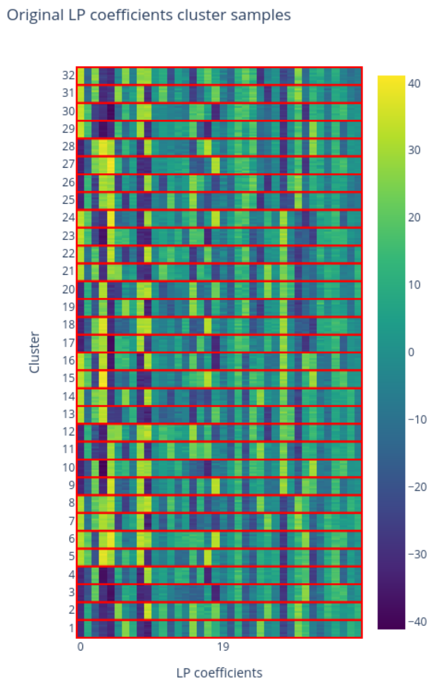
The results are the following:



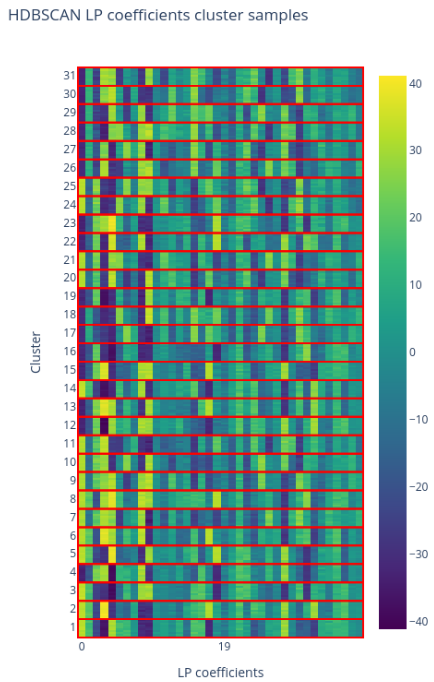
(a) Original cluster densities



(b) HDBSCAN clusters densities



(c) Original cluster samples



(d) HDBSCAN cluster samples

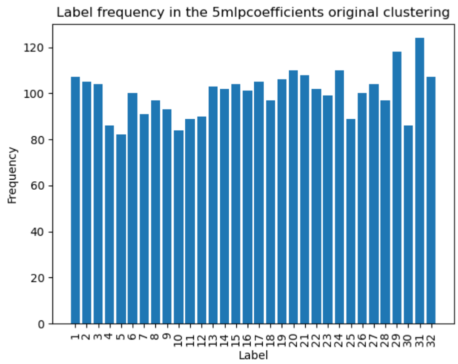
Figure 158: Comparison between original clustering and HDBSCAN clustering

7.2.4 Agglomerative clustering

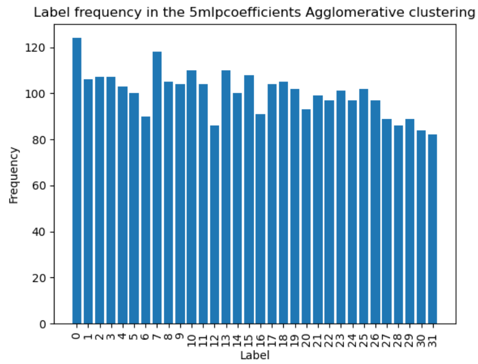
	Number of clusters
Original LP coefficients	4

Table 45: Agglomerative hyperparameter configuration for LP coefficients clustering

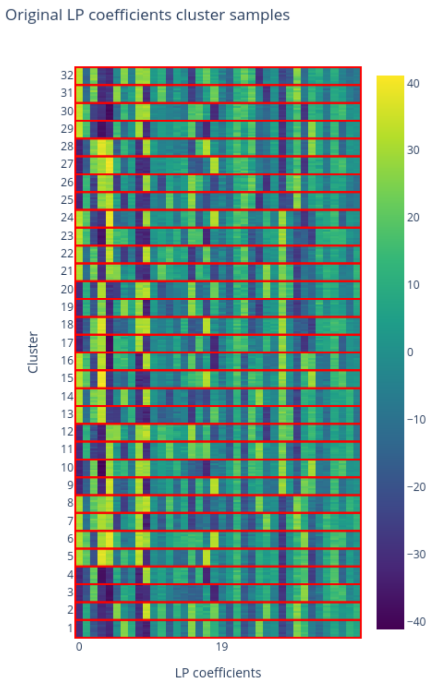
The results are the following:



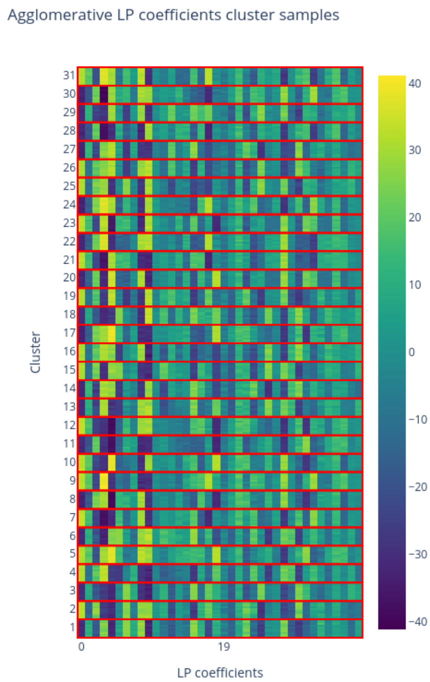
(a) Original cluster densities



(b) Agglomerative clusters densities



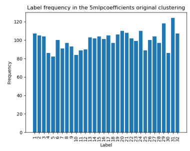
(c) Original cluster samples



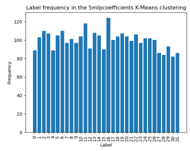
(d) Agglomerative cluster samples

Figure 159: Comparison between original clustering and Agglomerative clustering

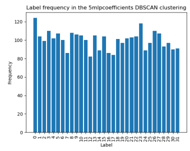
7.2.5 Summary



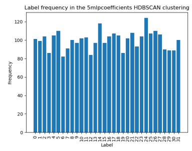
(a) Original cluster densities



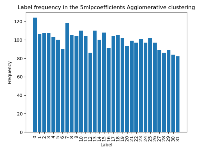
(b) K-means cluster densities



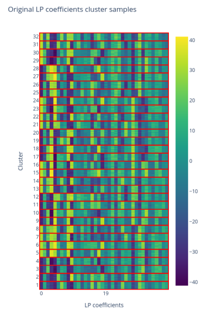
(c) DBSCAN cluster densities



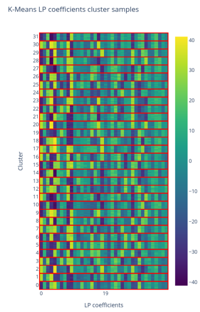
(d) HDBSCAN cluster densities



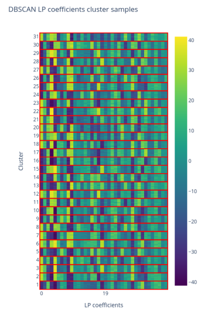
(e) Agglomerative cluster densities



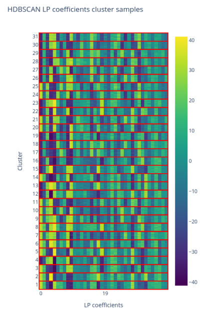
(f) Original cluster samples



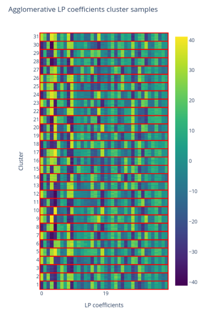
(g) K-means cluster samples



(h) DBSCAN cluster samples



(i) HDBSCAN cluster samples



(j) Agglomerative cluster samples

Figure 160: Comparison between clustering LP coefficients algorithms

	Original	**K-Means**	**DBSCAN**	**HDBSCAN**	**Agglomerative**
Original		1	1	1	1
K-Means			1	1	1
DBSCAN				1	1
HDBSCAN					1

Table 46: Normalized Mutual Information between original LP coefficients clusters

7.3 Output fluxes clustering

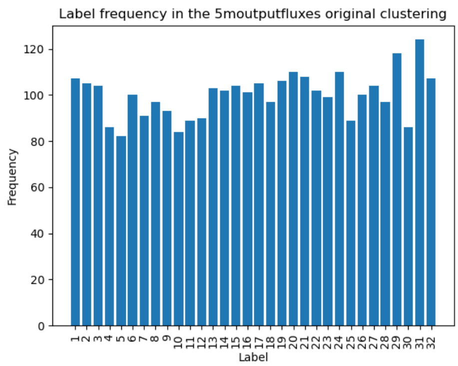
7.3.1 K-Means

As K-Means allows for the number of clusters to be defined, and we know that there are 4 in the original dataset, K-Means is used to find 4 clusters.

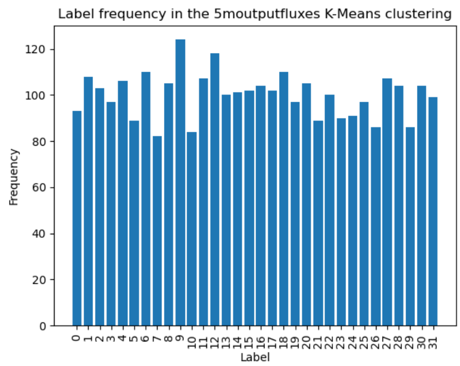
The results are the following:

	Number of clusters	**Number of initializations**
Original Output fluxes	32	100

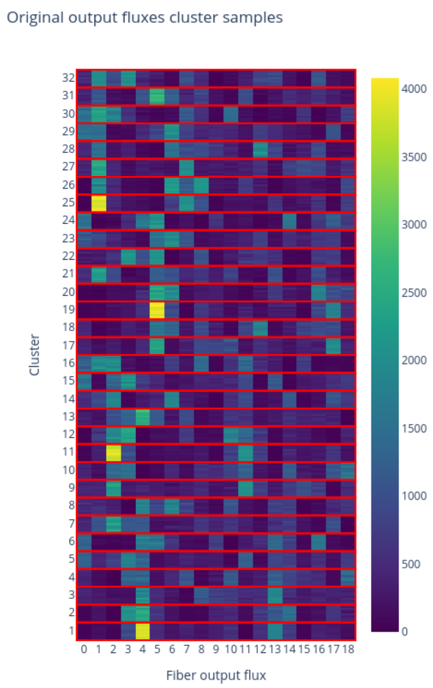
Table 47: K-Means hyperparameter configuration for c coefficients clustering



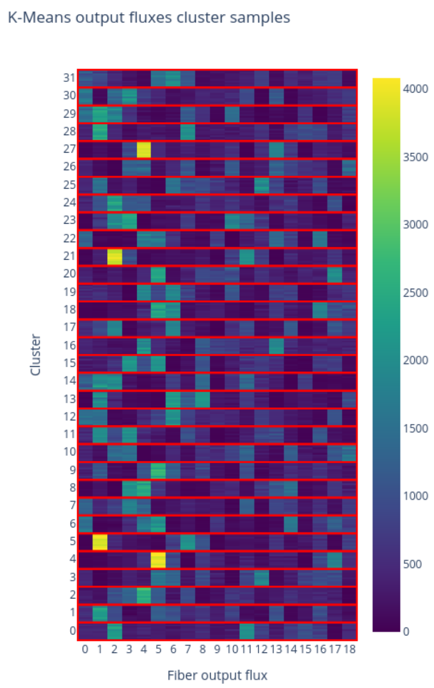
(a) Original cluster densities



(b) K-Means clusters densities



(c) Original cluster samples



(d) K-Means cluster samples

Figure 161: Comparison between original clustering and K-Means clustering from original Output fluxes

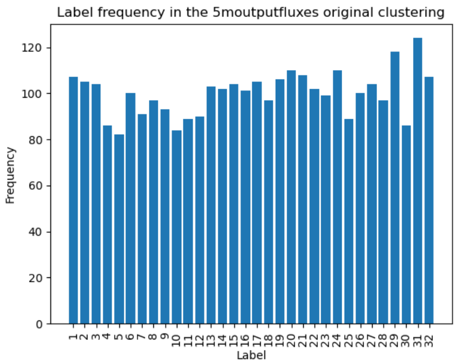
7.3.2 DBSCAN

A configuration that outputs 4 clusters is searched

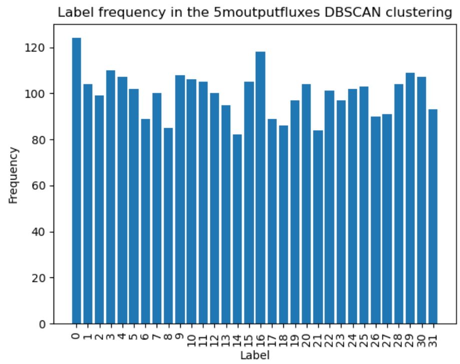
	Number of neighbours	**Epsilon**
Original Output fluxes	7	300

Table 48: DBSCAN hyperparameter configuration for Output fluxes clustering

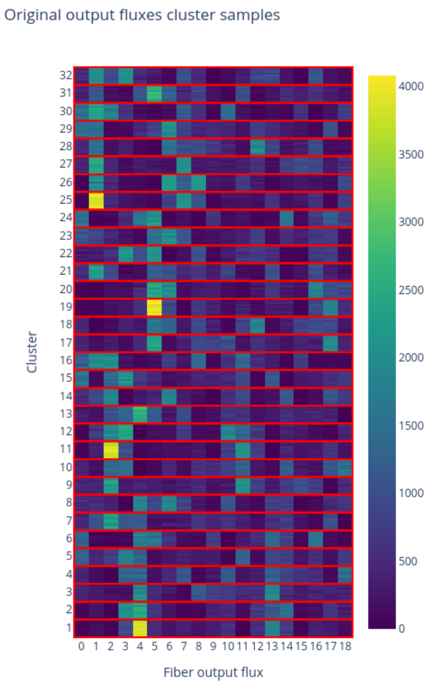
The results are the following:



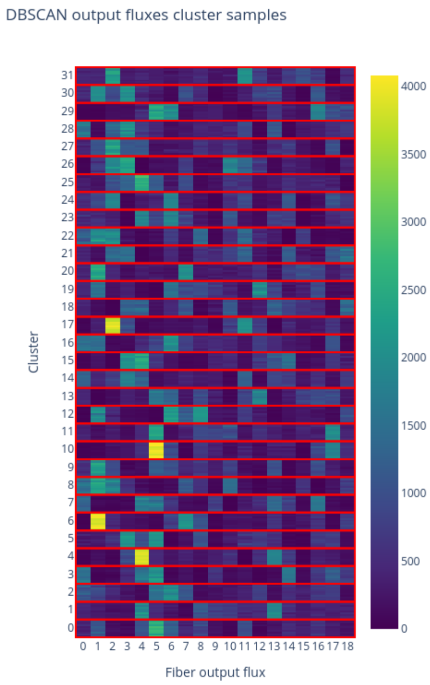
(a) Original cluster densities



(b) DBSCAN clusters densities



(c) Original cluster samples



(d) DBSCAN cluster samples

Figure 162: Comparison between original clustering and DBSCAN clustering

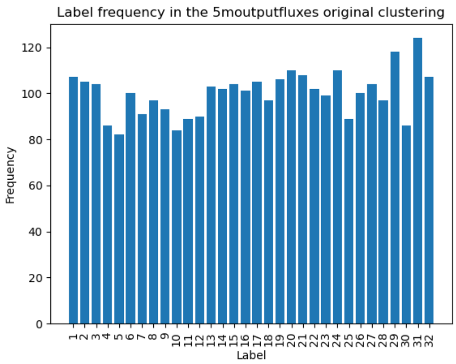
7.3.3 HDBSCAN

A configuration that outputs 4 clusters is searched.

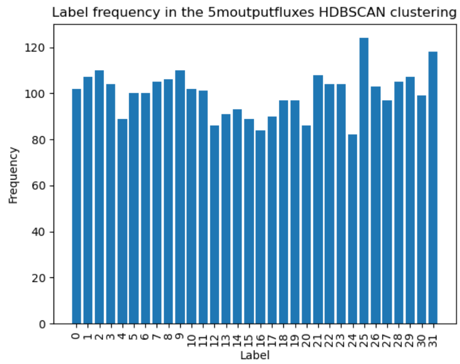
	Minimum cluster size
Original Output fluxes	21

Table 49: HDBSCAN hyperparameter configuration for Output fluxes clustering

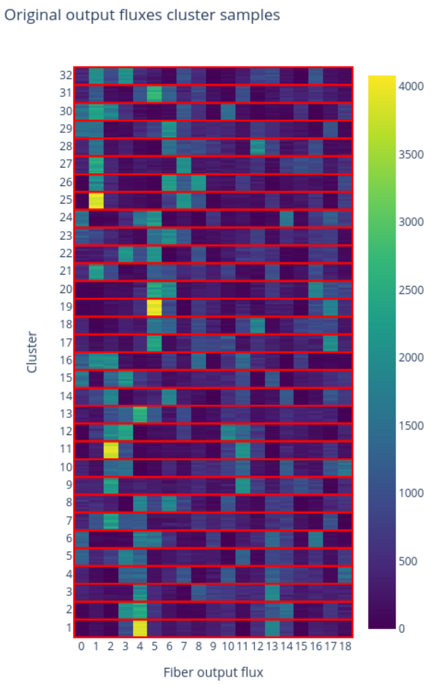
The results are the following:



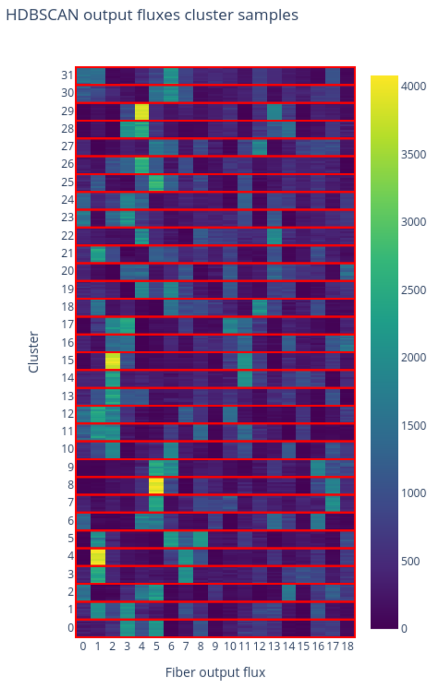
(a) Original cluster densities



(b) HDBSCAN clusters densities



(c) Original cluster samples



(d) HDBSCAN cluster samples

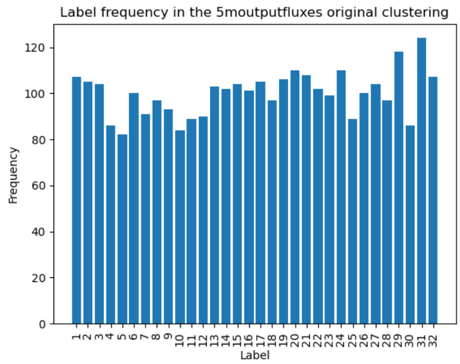
Figure 163: Comparison between original clustering and HDBSCAN clustering

7.3.4 Agglomerative clustering

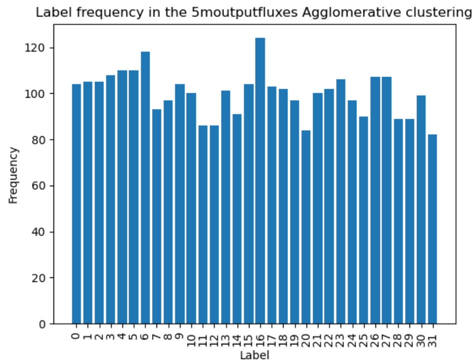
	Number of clusters
Original Output fluxes	32

Table 50: Agglomerative hyperparameter configuration for Output fluxes clustering

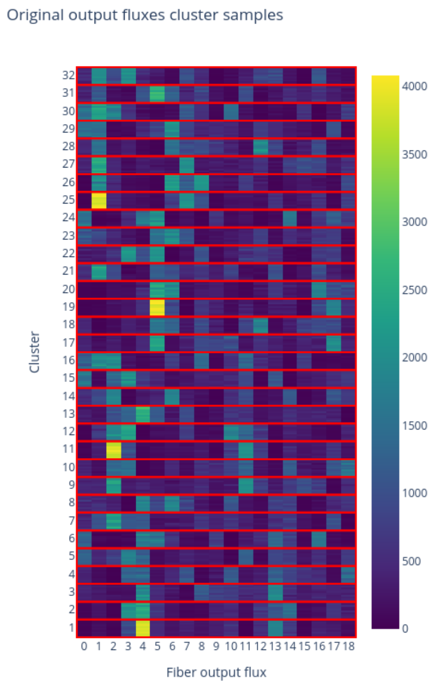
The results are the following:



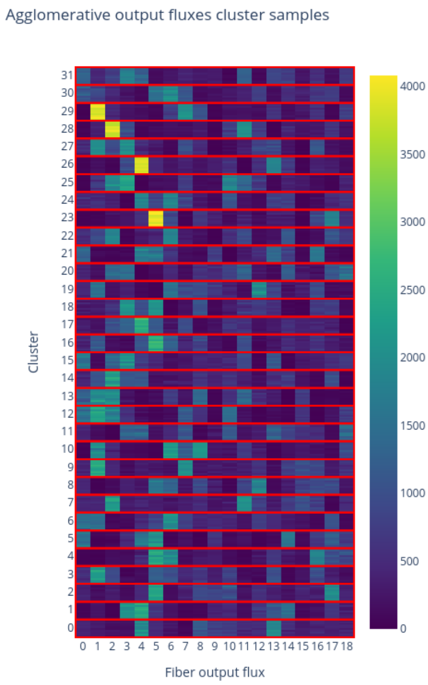
(a) Original cluster densities



(b) Agglomerative clusters densities



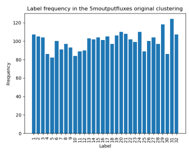
(c) Original cluster samples



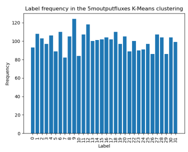
(d) Agglomerative cluster samples

Figure 164: Comparison between original clustering and Agglomerative clustering

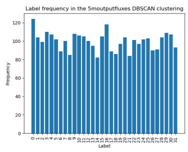
7.3.5 Summary



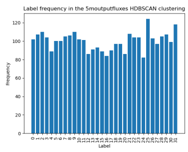
(a) Original cluster densities



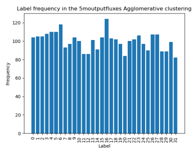
(b) K-means cluster densities



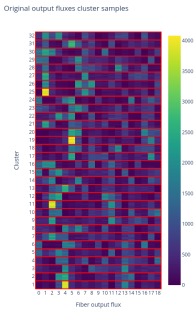
(c) DBSCAN cluster densities



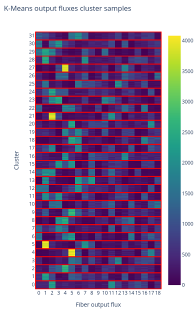
(d) HDBSCAN cluster densities



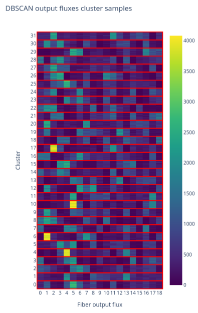
(e) Agglomerative cluster densities



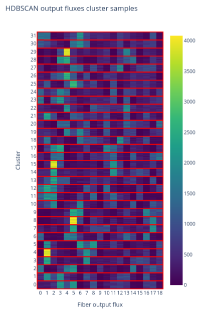
(f) Original cluster samples



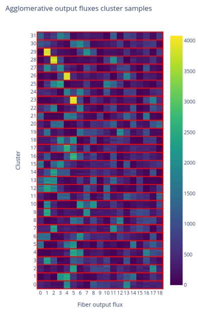
(g) K-means cluster samples



(h) DBSCAN cluster samples



(i) HDBSCAN cluster samples



(j) Agglomerative cluster samples

Figure 165: Comparison between clustering Output fluxes algorithms

	Original	**K-Means**	**DBSCAN**	**HDBSCAN**	**Agglomerative**
Original	\	1	0.998	1	1
K-Means		\	0.998	1	1
DBSCAN			\	0.998	0.998
HDBSCAN				\	1

Table 51: Normalized Mutual Information between original Output fluxes clusters

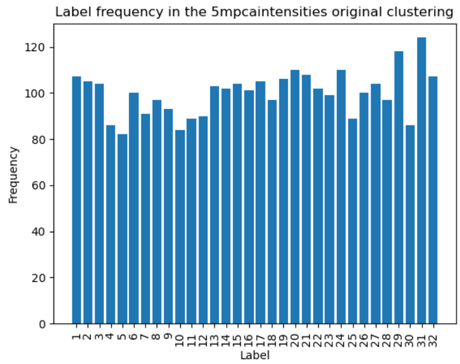
7.4 PSF Intensities clustering

7.4.1 K-Means

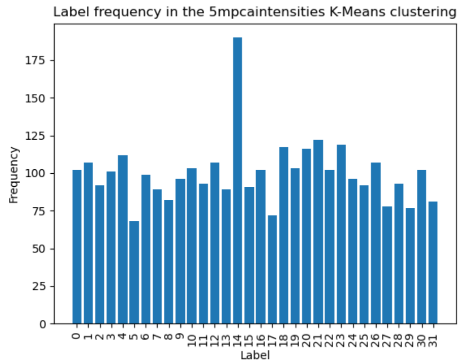
As K-Means allows for the number of clusters to be defined, and we know that there are 4 in the original dataset, K-Means is used to find 4 clusters.

	Number of clusters	**Number of initializations**
PCA PSF Intensities	32	200

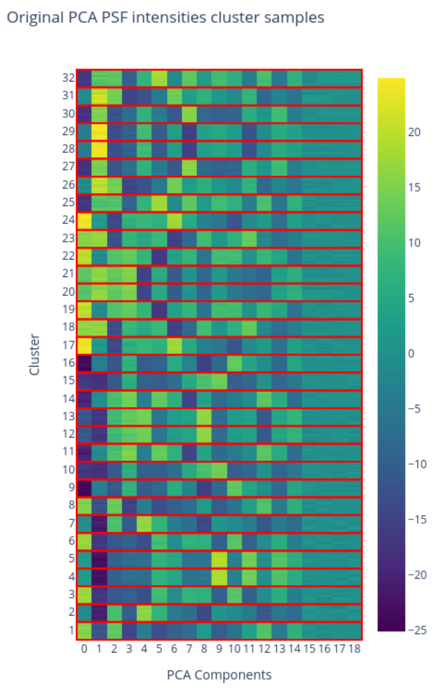
Table 52: K-Means hyperparameter configuration for c coefficients clustering



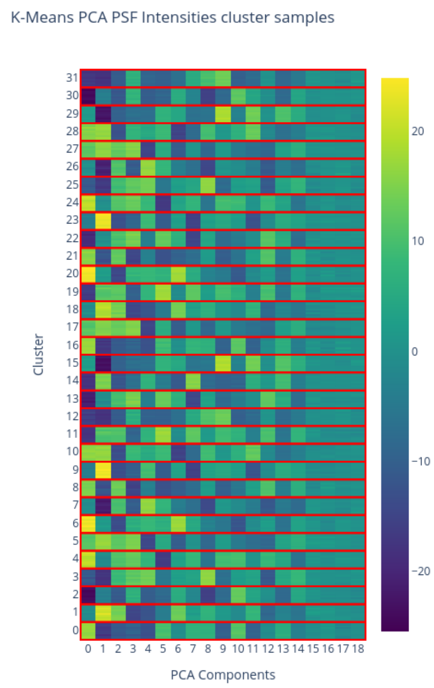
(a) Original cluster densities from PCA



(b) K-Means clusters densities from PCA



(c) Original cluster samples from PCA



(d) K-Means cluster samples from PCA

Figure 166: Comparison between original clustering and K-Means clustering from PCA of PSF Intensities

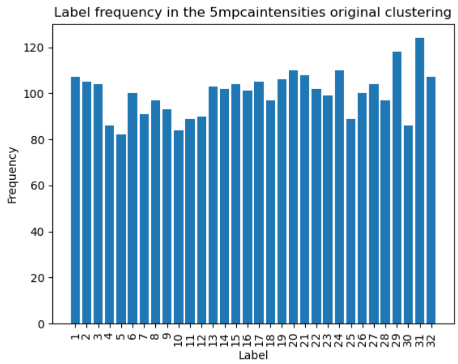
7.4.2 DBSCAN

A configuration that outputs 4 clusters is searched

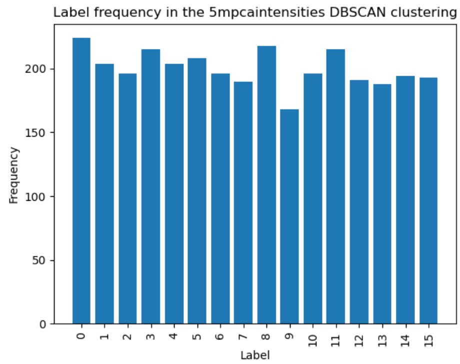
	Number of neighbours	**Epsilon**
PCA PSF Intensities	100	5

Table 53: DBSCAN hyperparameter configuration for PSF Intensities clustering

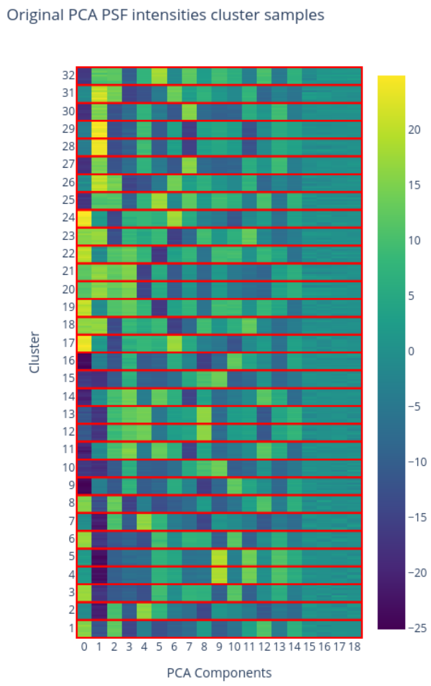
The results are the following:



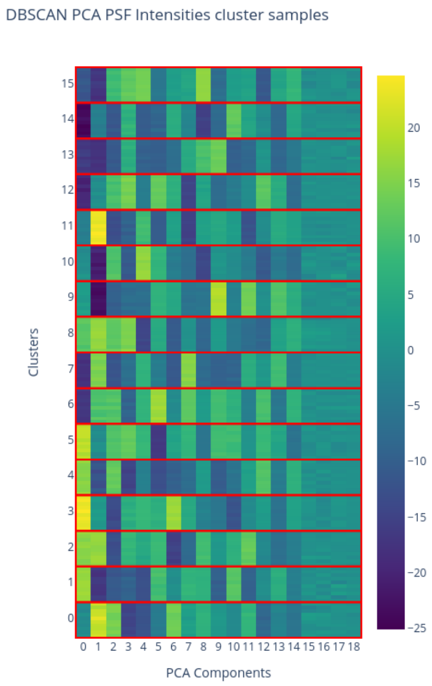
(a) Original cluster densities from PCA



(b) DBSCAN clusters densities from PCA



(c) Original cluster samples from PCA



(d) DBSCAN cluster samples from PCA

Figure 167: Comparison between original clustering and DBSCAN clustering from PCA of PSF Intensities

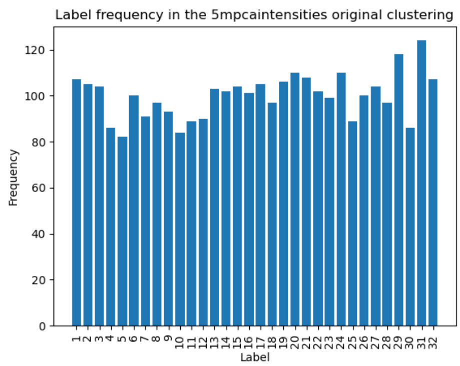
7.4.3 HDBSCAN

A configuration that outputs 4 clusters is searched.

