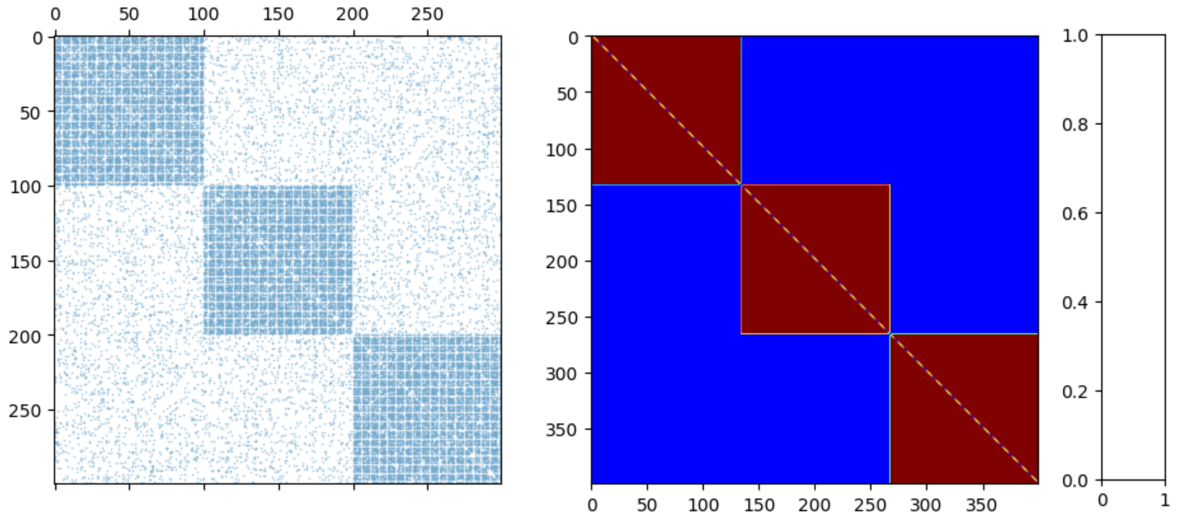
```
%load_ext autoreload
%autoreload 2
```

```
The autoreload extension is already loaded. To reload it, use:
  %reload_ext autoreload
```

```
import numpy as np
import matplotlib.pyplot as plt
from tqdm import tqdm


from pygraphon.graphons import graphon_log, graphon_logit_sum
from pygraphon.utils.utils_graph import edge_density
from pygraphon.graphons.StepGraphon import StepGraphon
from pygraphon.plots.plot_matrices import spy

from pygraphon.estimators import HistogramEstimator, SAS
from pygraphon.norm import MseProbaEdge
```

```
#Set the matrix of theta
graphon_matrix = np.array([[0.8,0.1,0.1],[0.1,0.8,0.1],[0.1,0.1,0.8]])
#set up normalised graphon with graphon
graphon = StepGraphon(graphon_matrix,1/3)
#graphon = graphon_logit_sum


A = graphon.draw(rho = 1,n = 300, exchangeable=False)
norm = MseProbaEdge(n_nodes = 500)

fig, ax = plt.subplots(1, 2, figsize=(10, 5))
fig.subplots_adjust(right=0.9)
cbar_ax = fig.add_axes([0.95, 0.15, 0.05, 0.7])
im = ax[1].imshow(graphon._get_edge_probabilities(400,False,True), cmap='jet')
_ = spy(A, ax=ax[0])
#ax[0].imshow(A.astype(int), cmap='jet')
plt.show()
```

