- Shin’s Lab -

Python for Data Visualization

Python for Data Visualization

-Chapter.2 Line Plot -

2-00. Intro to Line Plot

2-01. Line Plot Basics

2-02. Labels and Legend

2-03. Line Styles and Markers

2-04. Line Filling

2-05. Exercises

Python for Data Visualization

-Chapter.2 Line Plot -

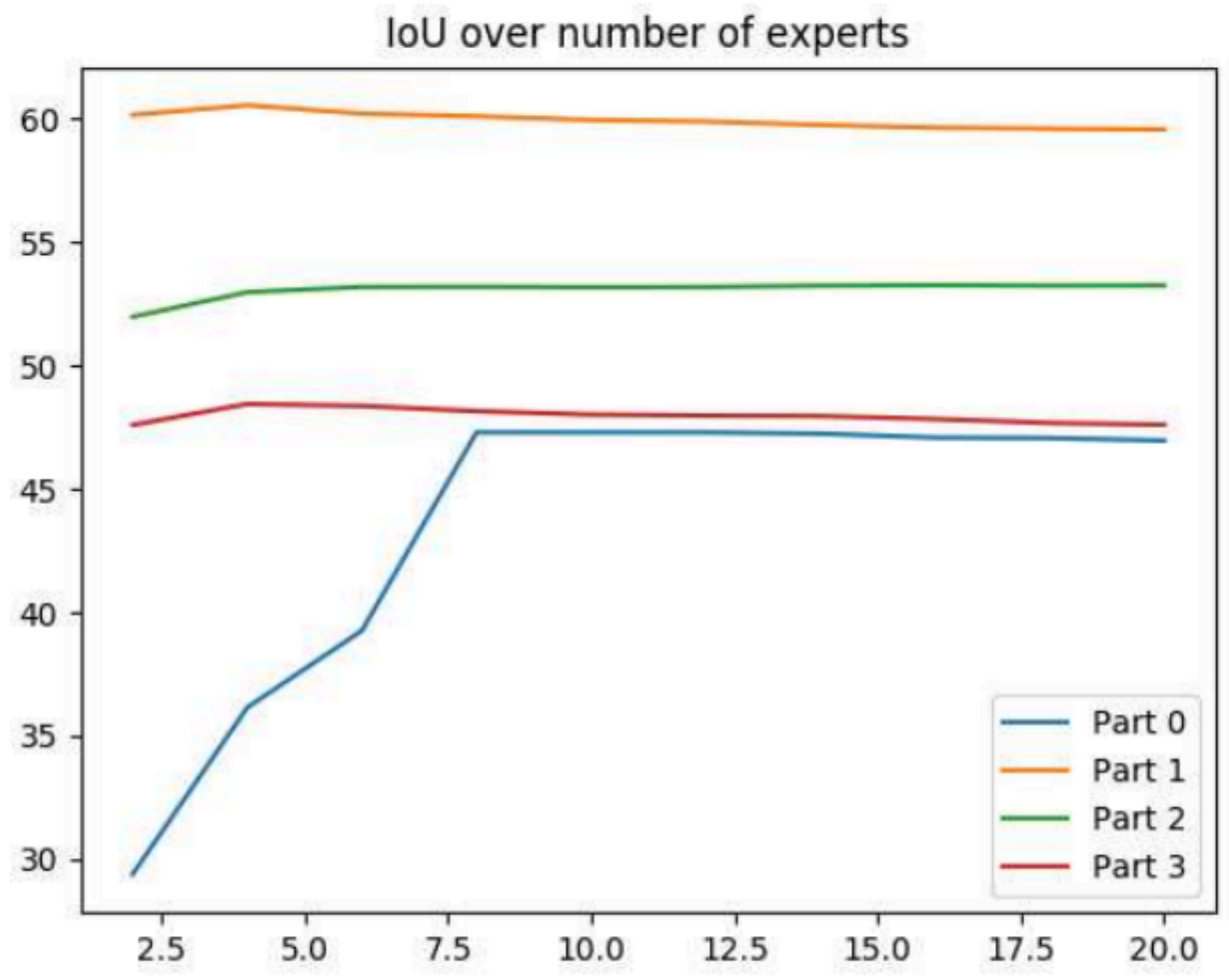
2-05. Exercises

1. Exercise 02-01
2. Exercise 02-02
3. Exercise 02-03
4. Exercise 02-04
5. Exercise 02-05
6. Exercise 02-06
7. Exercise 02-07
8. Exercise 02-08

1. Exercise 02-01

Feature Weighting and Boosting for Few-Shot Segmentation

Khoi Nguyen and Sinisa Todorovic
Oregon State University
Corvallis, OR 97330, USA
{nguyenkh,sinisa}@oregonstate.edu

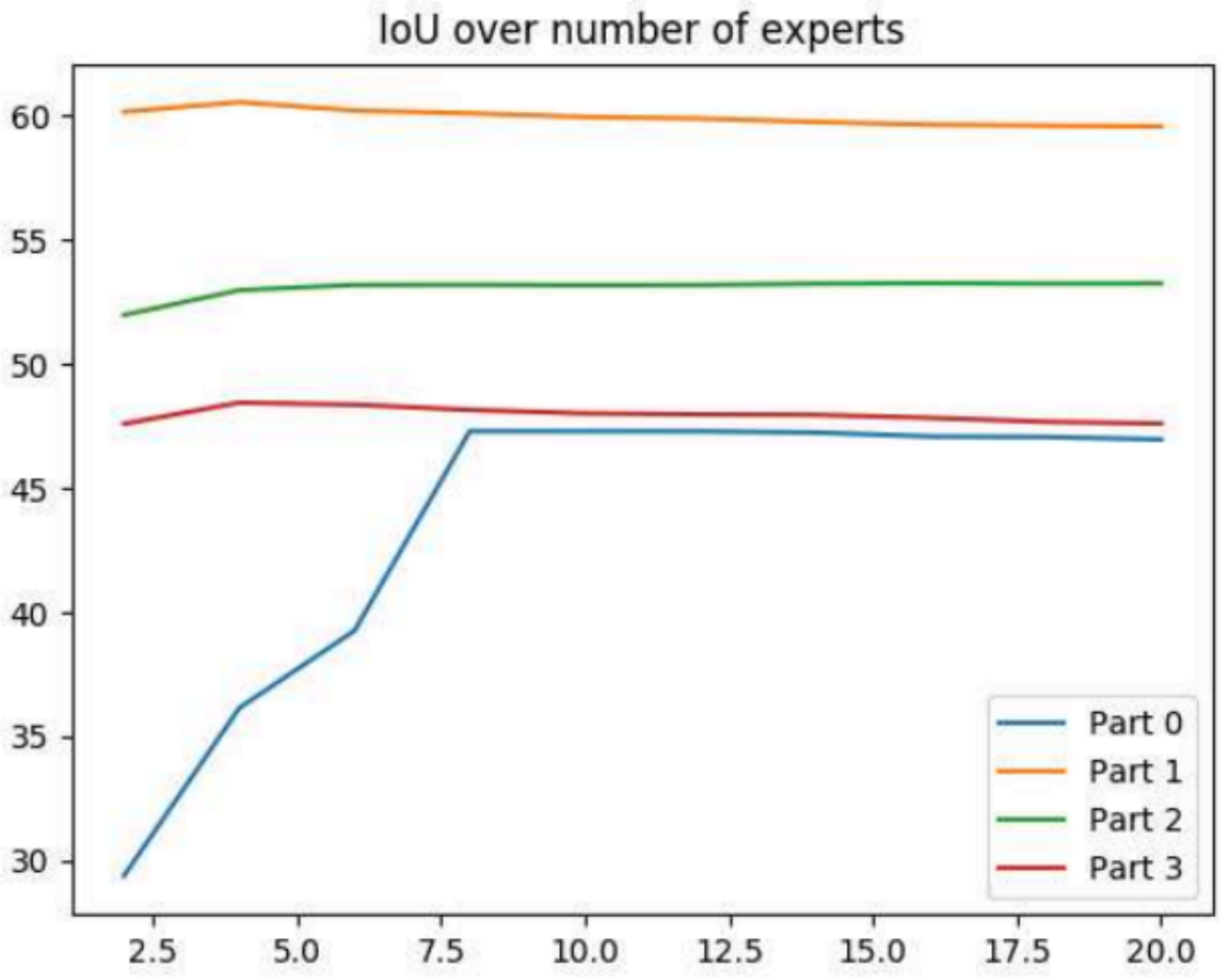


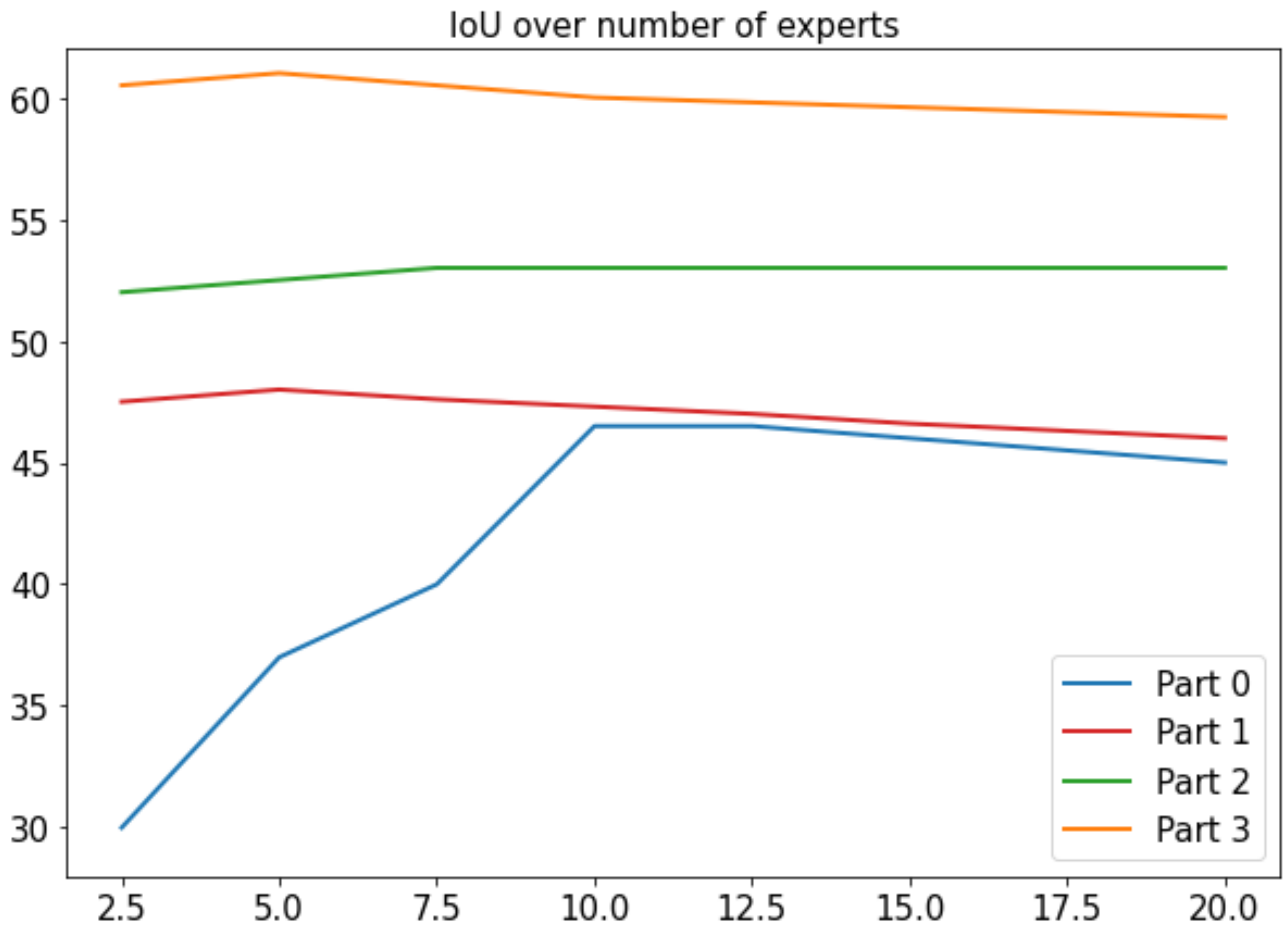
1. Exercise 02-01

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

### rcParam setting
plt.rcParams['lines.linewidth'] = 2

### data setting
x_loc = np.linspace(2.5, 20, 8)
data = np.array([[30, 37, 40, 46.5, 46.5, 46, 45.5, 45],
                 [47.5, 48, 47.6, 47.3, 47, 46.6, 46.3, 46],
                 [52, 52.5, 53, 53, 53, 53, 53, 53],
                 [60.5, 61, 60.5, 60, 59.8, 59.6, 59.4, 59.2]])
name_list = ['Part ' + str(i) for i in range(4)]

