

- Shin's Lab -

Python for Data Visualization

Python for Data Visualization

-Chapter.1 Matplotlib Anatomy -

1-01. Making Figures and Axes

1-02. Axes Customizing

1-03. Titles, Labels and Font Dict

1-04. Ticks and Ticklabels

1-05. Grid

1-06. Spines

1-07. Colors in Matplotlib

1-08. Matplotlib Styles and rcParams

Python for Data Visualization

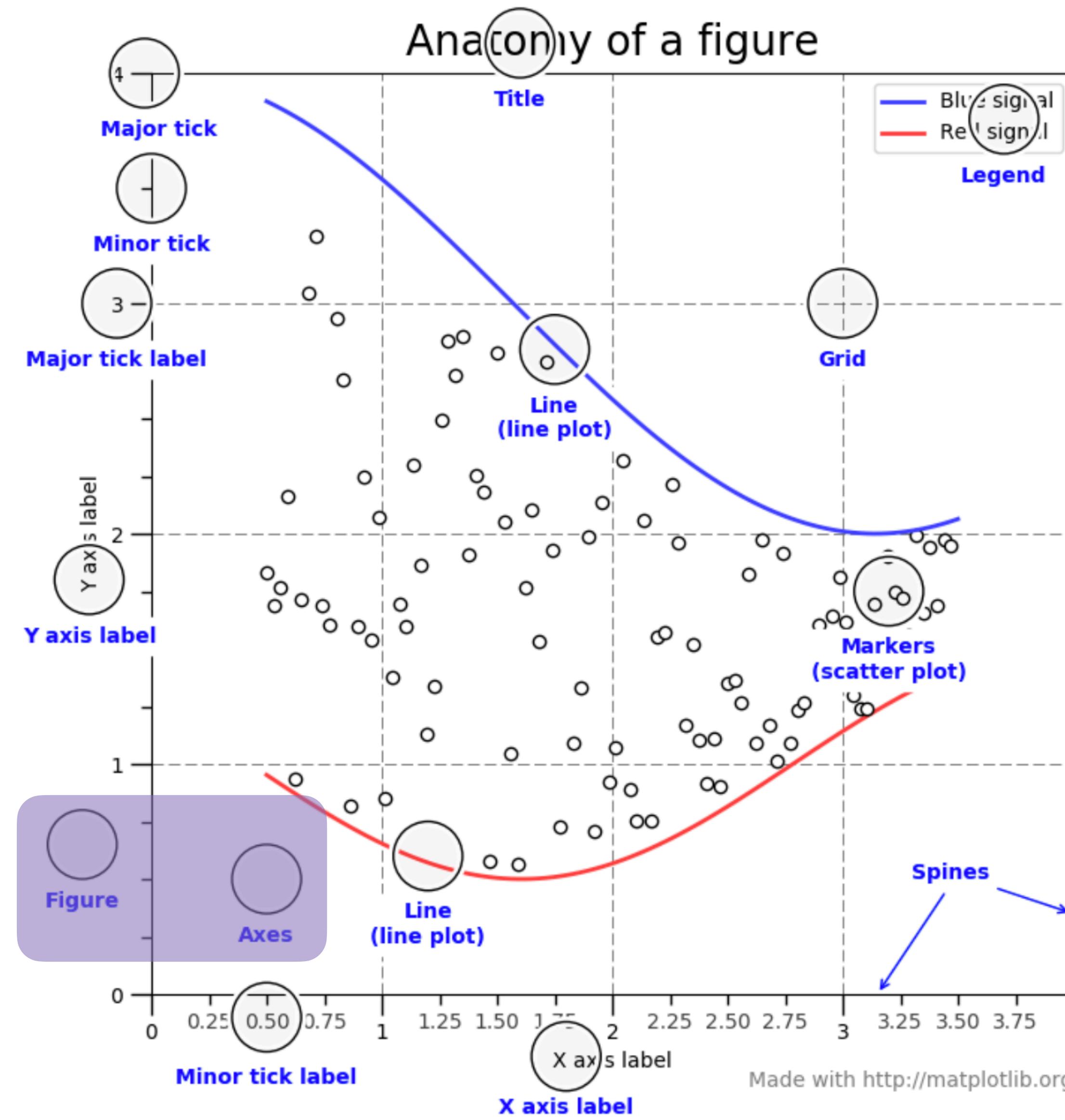
-Chapter.1 Matplotlib Anatomy -

1-01. Making Figures and Axes

1. Figures and Axes
2. plt.figure(Making Figures)
3. fig.add_subplot(Adding Subplots)
4. plt.subplots(Making Fig and Axes Simultaneously)
5. plt.subplot2grid(More Complex Arrangement)
6. Practice
7. fig.add_axes(Arbitrary Locations and Sizes of Axes)

Lecture_1-01 Making Figures and Axes

4



1. Figures and Axes

Adversarial Robustness vs. Model Compression, or Both?

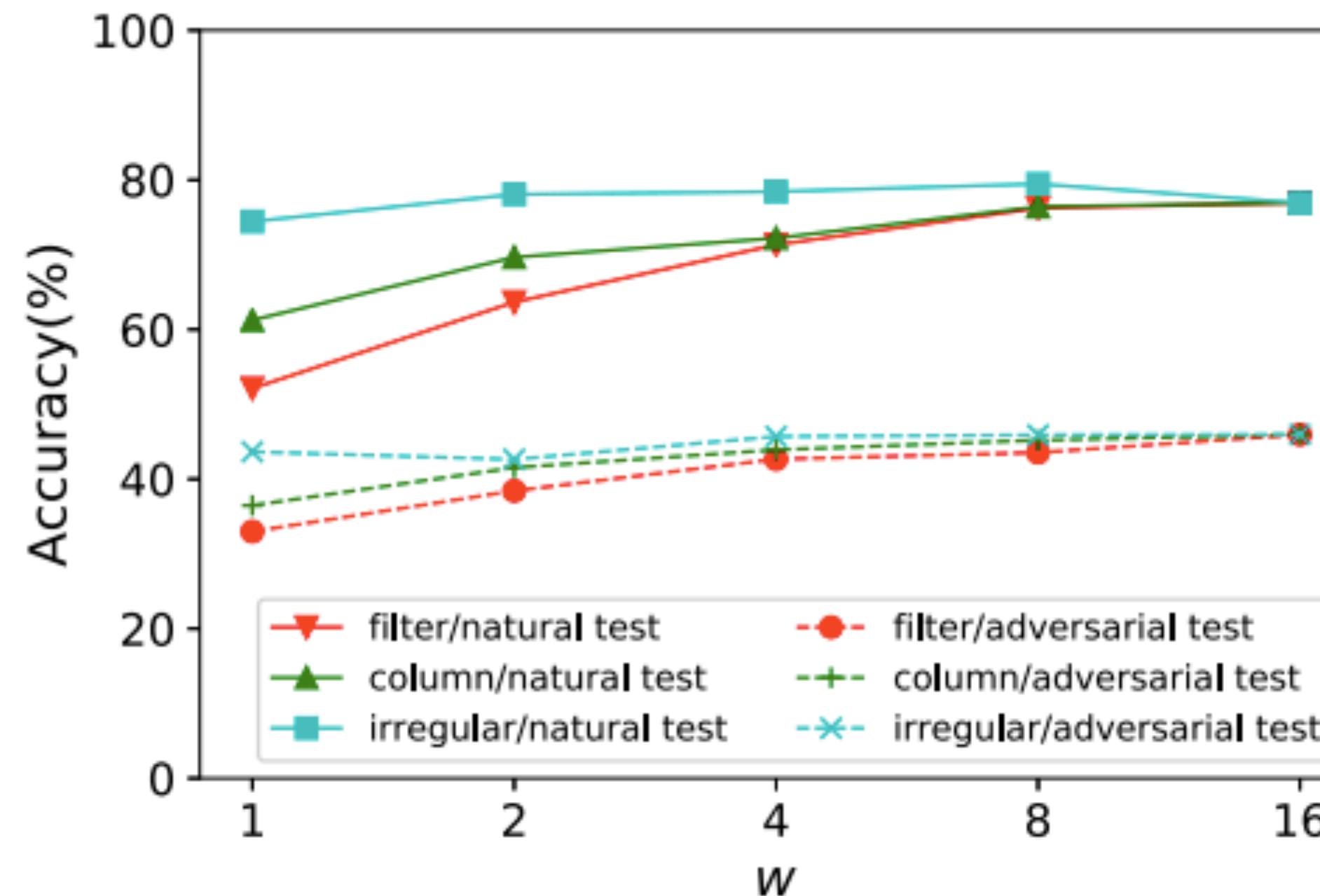
Shaokai Ye^{1*} Kaidi Xu^{2*} Sijia Liu³ Hao Cheng⁴ Jan-Henrik Lambrechts¹ Huan Zhang⁶

Aojun Zhou⁵ Kaisheng Ma¹⁺ Yanzhi Wang²⁺ Xue Lin²⁺

¹IIIS, Tsinghua University & IIISCT, China ²Northeastern University, USA

³MIT-IBM Watson AI Lab, IBM Research ⁴Xi'an Jiaotong University, China

⁵SenseTime Research, China ⁶University of California, Los Angeles, USA



1 figure

1 ax

ax1

ax2 => axes라고 부른다

Lecture. 1-01 Making Figures and Axes

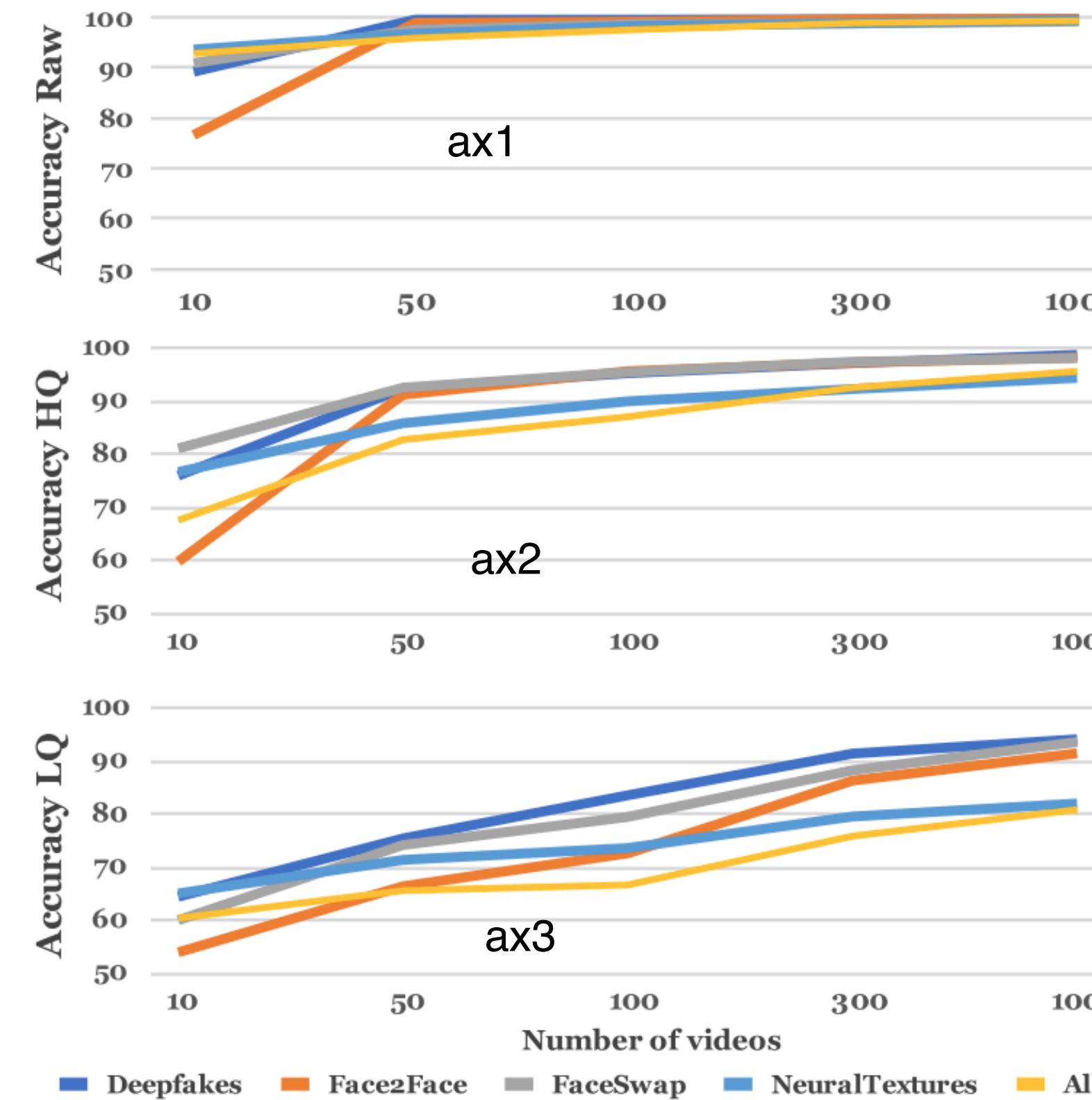
6

1. Figures and Axes

FaceForensics++: Learning to Detect Manipulated Facial Images

Andreas Rössler¹ Davide Cozzolino² Luisa Verdoliva² Christian Riess³
Justus Thies¹ Matthias Nießner¹

¹Technical University of Munich ²University Federico II of Naples ³University of Erlangen-Nuremberg

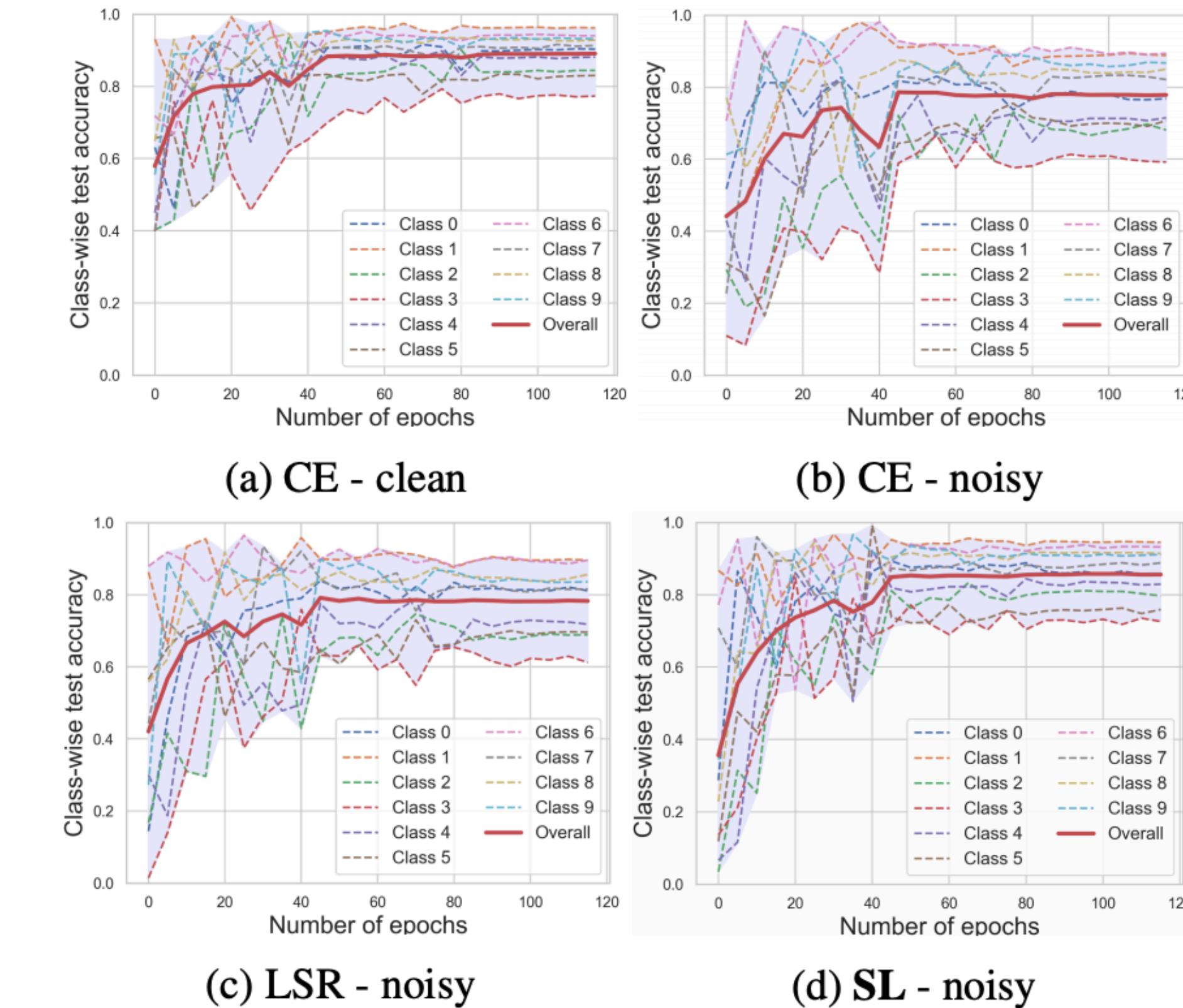


3 x 1 의 행렬

Symmetric Cross Entropy for Robust Learning with Noisy Labels

Yisen Wang^{1,*†} Xingjun Ma^{2,*†} Zaiyi Chen³ Yuan Luo¹ Jinfeng Yi⁴ James Bailey²

¹Shanghai Jiao Tong University ²The University of Melbourne ³Cainiao AI ⁴JD AI



2 x 2 의 행렬

Lecture_1-01 Making Figures and Axes

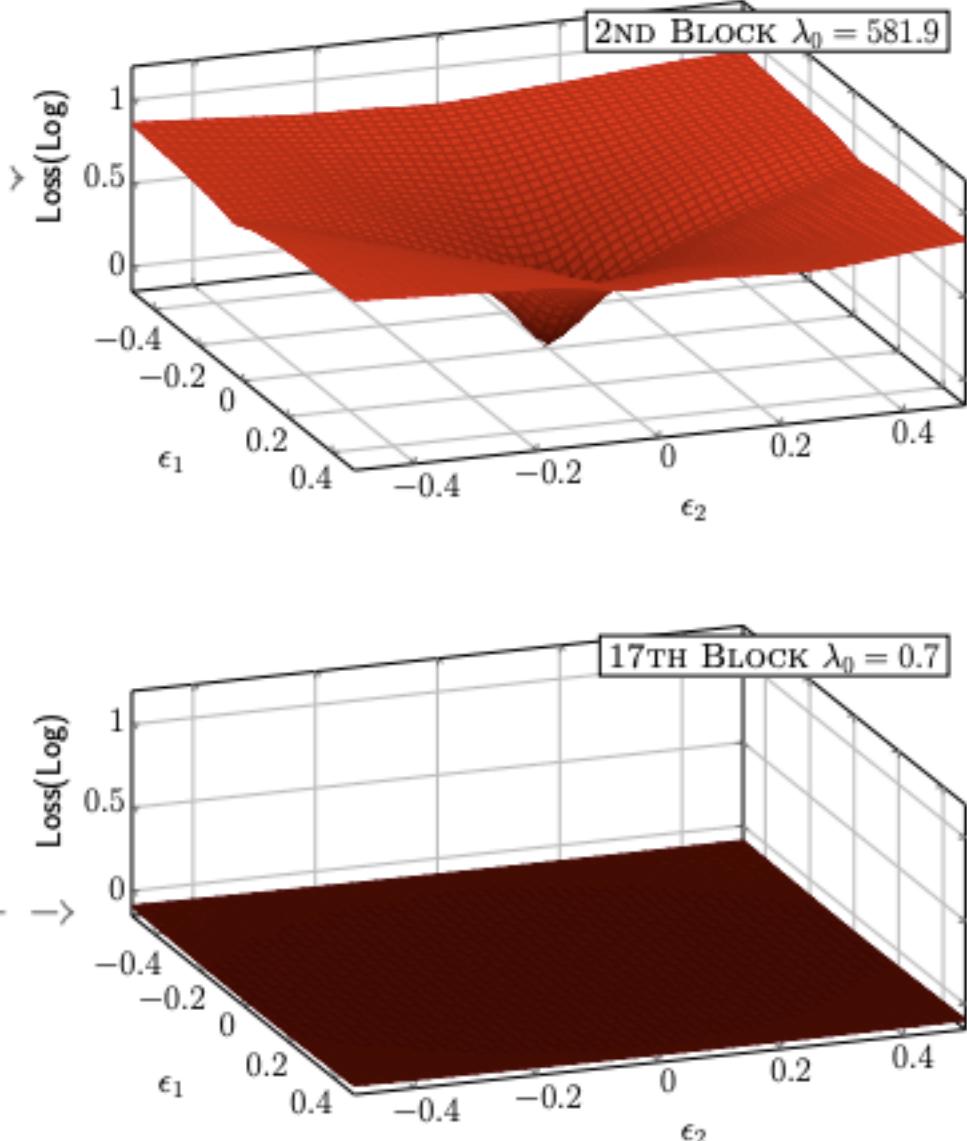
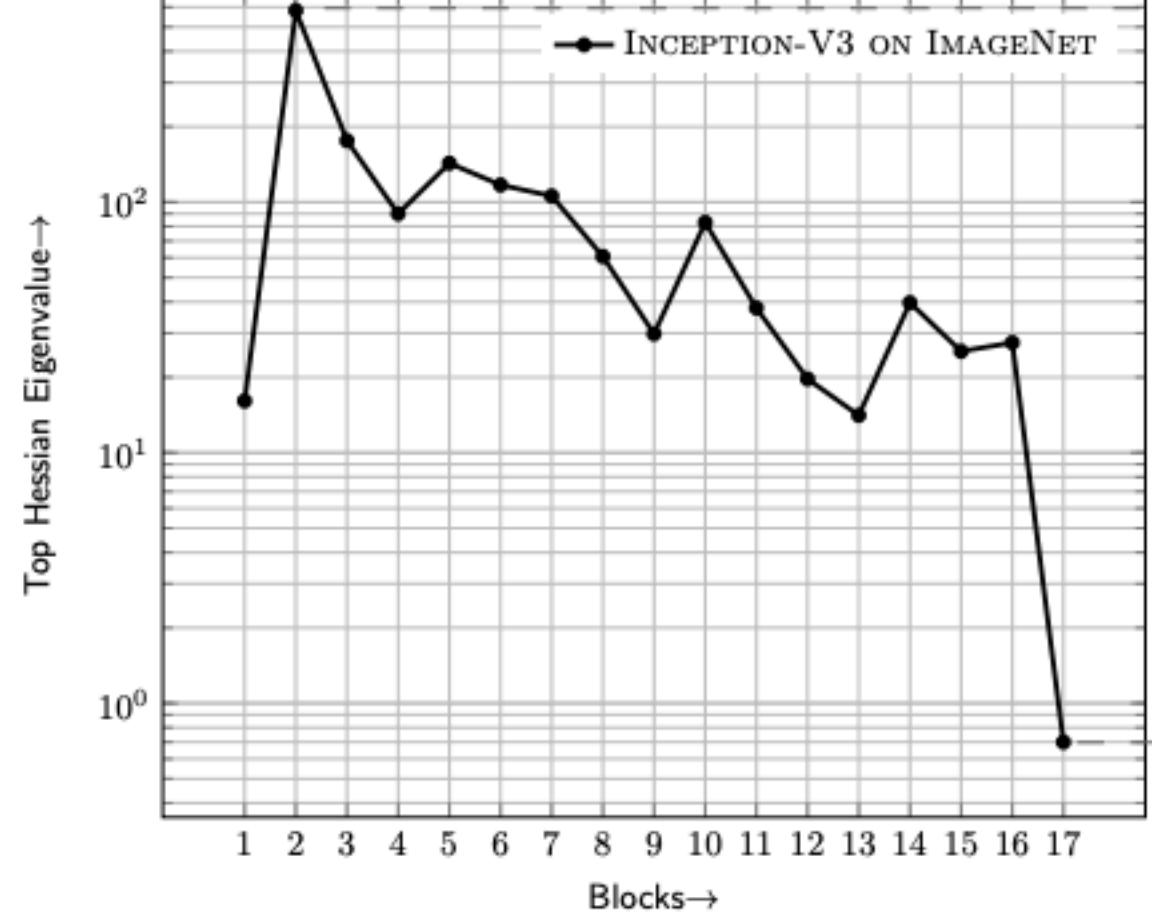
7

1. Figures and Axes

HAWQ: Hessian AWare Quantization of Neural Networks with Mixed-Precision

Zhen Dong*, Zhewei Yao*, Amir Gholami*, Michael W. Mahoney, Kurt Keutzer
University of California, Berkeley

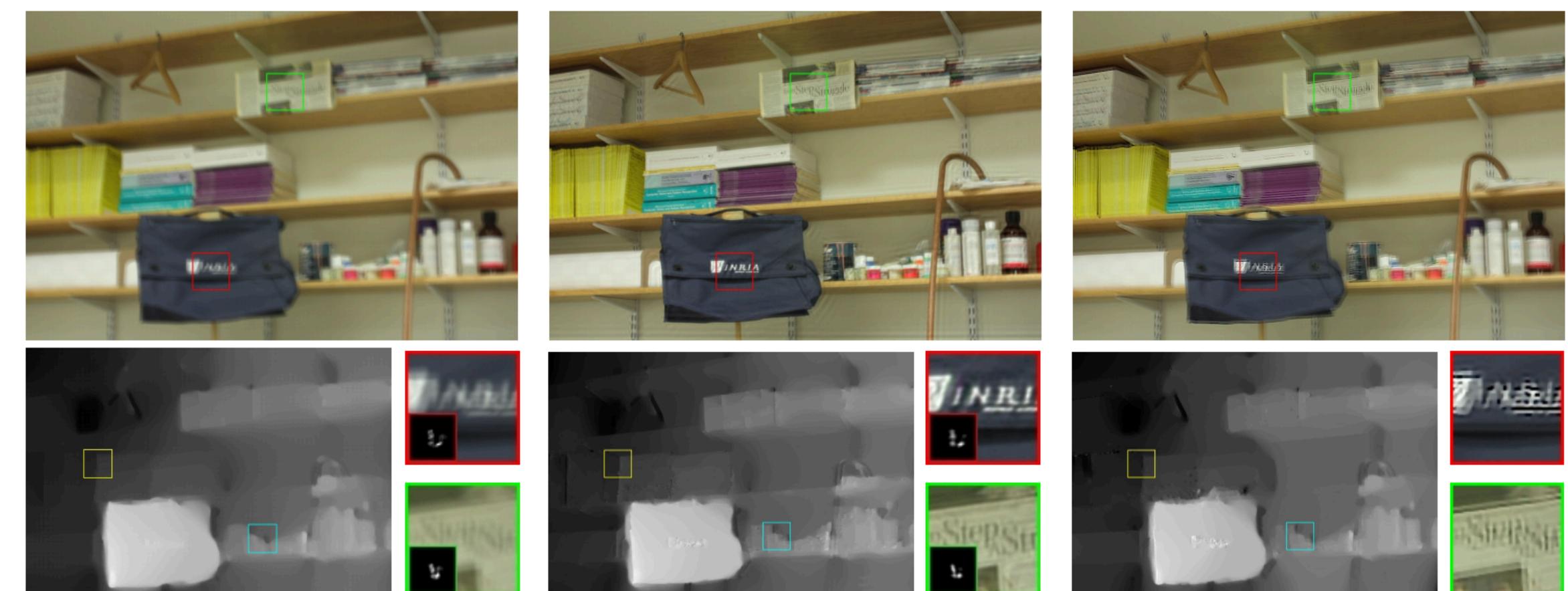
{zhendong, zheweyi, amirgh, mahoneymw, and keutzer}@berkeley.edu



Unconstrained Motion Deblurring for Dual-lens Cameras

M. R. Mahesh Mohan, Sharath Girish, and A. N. Rajagopalan
Indian Institute of Technology Madras

{ee14d023, ee15b058, raju}@ee.iitm.ac.in



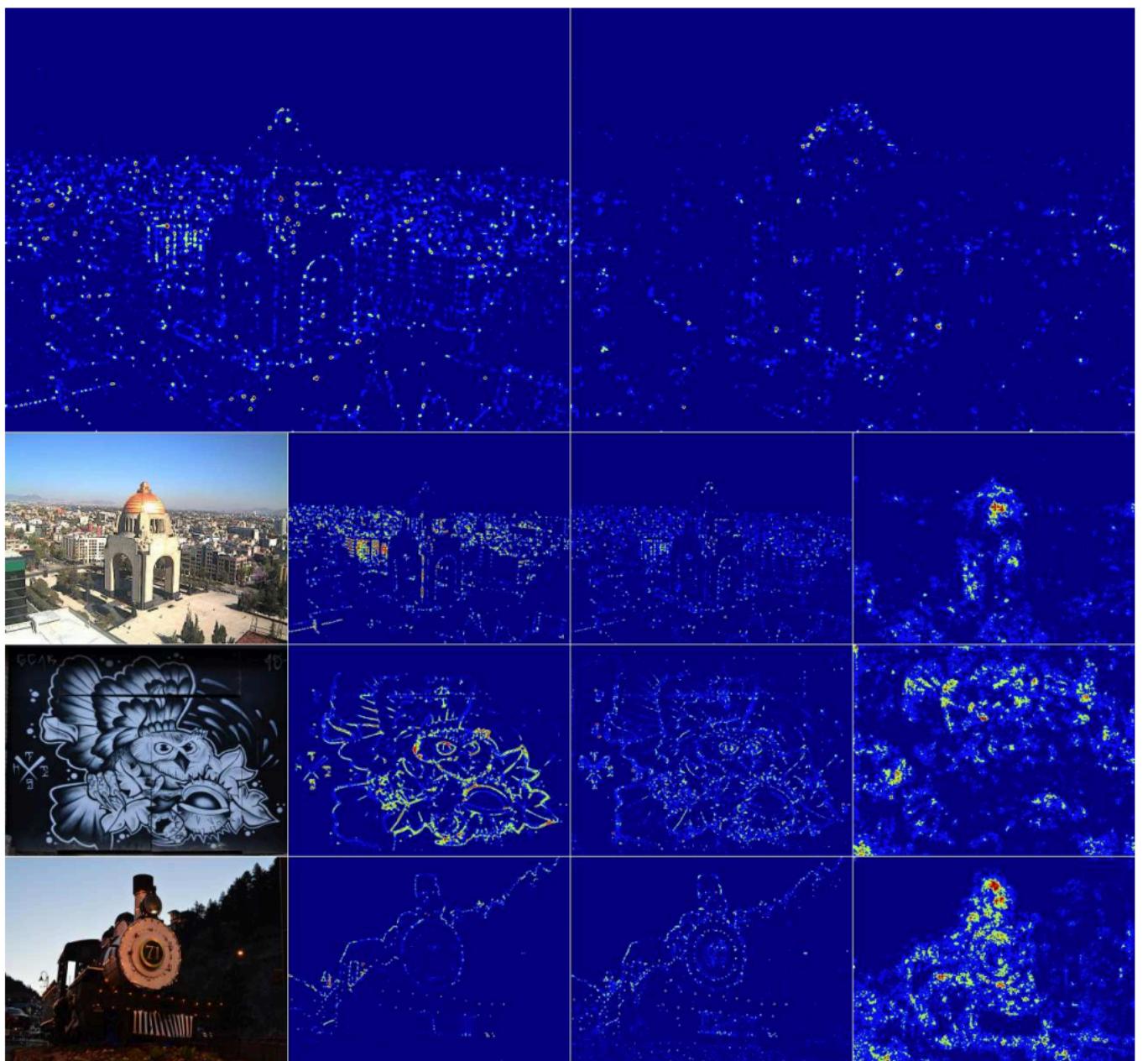
Lecture_1-01 Making Figures and Axes

8

1. Figures and Axes

ELF: Embedded Localisation of Features in Pre-Trained CNN

Assia Benbihi
UMI2958 GeorgiaTech-CNRS
Centrale Supélec
Université Paris-Saclay
Metz, France
abenbihi@georgiatech-metz.fr

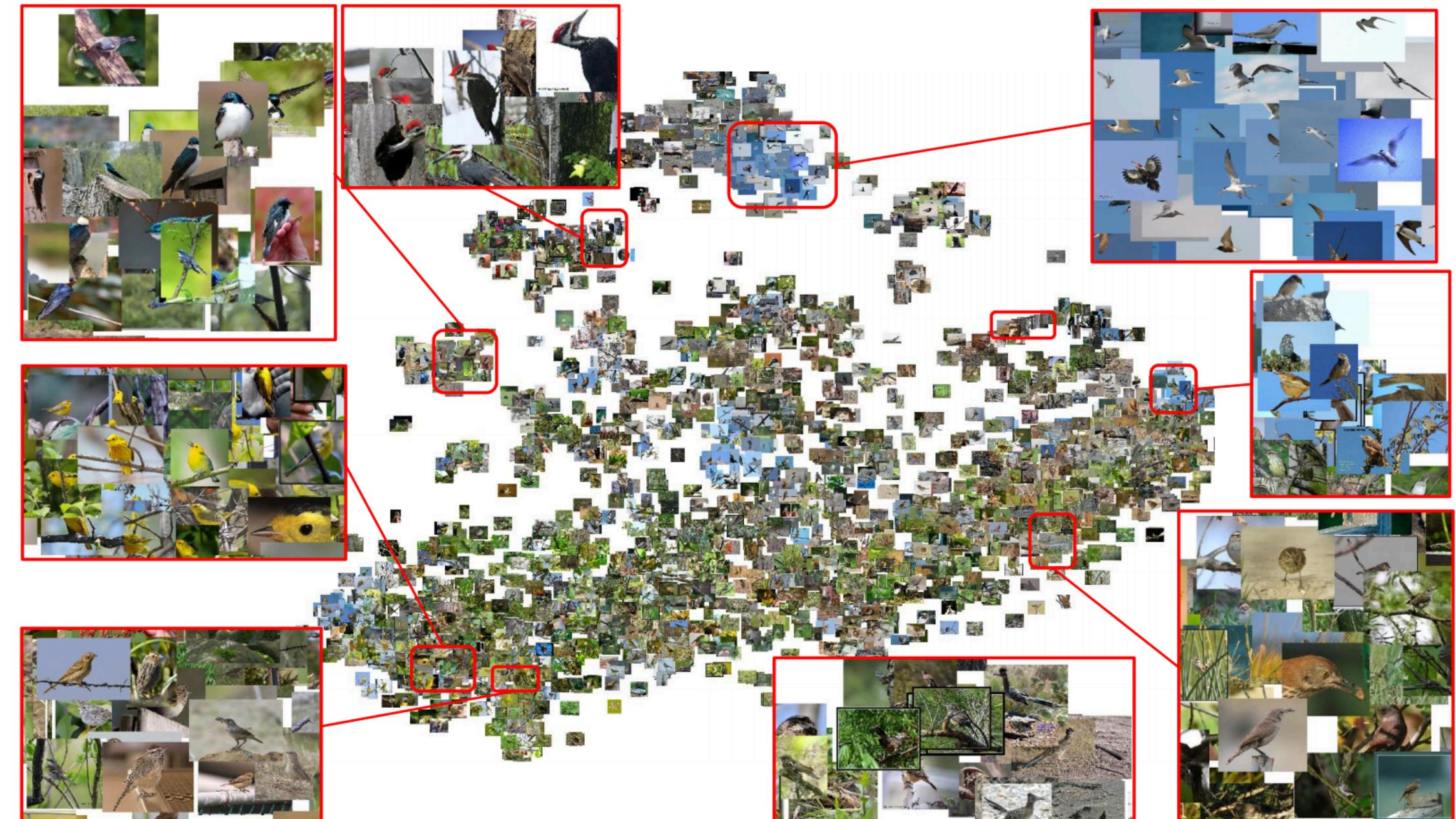


Shin's Lab

Sampling Wisely: Deep Image Embedding by Top- k Precision Optimization

Jing Lu^{1*} Chaofan Xu^{2,3*} Wei Zhang² Lingyu Duan⁴ Tao Mei²

¹Business Growth BU, JD ² JD AI Research ³Harbin Institute of Technology ⁴Peking University
lvjing12@jd.com, xuchaofan1994@126.com, wzhang.cu@gmail.com, lingyu@pku.edu.cn, tmei@live.com

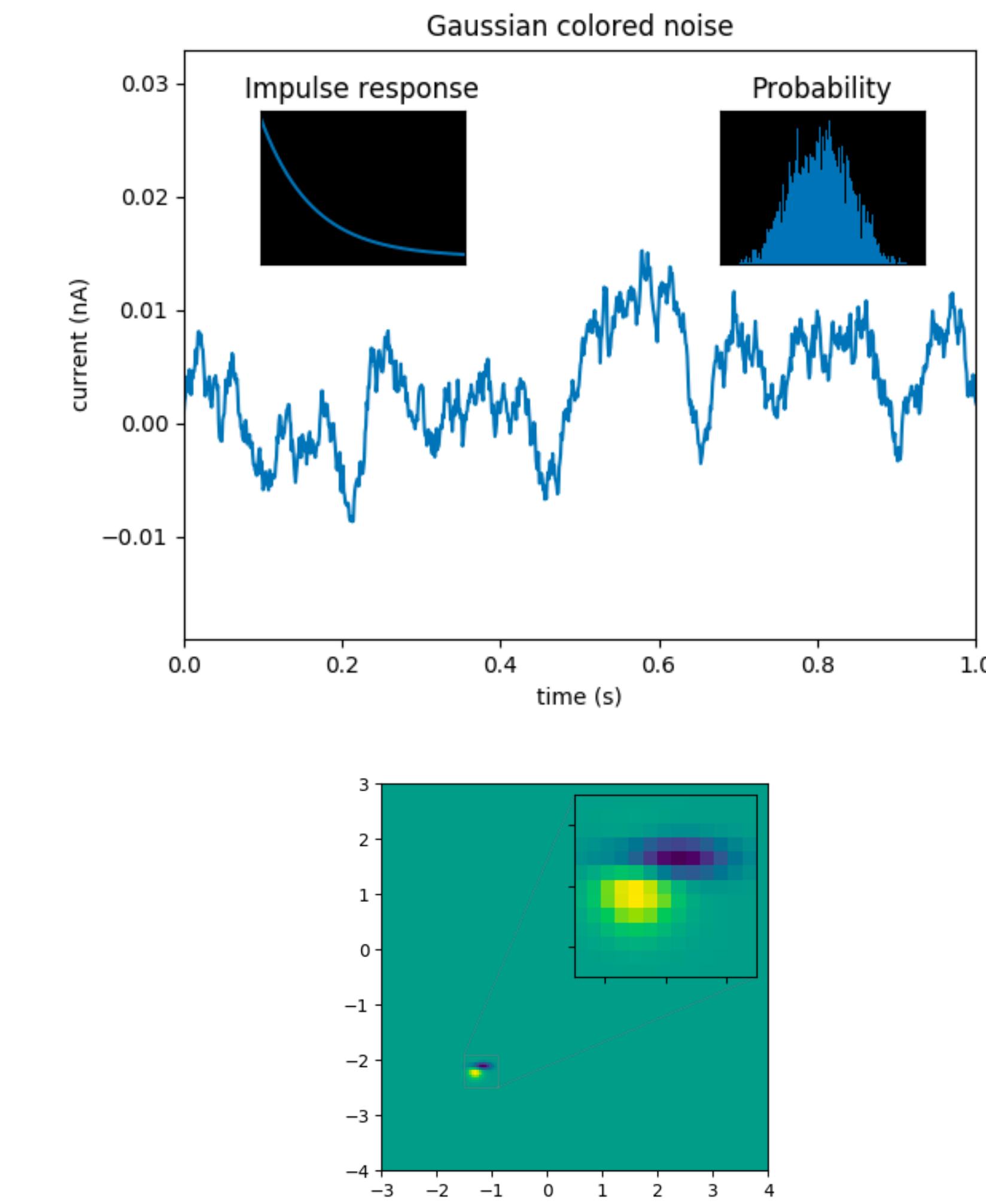
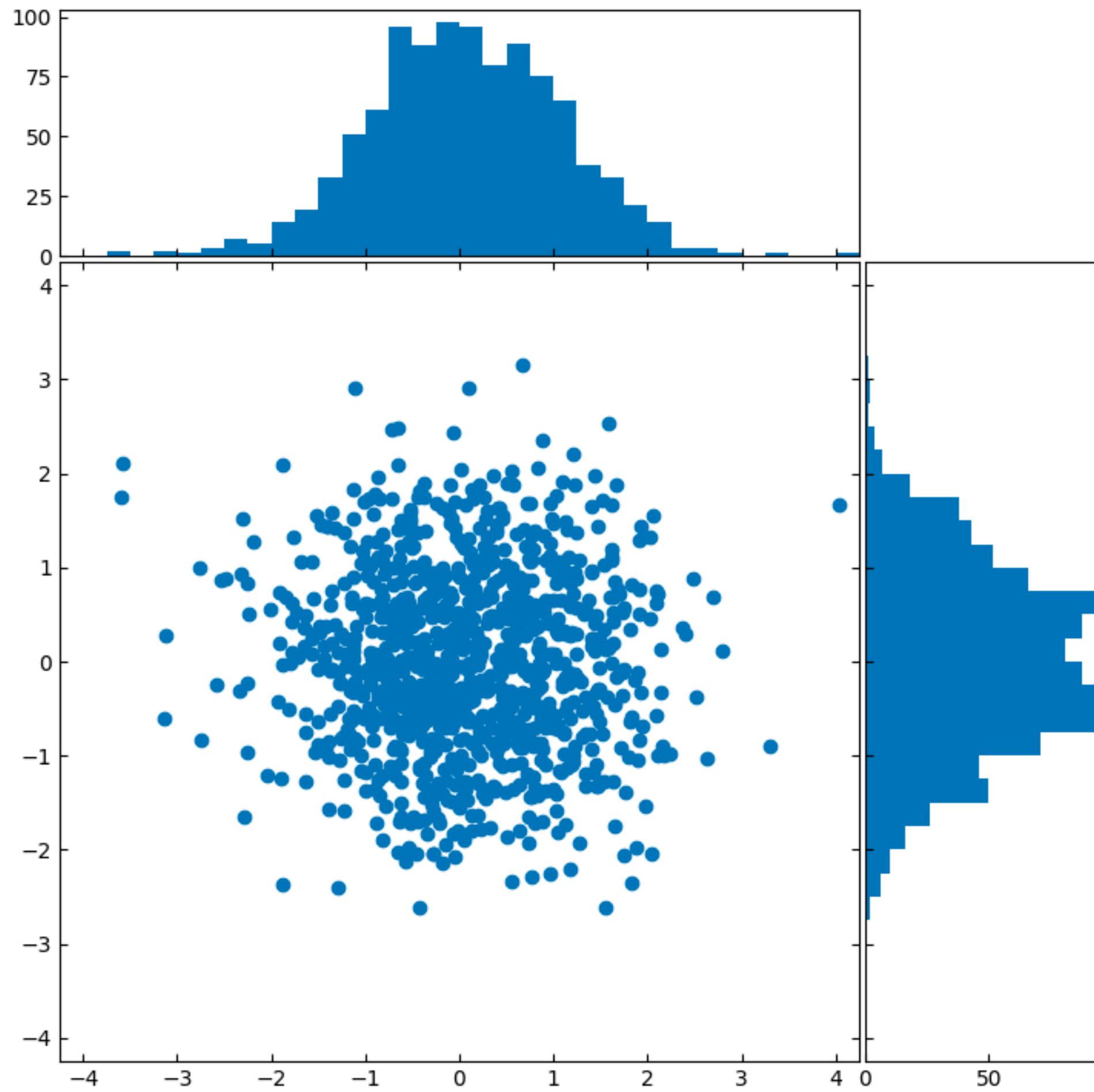


Python for Data Visualization

Lecture. 1-01 Making Figures and Axes

9

1. Figures and Axes



https://matplotlib.org/3.1.0/gallery/lines_bars_and_markers/scatter_hist.html#sphx-glr-gallery-lines-bars-and-markers-scatter-hist-py

Shin's Lab

https://matplotlib.org/3.1.0/gallery/subplots_axes_and_figures/axes_demo.html#sphx-glr-gallery-subplots-axes-and-figures-axes-demo-py

https://matplotlib.org/3.1.0/gallery/subplots_axes_and_figures/zoom_inset_axes.html#sphx-glr-gallery-subplots-axes-and-figures-zoom-inset-axes-py

Python for Data Visualization

2. plt.figure(Making Figures)

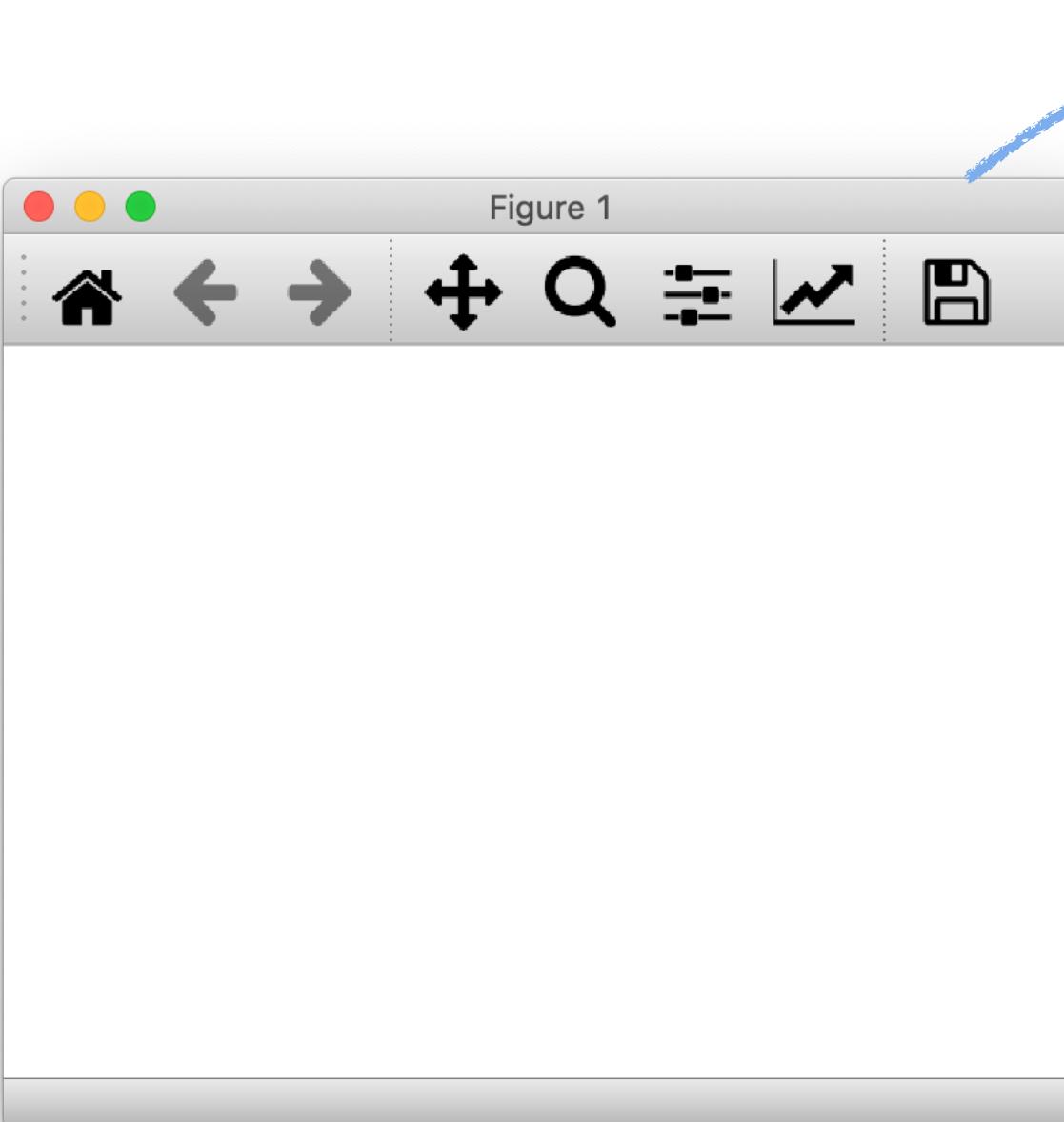
matplotlib.pyplot.figure

```
matplotlib.pyplot.figure(num=None, figsize=None, dpi=None, facecolor=None, edgecolor=None, frameon=True,  
FigureClass=<class 'matplotlib.figure.Figure'>, clear=False, **kwargs)
```

[source]

Create a new figure.