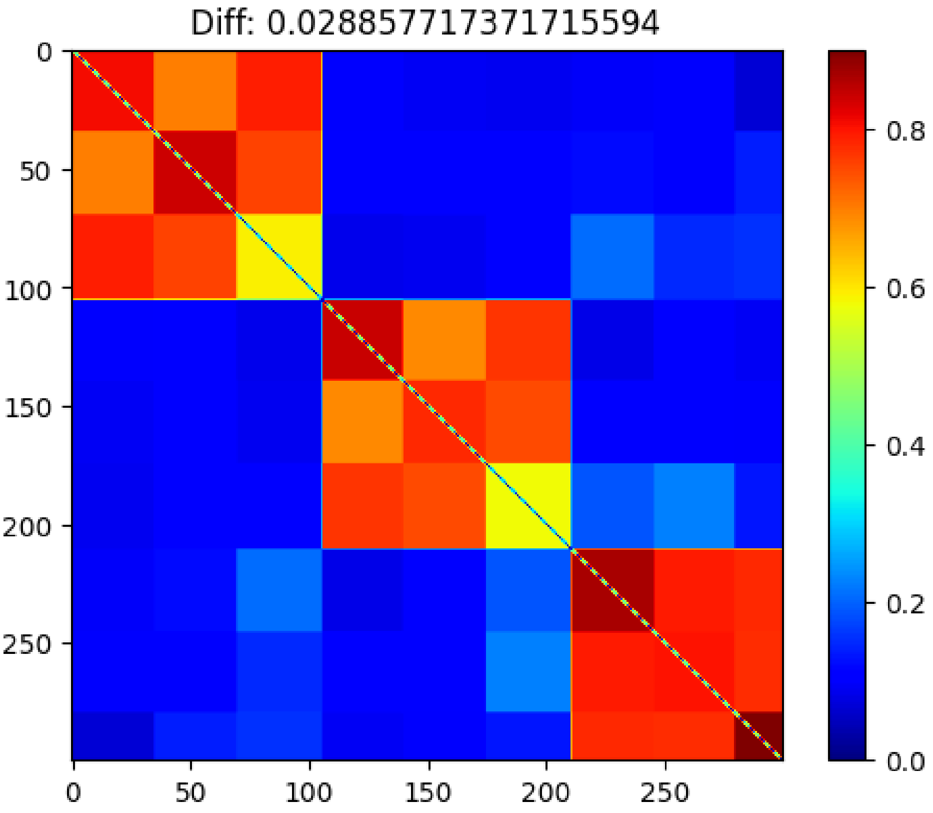


```
hist = HistogramEstimator(method = "matlab")
hist.fit(graph= A, bandwidthHist= None)
diff = norm(graphon, hist)
plt.imshow(hist.get_graphon()._get_edge_probabilities(300,False,True), cmap = "jet")
plt.title(f"Diff: {diff}")
plt.colorbar()
```

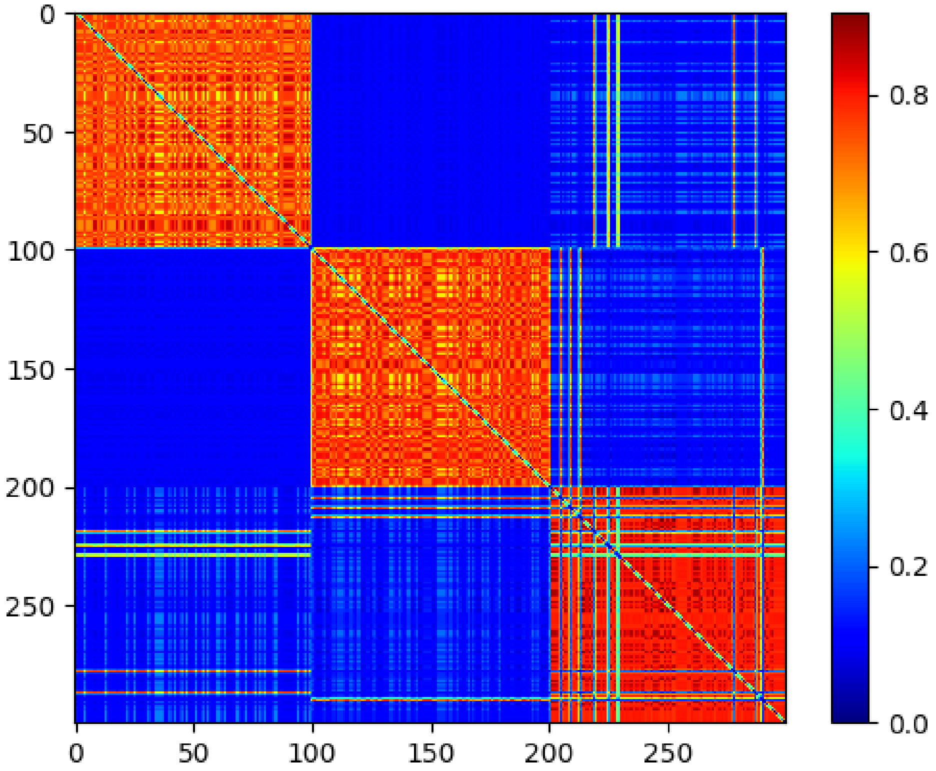
```
Initial log-likelihood: -21244.509161559272
fast normalized ll: -0.473679134036996
Initial normalized log-likelihood: -1.422845700995196
```

```
LL: -0.2730,  4 global improvements: 100%|      | 100/100 [01:35<00:00,  1.04it/s]
```

```
<matplotlib.colorbar.Colorbar at 0x1773dbe50>
```