### customizing setting
color_list = ['tab:blue', 'tab:red', 'tab:green', 'tab:orange']

### plotting
fig, ax = plt.subplots(figsize=(10, 7))
for line_idx in range(len(name_list)):
    ax.plot(x_loc, data[line_idx],
            color=color_list[line_idx],
            label=name_list[line_idx])
```

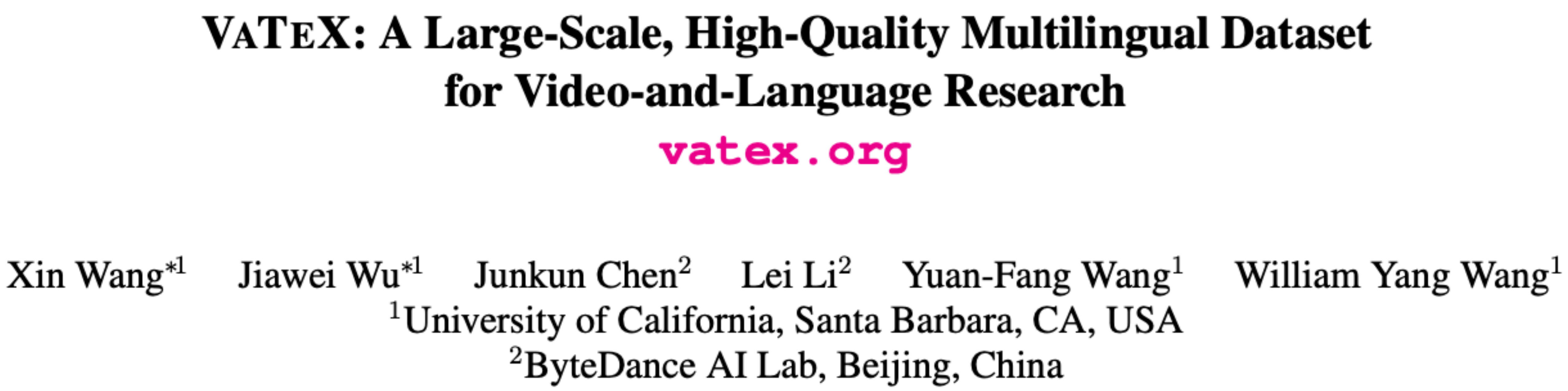
```python
### customizing
ax.set_ylim([28, 62])
ax.set_title('IoU over number of experts',
             fontsize=15)
ax.legend(loc='lower right',
          fontsize=15)
ax.tick_params(labelsize=15)
```

1. Exercise 02-01





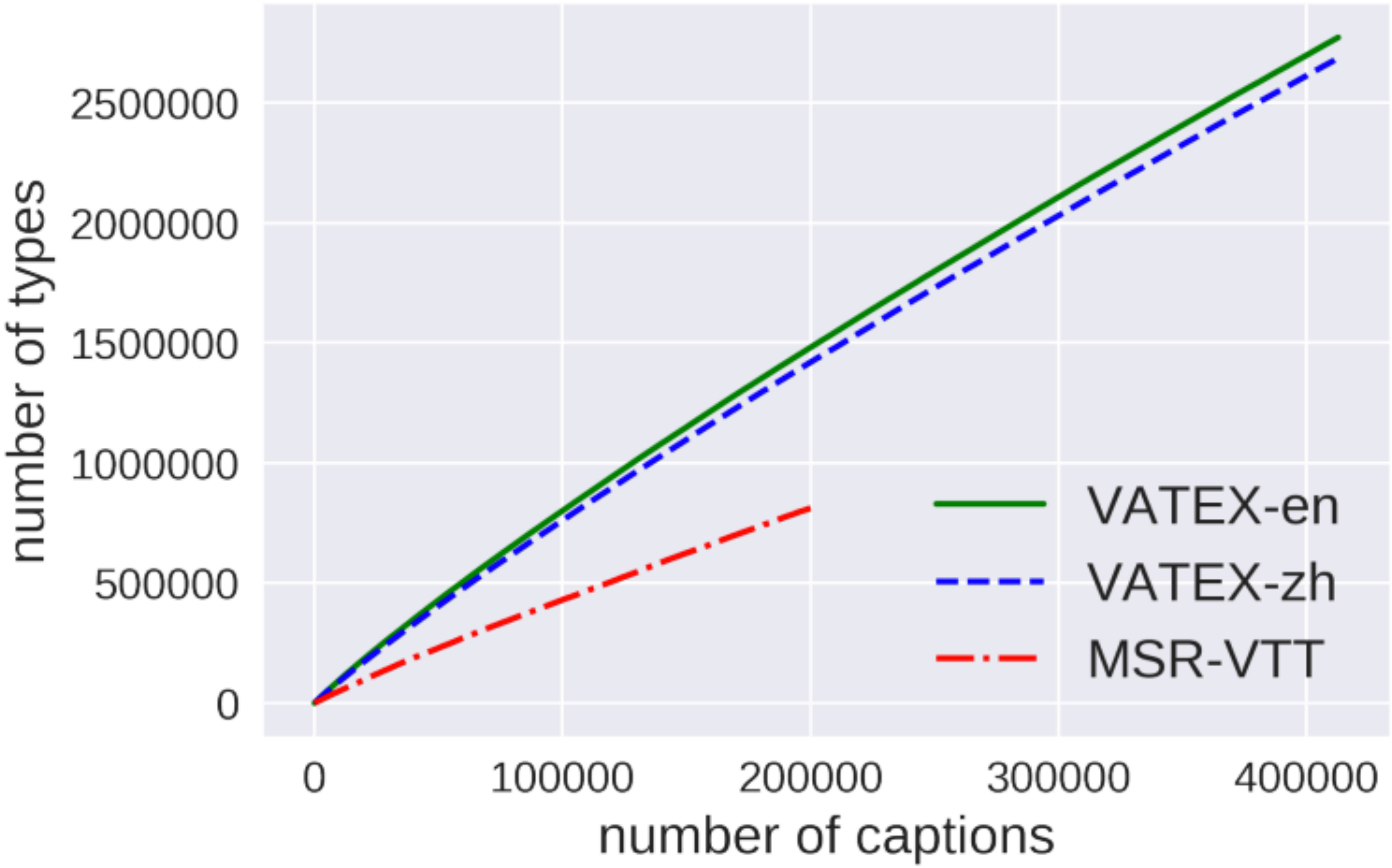
2. Exercise 02-02



VATEX: A Large-Scale, High-Quality Multilingual Dataset for Video-and-Language Research

vatex.org

Xin Wang*[1] Jiawei Wu*[1] Junkun Chen[2] Lei Li[2] Yuan-Fang Wang[1] William Yang Wang[1]
[1]University of California, Santa Barbara, CA, USA
[2]ByteDance AI Lab, Beijing, China

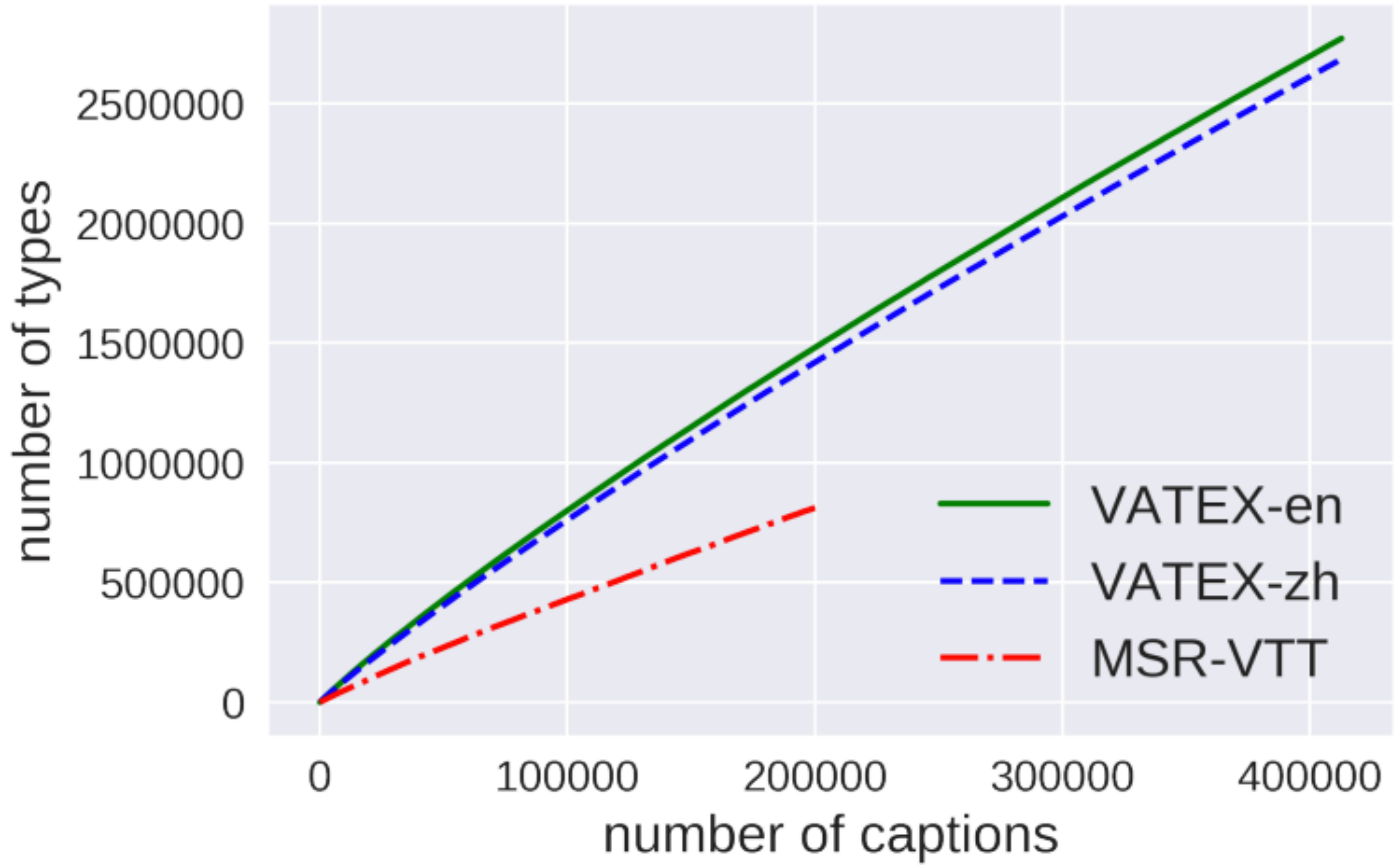


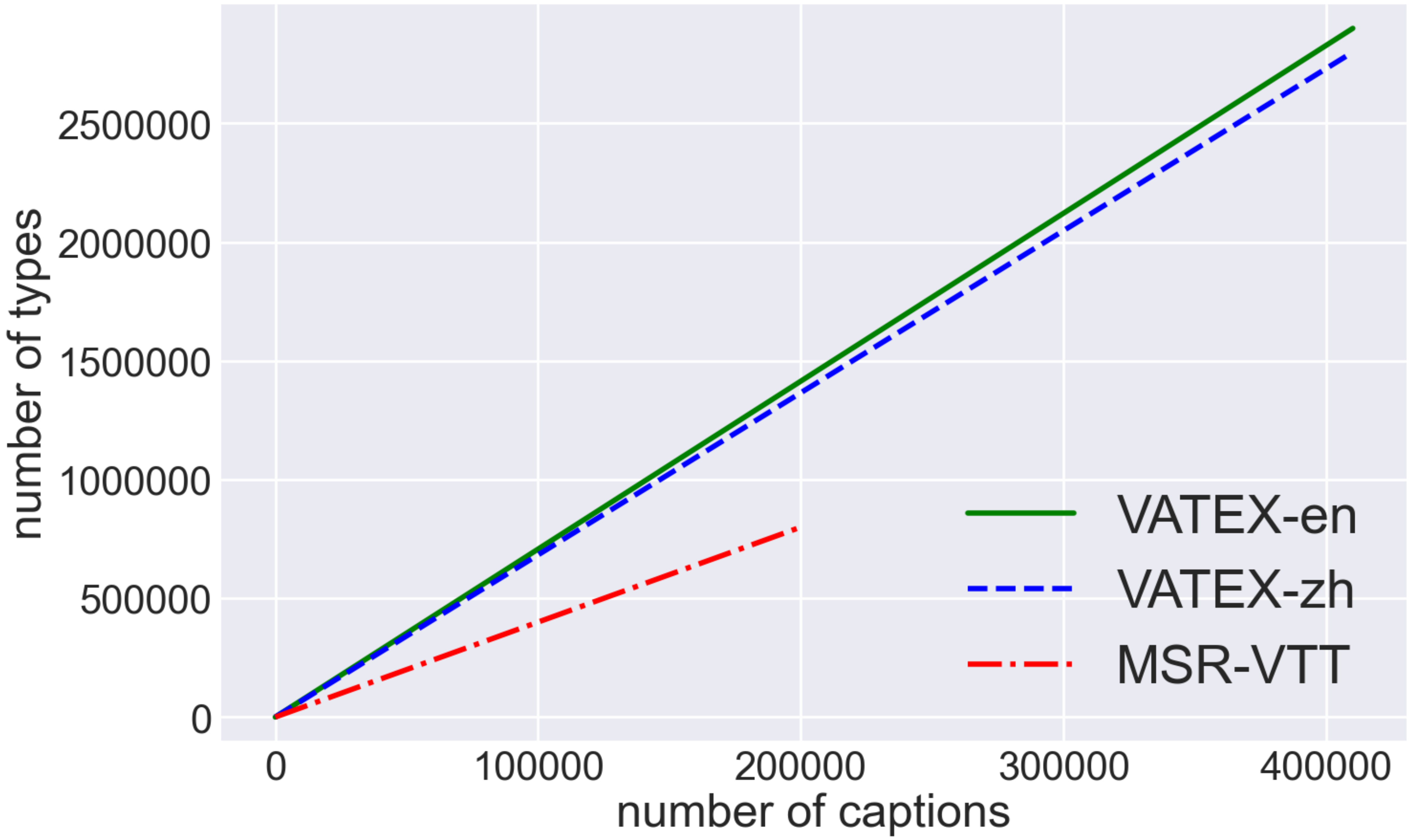
2. Exercise 02-02

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

### rcParam setting
plt.style.use('seaborn')
plt.rcParams['lines.linewidth'] = 4

### data setting
msr_x = np.linspace(0, 2E5, 10)
msr_y = np.linspace(0, 8E5, 10)

vatex_x = np.linspace(0, 4.1E5, 20)
vatex_en_y = np.linspace(0, 29E5, 20)
vatex_zh_y = np.linspace(0, 28E5, 20)

### plotting
fig, ax = plt.subplots(figsize=(16, 10))
ax.plot(vatex_x, vatex_en_y,
        color='g',
        label='VATEX-en')
ax.plot(vatex_x, vatex_zh_y,
        linestyle='--',
        color='b',
        label='VATEX-zh')
ax.plot(msr_x, msr_y,
        linestyle='-.',
        color='r',
        label='MSR-VTT')
```