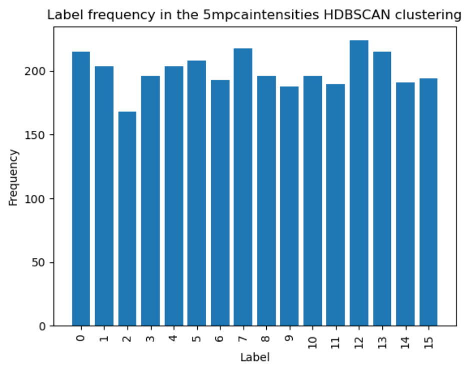
	Minimum cluster size
PCA PSF Intensities	100

Table 54: HDBSCAN hyperparameter configuration for PSF Intensities clustering

The results are the following:

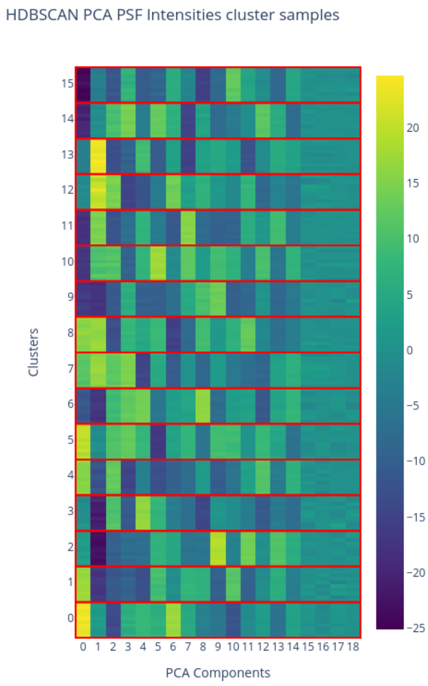


(a) Original cluster densities from PCA



(b) HDBSCAN clusters densities from PCA

(c) Original cluster samples from PCA



(d) HDBSCAN cluster samples from PCA

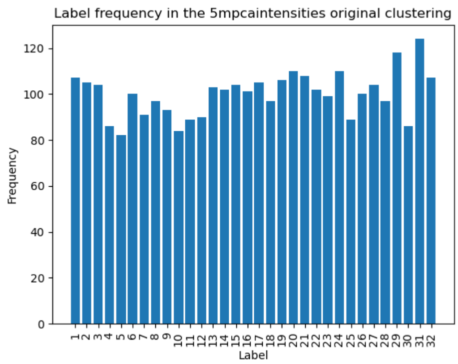
Figure 168: Comparison between original clustering and HDBSCAN clustering from PCA of PSF Intensities

7.4.4 Agglomerative clustering

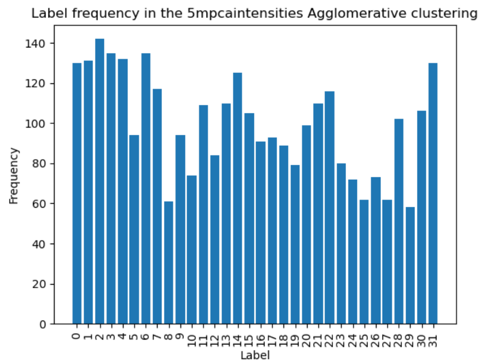
	Number of clusters
PCA PSF Intensities	32

Table 55: Agglomerative hyperparameter configuration for PSF Intensities clustering

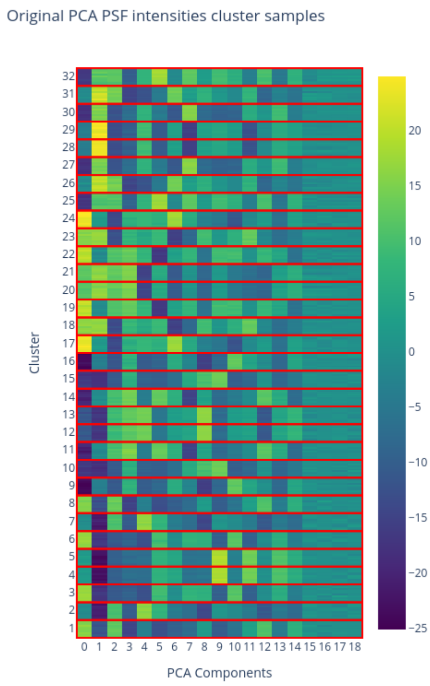
The results are the following:



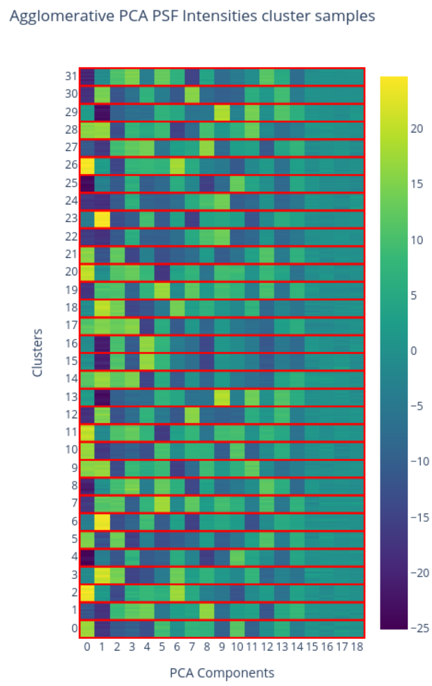
(a) Original cluster densities from PCA



(b) Agglomerative clusters densities from PCA



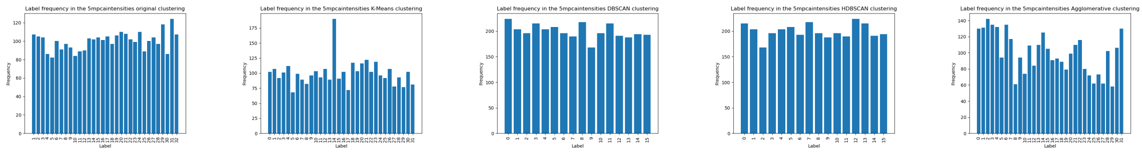
(c) Original cluster samples from PCA



(d) Agglomerative cluster samples from PCA

Figure 169: Comparison between original clustering and Agglomerative clustering

7.4.5 Summary



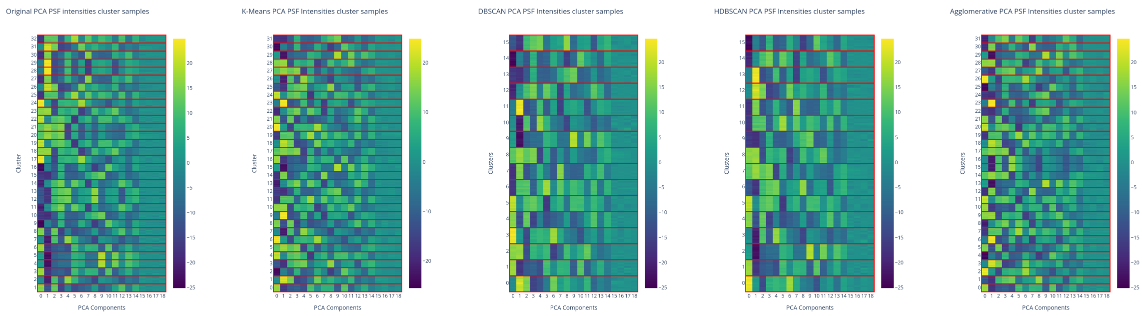
(a) Original cluster densities from PCA

(b) K-means cluster densities from PCA

(c) DBSCAN cluster densities from PCA

(d) HDBSCAN cluster densities from PCA

(e) Agglomerative cluster densities from PCA



(f) Original cluster samples from PCA

(g) K-means cluster samples from PCA

(h) DBSCAN cluster samples-from PCA

(i) HDBSCAN cluster samples from PCA

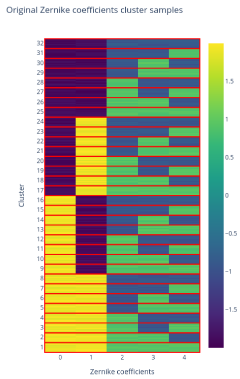
(j) Agglomerative cluster samples from PCA

Figure 170: Comparison between clustering PCA PSF Intensities algorithms

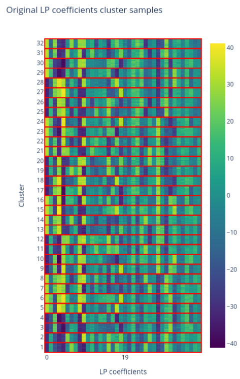
	Original	**K-Means**	**DBSCAN**	**HDBSCAN**	**Agglomerative**
Original		0.802	0.889	0.889	0.803
K-Means			0.891	0.891	0.909
DBSCAN				1	0.892
HDBSCAN					0.892

Table 56: Normalized Mutual Information between PCA PSF Intensities clusters

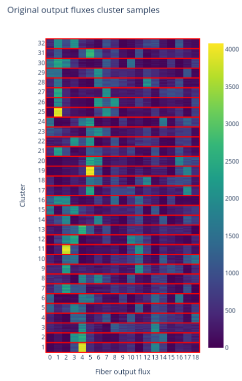
8 Dataset clusters comparison



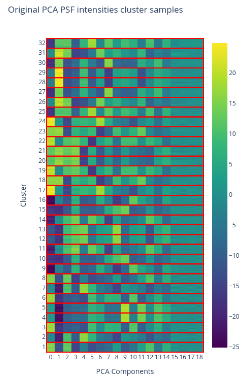
(a) Zernike coefficients



(b) LP coefficients

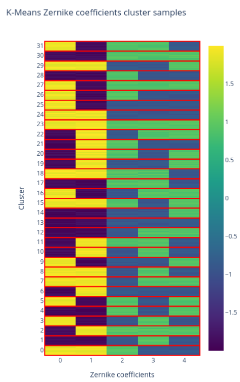


(c) Output fluxes

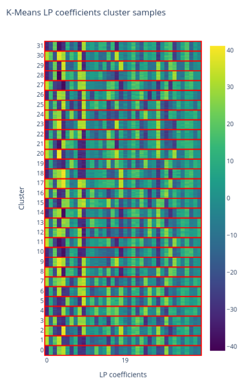


(d) PCA PSF Intensities

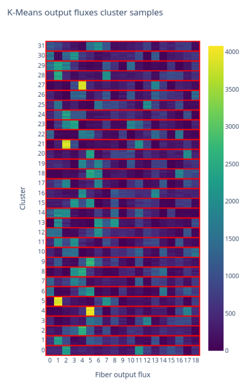
Figure 171: Original clusters from the datasets



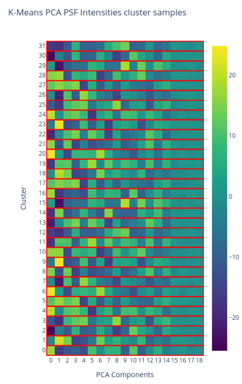
(a) Zernike coefficients



(b) LP coefficients

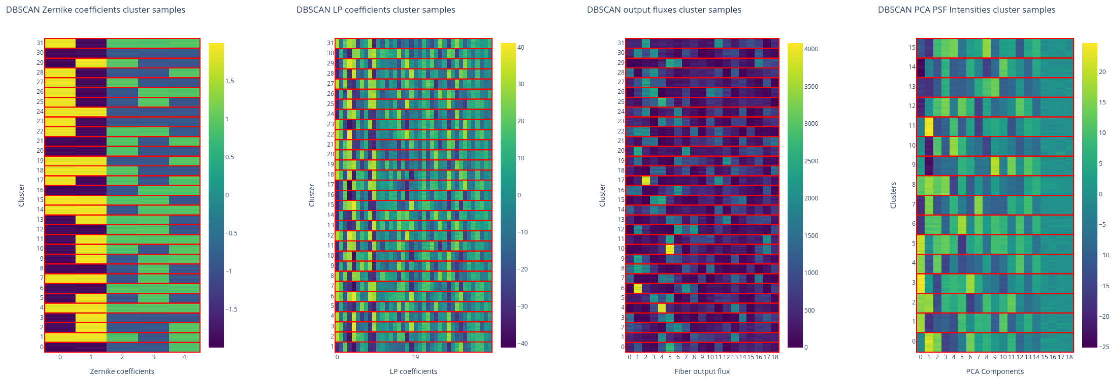


(c) Output fluxes



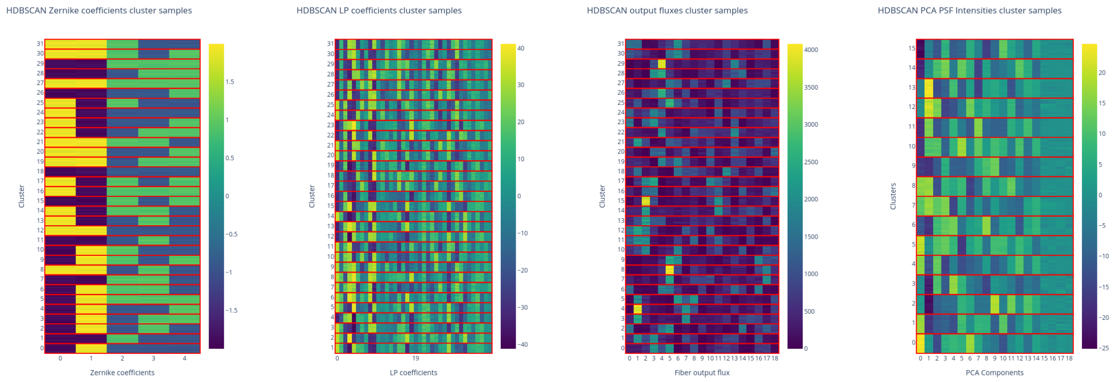
(d) PCA PSF Intensities

Figure 172: K-Means clusters from the datasets



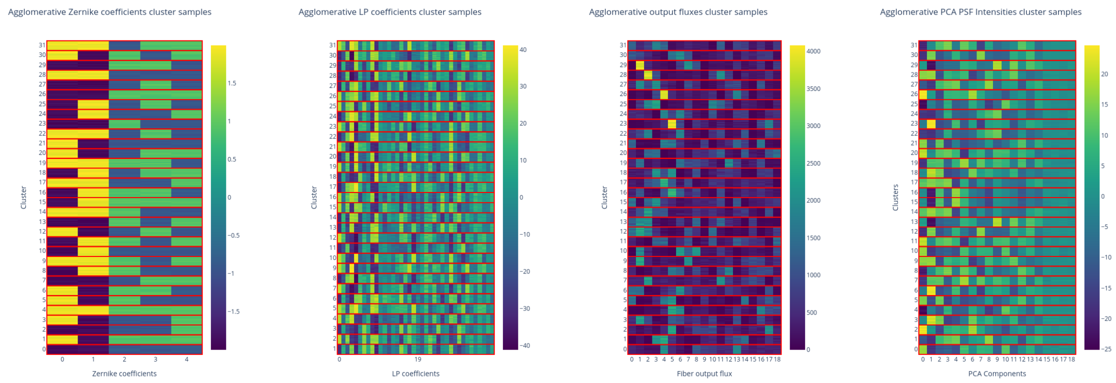
(a) Zernike coefficients (b) LP coefficients (c) Output fluxes (d) PCA PSF Intensities

Figure 173: DBSCAN clusters from the datasets



(a) Zernike coefficients (b) LP coefficients (c) Output fluxes (d) PCA PSF Intensities

Figure 174: HDBSCAN clusters from the datasets



(a) Zernike coefficients (b) LP coefficients (c) Output fluxes (d) PCA PSF Intensities

Figure 175: Agglomerative clusters from the datasets

Part VI

Mini Dataset 9 Zernike modes PL Information Determination

9 The data

9.1 Zernike coefficients dataset

A dataset of 6400 zernike coefficients is created for this report.In particular, each datapoint represent the coefficients of the first 5 Zernike modes, their values ranging between:

- The first 2 modes between[-2, -1.8] and [1.8, 2]
- Modes 4, 5 and 6 between [0.8, 1]
- Modes 7, 8, 9 and 10 between [-0.5, -0.3] and [0.3, 0.5]

These ranges create 64 original clusters that will be used as reference.

9.2 PSFs intensities dataset

A dataset of 6400 PSFs is created using the Zernike coefficients dataset.

9.3 LP mode coefficients dataset

A dataset of 6400 LP mode coefficients obtained from computing the overlap integral of the first 19 LP modes with the PSF dataset.

9.4 LP mode coefficients dataset

A dataset of 6400 PL output fluxes obtained from the PL transfer matrix and LP coefficients.

10 Preprocessing

10.1 PSF Intensities

The 6400x128x128 array is dimensionally reduced using PCA and UMAP both giving an array of 6400x19 projections of the PSF Intensities.

11 Clustering

A series of different clustering algorithms are used:

- K-Means
- DBSCAN

- HDBSCAN
- Agglomerative clustering

The clusters obtained will be compared the original clusters using NMI

11.1 Zernike coefficients clustering

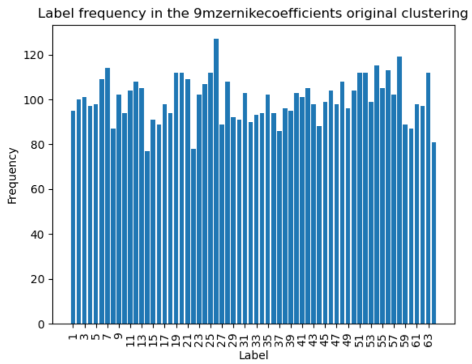
11.1.1 K-Means

As K-Means allows for the number of clusters to define, and we know that there are 4 in the original dataset, K-Means is used to find 4 clusters.

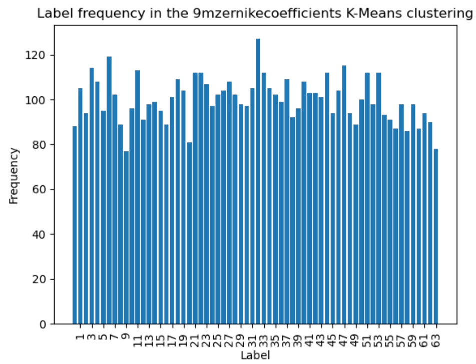
Number of clusters	**Number of initializations**
64	100

Table 57: K-Means hyperparameter configuration for Zernike coefficients clustering

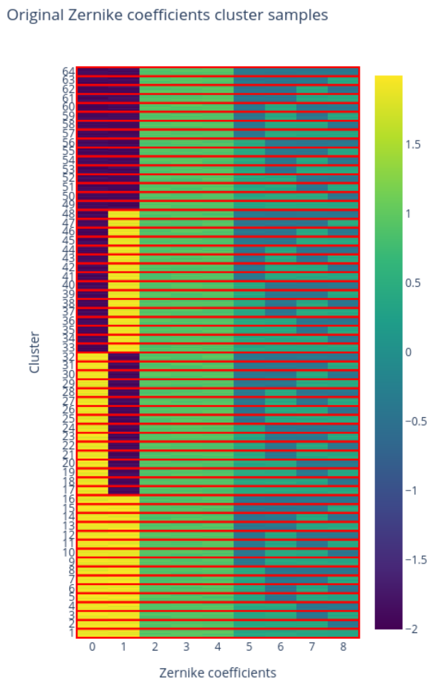
The results are the following:



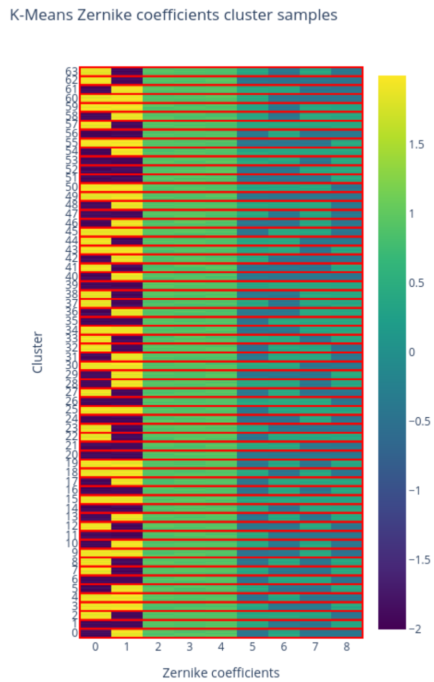
(a) Original cluster densities



(b) K-Means clusters densities



(c) Original cluster samples



(d) K-Means cluster samples

Figure 176: Comparison between original clustering and K-Means clustering

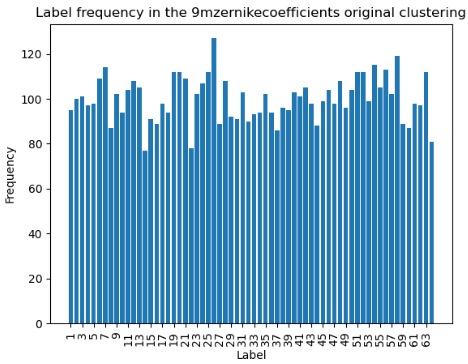
11.1.2 DBSCAN

A configuration that outputs 4 clusters is searched

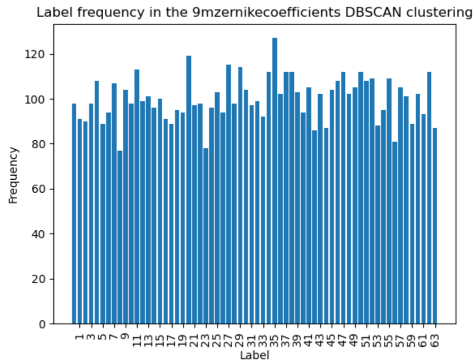
Number of neighbours	**Epsilon**
5	0.2

Table 58: DBSCAN hyperparameter configuration for Zernike coefficients clustering

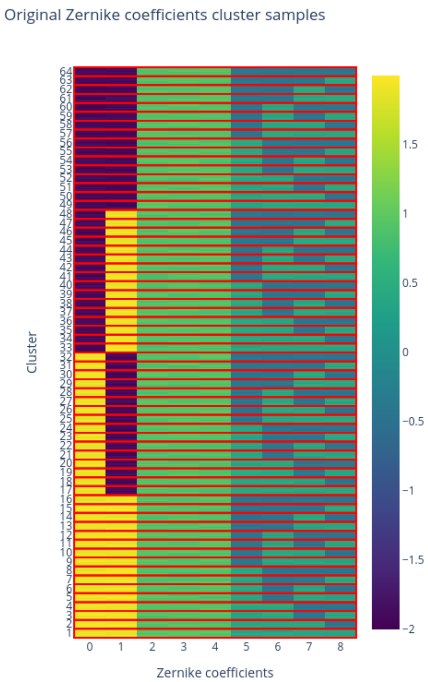
The results are the following:



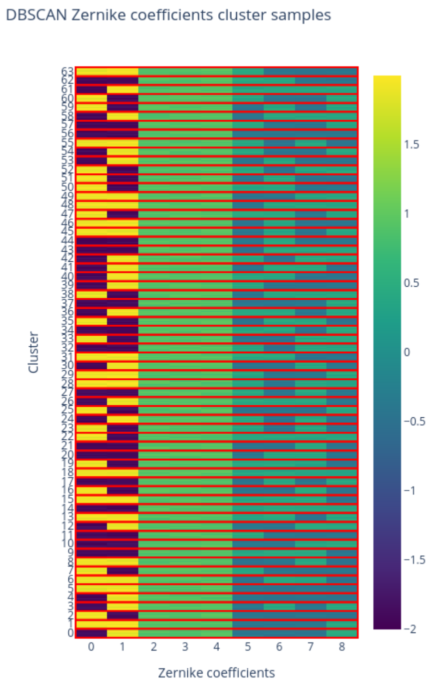
(a) Original cluster densities



(b) DBSCAN clusters densities



(c) Original cluster samples



(d) DBSCAN cluster samples

Figure 177: Comparison between original clustering and DBSCAN clustering

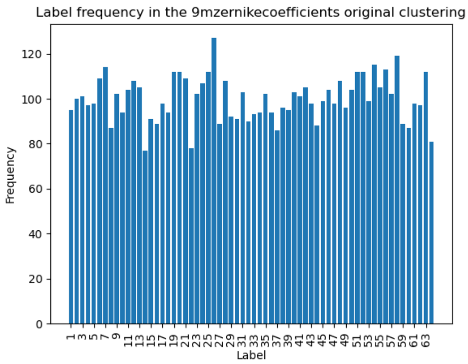
11.1.3 HDBSCAN

A configuration that outputs 4 clusters is searched.

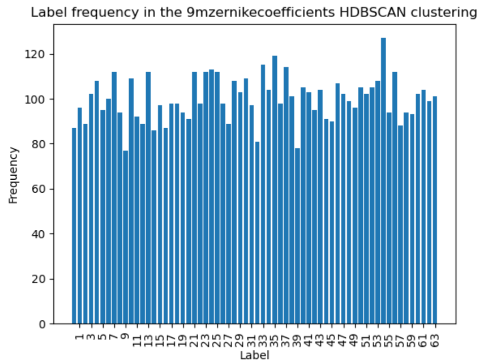
Minimum cluster size
50

Table 59: HDBSCAN hyperparameter configuration for Zernike coefficients clustering

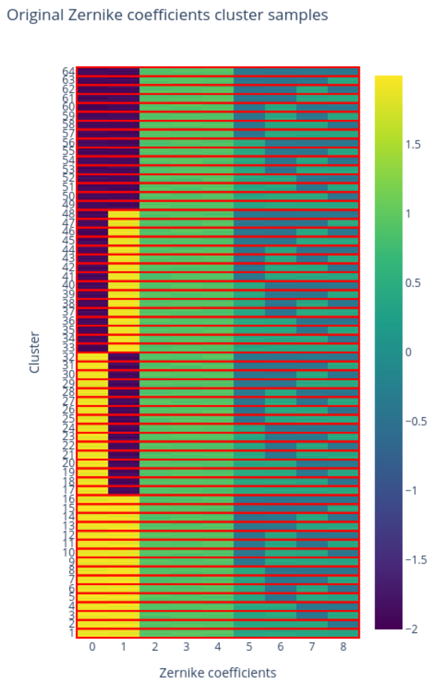
The results are the following:



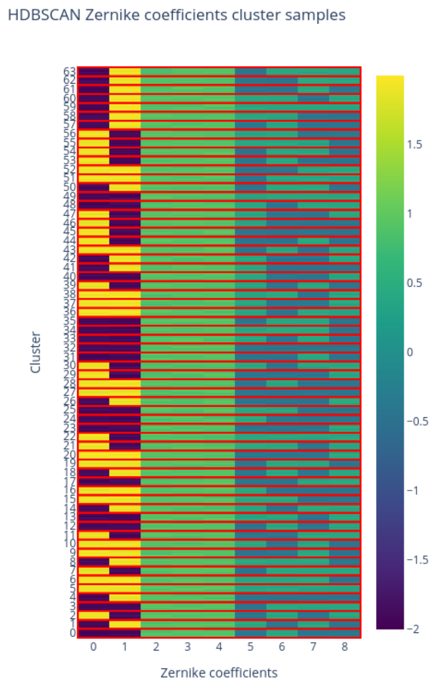
(a) Original cluster densities



(b) HDBSCAN clusters densities



(c) Original cluster samples



(d) HDBSCAN cluster samples

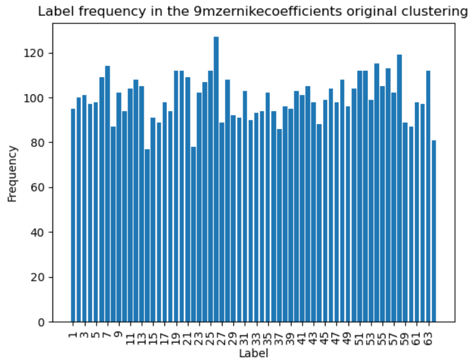
Figure 178: Comparison between original clustering and HDBSCAN clustering

11.1.4 Agglomerative clustering

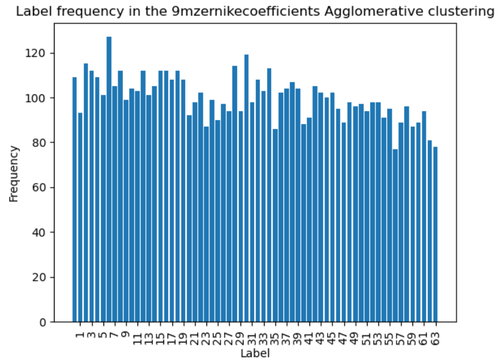
Number of clusters
64

Table 60: Agglomerative hyperparameter configuration for Zernike coefficients clustering

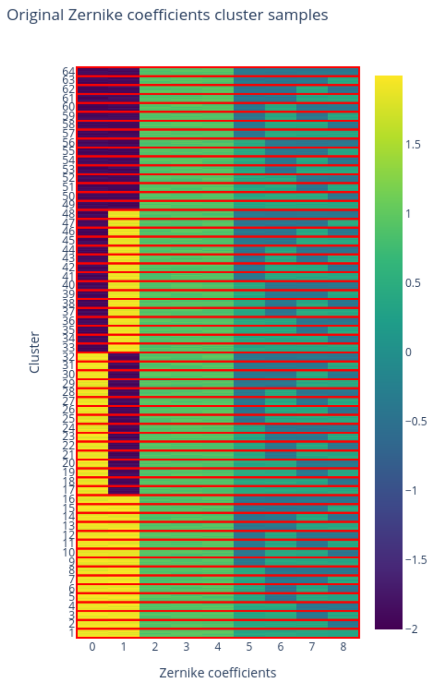
The results are the following:



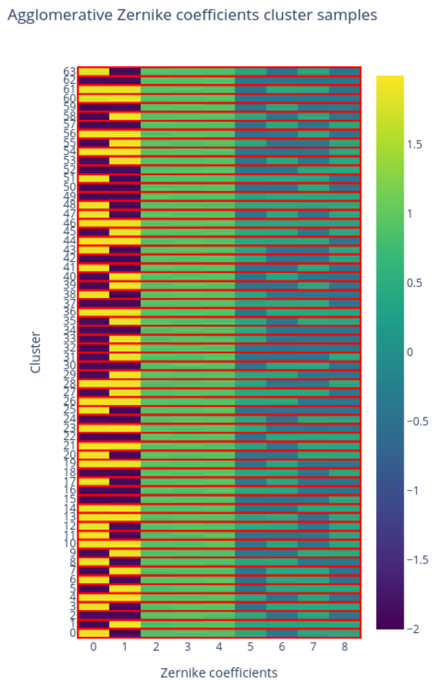
(a) Original cluster densities



(b) Agglomerative clusters densities



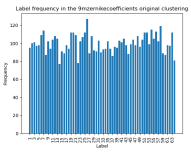
(c) Original cluster samples



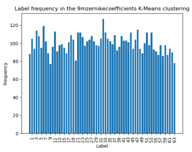
(d) Agglomerative cluster samples

Figure 179: Comparison between original clustering and Agglomerative clustering

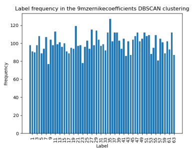
11.1.5 Summary



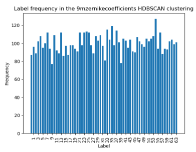
(a) Original cluster densities



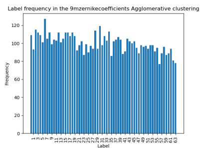
(b) K-means cluster densities



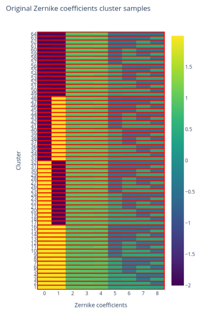
(c) DBSCAN cluster densities



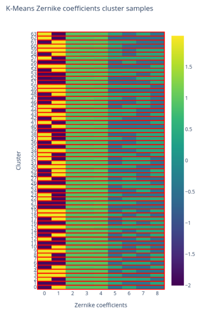
(d) HDBSCAN cluster densities



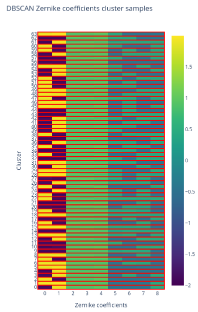
(e) Agglomerative cluster densities



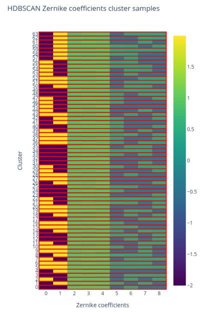
(f) Original cluster samples



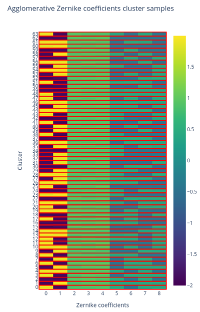
(g) K-means cluster samples



(h) DBSCAN cluster samples



(i) HDBSCAN cluster samples



(j) Agglomerative cluster samples

Figure 180: Comparison between clustering algorithms

	Original	**K-Means**	**DBSCAN**	**HDBSCAN**	**Agglomerative**
Original	\	1	1	1	1
K-Means		\	1	1	1
DBSCAN			\	1	1
HDBSCAN				\	1

Table 61: Normalized Mutual Information between clusters

11.2 LP coefficients clustering

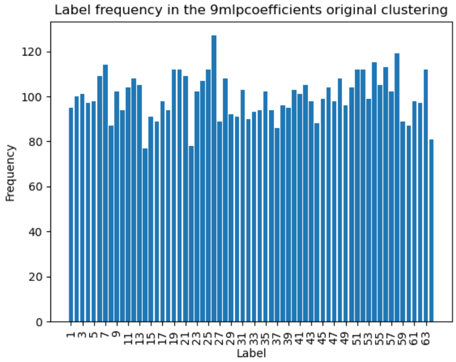
11.2.1 K-Means

As K-Means allows for the number of clusters to be defined, and we know that there are 4 in the original dataset, K-Means is used to find 4 clusters.

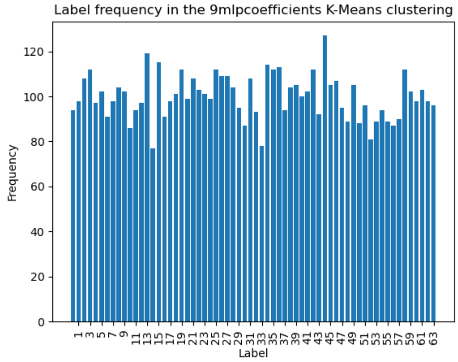
The results are the following:

	Number of clusters	**Number of initializations**
Original LP coefficients	32	100

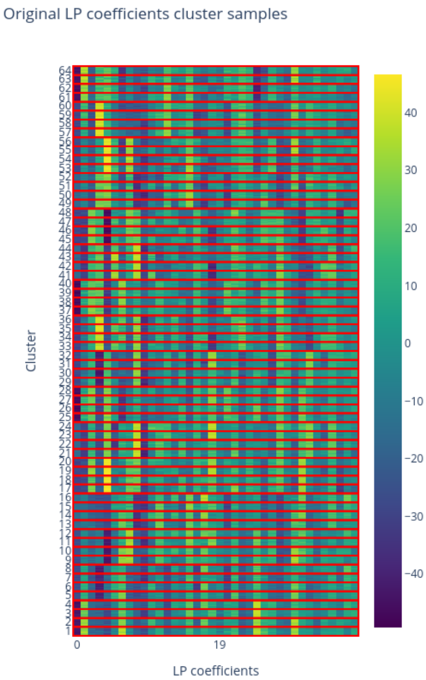
Table 62: K-Means hyperparameter configuration for c coefficients clustering



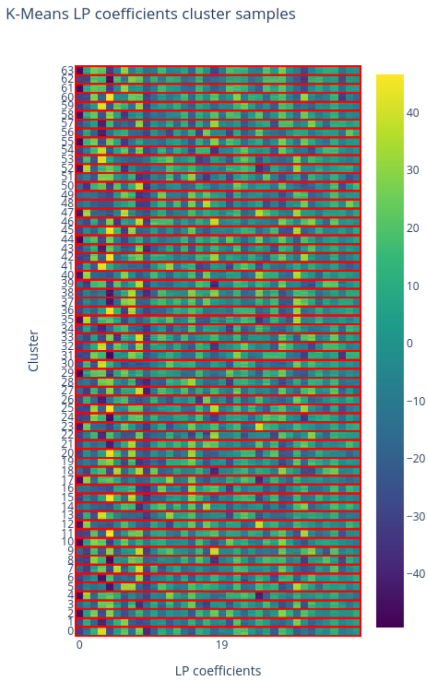
(a) Original cluster densities



(b) K-Means clusters densities



(c) Original cluster samples



(d) K-Means cluster samples

Figure 181: Comparison between original clustering and K-Means clustering from original LP coefficients

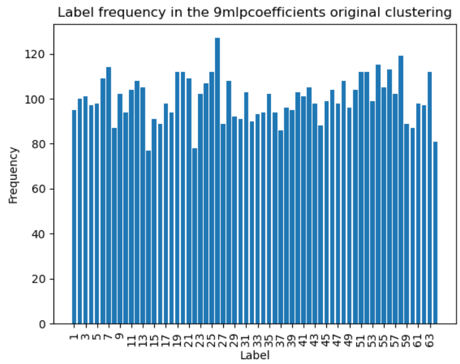
11.2.2 DBSCAN

A configuration that outputs 4 clusters is searched

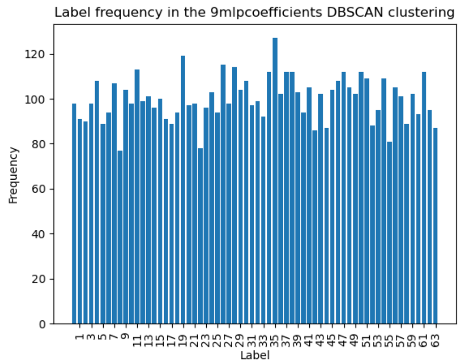
	Number of neighbours	**Epsilon**
Original LP coefficients	15	18

Table 63: DBSCAN hyperparameter configuration for LP coefficients clustering

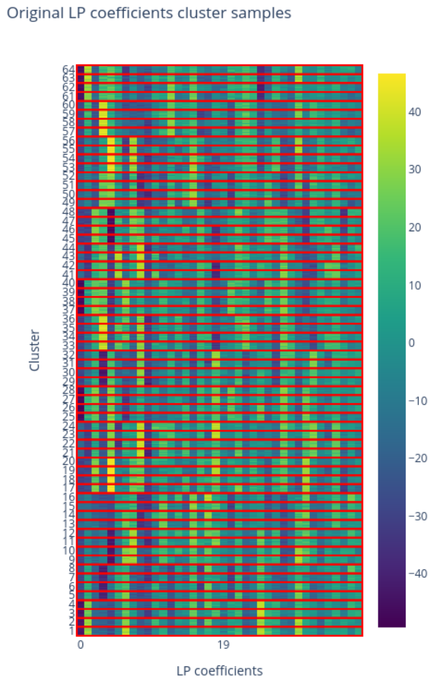
The results are the following:



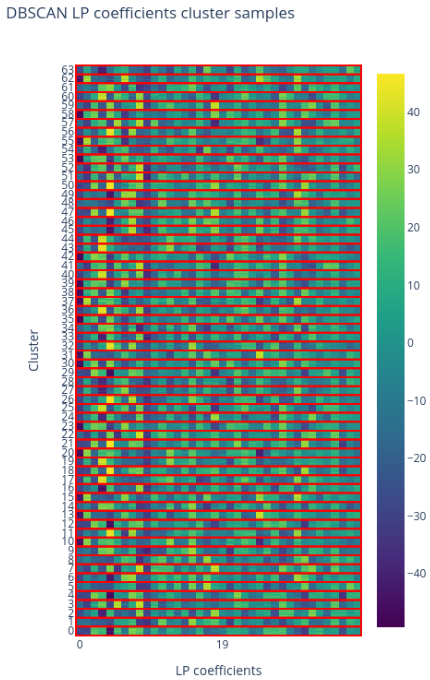
(a) Original cluster densities



(b) DBSCAN clusters densities



(c) Original cluster samples



(d) DBSCAN cluster samples

Figure 182: Comparison between original clustering and DBSCAN clustering

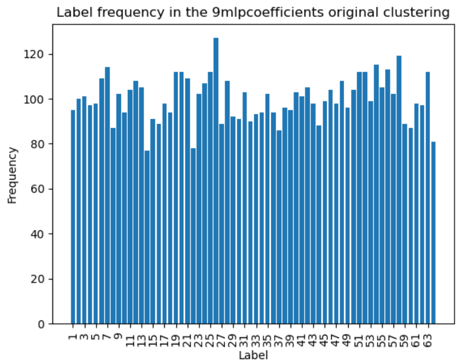
11.2.3 HDBSCAN

A configuration that outputs 4 clusters is searched.