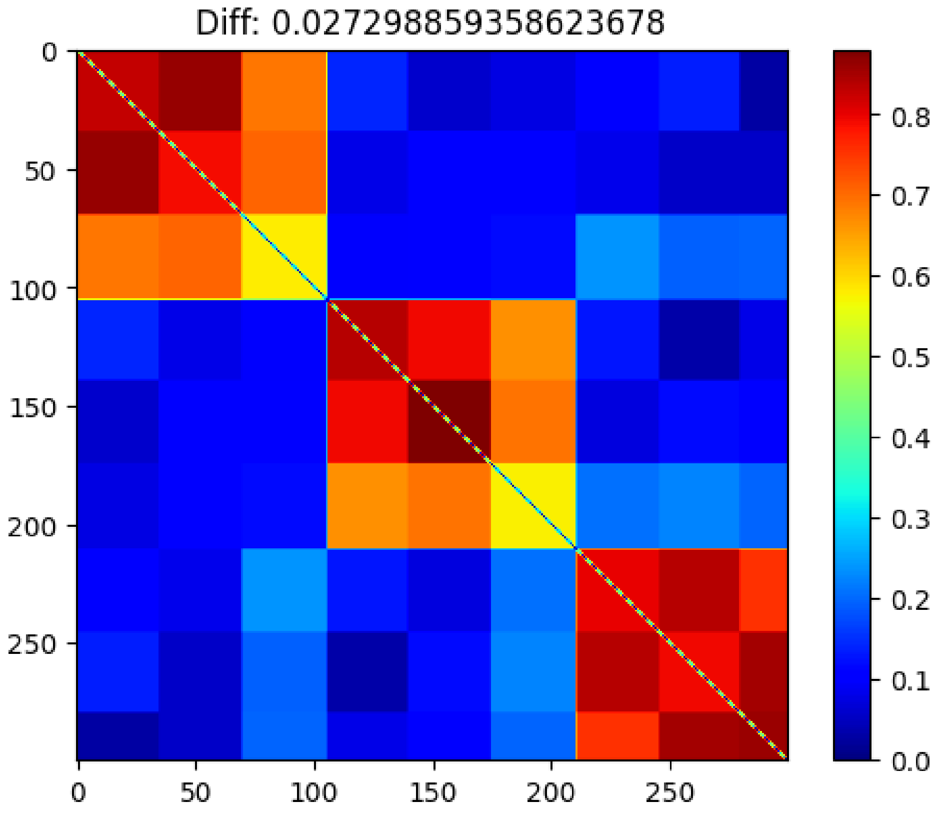
```
plt.imshow(hist.get_edge_connectivity(), cmap = "jet")
plt.colorbar()
plt.show()
```