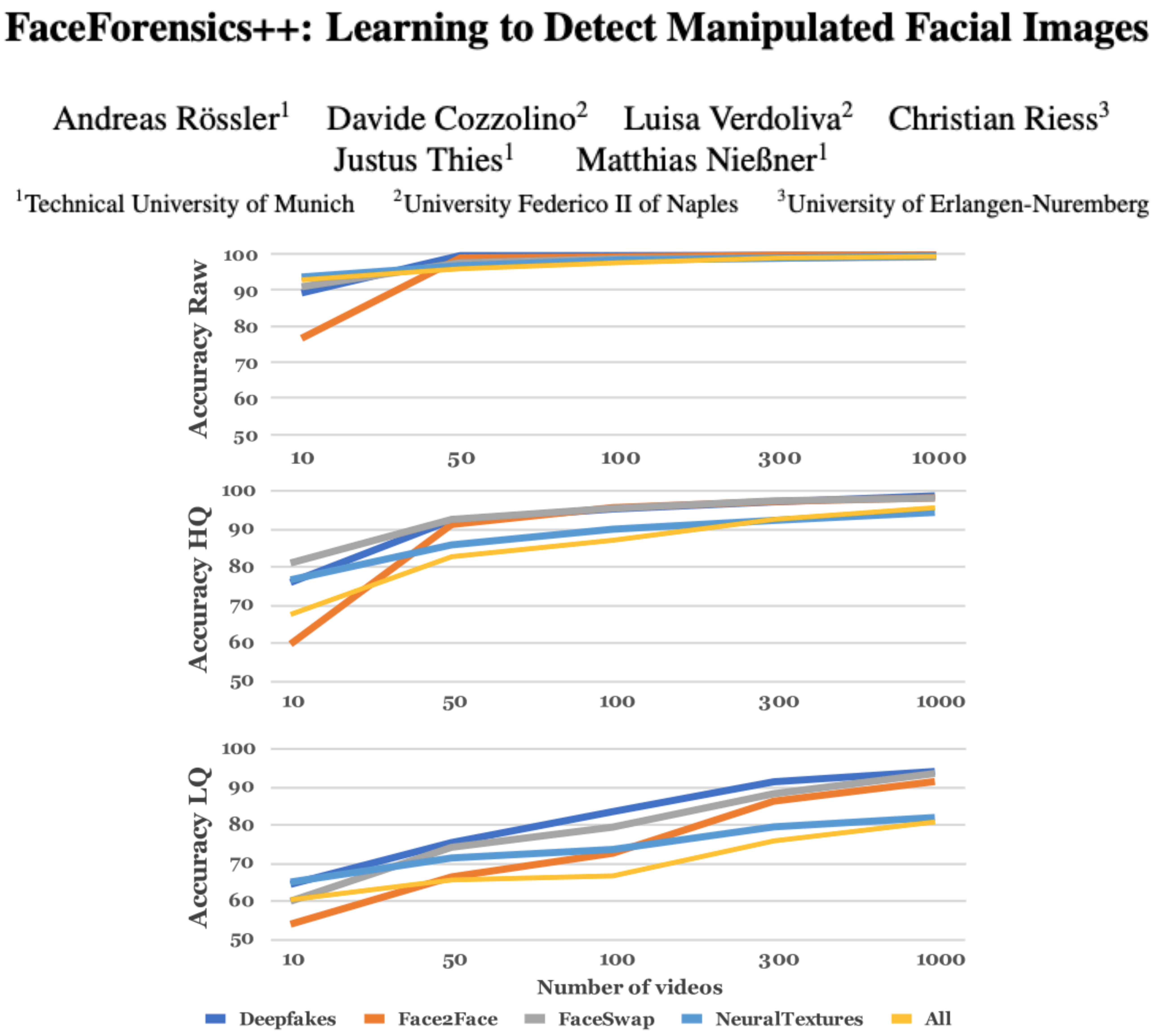
```
### axis customizing
ax.set_ylim([-1E5, 3E6])
y_ticks = np.linspace(0, 25E5, 6)
ax.set_yticks(y_ticks)

### tick & grid customizing
ax.ticklabel_format(axis='both',
                    style='plain')
ax.tick_params(labelsize=30)
ax.grid(linewidth=2)

### legend customizing
ax.legend(loc='lower right',
          fontsize=40)

### label customizing
ax.set_xlabel('number of captions',
              fontsize=35)
ax.set_ylabel('number of types',
              fontsize=35)
```

2. Exercise 02-02





3. Exercise 02-03



FaceForensics++: Learning to Detect Manipulated Facial Images

Andreas Rössler[1] Davide Cozzolino[2] Luisa Verdoliva[2] Christian Riess[3]
Justus Thies[1] Matthias Nießner[1]

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

3. Exercise 02-03

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

### rcParam setting
plt.rcParams['font.family'] = 'serif'

### data setting
data_dict_list = []
data_dict1 = {'DF':[90, 99, 100, 100, 100],
             'F2F':[77, 98, 99, 100, 100],
             'FS':[91,96,98,99,100],
             'NT':[94,95,97,98,100],
             'All':[93,94,96,97,100]}
data_dict_list.append(data_dict1)
data_dict2 = {'DF':[77, 91, 95, 98, 99],
             'F2F':[60, 91, 95, 98, 99],
             'FS':[81, 91, 95, 98, 99],
             'NT':[77, 85, 90, 92, 95],
             'All':[68, 82, 88, 92, 96]}
data_dict_list.append(data_dict2)
data_dict3 = {'DF':[66, 76, 83, 91, 94],
             'F2F':[55, 67, 72, 88, 91],
             'FS':[60, 75, 80, 88, 91],
             'NT':[66, 71, 72, 80, 82],
             'All':[60, 66, 67, 77, 81]}
data_dict_list.append(data_dict3)
line_name_list = ['Deepfakes', 'Face2Face', 'FaceSwap',
            'NeuralTextures', 'All']
ylabel_list = ['Accuracy Raw', 'Accuracy HQ', 'Accuracy LQ']
```

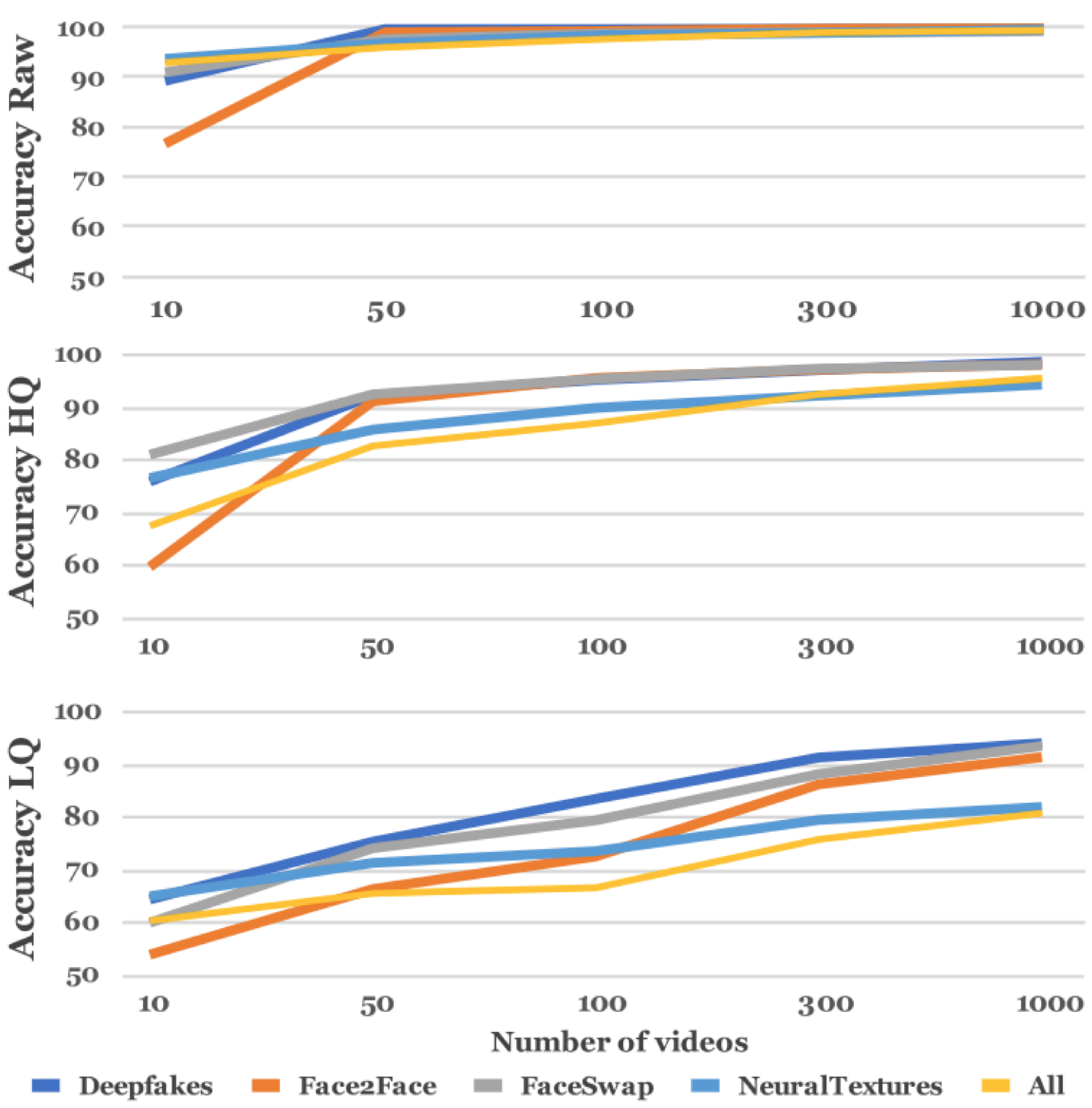
```
### font dict & color setting
label_dict = {'weight':'bold',
              'size':20}
tick_dict = {'weight':'semibold',
             'size':15}
color_list = ['royalblue', 'darkorange','gray',
              'cornflowerblue', 'gold']