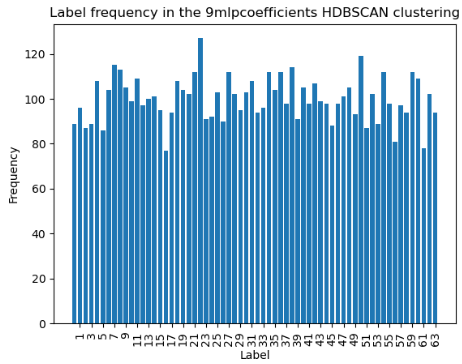
	Minimum cluster size
Original LP coefficients	21

Table 64: HDBSCAN hyperparameter configuration for LP coefficients clustering

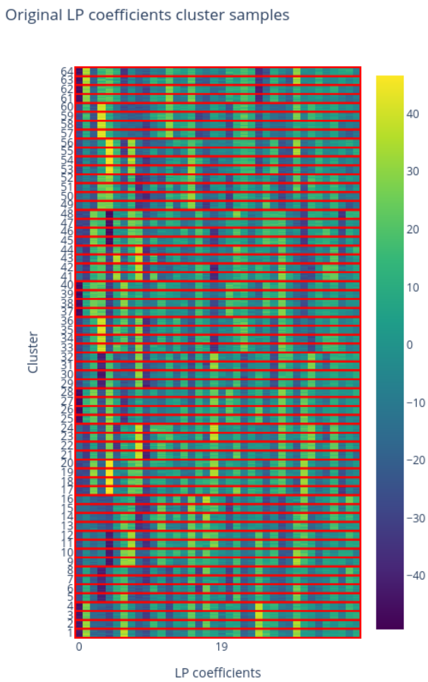
The results are the following:



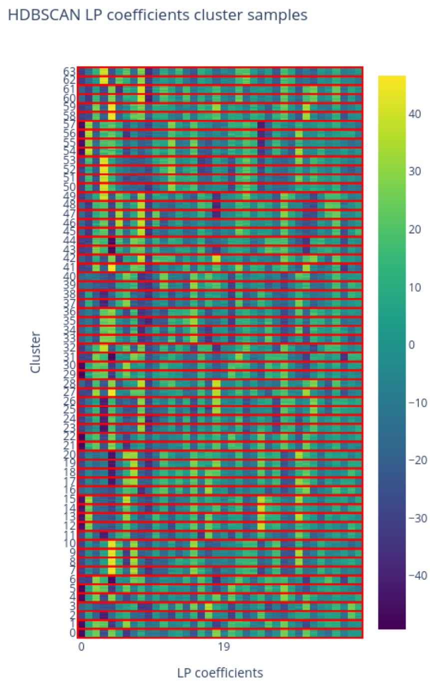
(a) Original cluster densities



(b) HDBSCAN clusters densities



(c) Original cluster samples



(d) HDBSCAN cluster samples

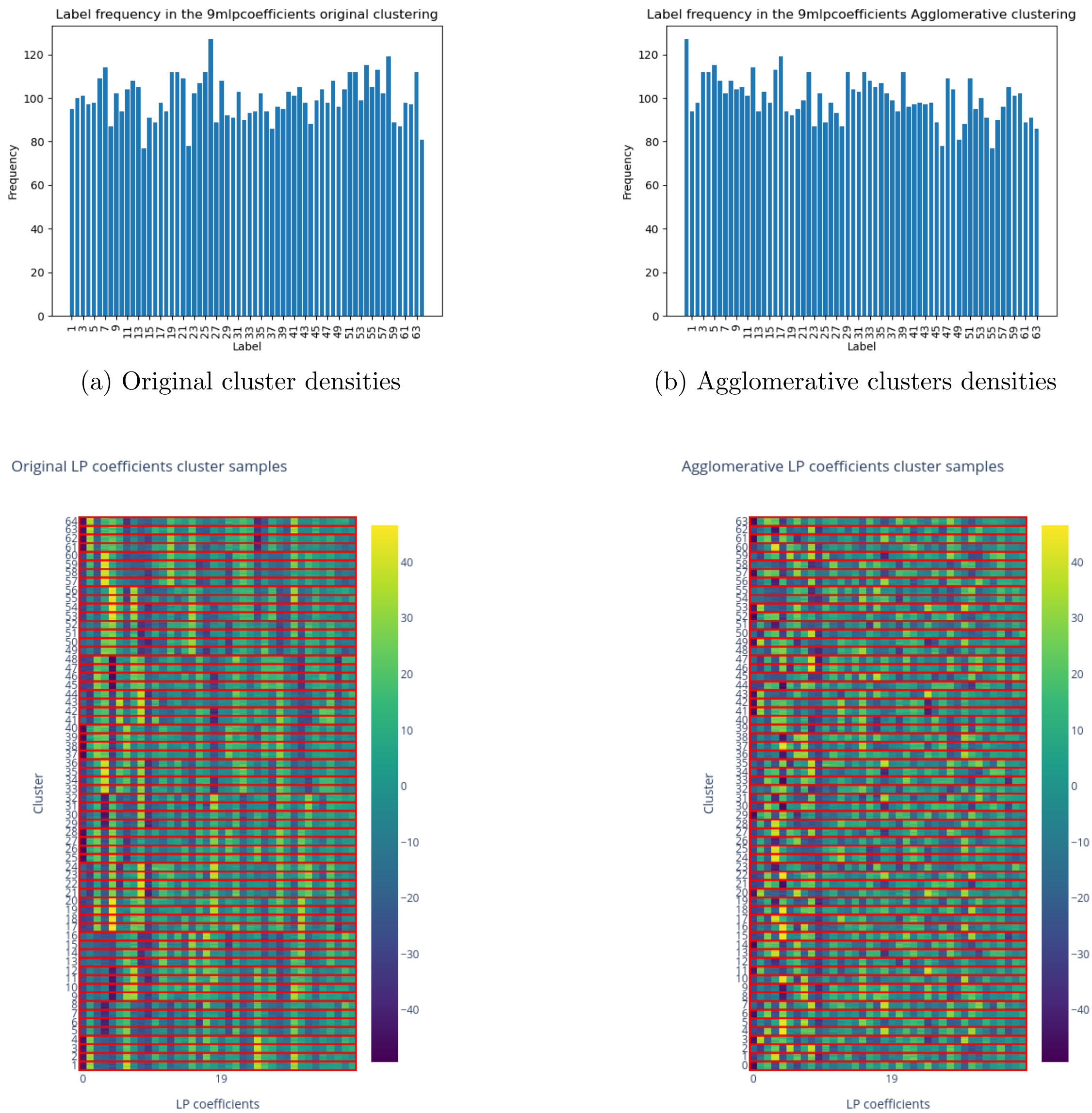
Figure 183: Comparison between original clustering and HDBSCAN clustering

11.2.4 Agglomerative clustering

	Number of clusters
Original LP coefficients	64

Table 65: Agglomerative hyperparameter configuration for LP coefficients clustering

The results are the following:



(a) Original cluster densities

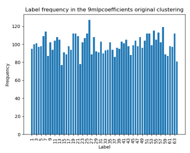
(b) Agglomerative clusters densities

(c) Original cluster samples

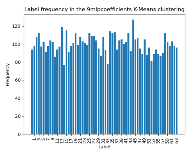
(d) Agglomerative cluster samples

Figure 184: Comparison between original clustering and Agglomerative clustering

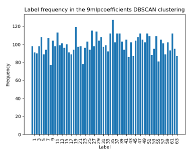
11.2.5 Summary



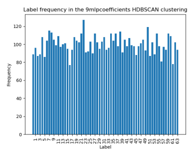
(a) Original cluster densities



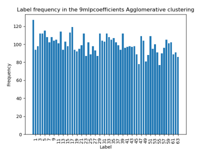
(b) K-means cluster densities



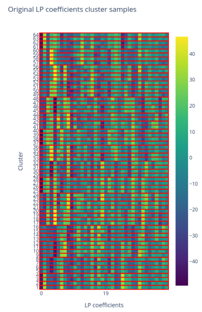
(c) DBSCAN cluster densities



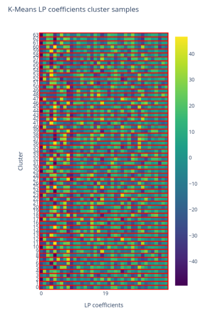
(d) HDBSCAN cluster densities



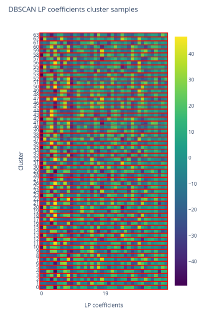
(e) Agglomerative cluster densities



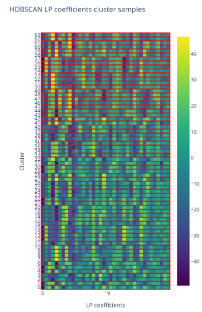
(f) Original cluster samples



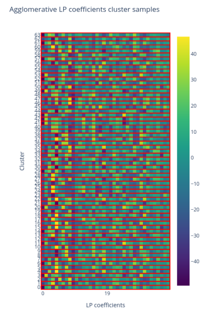
(g) K-means cluster samples



(h) DBSCAN cluster samples



(i) HDBSCAN cluster samples



(j) Agglomerative cluster samples

Figure 185: Comparison between clustering LP coefficients algorithms

	Original	**K-Means**	**DBSCAN**	**HDBSCAN**	**Agglomerative**
Original		1	1	1	1
K-Means			1	1	1
DBSCAN				1	1
HDBSCAN					1

Table 66: Normalized Mutual Information between original LP coefficients clusters

11.3 Output fluxes clustering

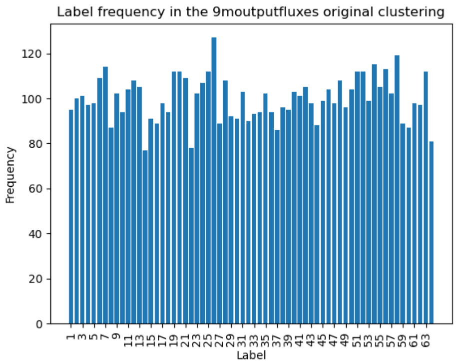
11.3.1 K-Means

As K-Means allows for the number of clusters to be defined, and we know that there are 4 in the original dataset, K-Means is used to find 4 clusters.

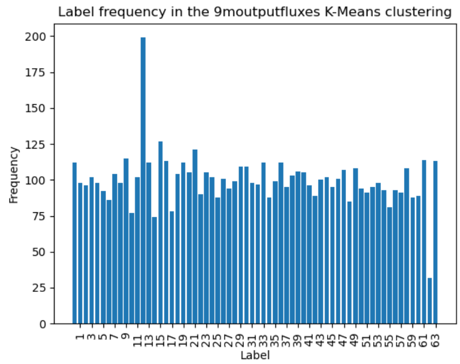
The results are the following:

	Number of clusters	**Number of initializations**
Original Output fluxes	64	100

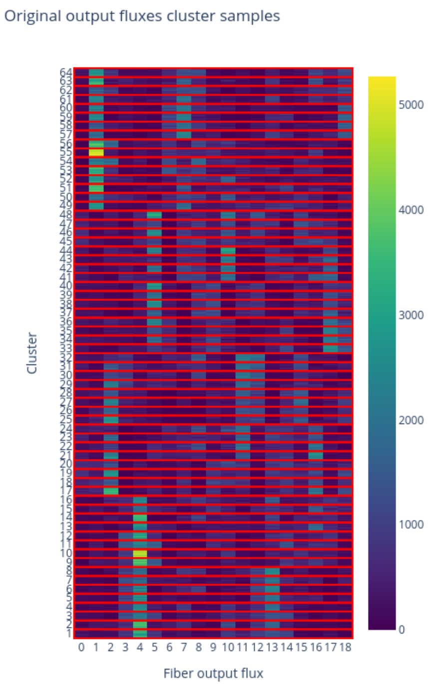
Table 67: K-Means hyperparameter configuration for c coefficients clustering



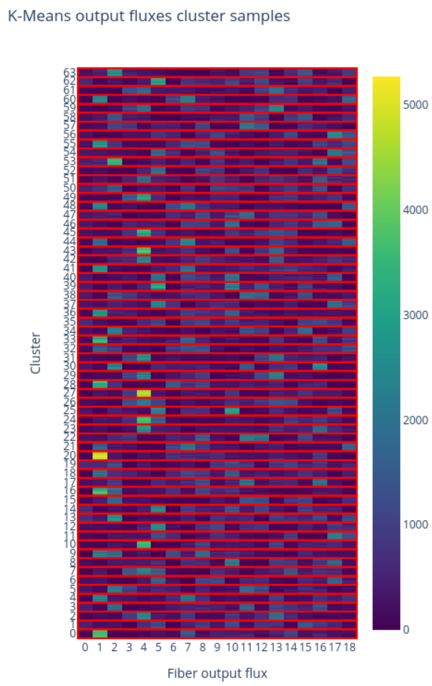
(a) Original cluster densities



(b) K-Means clusters densities



(c) Original cluster samples



(d) K-Means cluster samples

Figure 186: Comparison between original clustering and K-Means clustering from original Output fluxes

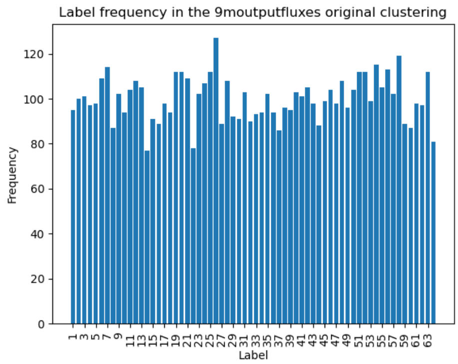
11.3.2 DBSCAN

A configuration that outputs 4 clusters is searched

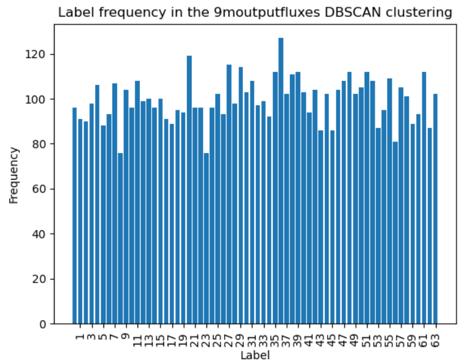
	Number of neighbours	**Epsilon**
Original Output fluxes	3	400

Table 68: DBSCAN hyperparameter configuration for Output fluxes clustering

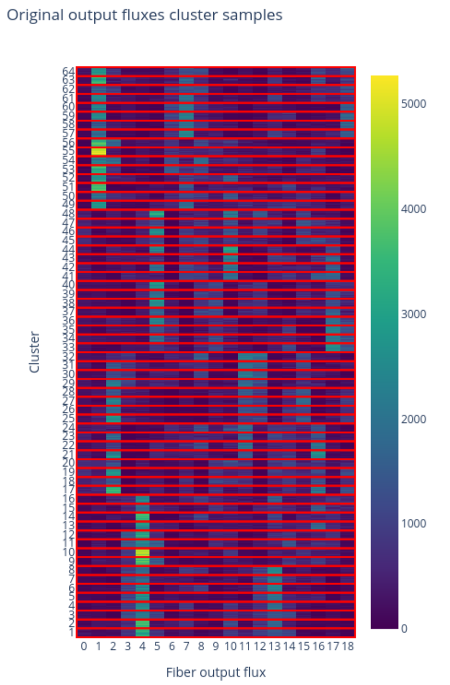
The results are the following:



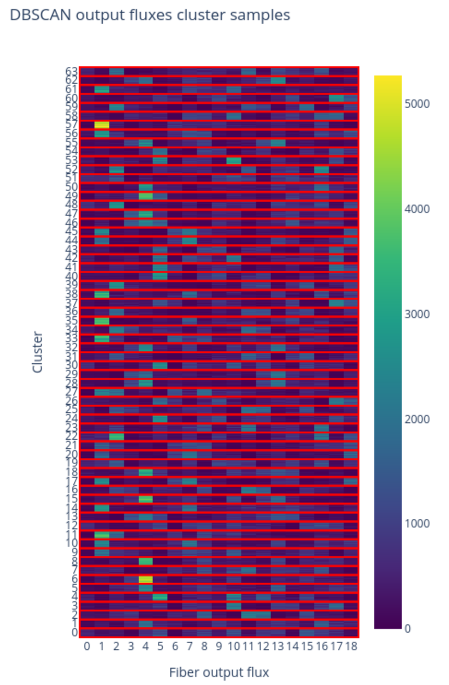
(a) Original cluster densities



(b) DBSCAN clusters densities



(c) Original cluster samples



(d) DBSCAN cluster samples

Figure 187: Comparison between original clustering and DBSCAN clustering

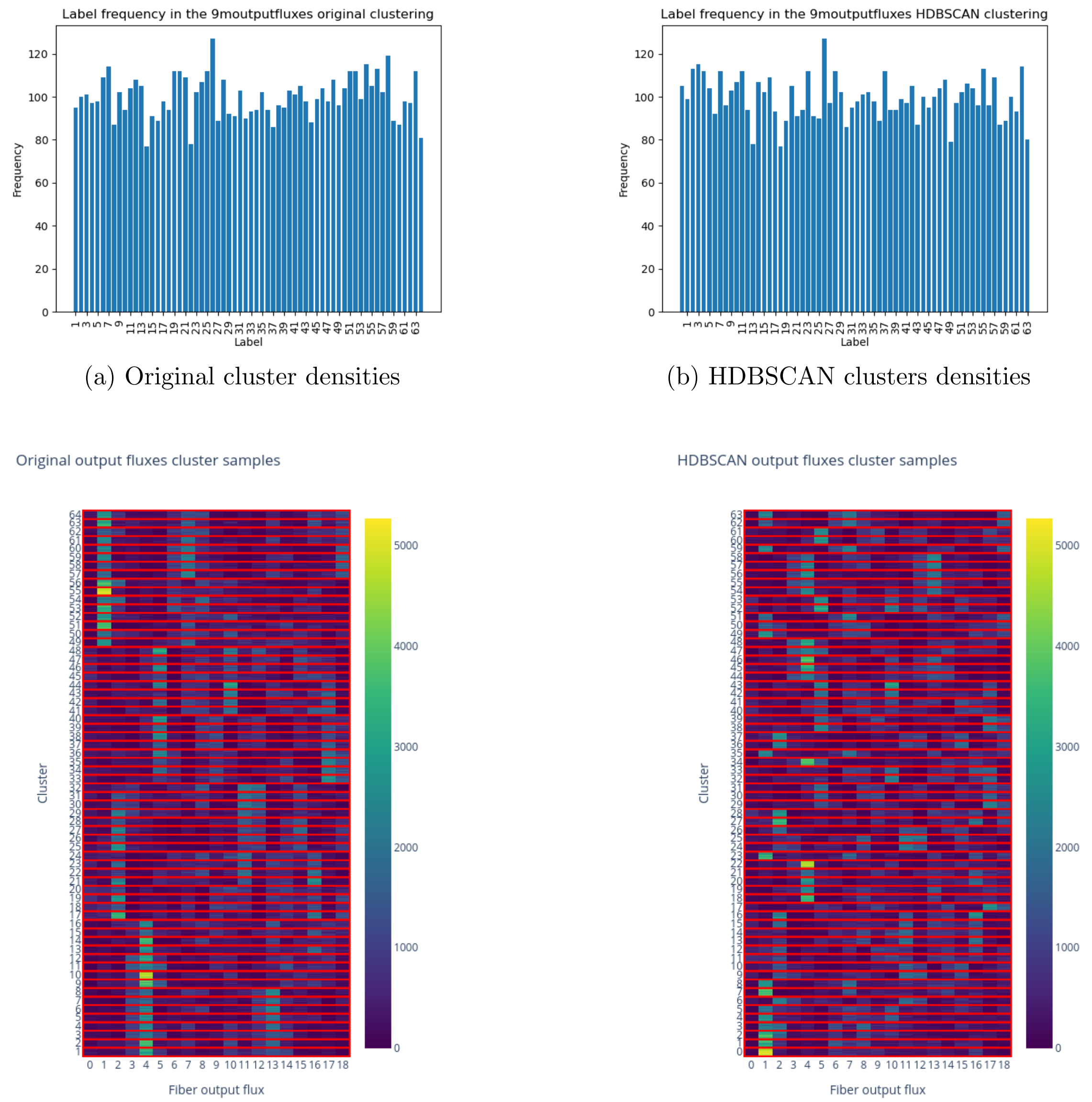
11.3.3 HDBSCAN

A configuration that outputs 4 clusters is searched.

	Minimum cluster size
Original Output fluxes	10

Table 69: HDBSCAN hyperparameter configuration for Output fluxes clustering

The results are the following:



(a) Original cluster densities

(b) HDBSCAN clusters densities

(c) Original cluster samples

(d) HDBSCAN cluster samples

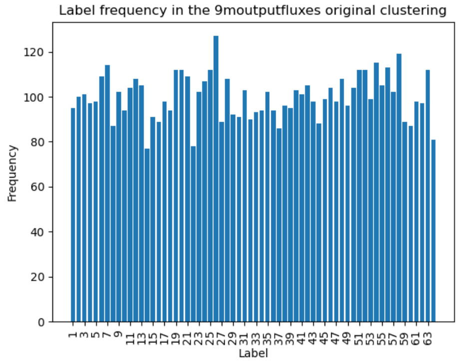
Figure 188: Comparison between original clustering and HDBSCAN clustering

11.3.4 Agglomerative clustering

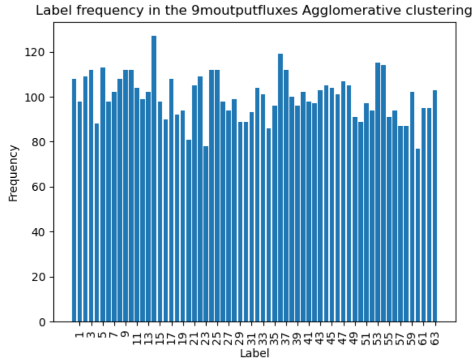
	Number of clusters
Original Output fluxes	64

Table 70: Agglomerative hyperparameter configuration for Output fluxes clustering

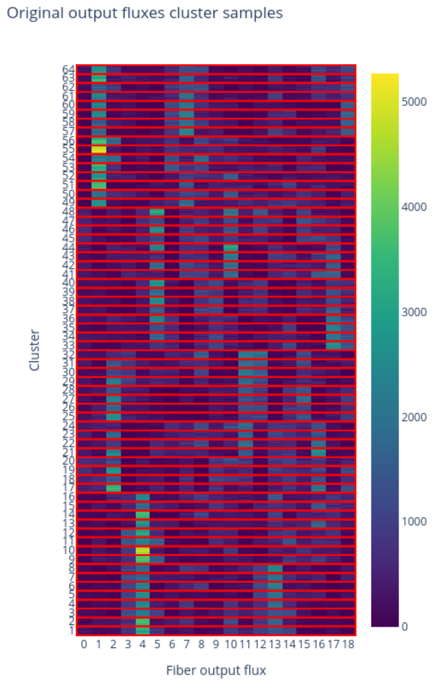
The results are the following:



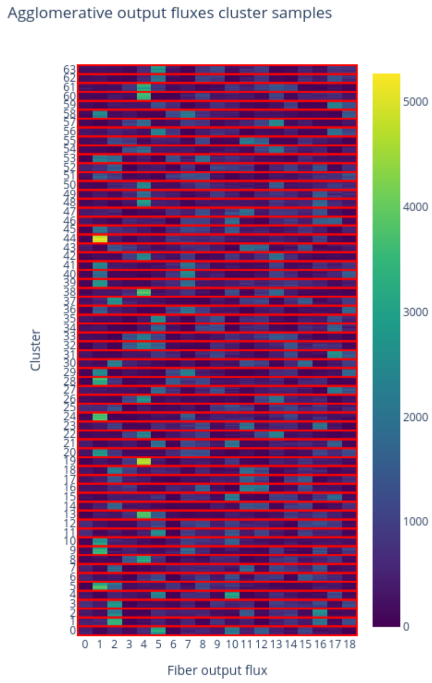
(a) Original cluster densities



(b) Agglomerative clusters densities



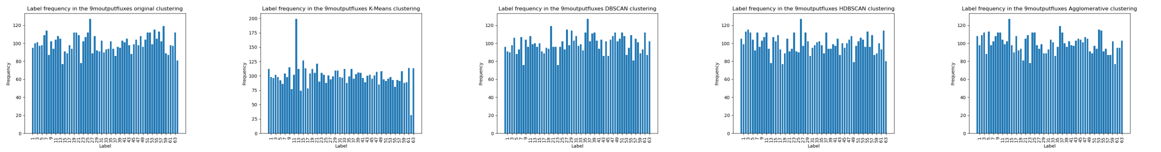
(c) Original cluster samples



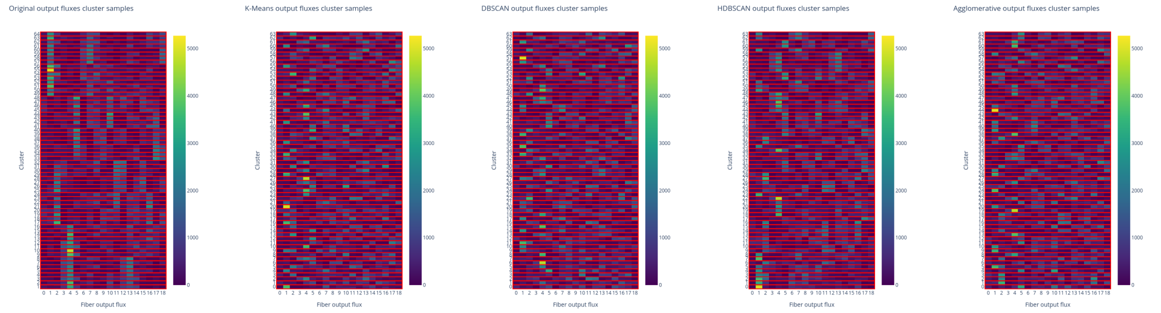
(d) Agglomerative cluster samples

Figure 189: Comparison between original clustering and Agglomerative clustering

11.3.5 Summary



(a) Original cluster densities (b) K-means cluster densities (c) DBSCAN cluster densities (d) HDBSCAN cluster densities (e) Agglomerative cluster densities



(f) Original cluster samples (g) K-means cluster samples (h) DBSCAN cluster samples (i) HDBSCAN cluster samples (j) Agglomerative cluster samples

Figure 190: Comparison between clustering Output fluxes algorithms

	Original	**K-Means**	**DBSCAN**	**HDBSCAN**	**Agglomerative**
Original		0.994	0.995	0.995	1
K-Means			0.990	0.991	0.994
DBSCAN				0.993	0.995
HDBSCAN					0.995

Table 71: Normalized Mutual Information between original Output fluxes clusters

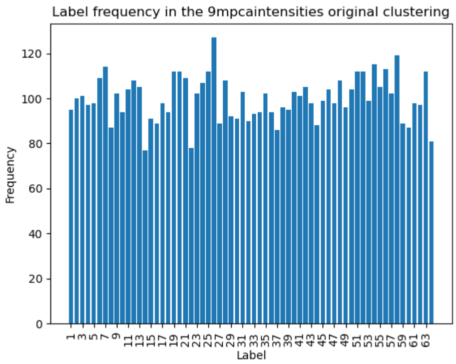
11.4 PSF Intensities clustering

11.4.1 K-Means

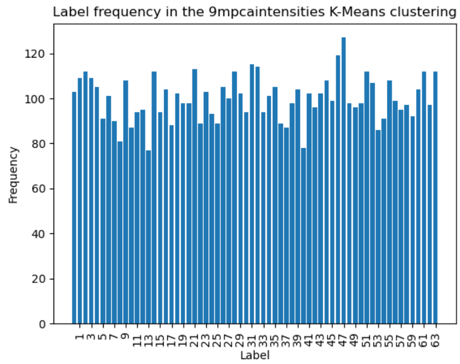
As K-Means allows for the number of clusters to be defined, and we know that there are 4 in the original dataset, K-Means is used to find 4 clusters.

	Number of clusters	**Number of initializations**
PCA PSF Intensities	64	100

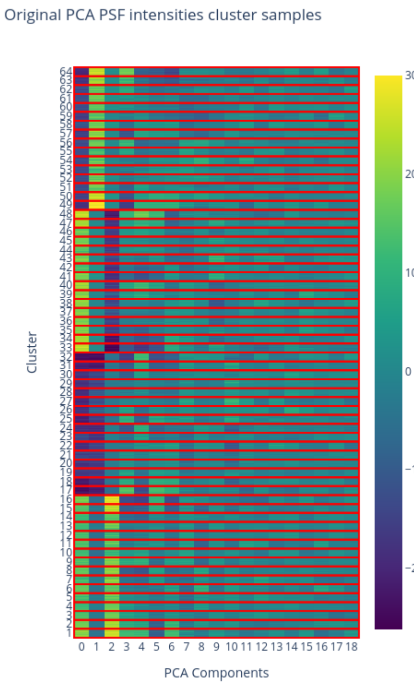
Table 72: K-Means hyperparameter configuration for c coefficients clustering



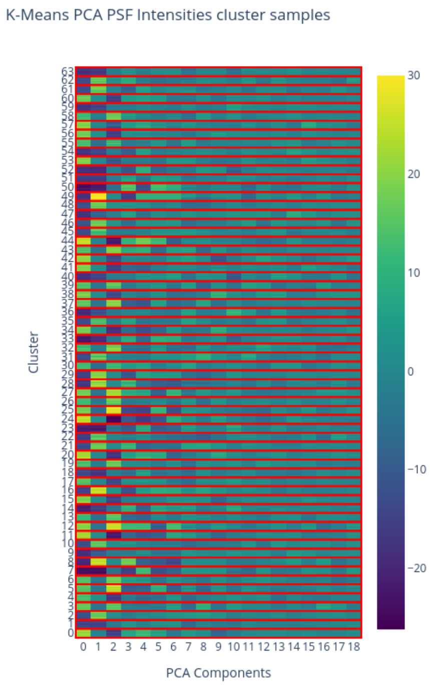
(a) Original cluster densities from PCA



(b) K-Means clusters densities from PCA



(c) Original cluster samples from PCA



(d) K-Means cluster samples from PCA

Figure 191: Comparison between original clustering and K-Means clustering from PCA of PSF Intensities

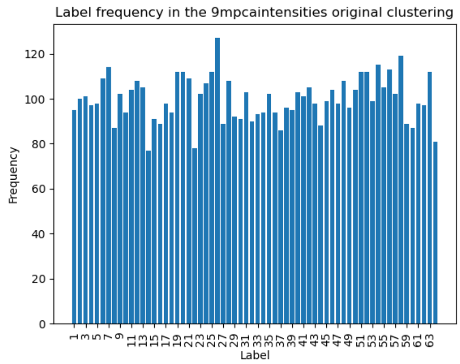
11.4.2 DBSCAN

A configuration that outputs 4 clusters is searched

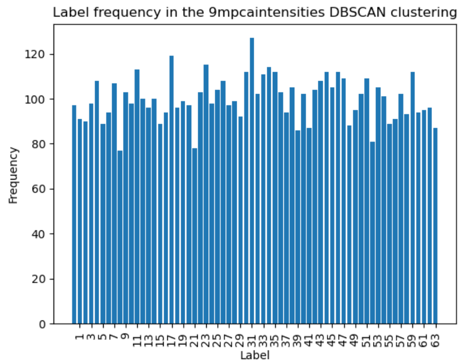
	Number of neighbours	**Epsilon**
PCA PSF Intensities	10	4

Table 73: DBSCAN hyperparameter configuration for PSF Intensities clustering

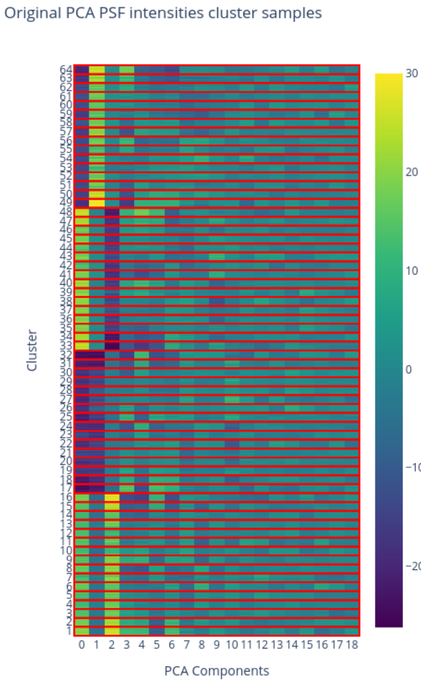
The results are the following:



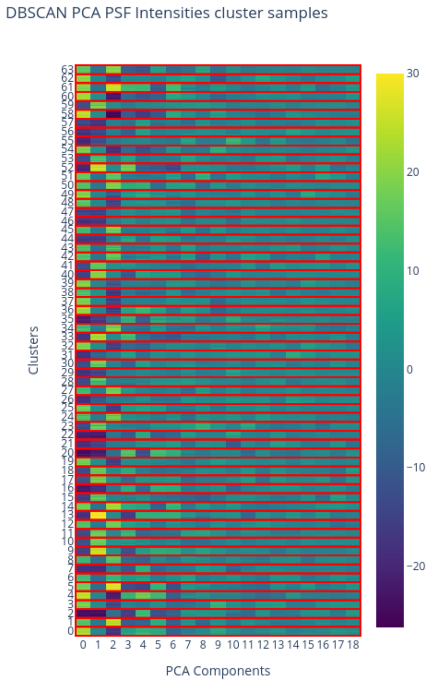
(a) Original cluster densities from PCA



(b) DBSCAN clusters densities from PCA



(c) Original cluster samples from PCA



(d) DBSCAN cluster samples from PCA

Figure 192: Comparison between original clustering and DBSCAN clustering from PCA of PSF Intensities

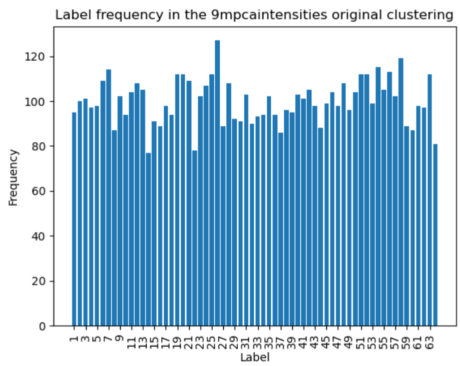
11.4.3 HDBSCAN

A configuration that outputs 4 clusters is searched.