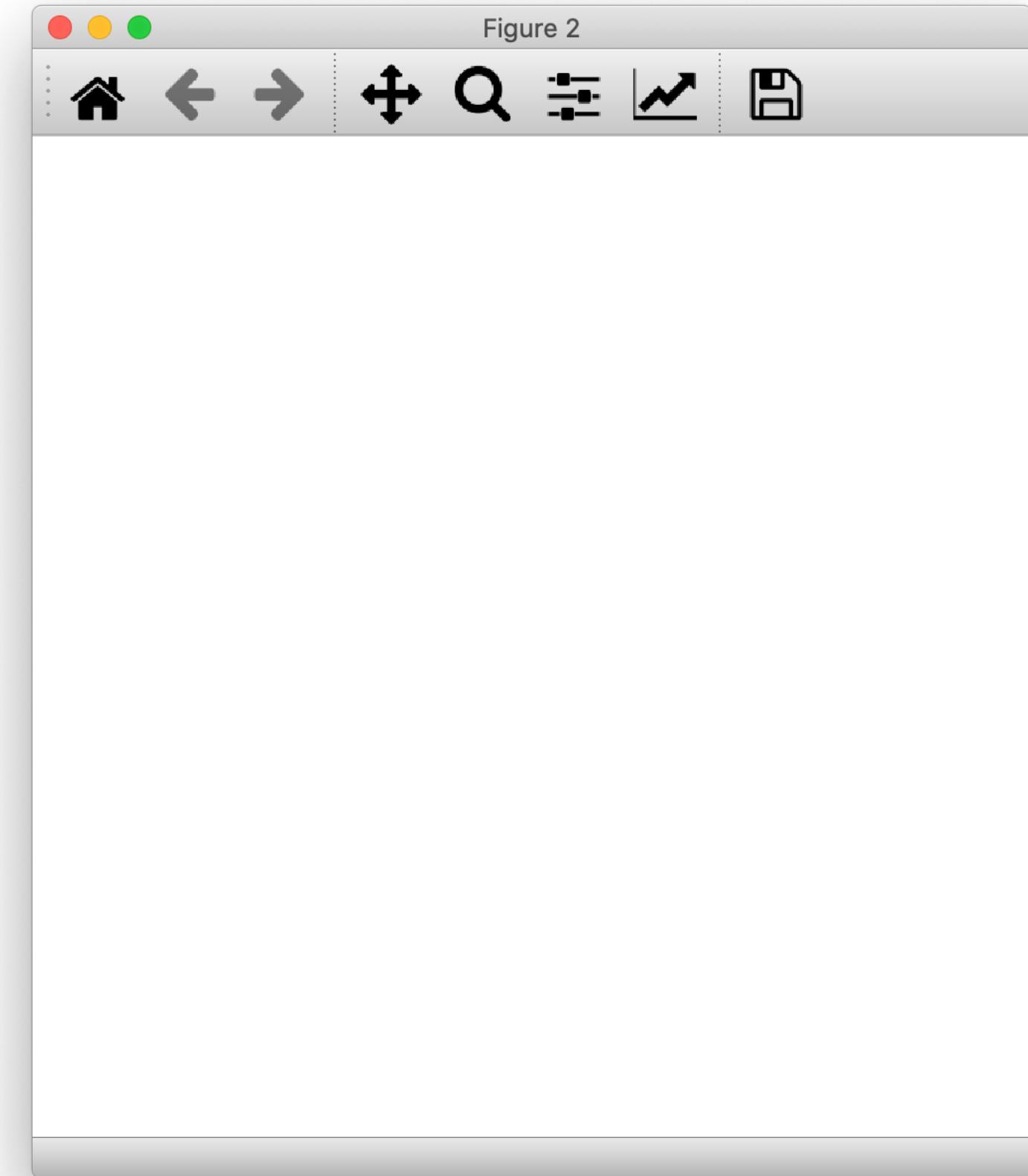
```
import matplotlib.pyplot as plt  
import numpy as np  
  
fig = plt.figure()
```



'fig' object
fig.suptitle()
fig.subplots_adjust()
⋮

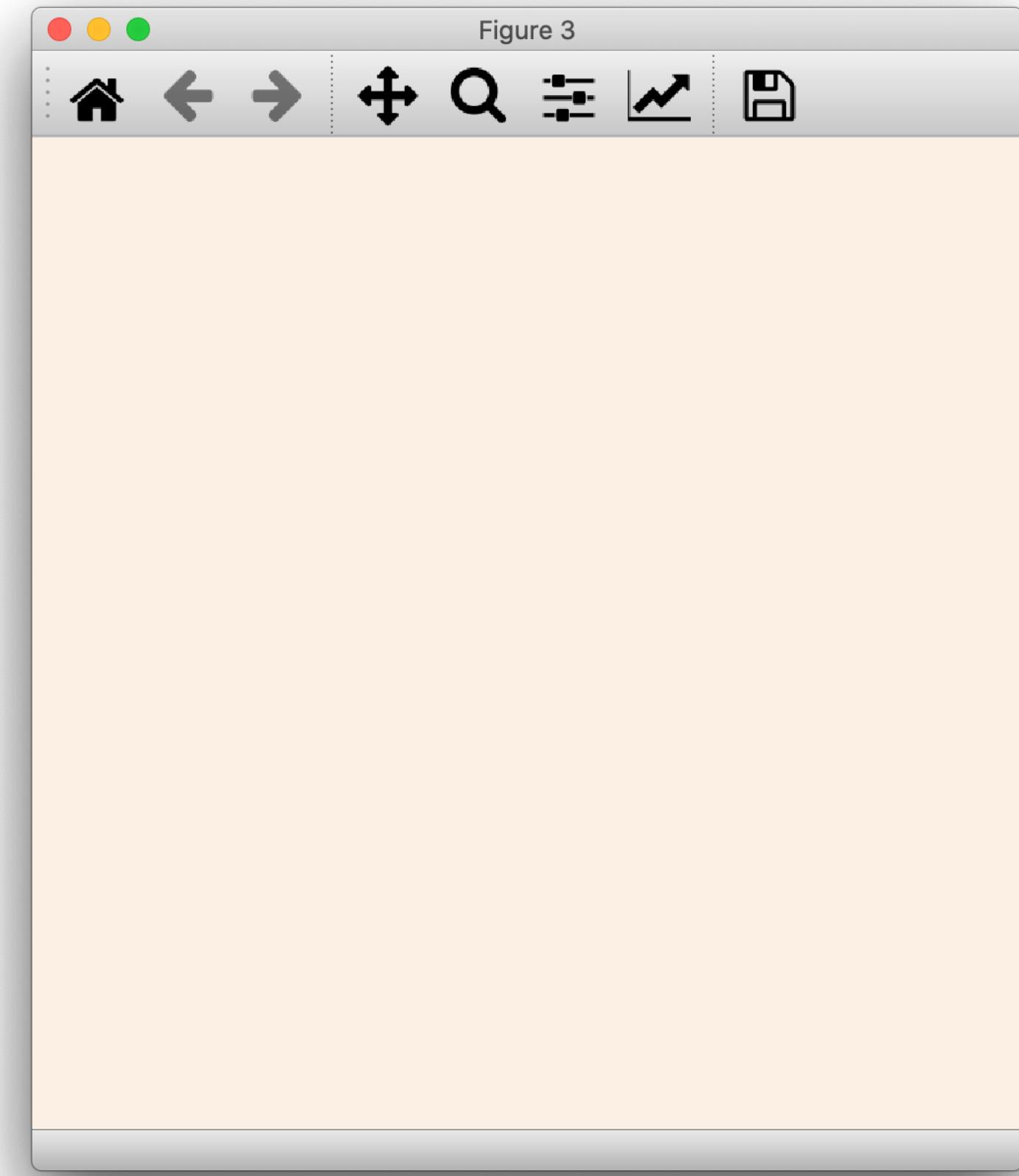
2. plt.figure(figsize Argument)

```
import matplotlib.pyplot as plt  
import numpy as np  
  
fig = plt.figure()  
fig = plt.figure(figsize=(7, 7))
```



2. plt.figure(Other Argument)

```
import matplotlib.pyplot as plt  
import numpy as np  
  
fig = plt.figure()  
fig = plt.figure(figsize=(7, 7))  
fig = plt.figure(figsize=(7, 7),  
                 facecolor='linen')
```



3. fig.add_subplot(Adding Subplots)

```
add_subplot(self, *args, **kwargs)
```

[\[source\]](#)

Add an `Axes` to the figure as part of a subplot arrangement.

Call signatures:

```
add_subplot(nrows, ncols, index, **kwargs)
add_subplot(pos, **kwargs)
add_subplot(ax)
add_subplot()
```

Usage

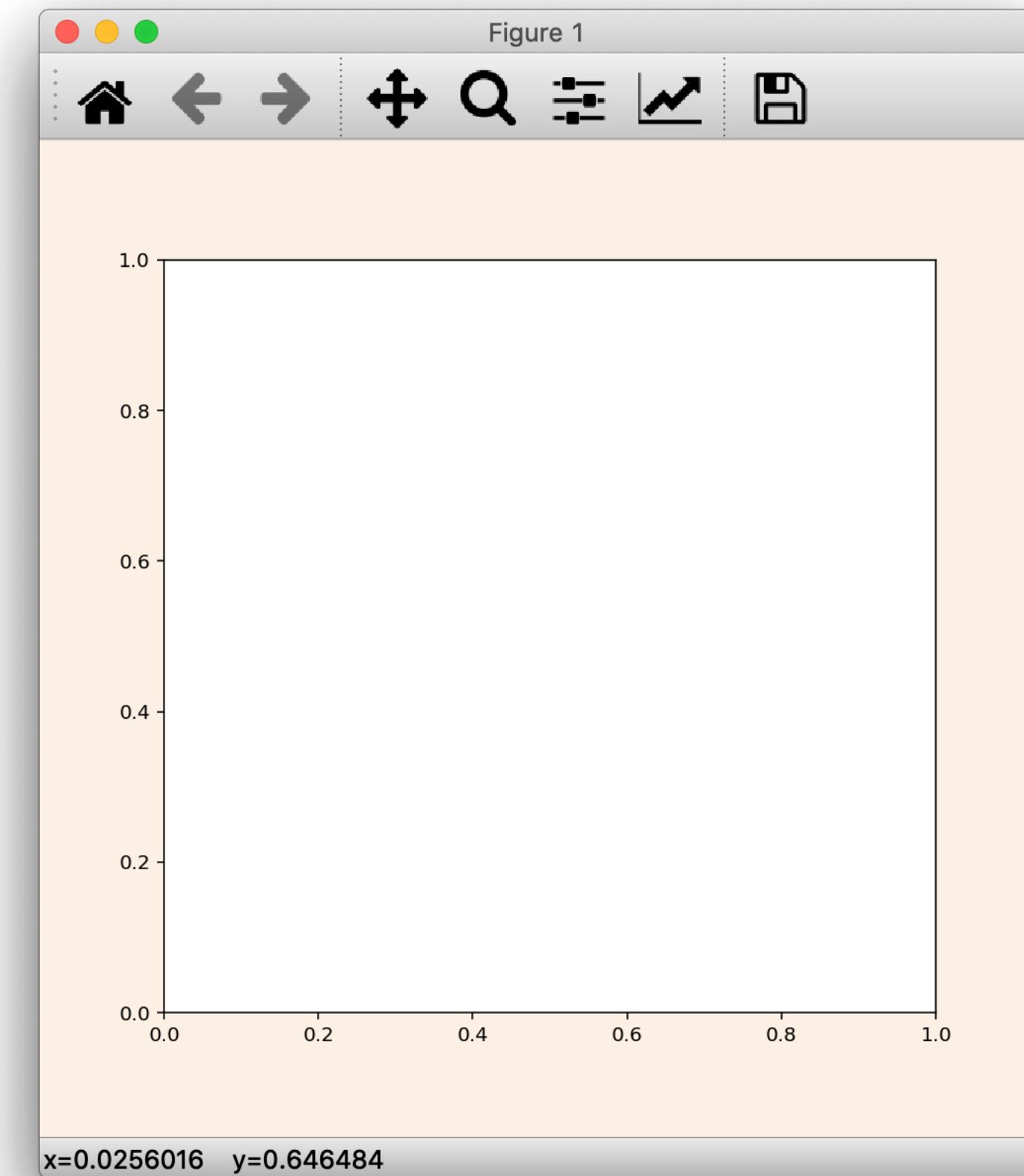
```
fig = plt.figure()
ax = fig.add_subplot()
```

3. fig.add_subplot(Adding Subplots)

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

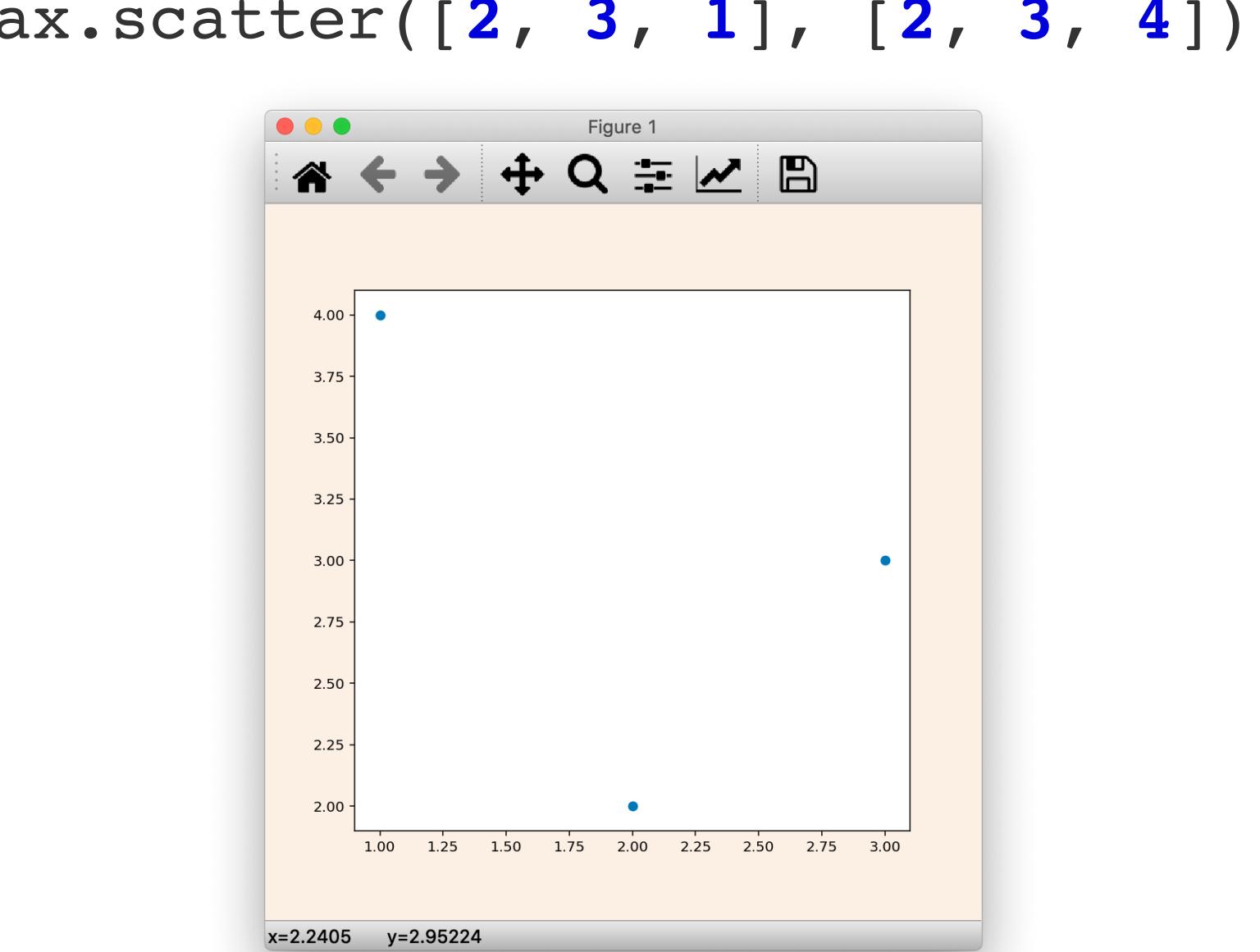
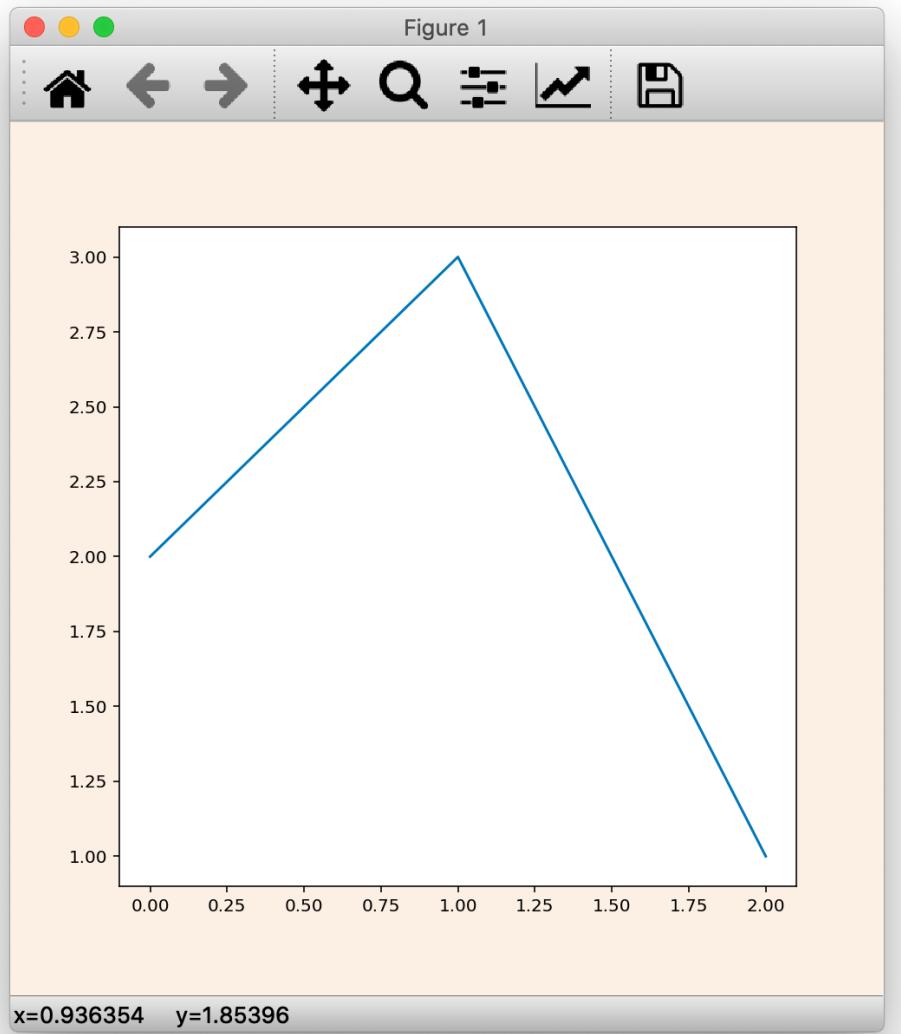
fig = plt.figure(figsize=(7, 7),
                 facecolor='linen')

ax = fig.add_subplot()
```

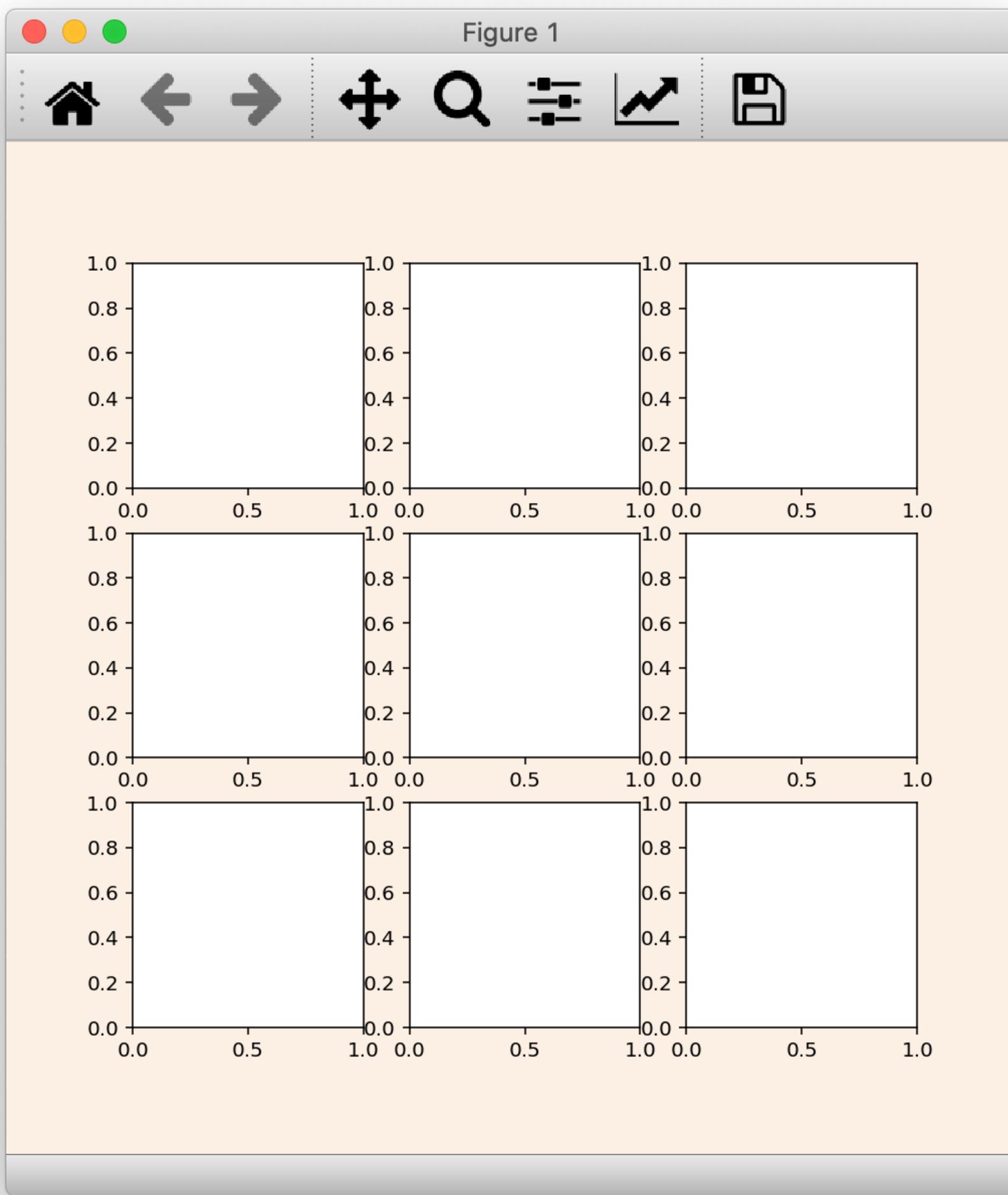


3. fig.add_subplot(Adding Subplots)

```
import matplotlib.pyplot as plt  
import numpy as np  
  
fig = plt.figure(figsize=(7, 7),  
                 facecolor='linen')  
  
ax = fig.add_subplot()  
  
    ax.plot([2, 3, 1])  
    ax.scatter([2, 3, 1], [2, 3, 4])
```



3. `fig.add_subplot(Axes Grid)`



3. fig.add_subplot(Arguments)

```
add_subplot(self, *args, **kwargs)
```

[\[source\]](#)

Add an **Axes** to the figure as part of a subplot arrangement.

Call signatures:

```
add_subplot(nrows, ncols, index, **kwargs)
add_subplot(pos, **kwargs)
add_subplot(ax)
add_subplot()
```

```
ax = fig.add_subplot(111)
```

Rows # Columns "i"th axes

```
ax = fig.add_subplot(1, 1, 1)
```

3. fig.add_subplot(Single Ax)

Adversarial Robustness vs. Model Compression, or Both?

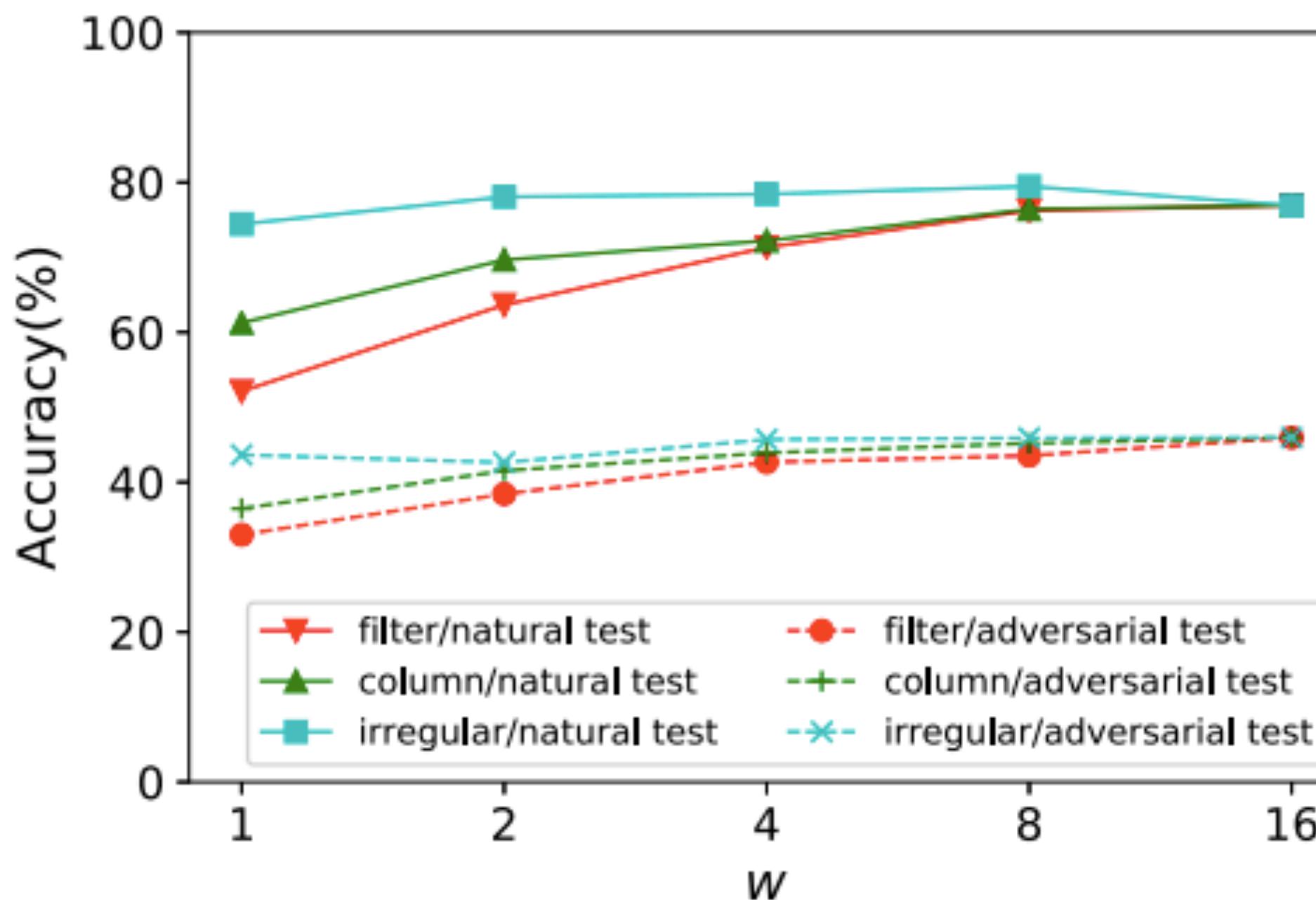
Shaokai Ye^{1*} Kaidi Xu^{2*} Sijia Liu³ Hao Cheng⁴ Jan-Henrik Lambrechts¹ Huan Zhang⁶

Aojun Zhou⁵ Kaisheng Ma¹⁺ Yanzhi Wang²⁺ Xue Lin²⁺

¹IIIS, Tsinghua University & IIISCT, China ²Northeastern University, USA

³MIT-IBM Watson AI Lab, IBM Research ⁴Xi'an Jiaotong University, China

⁵SenseTime Research, China ⁶University of California, Los Angeles, USA



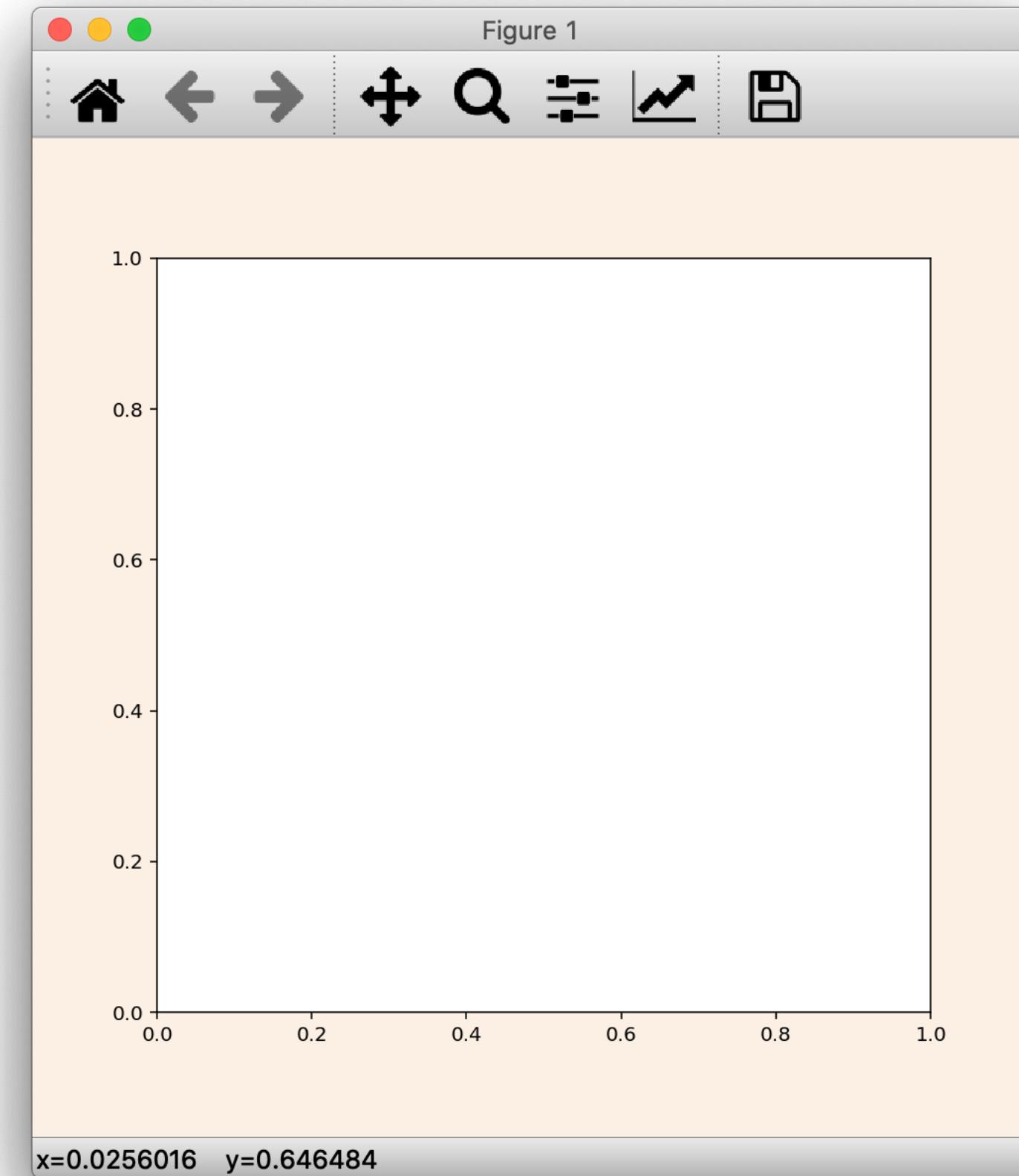
(a) VGG-16

3. fig.add_subplot(Single Ax)

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

fig = plt.figure(figsize=(7, 7),
                 facecolor='linen')

ax = fig.add_subplot(1, 1, 1)
ax = fig.add_subplot(111)
```

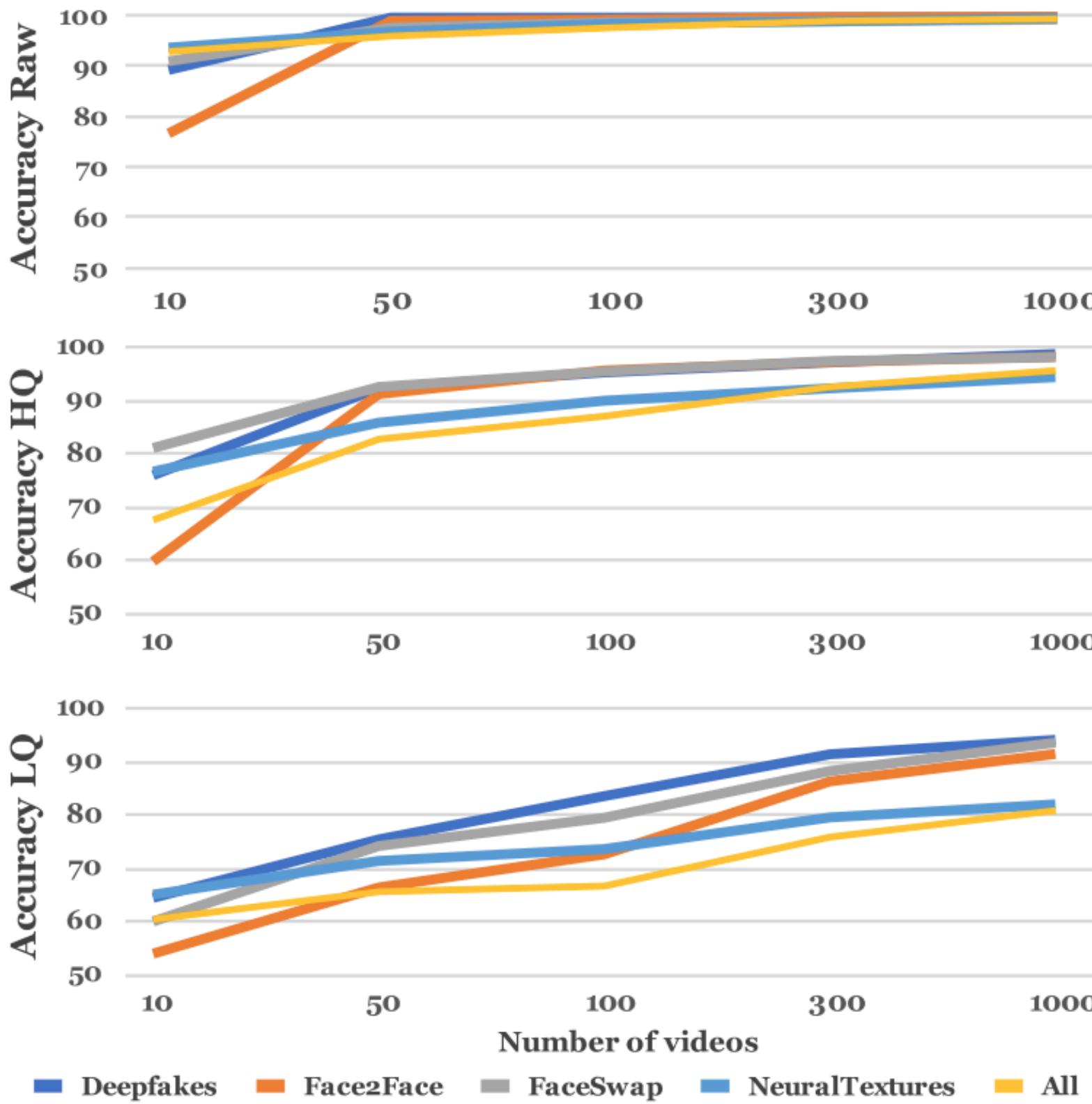


3. fig.add_subplot(1D Axes)

FaceForensics++: Learning to Detect Manipulated Facial Images

Andreas Rössler¹ Davide Cozzolino² Luisa Verdoliva² Christian Riess³
Justus Thies¹ Matthias Nießner¹

¹Technical University of Munich ²University Federico II of Naples ³University of Erlangen-Nuremberg

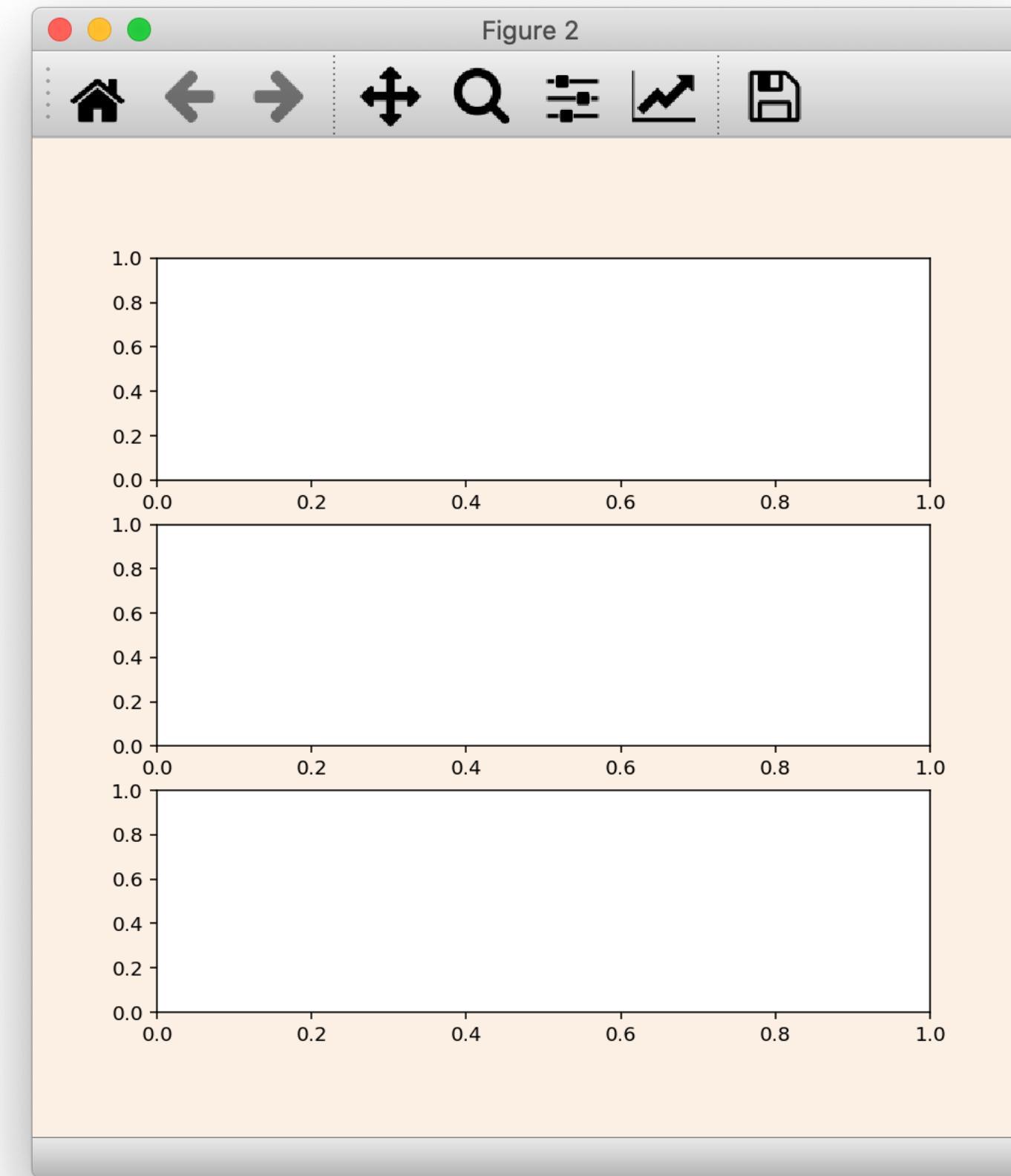


3. fig.add_subplot(1D Axes)