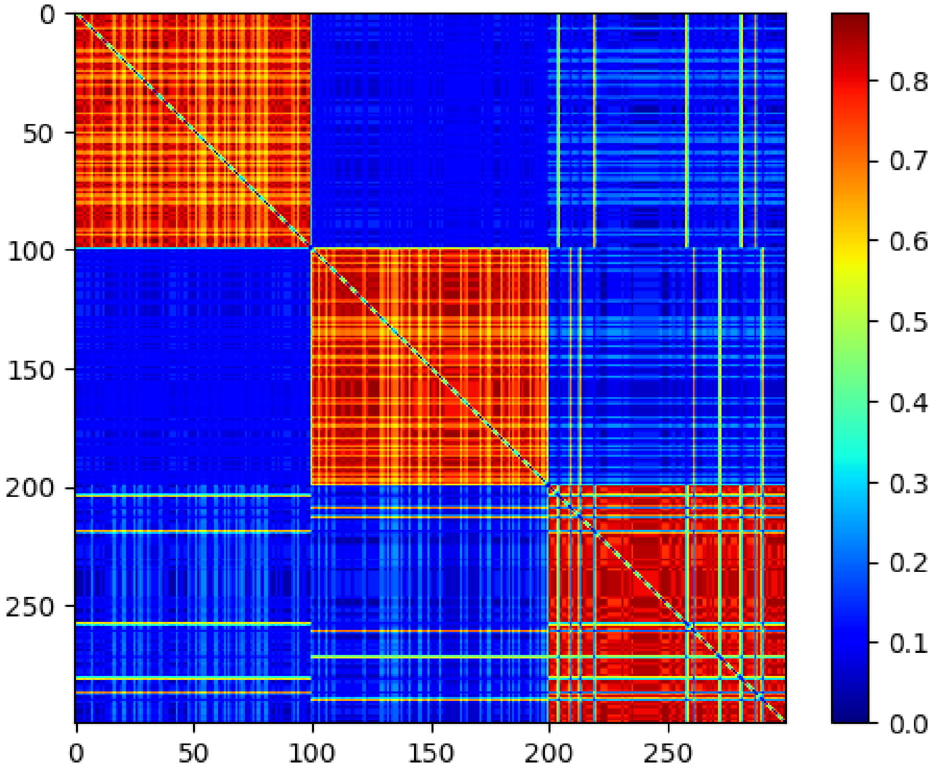
```python
my_hist = HistogramEstimator()
my_hist.fit(graph= A, bandwidthHist= None,tol = 1e-10)
diff = norm(graphon, my_hist)
plt.imshow(my_hist.get_graphon()._get_edge_probabilities(300,False,True), cmap = "jet")
plt.title(f"Diff: {diff}")
plt.colorbar()
```

```
Log likelihood: -0.4084:   6%|          | 6/100 [00:02<00:32,  2.93it/s]
```

```
<matplotlib.colorbar.Colorbar at 0x296c875e0>
```



```
plt.imshow(my_hist.get_edge_connectivity(), cmap = "jet")
plt.colorbar()
plt.show()
```



Test on pollblog dataset

yep that's the only one where we have a number

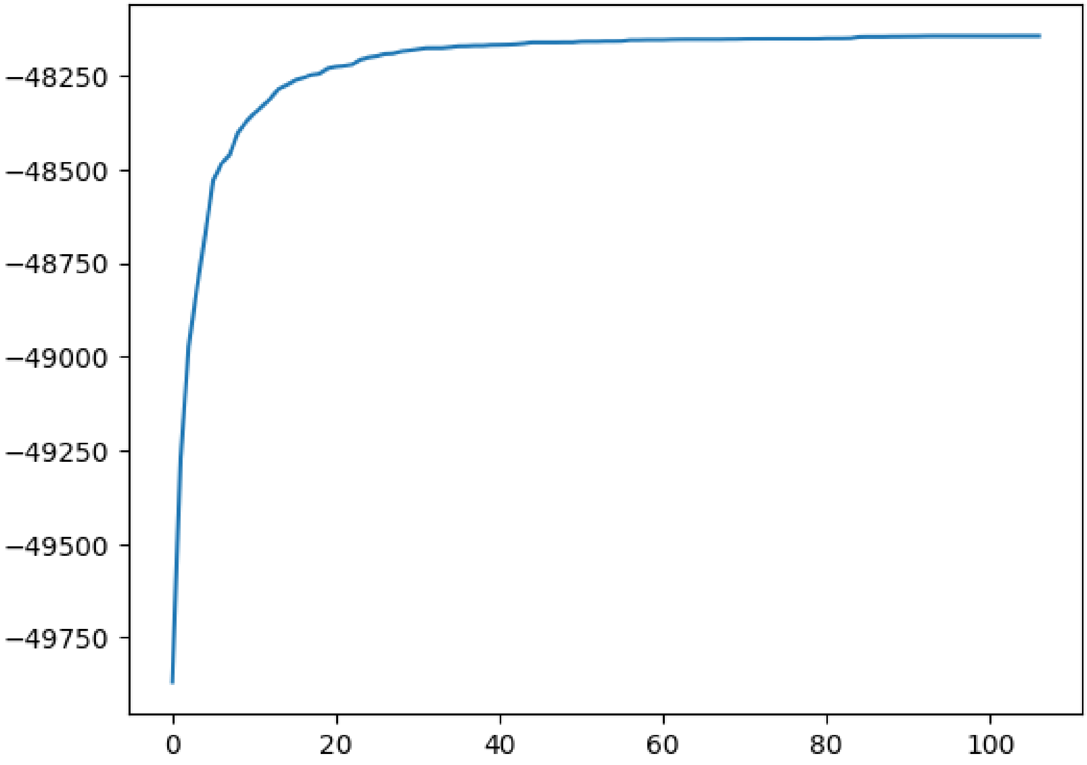
```
import networkx as nx
G = nx.read_edgelist("network.csv/edges.csv", delimiter = ",", create_using=nx.Graph())
A = nx.to_numpy_array(G)
A = A[np.sum(A, axis = 1) > 0].astype(int)
A[np.diag_indices_from(A)] = 0
```