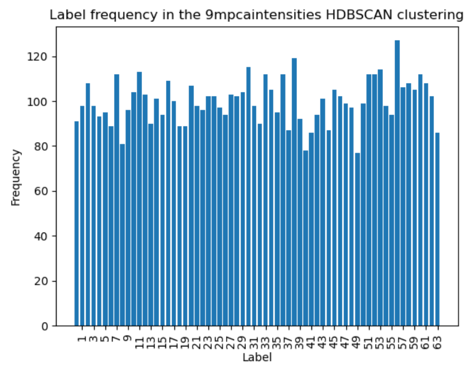
	Minimum cluster size
PCA PSF Intensities	15

Table 74: HDBSCAN hyperparameter configuration for PSF Intensities clustering

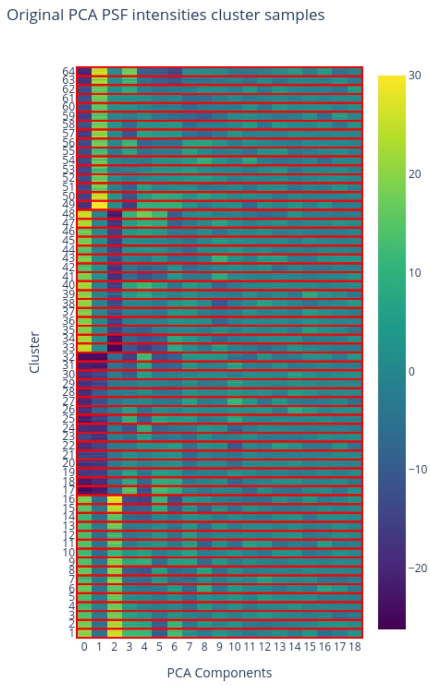
The results are the following:



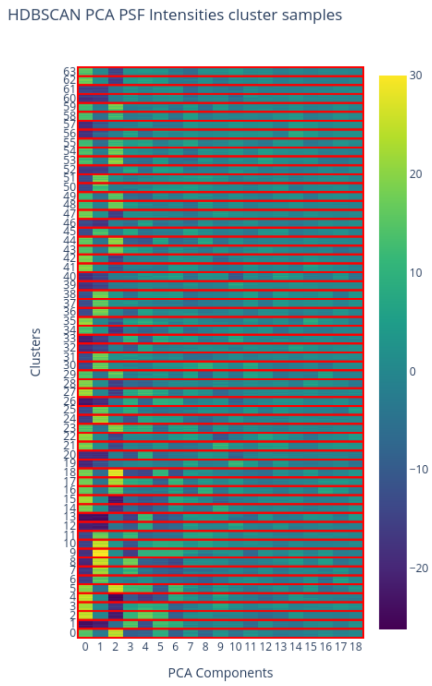
(a) Original cluster densities from PCA



(b) HDBSCAN clusters densities from PCA



(c) Original cluster samples from PCA



(d) HDBSCAN cluster samples from PCA

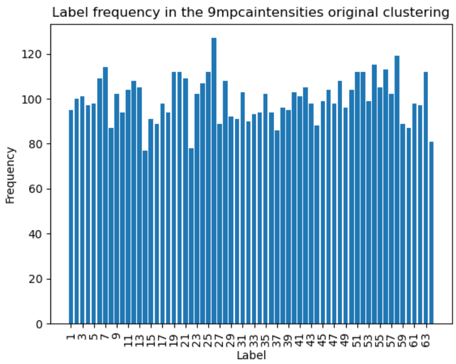
Figure 193: Comparison between original clustering and HDBSCAN clustering from PCA of PSF Intensities

11.4.4 Agglomerative clustering

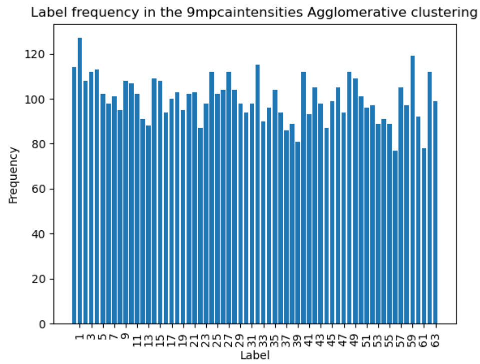
	Number of clusters
PCA PSF Intensities	64

Table 75: Agglomerative hyperparameter configuration for PSF Intensities clustering

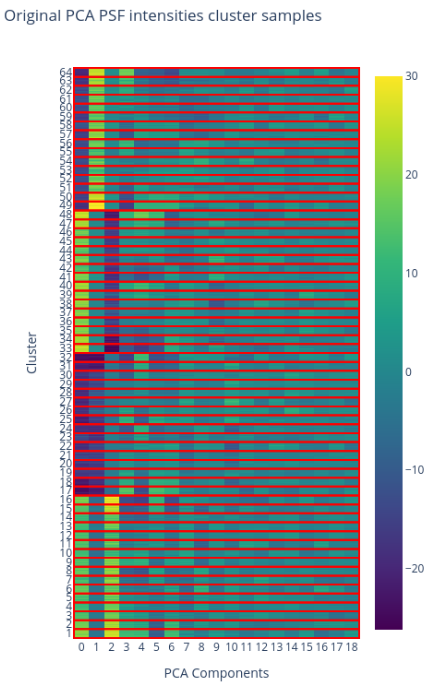
The results are the following:



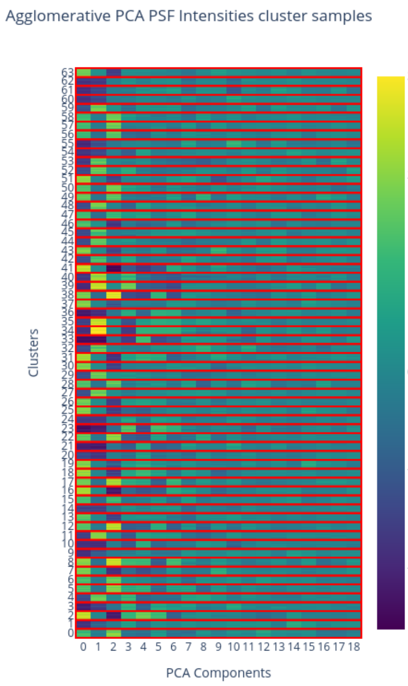
(a) Original cluster densities from PCA



(b) Agglomerative clusters densities from PCA



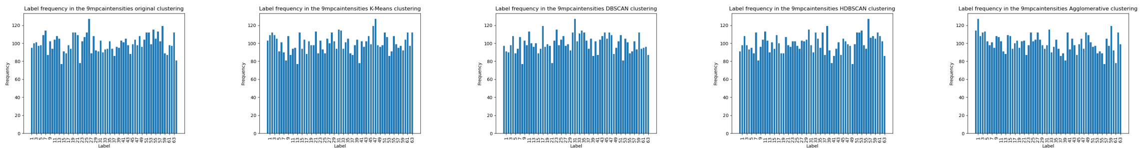
(c) Original cluster samples from PCA



(d) Agglomerative cluster samples from PCA

Figure 194: Comparison between original clustering and Agglomerative clustering

11.4.5 Summary



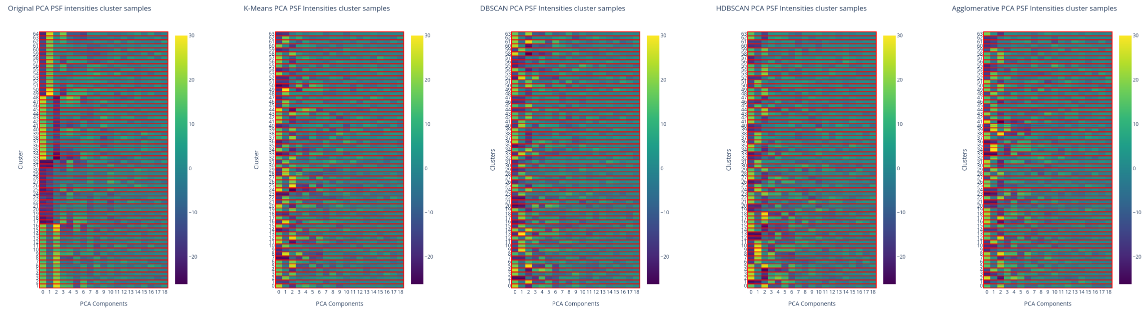
(a) Original cluster densities from PCA

(b) K-means cluster densities from PCA

(c) DBSCAN cluster densities from PCA

(d) HDBSCAN cluster densities from PCA

(e) Agglomerative cluster densities from PCA



(f) Original cluster samples from PCA

(g) K-means cluster samples from PCA

(h) DBSCAN cluster samples-from PCA

(i) HDBSCAN cluster samples from PCA

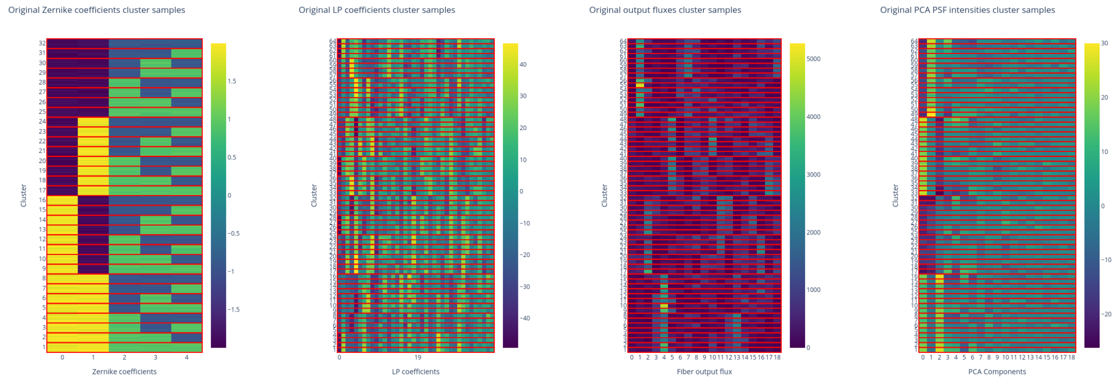
(j) Agglomerative cluster samples from PCA

Figure 195: Comparison between clustering PCA PSF Intensities algorithms

	Original	**K-Means**	**DBSCAN**	**HDBSCAN**	**Agglomerative**
Original	⟍	0.802	0.889	0.889	0.803
K-Means		⟍	0.891	0.891	0.909
DBSCAN			⟍	1	0.892
HDBSCAN				⟍	0.892

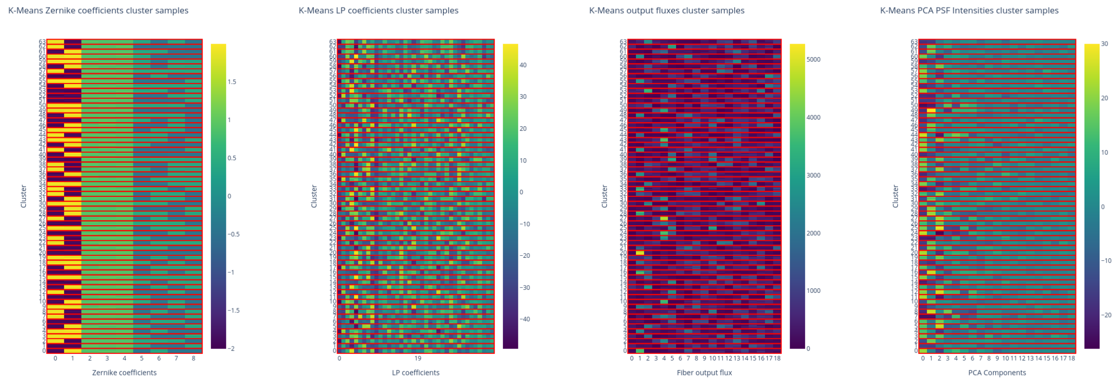
Table 76: Normalized Mutual Information between PCA PSF Intensities clusters

12 Dataset clusters comparison



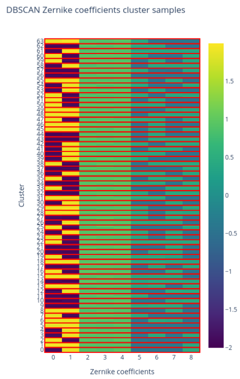
(a) Zernike coefficients (b) LP coefficients (c) Output fluxes (d) PCA PSF Intensities

Figure 196: Original clusters from the datasets

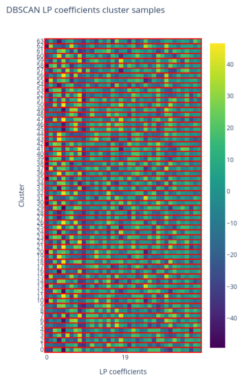


(a) Zernike coefficients (b) LP coefficients (c) Output fluxes (d) PCA PSF Intensities

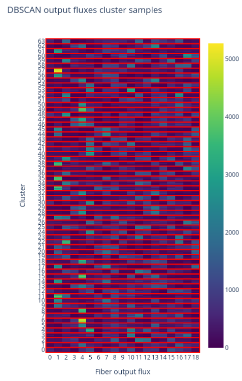
Figure 197: K-Means clusters from the datasets



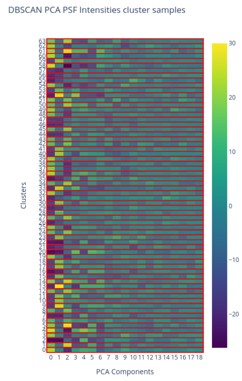
(a) Zernike coefficients



(b) LP coefficients

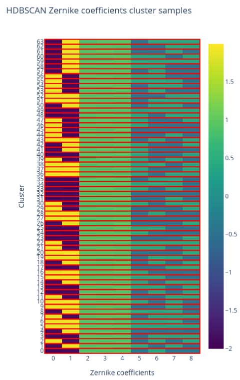


(c) Output fluxes

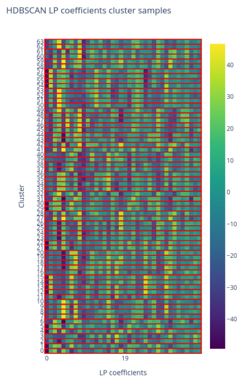


(d) PCA PSF Intensities

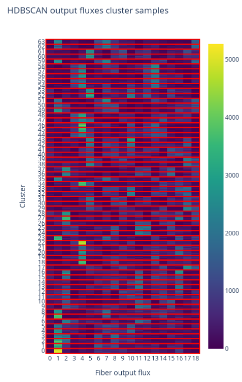
Figure 198: DBSCAN clusters from the datasets



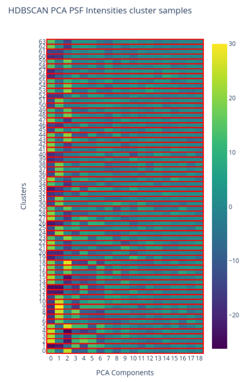
(a) Zernike coefficients



(b) LP coefficients

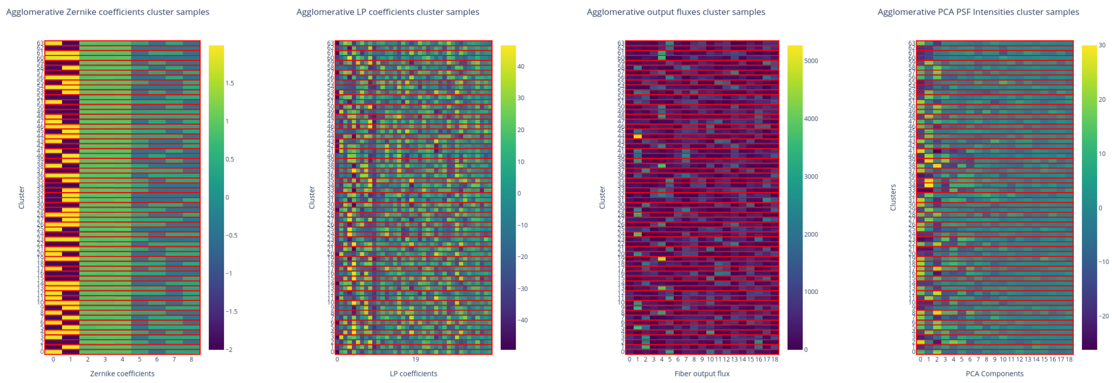


(c) Output fluxes



(d) PCA PSF Intensities

Figure 199: HDBSCAN clusters from the datasets



(a) Zernike coefficients (b) LP coefficients (c) Output fluxes (d) PCA PSF Intensities

Figure 200: Agglomerative clusters from the datasets

Part VII

Mini Dataset 14 Zernike modes PL Information Determination

13 The data

13.1 Zernike coefficients dataset

A dataset of 12800 zernike coefficients is created for this report.In particular, each datapoint represent the coefficients of the first 5 Zernike modes, their values ranging between:

- The first 2 modes between[-2, -1.8] and [1.8, 2]
- Modes 4, 5 and 6 between [0.8, 1]
- Modes 7, 8, 9 and 10 between [0.3, 0.5]

- Modes 11, 12, 13, 14 and 15 between [-0.5, -0.3] and [0.3, 0.5]

These ranges create 128 original clusters that will be used as reference.

13.2 PSFs intensities dataset

A dataset of 12800 PSFs is created using the Zernike coefficients dataset.

13.3 LP mode coefficients dataset

A dataset of 12800 LP mode coefficients obtained from computing the overlap integral of the first 19 LP modes with the PSF dataset.

13.4 LP mode coefficients dataset

A dataset of 12800 PL output fluxes obtained from the PL transfer matrix and LP coefficients.

14 Preprocessing

14.1 PSF Intensities

The 12800x128x128 array is dimensionally reduced using PCA and UMAP both giving an array of 12800x19 projections of the PSF Intensities.

15 Clustering

A series of different clustering algorithms are used:

- K-Means

- DBSCAN
- HDBSCAN
- Agglomerative clustering

The clusters obtained will be compared the original clusters using NMI

15.1 Zernike coefficients clustering

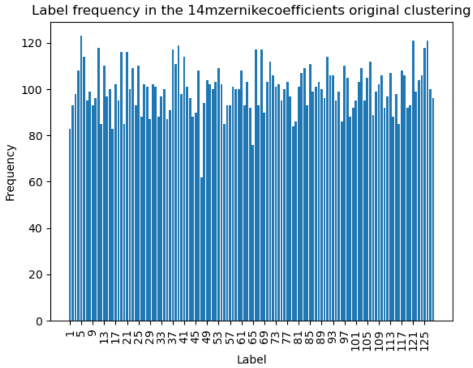
15.1.1 K-Means

As K-Means allows for the number of clusters to define, and we know that there are 4 in the original dataset, K-Means is used to find 128 clusters.

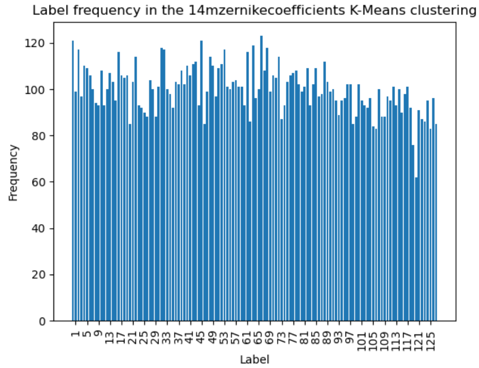
Number of clusters	**Number of initializations**
128	100

Table 77: K-Means hyperparameter configuration for Zernike coefficients clustering

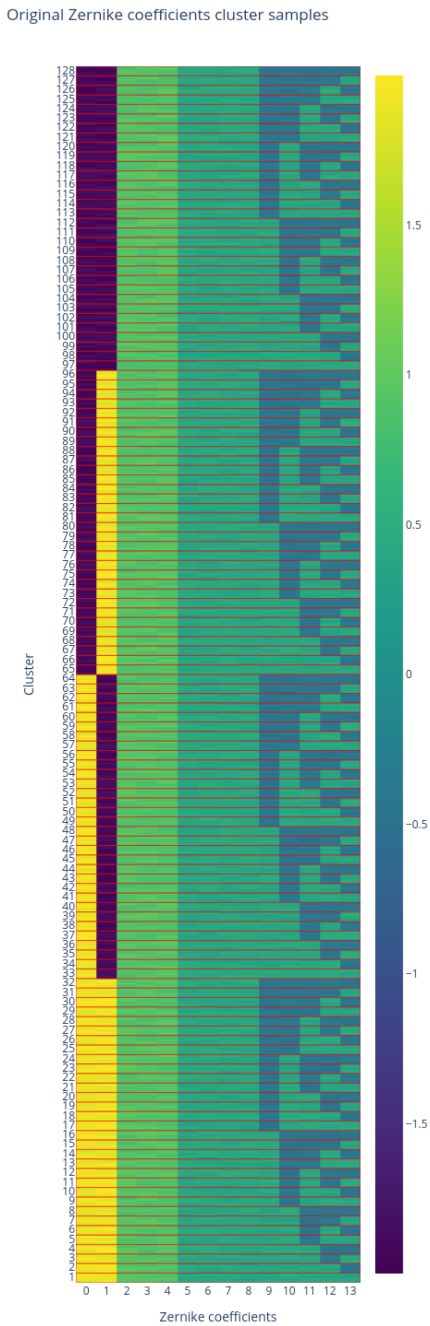
The results are the following:



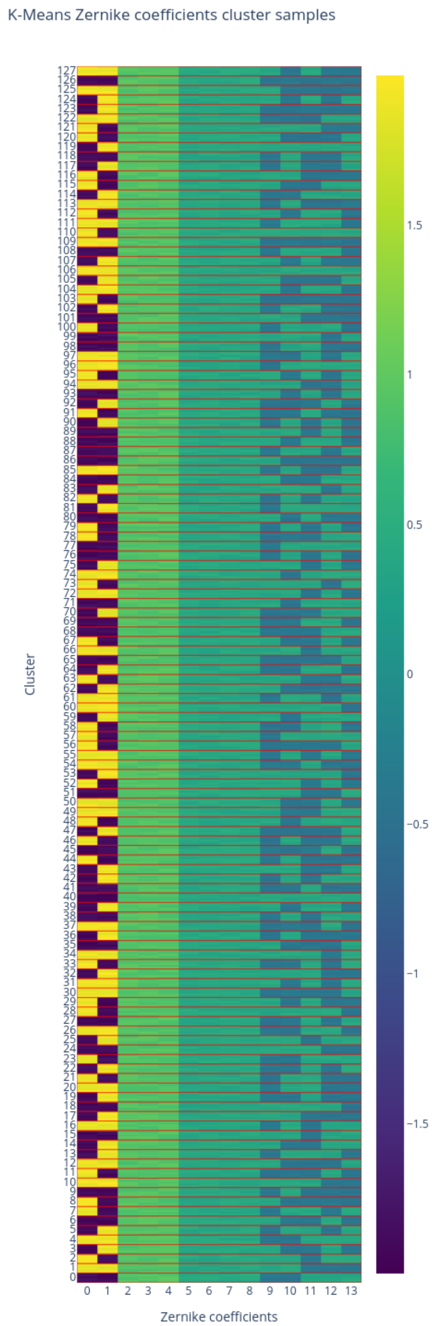
(a) Original cluster densities



(b) K-Means clusters densities



(c) Original cluster samples



(d) K-Means cluster samples

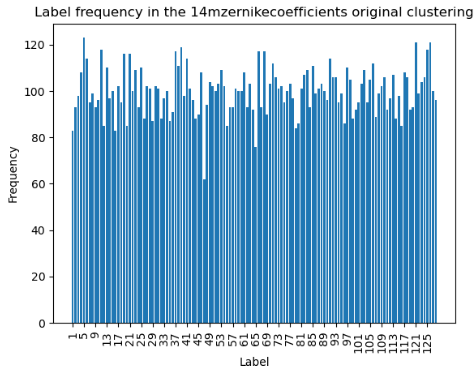
15.1.2 DBSCAN

A configuration that outputs 4 clusters is searched

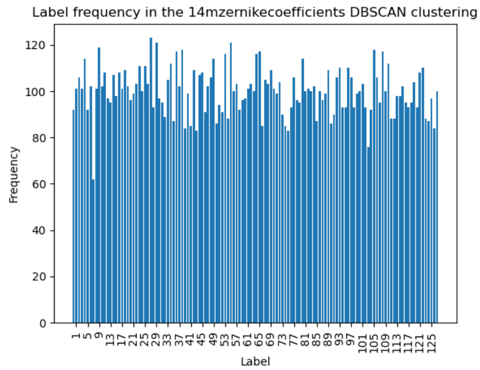
Number of neighbours	**Epsilon**
5	0.26

Table 78: DBSCAN hyperparameter configuration for Zernike coefficients clustering

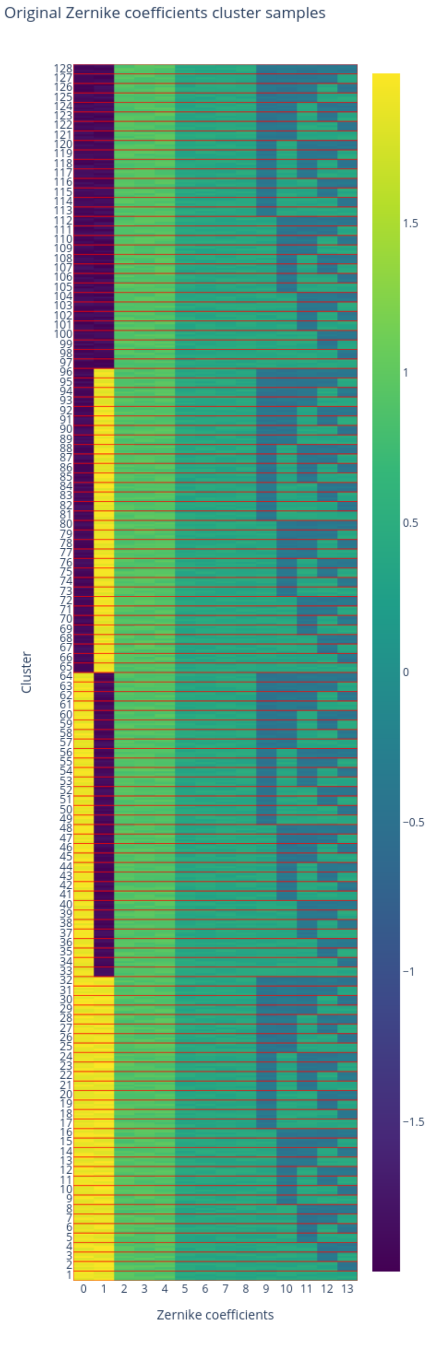
The results are the following:



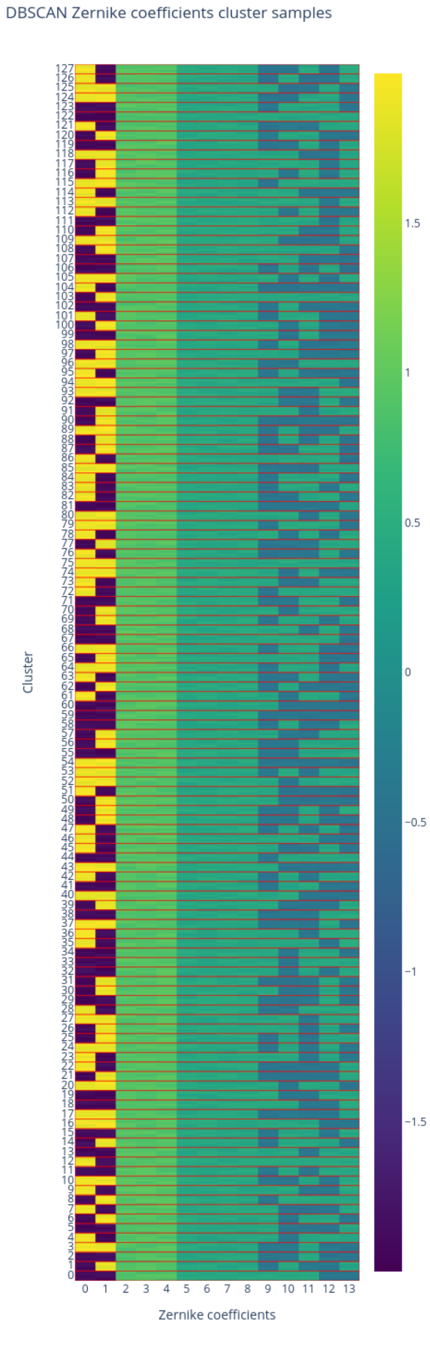
(a) Original cluster densities



(b) DBSCAN clusters densities



(c) Original cluster samples



(d) DBSCAN cluster samples

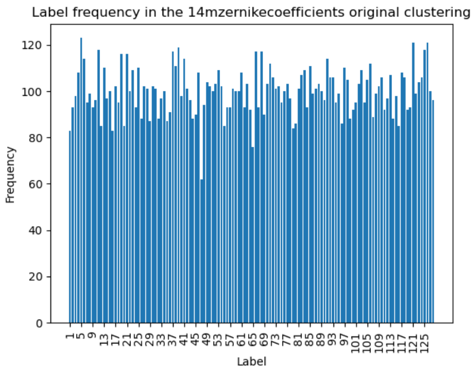
15.1.3 HDBSCAN

A configuration that outputs 4 clusters is searched.

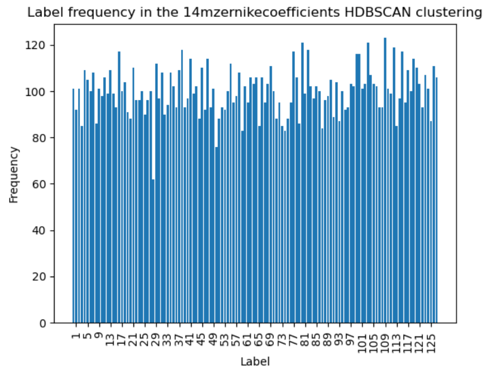
Minimum cluster size
50

Table 79: HDBSCAN hyperparameter configuration for Zernike coefficients clustering

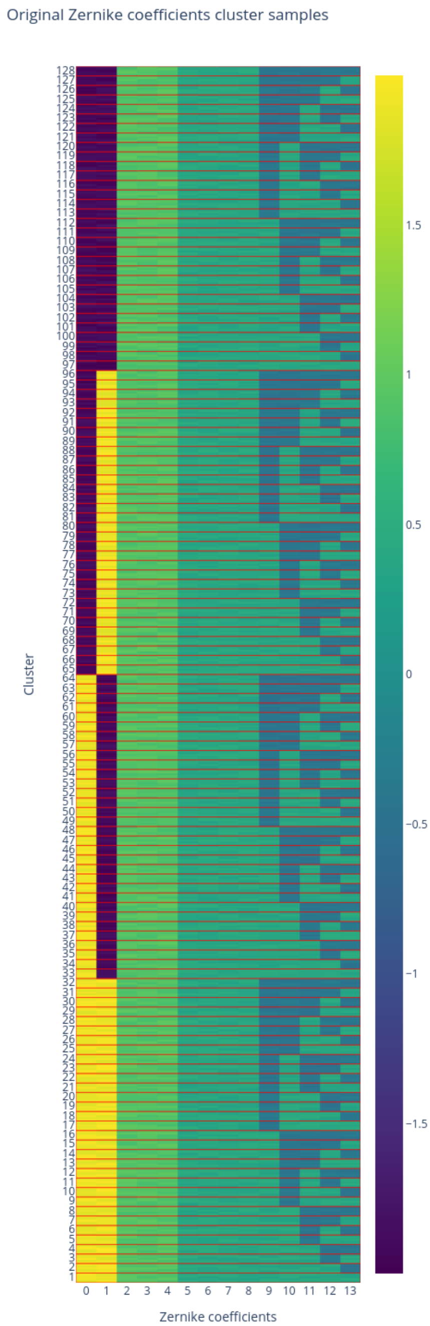
The results are the following:



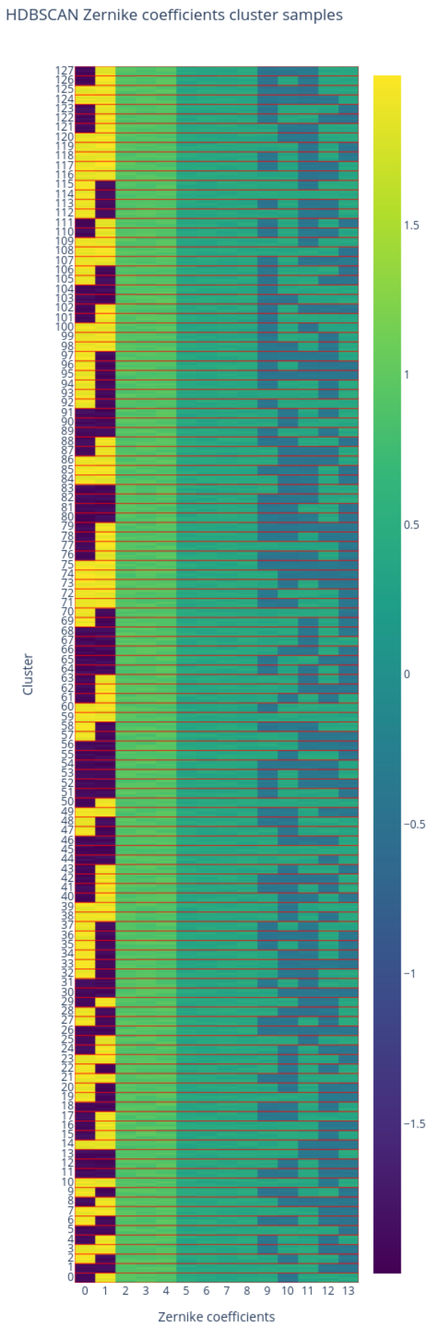
(a) Original cluster densities



(b) HDBSCAN clusters densities



(c) Original cluster samples



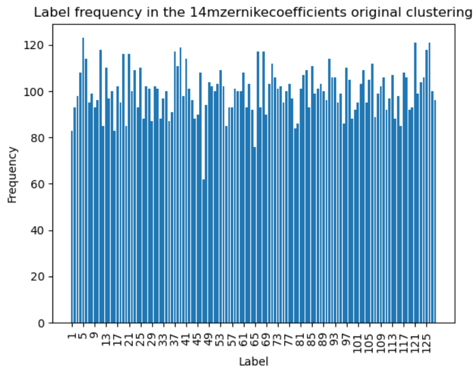
(d) HDBSCAN cluster samples

15.1.4 Agglomerative clustering

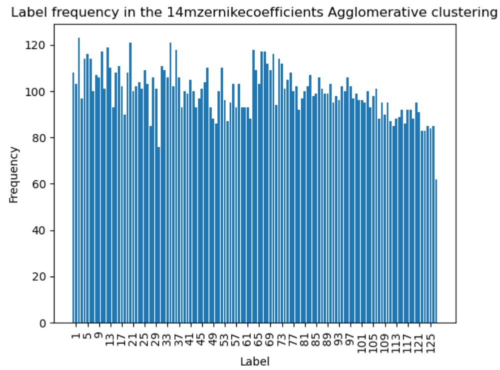
Number of clusters
128

Table 80: Agglomerative hyperparameter configuration for Zernike coefficients clustering

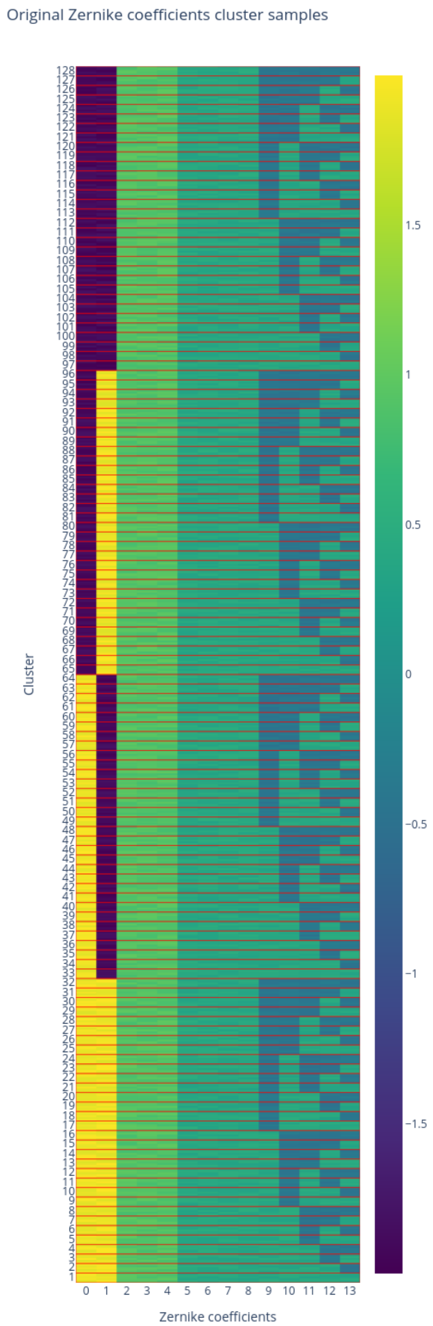
The results are the following:



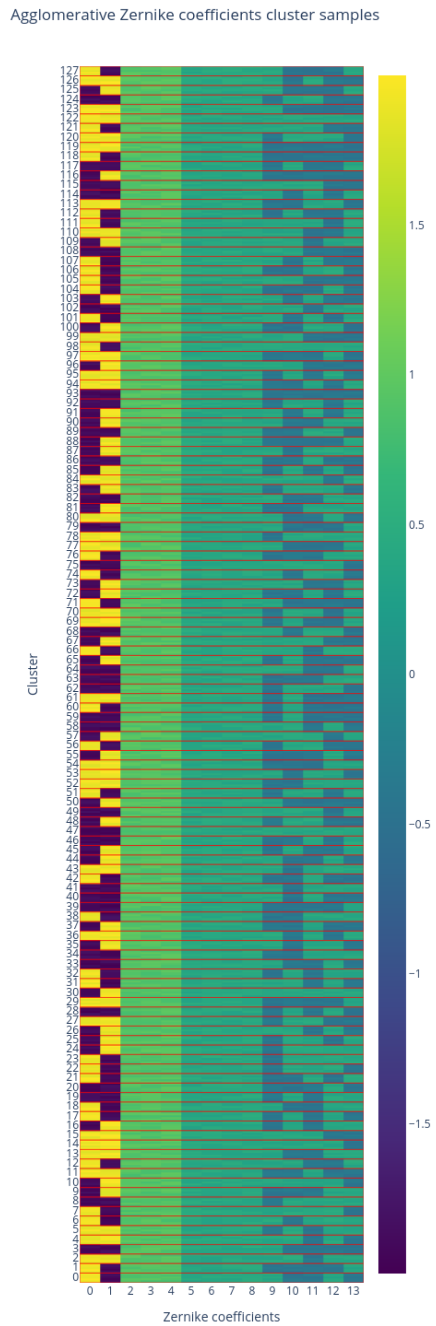
(a) Original cluster densities



(b) Agglomerative clusters densities

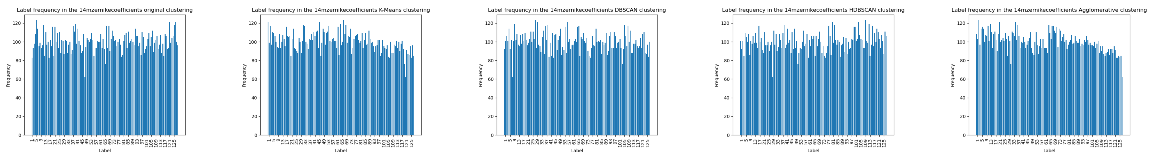


(c) Original cluster samples

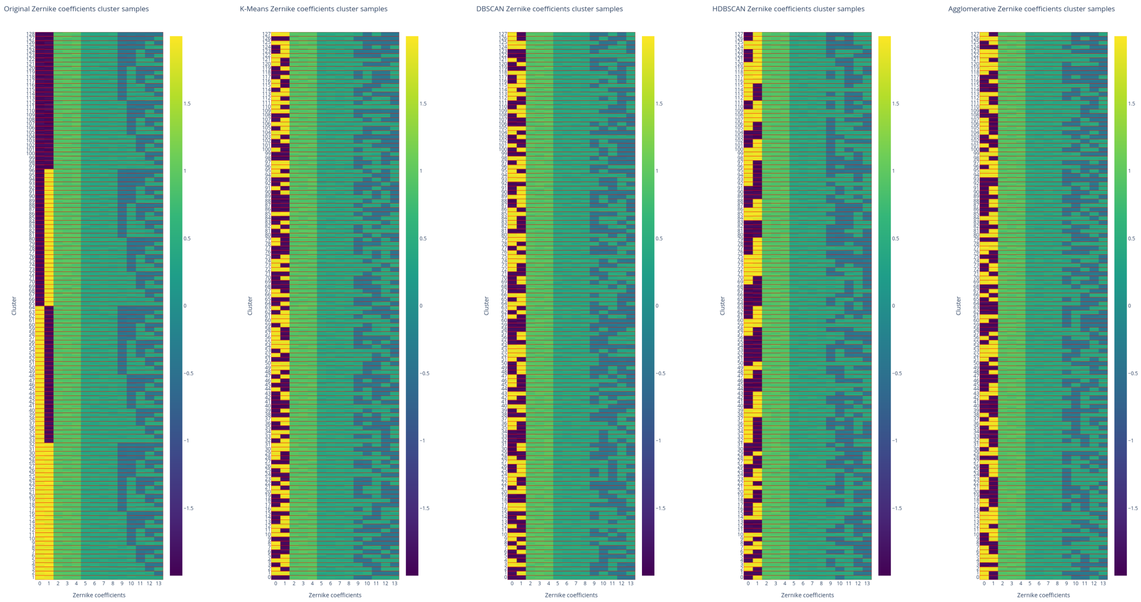


(d) Agglomerative cluster samples

15.1.5 Summary



(a) Original cluster densities (b) K-means cluster densities (c) DBSCAN cluster densities (d) HDBSCAN cluster densities (e) Agglomerative cluster densities



(f) Original cluster samples (g) K-means cluster samples (h) DBSCAN cluster samples (i) HDBSCAN cluster samples (j) Agglomerative cluster samples

Figure 205: Comparison between clustering algorithms

	Original	**K-Means**	**DBSCAN**	**HDBSCAN**	**Agglomerative**
Original		1	0.999	1	0.999
K-Means			0.999	1	1
DBSCAN				0.999	0.999
HDBSCAN					0.999

Table 81: Normalized Mutual Information between clusters

15.2 LP coefficients clustering

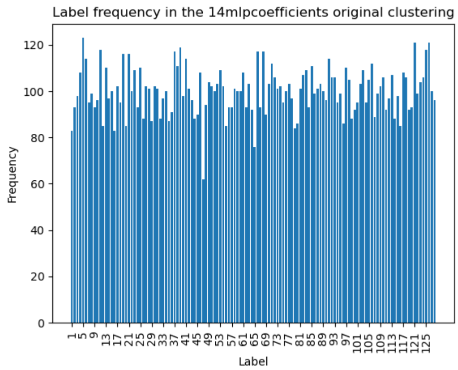
15.2.1 K-Means

As K-Means allows for the number of clusters to be defined, and we know that there are 4 in the original dataset, K-Means is used to find 128 clusters.

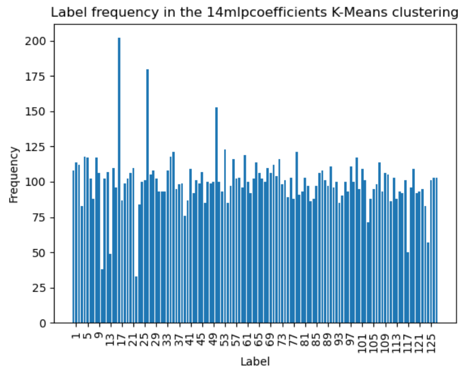
	Number of clusters	**Number of initializations**
Original LP coefficients	128	100

Table 82: K-Means hyperparameter configuration for c coefficients clustering

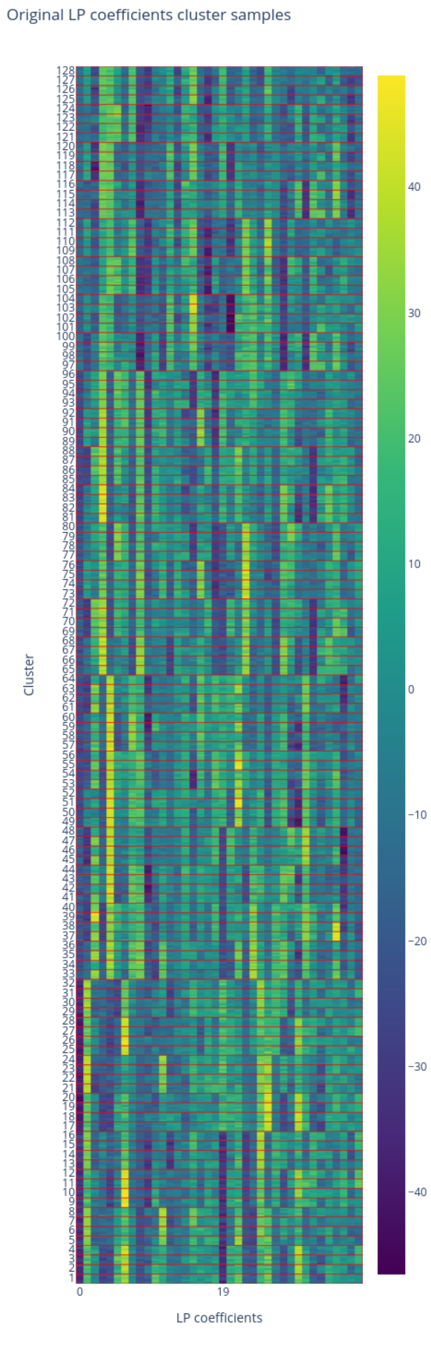
The results are the following:



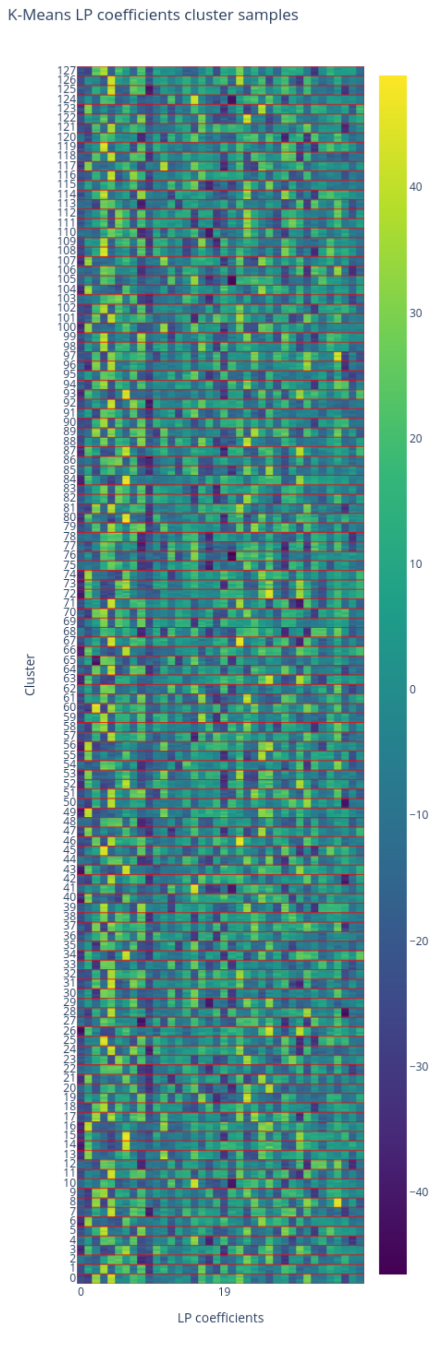
(a) Original cluster densities



(b) K-Means clusters densities



(c) Original cluster samples



(d) K-Means cluster samples

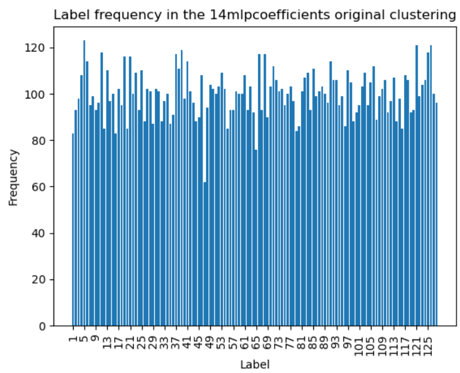
15.2.2 DBSCAN

A configuration that outputs 4 clusters is searched

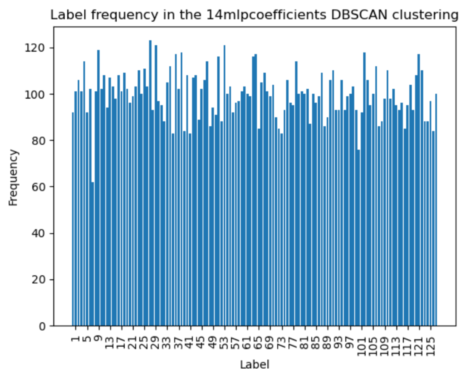
	Number of neighbours	**Epsilon**
Original LP coefficients	10	18.7

Table 83: DBSCAN hyperparameter configuration for LP coefficients clustering

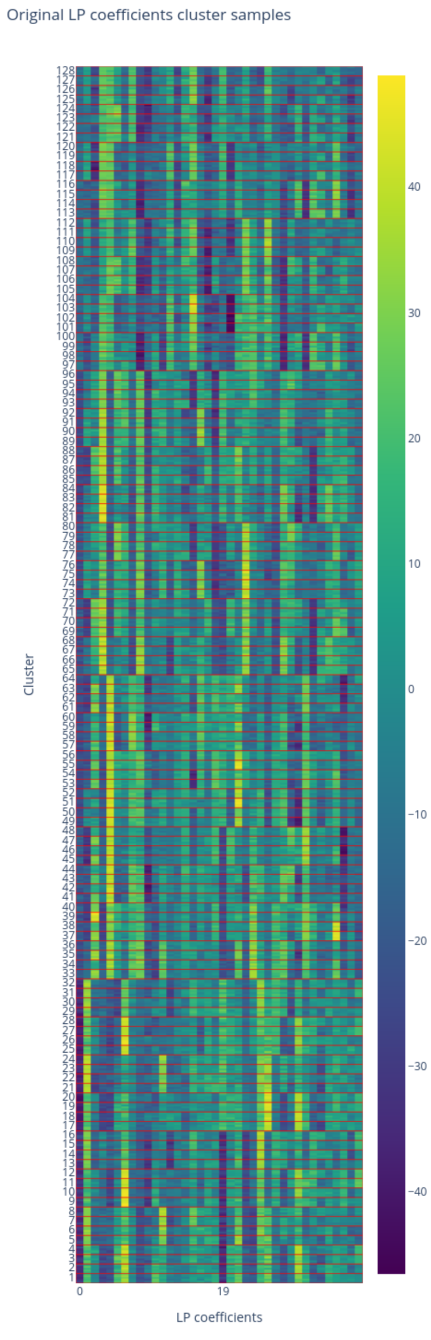
The results are the following:



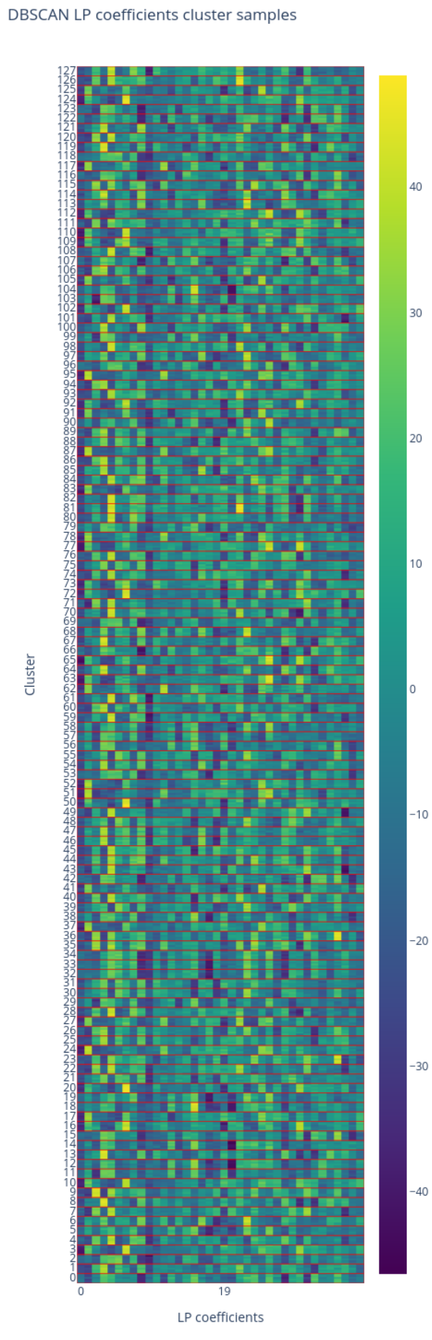
(a) Original cluster densities



(b) DBSCAN clusters densities



(c) Original cluster samples



(d) DBSCAN cluster samples

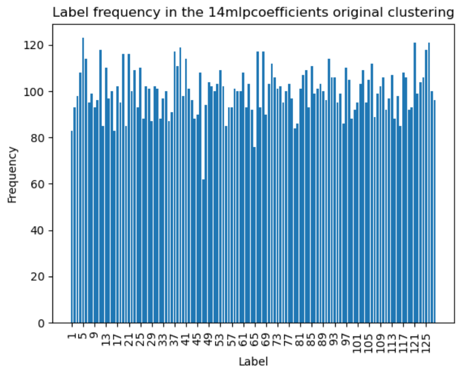
15.2.3 HDBSCAN

A configuration that outputs 4 clusters is searched.

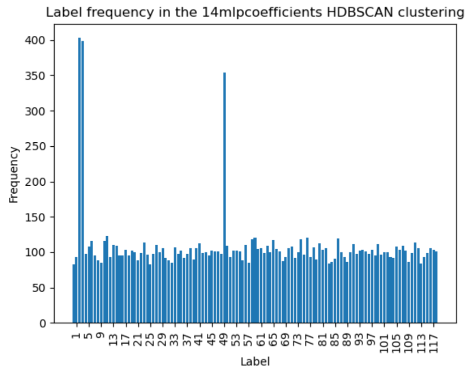
	Minimum cluster size
Original LP coefficients	5

Table 84: HDBSCAN hyperparameter configuration for LP coefficients clustering

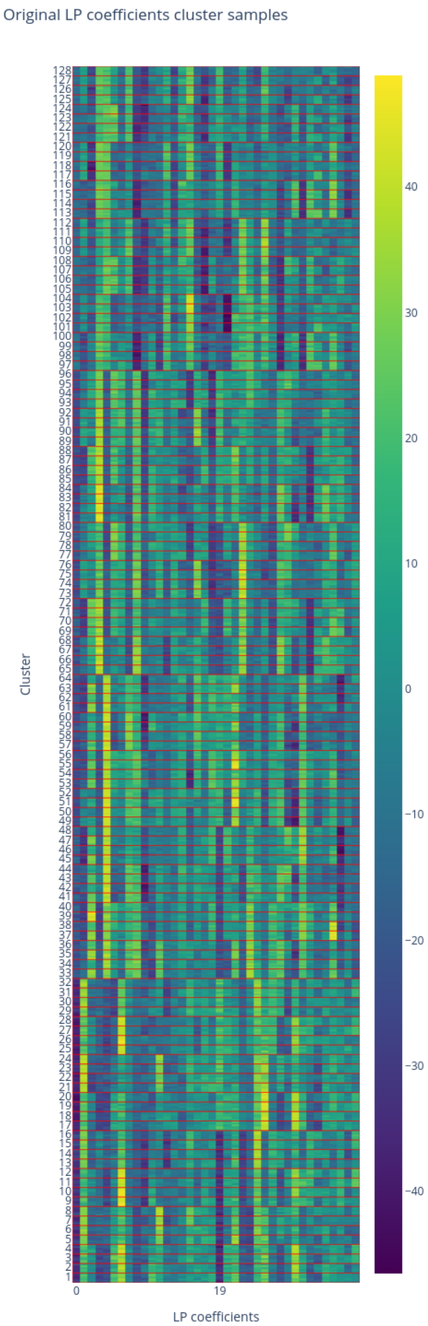
The results are the following:



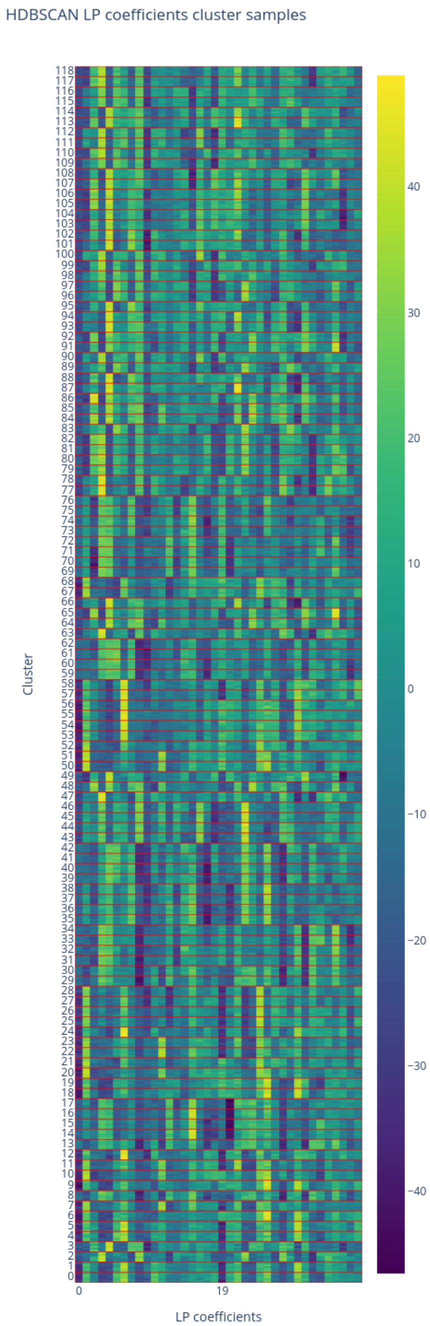
(a) Original cluster densities



(b) HDBSCAN clusters densities



(c) Original cluster samples



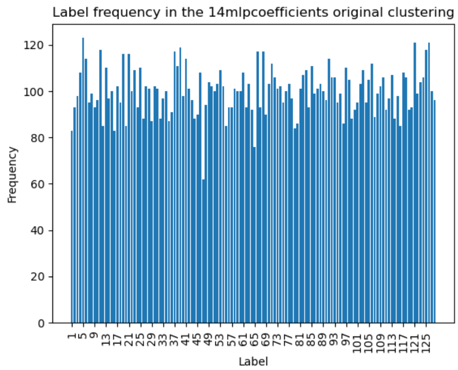
(d) HDBSCAN cluster samples

15.2.4 Agglomerative clustering

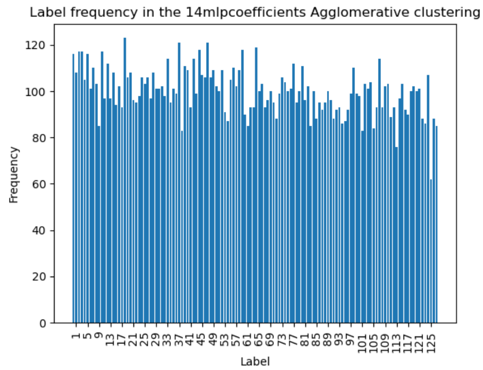
	Number of clusters
Original LP coefficients	128

Table 85: Agglomerative hyperparameter configuration for LP coefficients clustering

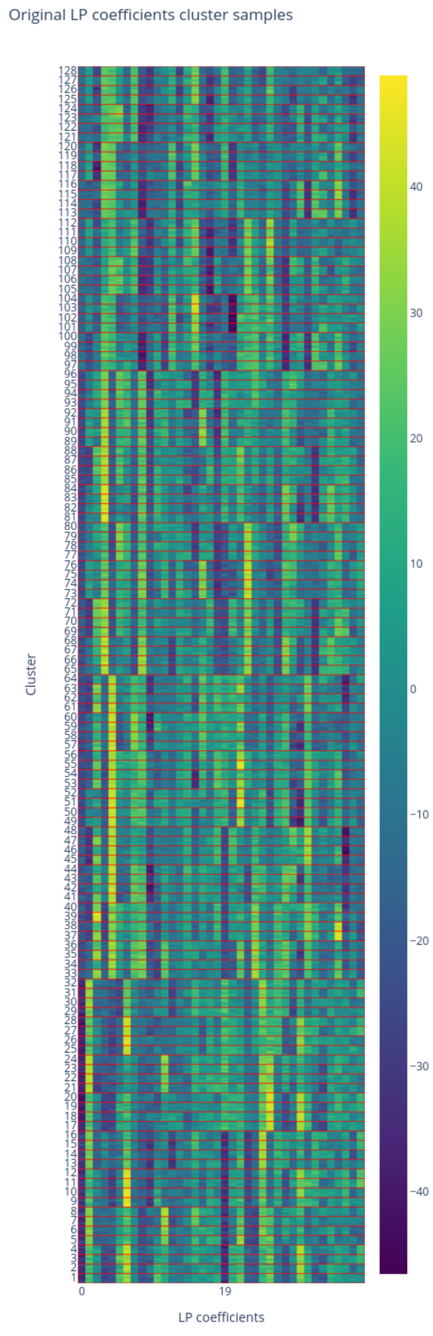
The results are the following:



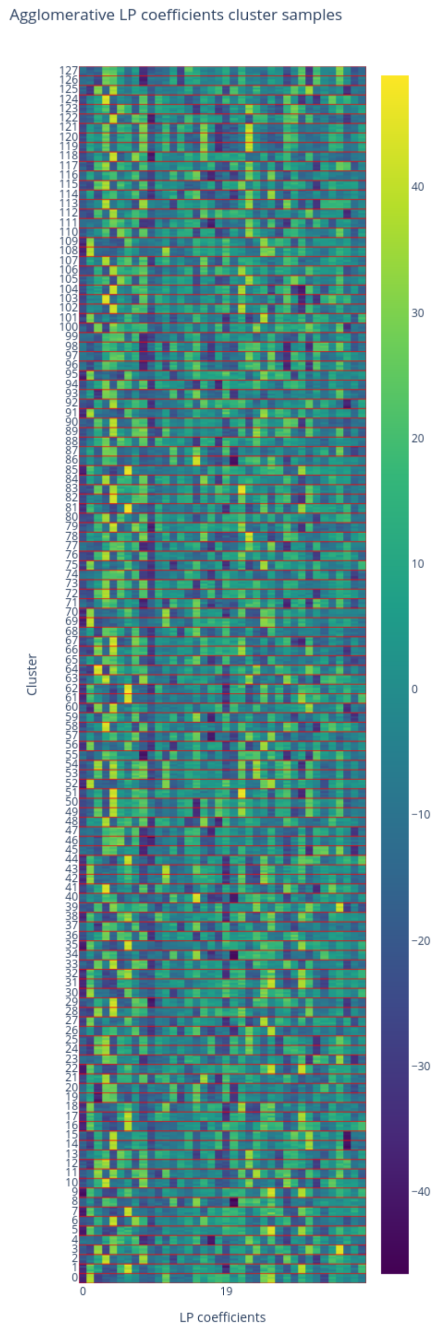
(a) Original cluster densities



(b) Agglomerative clusters densities

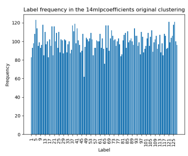


(c) Original cluster samples

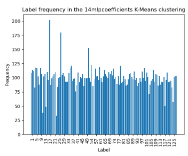


(d) Agglomerative cluster samples

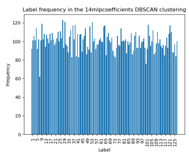
15.2.5 Summary



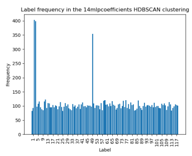
(a) Original cluster densities



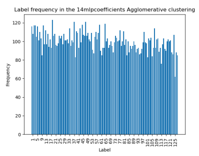
(b) K-means cluster densities



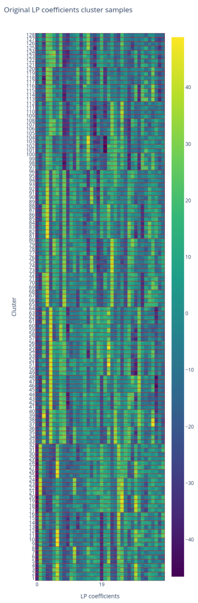
(c) DBSCAN cluster densities



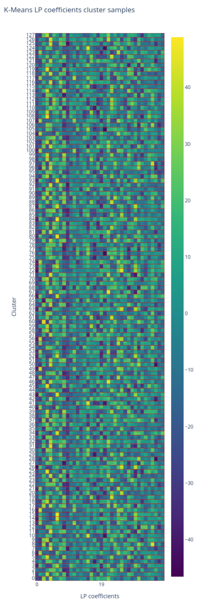
(d) HDBSCAN cluster densities



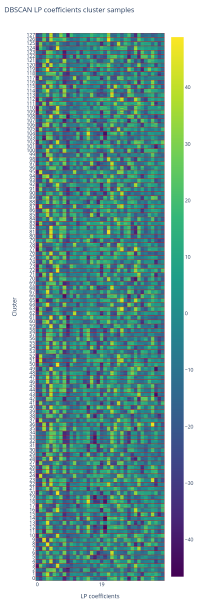
(e) Agglomerative cluster densities



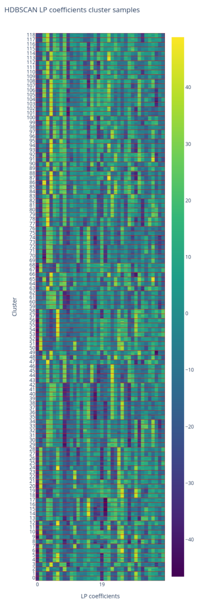
(f) Original cluster samples



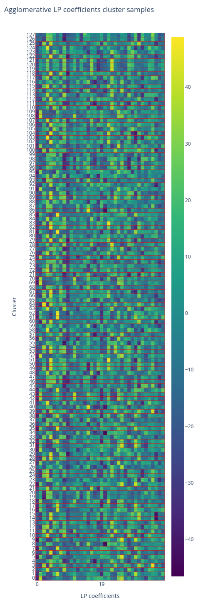
(g) K-means cluster samples



(h) DBSCAN cluster samples



(i) HDBSCAN cluster samples



(j) Agglomerative cluster samples

Figure 210: Comparison between clustering LP coefficients algorithms

	Original	**K-Means**	**DBSCAN**	**HDBSCAN**	**Agglomerative**
Original		1	1	1	1
K-Means			1	1	1
DBSCAN				1	1
HDBSCAN					1

Table 86: Normalized Mutual Information between original LP coefficients clusters

15.3 Output fluxes clustering

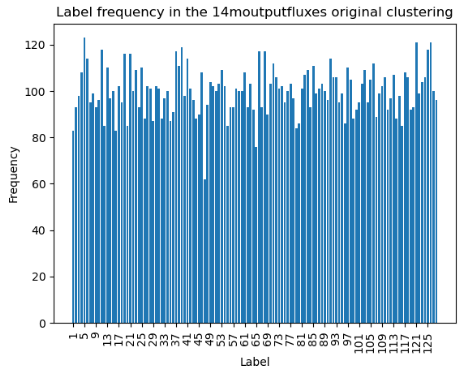
15.3.1 K-Means

As K-Means allows for the number of clusters to be defined, and we know that there are 4 in the original dataset, K-Means is used to find 128 clusters.

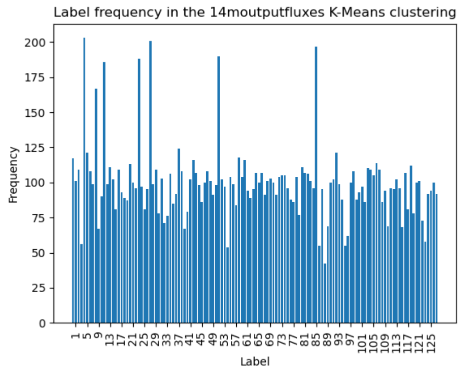
	Number of clusters	Number of initializations
Original Output fluxes	128	100

Table 87: K-Means hyperparameter configuration for output fluxes clustering

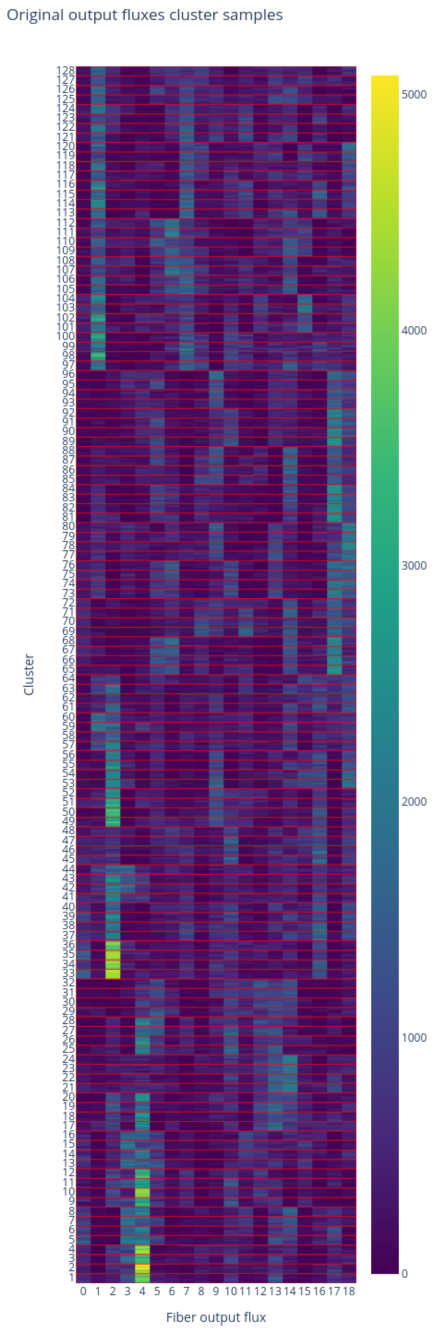
The results are the following:



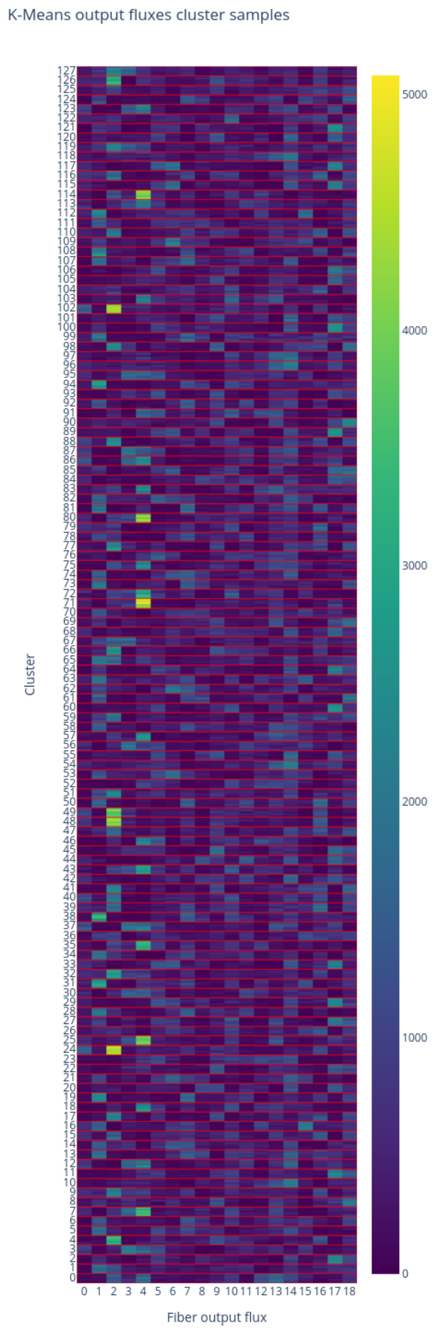
(a) Original cluster densities



(b) K-Means clusters densities



(c) Original cluster samples



(d) K-Means cluster samples

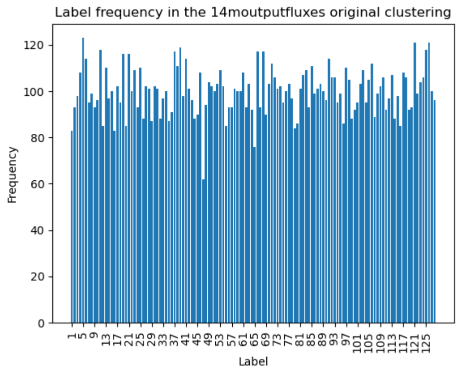
15.3.2 DBSCAN

A configuration that outputs 4 clusters is searched

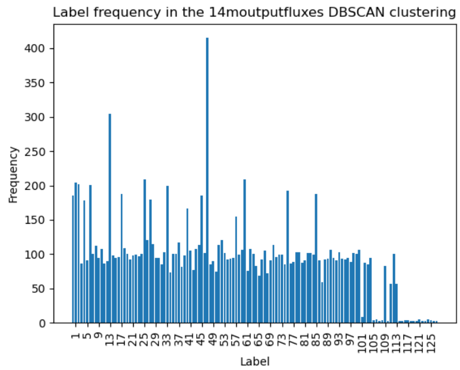
	Number of neighbours	**Epsilon**
Original Output fluxes	3	412

Table 88: DBSCAN hyperparameter configuration for Output fluxes clustering

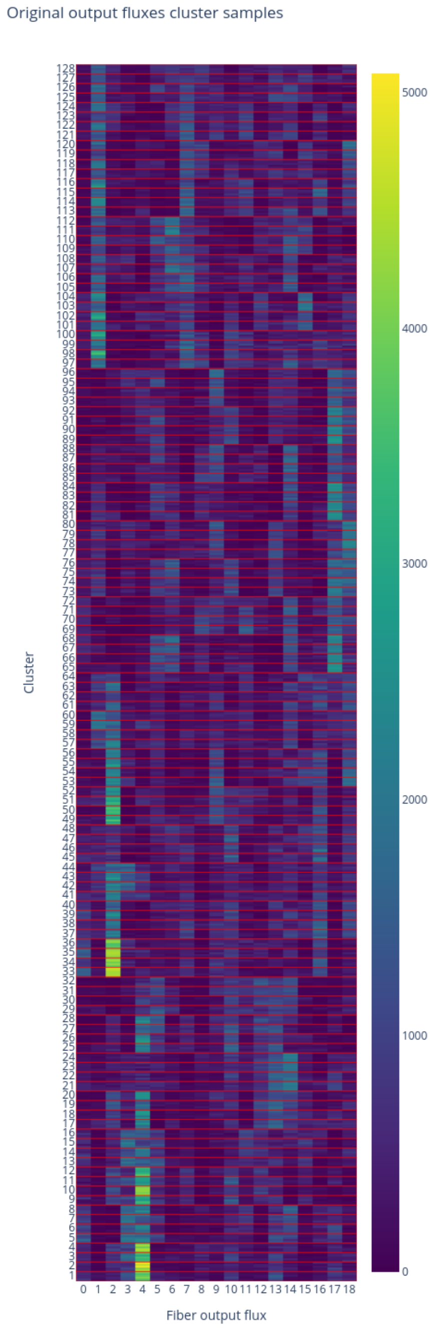
The results are the following:



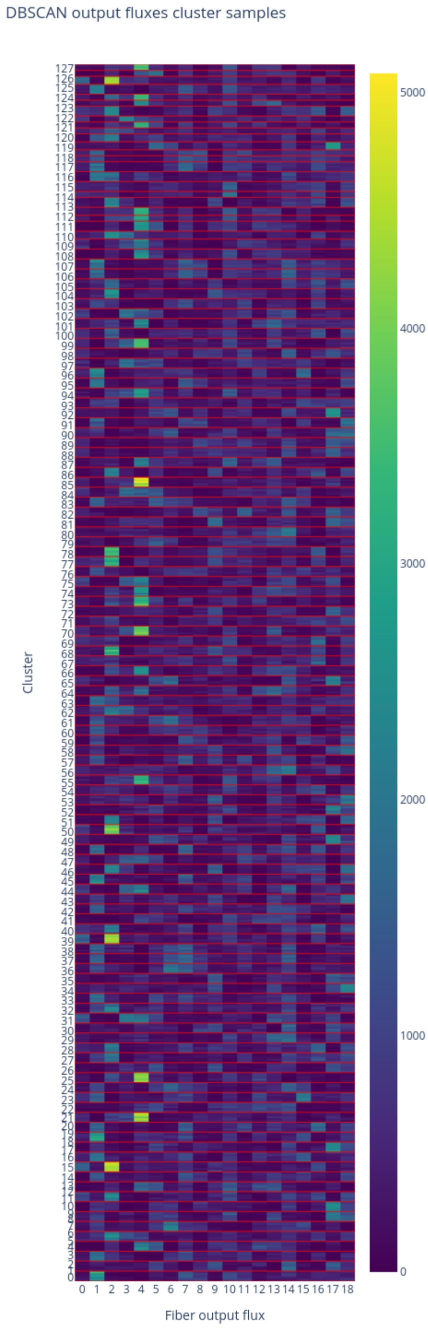
(a) Original cluster densities



(b) DBSCAN clusters densities



(c) Original cluster samples



(d) DBSCAN cluster samples

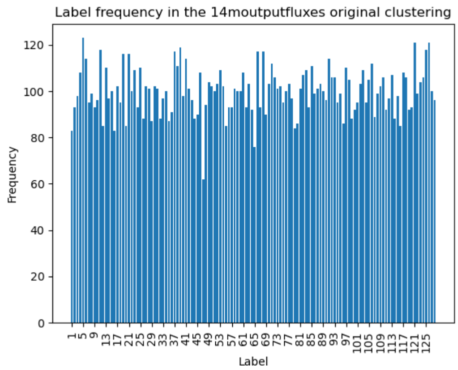
15.3.3 HDBSCAN

A configuration that outputs 4 clusters is searched.