### tick & ticklabel setting
x_ticks = [i for i in range(5)]
y_ticks = [i for i in range(50,101,10)]
x_ticklabels = [10, 50, 100, 300, 1000]
```

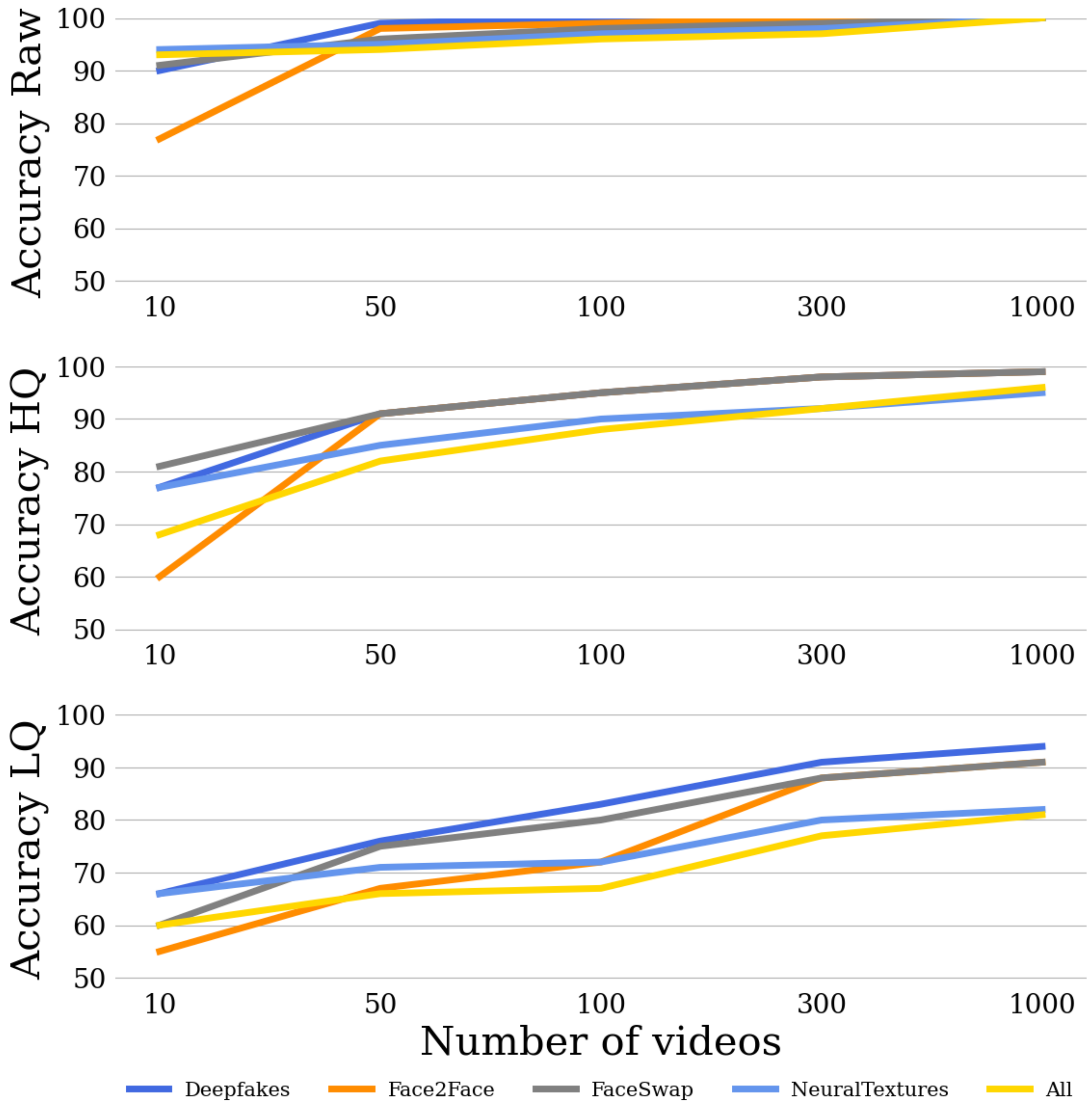
3. Exercise 02-03

```
fig, axes = plt.subplots(3,1,figsize = (13,13))
for ax_idx, ax in enumerate(axes.flat):
    ### axis limit customizing
    ax.set_ylim([49, 100])

    ### tick & ticklabel customizing
    ax.set_xticks(x_ticks)
    ax.set_xticklabels(x_ticklabels)
    ax.set_yticks(y_ticks)
    ax.set_ylabel(ylabel_list[ax_idx],
                  fontsize=30)
    ax.tick_params(labelsize=20,
                   left=False, bottom=False)

    ### plotting
    data = data_dict_list[ax_idx]
    for line_idx, (line_k, line_v) in enumerate(data.items()):
        ax.plot(x_ticks, line_v,
                color=color_list[line_idx],
                linewidth=5,
                label=line_name_list[line_idx])

    ### spine setting
    for spine_loc, spine in ax.spines.items():
        spine.set_visible(False)

    ### grid customizing
    ax.grid(axis='y')
    ax.grid(axis='x',
            linewidth=0)
```

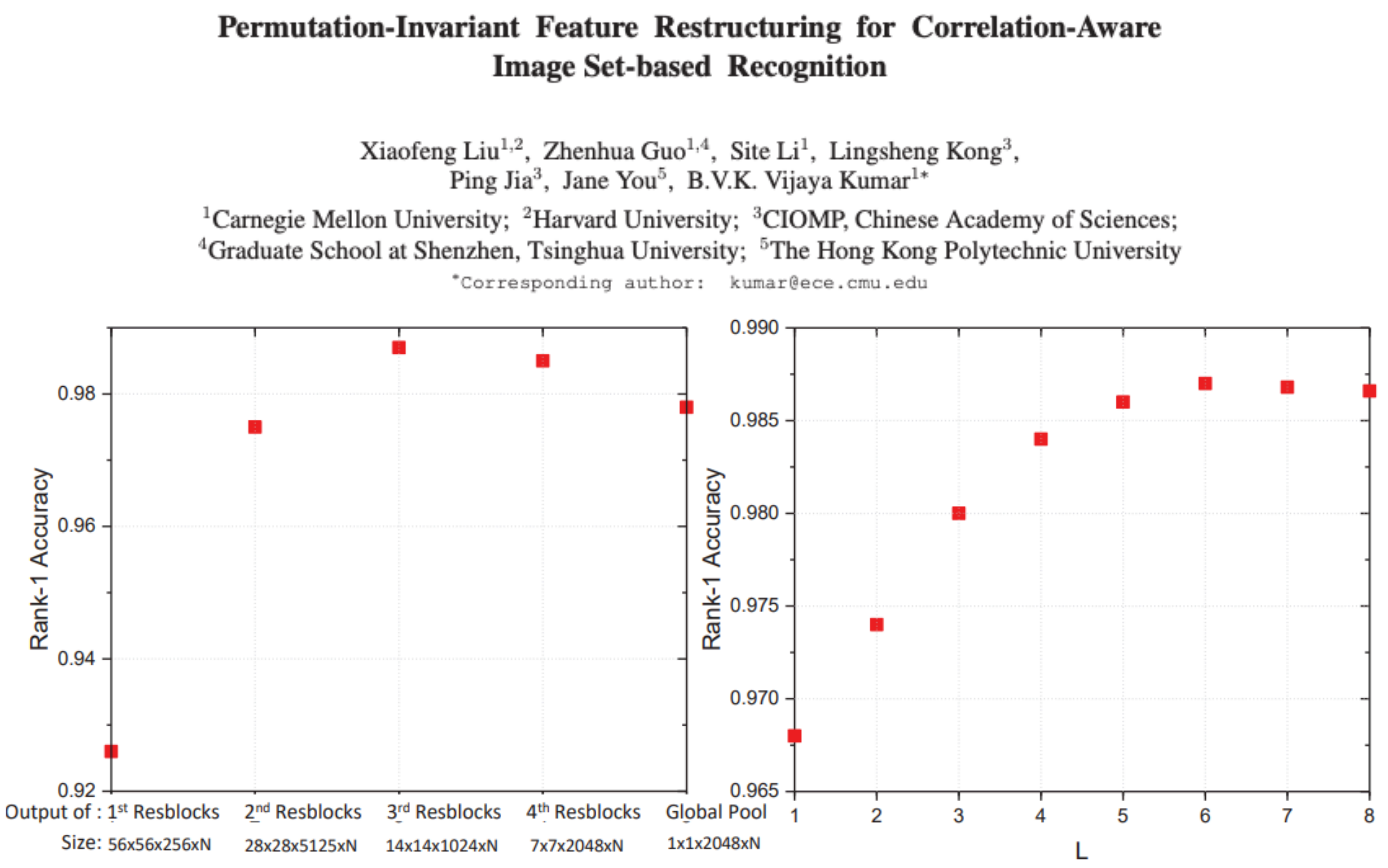
```
### label and legend customizing
axes[-1].set_xlabel('Number of videos',
                    fontsize=30)
axes[-1].legend(loc='upper center',
                bbox_to_anchor=(0.5, -0.3),
                fontsize=15,
                ncol=len(line_name_list),
                edgecolor='white')