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

fig = plt.figure(figsize=(7, 7),
                 facecolor='linen')

ax1 = fig.add_subplot(311)
ax2 = fig.add_subplot(312)
ax3 = fig.add_subplot(313)
```



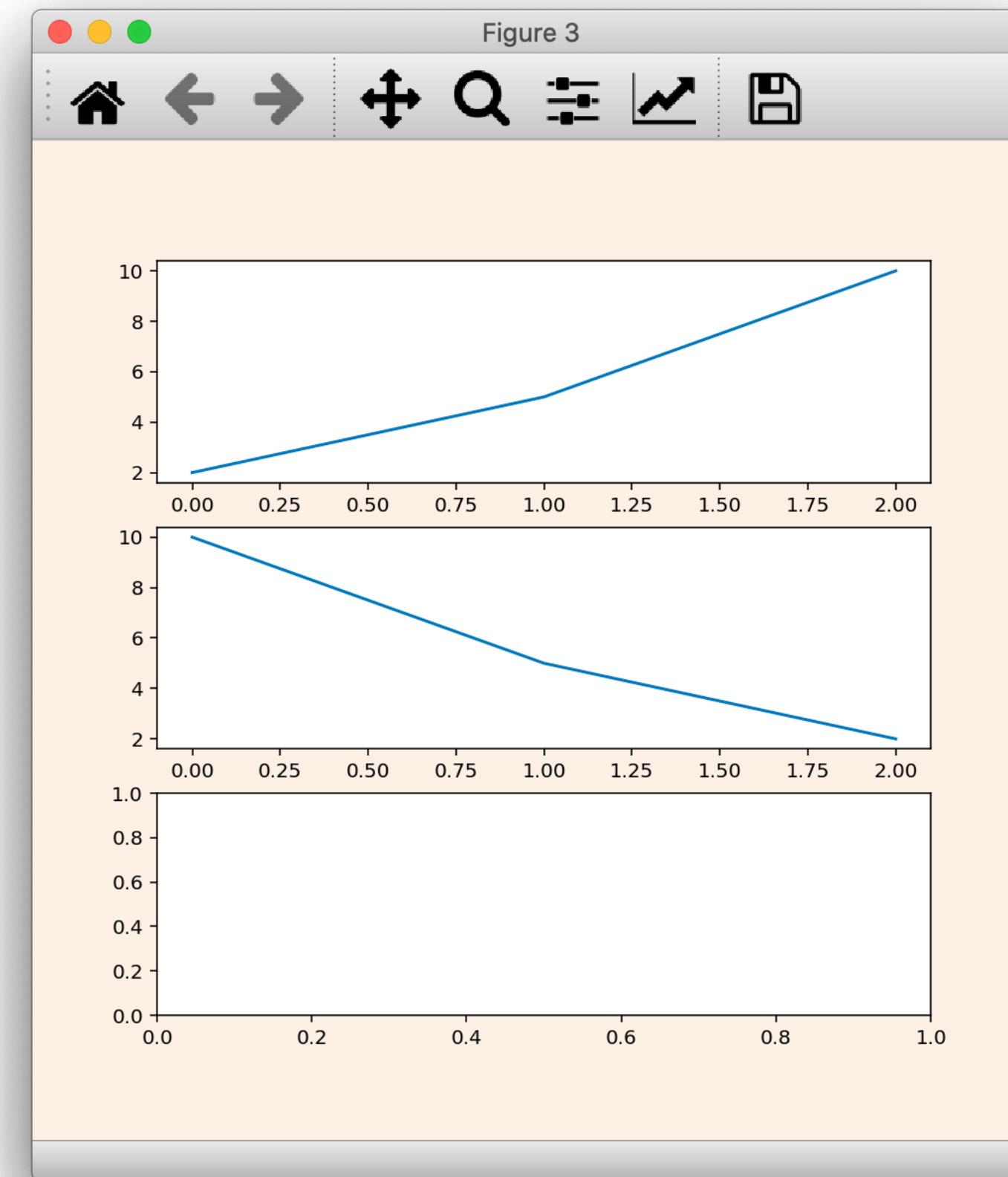
3. fig.add_subplot(1D Axes)

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

fig = plt.figure(figsize=(7, 7),
                 facecolor='linen')

ax1 = fig.add_subplot(311)
ax2 = fig.add_subplot(312)
ax3 = fig.add_subplot(313)

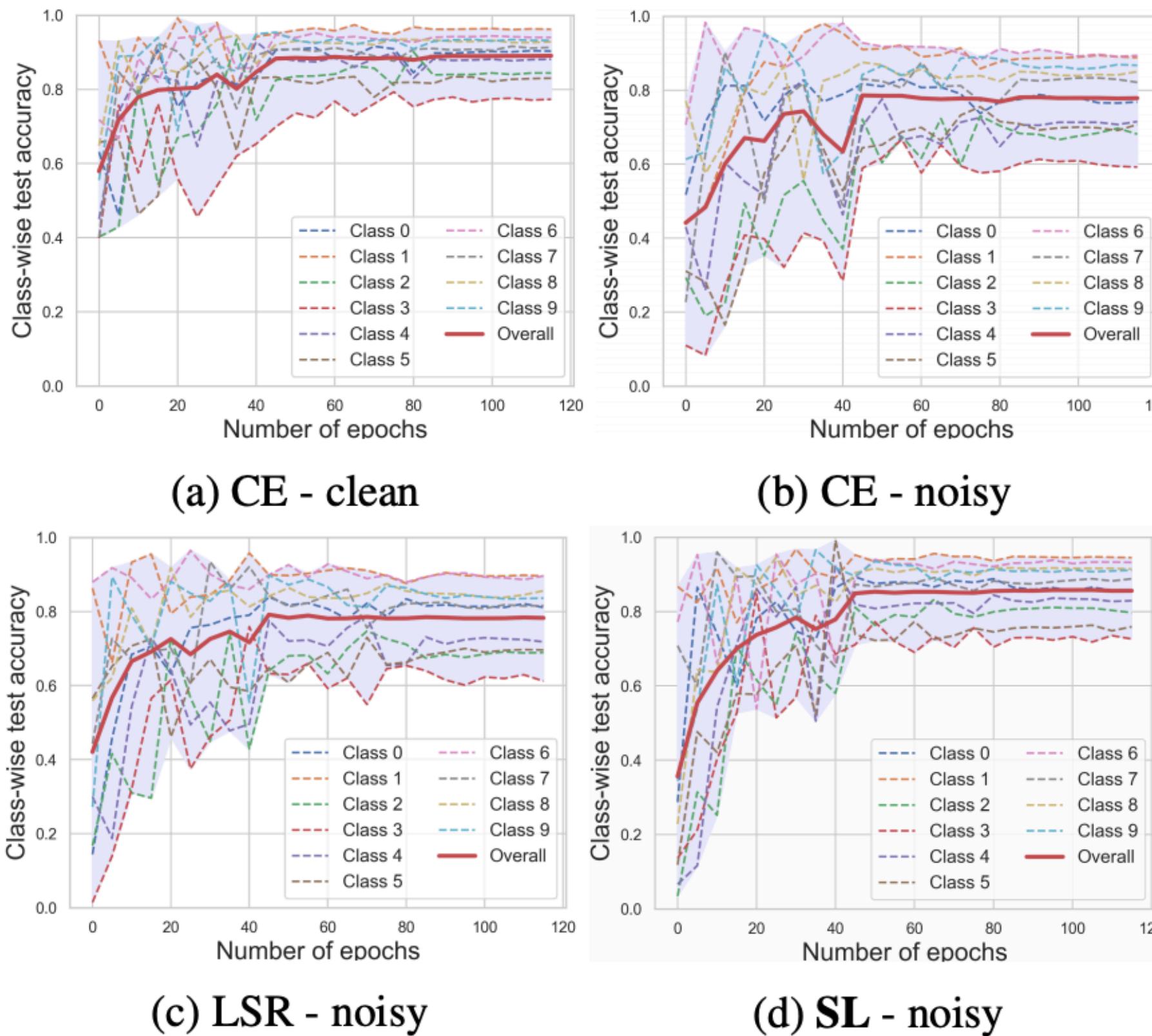
ax1.plot([2, 5, 10])
ax2.plot([10, 5, 2])
```



3. fig.add_subplot(2D Axes)

Symmetric Cross Entropy for Robust Learning with Noisy Labels

Yisen Wang^{1*}† Xingjun Ma^{2*}† Zaiyi Chen³ Yuan Luo¹ Jinfeng Yi⁴ James Bailey²
¹Shanghai Jiao Tong University ²The University of Melbourne ³Cainiao AI ⁴JD AI



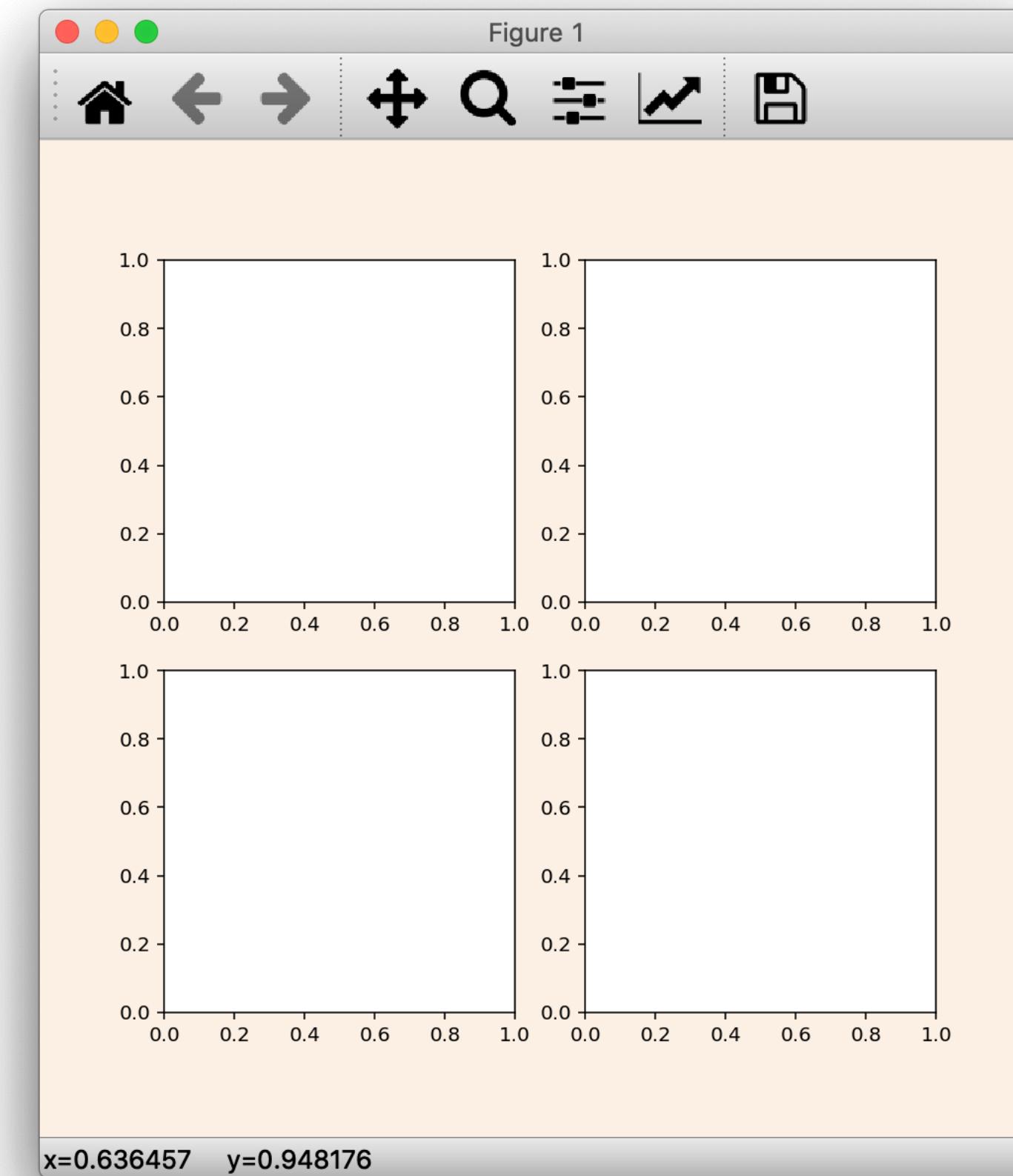
3. fig.add_subplot(2D Axes)

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

fig = plt.figure(figsize=(7, 7),
                 facecolor='linen')

ax1 = fig.add_subplot(311)
ax2 = fig.add_subplot(312)
ax3 = fig.add_subplot(313)

ax1 = fig.add_subplot(221)
ax2 = fig.add_subplot(222)
ax3 = fig.add_subplot(223)
ax4 = fig.add_subplot(224)
```

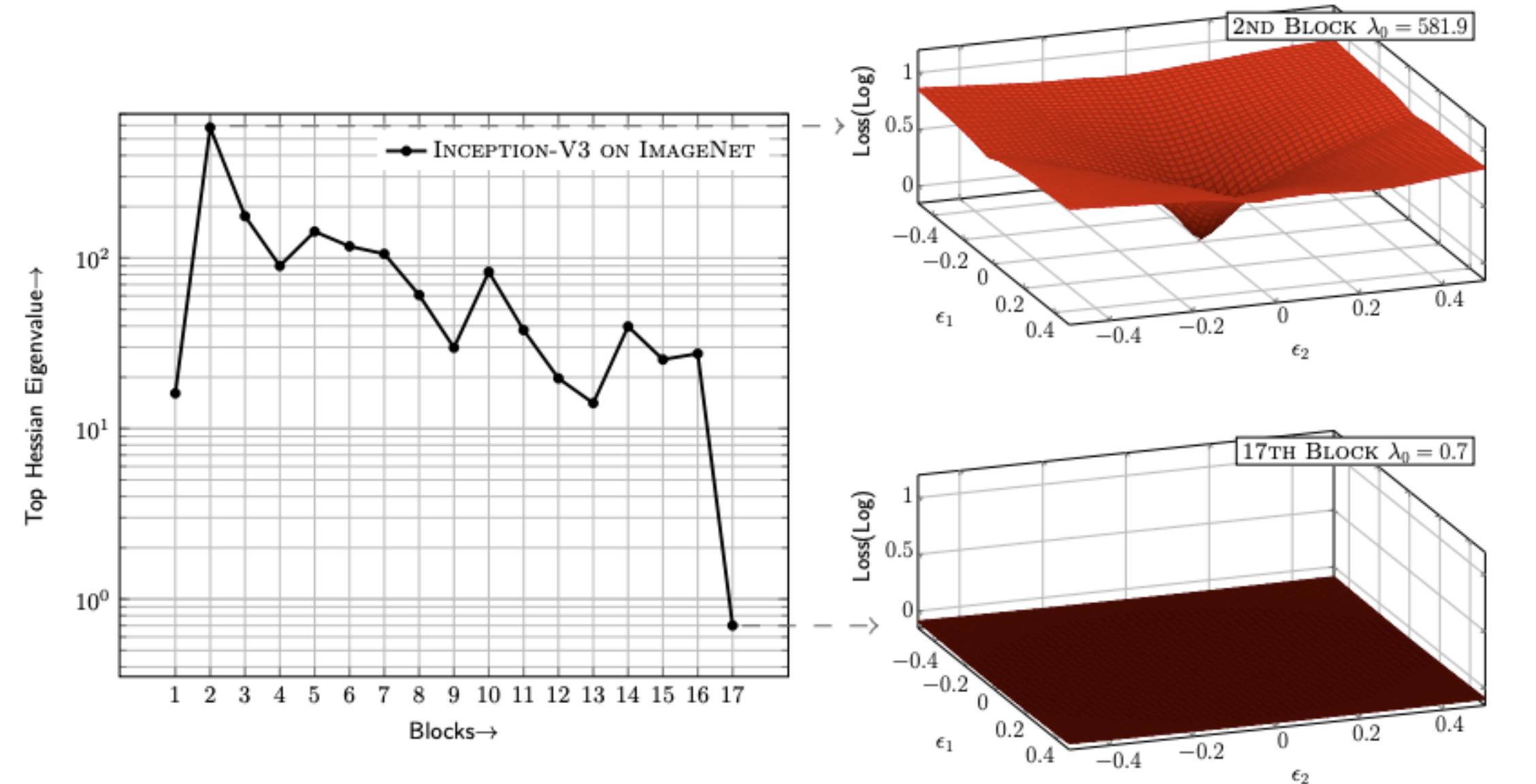


3. fig.add_subplot(Irregular Arrangement)

HAWQ: Hessian AWare Quantization of Neural Networks with Mixed-Precision

Zhen Dong*, Zhewei Yao*, Amir Gholami*, Michael W. Mahoney, Kurt Keutzer
University of California, Berkeley

{zhendong, zheweiy, amirgh, mahoneymw, and keutzer}@berkeley.edu

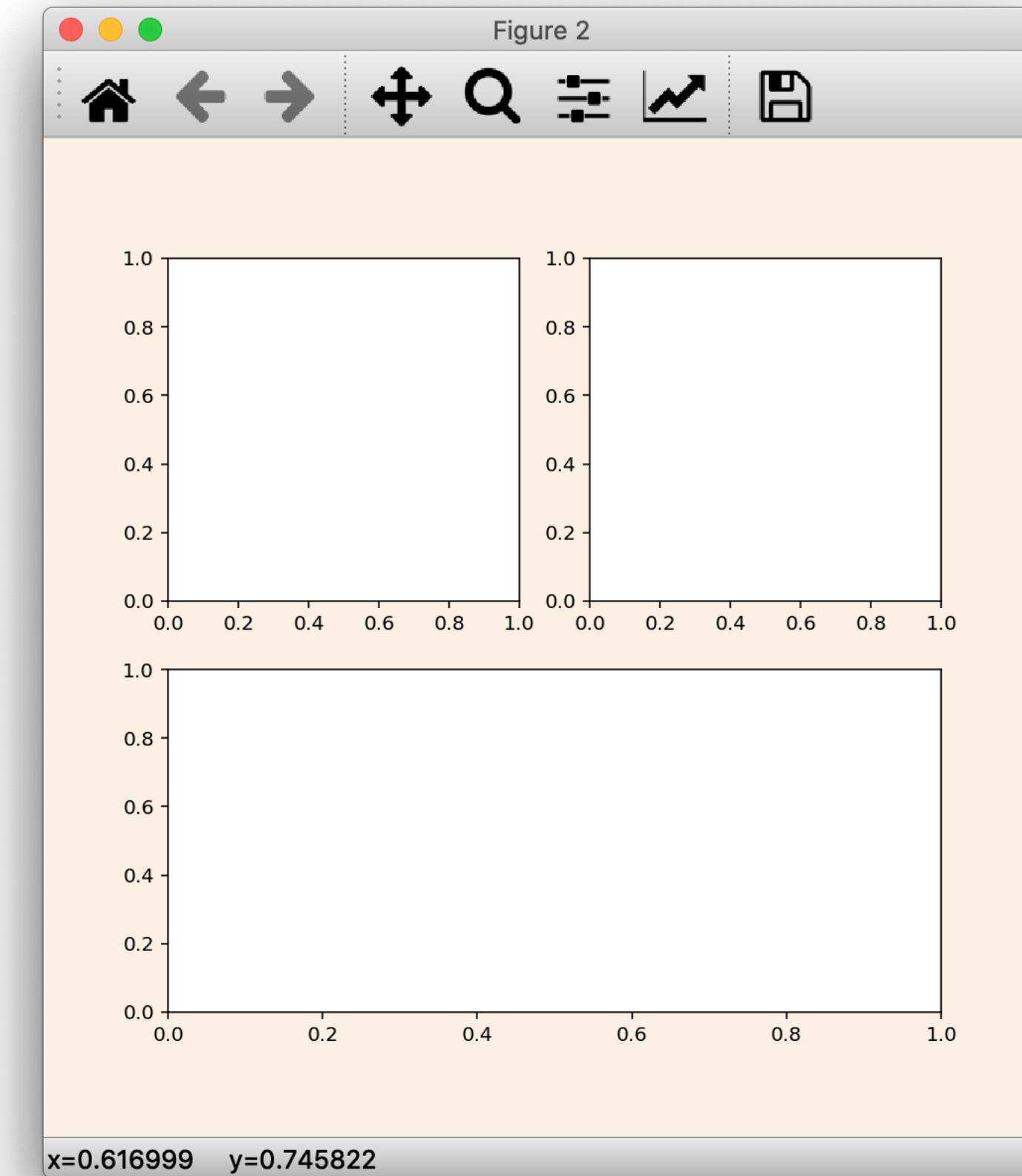


3. fig.add_subplot(Irregular Arrangement)

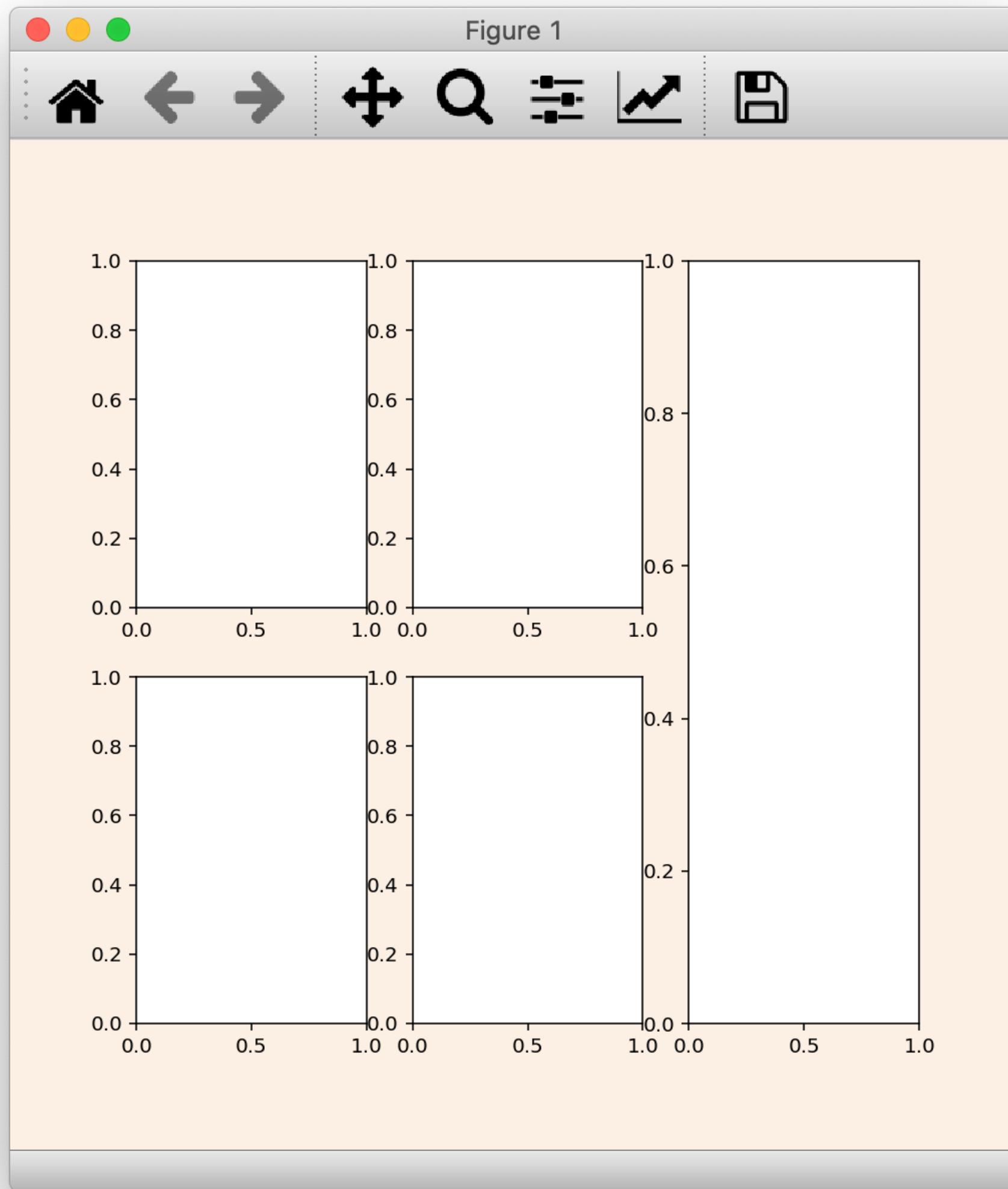
```
import matplotlib.pyplot as plt
import numpy as np

fig = plt.figure(figsize=(7, 7),
                 facecolor='linen')

ax1 = fig.add_subplot(221)
ax2 = fig.add_subplot(222)
ax3 = fig.add_subplot(212)
```



3. fig.add_subplot(Irregular Arrangement)

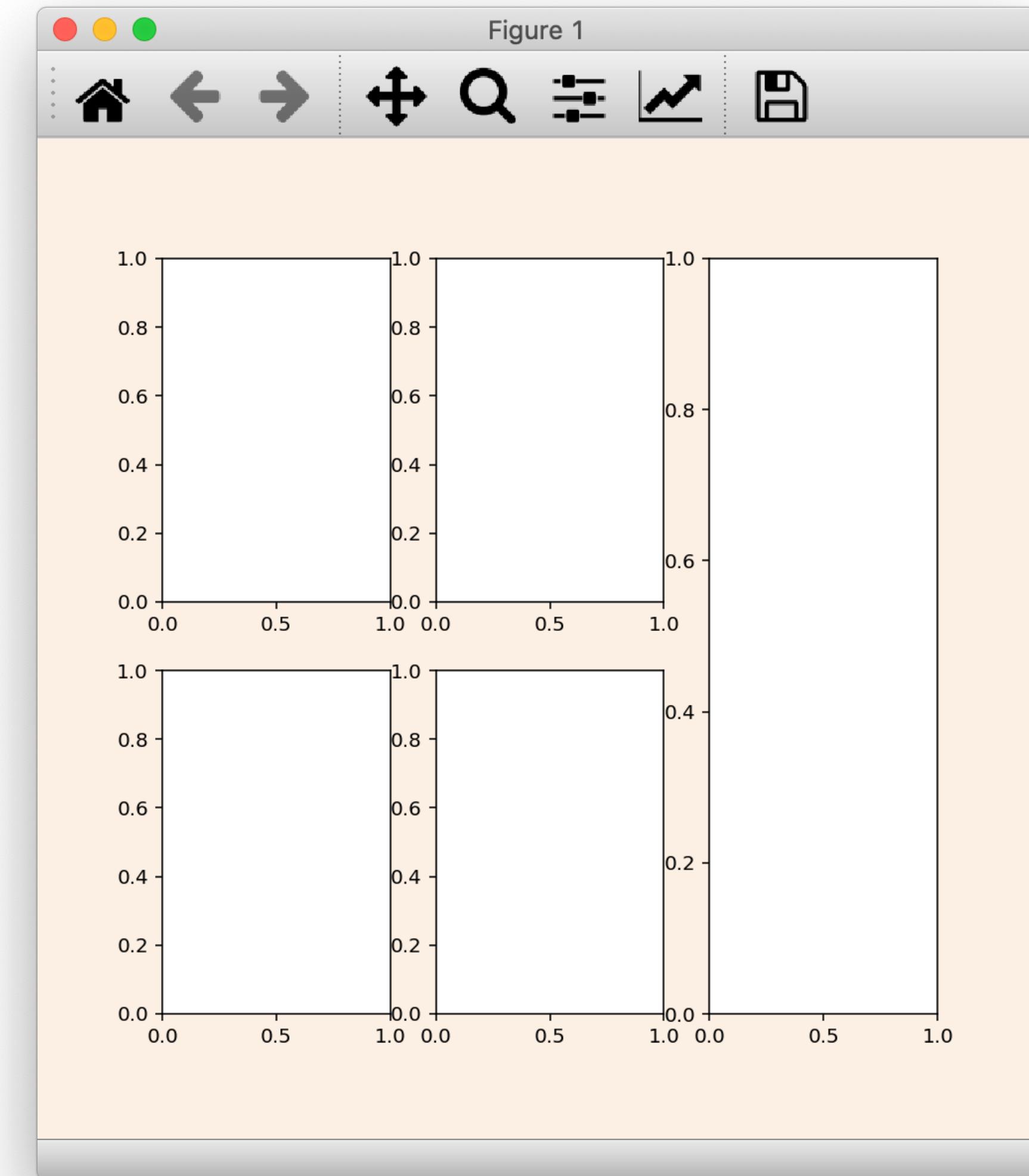


3. fig.add_subplot(Irregular Arrangement)

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

fig = plt.figure(figsize=(7, 7),
                 facecolor='linen')

ax1 = fig.add_subplot(231)
ax2 = fig.add_subplot(232)
ax3 = fig.add_subplot(234)
ax4 = fig.add_subplot(235)
ax5 = fig.add_subplot(133)
```



3. fig.add_subplot(Other Arguments)

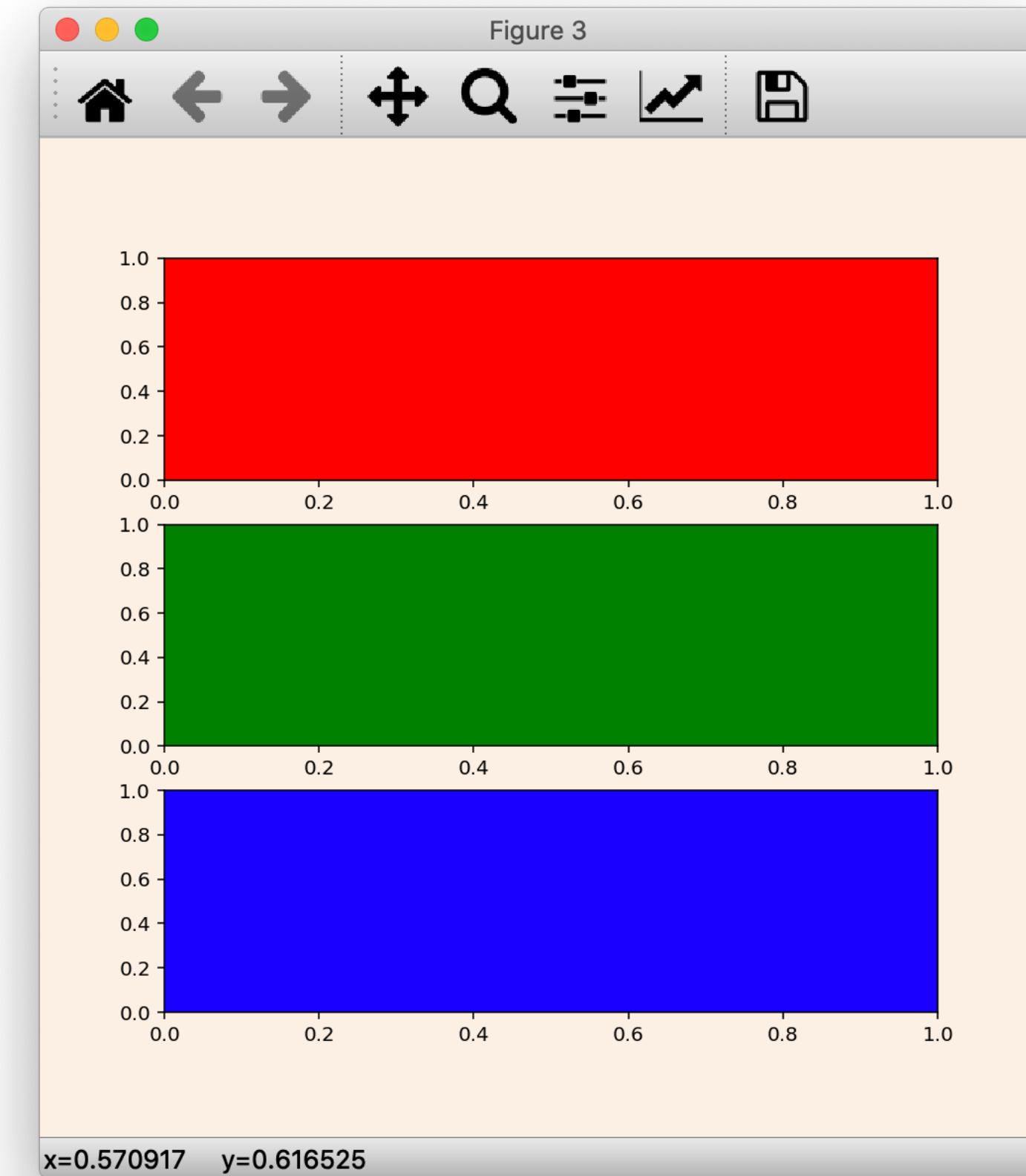
```
import matplotlib.pyplot as plt
import numpy as np

fig = plt.figure(figsize=(7, 7),
                 facecolor='linen')

ax1 = fig.add_subplot(311,
                      facecolor='r')

ax2 = fig.add_subplot(312,
                      facecolor='g')

ax3 = fig.add_subplot(313,
                      facecolor='b')
```



4. plt.subplots(Making Fig and Axes Simultaneously)

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

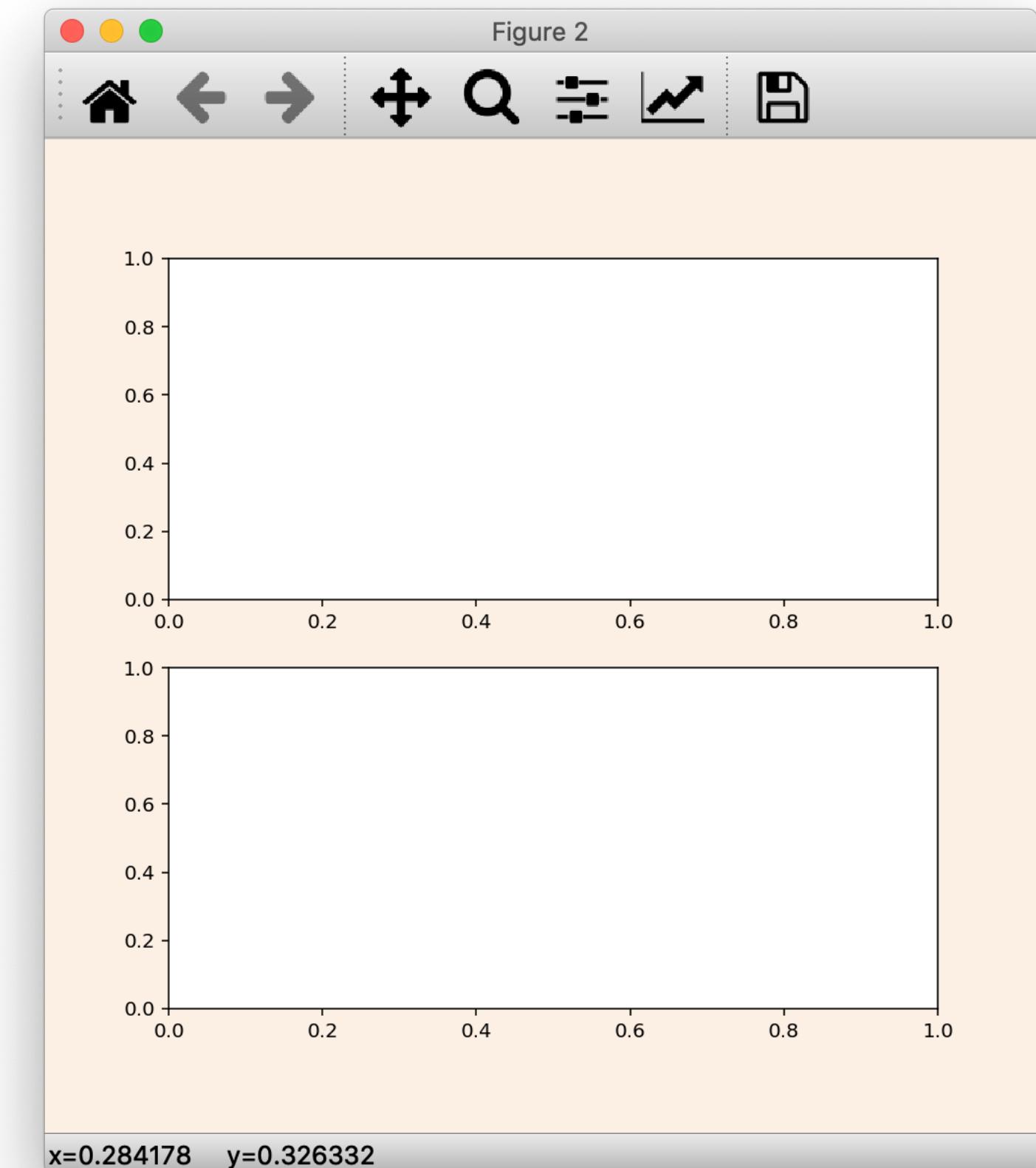
fig = plt.figure()
ax1 = fig.add_subplot(211)
ax2 = fig.add_subplot(212)

print(ax1)      AxesSubplot(0.125,0.536818;0.775x0.343182)
print(ax2)      AxesSubplot(0.125,0.125;0.775x0.343182)

fig, axes = plt.subplots(nrows=2, ncols=1)
print(axes)    [<matplotlib.axes._subplots.AxesSubplot object at 0x7ff8b7d95e10>
                <matplotlib.axes._subplots.AxesSubplot object at 0x7ff8b7dd63d0>]

print(type(axes)) <class 'numpy.ndarray'>

print(axes[0])  AxesSubplot(0.125,0.536818;0.775x0.343182)
print(axes[1])  AxesSubplot(0.125,0.125;0.775x0.343182)
```



4. plt.subplots(nrows and ncols)

matplotlib.pyplot.subplots

```
matplotlib.pyplot.subplots(nrows=1, ncols=1, *, sharex=False, sharey=False, squeeze=True, subplot_kw=None,  
gridspec_kw=None, **fig_kw)
```

[\[source\]](#)

Create a figure and a set of subplots.

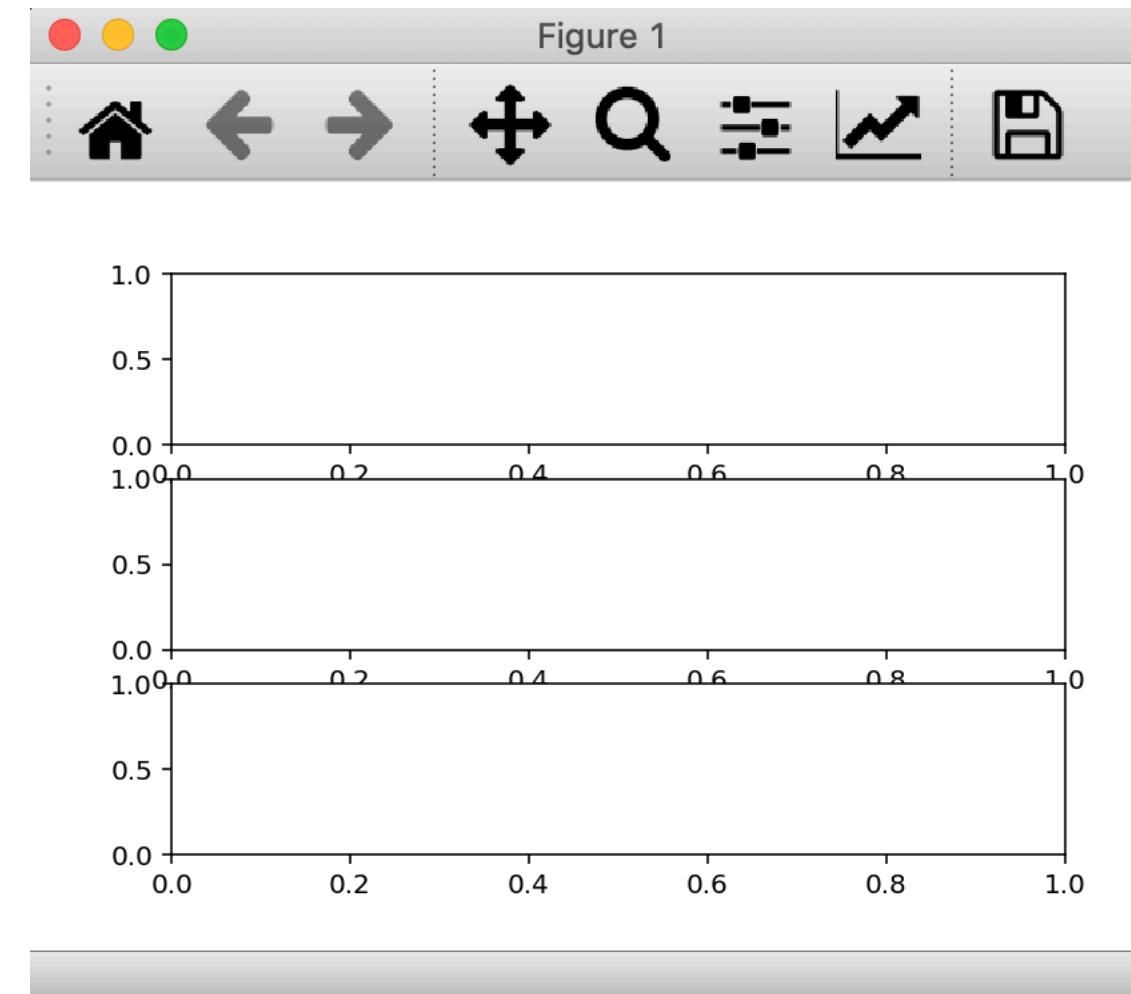
This utility wrapper makes it convenient to create common layouts of subplots, including the enclosing figure object, in a single call.

Usage

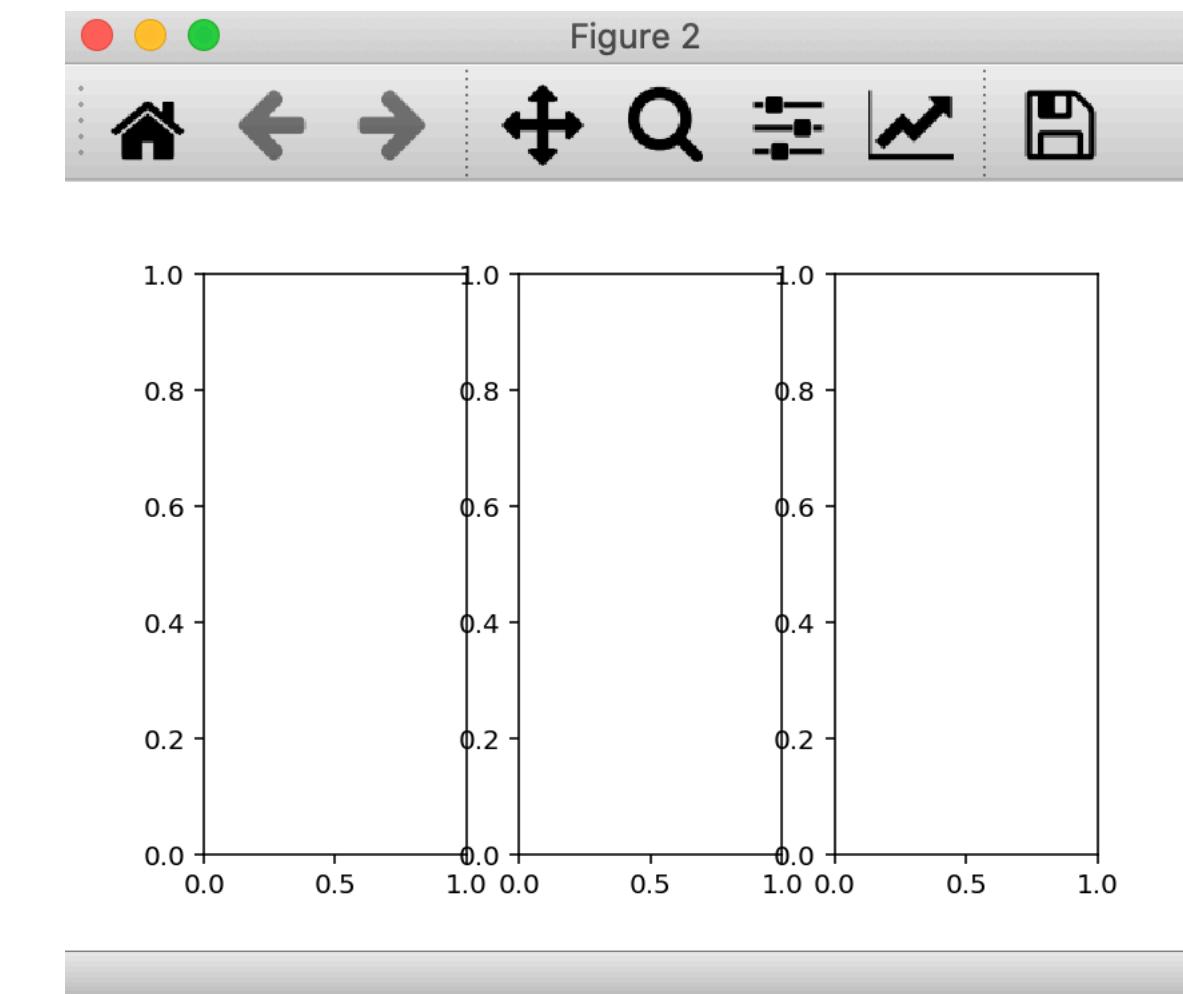
```
fig, axes = plt.subplots()
```

4. plt.subplots(nrows and ncols)