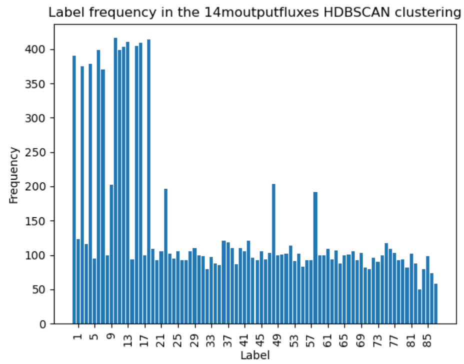
	Minimum cluster size
Original Output fluxes	4

Table 89: HDBSCAN hyperparameter configuration for Output fluxes clustering

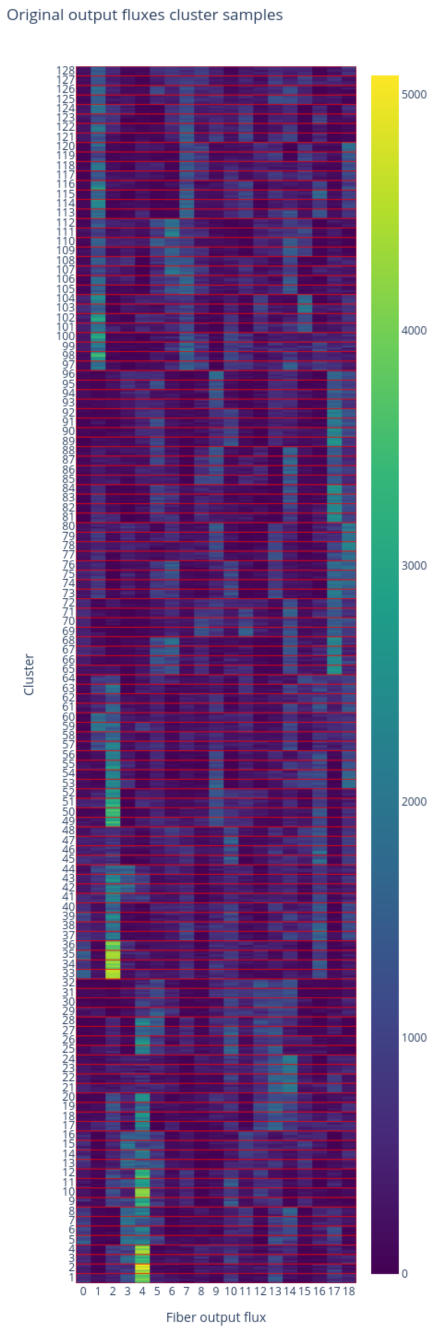
The results are the following:



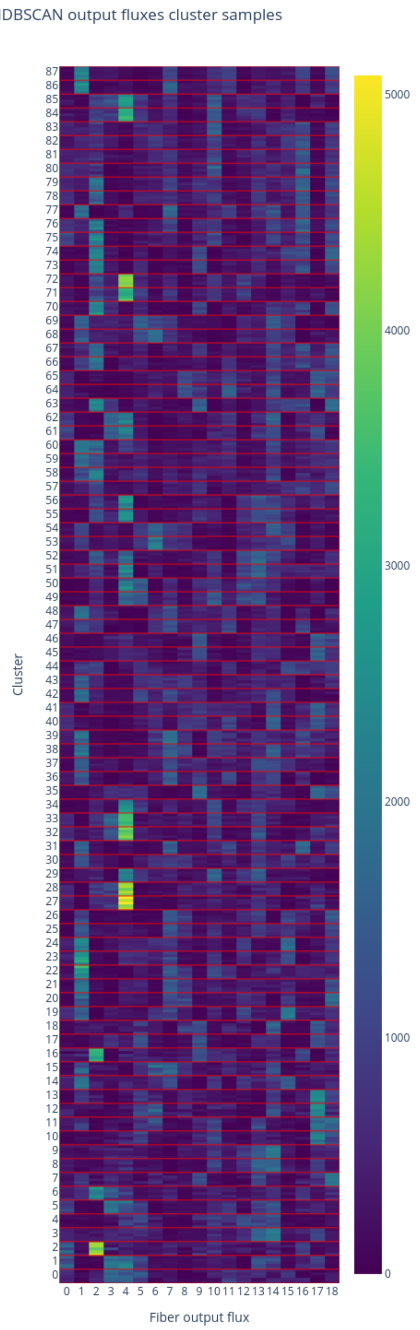
(a) Original cluster densities



(b) HDBSCAN clusters densities



(c) Original cluster samples



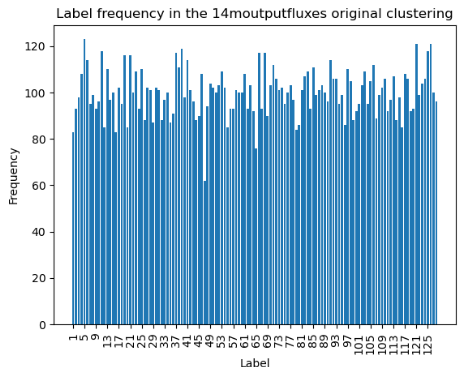
(d) HDBSCAN cluster samples

15.3.4 Agglomerative clustering

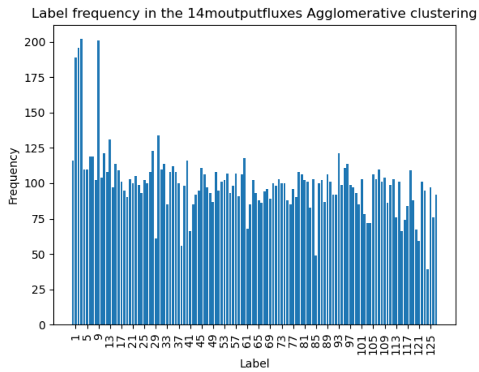
	Number of clusters
Original Output fluxes	128

Table 90: Agglomerative hyperparameter configuration for Output fluxes clustering

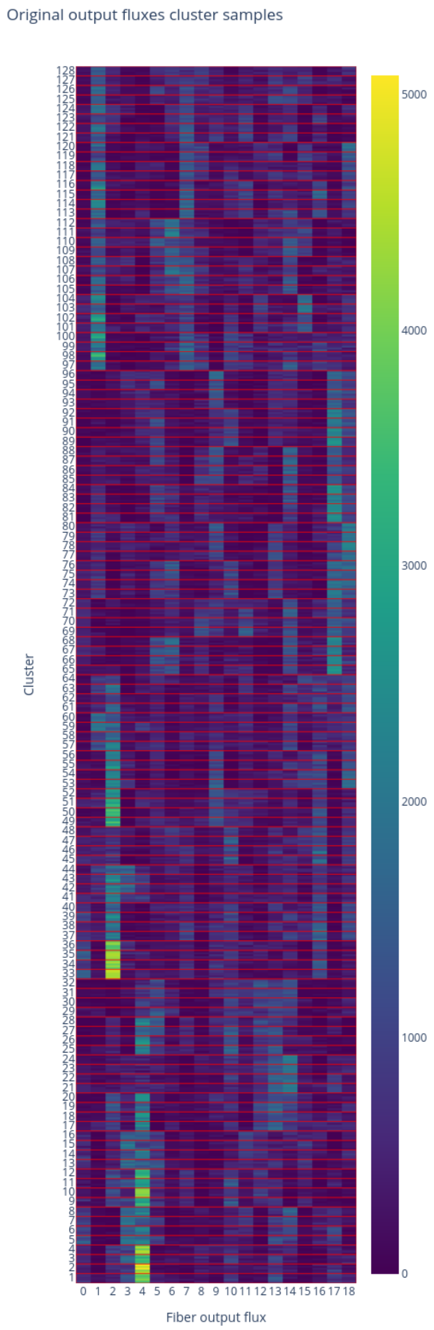
The results are the following:



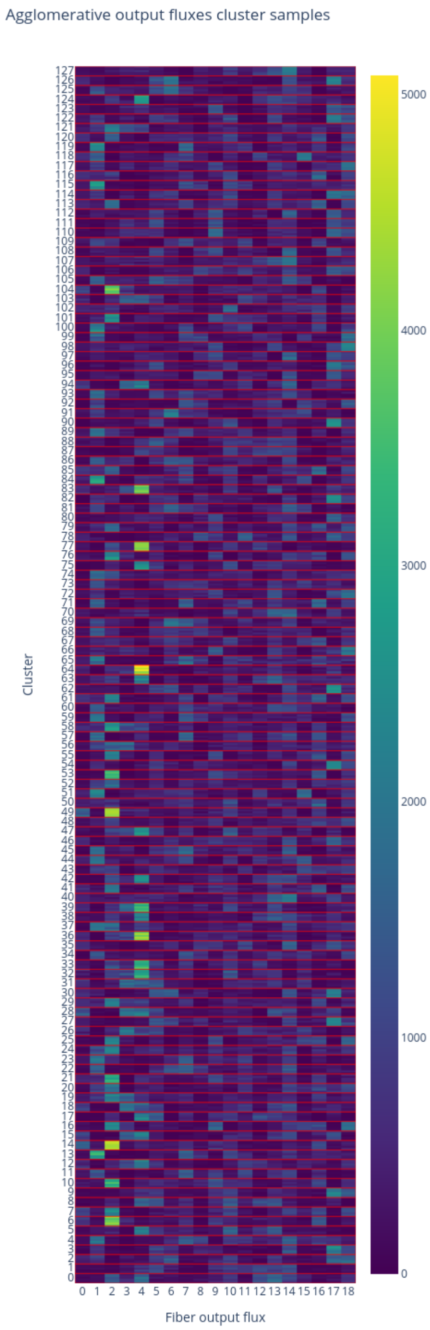
(a) Original cluster densities



(b) Agglomerative clusters densities

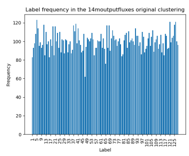


(c) Original cluster samples

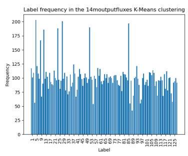


(d) Agglomerative cluster samples

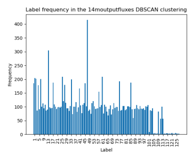
15.3.5 Summary



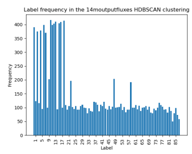
(a) Original cluster densities



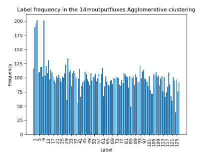
(b) K-means cluster densities



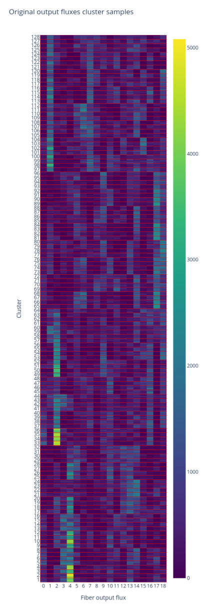
(c) DBSCAN cluster densities



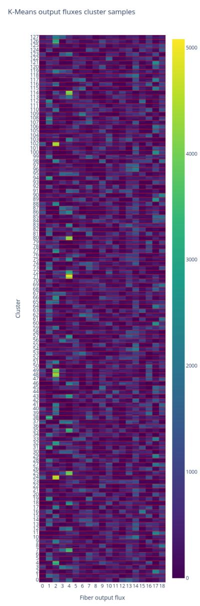
(d) HDBSCAN cluster densities



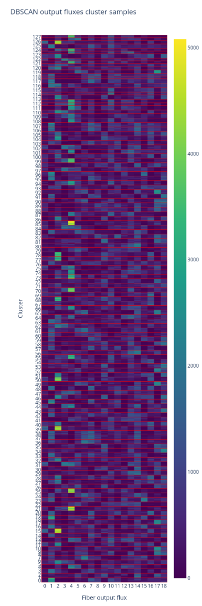
(e) Agglomerative cluster densities



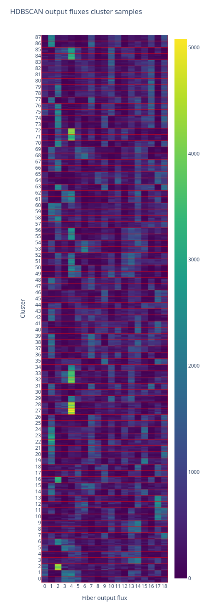
(f) Original cluster samples



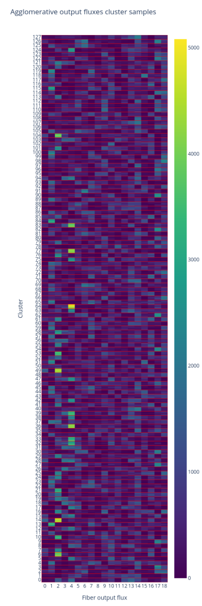
(g) K-means cluster samples



(h) DBSCAN cluster samples



(i) HDBSCAN cluster samples



(j) Agglomerative cluster samples

Figure 215: Comparison between clustering Output fluxes algorithms

	Original	**K-Means**	**DBSCAN**	**HDBSCAN**	**Agglomerative**
Original		0.969	0.939	0.925	0.982
K-Means			0.929	0.923	0.970
DBSCAN				0.922	0.938
HDBSCAN					0.925

Table 91: Normalized Mutual Information between original Output fluxes clusters

Part VIII

Normalized Mutual Information Analysis

16 The data

16.1 Zernike coefficients dataset

8 datasets of zernike coefficients are created, each of the dataset contain 5000 datapoints

- **2 mode dataset**: 2 Zernike modes coefficients, their RMSE in the range [-1,1]
- **5 mode dataset**: 5 Zernike modes coefficients, their RMSE in the range:
  - Modes 2 and 3: [-0.4, 0.4]
  - Modes 4, 5 and 6: [-0.4, 0.4]
- **9 mode dataset**: 9 Zernike mode coefficients, their RMSE in the range:
  - Modes 2 and 3: [-0.22, 0.22]
  - Modes 4, 5 and 6: [-0.22, 0.22]
  - Modes 7, 8, 9 and 10: [-0.22, 0.22]
- **14 mode dataset**: 14 Zernike mode coefficients, their RMSE in the range:
  - Modes 2 and 3: [-0.142, 0.142]
  - Modes 4, 5 and 6: [-0.142, 0.142]
  - Modes 7, 8, 9 and 10: [-0.142, 0.142]
  - Modes 11, 12, 13, 14 and 15: [-0.142, 0.142]

- **20 mode dataset**: 20 Zernike mode coefficients, their RMSE in the range:
  - Modes 2 and 3: [-0.1, 0.1]
  - Modes 4, 5 and 6: [-0.1, 0.1]
  - Modes 7, 8, 9 and 10: [-0.1, 0.1]
  - Modes 11, 12, 13, 14 and 15: [-0.1, 0.1]
  - Modes 16, 17, 18, 19, 20 and 21: [-0.1, 0.1]
- **27 mode dataset**: 27 Zernike mode coefficients, their RMSE in the range:
  - Modes 2 and 3: [-0.07, 0.07]
  - Modes 4, 5 and 6: [-0.07, 0.07]
  - Modes 7, 8, 9 and 10: [-0.07, 0.07]
  - Modes 11, 12, 13, 14 and 15: [-0.07, 0.07]
  - Modes 16, 17, 18, 19, 20 and 21: [-0.07, 0.07]
  - Modes 22, 23, 24, 25, 26, 27 and 28: [-0.07, 0.07]
- **35 mode dataset**: 35 Zernike mode coefficients, their RMSE in the range:
  - Modes 2 and 3: [-0.05, 0.05]
  - Modes 4, 5 and 6: [-0.05, 0.05]
  - Modes 7, 8, 9 and 10: [-0.05, 0.05]
  - Modes 11, 12, 13, 14 and 15: [-0.05, 0.05]
  - Modes 16, 17, 18, 19, 20 and 21: [-0.05, 0.05]
  - Modes 22, 23, 24, 25, 26, 27 and 28: [-0.05, 0.05]
  - Modes 29, 30, 31, 32, 33, 34, 35 and 36: [-0.05, 0.05]
- **44 mode dataset**: 44 Zernike mode coefficients, their RMSE in the range:

- Modes 2 and 3: [-0.04, 0.04]
- Modes 4, 5 and 6: [-0.04, 0.04]
- Modes 7, 8, 9 and 10: [-0.04, 0.04]
- Modes 11, 12, 13, 14 and 15: [-0.04, 0.04]
- Modes 16, 17, 18, 19, 20 and 21: [-0.04, 0.04]
- Modes 22, 23, 24, 25, 26, 27 and 28: [-0.04, 0.04]
- Modes 29, 30, 31, 32, 33, 34, 35 and 36: [-0.04, 0.04]
- Modes 37, 38, 39, 40, 41, 42, 43, 44 and 45: [-0.04, 0.04]

16.2 PSFs intensities dataset

8 datasets of 5000 PSF intensities are created from the 5 zernike coefficients dataset, each datapoint being a 128x128 matrix

16.3 LP mode coefficients dataset

8 datasets of 5000 LP coefficients are created from the 5 zernike coefficients dataset, each datapoint being a 19x2 matrix dividing the real and imaginary part of the LP coefficients

16.4 Output fluxes dataset

The output fluxes are obtained using the 19 mode PL transfer matrix. 8 datasets of 5000 Output fluxes are created from the 5 zernike coefficients dataset, each datapoint being a 19x1 vector

17 Preprocessing

17.1 PSF Intensities

The 5000x128x128 PSF intensities datasets are dimensionally reduced using UMAP giving an array of 5000x19 projections of the PSF Intensities for each dataset.

Number of neighbors	**Min distance**	**Number of components**
15	0.3	19

Table 92: UMAP hyperparameter configurations for PSF Intensities dimensionality reduction

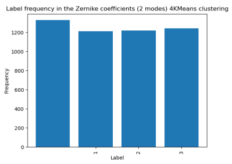
17.2 Clustering

K-Means is used to find clusters in all the datasets. KMeans is set to produce the following number of clusters:

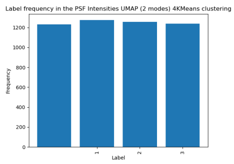
- 4
- 8
- 16
- 32
- 64
- 100
- 250
- 500
- 1000

The results are the following:

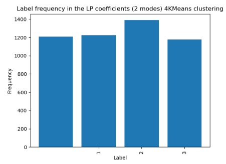
17.2.1 2 Zernike modes datasets clusters densities



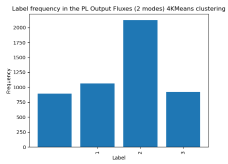
(a) 4KMeans for Zernike coefficients



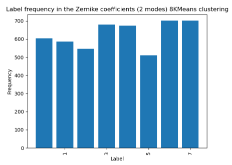
(b) 4KMeans PSF Intensities UMAP



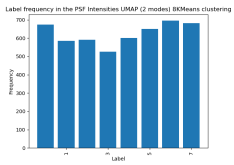
(c) 4KMeans LP coefficients



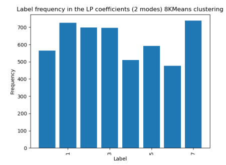
(d) 4KMeans Output Fluxes



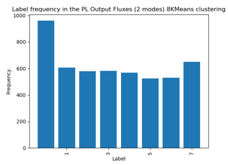
(a) 8KMeans for Zernike coefficients



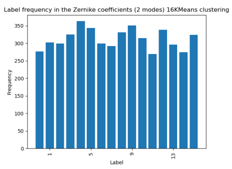
(b) 8KMeans PSF Intensities UMAP



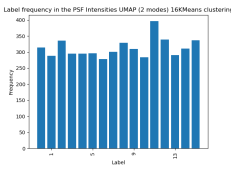
(c) 8KMeans LP coefficients



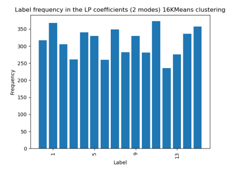
(d) 8KMeans Output Fluxes



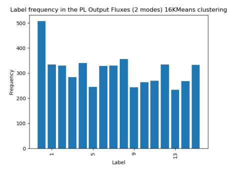
(a) 16KMeans for Zernike coefficients



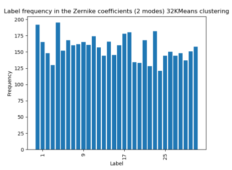
(b) 16KMeans PSF Intensities UMAP



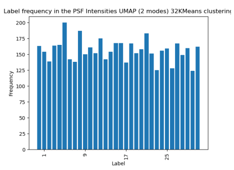
(c) 16KMeans LP coefficients



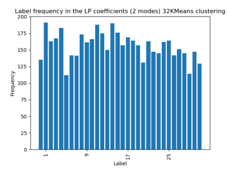
(d) 16KMeans Output Fluxes



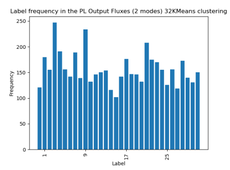
(a) 32KMeans for Zernike coefficients



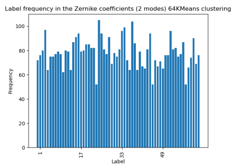
(b) 32KMeans PSF Intensities UMAP



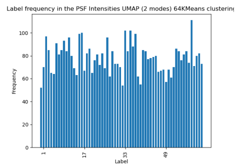
(c) 32KMeans LP coefficients



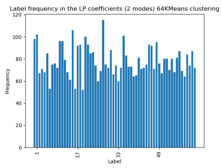
(d) 32KMeans Output Fluxes



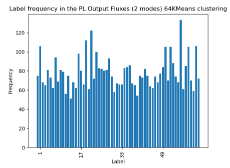
(a) 64KMeans for Zernike coefficients



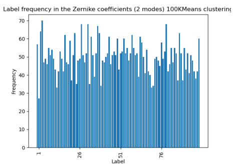
(b) 64KMeans PSF Intensities UMAP



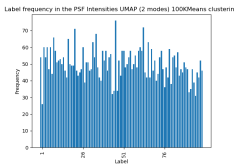
(c) 64KMeans LP coefficients



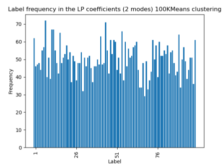
(d) 64KMeans Output Fluxes



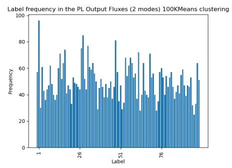
(a) 100KMeans for Zernike coefficients



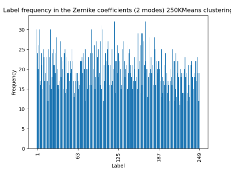
(b) 100KMeans PSF Intensities UMAP



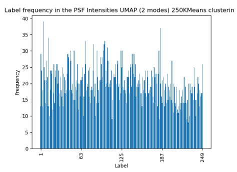
(c) 100KMeans LP coefficients



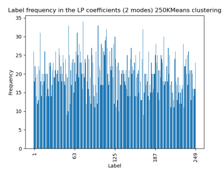
(d) 100KMeans Output Fluxes



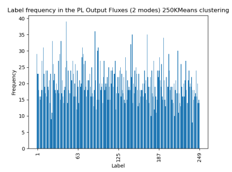
(a) 250KMeans for Zernike coefficients



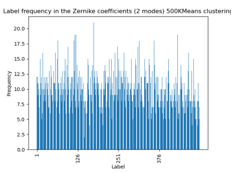
(b) 250KMeans PSF Intensities UMAP



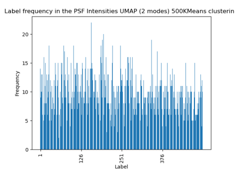
(c) 250KMeans LP coefficients



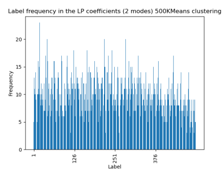
(d) 250KMeans Output Fluxes



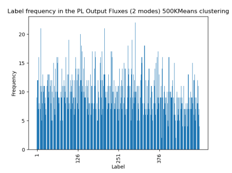
(a) 500KMeans for Zernike coefficients



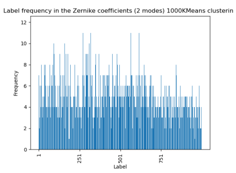
(b) 500KMeans PSF Intensities UMAP



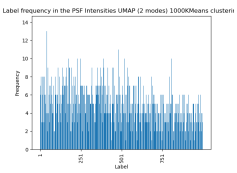
(c) 500KMeans LP coefficients



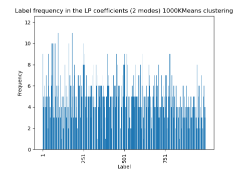
(d) 500KMeans Output Fluxes



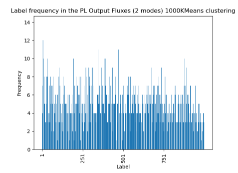
(a) 1000KMeans for Zernike coefficients



(b) 1000KMeans PSF Intensities UMAP

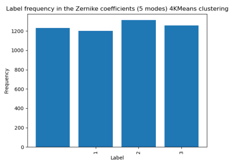


(c) 1000KMeans LP coefficients

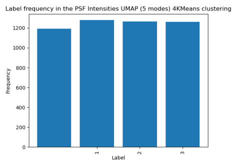


(d) 1000KMeans Output Fluxes

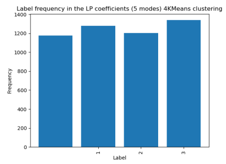
17.2.2 5 Zernike modes datasets clusters densities



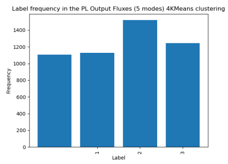
(a) 4KMeans for Zernike coefficients



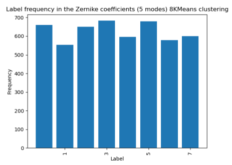
(b) 4KMeans PSF Intensities UMAP



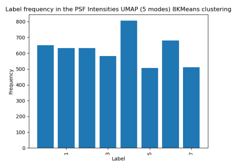
(c) 4KMeans LP coefficients



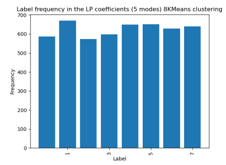
(d) 4KMeans Output Fluxes



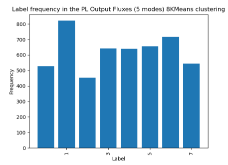
(a) 8KMeans for Zernike coefficients



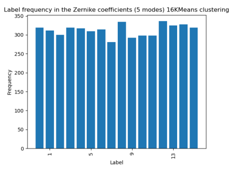
(b) 8KMeans PSF Intensities UMAP



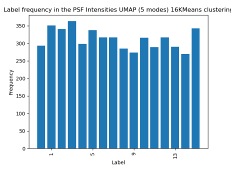
(c) 8KMeans LP coefficients



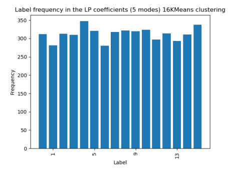
(d) 8KMeans Output Fluxes



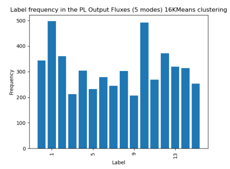
(a) 16KMeans for Zernike coefficients



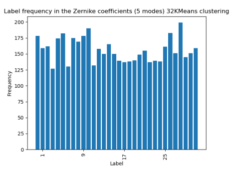
(b) 16KMeans PSF Intensities UMAP



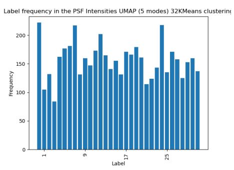
(c) 16KMeans LP coefficients



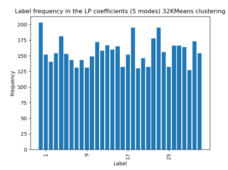
(d) 16KMeans Output Fluxes



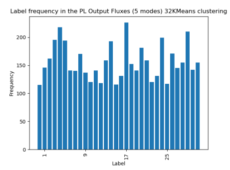
(a) 32KMeans for Zernike coefficients



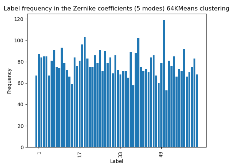
(b) 32KMeans PSF Intensities UMAP



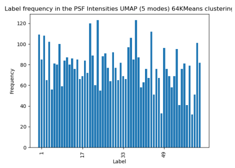
(c) 32KMeans LP coefficients



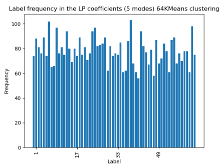
(d) 32KMeans Output Fluxes



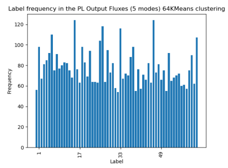
(a) 64KMeans for Zernike coefficients



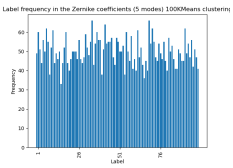
(b) 64KMeans PSF Intensities UMAP



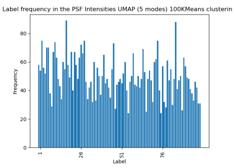
(c) 64KMeans LP coefficients



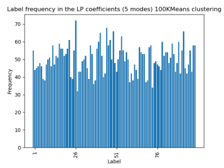
(d) 64KMeans Output Fluxes



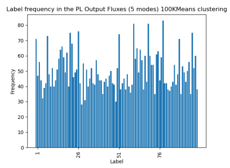
(a) 100KMeans for Zernike coefficients



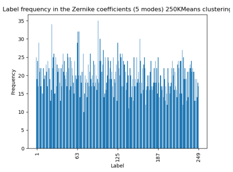
(b) 100KMeans PSF Intensities UMAP



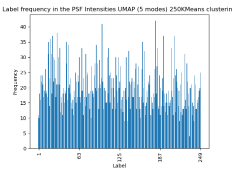
(c) 100KMeans LP coefficients



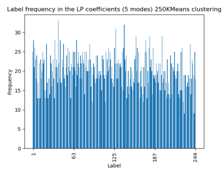
(d) 100KMeans Output Fluxes



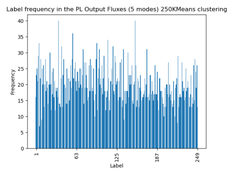
(a) 250KMeans for Zernike coefficients



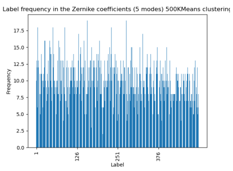
(b) 250KMeans PSF Intensities UMAP



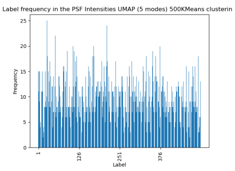
(c) 250KMeans LP coefficients



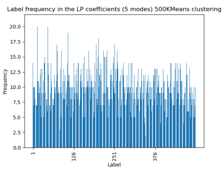
(d) 250KMeans Output Fluxes



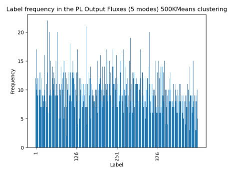
(a) 500KMeans for Zernike coefficients



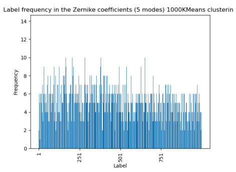
(b) 500KMeans PSF Intensities UMAP



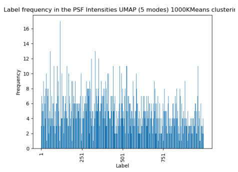
(c) 500KMeans LP coefficients



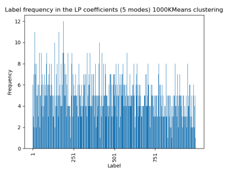
(d) 500KMeans Output Fluxes



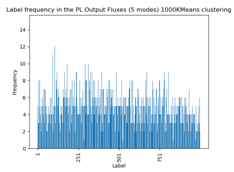
(a) 1000KMeans for Zernike coefficients



(b) 250KMeans PSF Intensities UMAP

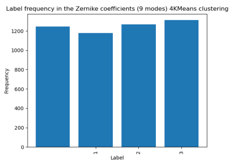


(c) 250KMeans LP coefficients

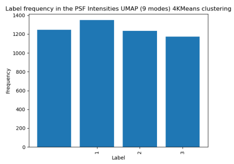


(d) 250KMeans Output Fluxes

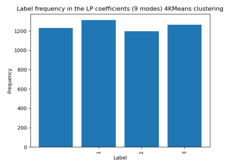
17.2.3 9 Zernike modes datasets clusters densities