### axis adjustment customizing
fig.subplots_adjust(hspace=0.3)
```

3. Exercise 02-03





4. Exercise 02-04



Permutation-Invariant Feature Restructuring for Correlation-Aware Image Set-based Recognition

Xiaofeng Liu[1,2], Zhenhua Guo[1,4], Site Li[1], Lingsheng Kong[3],
Ping Jia[3], Jane You[5], B.V.K. Vijaya Kumar[1*]

[1]Carnegie Mellon University; [2]Harvard University; [3]CIOMP, Chinese Academy of Sciences;
[4]Graduate School at Shenzhen, Tsinghua University; [5]The Hong Kong Polytechnic University

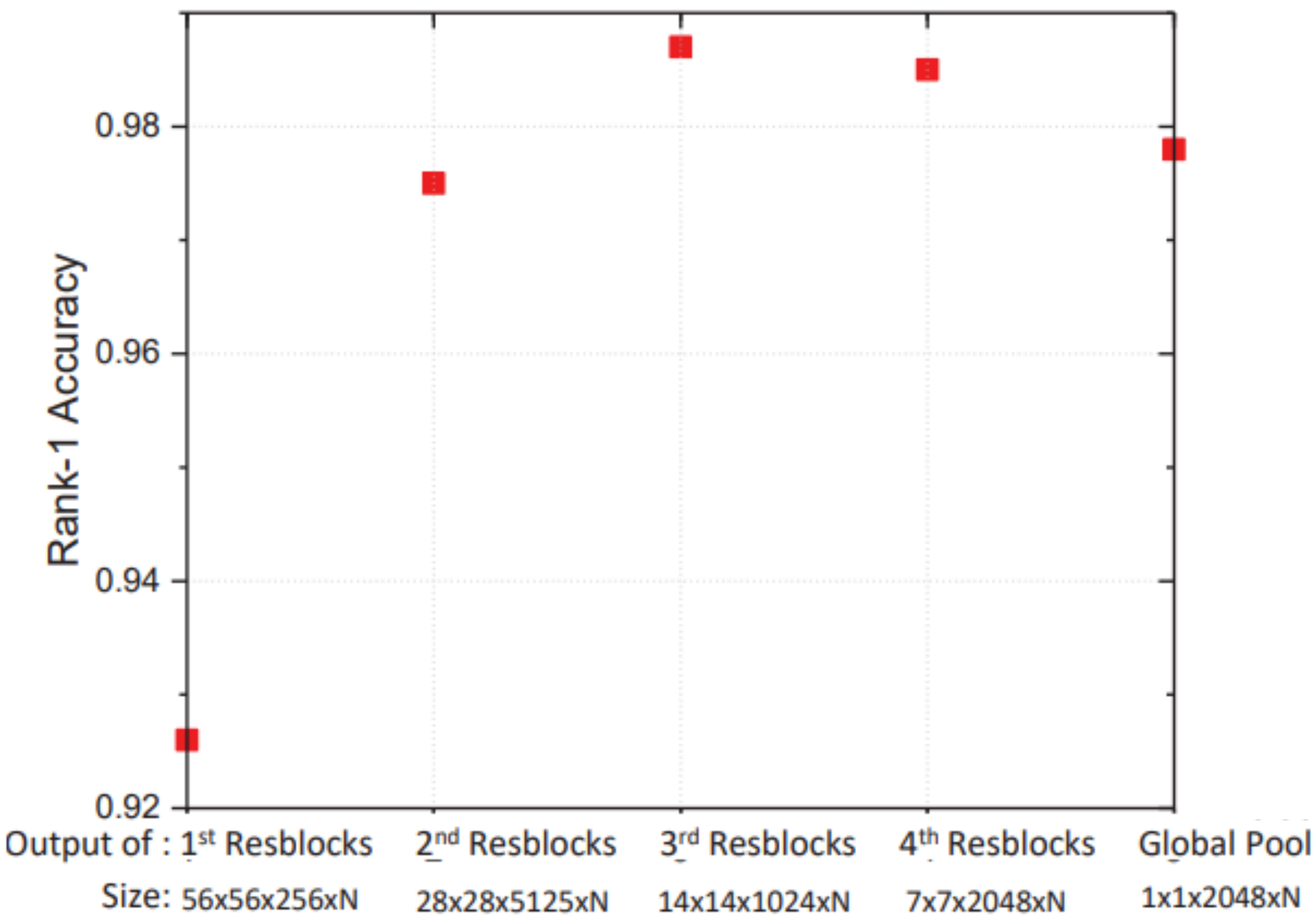
*Corresponding author: kumar@ece.cmu.edu

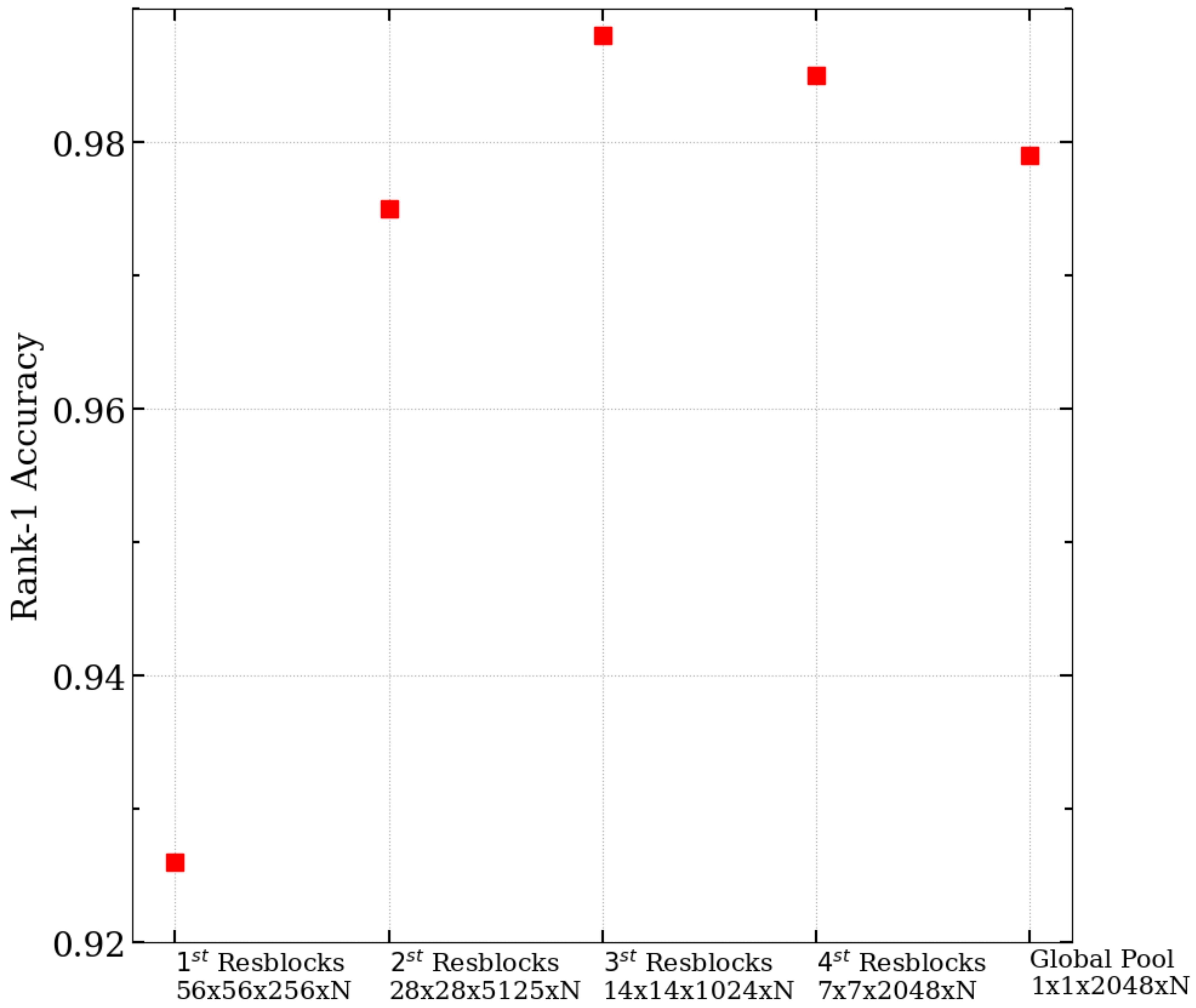
4. Exercise 02-04

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

### data setting
data1 = [0.926, 0.975, 0.988, 0.985, 0.979]

### plotting
fig, ax = plt.subplots(figsize=(10, 10))
ax.set_ylim([0.92, 0.99])
ax.plot(data1,'s',
        c='red',
        markersize=10)

### tick & ticklabel setting
xticks = [t for t in range(5)]
xtick_labels = [r'$1^{st}$ ' + 'Resblocks' + '\n56x56x256xN',
                r'$2^{st}$ ' + 'Resblocks' + '\n28x28x5125xN',
                r'$3^{st}$ ' + 'Resblocks' + '\n14x14x1024xN',
                r'$4^{st}$ ' + 'Resblocks' + '\n7x7x2048xN',
                'Global Pool' + '\n1x1x2048xN']
major_yticks = np.linspace(0.92, 0.98, 4)
minor_yticks = np.linspace(0.92, 0.99, 8)
```

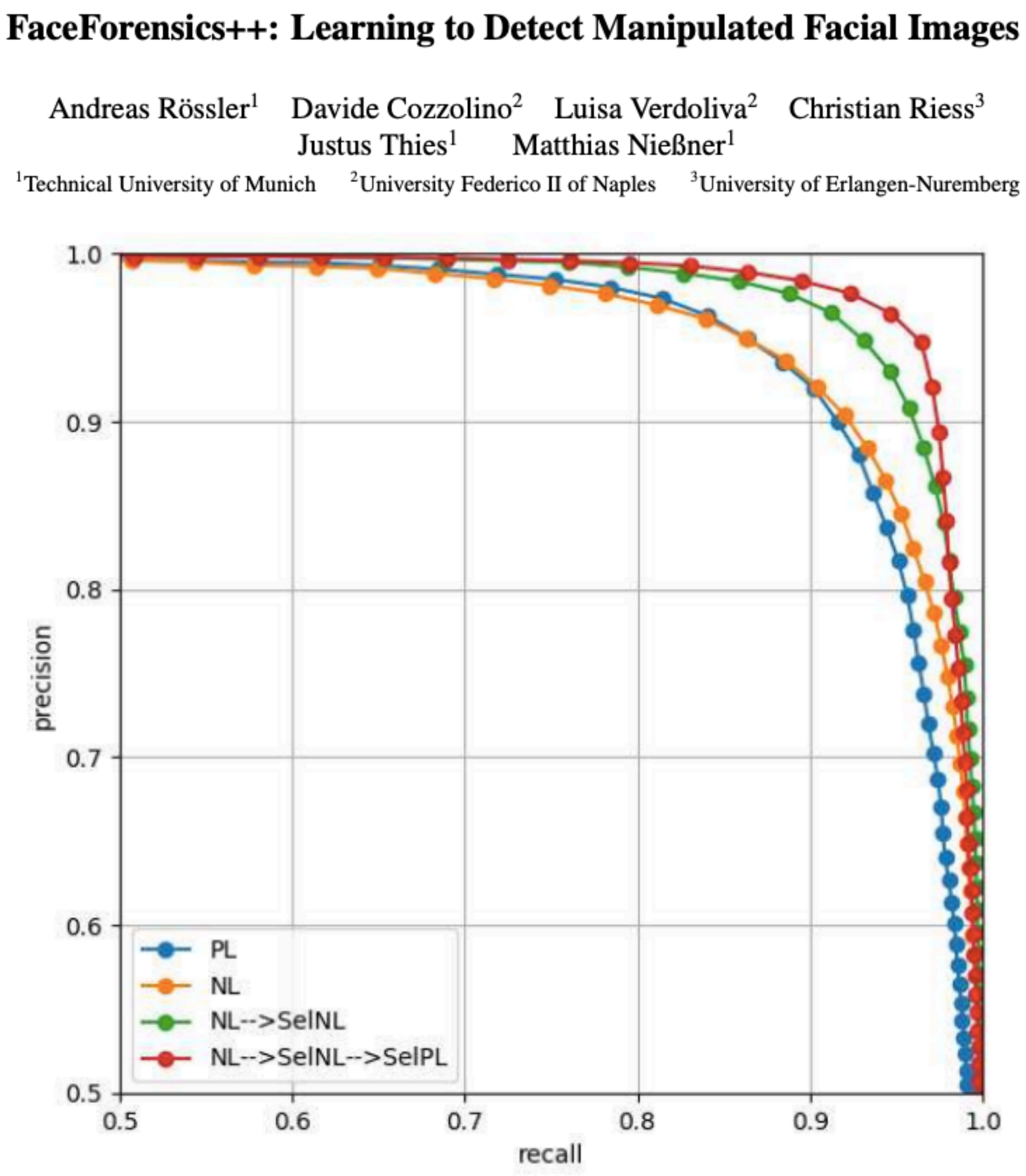
```python
### tick & ticklabel customizing
ax.set_xticks(xticks)
ax.set_xticklabels(xtick_labels,
                   ha='left')
ax.set_yticks(major_yticks)
ax.set_yticks(minor_yticks,
              minor=True)
ax.tick_params(axis='y',
               labelsize=20)
ax.tick_params(axis='x',
               labelsize=15)
ax.tick_params(which='major',
               length=7,
               width=1.5,
               direction='in',
               right=True, top=True)
ax.tick_params(which='minor',
               length=4,
               width=1.5,
               direction='in',
               right=True)

### grid and label customizing
ax.grid(linestyle=':')

ax.set_ylabel('Rank-1 Accuracy',
              fontsize=20)
```

4. Exercise 02-04





5. Exercise 02-05



FaceForensics++: Learning to Detect Manipulated Facial Images

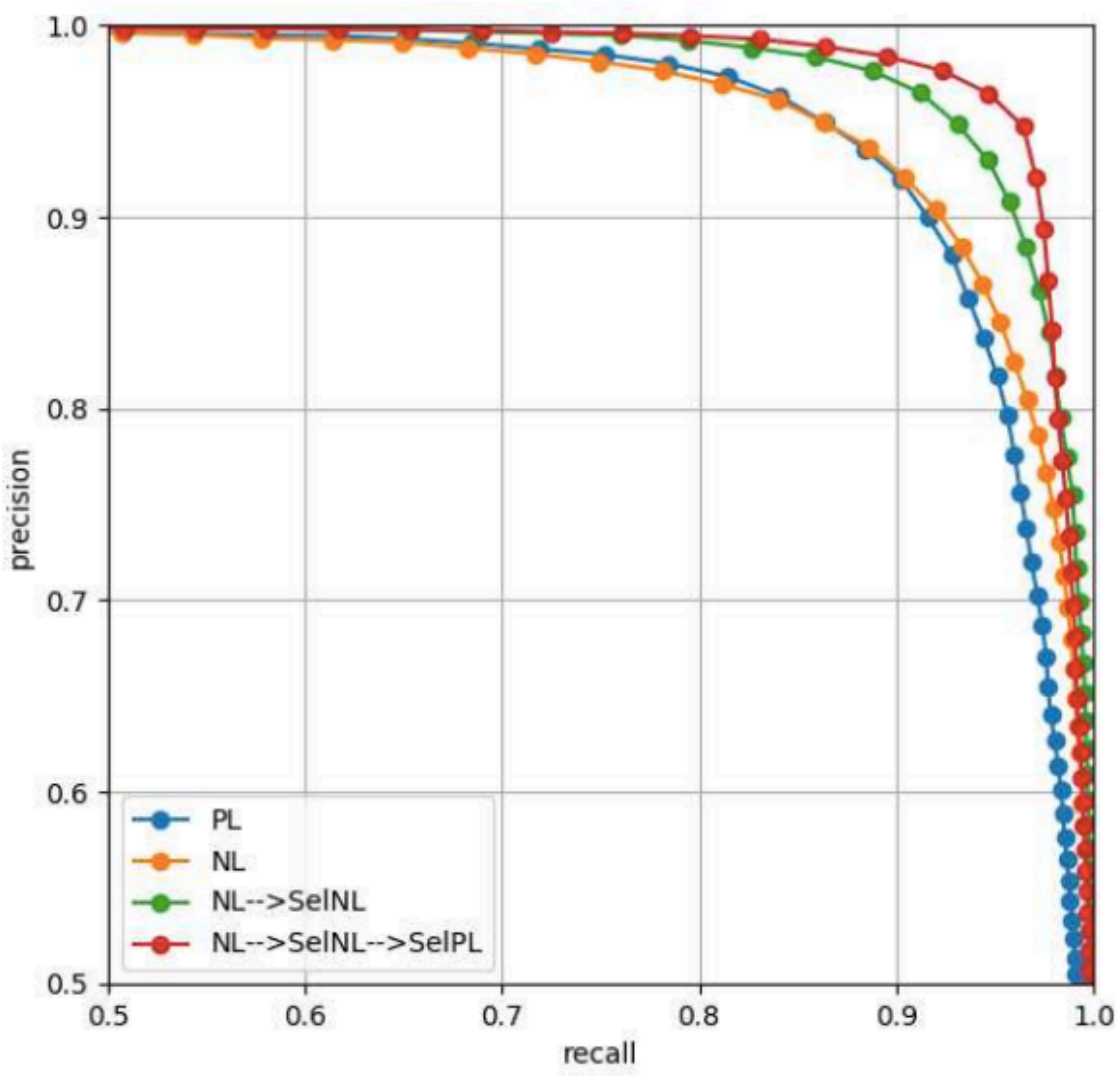
Andreas Rössler[1] Davide Cozzolino[2] Luisa Verdoliva[2] Christian Riess[3]
Justus Thies[1] Matthias Nießner[1]
[1]Technical University of Munich [2]University Federico II of Naples [3]University of Erlangen-Nuremberg