```
import matplotlib.pyplot as plt  
import numpy as np  
  
fig, axes = plt.subplots(nrows=3, ncols=1)  
  
print(axes[0])  
print(axes[1])  
print(axes[2])  
  
AxesSubplot(0.125,0.657941;0.775x0.222059)  
AxesSubplot(0.125,0.391471;0.775x0.222059)  
AxesSubplot(0.125,0.125;0.775x0.222059)
```



```
fig, axes = plt.subplots(nrows=1, ncols=3)  
  
print(axes[0])  
print(axes[1])  
print(axes[2])  
  
AxesSubplot(0.125,0.125;0.227941x0.755)  
AxesSubplot(0.398529,0.125;0.227941x0.755)  
AxesSubplot(0.672059,0.125;0.227941x0.755)
```



Lecture_1-01 Making Figures and Axes

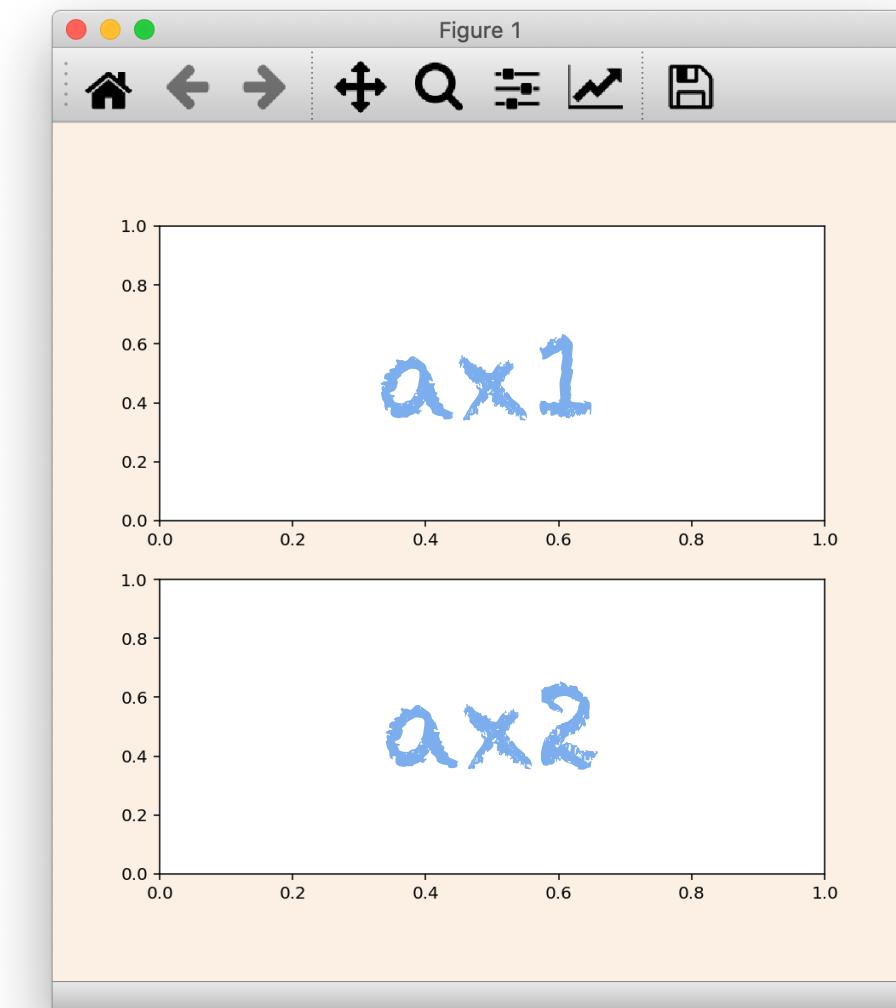
33

4. plt.subplots(nrows and ncols)

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

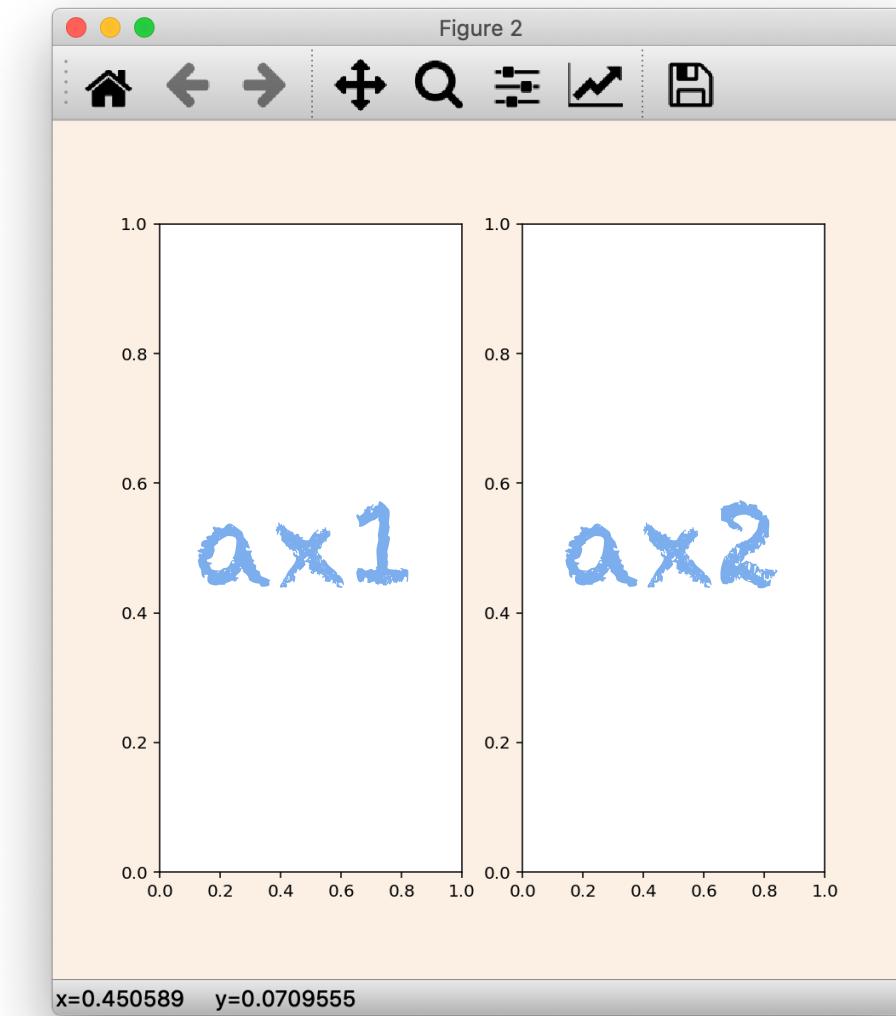
fig, (ax1, ax2) = plt.subplots(2, 1,
                             figsize=(7, 7),
                             facecolor='linen')

print(ax1)      AxesSubplot(0.125,0.536818;0.775x0.343182)
print(ax2)      AxesSubplot(0.125,0.125;0.775x0.343182)
```



```
fig, (ax1, ax2) = plt.subplots(1, 2,
                             figsize=(7, 7),
                             facecolor='linen')

print(ax1)      AxesSubplot(0.125,0.125;0.352273x0.755)
print(ax2)      AxesSubplot(0.547727,0.125;0.352273x0.755)
```



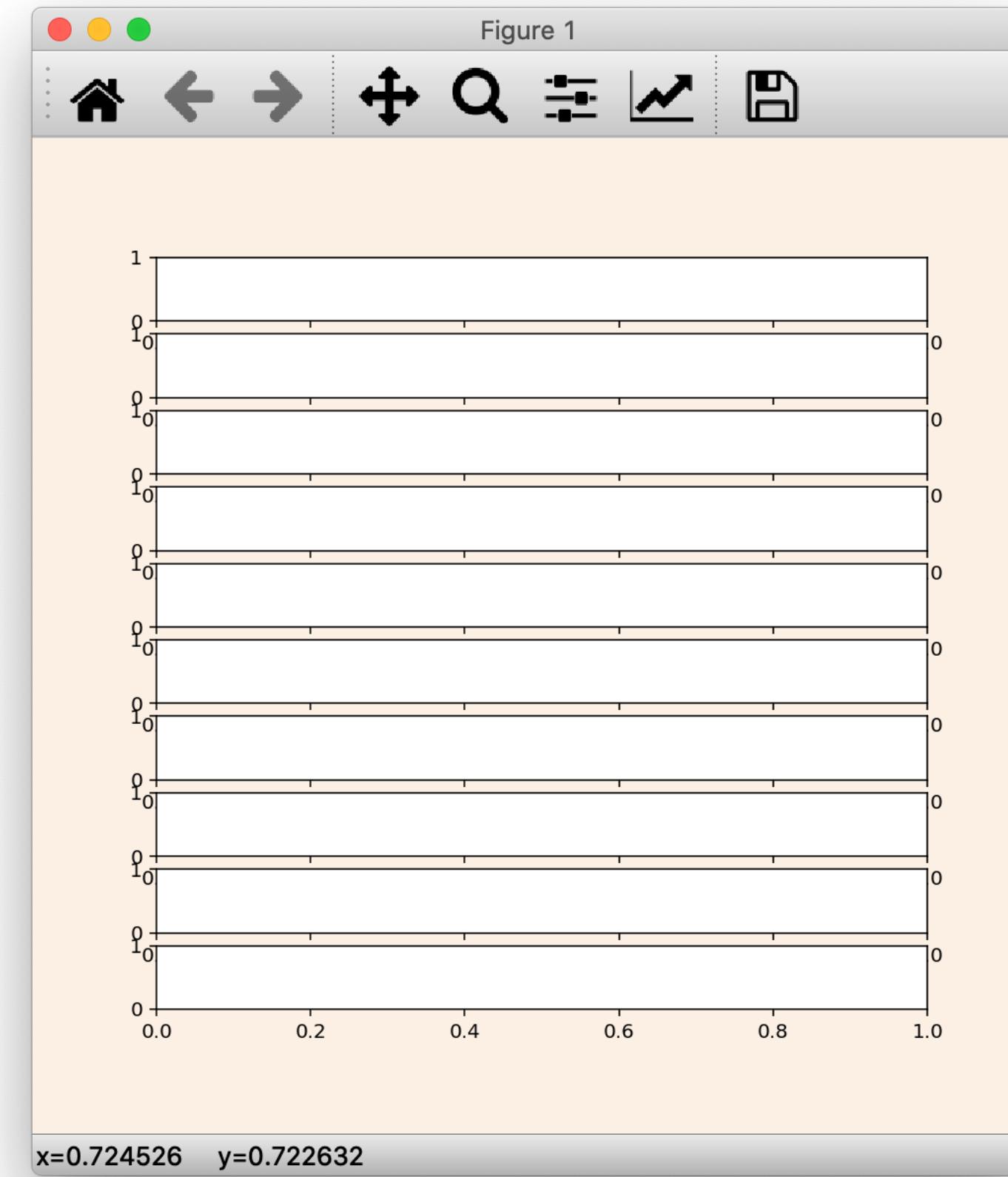
4. plt.subplots(nrows and ncols)

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

fig, axes = plt.subplots(10, 1,
                       figsize=(7, 7),
                       facecolor='linen')

for ax in axes:
    print(ax)

AxesSubplot(0.125,0.816017;0.775x0.0639831)
AxesSubplot(0.125,0.739237;0.775x0.0639831)
AxesSubplot(0.125,0.662458;0.775x0.0639831)
AxesSubplot(0.125,0.585678;0.775x0.0639831)
AxesSubplot(0.125,0.508898;0.775x0.0639831)
AxesSubplot(0.125,0.432119;0.775x0.0639831)
AxesSubplot(0.125,0.355339;0.775x0.0639831)
AxesSubplot(0.125,0.278559;0.775x0.0639831)
AxesSubplot(0.125,0.20178;0.775x0.0639831)
AxesSubplot(0.125,0.125;0.775x0.0639831)
```



4. plt.subplots(Indexing 2-D Axes)

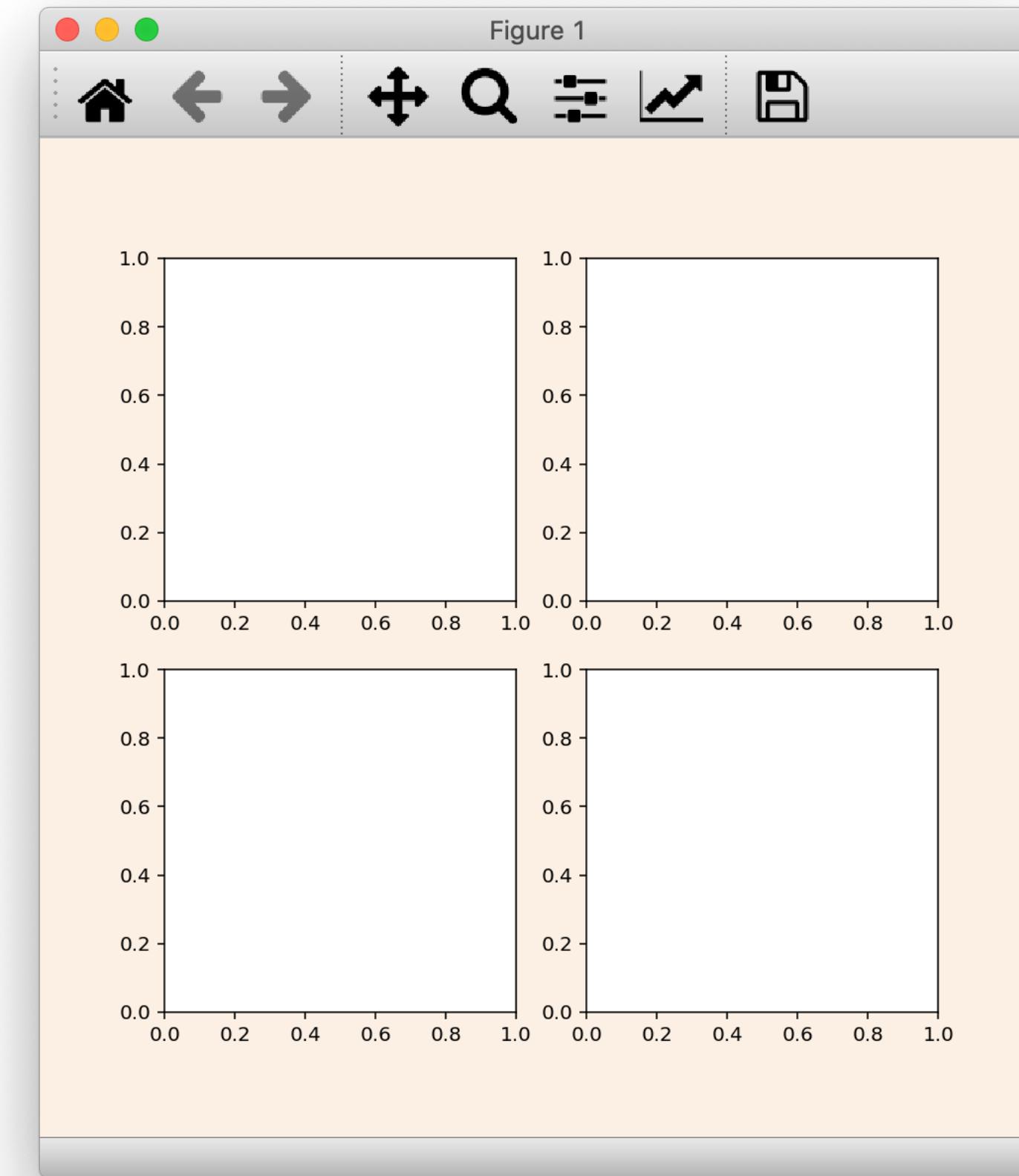
```
import matplotlib.pyplot as plt
import numpy as np

fig, axes = plt.subplots(2, 2,
                        figsize=(7, 7),
                        facecolor='linen')

print(axes)

[[<matplotlib.axes._subplots.AxesSubplot object at 0x7ff8b547f050>
 <matplotlib.axes._subplots.AxesSubplot object at 0x7ff8b89e3990>]
 [<matplotlib.axes._subplots.AxesSubplot object at 0x7ff8b53297d0>
 <matplotlib.axes._subplots.AxesSubplot object at 0x7ff8b5365e10>]]

print(axes.shape)  (2, 2)
```



4. plt.subplots(Indexing 2-D Axes)

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

fig, axes = plt.subplots(2, 2,
                        figsize=(7, 7),
                        facecolor='linen')

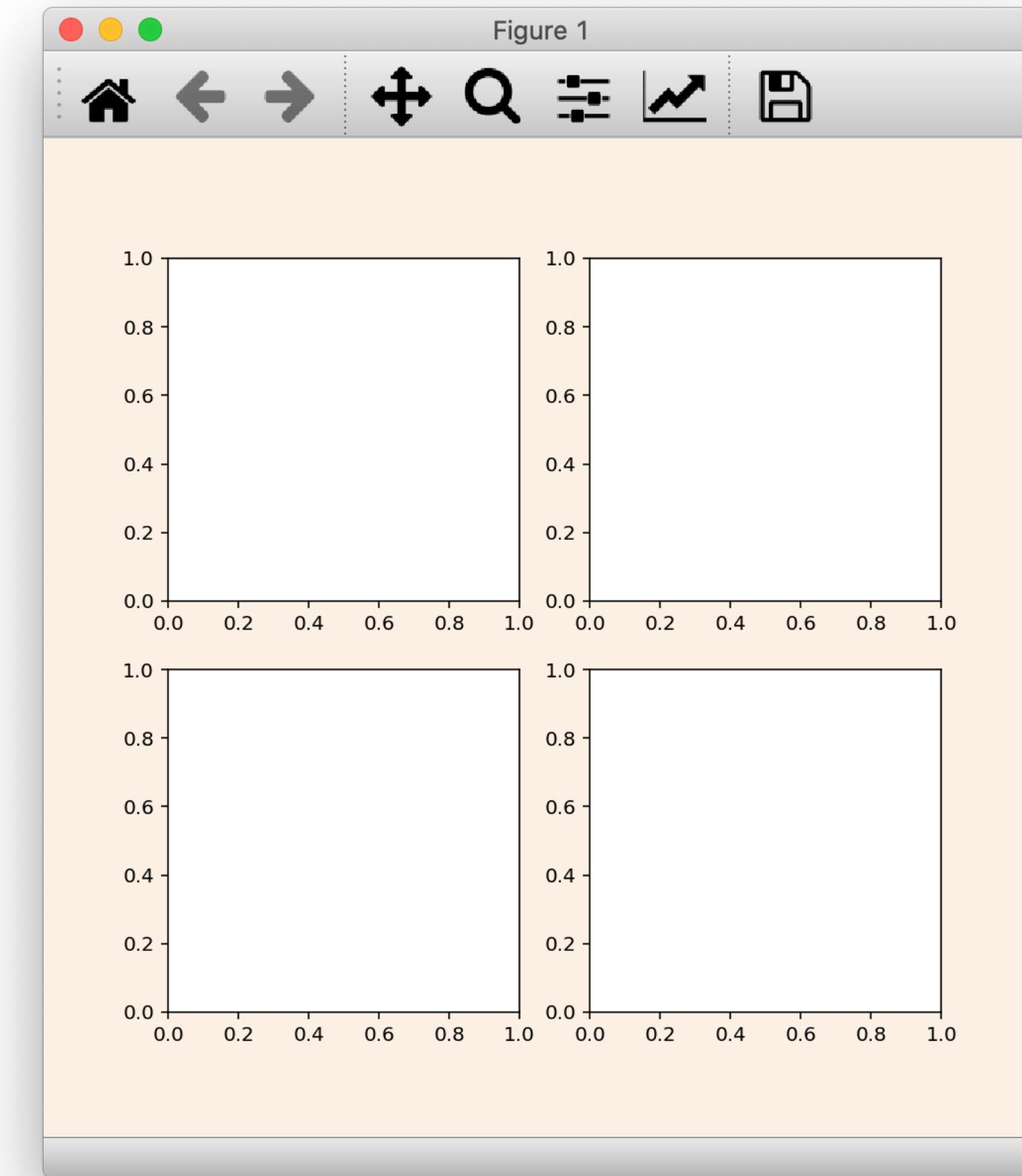
print(axes)

[[<matplotlib.axes._subplots.AxesSubplot object at 0x7ff8b547f050>
 <matplotlib.axes._subplots.AxesSubplot object at 0x7ff8b89e3990>]
 [<matplotlib.axes._subplots.AxesSubplot object at 0x7ff8b53297d0>
 <matplotlib.axes._subplots.AxesSubplot object at 0x7ff8b5365e10>]]

print(axes[0]) [<matplotlib.axes._subplots.AxesSubplot object at 0x7f884ddc68d0>
 <matplotlib.axes._subplots.AxesSubplot object at 0x7f884d8892d0>]

print(axes[1]) [<matplotlib.axes._subplots.AxesSubplot object at 0x7f8852ec1850>
 <matplotlib.axes._subplots.AxesSubplot object at 0x7f8852ecddd0>]

print(axes[0, 0]) AxesSubplot(0.125,0.536818;0.352273x0.343182)
print(axes[0, 1]) AxesSubplot(0.547727,0.536818;0.352273x0.343182)
print(axes[1, 0]) AxesSubplot(0.125,0.125;0.352273x0.343182)
print(axes[1, 1]) AxesSubplot(0.547727,0.125;0.352273x0.343182)
```



4. plt.subplots(ndarray.flat and enumerate)

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

test_np = np.array([[1, 2], [3, 4]])
print(test_np) [[1 2]
                [3 4]]
print(test_np[0]) [1 2]
print(test_np[1]) [3 4]
```

```
for val in test_np:
    print(val) [1 2]
                [3 4]
```

```
print(test_np[0, 0]) 1
print(test_np[0, 1]) 2
print(test_np[1, 0]) 3
print(test_np[1, 1]) 4
```

```
for val in test_np.flat:
    print(val) 1
                2
                3
                4
```

4. plt.subplots(ndarray.flat and enumerate)

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

```
idx = 0
test_list = [10, 11, 12, 13]
```

```
for val in test_list:
    print(idx, val)
    idx += 1
```

0	10
1	11
2	12
3	13

```
for idx, val in enumerate(test_list):
    print(idx, val)
```

0	10
1	11
2	12
3	13

4. plt.subplots(Axes Indexing)

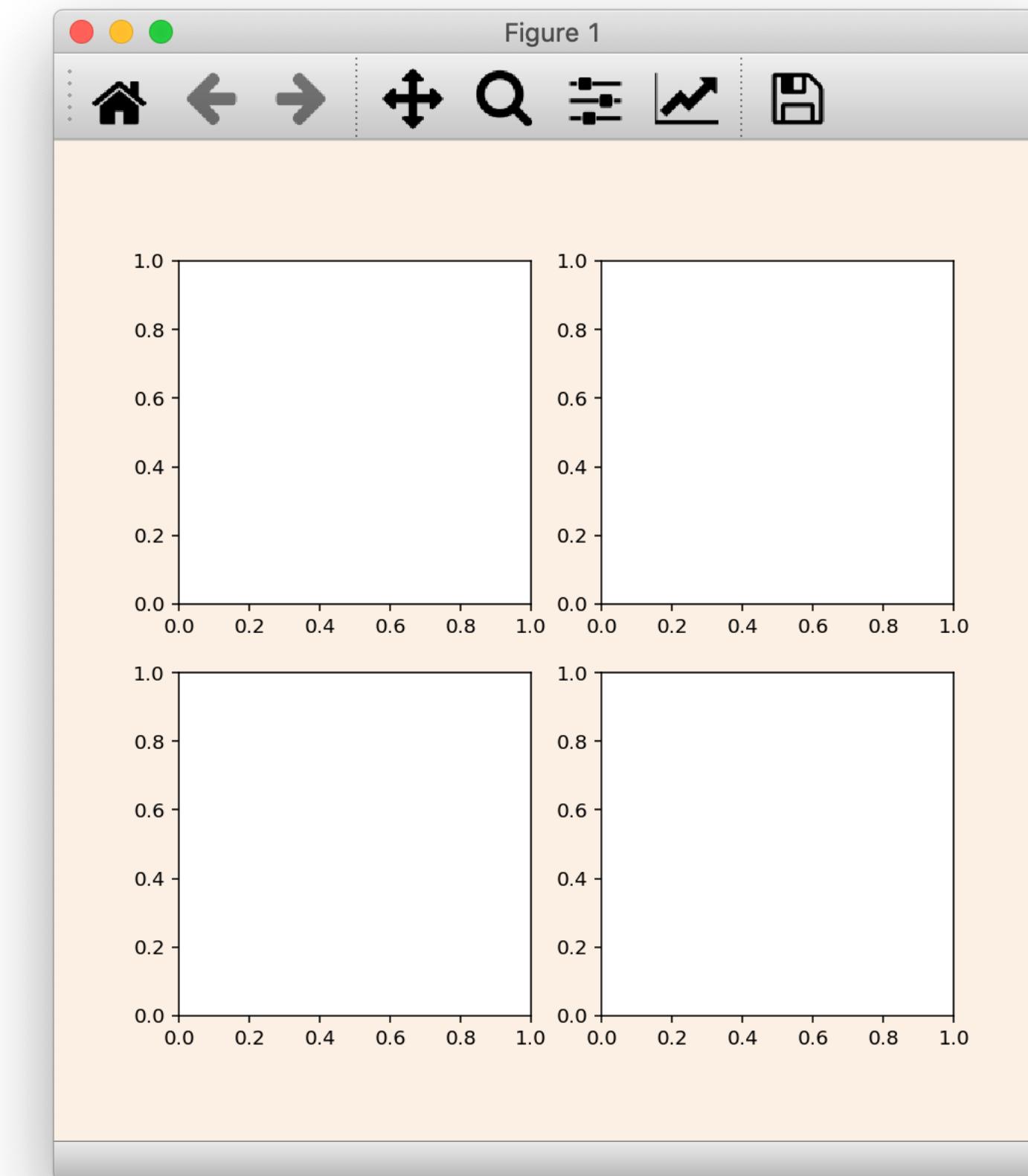
```
import matplotlib.pyplot as plt
import numpy as np

fig, axes = plt.subplots(nrows=2, ncols=2,
                        figsize=(7, 7),
                        facecolor='linen')

for ax in axes.flat:
    print(ax) AxesSubplot(0.125,0.536818;0.352273x0.343182)
    AxesSubplot(0.547727,0.536818;0.352273x0.343182)
    AxesSubplot(0.125,0.125;0.352273x0.343182)
    AxesSubplot(0.547727,0.125;0.352273x0.343182)
```

```
for val in test_np.flat:
    print(val)

for idx, val in enumerate(test_list):
    print(idx, val)
```



4. plt.subplots(Axes Indexing)

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

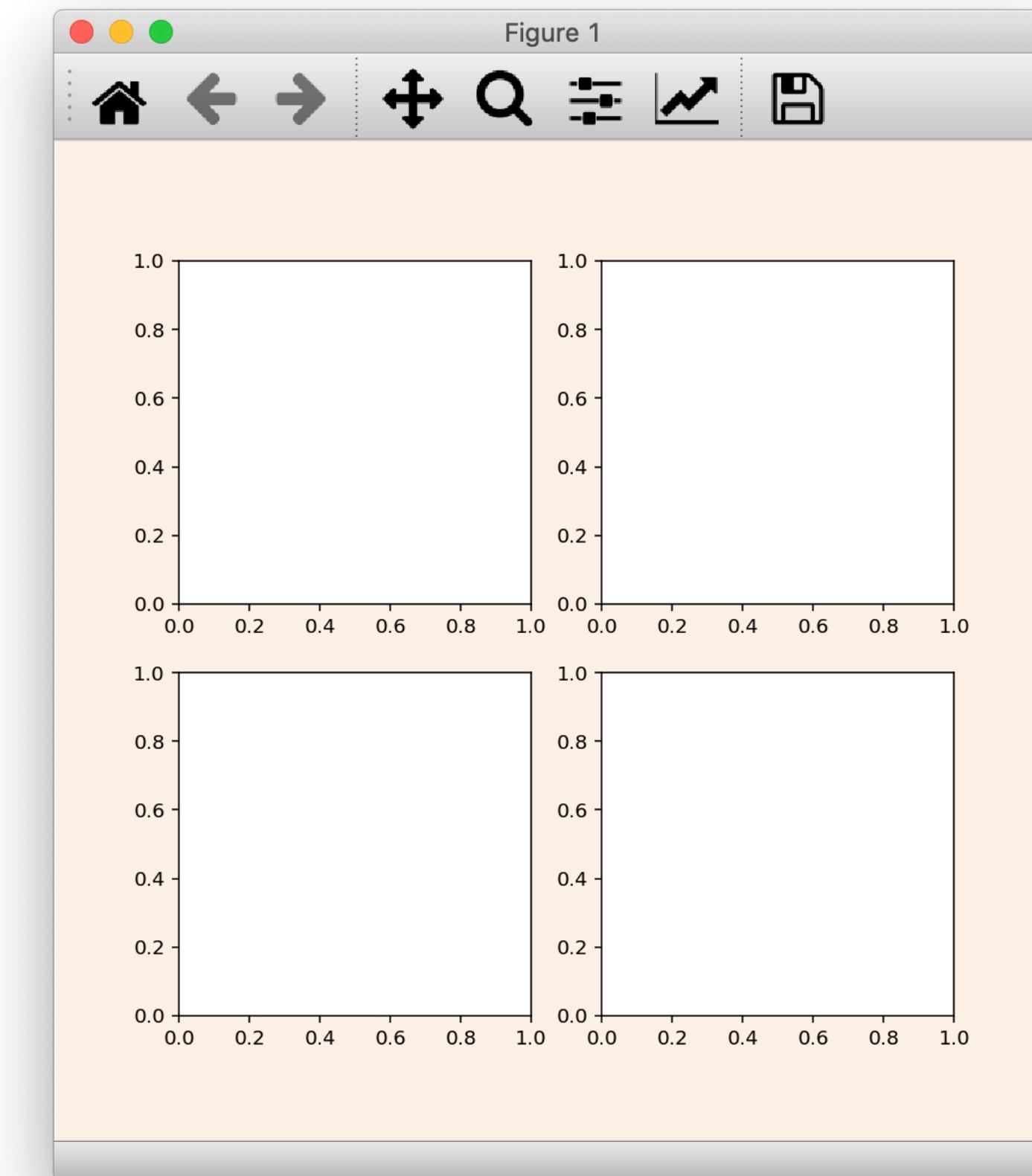
fig, axes = plt.subplots(nrows=2, ncols=2,
                        figsize=(7, 7),
                        facecolor='linen')

for ax_idx, ax in enumerate(axes.flat):
    print(ax_idx, ax)

0 AxesSubplot(0.125,0.536818;0.352273x0.343182)
1 AxesSubplot(0.547727,0.536818;0.352273x0.343182)
2 AxesSubplot(0.125,0.125;0.352273x0.343182)
3 AxesSubplot(0.547727,0.125;0.352273x0.343182)
```

```
for val in test_np.flat:
    print(val)

for idx, val in enumerate(test_list):
    print(idx, val)
```



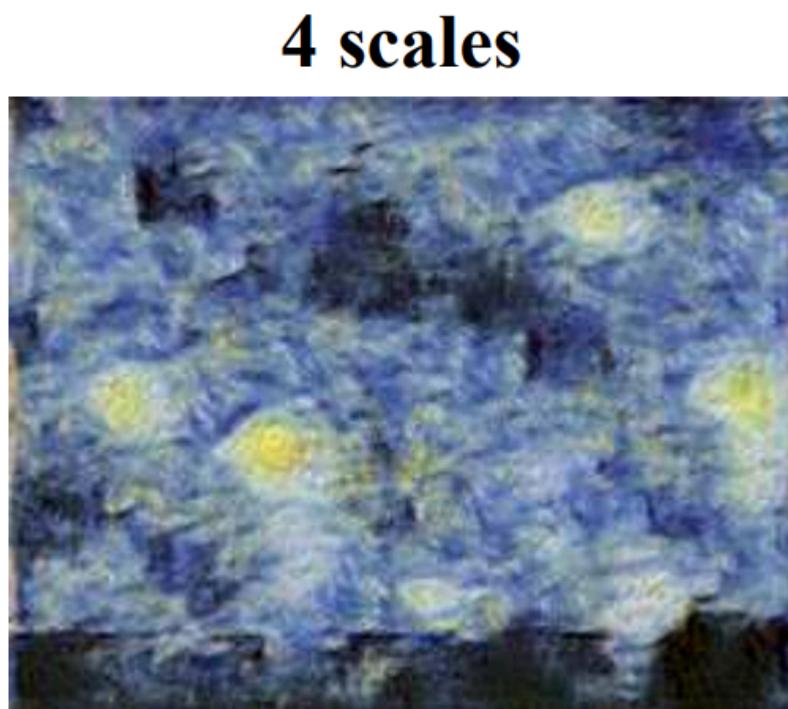
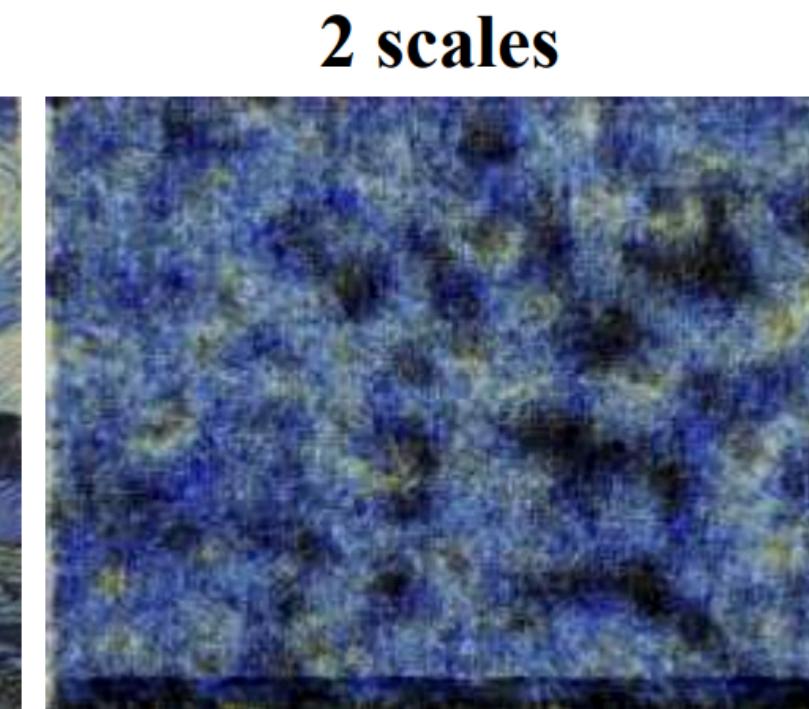
4. plt.subplots(Axes Indexing Example)

SinGAN: Learning a Generative Model from a Single Natural Image

Tamar Rott Shaham
Technion

Tali Dekel
Google Research

Tomer Michaeli
Technion



```
title_list = ['Training Image', '2 scales',
              '4 scales', '5 scales',
              '6 scales', '8 scales']
```

```
print(title_list[0])      Training Image
print(title_list[1])      2 scales
print(title_list[2])      4 scales
print(title_list[3])      5 scales
print(title_list[4])      6 scales
print(title_list[5])      8 scales
```

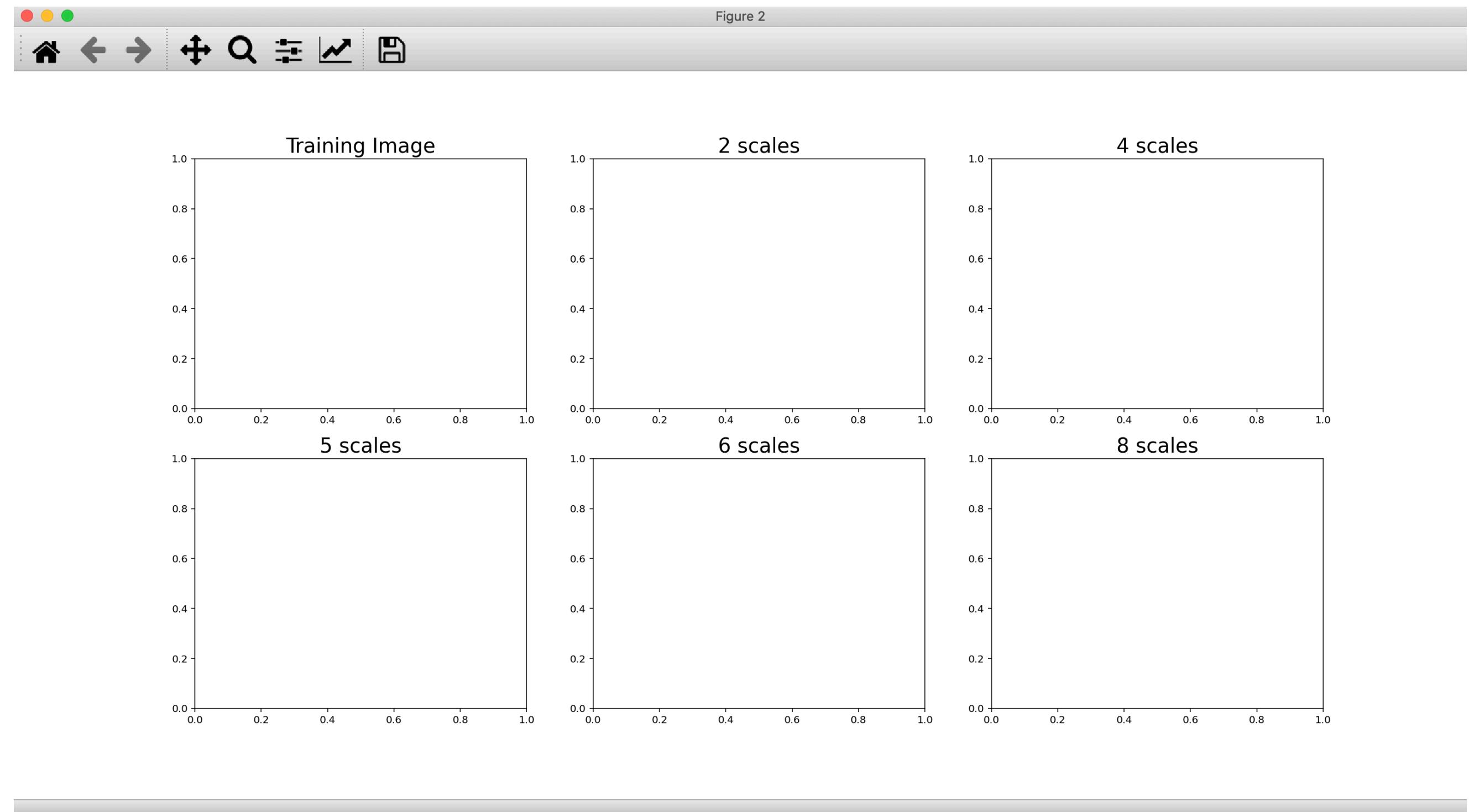
4. plt.subplots(Axes Indexing Example)