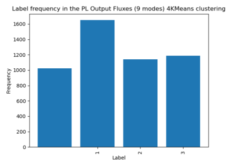
(a) 4KMeans for Zernike coefficients



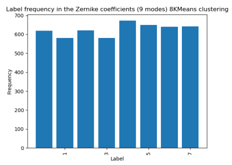
(b) 4KMeans PSF Intensities UMAP



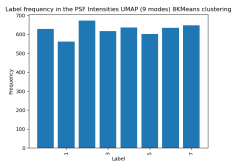
(c) 4KMeans LP coefficients



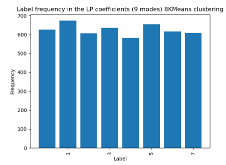
(d) 4KMeans Output Fluxes



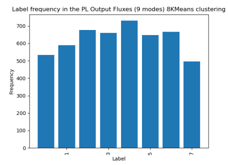
(a) 8KMeans for Zernike coefficients



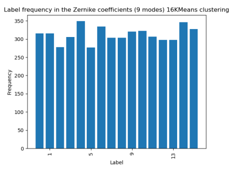
(b) 8KMeans PSF Intensities UMAP



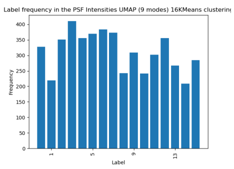
(c) 8KMeans LP coefficients



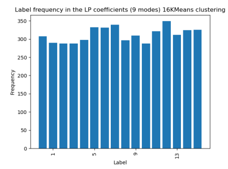
(d) 8KMeans Output Fluxes



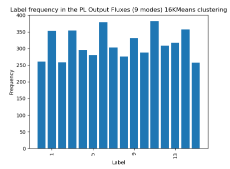
(a) 16KMeans for Zernike coefficients



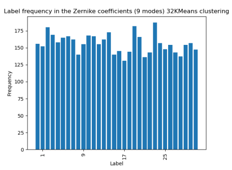
(b) 16KMeans PSF Intensities UMAP



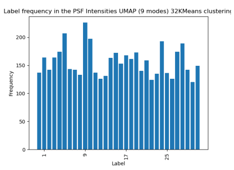
(c) 16KMeans LP coefficients



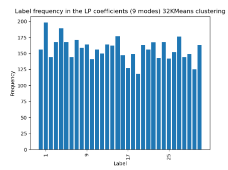
(d) 16KMeans Output Fluxes



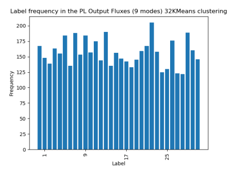
(a) 32KMeans for Zernike coefficients



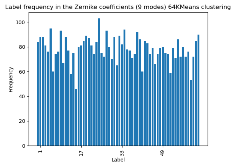
(b) 32KMeans PSF Intensities UMAP



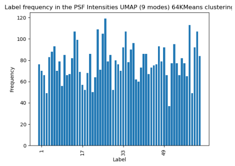
(c) 32KMeans LP coefficients



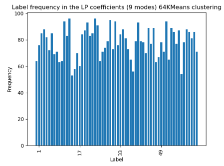
(d) 32KMeans Output Fluxes



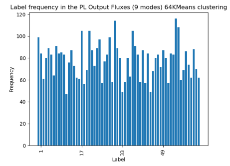
(a) 64KMeans for Zernike coefficients



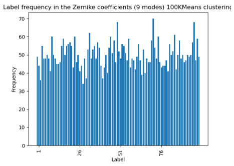
(b) 64KMeans PSF Intensities UMAP



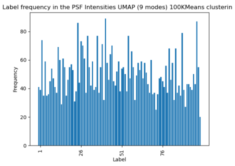
(c) 64KMeans LP coefficients



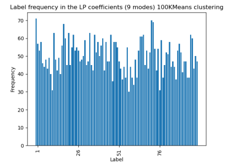
(d) 64KMeans Output Fluxes



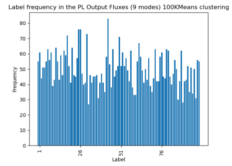
(a) 100KMeans for Zernike coefficients



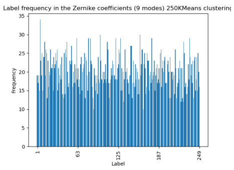
(b) 100KMeans PSF Intensities UMAP



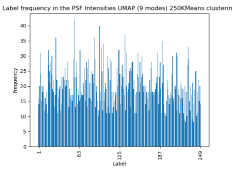
(c) 100KMeans LP coefficients



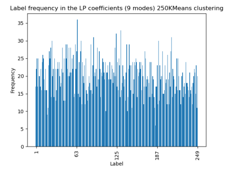
(d) 100KMeans Output Fluxes



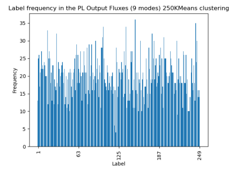
(a) 250KMeans for Zernike coefficients



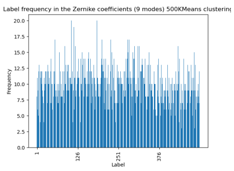
(b) 250KMeans PSF Intensities UMAP



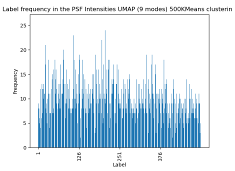
(c) 250KMeans LP coefficients



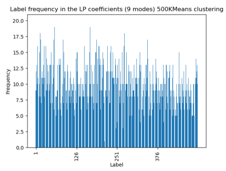
(d) 250KMeans Output Fluxes



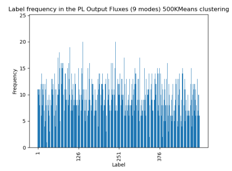
(a) 500KMeans for Zernike coefficients



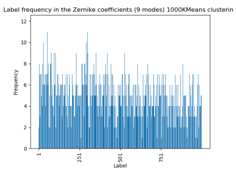
(b) 500KMeans PSF Intensities UMAP



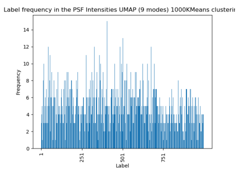
(c) 500KMeans LP coefficients



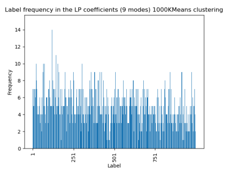
(d) 500KMeans Output Fluxes



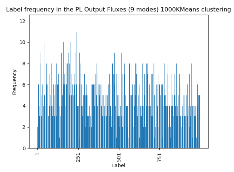
(a) 1000KMeans for Zernike coefficients



(b) 1000KMeans PSF Intensities UMAP

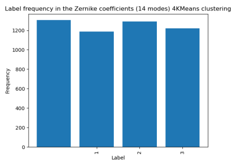


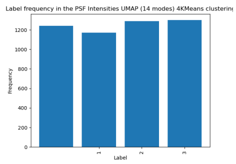
(c) 1000KMeans LP coefficients

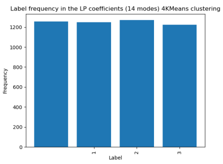


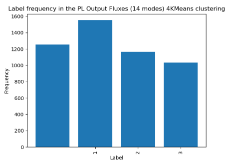
(d) 1000KMeans Output Fluxes

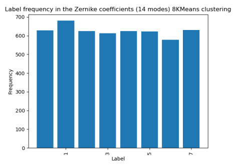
17.2.4 14 Zernike modes datasets clusters densities

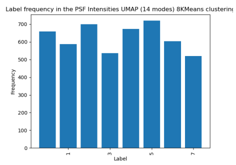

(a) 4KMeans for Zernike coefficients

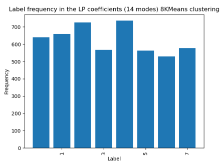

(b) 4KMeans PSF Intensities UMAP

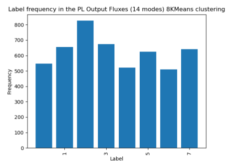

(c) 4KMeans LP coefficients

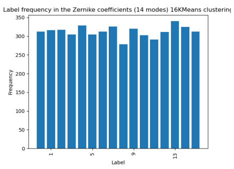

(d) 4KMeans Output Fluxes

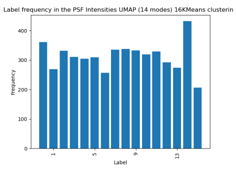

(a) 8KMeans for Zernike coefficients

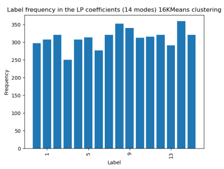

(b) 8KMeans PSF Intensities UMAP

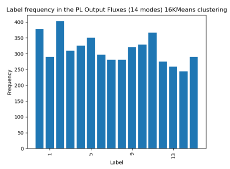

(c) 8KMeans LP coefficients

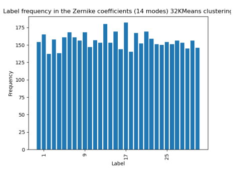

(d) 8KMeans Output Fluxes

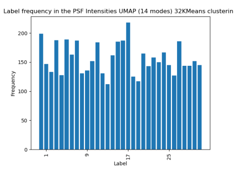

(a) 16KMeans for Zernike coefficients

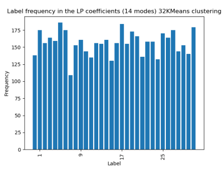

(b) 16KMeans PSF Intensities UMAP

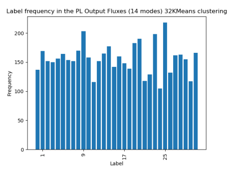

(c) 16KMeans LP coefficients

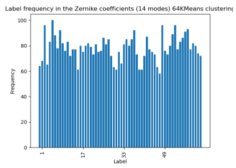

(d) 16KMeans Output Fluxes


(a) 32KMeans for Zernike coefficients

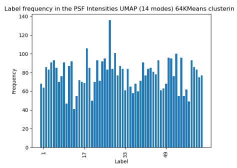

(b) 32KMeans PSF Intensities UMAP


(c) 32KMeans LP coefficients

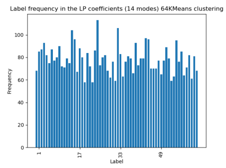

(d) 32KMeans Output Fluxes



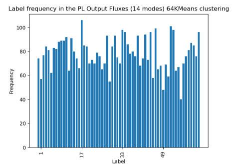
(a) 64KMeans for Zernike coefficients



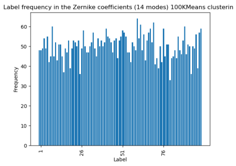
(b) 64KMeans PSF Intensities UMAP



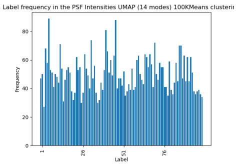
(c) 64KMeans LP coefficients



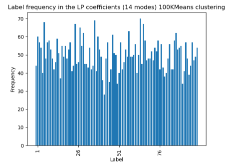
(d) 64KMeans Output Fluxes



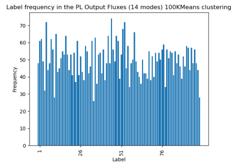
(a) 100KMeans for Zernike coefficients



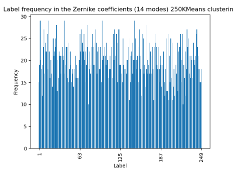
(b) 100KMeans PSF Intensities UMAP



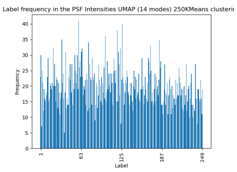
(c) 100KMeans LP coefficients



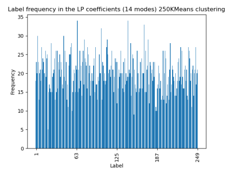
(d) 100KMeans Output Fluxes



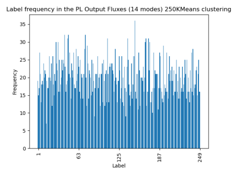
(a) 250KMeans for Zernike coefficients



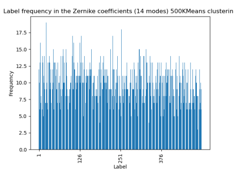
(b) 250KMeans PSF Intensities UMAP



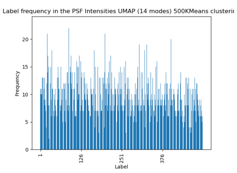
(c) 250KMeans LP coefficients



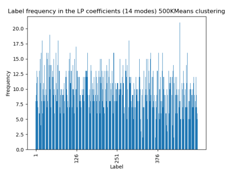
(d) 250KMeans Output Fluxes



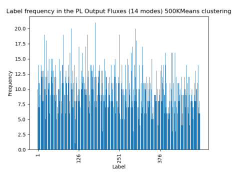
(a) 500KMeans for Zernike coefficients



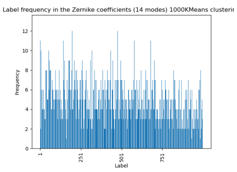
(b) 500KMeans PSF Intensities UMAP



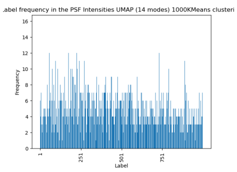
(c) 500KMeans LP coefficients



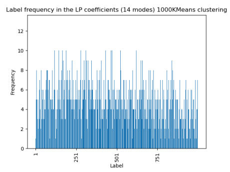
(d) 500KMeans Output Fluxes



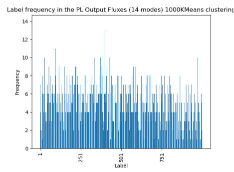
(a) 1000KMeans for Zernike coefficients



(b) 1000KMeans PSF Intensities UMAP

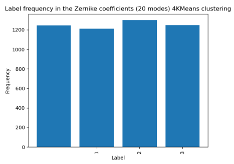


(c) 1000KMeans LP coefficients

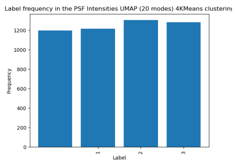


(d) 1000KMeans Output Fluxes

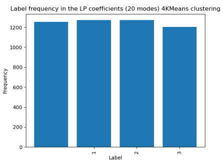
17.2.5 20 Zernike modes datasets clusters densities



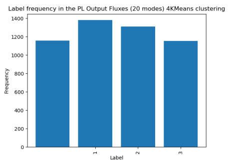
(a) 4KMeans for Zernike coefficients



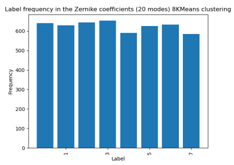
(b) 4KMeans PSF Intensities UMAP



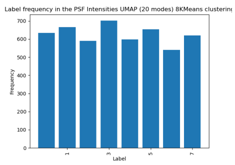
(c) 4KMeans LP coefficients



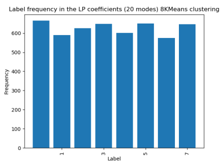
(d) 4KMeans Output Fluxes



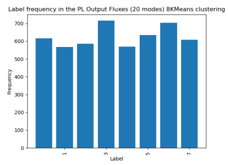
(a) 8KMeans for Zernike coefficients



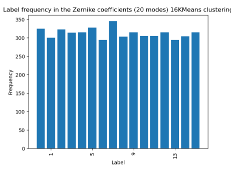
(b) 8KMeans PSF Intensities UMAP



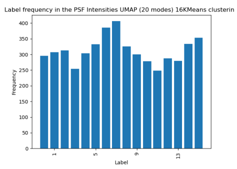
(c) 8KMeans LP coefficients



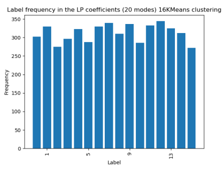
(d) 8KMeans Output Fluxes



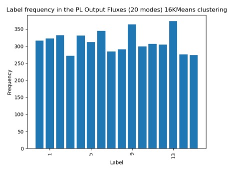
(a) 16KMeans for Zernike coefficients



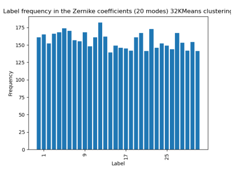
(b) 16KMeans PSF Intensities UMAP



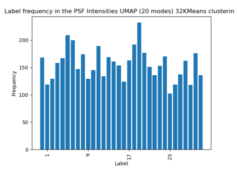
(c) 16KMeans LP coefficients



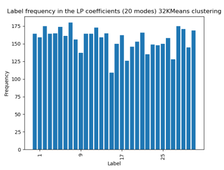
(d) 16KMeans Output Fluxes



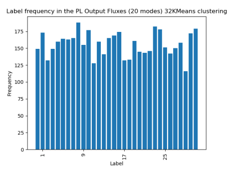
(a) 32KMeans for Zernike coefficients



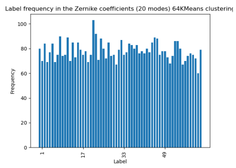
(b) 32KMeans PSF Intensities UMAP

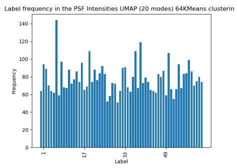


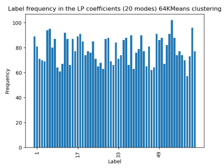
(c) 32KMeans LP coefficients

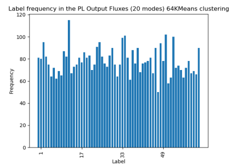


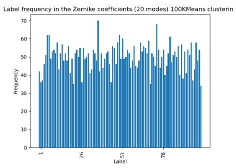
(d) 32KMeans Output Fluxes

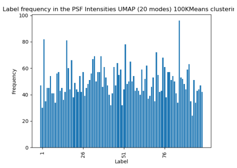

(a) 64KMeans for Zernike coefficients

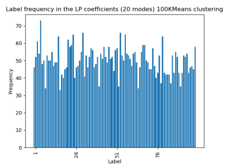

(b) 64KMeans PSF Intensities UMAP

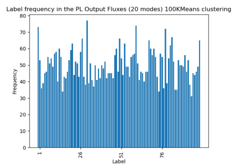

(c) 64KMeans LP coefficients

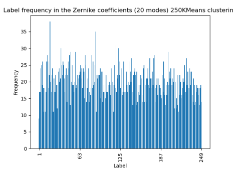

(d) 64KMeans Output Fluxes

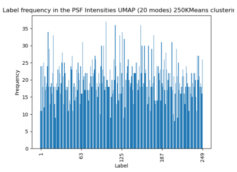

(a) 100KMeans for Zernike coefficients

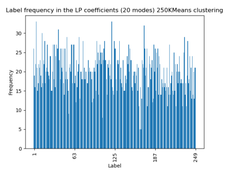

(b) 100KMeans PSF Intensities UMAP

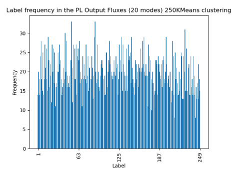

(c) 100KMeans LP coefficients

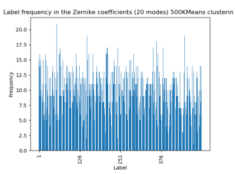

(d) 100KMeans Output Fluxes

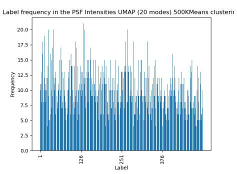

(a) 250KMeans for Zernike coefficients

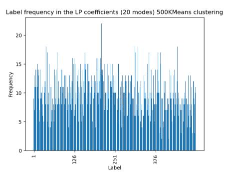

(b) 250KMeans PSF Intensities UMAP

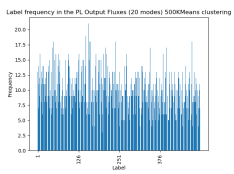

(c) 250KMeans LP coefficients

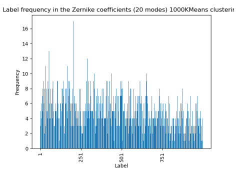

(d) 250KMeans Output Fluxes


(a) 500KMeans for Zernike coefficients

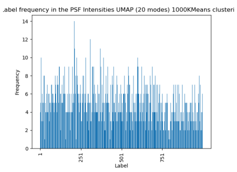

(b) 500KMeans PSF Intensities UMAP


(c) 500KMeans LP coefficients

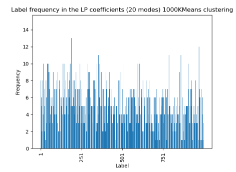

(d) 500KMeans Output Fluxes



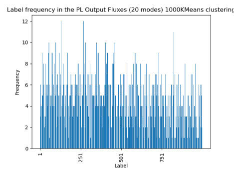
(a) 1000KMeans for Zernike coefficients



(b) 1000KMeans PSF Intensities UMAP

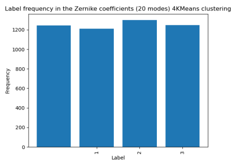


(c) 1000KMeans LP coefficients

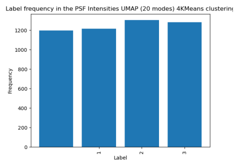


(d) 1000KMeans Output Fluxes

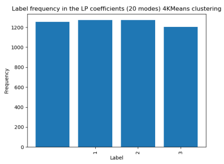
17.2.6 27 Zernike modes datasets clusters densities



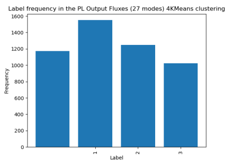
(a) 4KMeans for Zernike coefficients



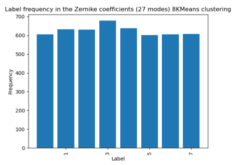
(b) 4KMeans PSF Intensities UMAP



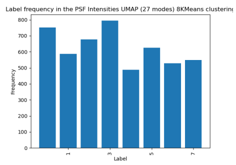
(c) 4KMeans LP coefficients



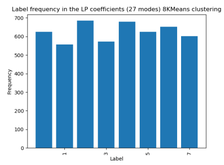
(d) 4KMeans Output Fluxes



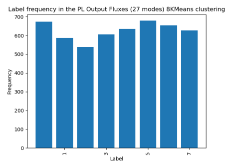
(a) 8KMeans for Zernike coefficients



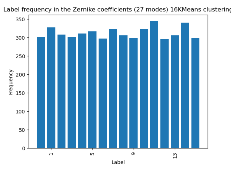
(b) 8KMeans PSF Intensities UMAP



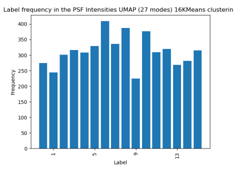
(c) 8KMeans LP coefficients



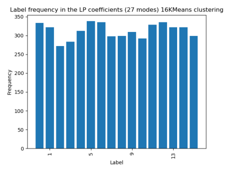
(d) 8KMeans Output Fluxes



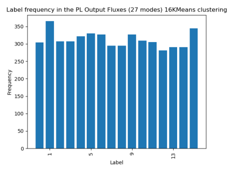
(a) 16KMeans for Zernike coefficients



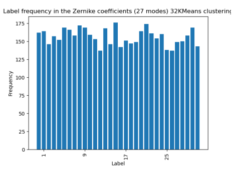
(b) 16KMeans PSF Intensities UMAP



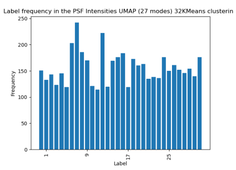
(c) 16KMeans LP coefficients



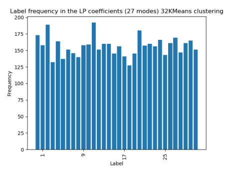
(d) 16KMeans Output Fluxes



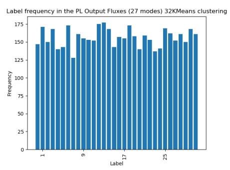
(a) 32KMeans for Zernike coefficients



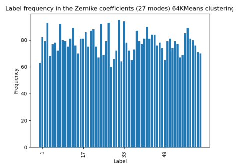
(b) 32KMeans PSF Intensities UMAP



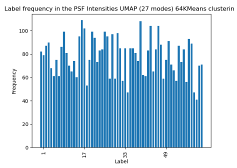
(c) 32KMeans LP coefficients



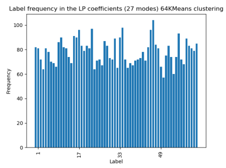
(d) 32KMeans Output Fluxes



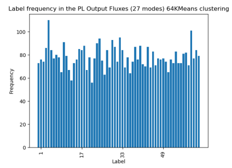
(a) 64KMeans for Zernike coefficients



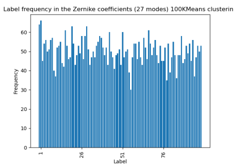
(b) 64KMeans PSF Intensities UMAP



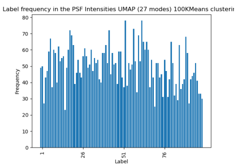
(c) 64KMeans LP coefficients



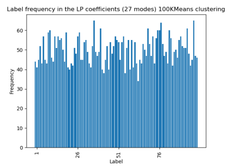
(d) 64KMeans Output Fluxes



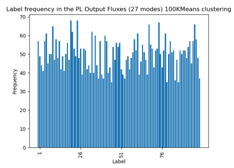
(a) 100KMeans for Zernike coefficients



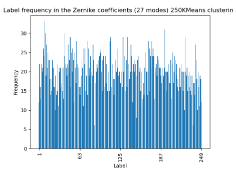
(b) 100KMeans PSF Intensities UMAP



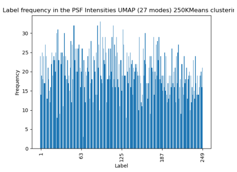
(c) 100KMeans LP coefficients



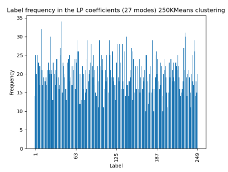
(d) 100KMeans Output Fluxes



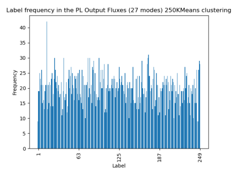
(a) 250KMeans for Zernike coefficients



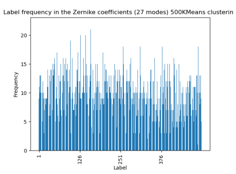
(b) 250KMeans PSF Intensities UMAP



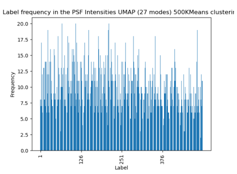
(c) 250KMeans LP coefficients



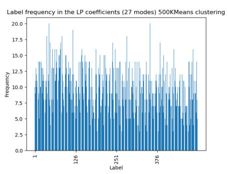
(d) 250KMeans Output Fluxes



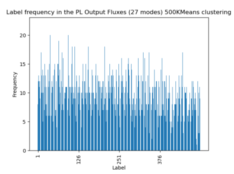
(a) 500KMeans for Zernike coefficients



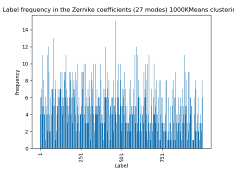
(b) 500KMeans PSF Intensities UMAP



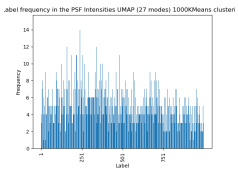
(c) 500KMeans LP coefficients



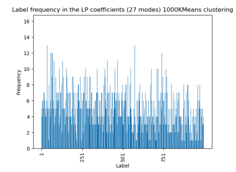
(d) 500KMeans Output Fluxes



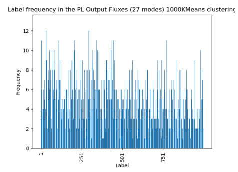
(a) 1000KMeans for Zernike coefficients



(b) 1000KMeans PSF Intensities UMAP

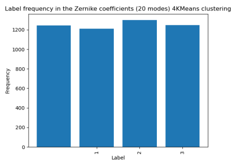


(c) 1000KMeans LP coefficients

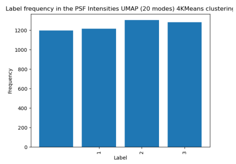


(d) 1000KMeans Output Fluxes

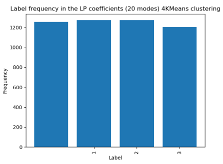
17.2.7 35 Zernike modes datasets clusters densities



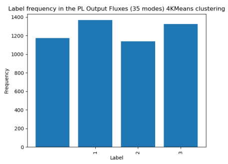
(a) 4KMeans for Zernike coefficients



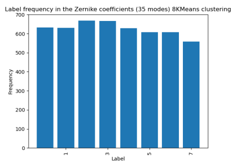
(b) 4KMeans PSF Intensities UMAP



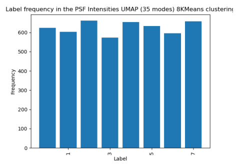
(c) 4KMeans LP coefficients



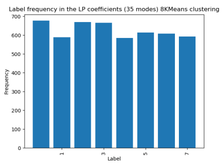
(d) 4KMeans Output Fluxes



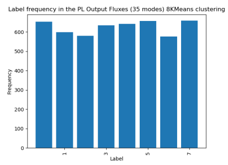
(a) 8KMeans for Zernike coefficients



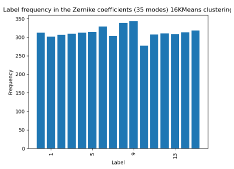
(b) 8KMeans PSF Intensities UMAP



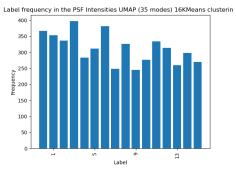
(c) 8KMeans LP coefficients



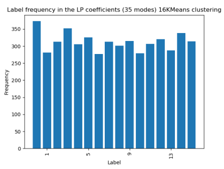
(d) 8KMeans Output Fluxes



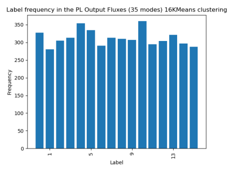
(a) 16KMeans for Zernike coefficients



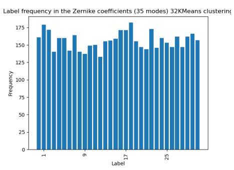
(b) 16KMeans PSF Intensities UMAP



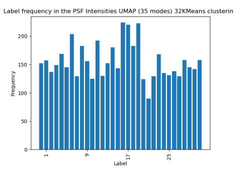
(c) 16KMeans LP coefficients



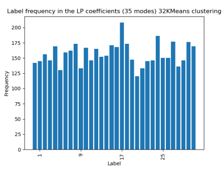
(d) 16KMeans Output Fluxes



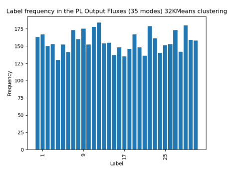
(a) 32KMeans for Zernike coefficients



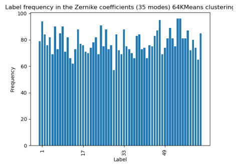
(b) 32KMeans PSF Intensities UMAP



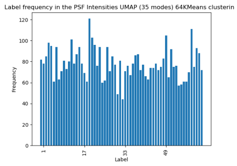
(c) 32KMeans LP coefficients



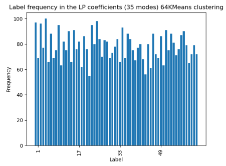
(d) 32KMeans Output Fluxes



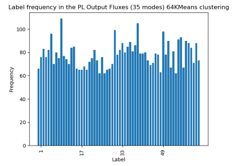
(a) 64KMeans for Zernike coefficients



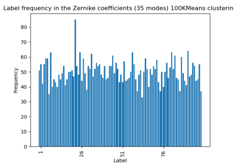
(b) 64KMeans PSF Intensities UMAP



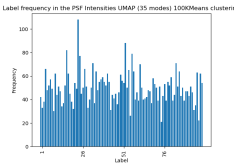
(c) 64KMeans LP coefficients



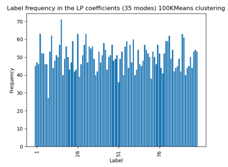
(d) 64KMeans Output Fluxes



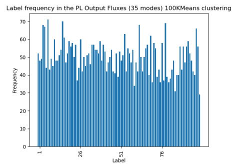
(a) 100KMeans for Zernike coefficients



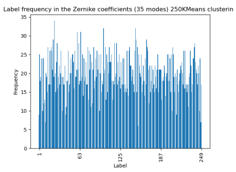
(b) 100KMeans PSF Intensities UMAP



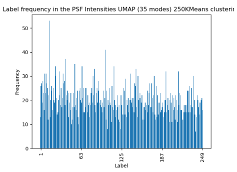
(c) 100KMeans LP coefficients



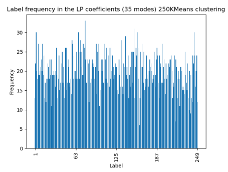
(d) 100KMeans Output Fluxes



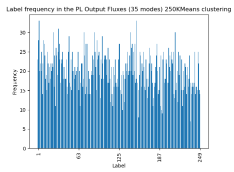
(a) 250KMeans for Zernike coefficients



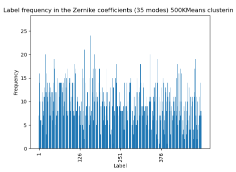
(b) 250KMeans PSF Intensities UMAP



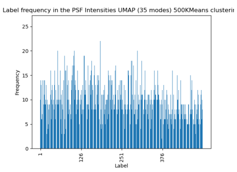
(c) 250KMeans LP coefficients



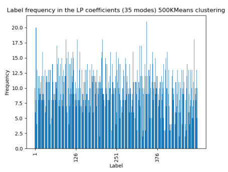
(d) 250KMeans Output Fluxes



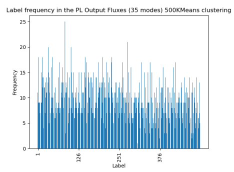
(a) 500KMeans for Zernike coefficients



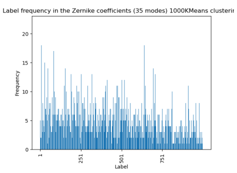
(b) 500KMeans PSF Intensities UMAP



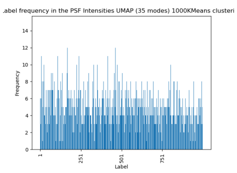
(c) 500KMeans LP coefficients



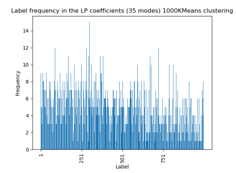
(d) 500KMeans Output Fluxes



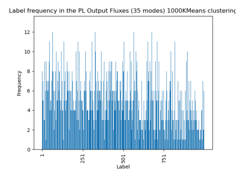
(a) 1000KMeans for Zernike coefficients



(b) 1000KMeans PSF Intensities UMAP

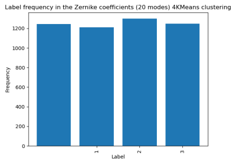


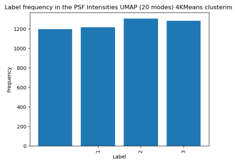
(c) 1000KMeans LP coefficients

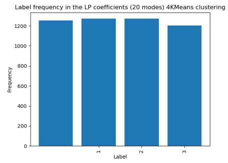


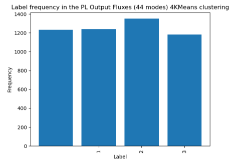
(d) 1000KMeans Output Fluxes

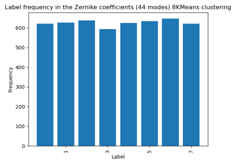
17.2.8 44 Zernike modes datasets clusters densities