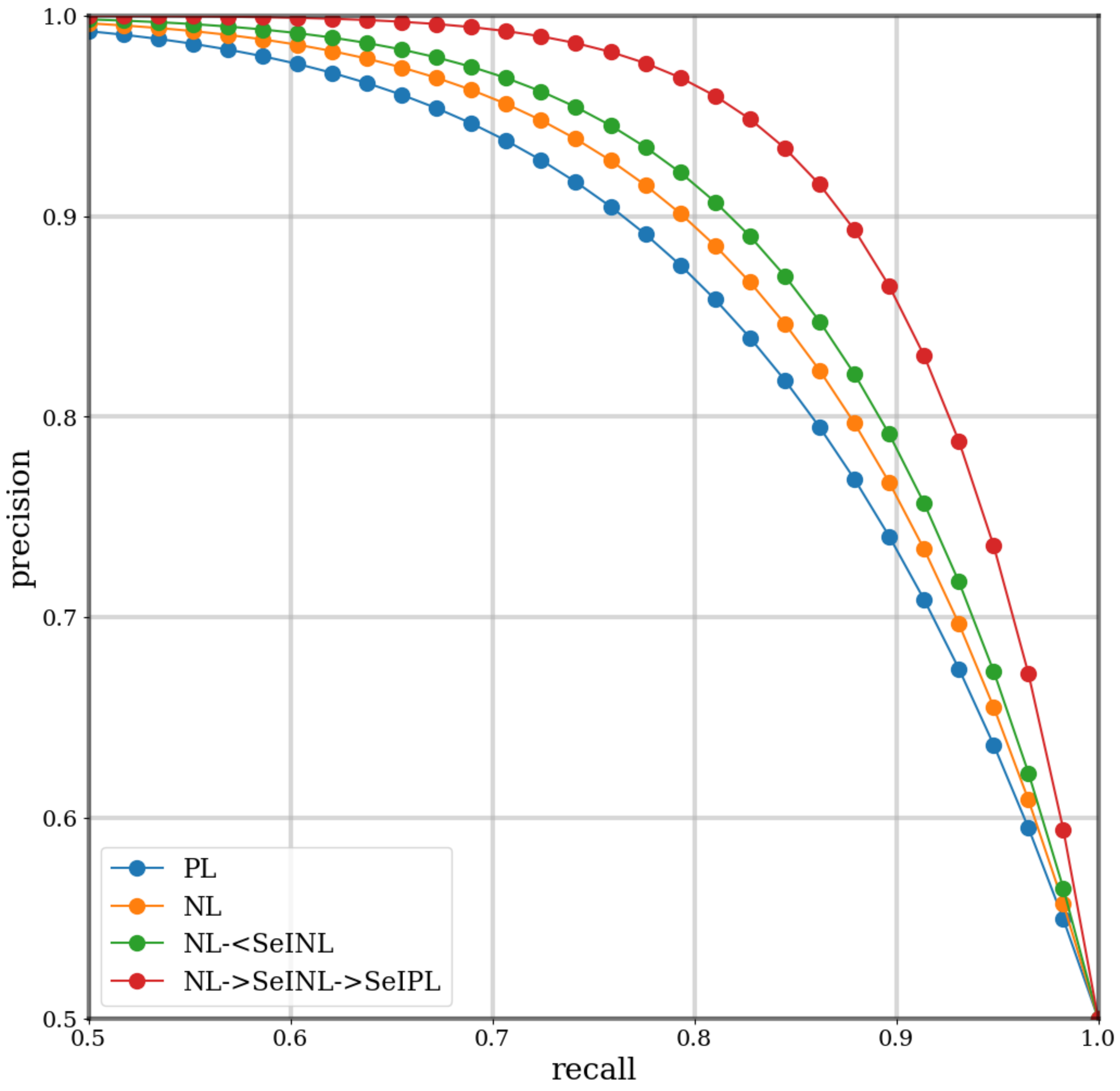
5. Exercise 02-05

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

### data setting
name_list = ['PL', 'NL', 'NL-<SeINL',
             'NL->SeINL->SeIPL']
x_data = np.linspace(0.5, 1, 30)
y_data = np.empty(shape = (0,len(x_data)))
gamma_list = [6,7, 8, 12]
for i in range(len(gamma_list)):
    y = (-0.5)*np.power(x_data, gamma_list[i]) + 1
    y = y.reshape(1,len(x_data))

    y_data = np.vstack((y_data, y))

### plotting
fig, ax = plt.subplots(figsize = (12,12))

for i in range(len(name_list)):
    ax.plot(x_data, y_data[i],
            label = name_list[i],
            marker = 'o',
            markersize = 10)

### spine customizing
for spine_idx, spine_loc in enumerate(ax.spines):
    ax.spines[spine_loc].set_linewidth(3)
    ax.spines[spine_loc].set_alpha(0.5)
```

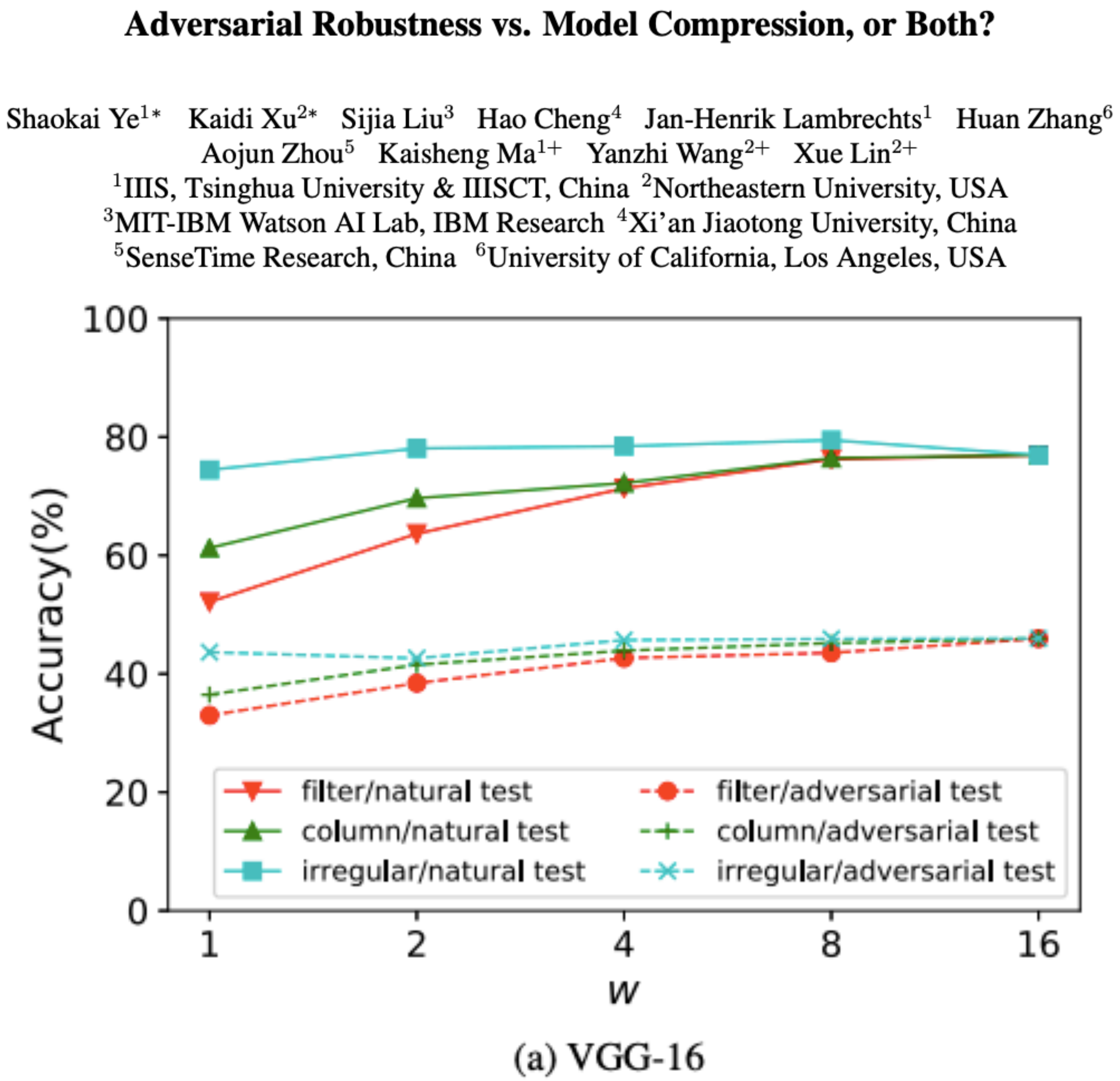
```python
### axis limit customizing
ax.set_xlim([0.5, 1.0])
ax.set_ylim([0.5, 1.0])

### legend customizing
ax.legend(loc = 'lower left',
          fontsize = 17)

### tick & ticklabel customizing
tick_dict = {'size':15}
ticks = [round(0.1*i,1) for i in range(5, 11)]

ax.set_xticks(ticks)
ax.set_xticklabels(ticks,
                   fontdict = tick_dict)
ax.set_yticks(ticks)
ax.set_yticklabels(ticks,
                   fontdict = tick_dict)

### label customizing
label_dict = {'size':20}
ax.set_xlabel("recall", fontdict = label_dict)
ax.set_ylabel("precision", fontdict = label_dict)