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

title_list = ['Training Image', '2 scales',
              '4 scales', '5 scales',
              '6 scales', '8 scales']

fig, axes = plt.subplots(2, 3,
                        figsize=(20, 10))

for ax_idx, ax in enumerate(axes.flat):
    ax.set_title(title_list[ax_idx],
                 fontsize=20)
```

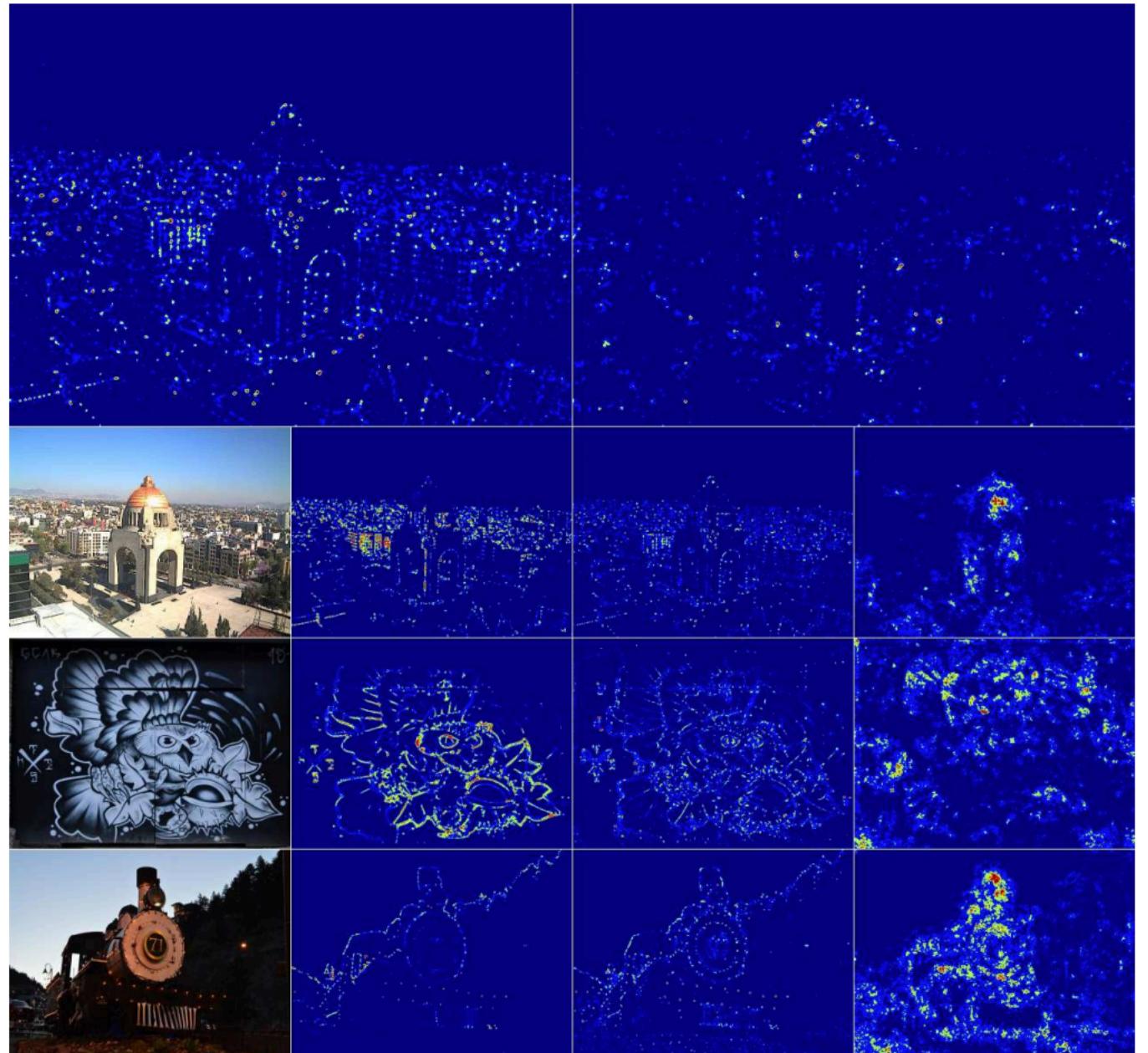


5. plt.subplot2grid(More Complex Arrangement)

ELF: Embedded Localisation of Features in Pre-Trained CNN

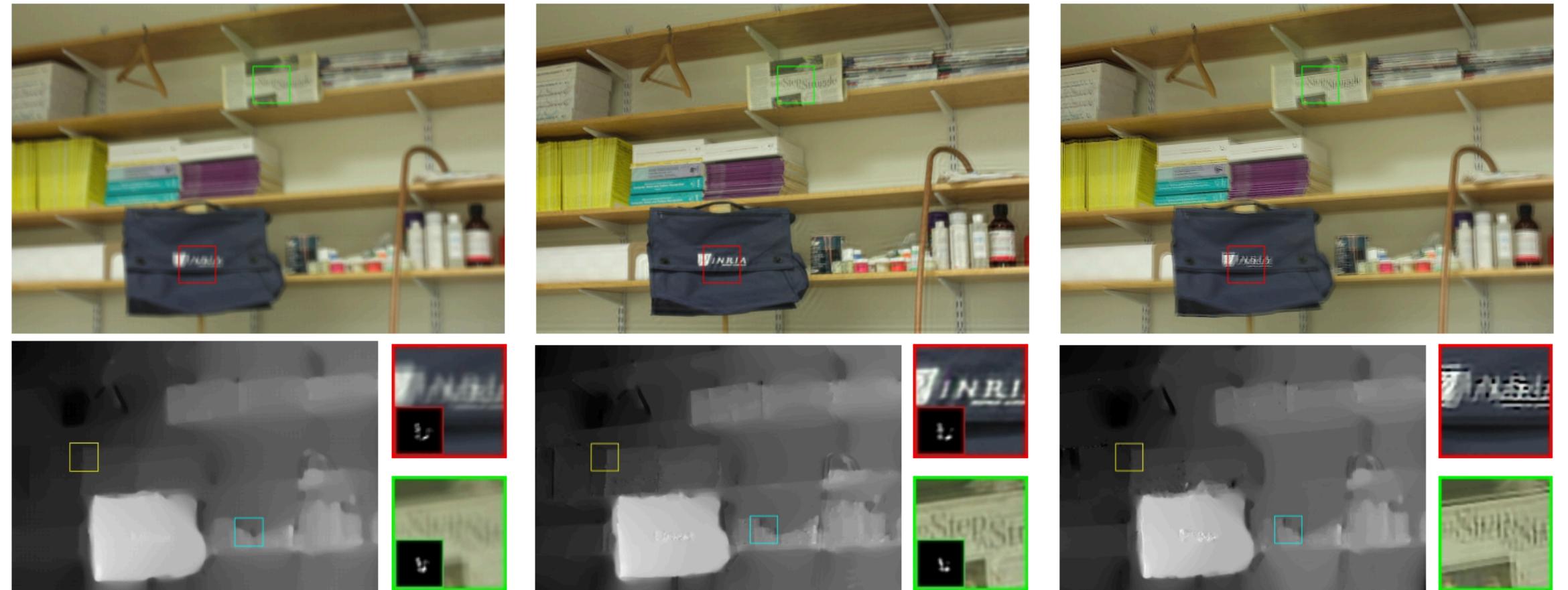
Assia Benbihi
UMI2958 GeorgiaTech-CNRS
Centrale Supélec
Université Paris-Saclay
Metz, France

abenbihi@georgiatech-metz.fr

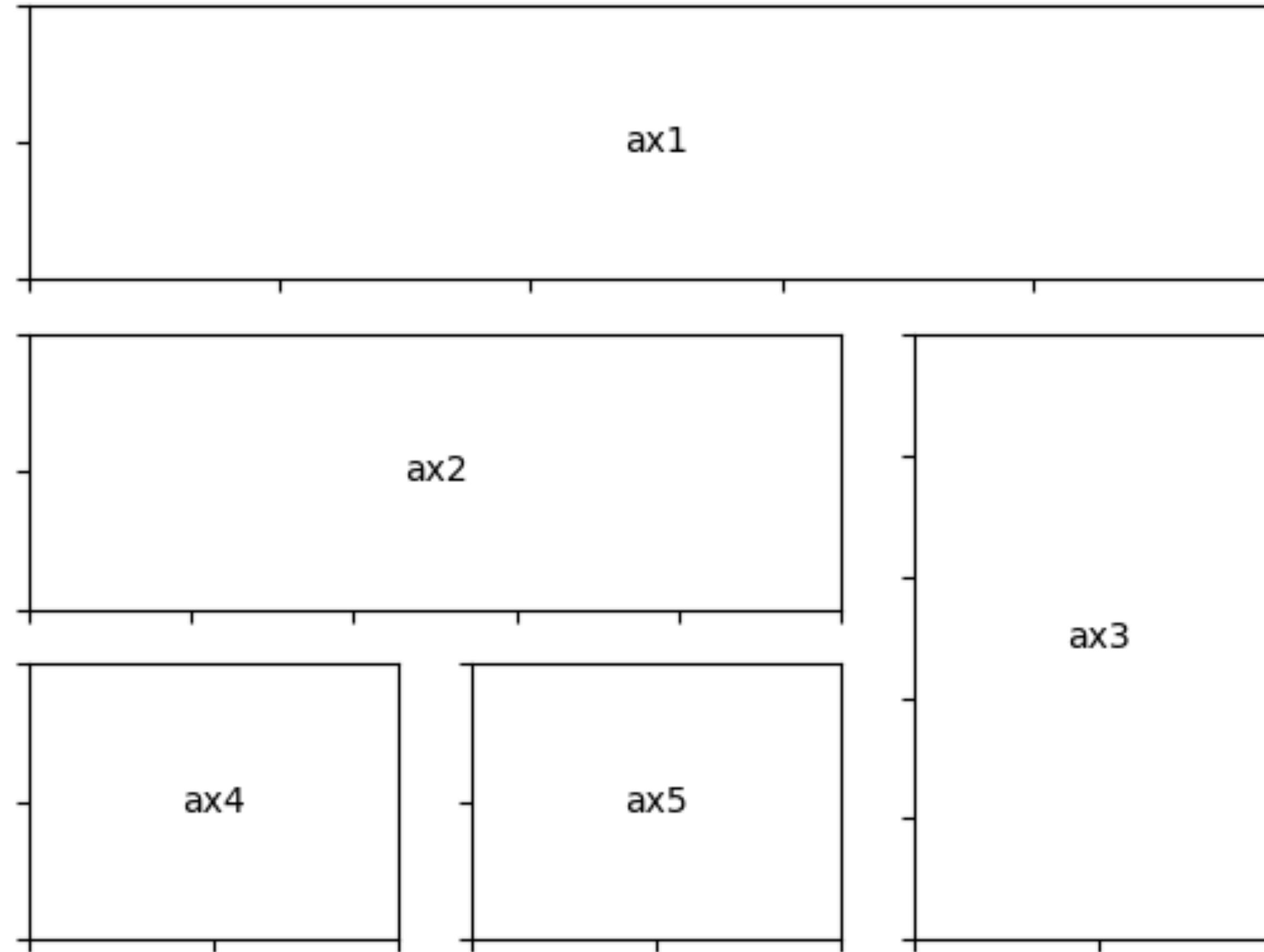


Unconstrained Motion Deblurring for Dual-lens Cameras

M. R. Mahesh Mohan, Sharath Girish, and A. N. Rajagopalan
Indian Institute of Technology Madras
{ee14d023, ee15b058, raju}@ee.iitm.ac.in



5. plt.subplot2grid(More Complex Arrangement)



5. plt.subplot2grid(Arguments)

matplotlib.pyplot.subplot2grid

```
matplotlib.pyplot.subplot2grid(shape, loc, rowspan=1, colspan=1, fig=None, **kwargs)
```

[\[source\]](#)

Create a subplot at a specific location inside a regular grid.

Parameters:

shape : (int, int)

Number of rows and of columns of the grid in which to place axis.

loc : (int, int)

Row number and column number of the axis location within the grid.

rowspan : int, default: 1

Number of rows for the axis to span to the right.

colspan : int, default: 1

Number of columns for the axis to span downwards.

Usage

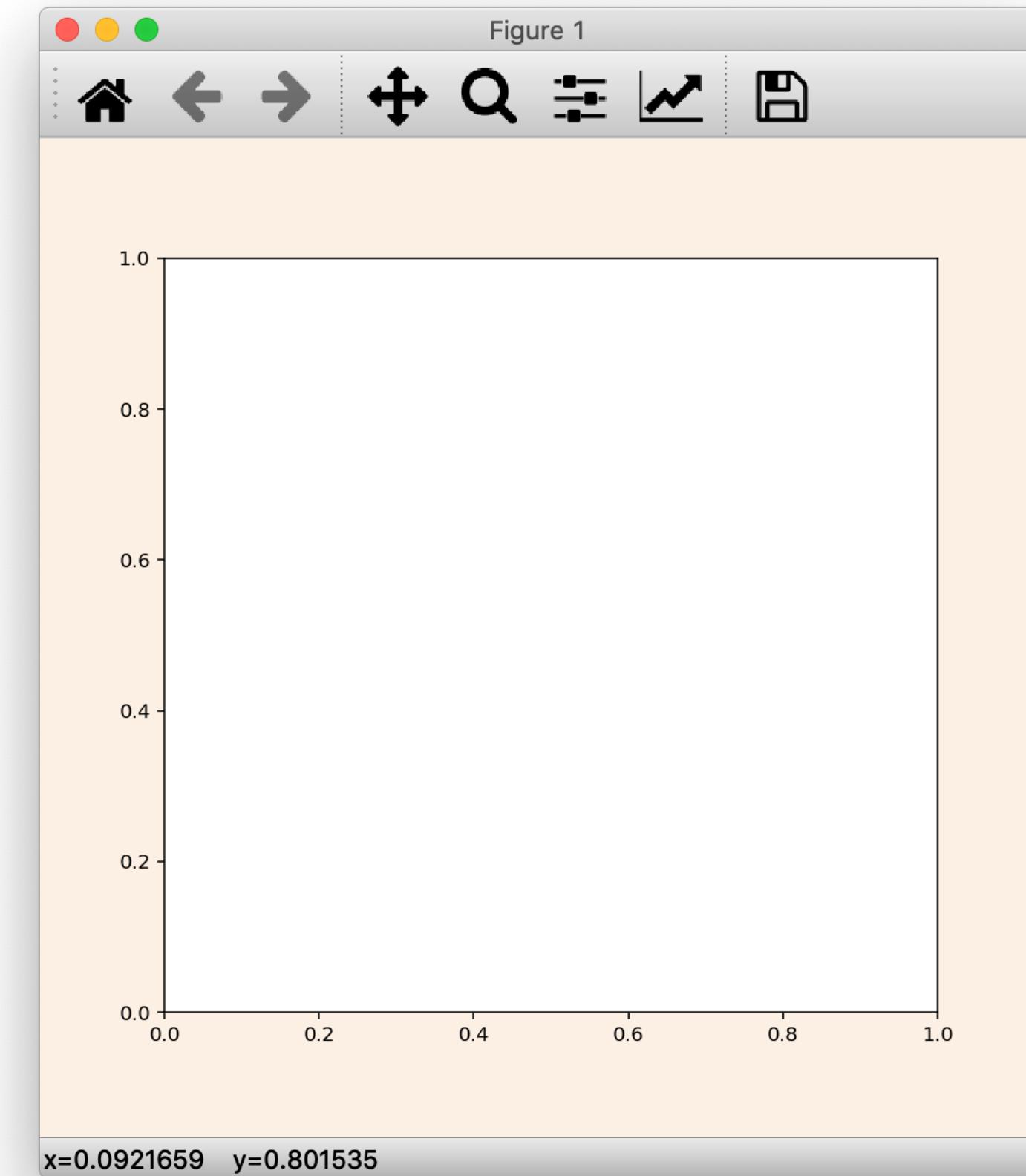
```
fig = plt.figure()
ax = plt.subplot2grid(( , ), ( , ),
                      rowspan=, colspan=,
                      fig=fig)
```

5. plt.subplot2grid(Single Ax)

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

fig = plt.figure(figsize=(7, 7),
                 facecolor='linen')

ax1 = plt.subplot2grid((1, 1), (0, 0), fig=fig)
```



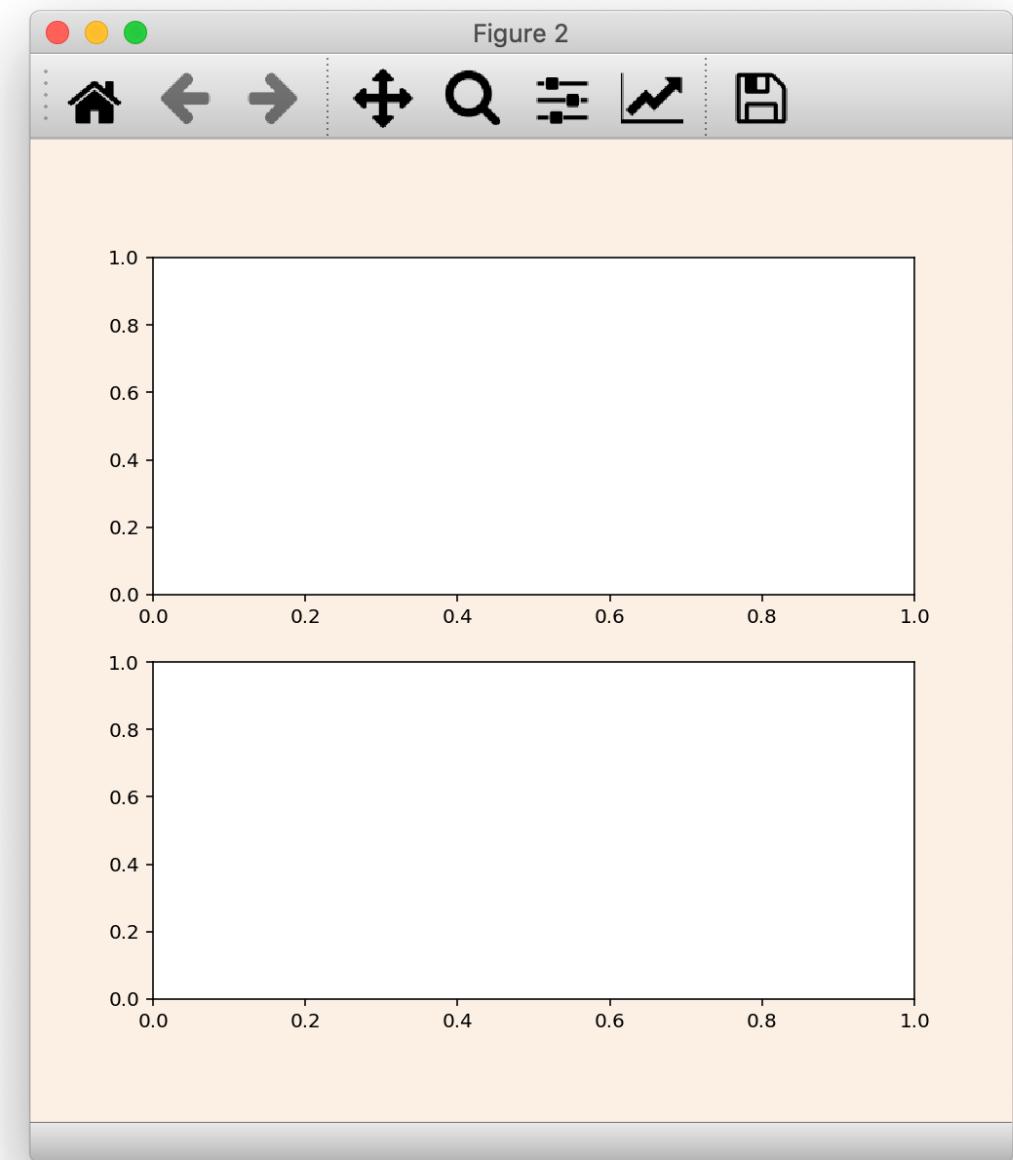
5. plt.subplot2grid(Single Ax)

```
import matplotlib.pyplot as plt
```

```
import numpy as np
```

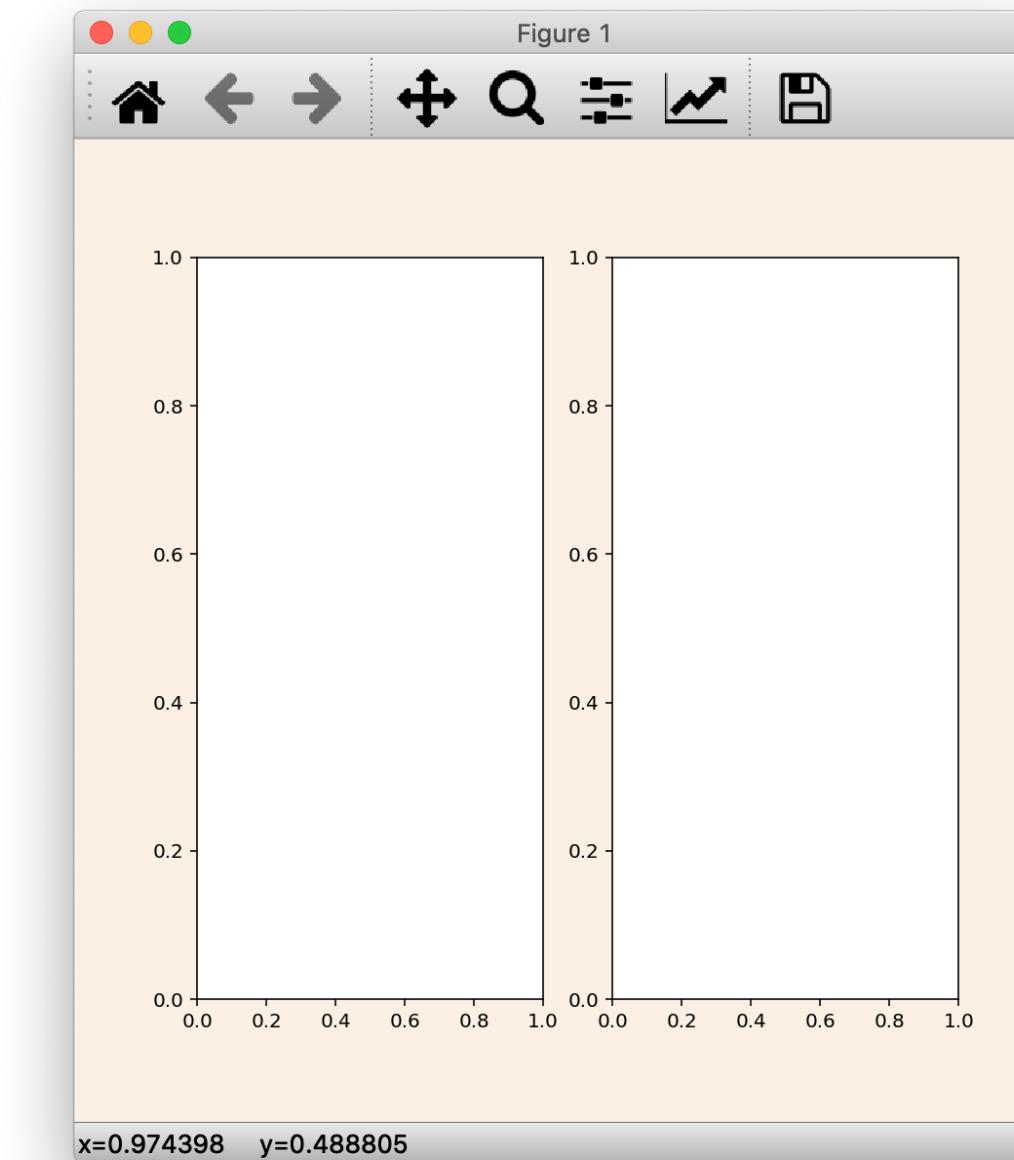
```
fig = plt.figure(figsize=(7, 7),  
                 facecolor='linen')
```

```
ax1 = plt.subplot2grid((2, 1), (0, 0), fig=fig)  
ax2 = plt.subplot2grid((2, 1), (1, 0), fig=fig)
```



```
fig = plt.figure(figsize=(7, 7),  
                 facecolor='linen')
```

```
ax1 = plt.subplot2grid((1, 2), (0, 0), fig=fig)  
ax2 = plt.subplot2grid((1, 2), (0, 1), fig=fig)
```



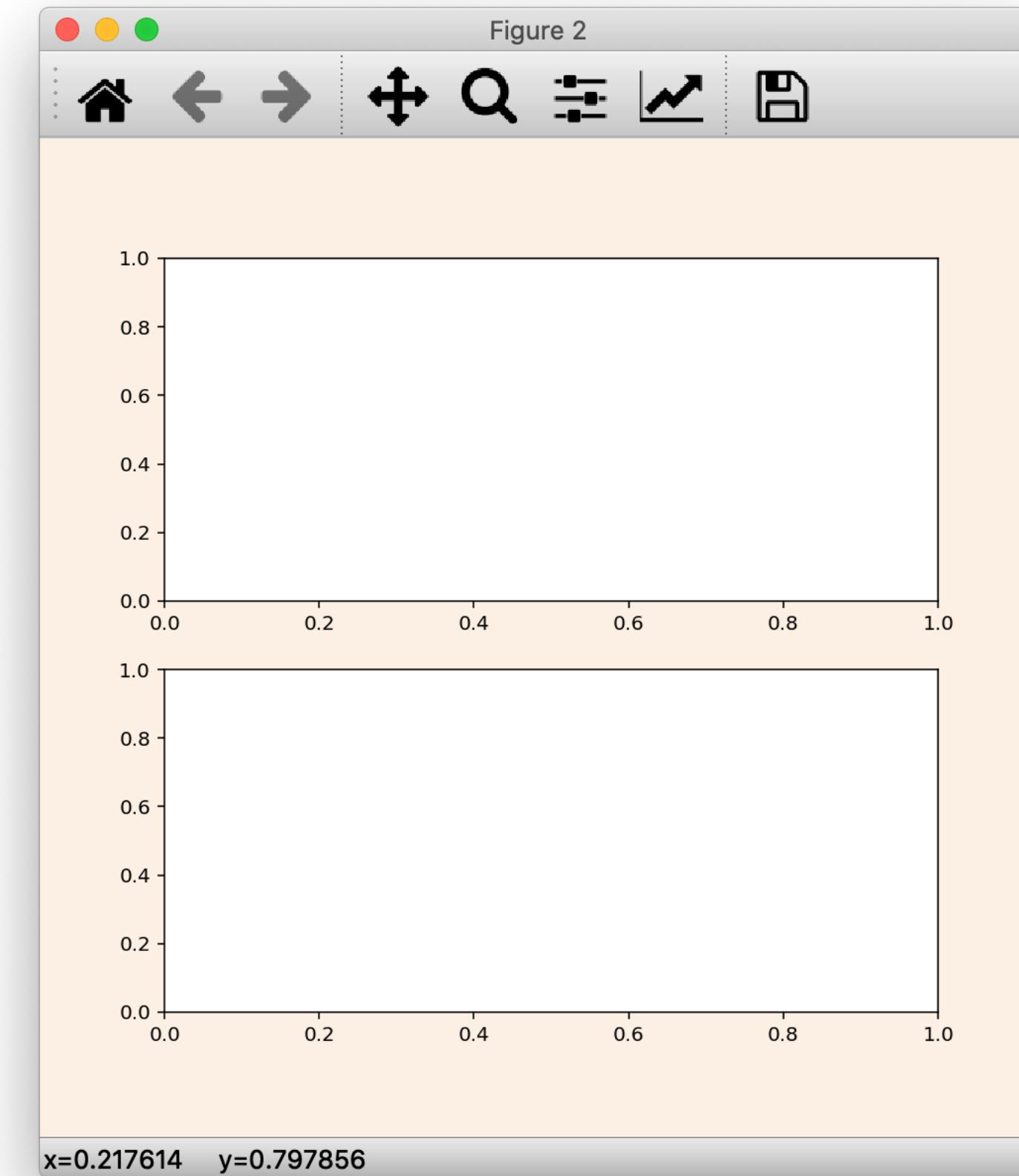
5. plt.subplot2grid(Comparison with Other APIs)

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

fig = plt.figure(figsize=(7, 7),
                 facecolor='linen')
ax1 = fig.add_subplot(211)
ax2 = fig.add_subplot(212)

-----
fig, axes = plt.subplots(2, 1,
                        figsize=(7, 7),
                        facecolor='linen')

-----
fig = plt.figure(figsize=(7, 7),
                 facecolor='linen')
ax1 = plt.subplot2grid((2, 1), (0, 0), fig=fig)
ax2 = plt.subplot2grid((2, 1), (1, 0), fig=fig)
```



5. plt.subplot2grid(Axes Arrangement)

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

fig = plt.figure(figsize=(7, 7),
                 facecolor='linen')

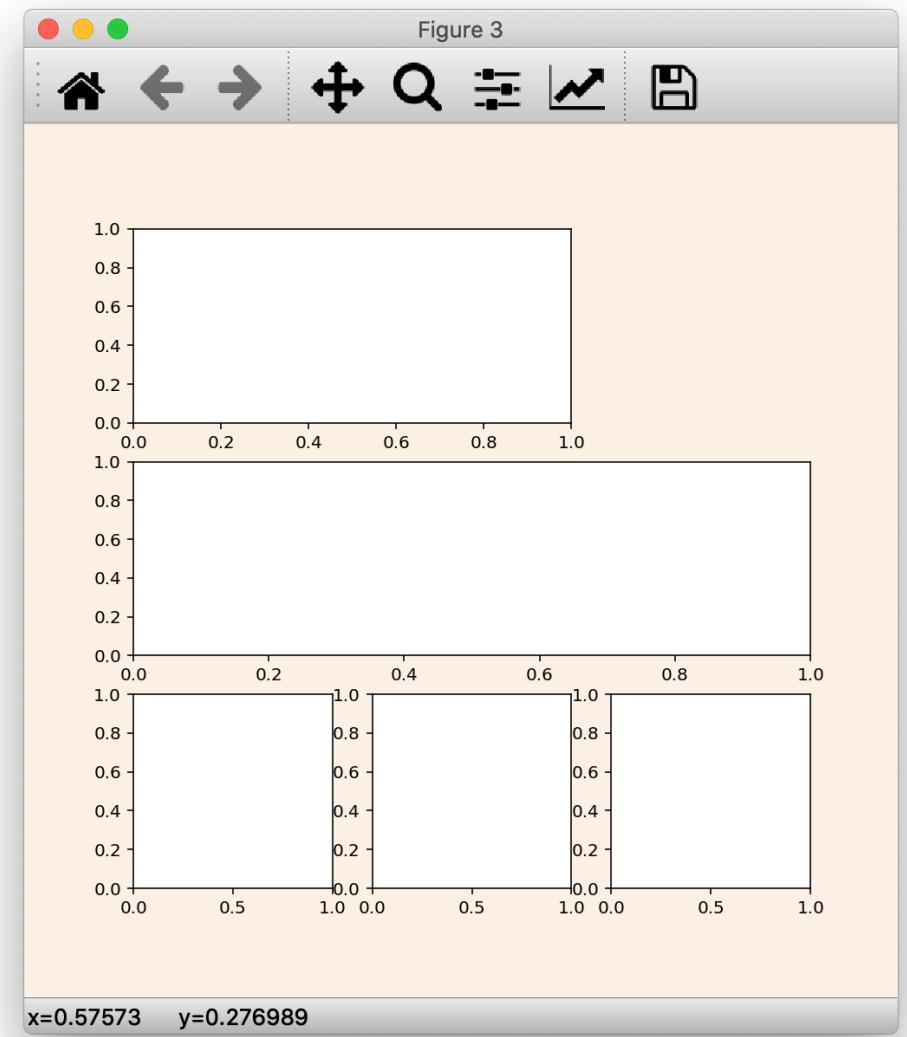
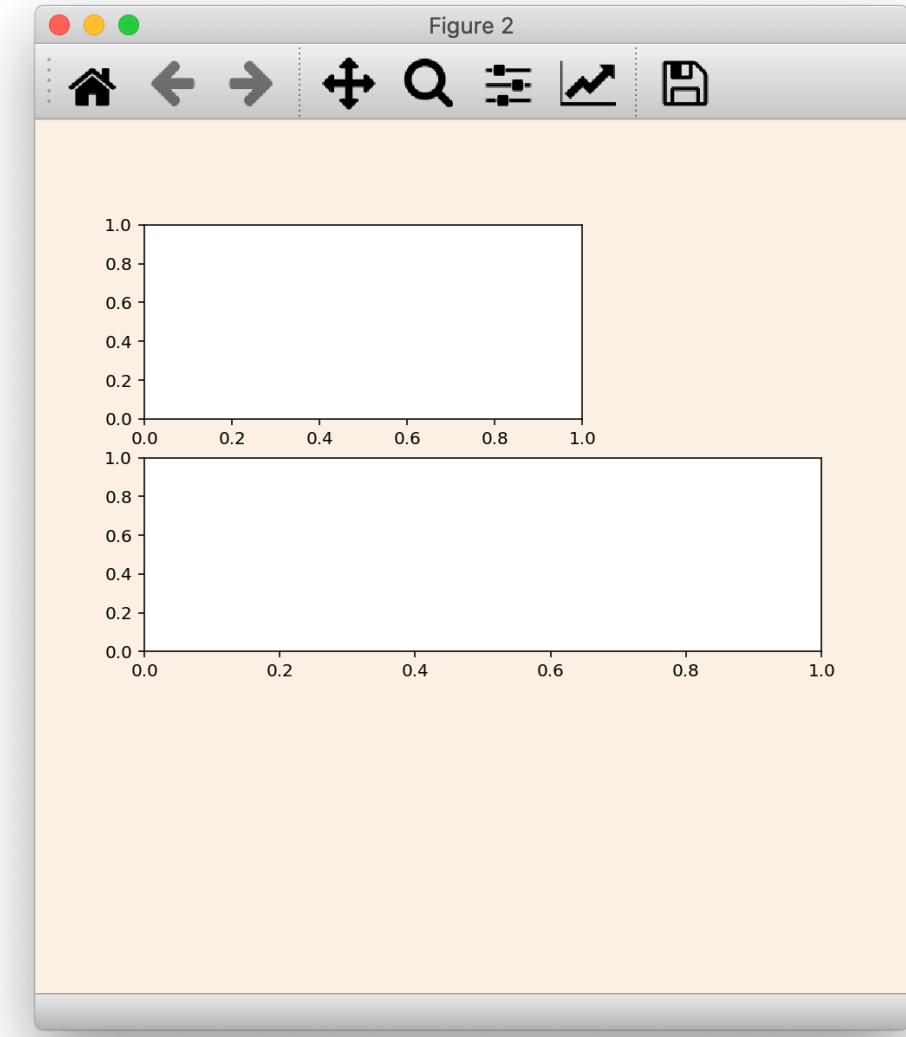
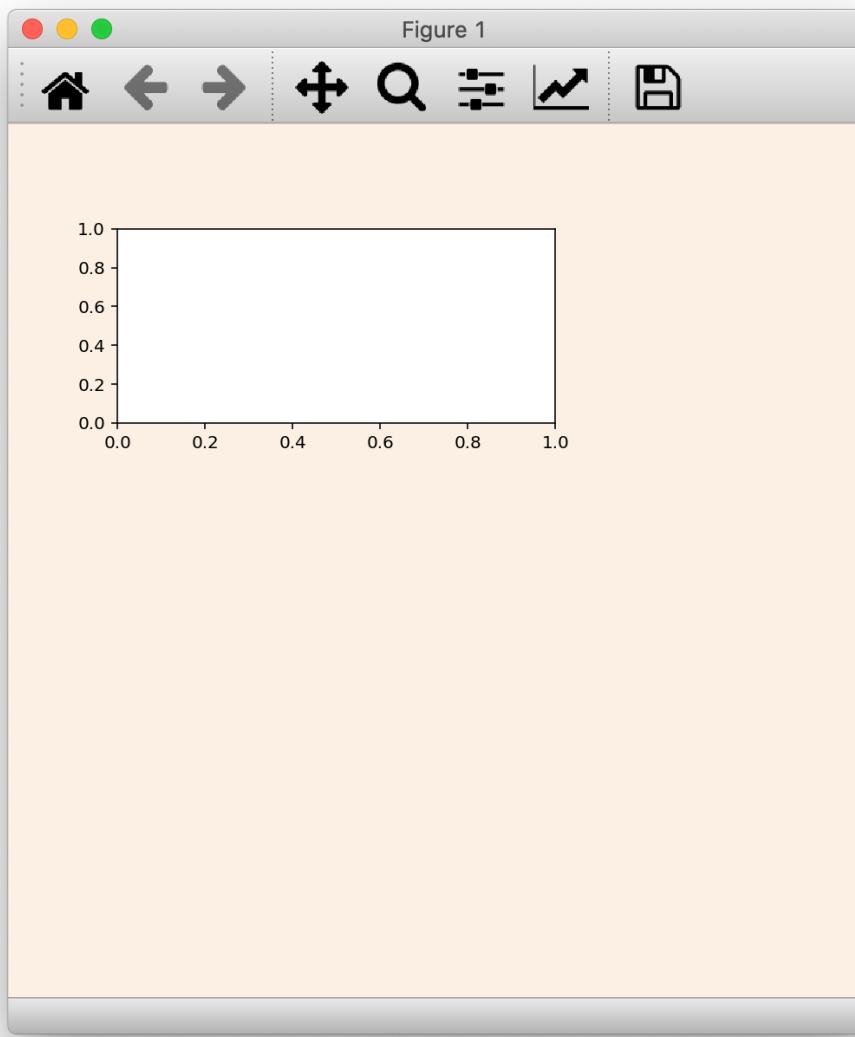
ax1 = plt.subplot2grid((3, 3), (0, 0),
                      colspan=2,
                      fig=fig)

ax2 = plt.subplot2grid((3, 3), (1, 0),
                      colspan=3,
                      fig=fig)

ax3 = plt.subplot2grid((3, 3), (2, 0),
                      fig=fig)

ax4 = plt.subplot2grid((3, 3), (2, 1),
                      fig=fig)

ax5 = plt.subplot2grid((3, 3), (2, 2),
                      fig=fig)
```



5. plt.subplot2grid(Comparison with Other APIs)

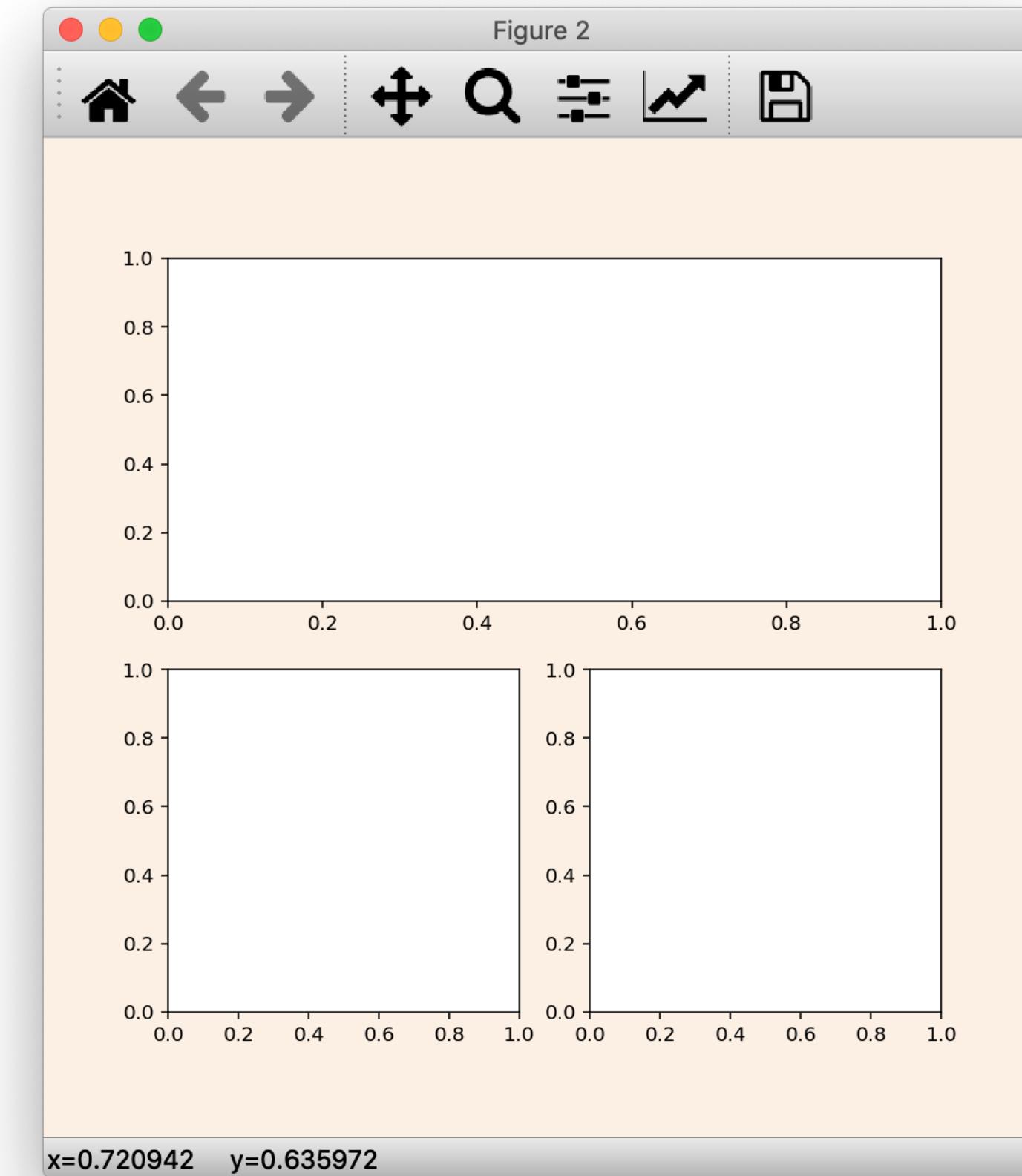
```
import matplotlib.pyplot as plt
import numpy as np

fig = plt.figure(figsize=(7, 7),
                 facecolor='linen')

ax1 = fig.add_subplot(211)
ax2 = fig.add_subplot(223)
ax3 = fig.add_subplot(224)

-----
fig = plt.figure(figsize=(7, 7),
                 facecolor='linen')

ax1 = plt.subplot2grid((2, 2), (0, 0),
                      colspan=2,
                      fig=fig)
ax2 = plt.subplot2grid((2, 2), (1, 0),
                      fig=fig)
ax3 = plt.subplot2grid((2, 2), (1, 1),
                      fig=fig)
```



5. plt.subplot2grid(Comparison with Other APIs)

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

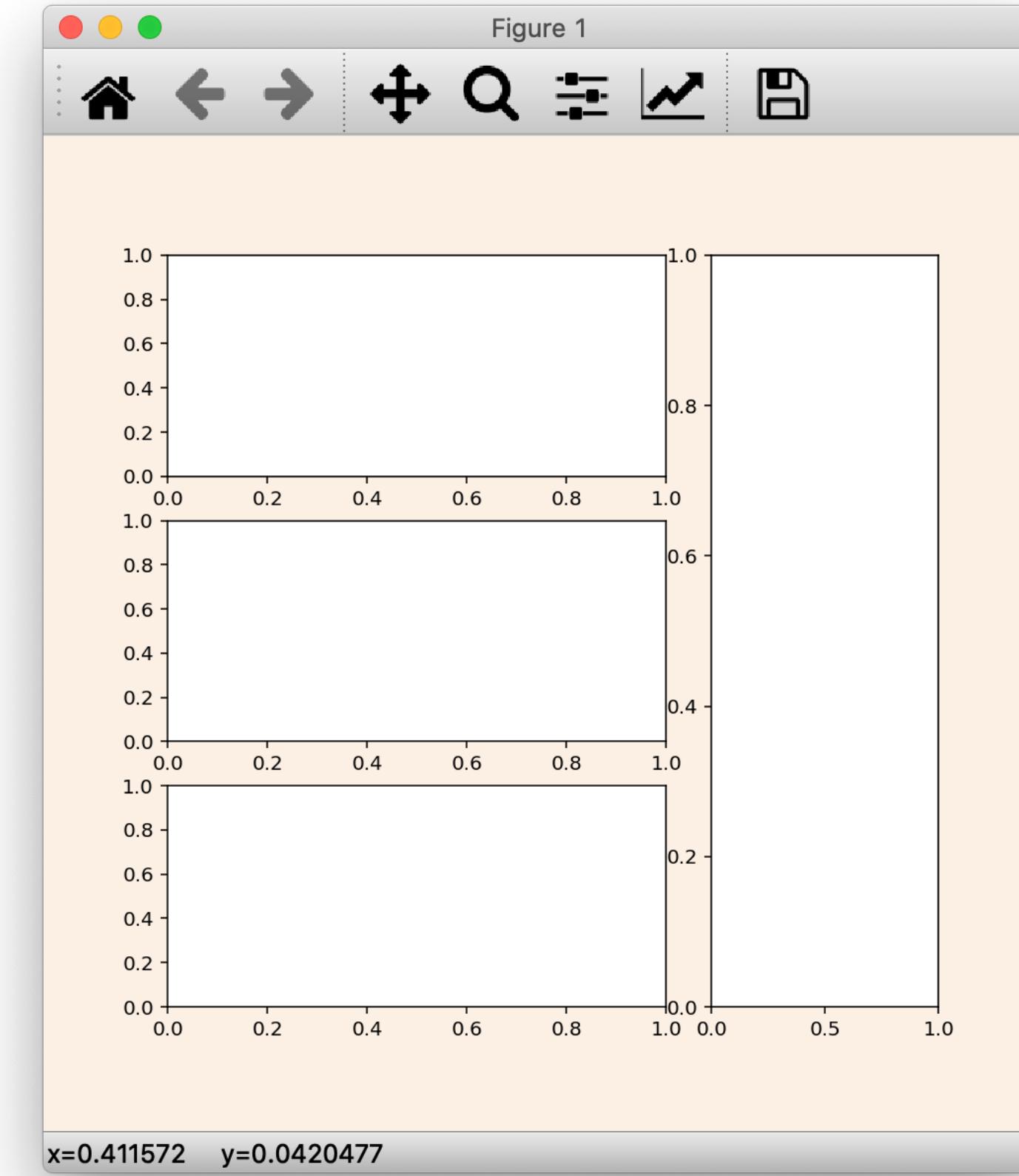
fig = plt.figure(figsize=(7, 7),
                 facecolor='linen')

ax1 = plt.subplot2grid((3, 3), (0, 0),
                      colspan=2,
                      fig=fig)