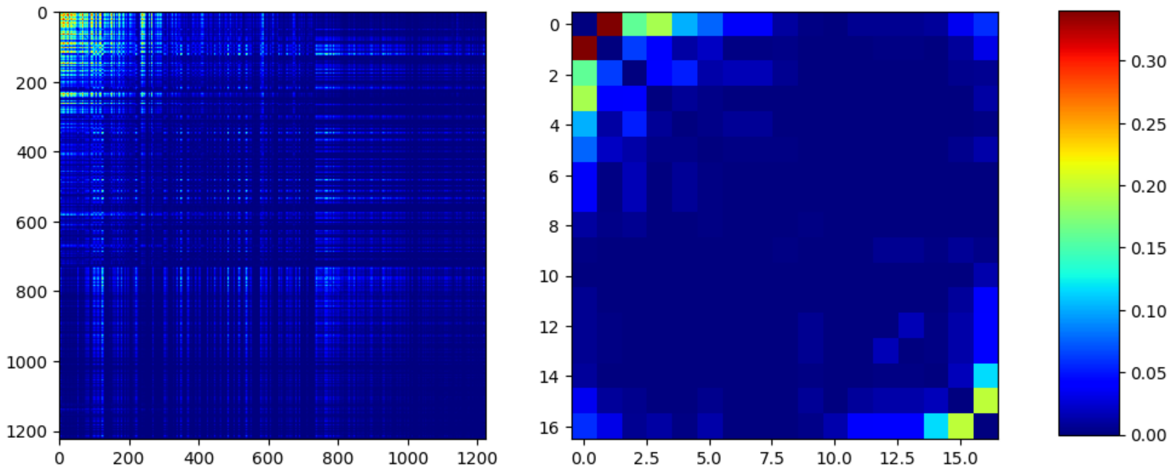
```
my_hist = HistogramEstimator()
my_hist.fit(graph= A, bandwidthHist= None,absTol = 1e-10, past_non_improving= 100)

fig, ax = plt.subplots(1, 2, figsize=(10, 5))
fig.subplots_adjust(right=0.9)
cbar_ax = fig.add_axes([0.95, 0.15, 0.05, 0.7])
im = ax[1].imshow(my_hist.get_graphon()._get_edge_probabilities(17,False,True), cmap='jet'
ax[0].imshow(my_hist.get_edge_connectivity(), cmap='jet')
plt.colorbar(im, cax=cbar_ax)
plt.show()
```

```
h 73
```

```
Log likelihood: -0.0643:  21%|          | 105/500 [00:38<02:23,  2.76it/s]
```

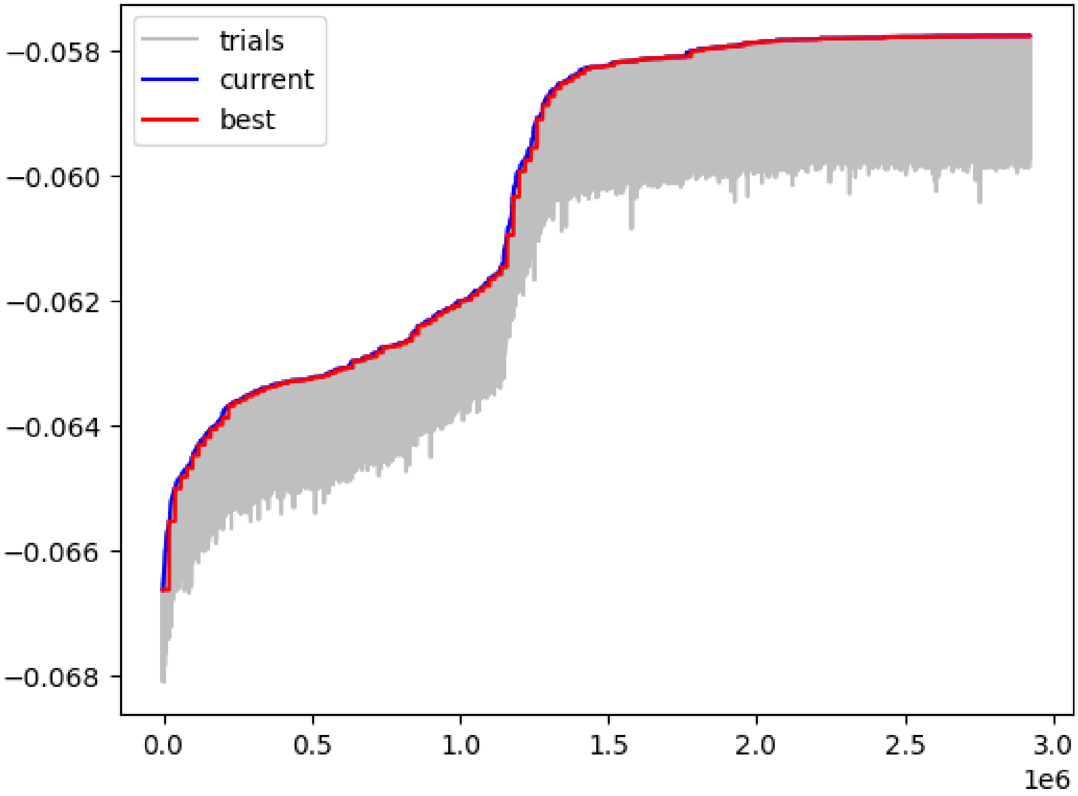
```
Converged with normalized ll: -2.8803
```