### grid customizing
ax.grid(linewidth = 3,
        alpha = 0.5)
```

5. Exercise 02-05





6. Exercise 02-06



Adversarial Robustness vs. Model Compression, or Both?

Shaokai Ye[1*] Kaidi Xu[2*] Sijia Liu[3] Hao Cheng[4] Jan-Henrik Lambrechts[1] Huan Zhang[6]
Aojun Zhou[5] Kaisheng Ma[1+] Yanzhi Wang[2+] Xue Lin[2+]
[1]IIIS, Tsinghua University & IIISCT, China [2]Northeastern University, USA
[3]MIT-IBM Watson AI Lab, IBM Research [4]Xi'an Jiaotong University, China
[5]SenseTime Research, China [6]University of California, Los Angeles, USA

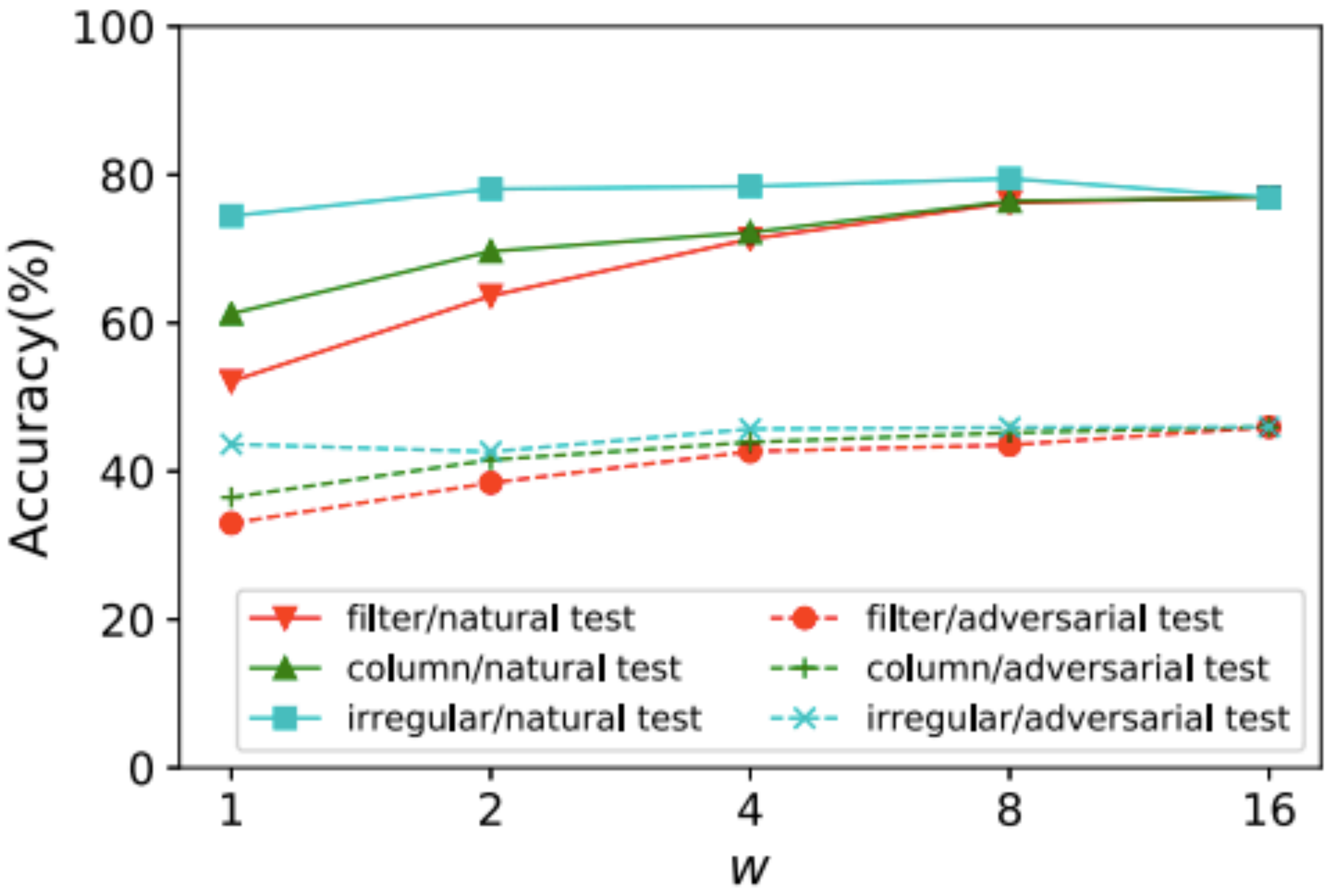
(a) VGG-16

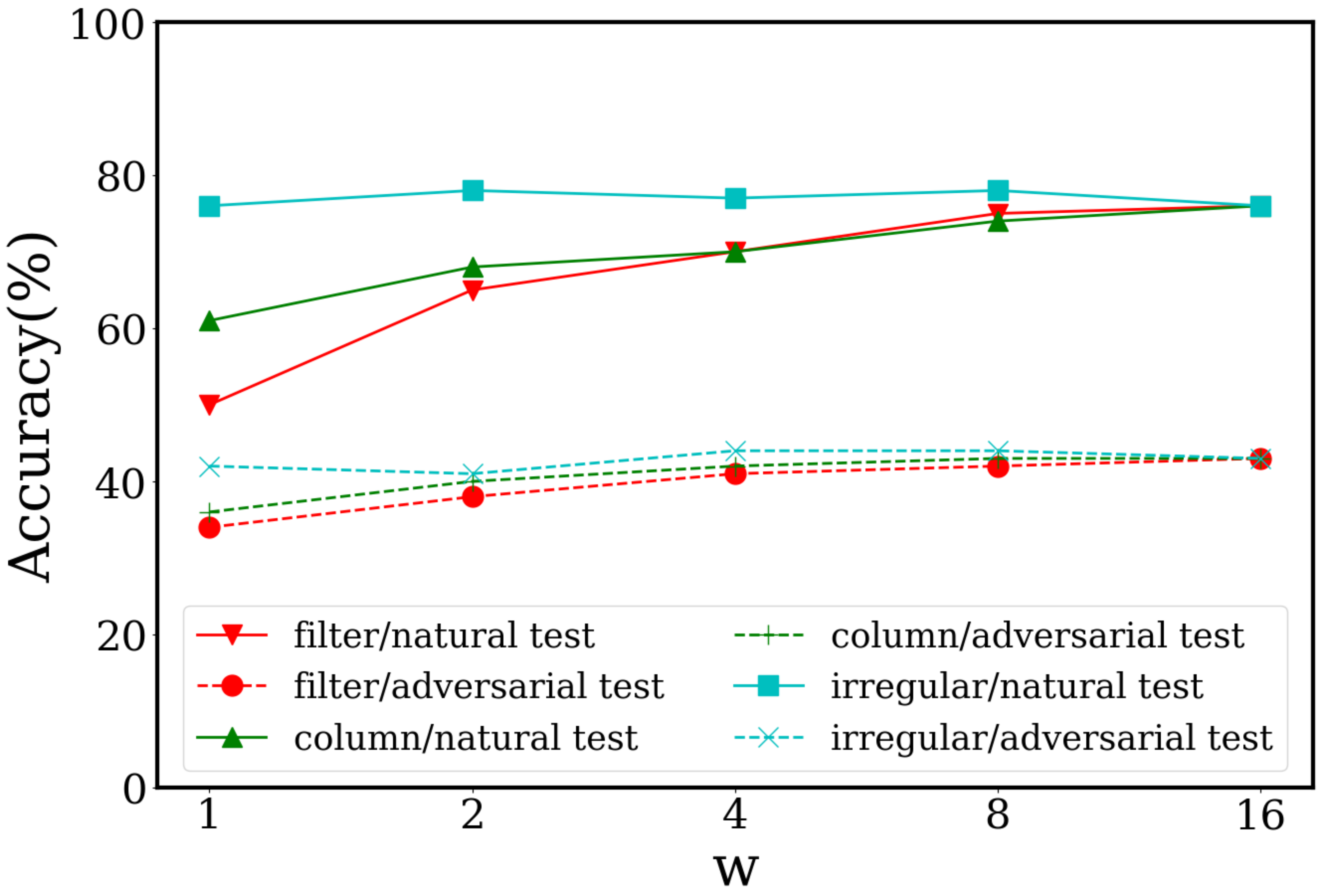
6. Exercise 02-06

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

### data setting
name_list = ['filter/natural test', 'filter/adversarial test',
             'column/natural test', 'column/adversarial test',
             'irregular/natural test', 'irregular/adversarial test']
x_loc = [2**i for i in range(5)]
data_dict = {'FN':[50, 65, 70, 75, 76],
             'FA':[34, 38, 41, 42, 43],
             'CN':[61, 68, 70, 74, 76],
             'CA':[36, 40, 42, 43, 43],
             'IN':[76, 78, 77, 78, 76],
             'IA':[42, 41, 44, 44, 43]}

### line setting
color_list = ['r', 'r', 'g', 'g', 'c', 'c']
marker_list = ['v', 'o', '^', '+', 's', 'x']
linestyle_list = ['-', '--', '-', '--', '-', '--']

### plotting
fig, ax = plt.subplots(figsize = (15,10))
ax.set_xscale('log',  basex=2)
for data_idx, ex_name in enumerate(data_dict):
    ax.plot(x_loc, data_dict[ex_name],
            color = color_list[data_idx],
            linewidth = 2,
            marker = marker_list[data_idx],
            markersize = 15,
            linestyle = linestyle_list[data_idx],
            label = name_list[data_idx])
```

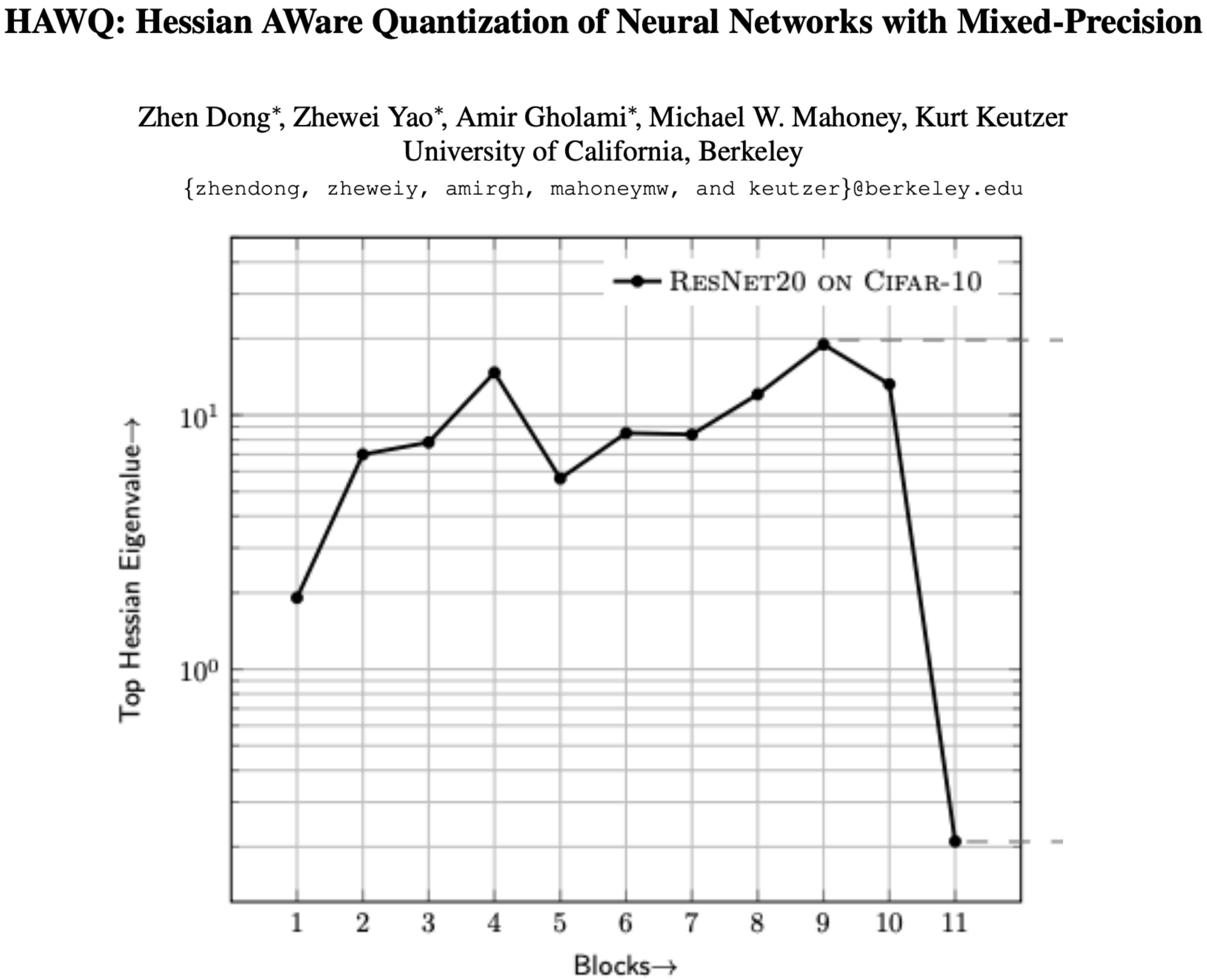
```python
### legend customizing
ax.legend(fontsize = 25,
          ncol = 2,
          loc = 'lower center',
          bbox_to_anchor = (0.5, 0))

### spine customizing
for spine_idx, spine_key in enumerate(ax.spines):
    ax.spines[spine_key].set_linewidth(3)

### tick & ticklabel customizing
x_ticks = [2**i for i in range(5)]
x_ticklabels = x_ticks
y_ticks = [20*i for i in range(6)]
y_ticklabels = y_ticks

ax.set_xticks(x_ticks)
ax.set_xticklabels(x_ticklabels,
                   size = 30)
ax.set_yticks(y_ticks)
ax.set_yticklabels(y_ticklabels,
                   size = 30)

### axis limit customizing
ax.set_ylim([0, 100])

### label customizing
label_dict = {'size':40}
ax.set_xlabel('w', fontdict = label_dict)
ax.set_ylabel('Accuracy(%)', fontdict = label_dict)
```