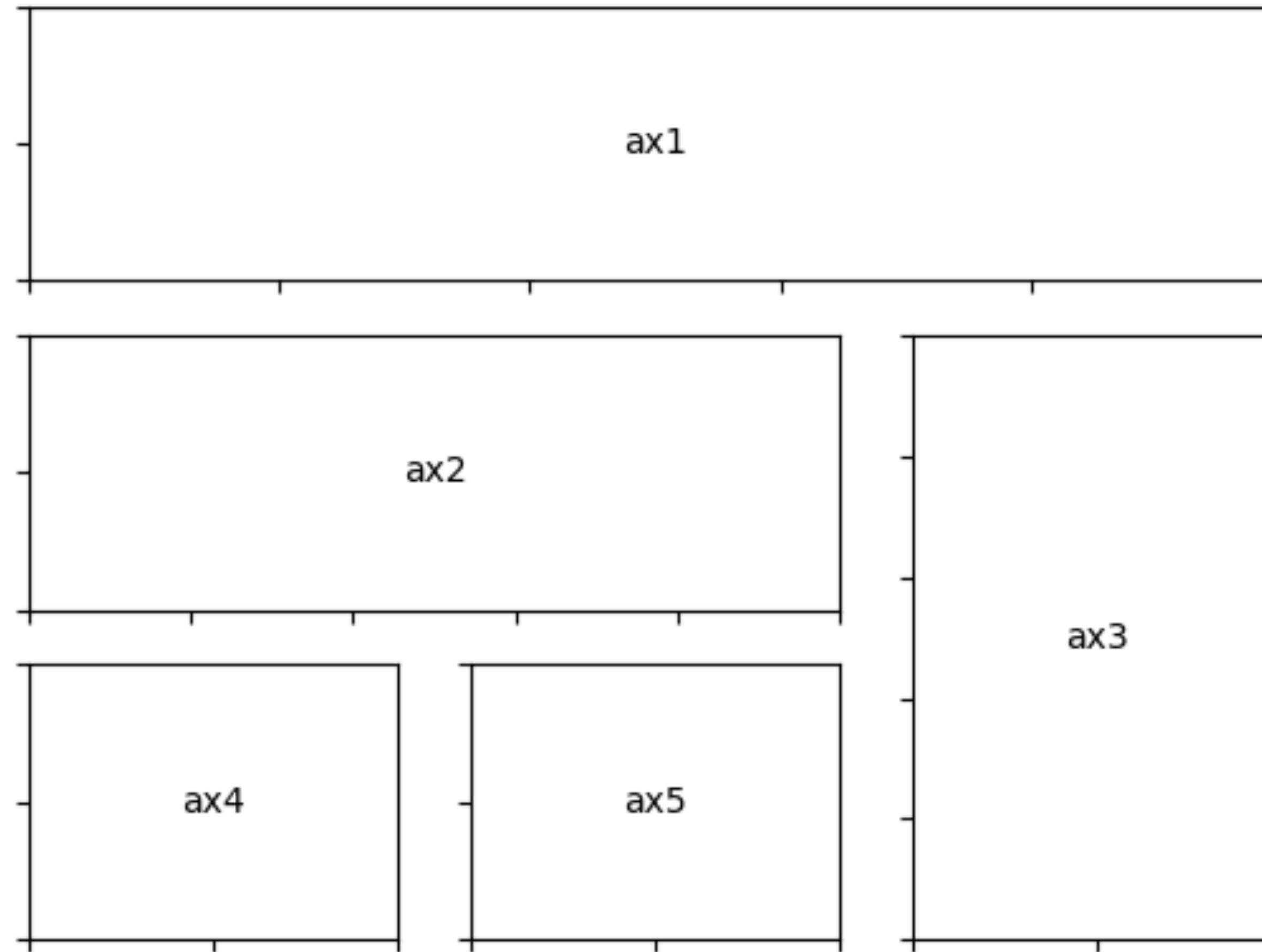
ax2 = plt.subplot2grid((3, 3), (1, 0),
                      colspan=2,
                      fig=fig)

ax3 = plt.subplot2grid((3, 3), (2, 0),
                      colspan=2,
                      fig=fig)

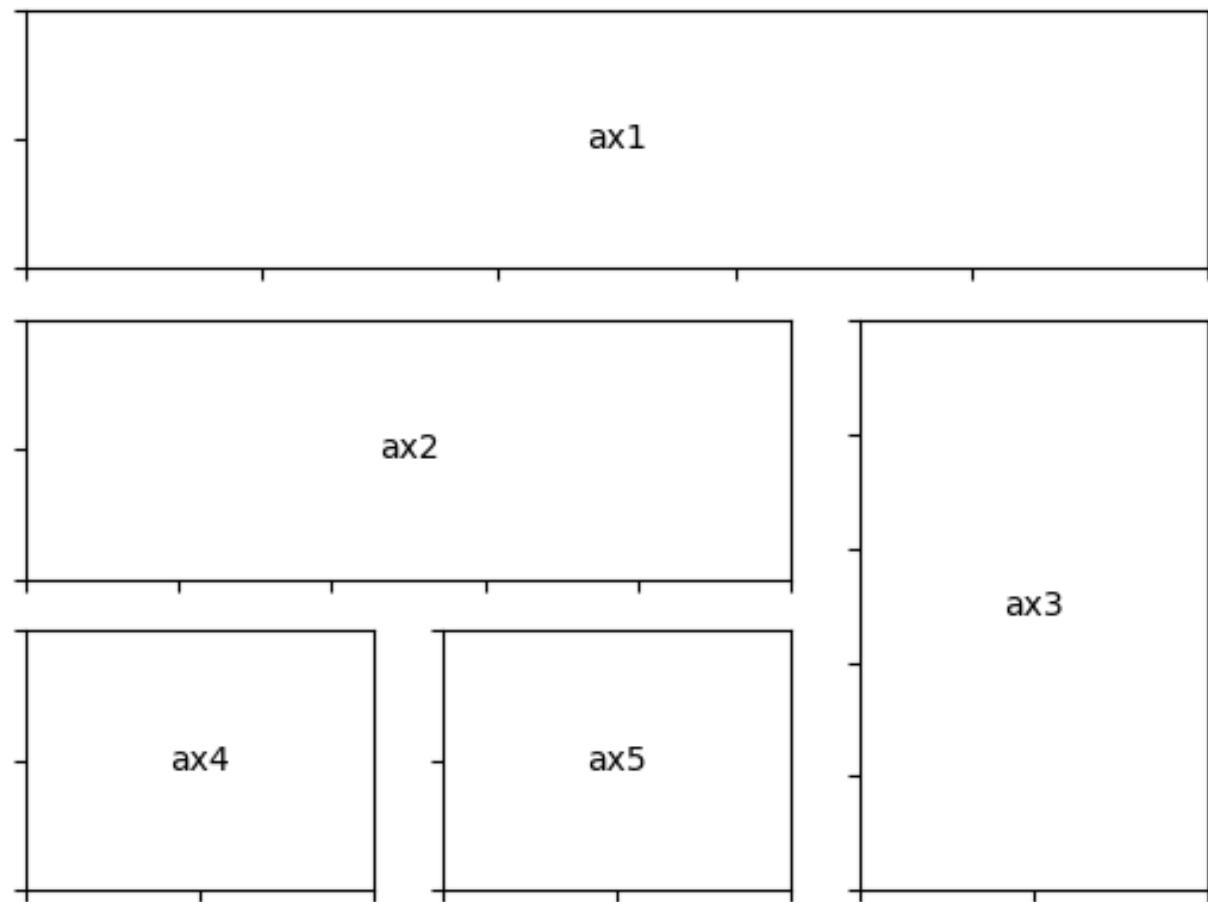
ax4 = plt.subplot2grid((3, 3), (0, 2),
                      rowspan=3,
                      fig=fig)
```



5. plt.subplot2grid(Example)



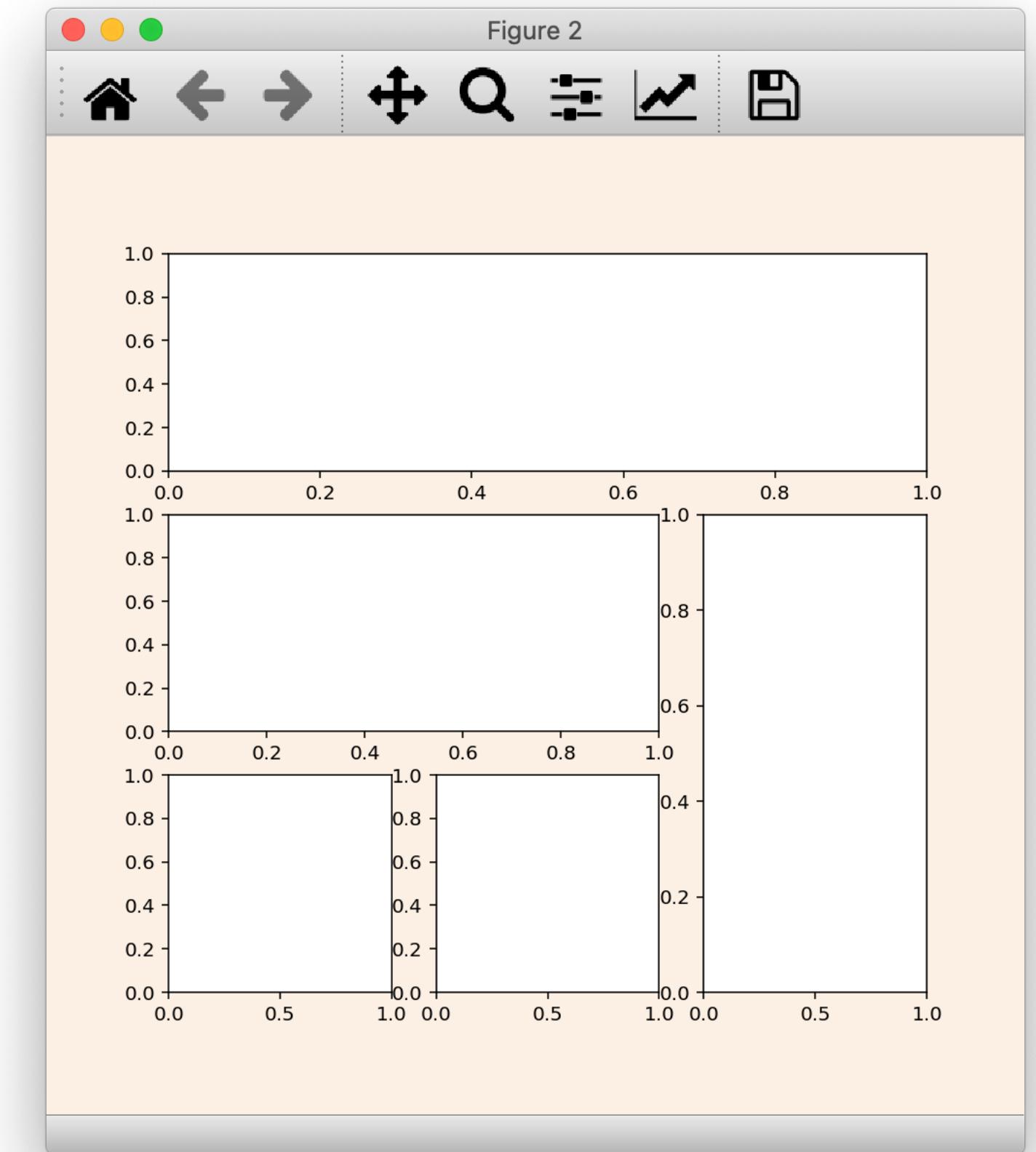
5. plt.subplot2grid(Example)



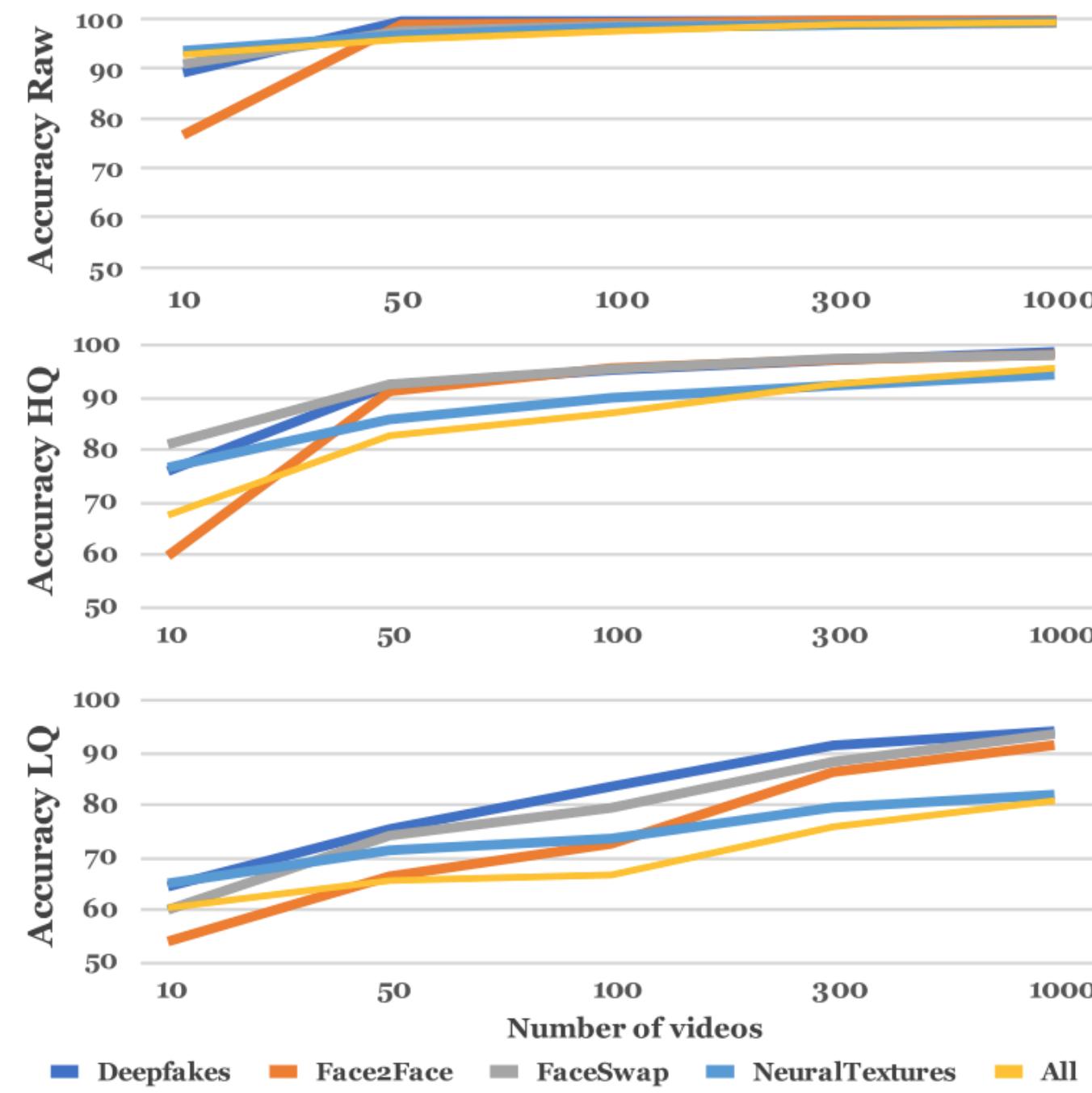
```
import matplotlib.pyplot as plt
import numpy as np

fig = plt.figure(figsize = (7,7),
                  facecolor = 'linen')

ax1 = plt.subplot2grid((3, 3), (0, 0),
                      colspan=3,
                      fig = fig)
ax2 = plt.subplot2grid((3, 3), (1, 0),
                      colspan=2,
                      fig = fig)
ax3 = plt.subplot2grid((3, 3), (1, 2),
                      rowspan=2,
                      fig = fig)
ax4 = plt.subplot2grid((3, 3), (2, 0),
                      fig = fig)
ax5 = plt.subplot2grid((3, 3), (2, 1),
                      fig = fig)
```



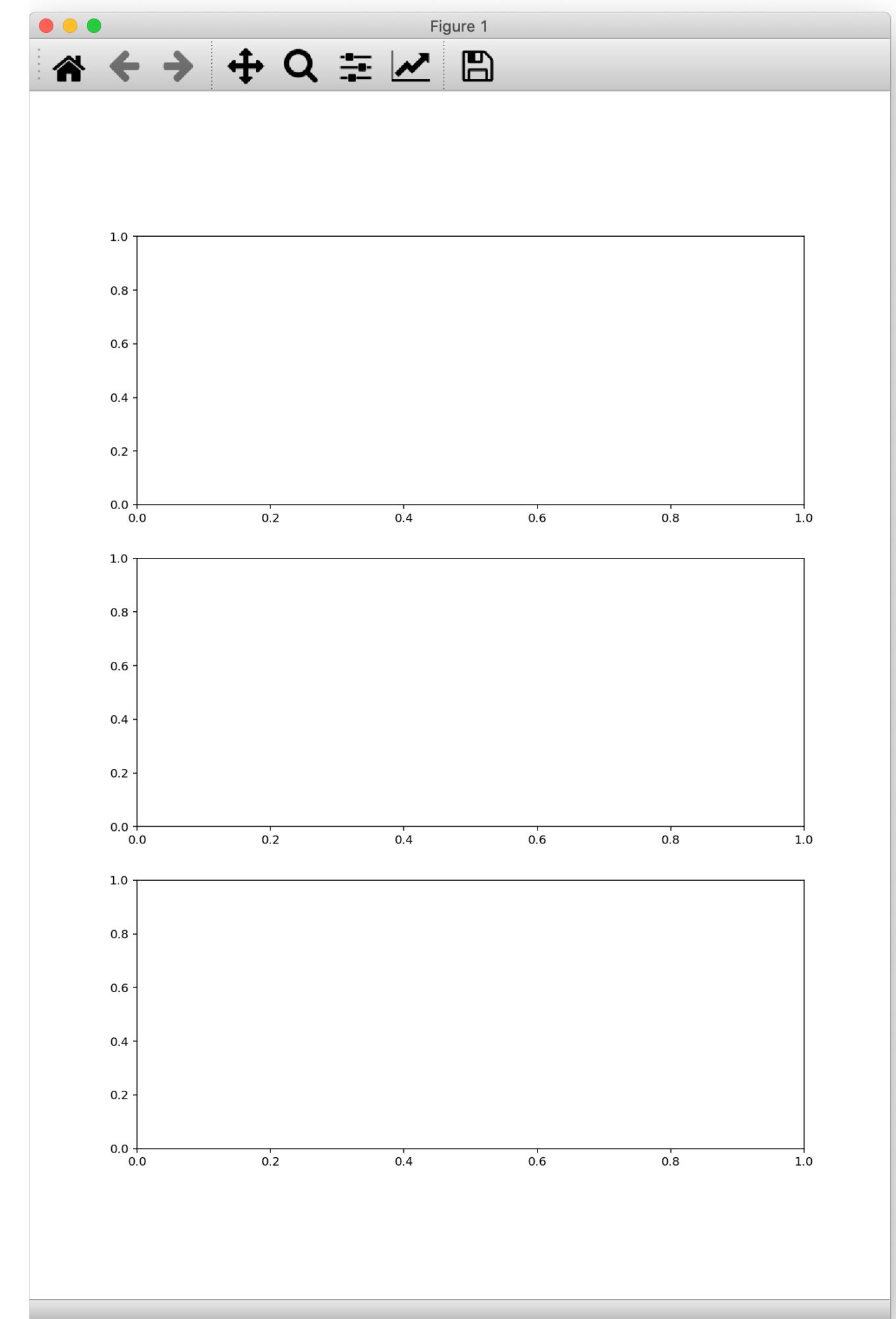
6. Practice.1



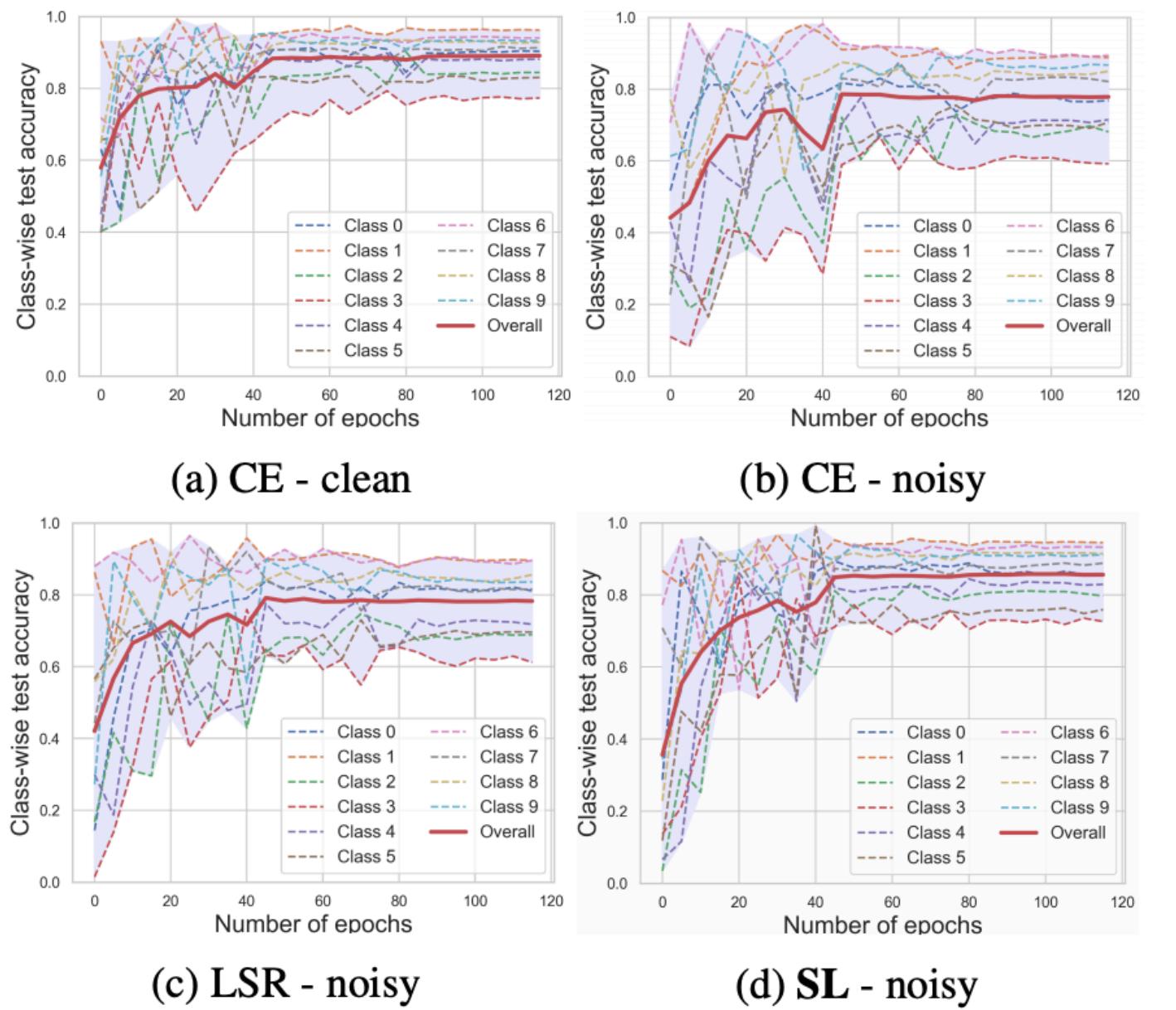
```
fig = plt.figure(figsize=(10, 20))
ax1 = fig.add_subplot(311)
ax2 = fig.add_subplot(312)
ax3 = fig.add_subplot(313)
```

```
fig, axes = plt.subplots(3, 1, figsize=(10, 20))
```

```
fig = plt.figure(figsize=(10, 20))
ax1 = plt.subplot2grid((3, 1), (0, 0), fig=fig)
ax2 = plt.subplot2grid((3, 1), (1, 0), fig=fig)
ax3 = plt.subplot2grid((3, 1), (2, 0), fig=fig)
```



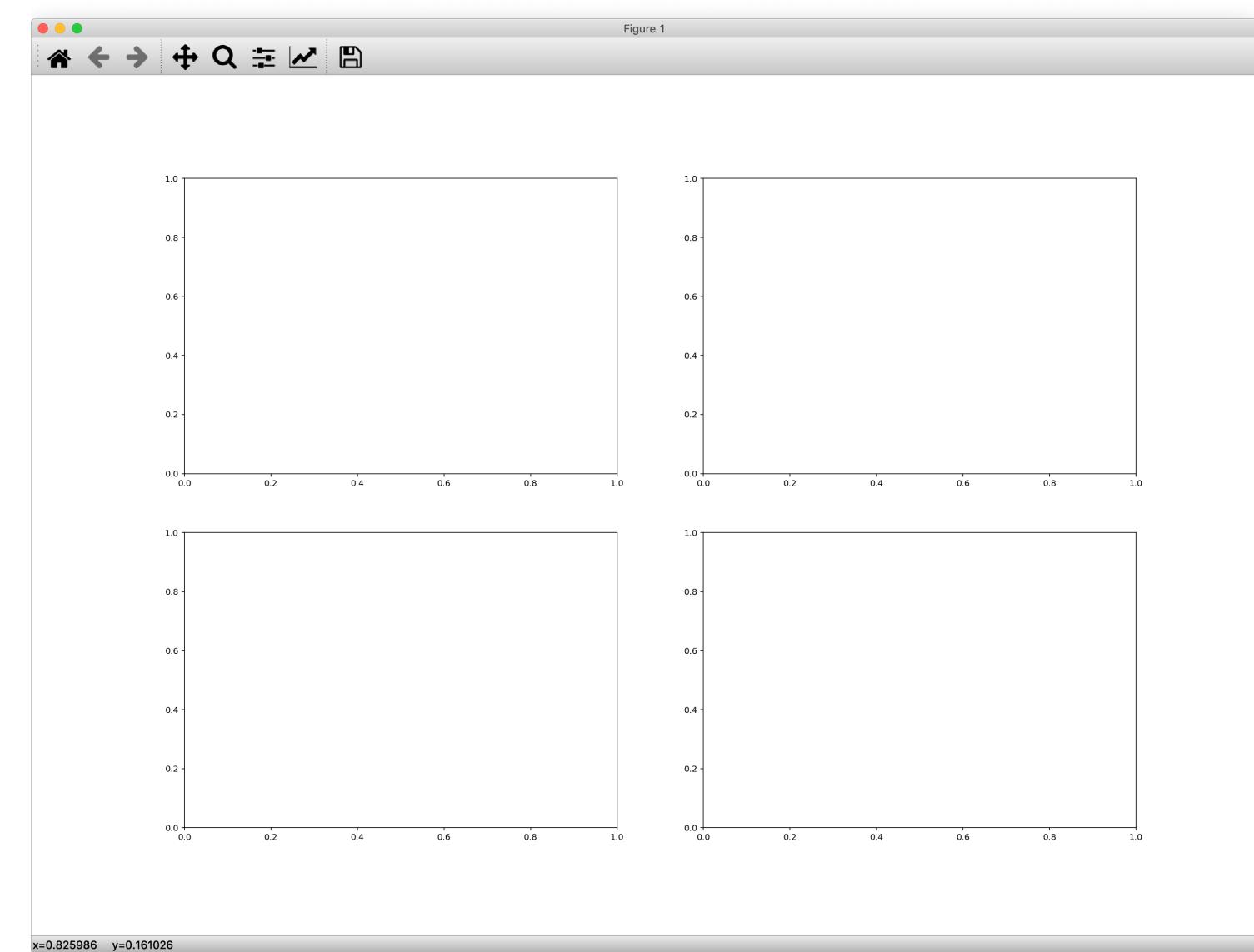
6. Practice.2



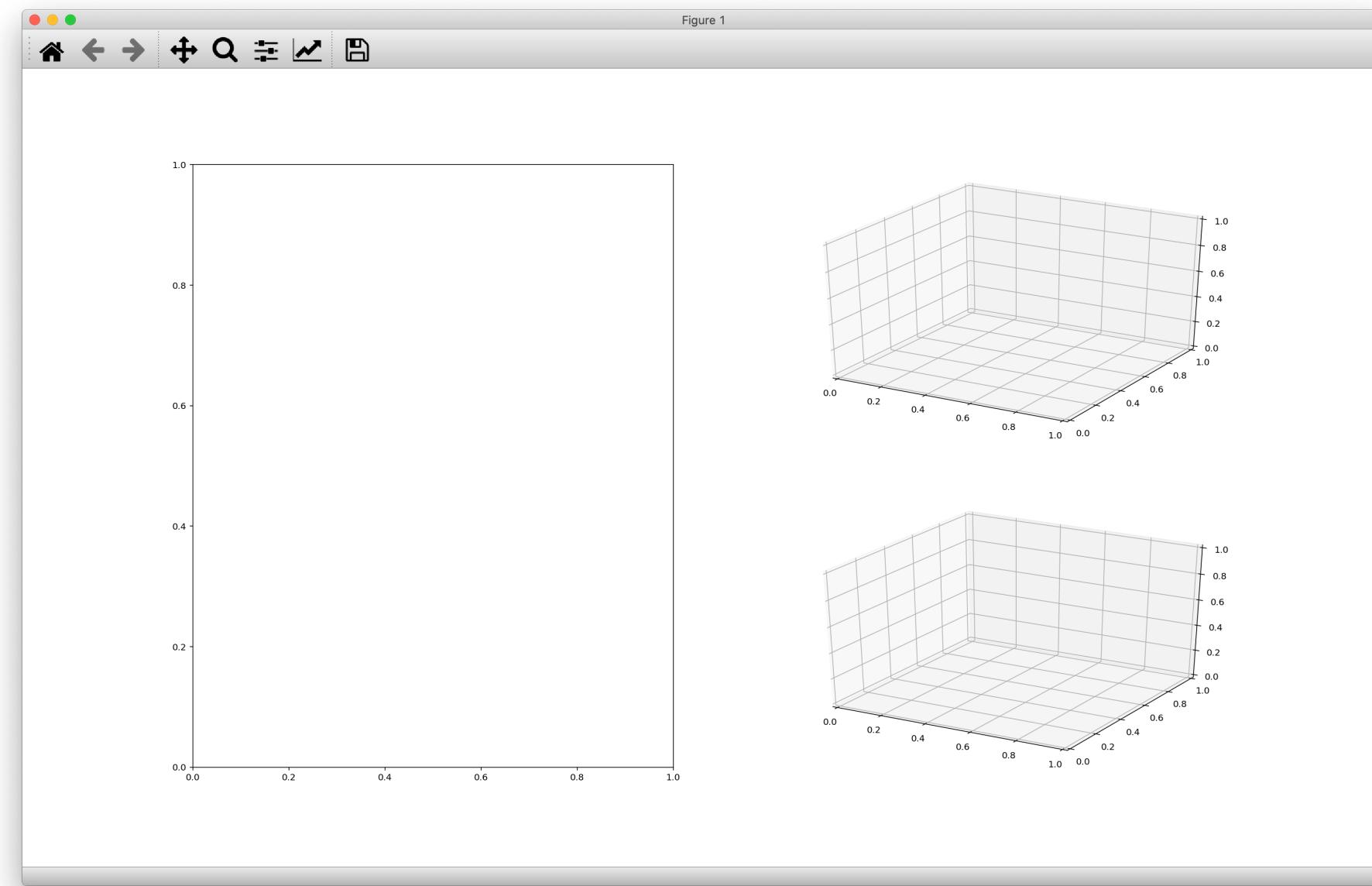
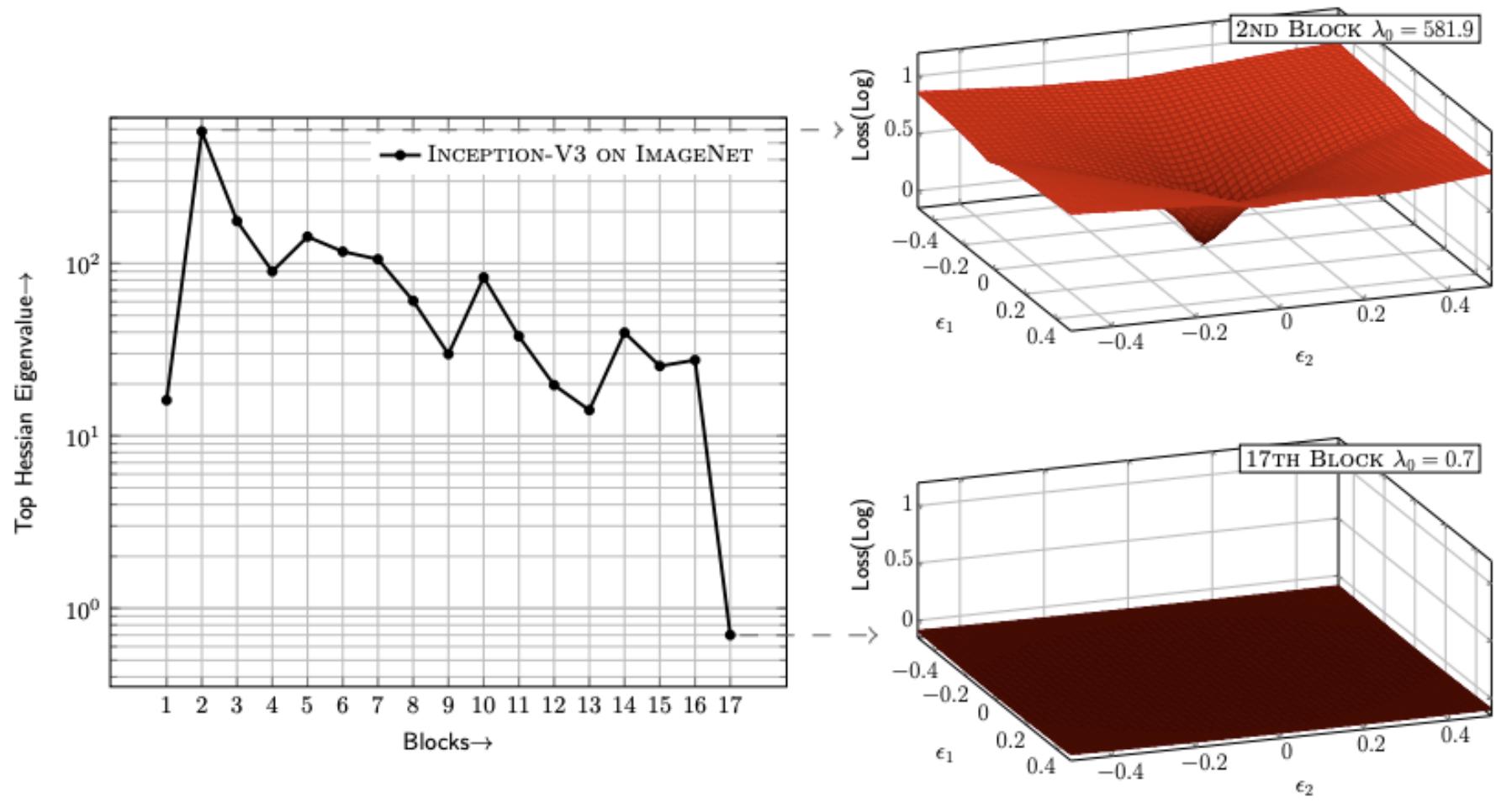
```
fig = plt.figure(figsize=(20, 20))
ax1_1 = fig.add_subplot(221)
ax1_2 = fig.add_subplot(222)
ax2_1 = fig.add_subplot(223)
ax2_2 = fig.add_subplot(224)
```

```
fig, axes = plt.subplots(2, 2, figsize=(20, 20))
```

```
fig = plt.figure(figsize=(20, 20))
ax1_1 = plt.subplot2grid((2, 2), (0, 0), fig=fig)
ax1_2 = plt.subplot2grid((2, 2), (0, 1), fig=fig)
ax2_1 = plt.subplot2grid((2, 2), (1, 0), fig=fig)
ax2_2 = plt.subplot2grid((2, 2), (1, 1), fig=fig)
```



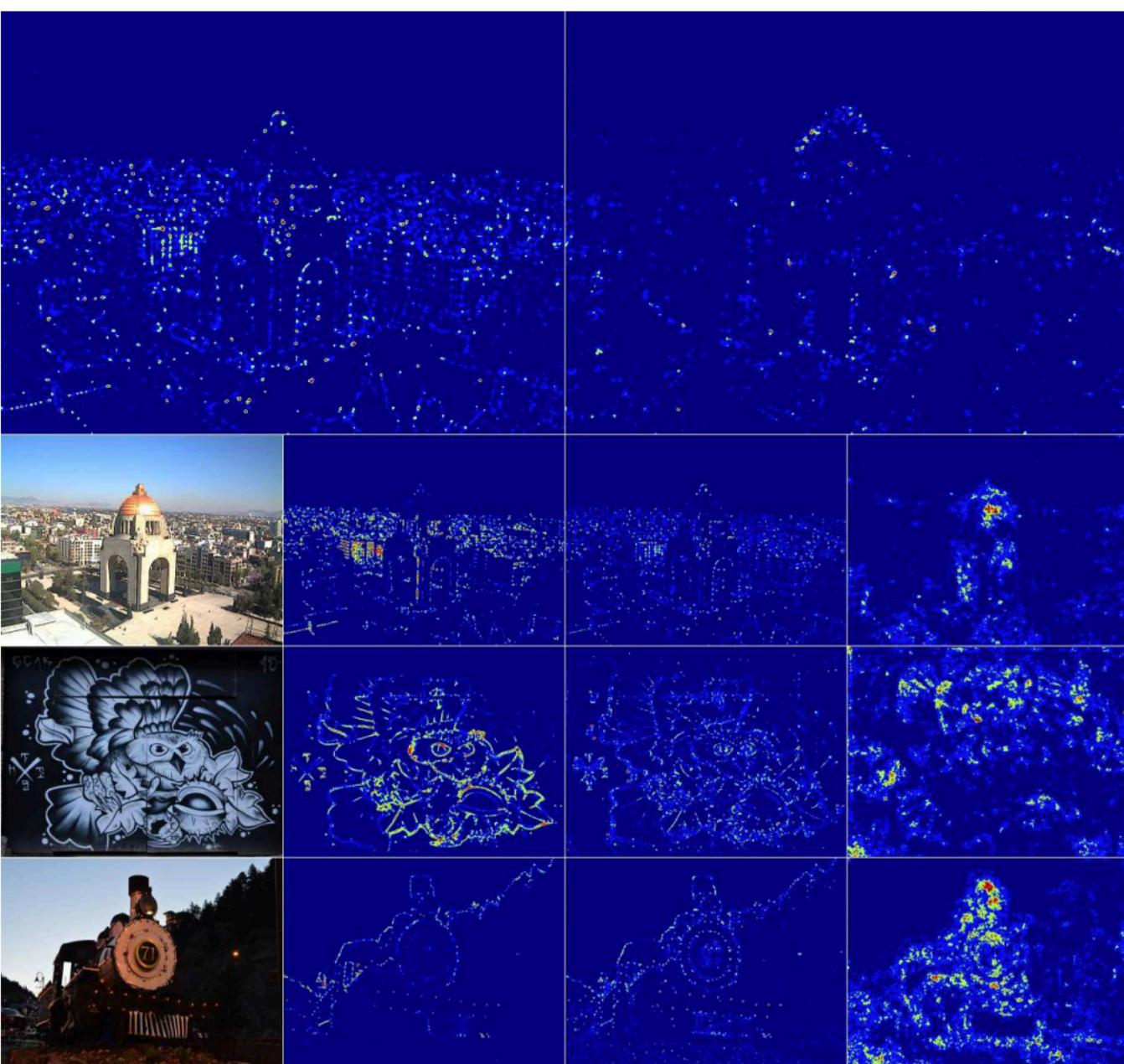
6. Practice.3



```
fig = plt.figure(figsize=(25, 20))
ax1 = fig.add_subplot(121)
ax2 = fig.add_subplot(222,
                     projection='3d')
ax3 = fig.add_subplot(224,
                     projection='3d')
```

```
fig = plt.figure(figsize=(25, 20))
ax1 = plt.subplot2grid((2, 2), (0, 0),
                      rowspan=2,
                      fig=fig)
ax2 = plt.subplot2grid((2, 2), (0, 1),
                      projection='3d'
                      fig=fig))
ax3 = plt.subplot2grid((2, 2), (1, 1),
                      projection='3d'
                      fig=fig))
```

6. Practice.4



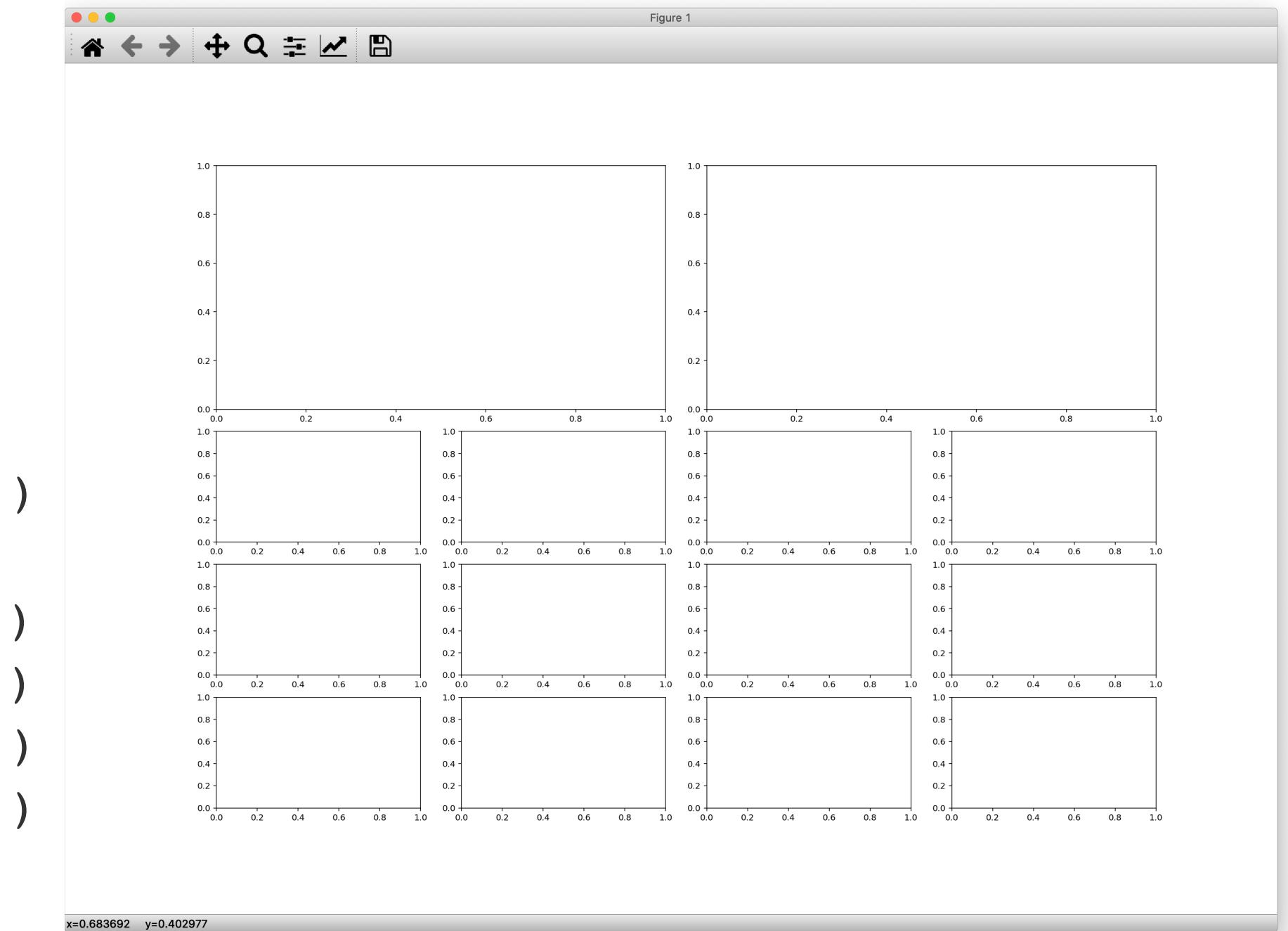
```
fig = plt.figure(figsize=(20, 20))

ax1 = plt.subplot2grid((5, 4), (0, 0),
                      rowspan=2, colspan=2)
ax2 = plt.subplot2grid((5, 4), (0, 2),
                      rowspan=2, colspan=2)

ax3 = plt.subplot2grid((5, 4), (2, 0))
ax4 = plt.subplot2grid((5, 4), (2, 1))
ax5 = plt.subplot2grid((5, 4), (2, 2))
ax6 = plt.subplot2grid((5, 4), (2, 3))

ax7 = plt.subplot2grid((5, 4), (3, 0))
ax8 = plt.subplot2grid((5, 4), (3, 1))
ax9 = plt.subplot2grid((5, 4), (3, 2))
ax10 = plt.subplot2grid((5, 4), (3, 3))

ax11 = plt.subplot2grid((5, 4), (4, 0))
ax12 = plt.subplot2grid((5, 4), (4, 1))
ax13 = plt.subplot2grid((5, 4), (4, 2))
ax14 = plt.subplot2grid((5, 4), (4, 3))
```



6. Practice.4