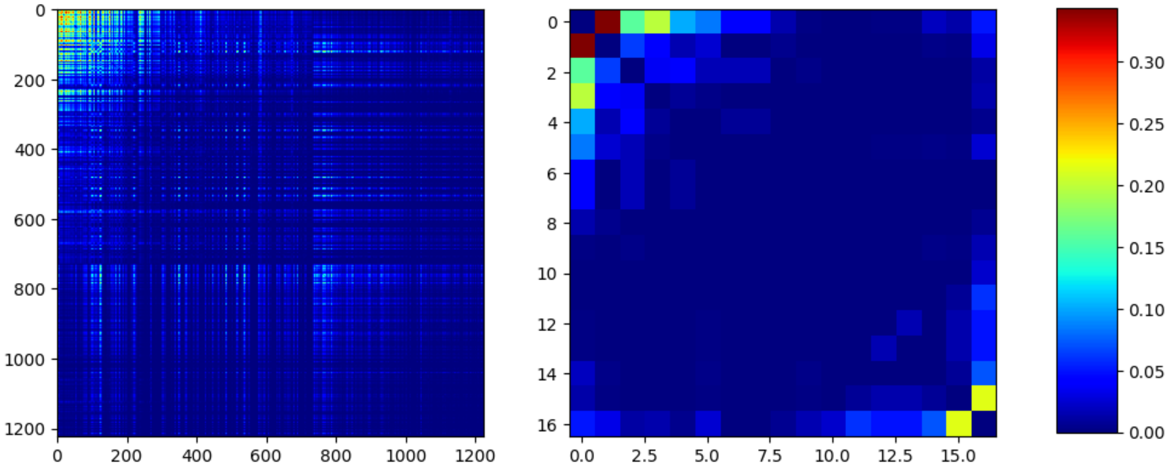




```
hist = HistogramEstimator(method = "matlab")
hist.fit(graph= A, bandwidthHist= None)
fig, ax = plt.subplots(1, 2, figsize=(10, 5))
fig.subplots_adjust(right=0.9)
cbar_ax = fig.add_axes([0.95, 0.15, 0.05, 0.7])
im = ax[1].imshow(hist.get_graphon()._get_edge_probabilities(17,False,True), cmap='jet')
ax[0].imshow(hist.get_edge_connectivity(), cmap='jet')
plt.colorbar(im, cax=cbar_ax)
plt.show()
```

```
h 73
Initial log-likelihood: -49868.09652905405
fast normalized ll: -0.06662617976936341
Initial normalized log-likelihood: -2.9834338336257282

LL: -0.0578,  3 global improvements:  29%|         | 145/500 [02:11<05:21,  1.10it/s]

3 consecutive likelihood improvements less than specified tolerance; quitting now
Best LL: -0.0578
```


trials
current
best
−0.058
−0.060
−0.062
−0.064
−0.066
−0.068
0.0
0.5
1.0
1.5
2.0
2.5
3.0
1e6


0
200
400
600
800
1000
1200
0
200
400
600
800
1000
1200
0
2
4
6
8
10
12
14
16
0.0
2.5
5.0
7.5
10.0
12.5
15.0
0.30
0.25
0.20
0.15
0.10
0.05
0.00