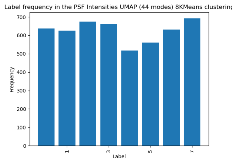

(a) 4KMeans for Zernike coefficients

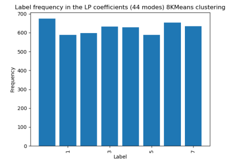

(b) 4KMeans PSF Intensities UMAP

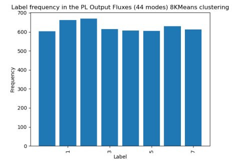

(c) 4KMeans LP coefficients

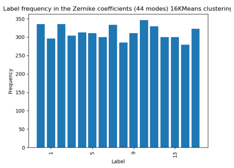

(d) 4KMeans Output Fluxes

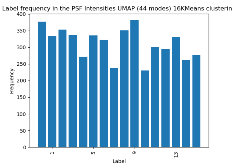

(a) 8KMeans for Zernike coefficients

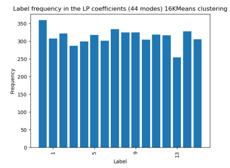

(b) 8KMeans PSF Intensities UMAP

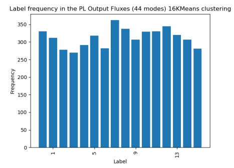

(c) 8KMeans LP coefficients

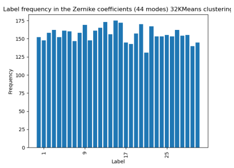

(d) 8KMeans Output Fluxes

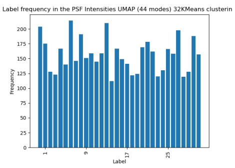

(a) 16KMeans for Zernike coefficients

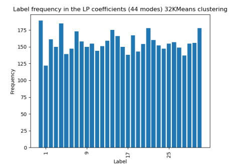

(b) 16KMeans PSF Intensities UMAP

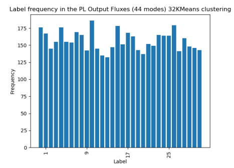

(c) 16KMeans LP coefficients

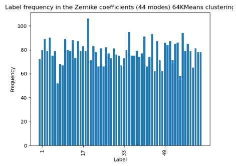

(d) 16KMeans Output Fluxes


(a) 32KMeans for Zernike coefficients

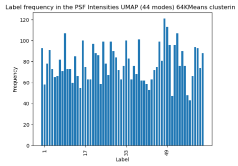

(b) 32KMeans PSF Intensities UMAP


(c) 32KMeans LP coefficients

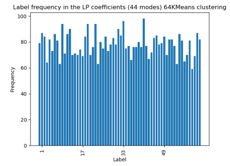

(d) 32KMeans Output Fluxes



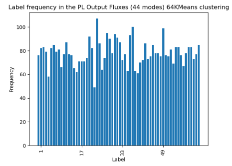
(a) 64KMeans for Zernike coefficients



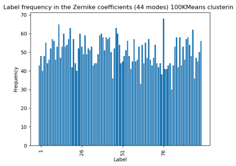
(b) 64KMeans PSF Intensities UMAP



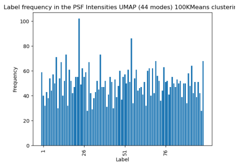
(c) 64KMeans LP coefficients



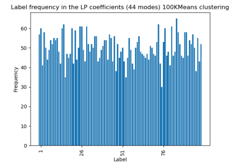
(d) 64KMeans Output Fluxes



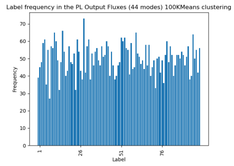
(a) 100KMeans for Zernike coefficients



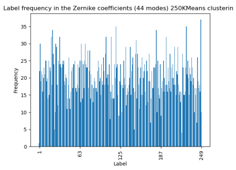
(b) 100KMeans PSF Intensities UMAP



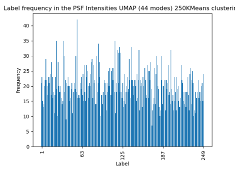
(c) 100KMeans LP coefficients



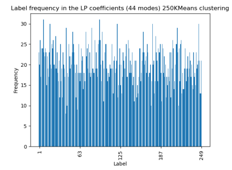
(d) 100KMeans Output Fluxes



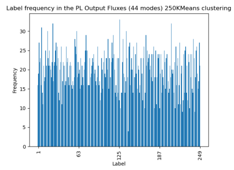
(a) 250KMeans for Zernike coefficients



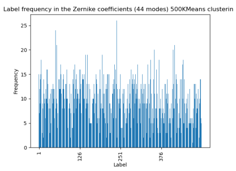
(b) 250KMeans PSF Intensities UMAP



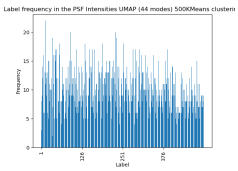
(c) 250KMeans LP coefficients



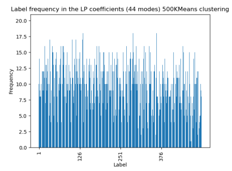
(d) 250KMeans Output Fluxes



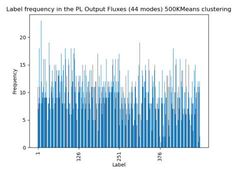
(a) 500KMeans for Zernike coefficients



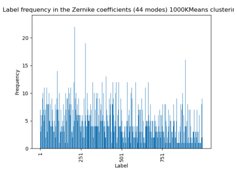
(b) 500KMeans PSF Intensities UMAP



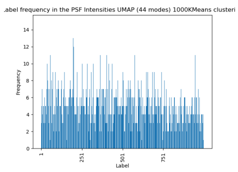
(c) 500KMeans LP coefficients



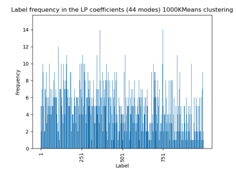
(d) 500KMeans Output Fluxes



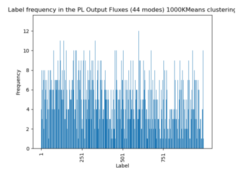
(a) 1000KMeans for Zernike coefficients



(b) 1000KMeans PSF Intensities UMAP



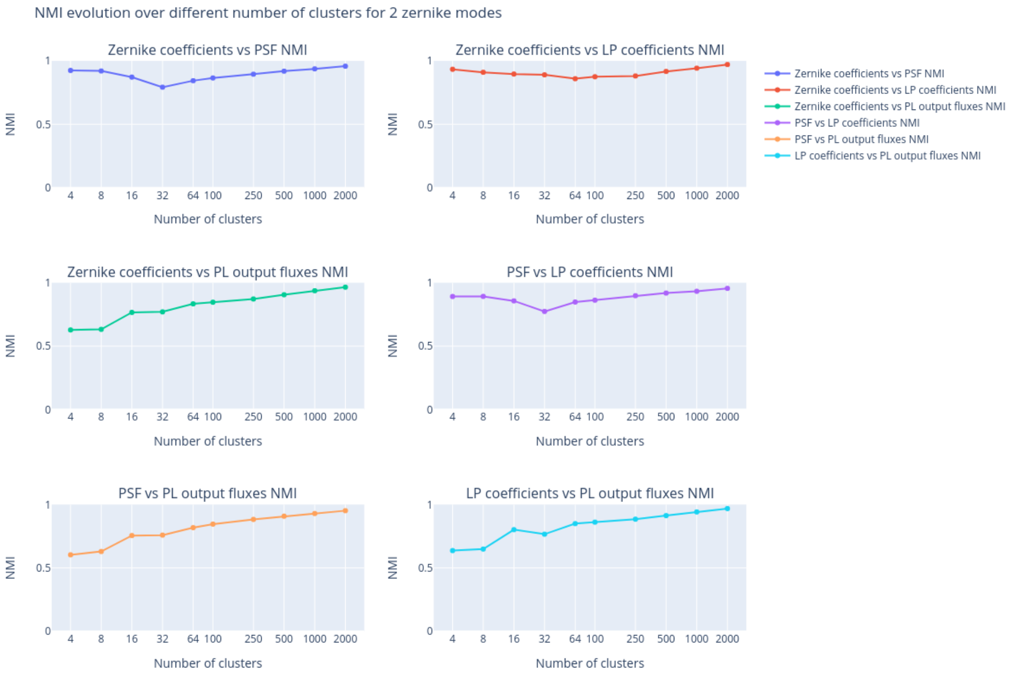
(c) 1000KMeans LP coefficients



(d) 1000KMeans Output Fluxes

17.3 Normalized Mutual Information evolution

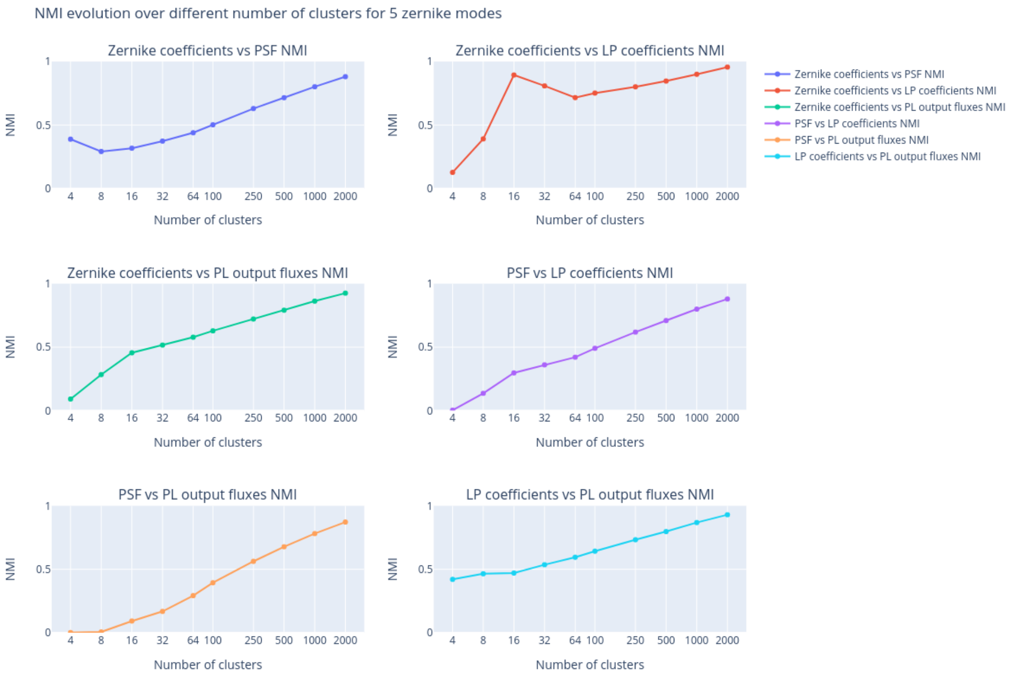
17.3.1 NMI evolution over number of clusters for 2 zernike mode related datasets



Clusters	Z vs PSF	Z vs LP	Z vs PL	PSF vs LP	PSF vs PL	LP vs PL
4	0.925	0.934	0.628	0.892	0.604	0.637
8	0.921	0.910	0.633	0.892	0.630	0.649
16	0.872	0.897	0.766	0.857	0.756	0.802
32	0.793	0.891	0.770	0.774	0.759	0.766
64	0.844	0.860	0.834	0.848	0.818	0.850
100	0.865	0.875	0.846	0.863	0.846	0.861
250	0.895	0.881	0.872	0.898	0.883	0.885
500	0.919	0.917	0.905	0.919	0.908	0.913
1000	0.937	0.943	0.937	0.933	0.930	0.942
2000	0.959	0.972	0.965	0.956	0.952	0.969

Table 93: NMI Analysis for Different Numbers of Clusters

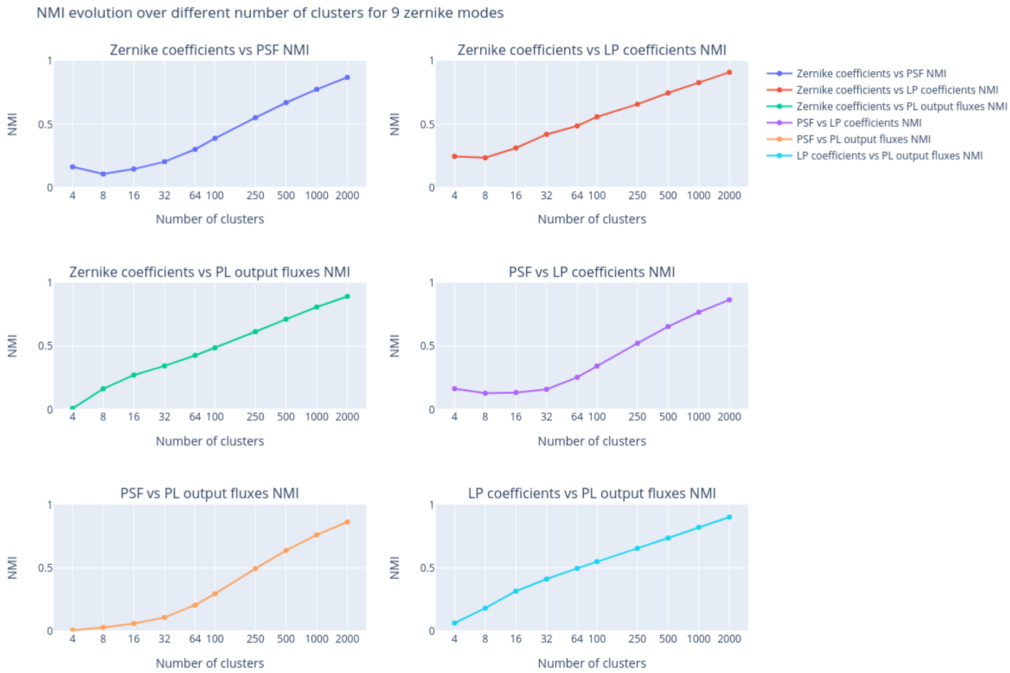
17.3.2 NMI evolution over number of clusters for 5 zernike mode related datasets



Clusters	Z vs PSF	Z vs LP	Z vs PL	PSF vs LP	PSF vs PL	LP vs PL
4	0.3884	0.1268	0.0914	0.0033	0.0016	0.4207
8	0.2913	0.3905	0.2838	0.1361	0.0064	0.4649
16	0.3172	0.8939	0.4555	0.2970	0.0925	0.4705
32	0.3731	0.8080	0.5162	0.3606	0.1683	0.5361
64	0.4390	0.7155	0.5783	0.4208	0.2927	0.5950
100	0.5018	0.7515	0.6286	0.4911	0.3940	0.6427
250	0.6296	0.8008	0.7217	0.6187	0.5626	0.7328
500	0.7151	0.8462	0.7910	0.7095	0.6774	0.7972
1000	0.8012	0.8996	0.8626	0.7995	0.7812	0.8683
2000	0.8807	0.9562	0.9255	0.8795	0.8723	0.9299

Table 94: NMI Analysis for Different Numbers of Clusters

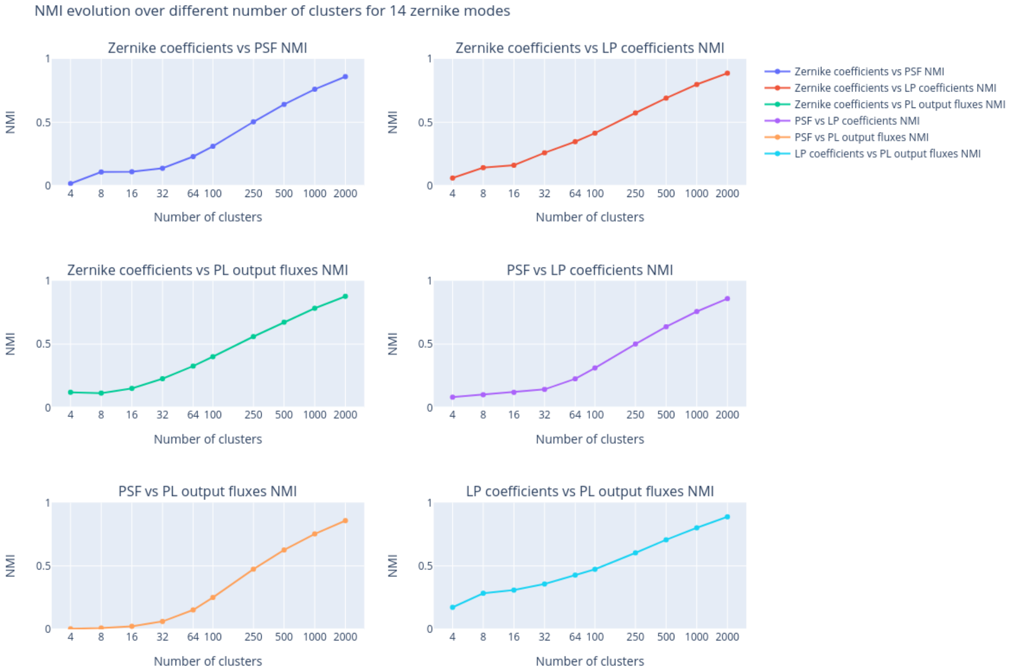
17.3.3 NMI evolution over number of clusters for 9 zernike mode related datasets



Clusters	Z vs PSF	Z vs LP	Z vs PL	PSF vs LP	PSF vs PL	LP vs PL
4	0.164	0.247	0.009	0.165	0.009	0.065
8	0.108	0.236	0.164	0.128	0.031	0.183
16	0.147	0.313	0.272	0.134	0.061	0.317
32	0.205	0.421	0.344	0.160	0.110	0.413
64	0.303	0.487	0.427	0.254	0.207	0.497
100	0.389	0.559	0.487	0.343	0.296	0.550
250	0.552	0.658	0.614	0.522	0.495	0.655
500	0.671	0.747	0.712	0.655	0.638	0.736
1000	0.776	0.829	0.809	0.768	0.761	0.820
2000	0.870	0.910	0.892	0.866	0.862	0.902

Table 95: NMI Analysis for Different Numbers of Clusters

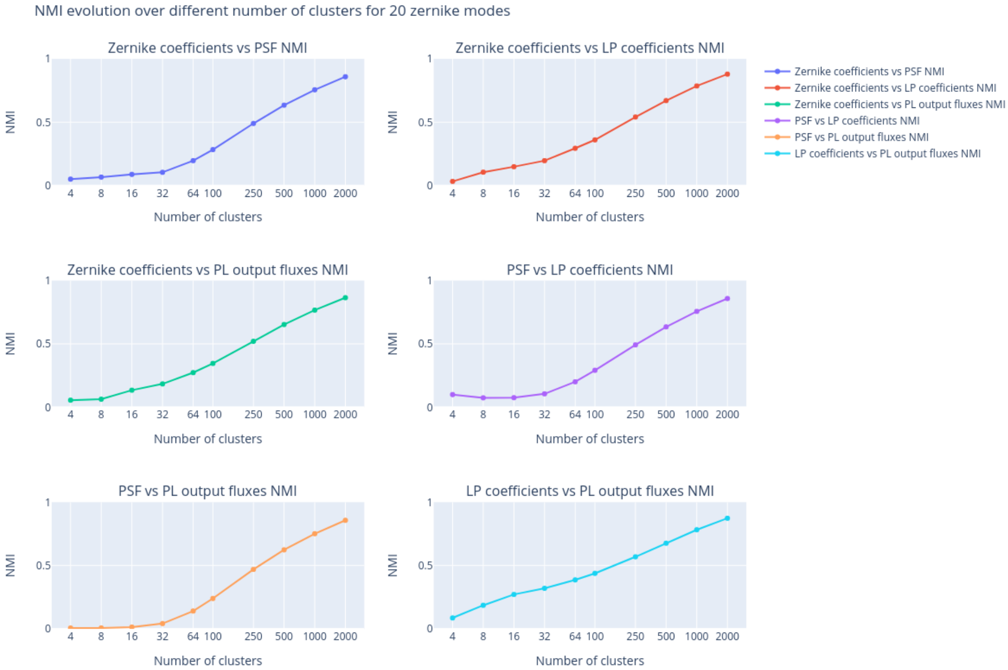
17.3.4 NMI evolution over number of clusters for 14 zernike mode related datasets



Clusters	Z vs PSF	Z vs LP	Z vs PL	PSF vs LP	PSF vs PL	LP vs PL
4	0.017	0.060	0.120	0.082	0.004	0.174
8	0.108	0.142	0.114	0.101	0.010	0.285
16	0.110	0.161	0.151	0.120	0.024	0.310
32	0.137	0.259	0.227	0.143	0.062	0.358
64	0.230	0.347	0.327	0.226	0.153	0.428
100	0.311	0.414	0.401	0.312	0.252	0.474
250	0.504	0.574	0.560	0.501	0.475	0.604
500	0.641	0.691	0.673	0.637	0.626	0.706
1000	0.761	0.799	0.783	0.757	0.753	0.801
2000	0.860	0.888	0.877	0.859	0.858	0.888

Table 96: NMI Analysis for Different Numbers of Clusters

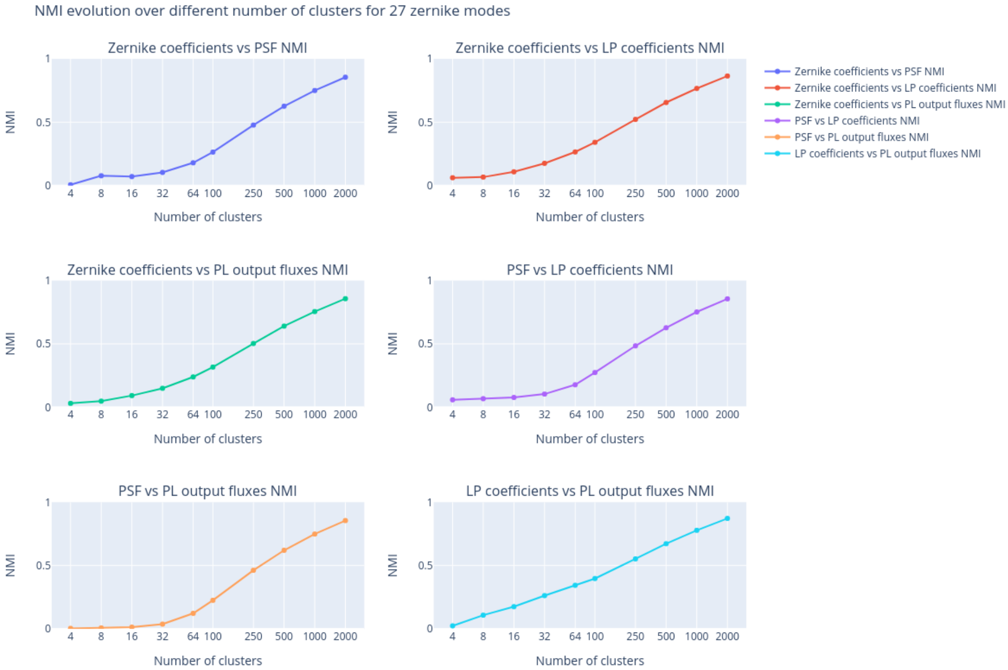
17.3.5 NMI evolution over number of clusters for 20 zernike mode related datasets



Clusters	Z vs PSF	Z vs LP	Z vs PL	PSF vs LP	PSF vs PL	LP vs PL
4	0.051	0.033	0.056	0.100	0.007	0.087
8	0.067	0.106	0.063	0.074	0.007	0.186
16	0.088	0.149	0.134	0.075	0.014	0.272
32	0.105	0.196	0.184	0.106	0.042	0.320
64	0.196	0.294	0.273	0.201	0.141	0.387
100	0.284	0.361	0.346	0.292	0.240	0.438
250	0.490	0.541	0.520	0.492	0.470	0.569
500	0.635	0.671	0.653	0.635	0.624	0.676
1000	0.756	0.786	0.766	0.757	0.752	0.783
2000	0.859	0.879	0.865	0.858	0.857	0.873

Table 97: NMI Analysis for Different Numbers of Clusters

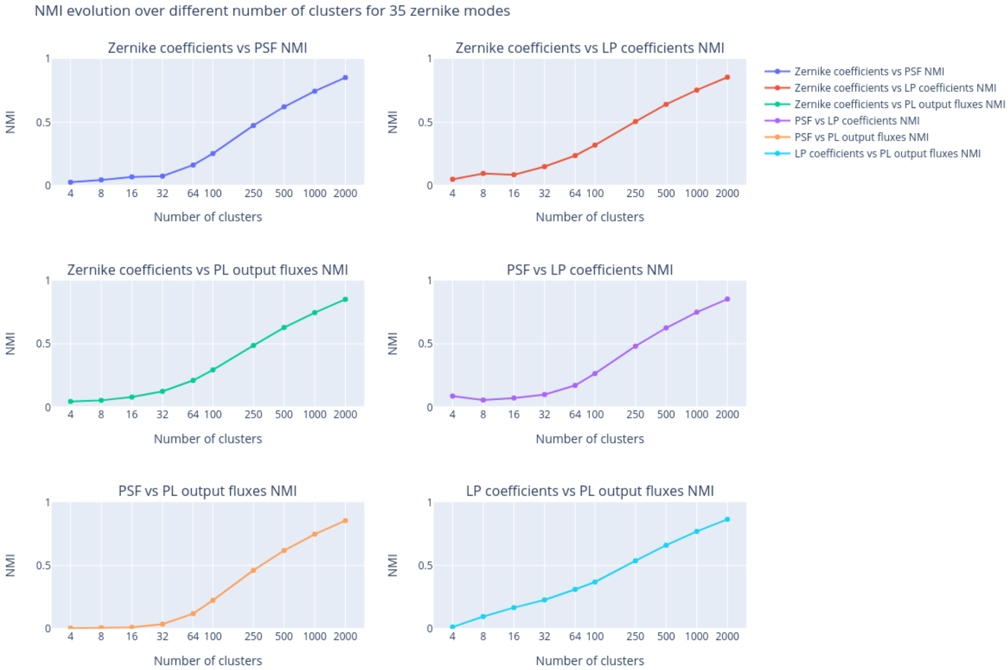
17.3.6 NMI evolution over number of clusters for 27 zernike mode related datasets



Clusters	Z vs PSF	Z vs LP	Z vs PL	PSF vs LP	PSF vs PL	LP vs PL
4	0.006	0.062	0.030	0.059	0.004	0.024
8	0.077	0.067	0.048	0.067	0.007	0.108
16	0.071	0.108	0.092	0.077	0.014	0.175
32	0.104	0.175	0.149	0.104	0.037	0.263
64	0.180	0.265	0.239	0.177	0.123	0.344
100	0.264	0.341	0.317	0.274	0.226	0.397
250	0.477	0.522	0.503	0.485	0.463	0.552
500	0.626	0.656	0.640	0.627	0.620	0.672
1000	0.750	0.767	0.755	0.752	0.749	0.778
2000	0.855	0.865	0.857	0.856	0.855	0.872

Table 98: NMI Analysis for Different Numbers of Clusters

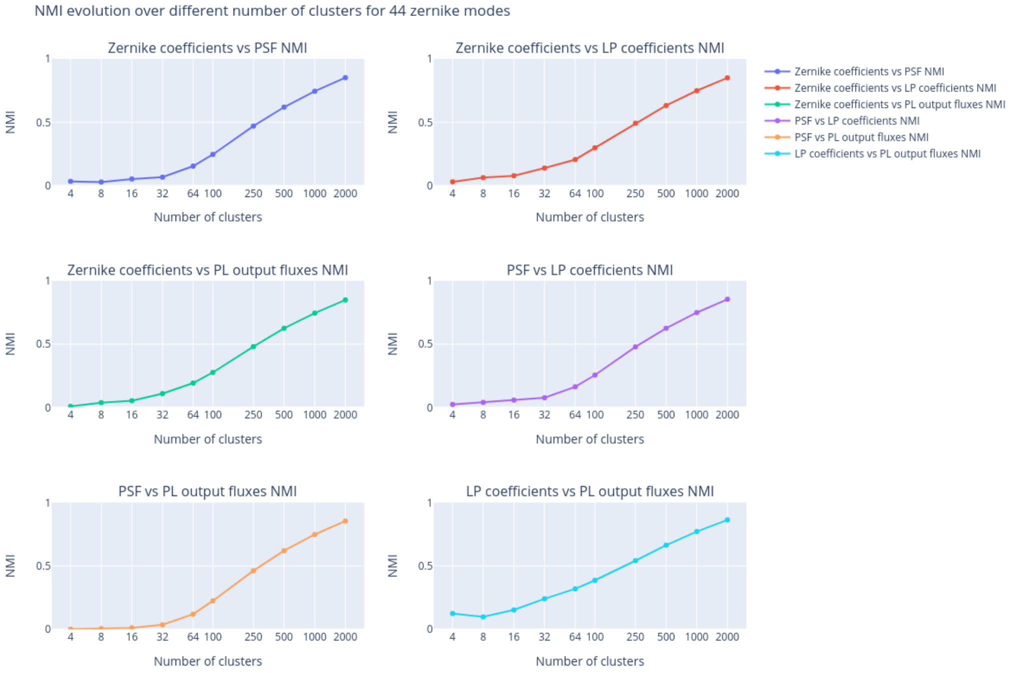
17.3.7 NMI evolution over number of clusters for 35 zernike mode related datasets



Clusters	Z vs PSF	Z vs LP	Z vs PL	PSF vs LP	PSF vs PL	LP vs PL
4	0.026	0.049	0.046	0.088	0.006	0.015
8	0.045	0.095	0.054	0.056	0.009	0.097
16	0.068	0.085	0.080	0.073	0.013	0.168
32	0.074	0.149	0.125	0.100	0.037	0.229
64	0.162	0.236	0.212	0.172	0.120	0.312
100	0.253	0.319	0.295	0.265	0.226	0.370
250	0.474	0.506	0.487	0.482	0.462	0.538
500	0.622	0.641	0.629	0.626	0.619	0.660
1000	0.745	0.754	0.747	0.750	0.748	0.769
2000	0.853	0.855	0.852	0.854	0.855	0.865

Table 99: NMI Analysis for Different Numbers of Clusters

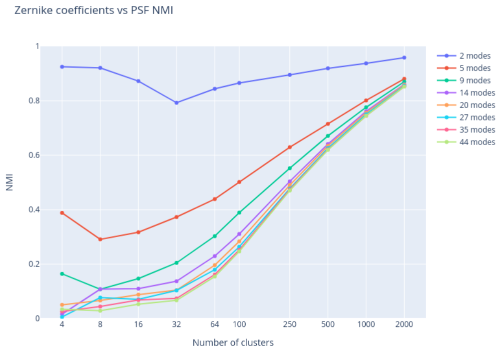
17.3.8 NMI evolution over number of clusters for 44 zernike mode related datasets



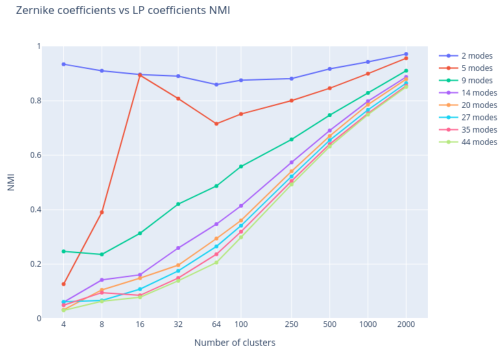
Clusters	Z vs PSF	Z vs LP	Z vs PL	PSF vs LP	PSF vs PL	LP vs PL
4	0.034	0.031	0.009	0.024	0.002	0.125
8	0.029	0.064	0.039	0.040	0.006	0.098
16	0.053	0.079	0.054	0.058	0.012	0.153
32	0.067	0.139	0.110	0.077	0.036	0.241
64	0.155	0.206	0.193	0.164	0.120	0.320
100	0.247	0.299	0.277	0.256	0.225	0.387
250	0.470	0.493	0.481	0.478	0.463	0.542
500	0.620	0.632	0.625	0.625	0.621	0.664
1000	0.745	0.749	0.745	0.749	0.749	0.772
2000	0.852	0.851	0.849	0.854	0.855	0.864

Table 100: NMI Analysis for Different Numbers of Clusters

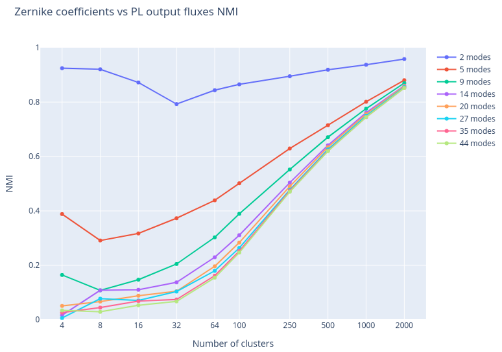
17.3.9 NMI evolution over number of zernike modes



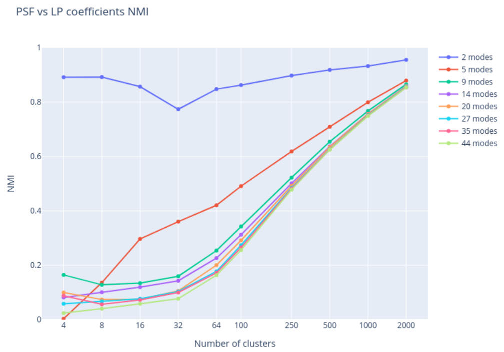
(a) NMI evolution over number of clusters for Zernike coefficients vs PSF



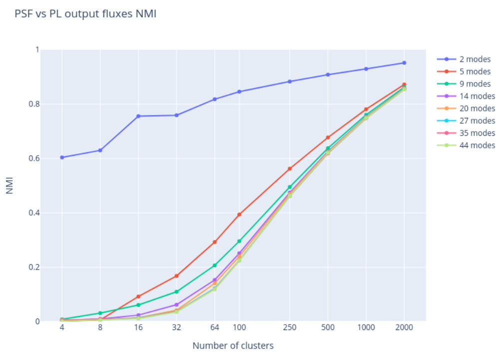
(b) NMI evolution over number of clusters for Zernike coefficients vs LP coefficients



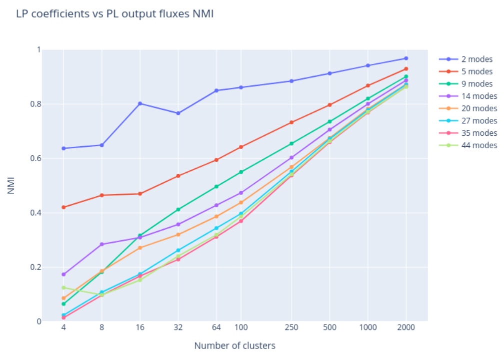
(c) NMI evolution over number of clusters for Zernike coefficients vs PL output fluxes



(d) NMI evolution over number of clusters for PSF vs LP coefficients



(e) NMI evolution over number of clusters for PSF vs PL output fluxes



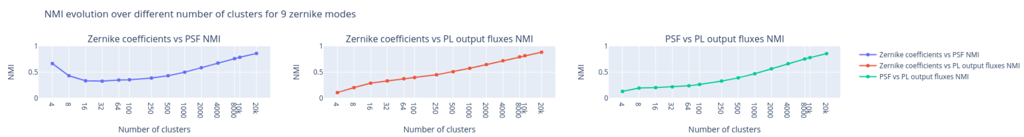
(f) NMI evolution over number of clusters for LP coefficients vs PL output fluxes

17.4 Normalized Mutual Information evolution for a big dataset

This dataset is made from 75000 combinations of 9 zernike modes their RMSE ranging from:

- Modes 2,3 between [-0.8, 0.8]
- Modes 4,5,6 between [-0.6, 0.6]
- Modes 7,8,9,10 between [-0.4, 0.4]

17.4.1 NMI evolution over number of clusters for 9 zernike mode related datasets



Clusters	Z vs PSF	Z vs PL	PSF vs PL
4	0.667	0.109	0.133
8	0.433	0.205	0.196
16	0.336	0.291	0.205
32	0.329	0.334	0.220
64	0.351	0.375	0.240
100	0.356	0.399	0.267
250	0.389	0.452	0.331
500	0.433	0.510	0.393
1000	0.501	0.577	0.472
2000	0.588	0.648	0.567
4000	0.677	0.722	0.664
8000	0.763	0.796	0.755
10000	0.789	0.819	0.782
20000	0.863	0.886	0.859

Table 101: NMI Analysis for Different Numbers of Clusters