```
fig = plt.figure(figsize=(20, 20))

ax1 = plt.subplot2grid((5, 4), (0, 0),
                      rowspan=2, colspan=2)
ax2 = plt.subplot2grid((5, 4), (0, 2),
                      rowspan=2, colspan=2)

ax3 = plt.subplot2grid((5, 4), (2, 0))
ax4 = plt.subplot2grid((5, 4), (2, 1))
ax5 = plt.subplot2grid((5, 4), (2, 2))
ax6 = plt.subplot2grid((5, 4), (2, 3))

ax7 = plt.subplot2grid((5, 4), (3, 0))
ax8 = plt.subplot2grid((5, 4), (3, 1))
ax9 = plt.subplot2grid((5, 4), (3, 2))
ax10 = plt.subplot2grid((5, 4), (3, 3))

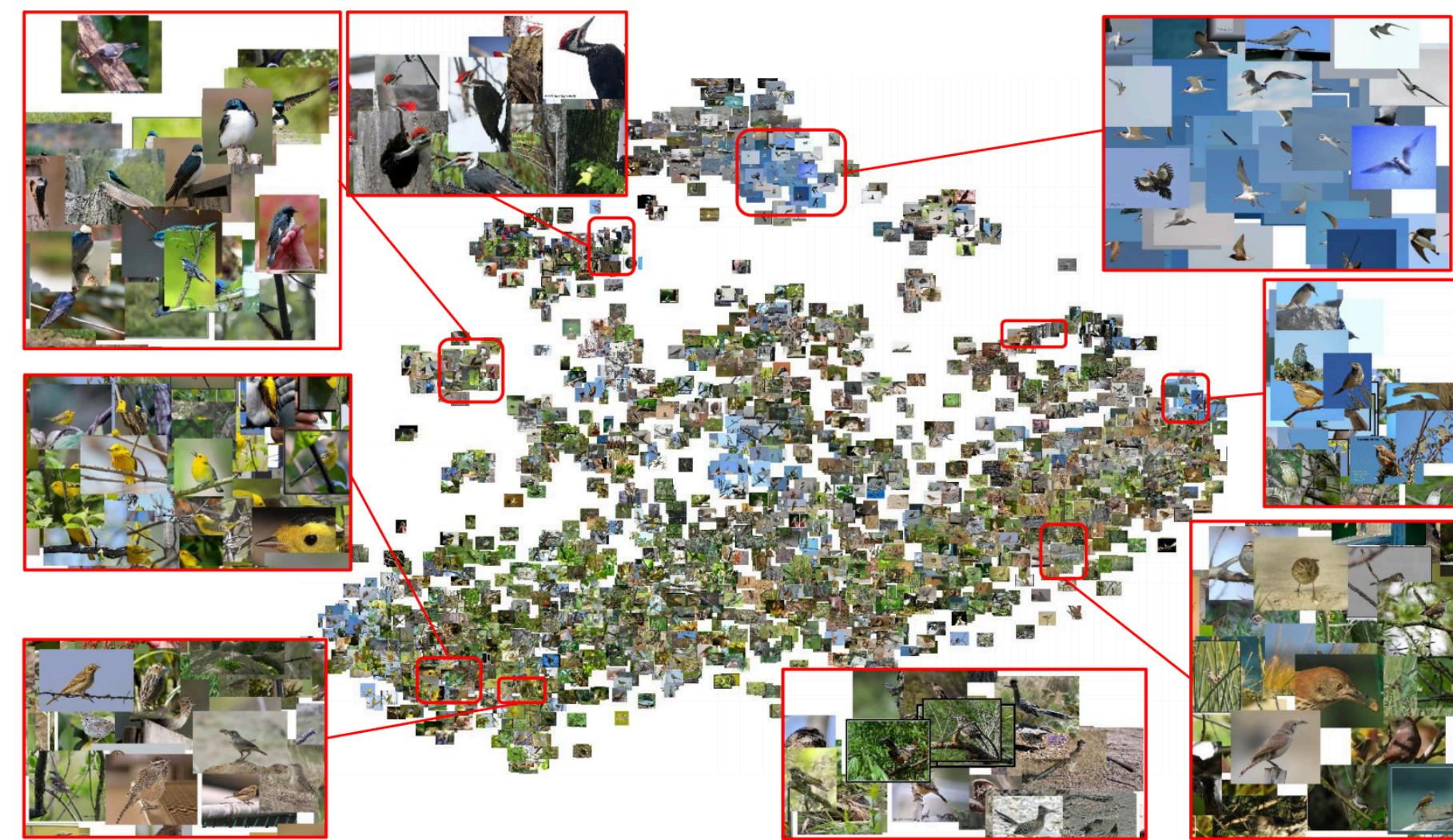
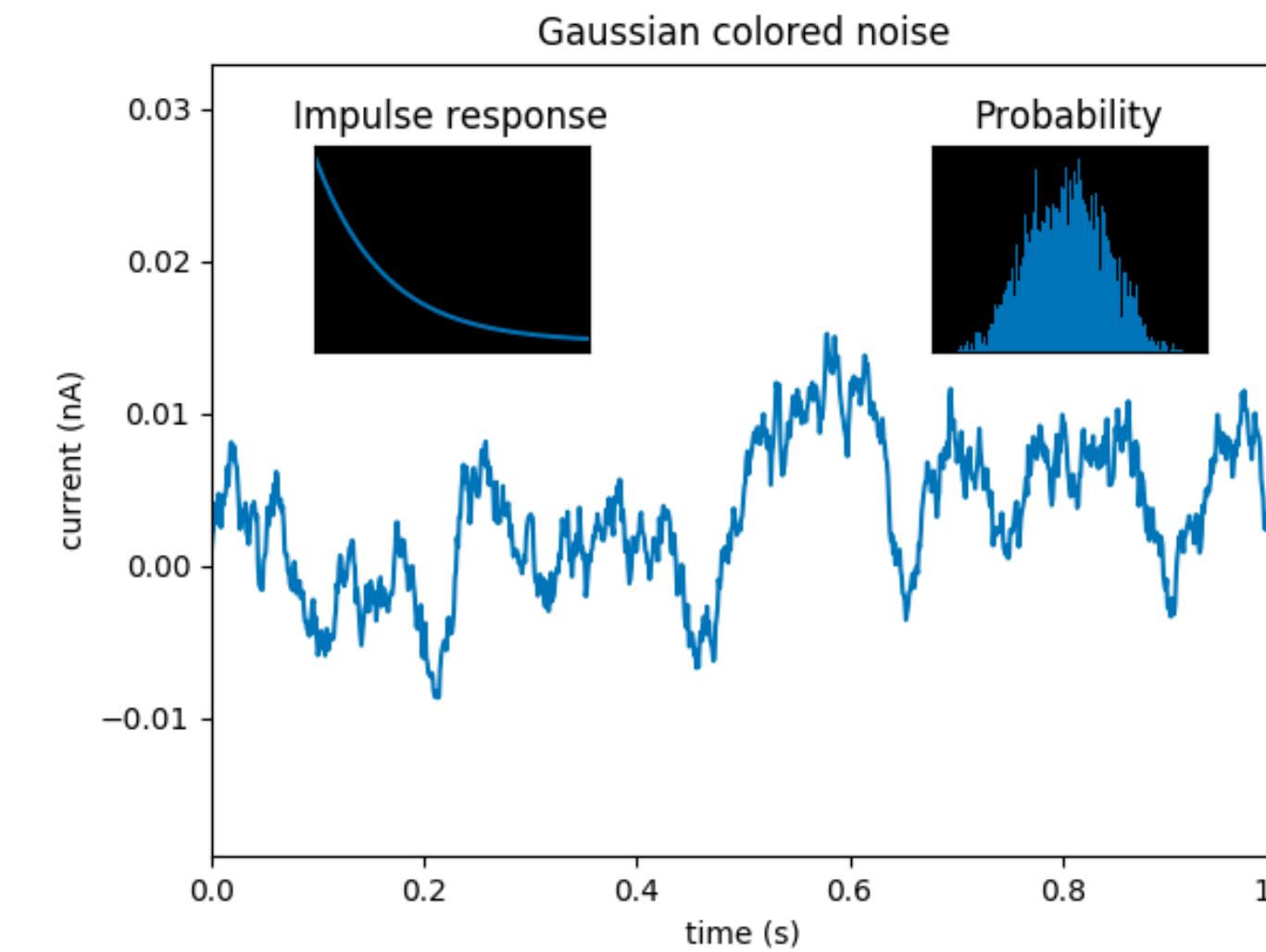
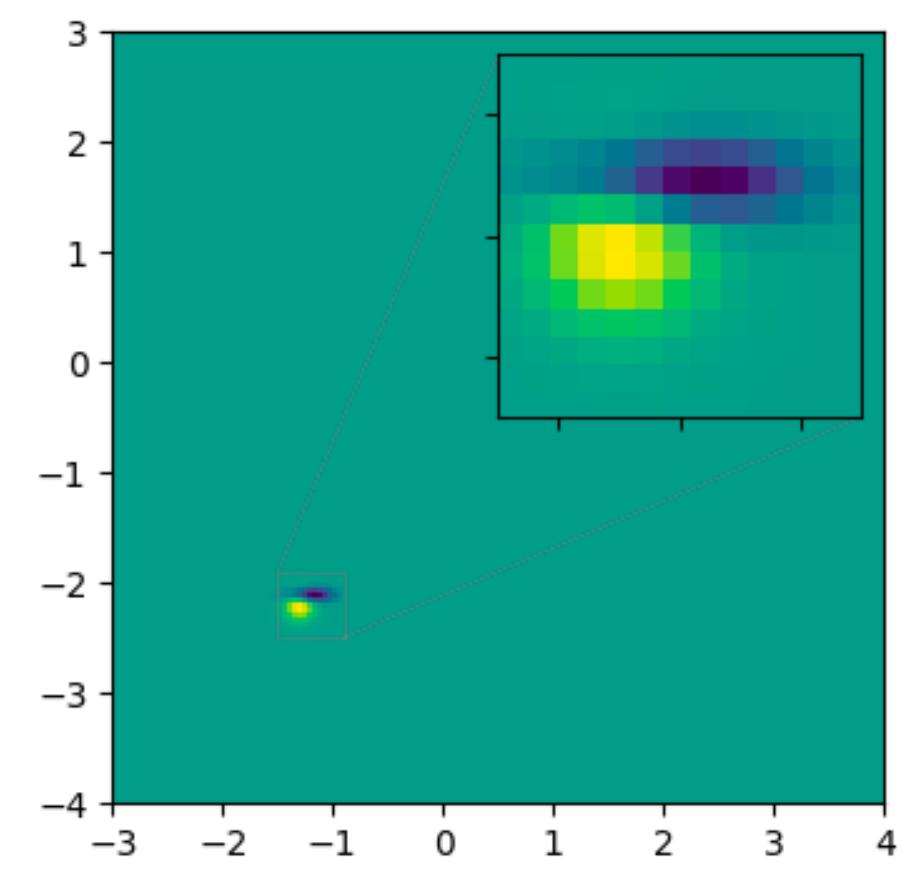
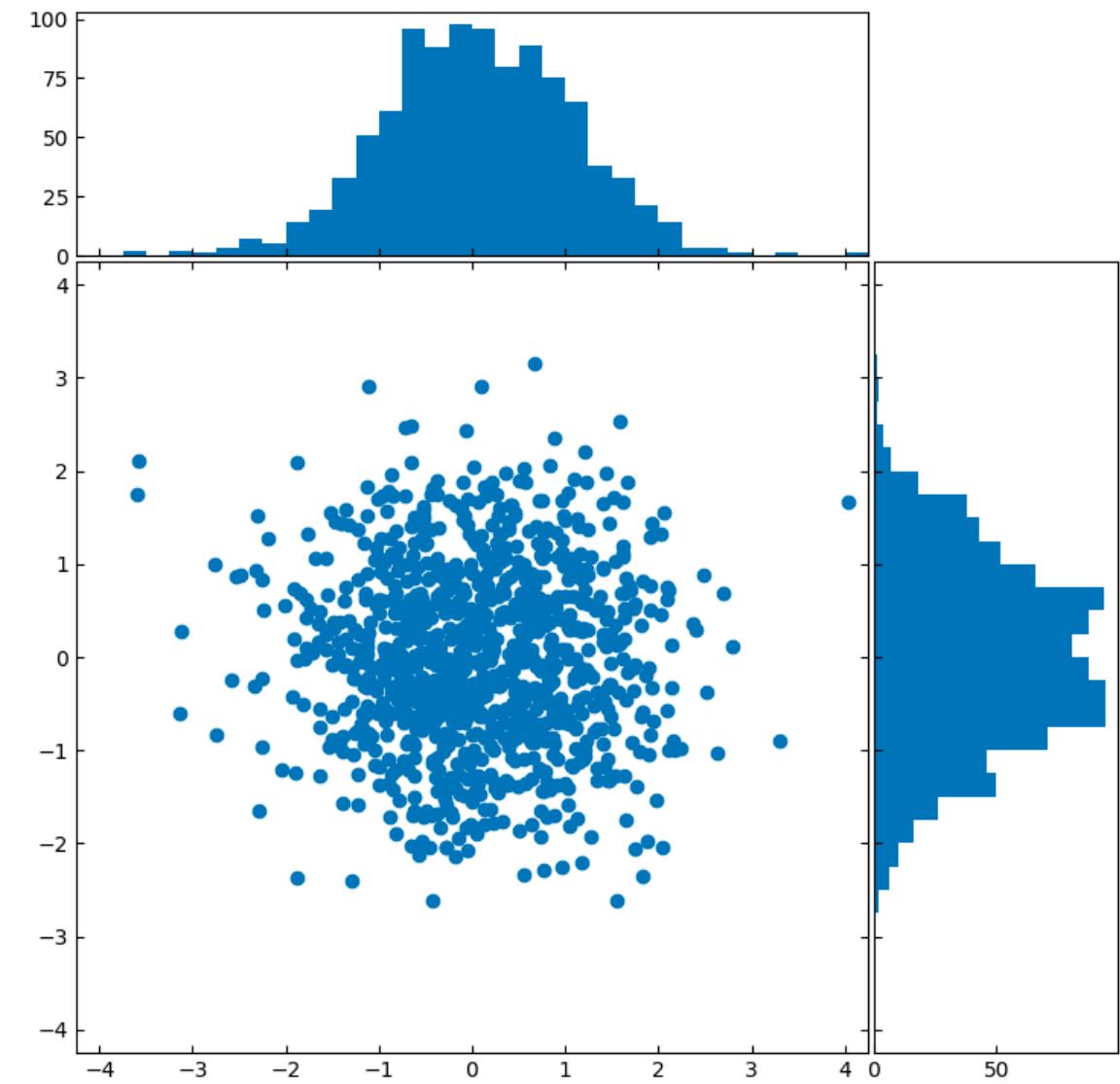
ax11 = plt.subplot2grid((5, 4), (4, 0))
ax12 = plt.subplot2grid((5, 4), (4, 1))
ax13 = plt.subplot2grid((5, 4), (4, 2))
ax14 = plt.subplot2grid((5, 4), (4, 3))
```

```
fig = plt.figure(figsize=(20, 20))

axes = np.empty(shape=(0,))
axes = np.append(axes, plt.subplot2grid((5, 4), (0, 0), rowspan=2, colspan=2))
axes = np.append(axes, plt.subplot2grid((5, 4), (0, 2), rowspan=2, colspan=2))

for r_idx in range(2, 5):
    for c_idx in range(4):
        axes = np.append(axes, plt.subplot2grid((5, 4), (r_idx, c_idx)))
```

7. fig.add_axes(Arbitrary Locations and Sizes of Axes)



7. fig.add_axes(rect Argument)

```
add_axes(self, *args, **kwargs)
```

[\[source\]](#)

Add an axes to the figure.

Call signatures:

```
add_axes(rect, projection=None, polar=False, **kwargs)
add_axes(ax)
```



Parameters:

rect : sequence of float

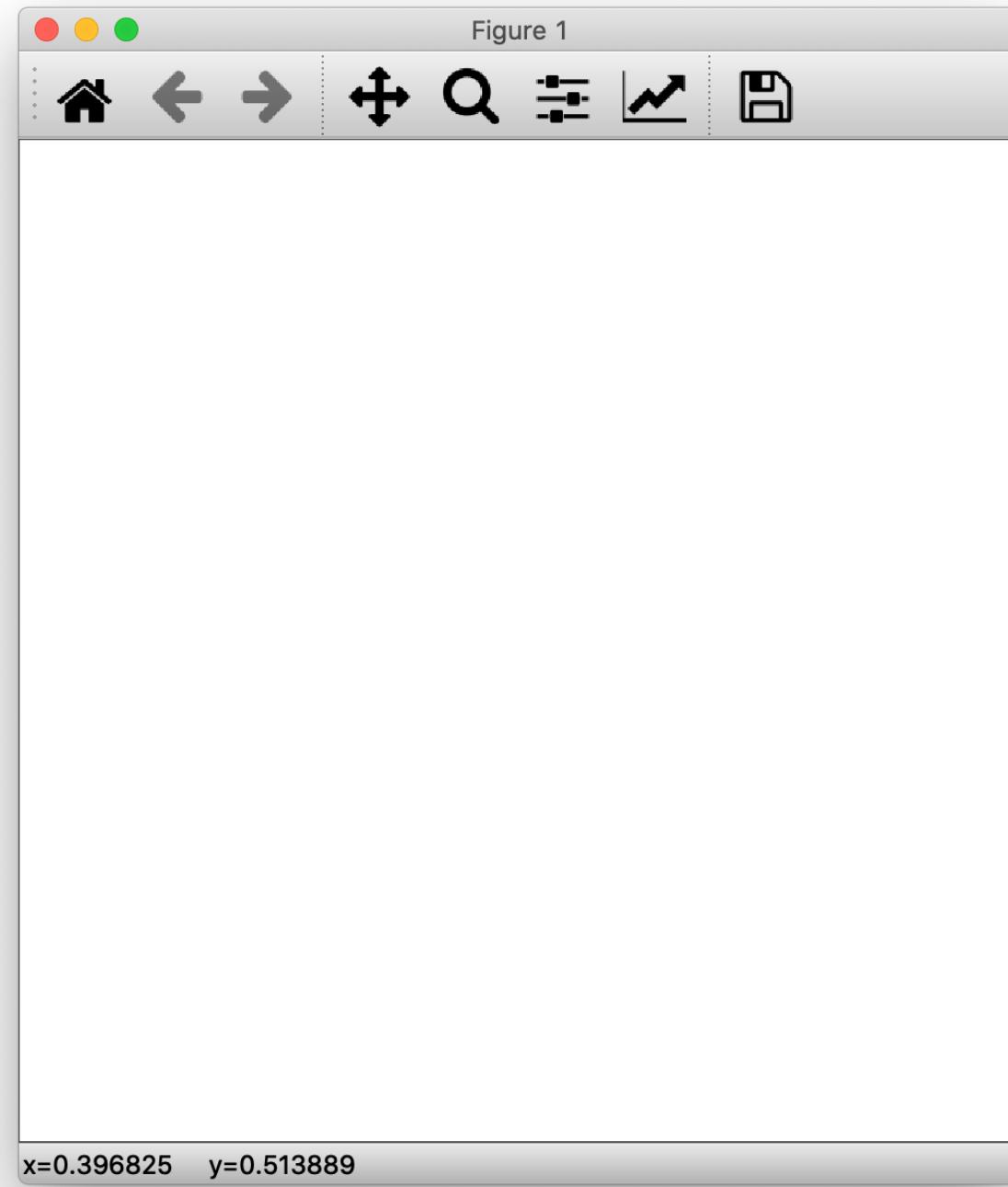
The dimensions [left, bottom, width, height] of the new axes. All quantities are in fractions of figure width and height.

Usage

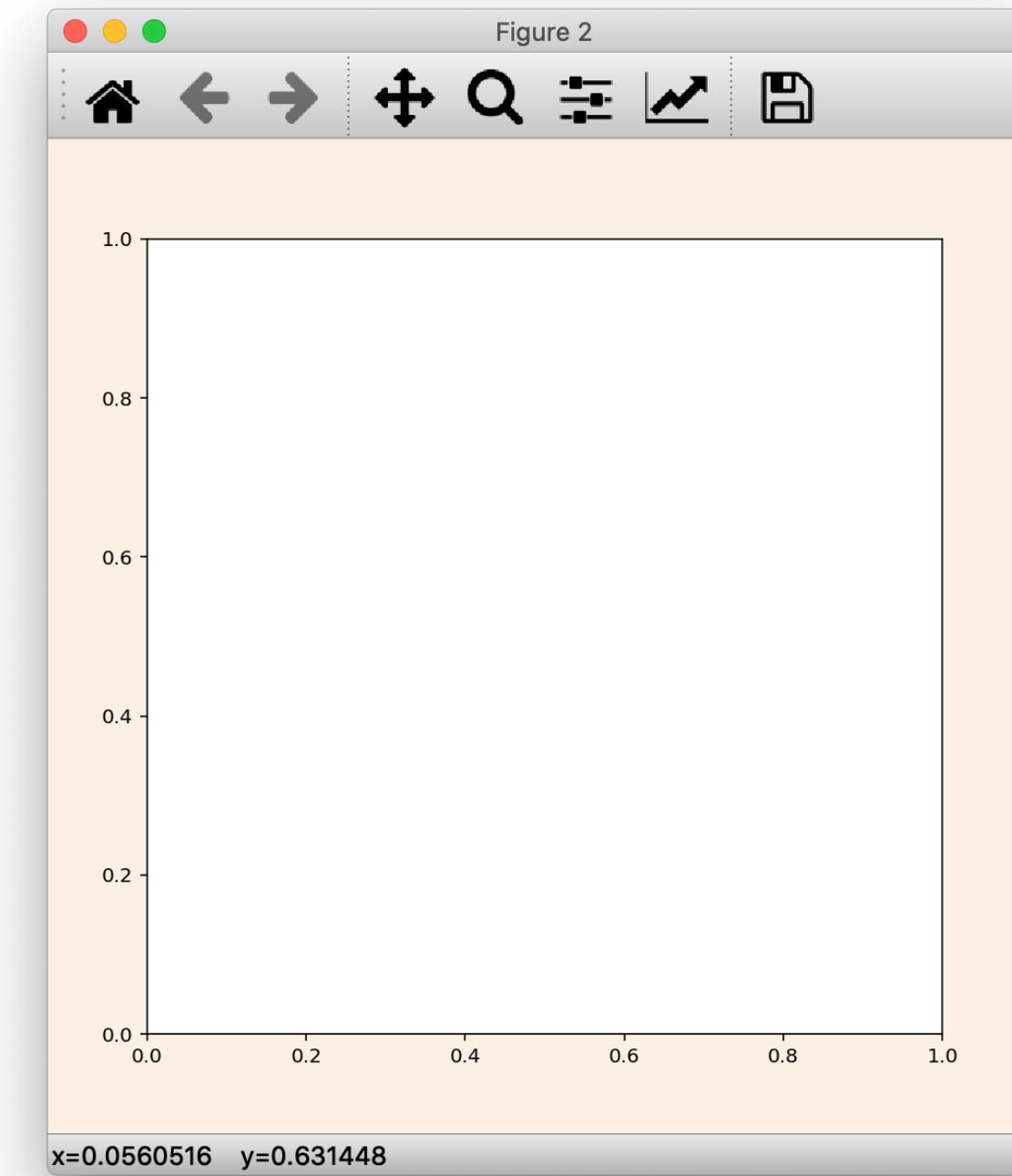
```
fig = plt.figure()
ax = fig.add_axes([left, bottom, width, height])
```

7. fig.add_axes(rect Argument)

```
import matplotlib.pyplot as plt  
import numpy as np  
  
fig = plt.figure(figsize=(7, 7),  
                 facecolor='linen')  
  
ax = fig.add_axes([0, 0, 1, 1])
```



```
fig = plt.figure(figsize=(7, 7),  
                 facecolor='linen')  
  
ax = fig.add_axes([0.1, 0.1, 0.8, 0.8])
```



7. fig.add_axes(rect Argument)

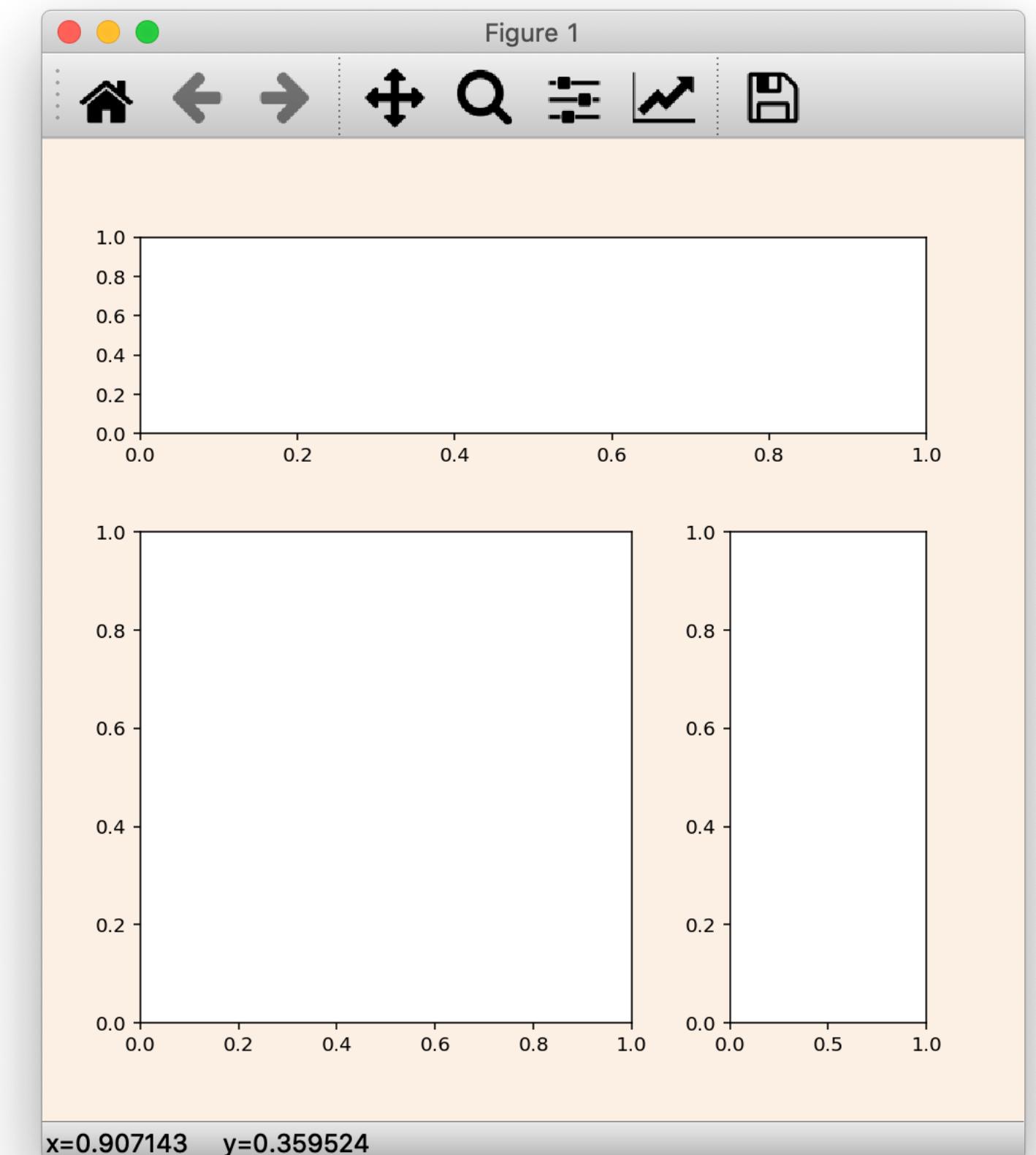
```
import matplotlib.pyplot as plt
import numpy as np

fig = plt.figure(figsize=(7, 7),
                 facecolor='linen')

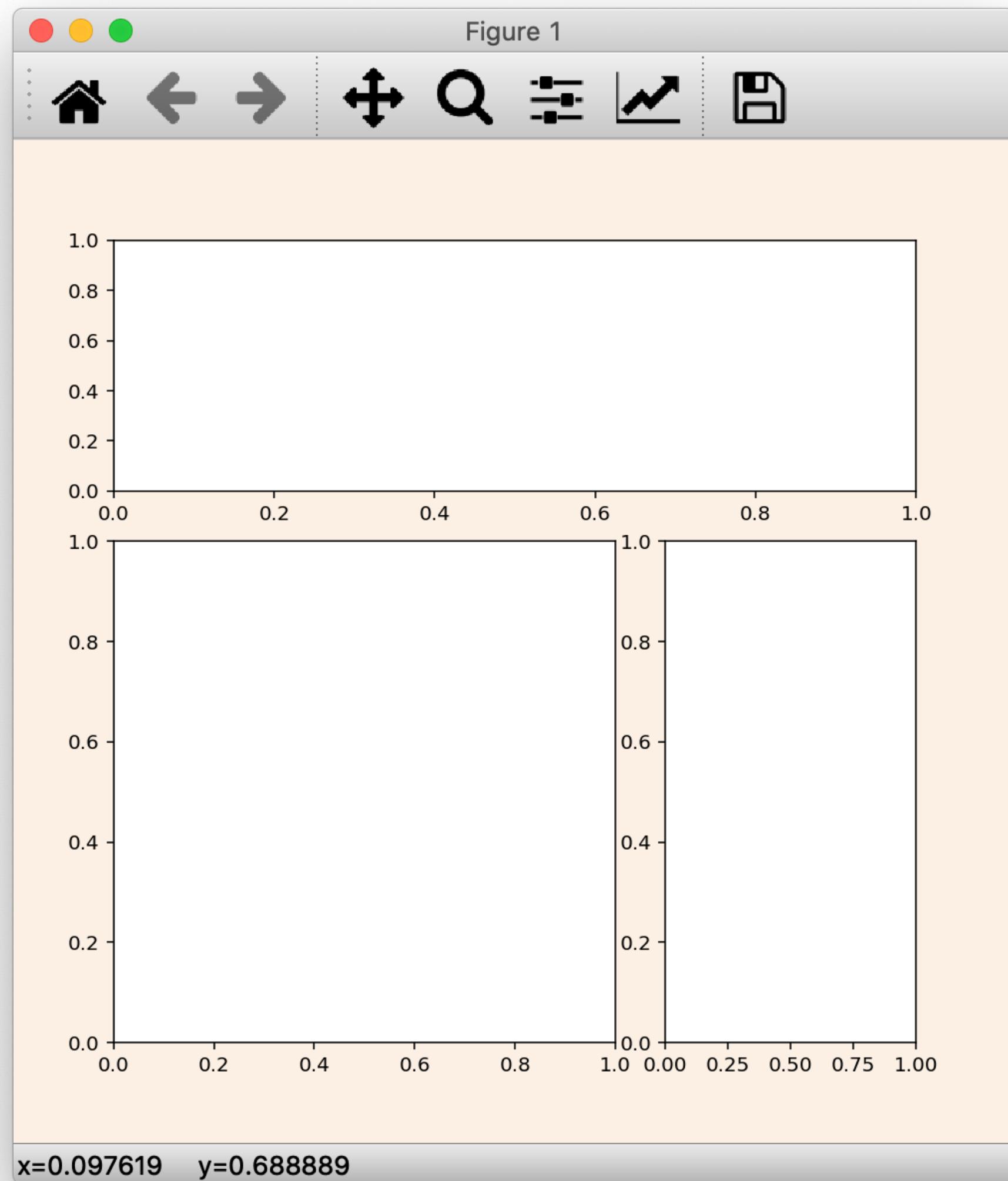
ax1 = fig.add_axes([0.1, 0.1, 0.5, 0.5])
ax2 = fig.add_axes([0.7, 0.1, 0.2, 0.5])
ax3 = fig.add_axes([0.1, 0.7, 0.8, 0.2])

-----
rect1 = [0.1, 0.1, 0.5, 0.5]
rect2 = [0.7, 0.1, 0.2, 0.5]
rect3 = [0.1, 0.7, 0.8, 0.2]

ax1 = fig.add_axes(rect1)
ax2 = fig.add_axes(rect2)
ax3 = fig.add_axes(rect3)
```



7. `fig.add_axes(Auto-scaled Axes)`



7. fig.add_axes(Auto-scaled Axes)

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

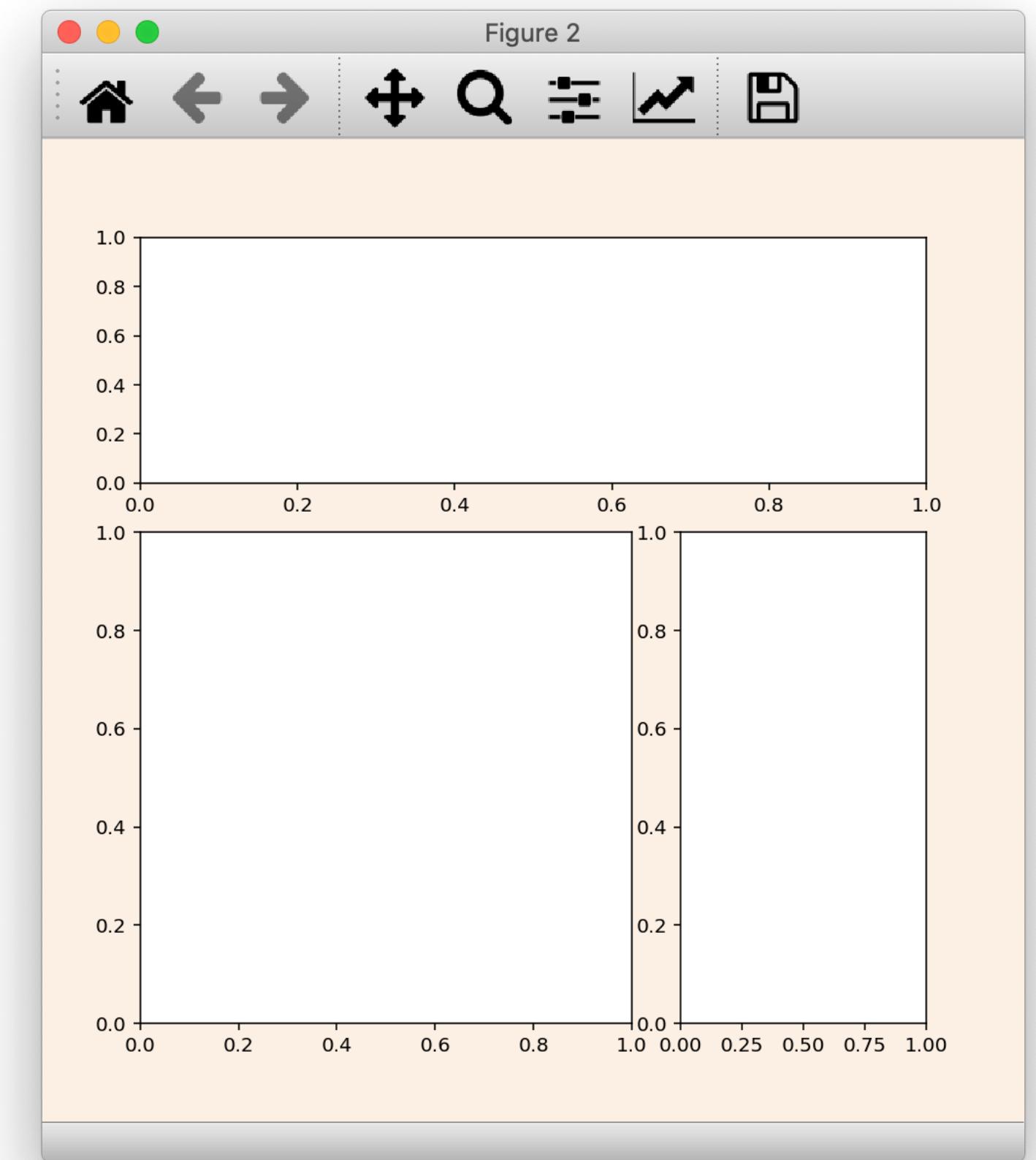
left, bottom = 0.1, 0.1
spacing = 0.05
width1, height1 = 0.5, 0.5

width2 = 1 - (2*left + width1 + spacing)
height2 = 1 - (2*bottom + height1 + spacing)

rect1 = [left, bottom, width1, height1]
rect2 = [left, bottom + height1 + spacing, 1 - 2*left, height2]
rect3 = [left + width1 + spacing, bottom, width2, height1]

fig = plt.figure(figsize=(7, 7),
                 facecolor='linen')

ax1 = fig.add_axes(rect1)
ax2 = fig.add_axes(rect2)
ax3 = fig.add_axes(rect3)
```

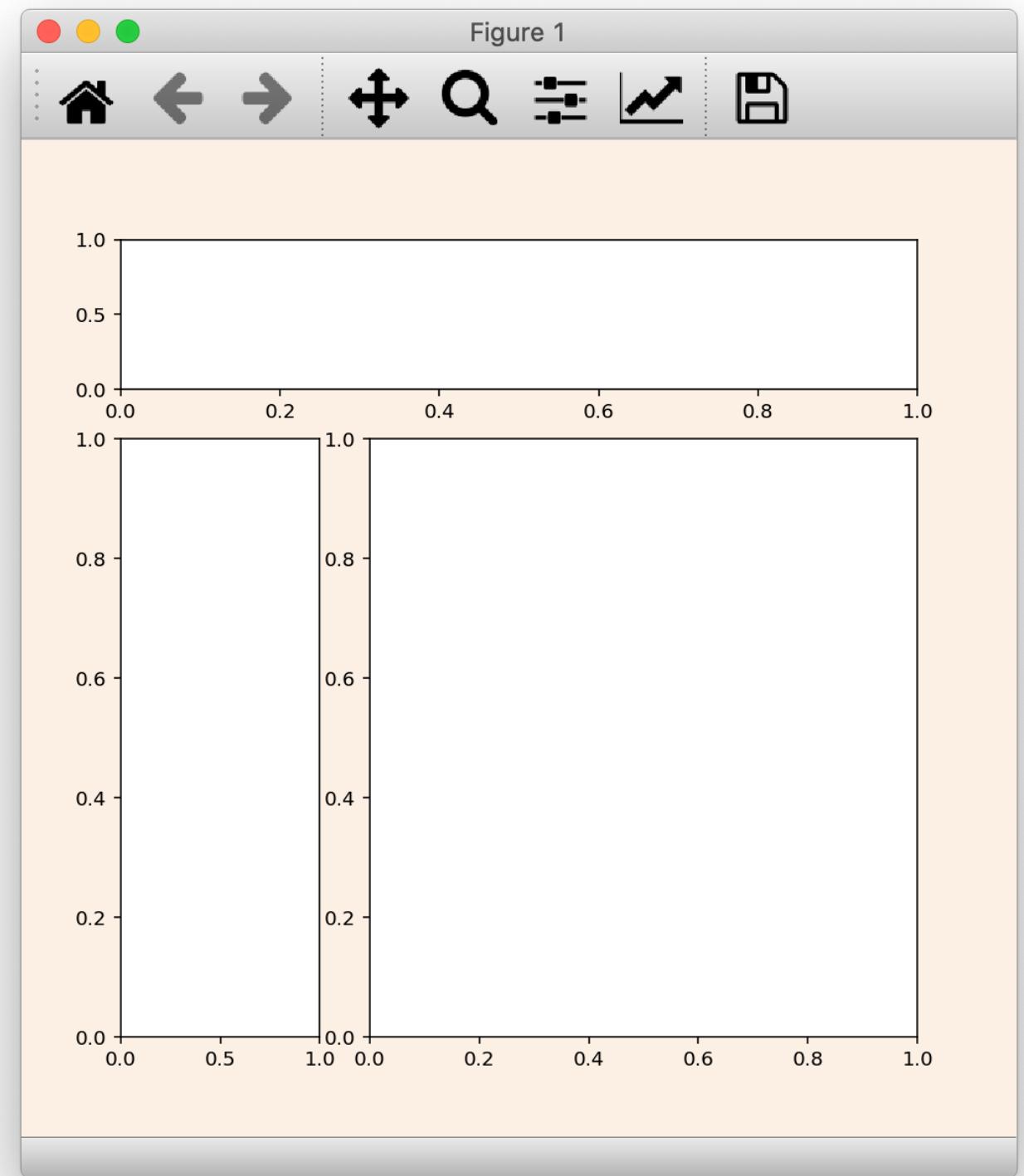
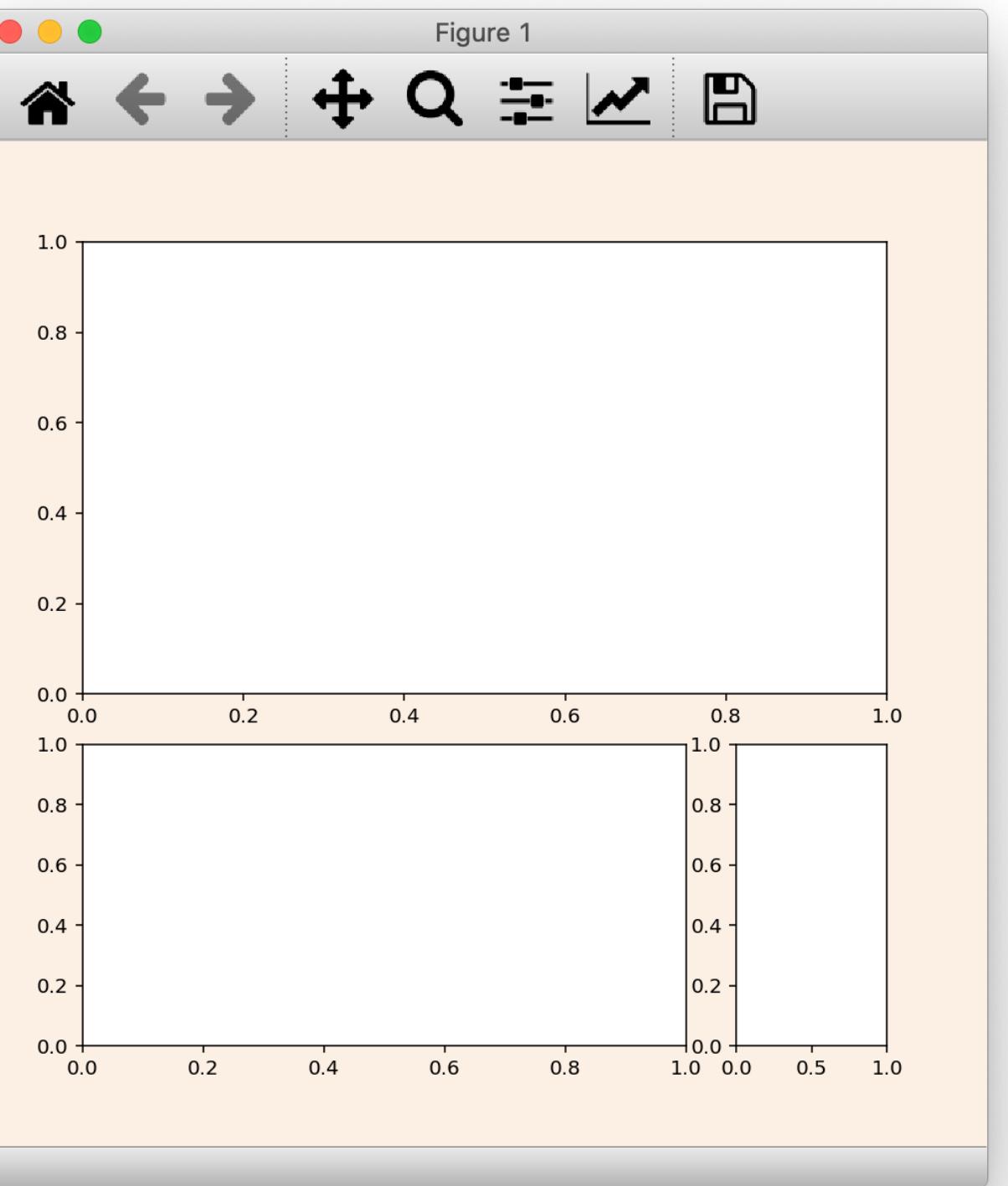
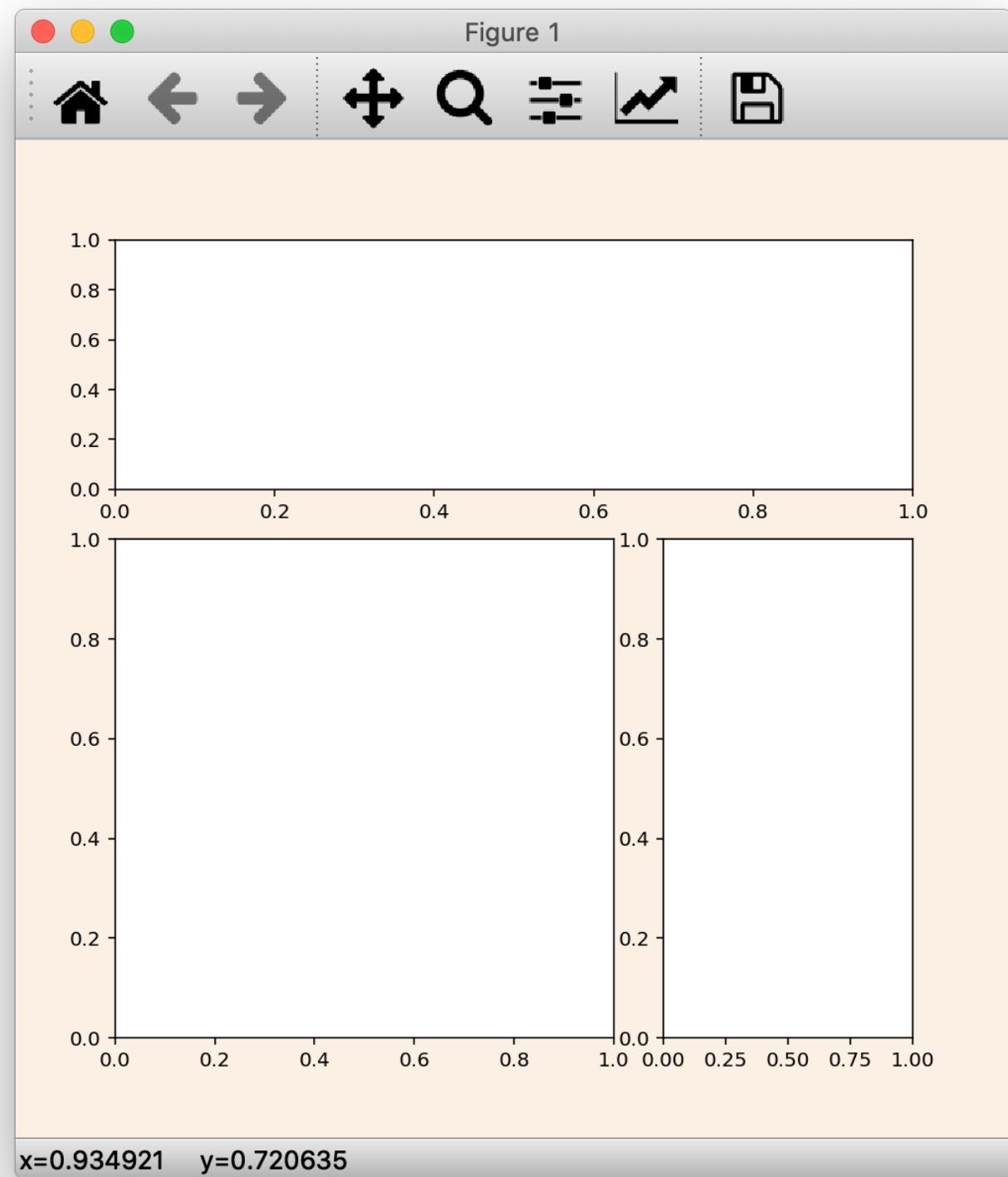


7. fig.add_axes(Auto-scaled Axes)

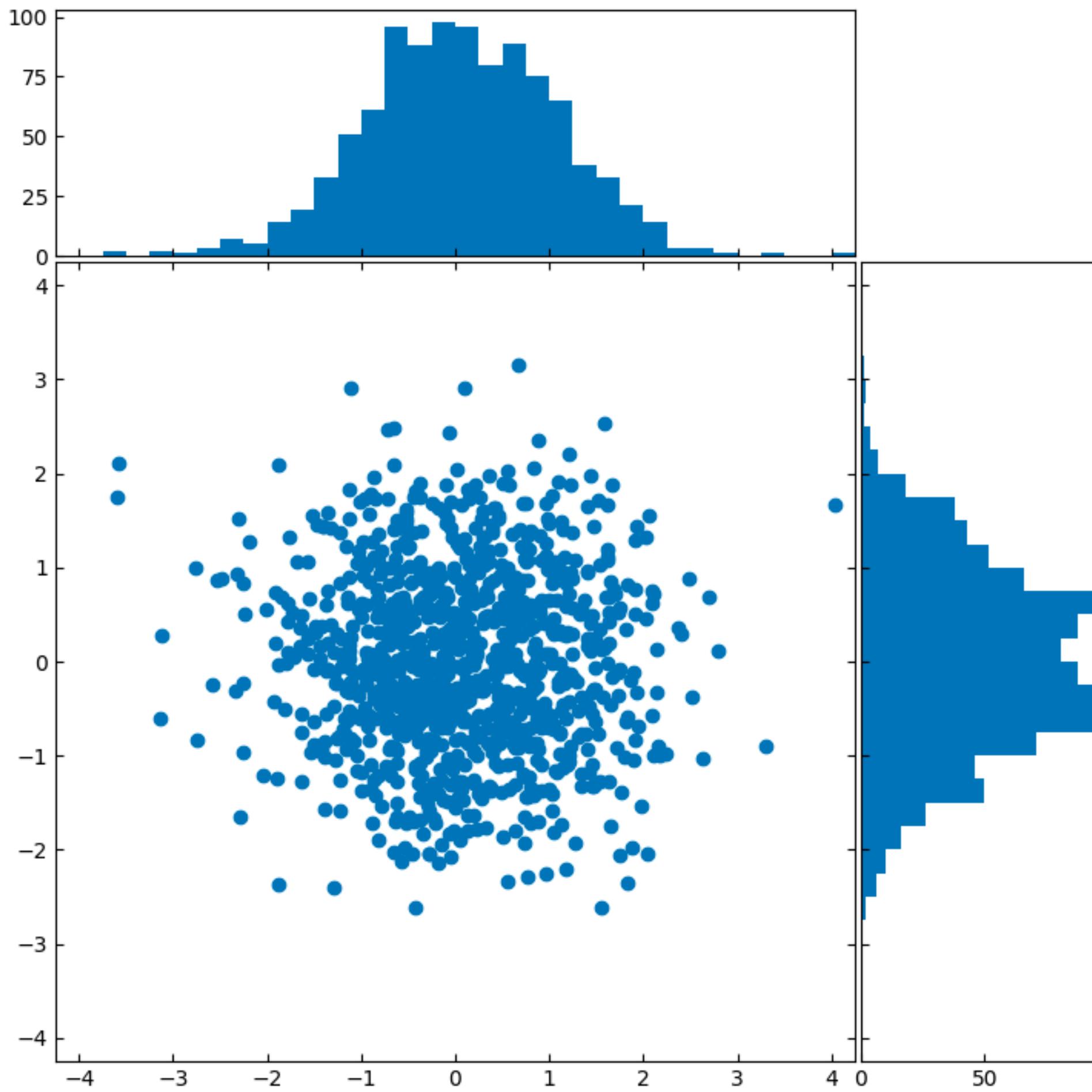
```
left, bottom = 0.1, 0.1  
spacing = 0.05  
width1, height1 = 0.5, 0.5
```

```
left, bottom = 0.1, 0.1  
spacing = 0.05  
width1, height1 = 0.6, 0.3
```

```
left, bottom = 0.1, 0.1  
spacing = 0.05  
width1, height1 = 0.2, 0.6
```



7. fig.add_axes(Example)



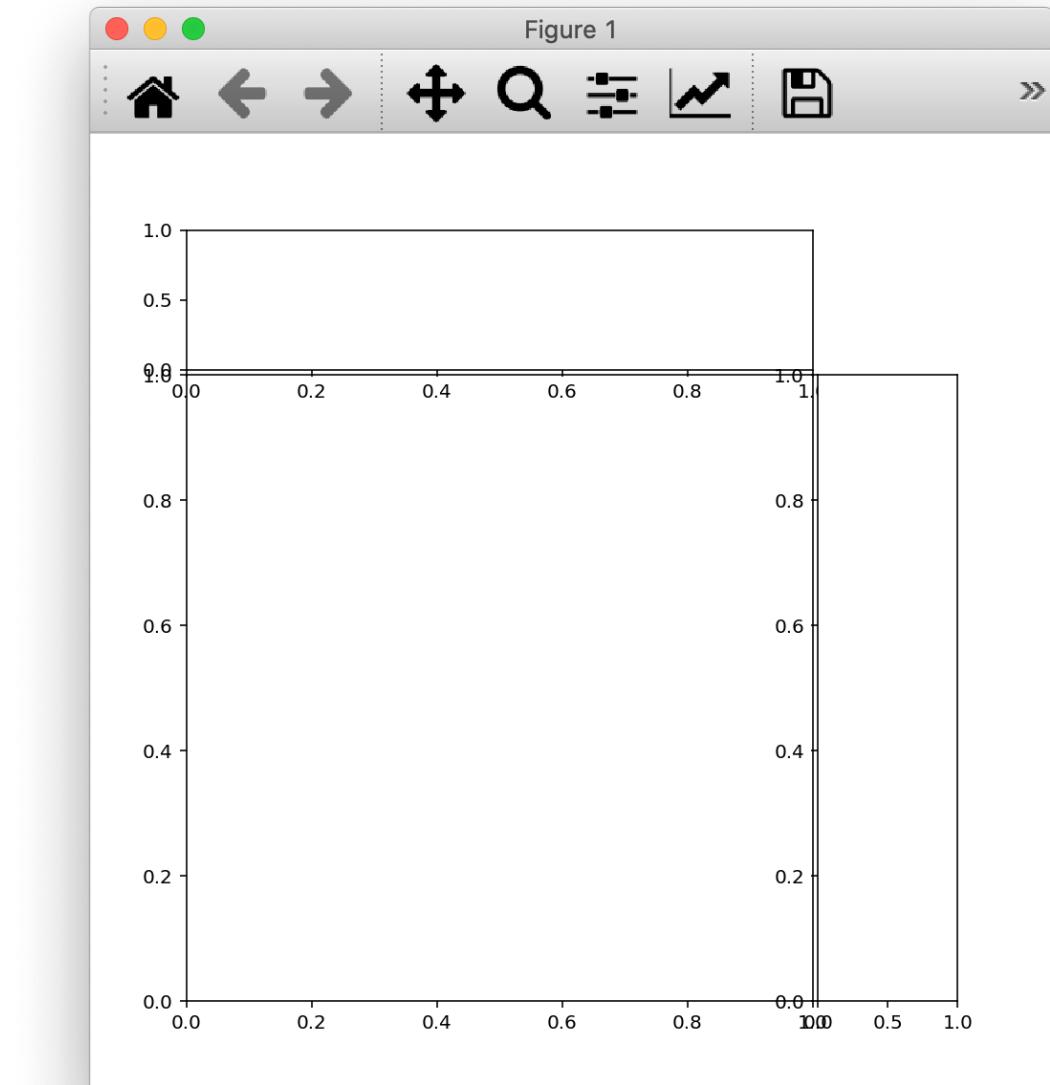
```
import matplotlib.pyplot as plt
import numpy as np

left, bottom = 0.1, 0.1
spacing = 0.005

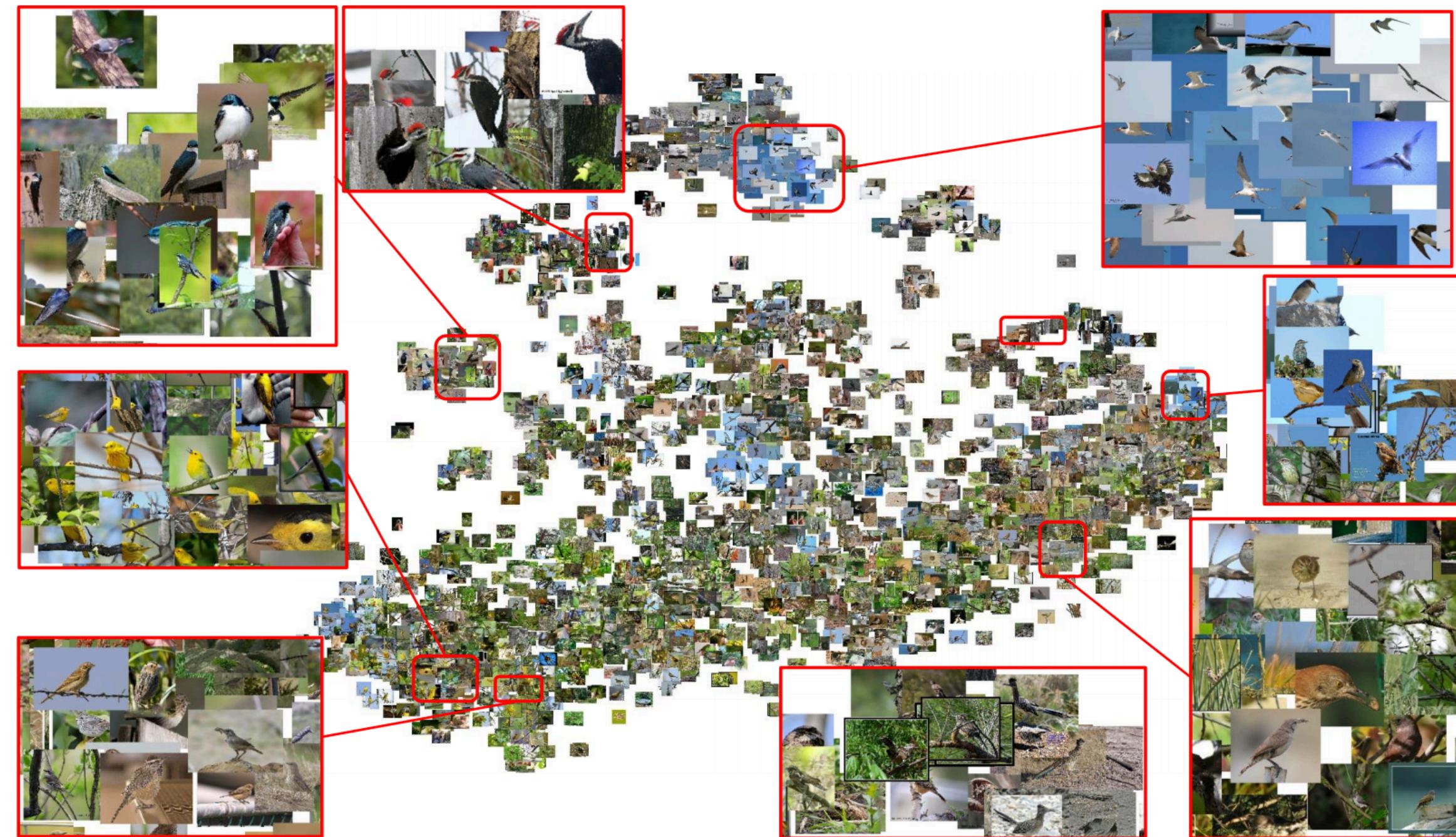
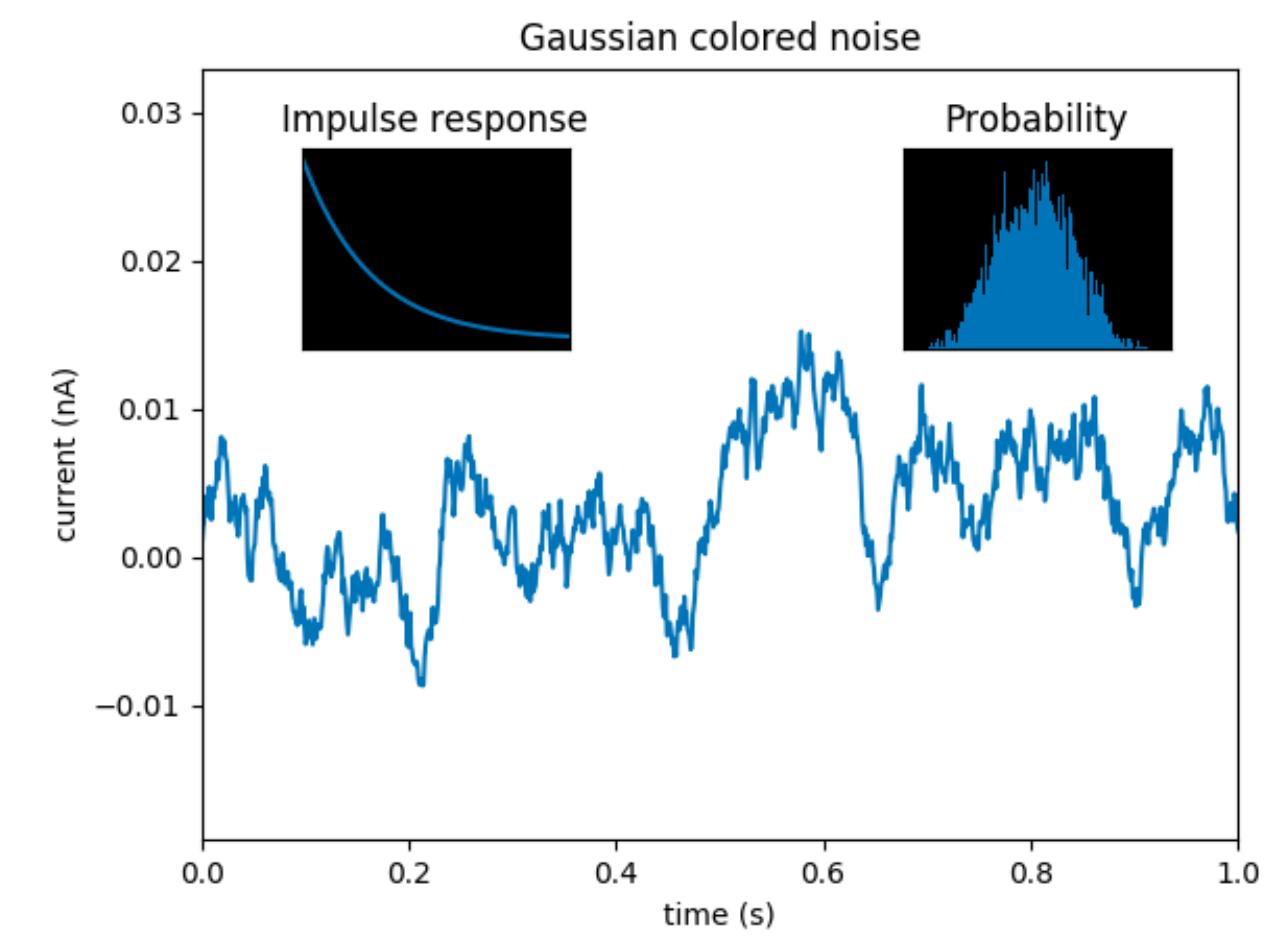
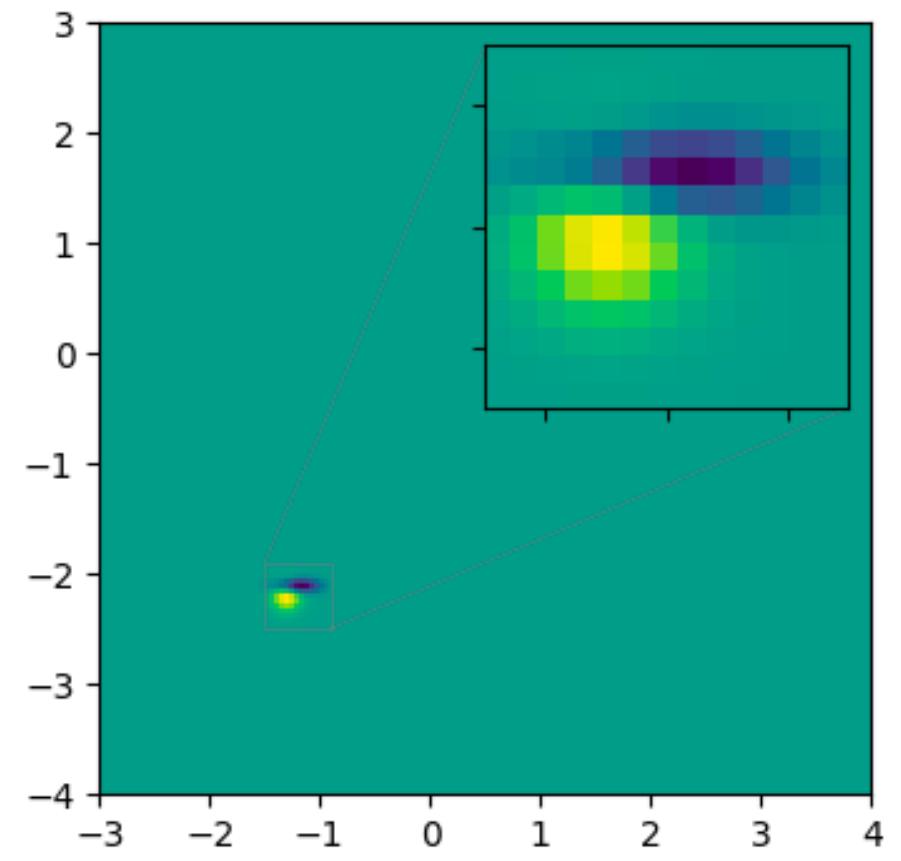
width1, height1, = 0.65, 0.65
width2 = 1 - (2*left + width1 + spacing)
height2 = 1 - (2*bottom + height1 + spacing)

rect1 = [left, bottom, width1, height1]
rect2 = [left, bottom + height1 + spacing, width1, height2]
rect3 = [left + width1 + spacing, bottom, width2, height1]

fig = plt.figure(figsize=(7, 7))
ax1 = fig.add_axes(rect1)
ax2 = fig.add_axes(rect2)
ax3 = fig.add_axes(rect3)
```



7. fig.add_axes(Zoom Axes)

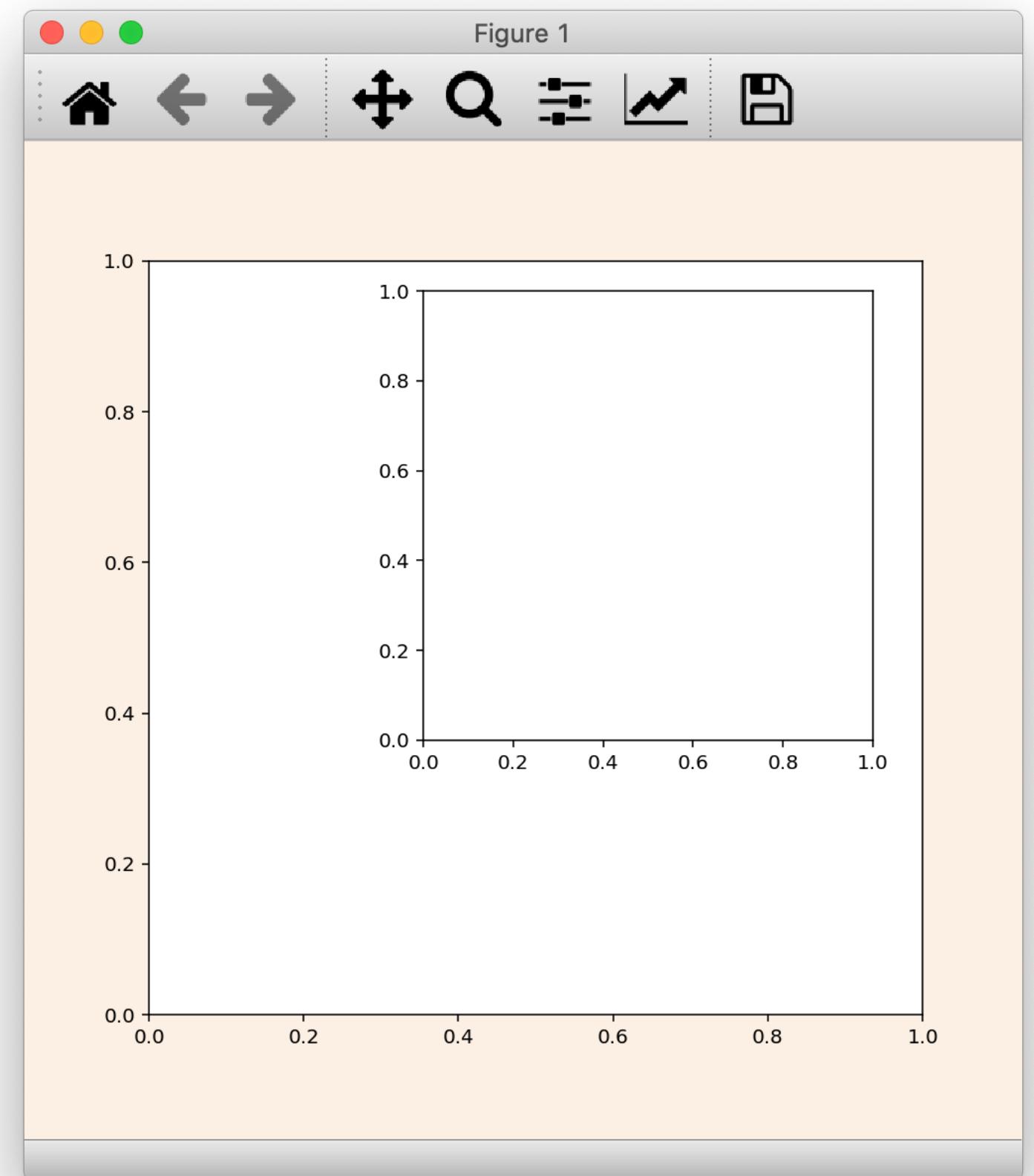


7. fig.add_axes(Zoom Axes)

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

fig = plt.figure(figsize=(7, 7),
                 facecolor='linen')

ax = fig.add_subplot(111)
ax_zoom = fig.add_axes([0.4, 0.4, 0.45, 0.45])
```



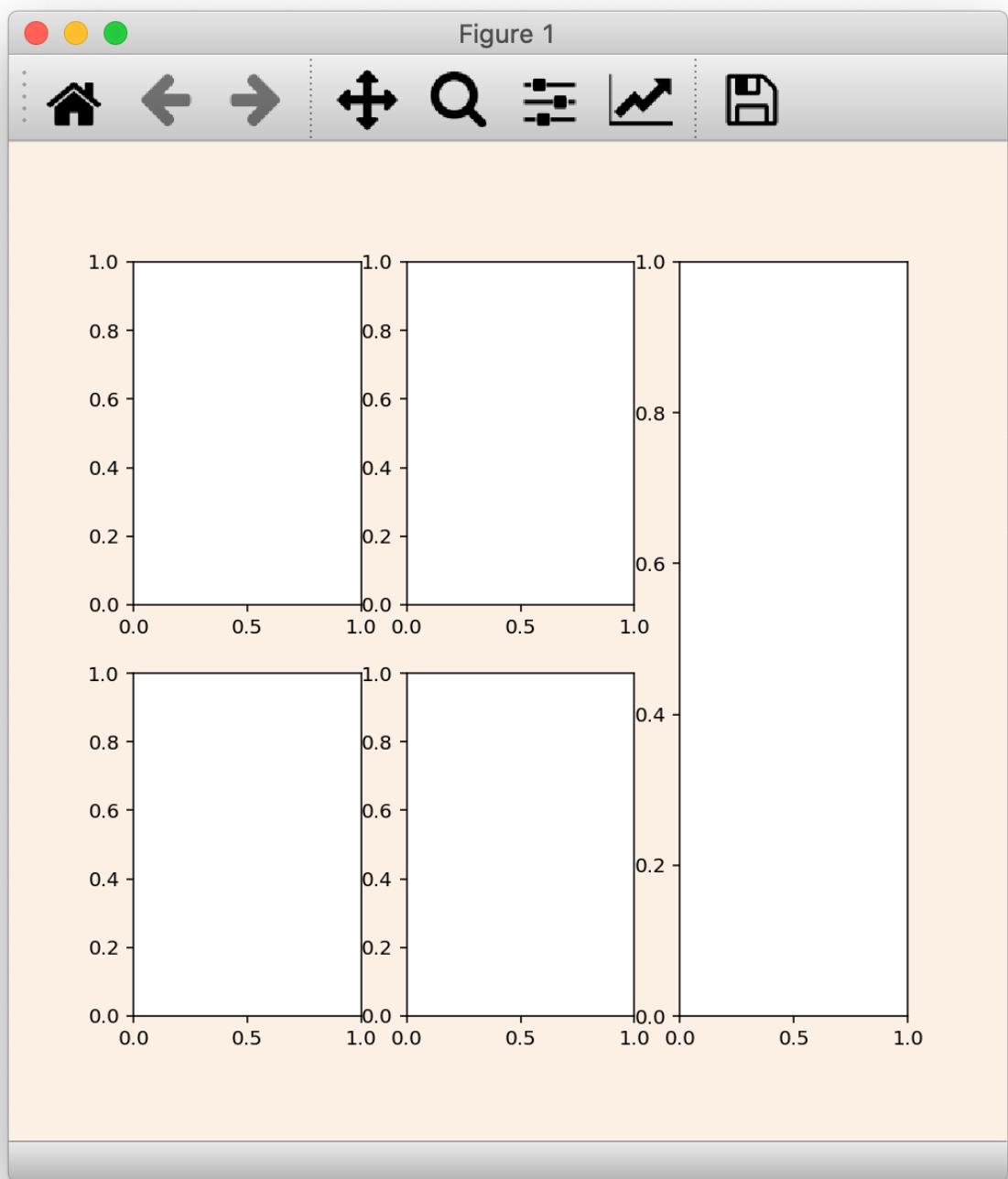
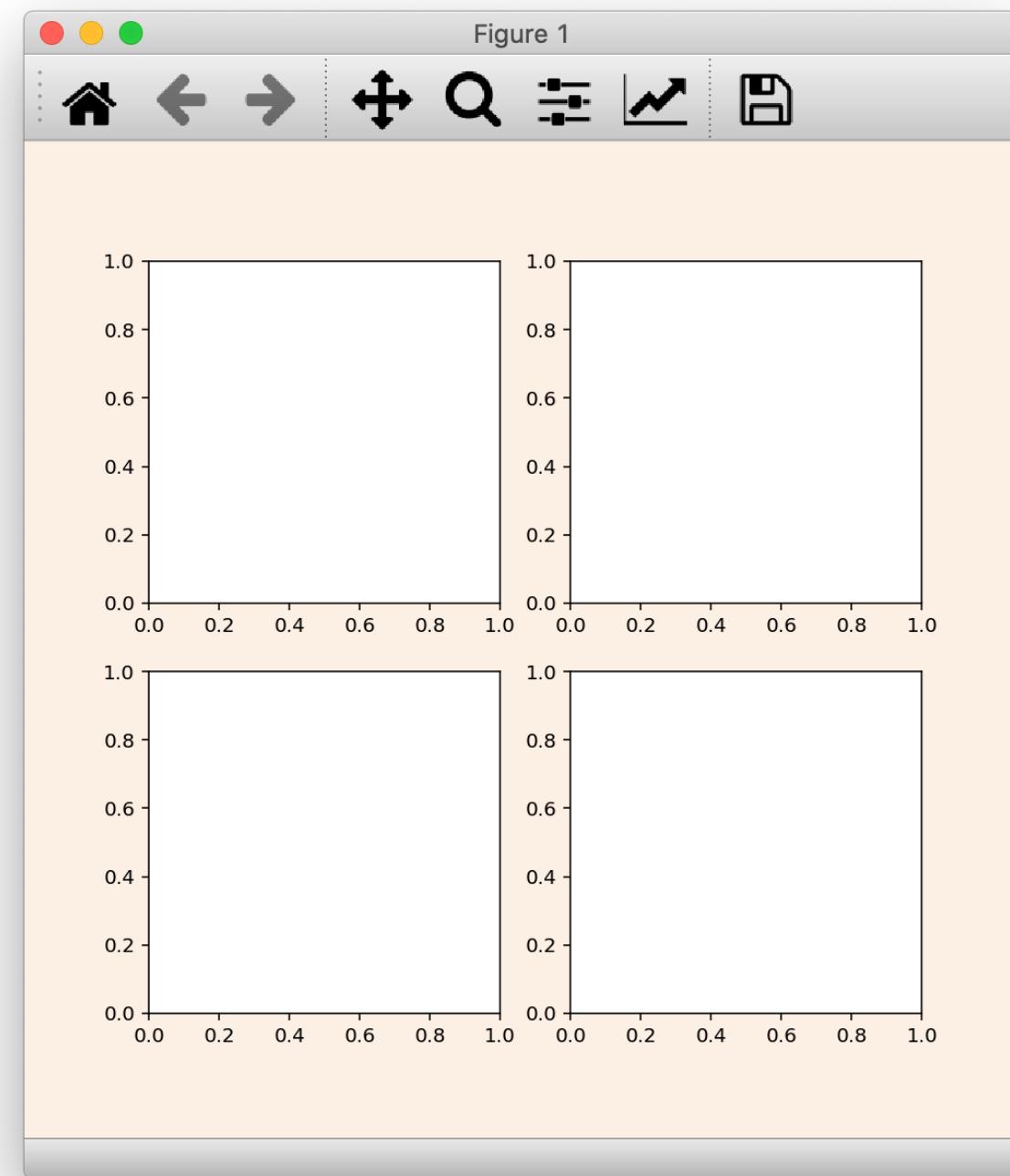
- Making Figures and Axes Review

```
fig = plt.figure()  
  
ax = fig.add_subplot()  
  
ax = fig.add_axes()  
  
ax = plt.subplot2grid(fig=fig)  
  
fig, axes = plt.subplots()
```

Lecture_1-01 Making Figures and Axes

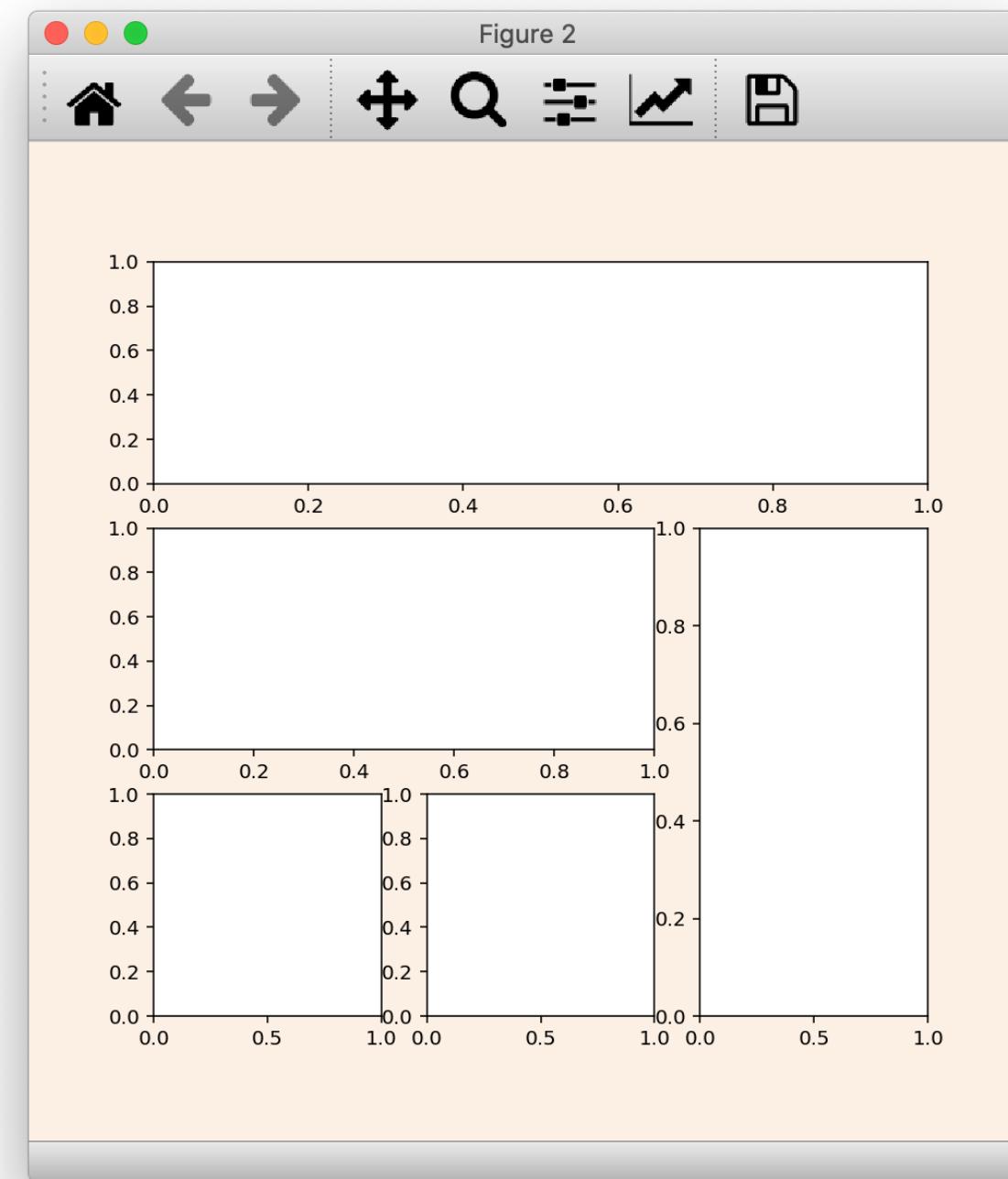
70

- Making Figures and Axes Review

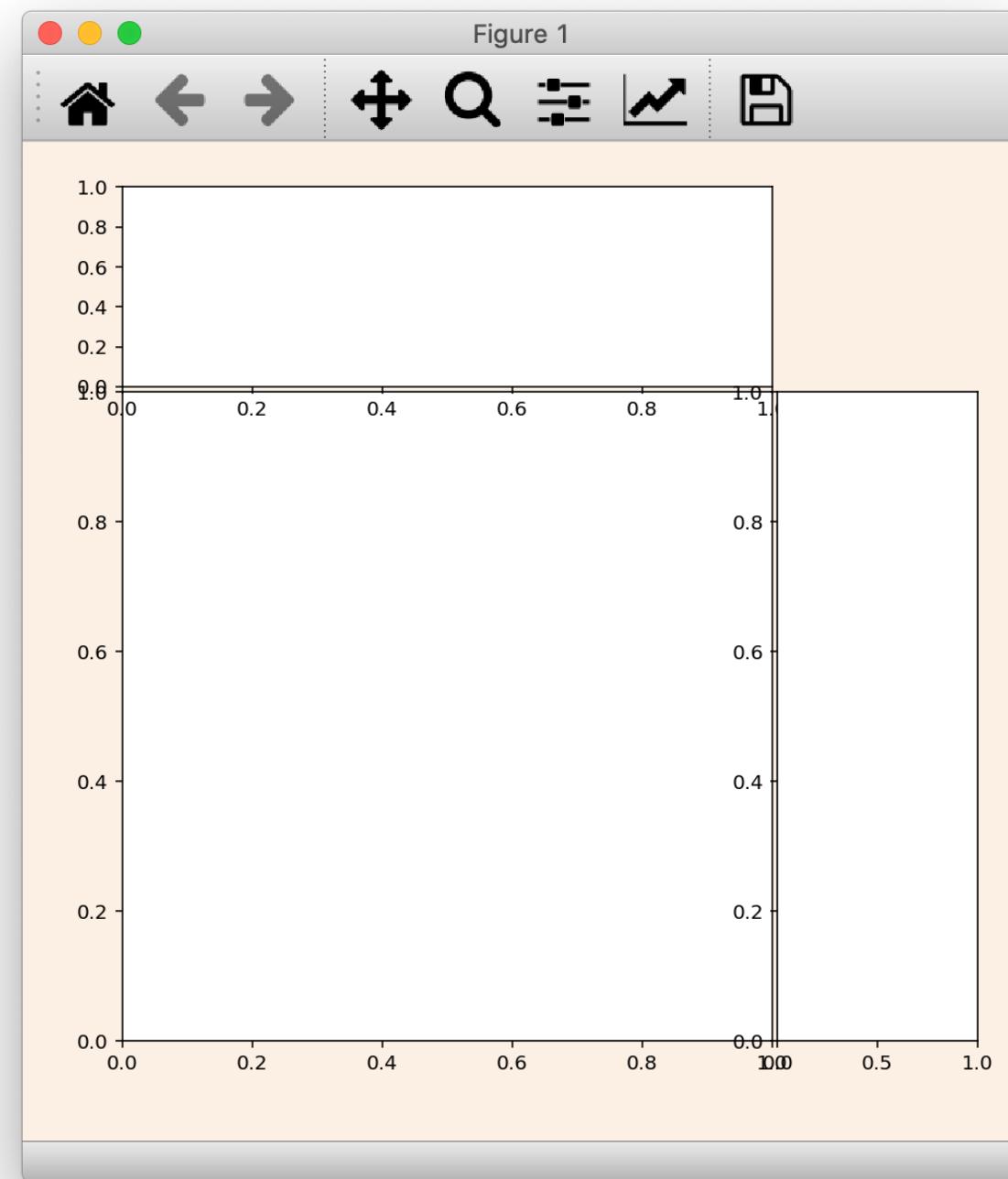


```
fig, axes = plt.subplots()
```

```
fig = plt.figure()  
ax = fig.add_subplot()
```



```
fig = plt.figure()  
ax = plt.subplot2grid(fig=fig)
```



```
ax = fig.add_axes()
```

Python for Data Visualization

-Chapter.1 Matplotlib Anatomy -

1-01. Making Figures and Axes

1. Figures and Axes
2. `plt.figure(Making Figures)`
3. `fig.add_subplot(Adding Subplots)`
4. `plt.subplots(Making Fig and Axes Simultaneously)`
5. `plt.subplot2grid(More Complex Arrangement)`
6. Practice
7. `fig.add_axes(Arbitrary Locations and Sizes of Axes)`