6. Exercise 02-06





7. Exercise 02-07



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

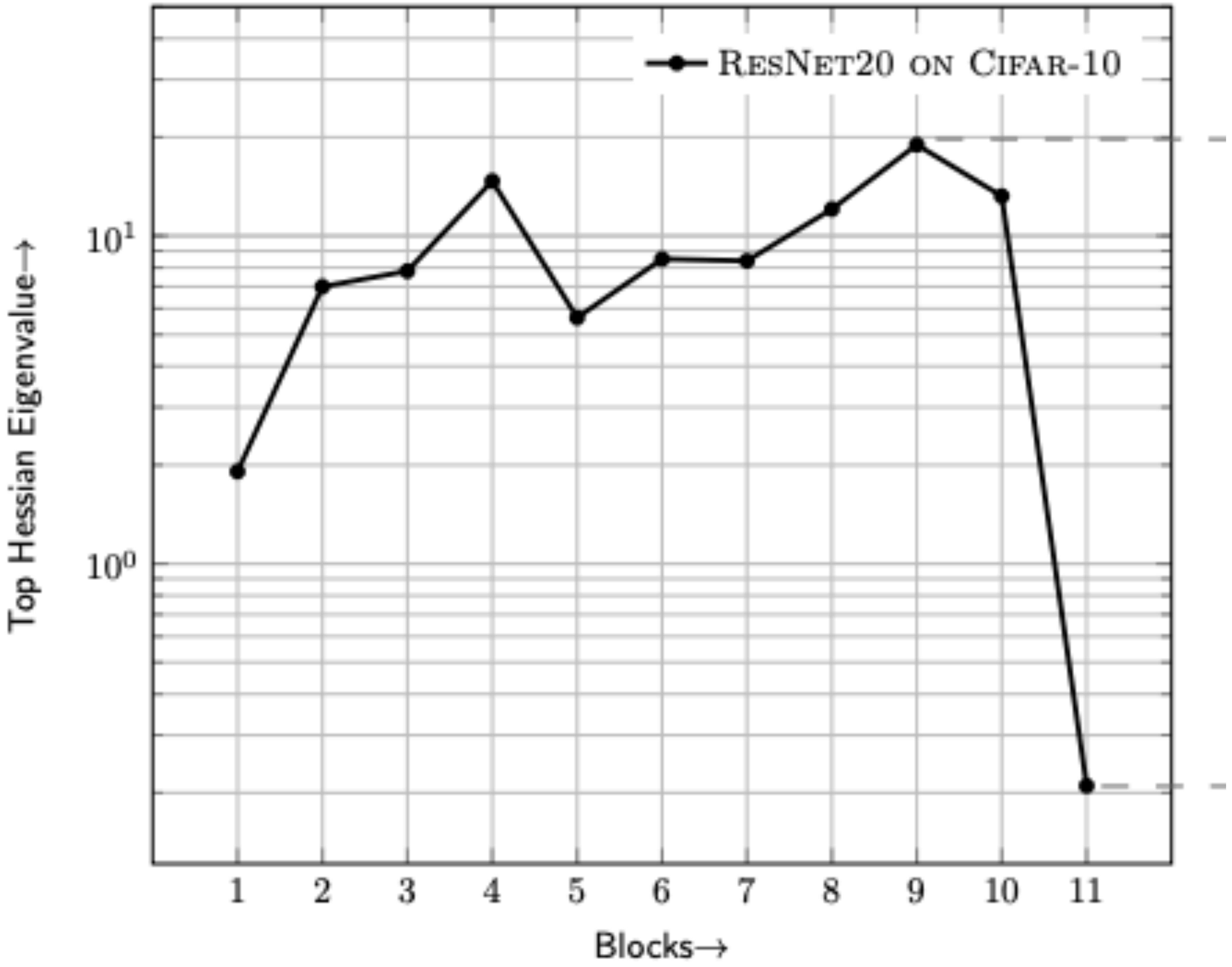
Zhen Dong*, Zhewei Yao*, Amir Gholami*, Michael W. Mahoney, Kurt Keutzer
University of California, Berkeley
{zhendong, zheweiy, amirgh, mahoneymw, and keutzer}@berkeley.edu

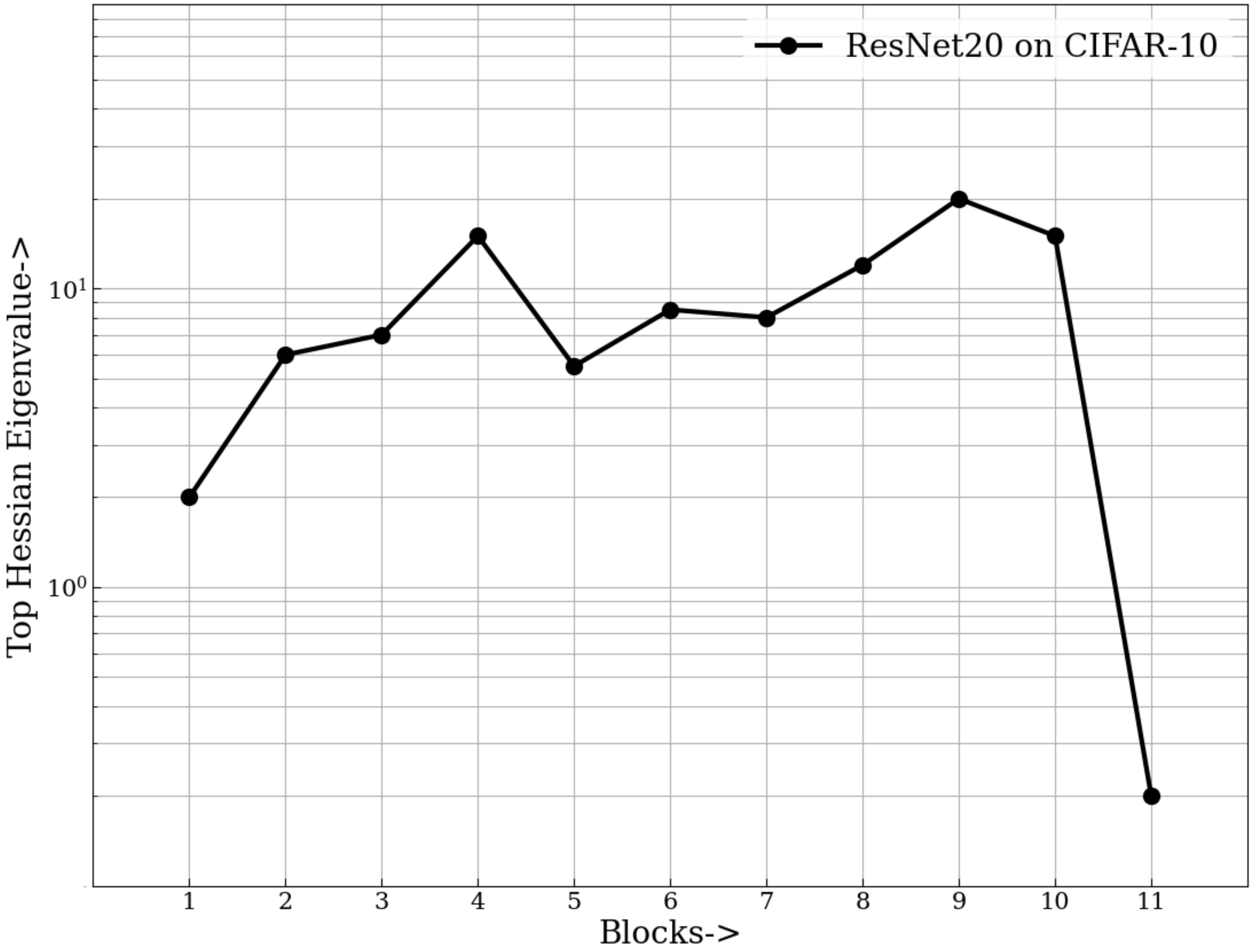
7. Exercise 02-07

```python
import matplotlib.pyplot as plt

### data setting
data = [2, 6, 7, 15, 5.5, 8.5, 8, 12, 20, 15, 0.2]

fig, ax = plt.subplots(figsize=(13, 10))
### axis customizing
ax.set_yscale('log')
ax.set_xlim([0, 12])
ax.set_ylim([0.1, 50])

### tick & ticklabel customizing
major_yticks = [1, 10]
minor_yticks = [j*10**i for i in range(-1, -1+3) for j in range(1, 1+9)]
major_xticks = [i for i in range(1, 12)]

ax.set_yticks(major_yticks)
ax.set_yticks(minor_yticks,
              minor=True)
ax.set_xticks(major_xticks)

ax.tick_params(labelsize=15,
               direction='in',
               length=5)
ax.tick_params(which='minor',
               labelsize=0,
               direction='in',
               length=3)
```

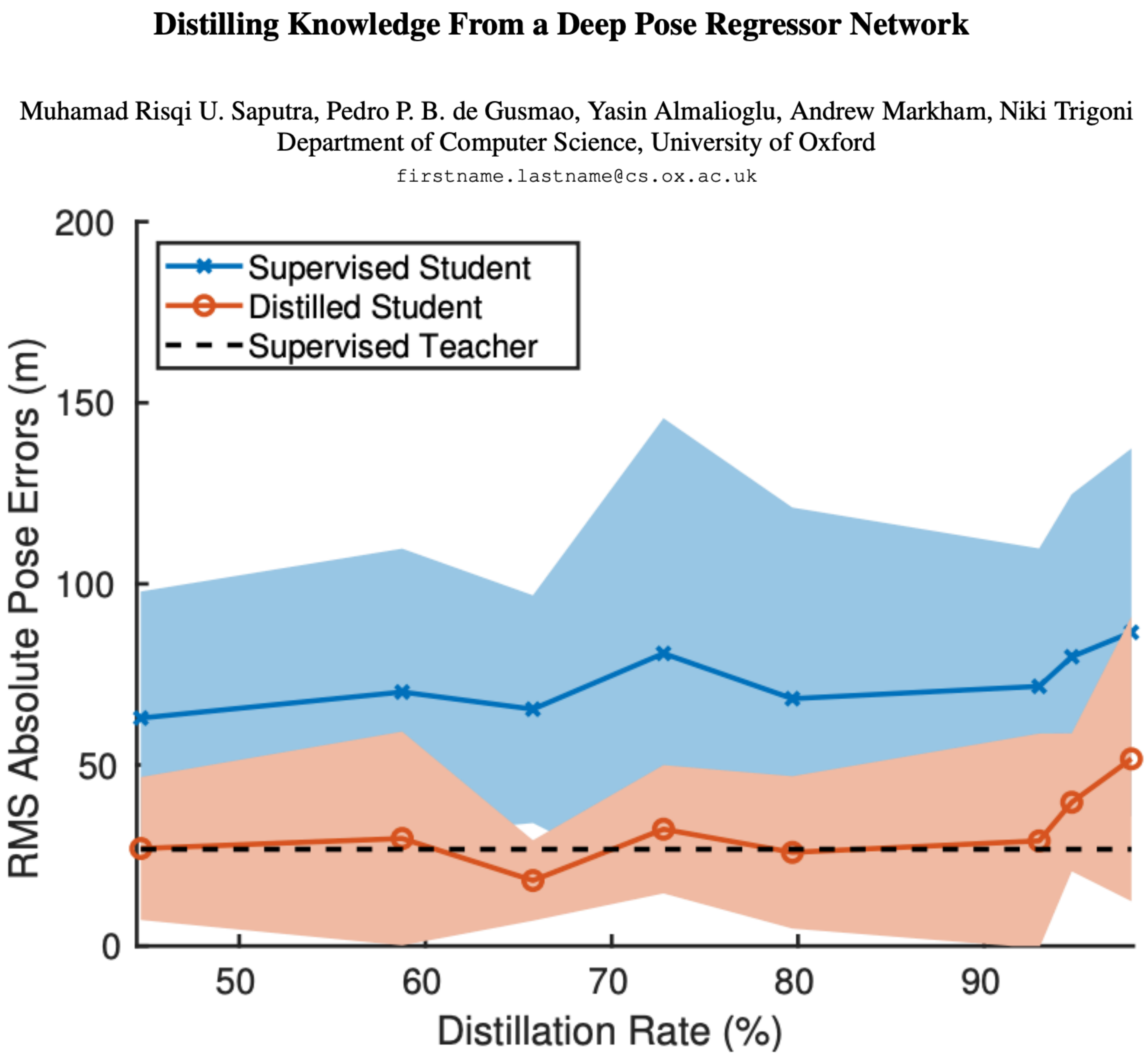
```python
### grid customizing
ax.grid(which='both')

### plotting
ax.plot(major_xticks, data,
        color='black',
        marker='o',
        label='ResNet20 on CIFAR-10',
        linewidth=3,
        markersize=10)

### legend customizing
ax.legend(loc='upper right',
          fontsize=20,
          edgecolor='white',
          facecolor='white')

### label customizing
ax.set_xlabel('Blocks->',
              fontsize=20)
ax.set_ylabel('Top Hessian Eigenvalue->',
              fontsize=20)
```

7. Exercise 02-07





8. Exercise 02-08



Distilling Knowledge From a Deep Pose Regressor Network

Muhamad Risqi U. Saputra, Pedro P. B. de Gusmao, Yasin Almalioglu, Andrew Markham, Niki Trigoni
Department of Computer Science, University of Oxford
firstname.lastname@cs.ox.ac.uk

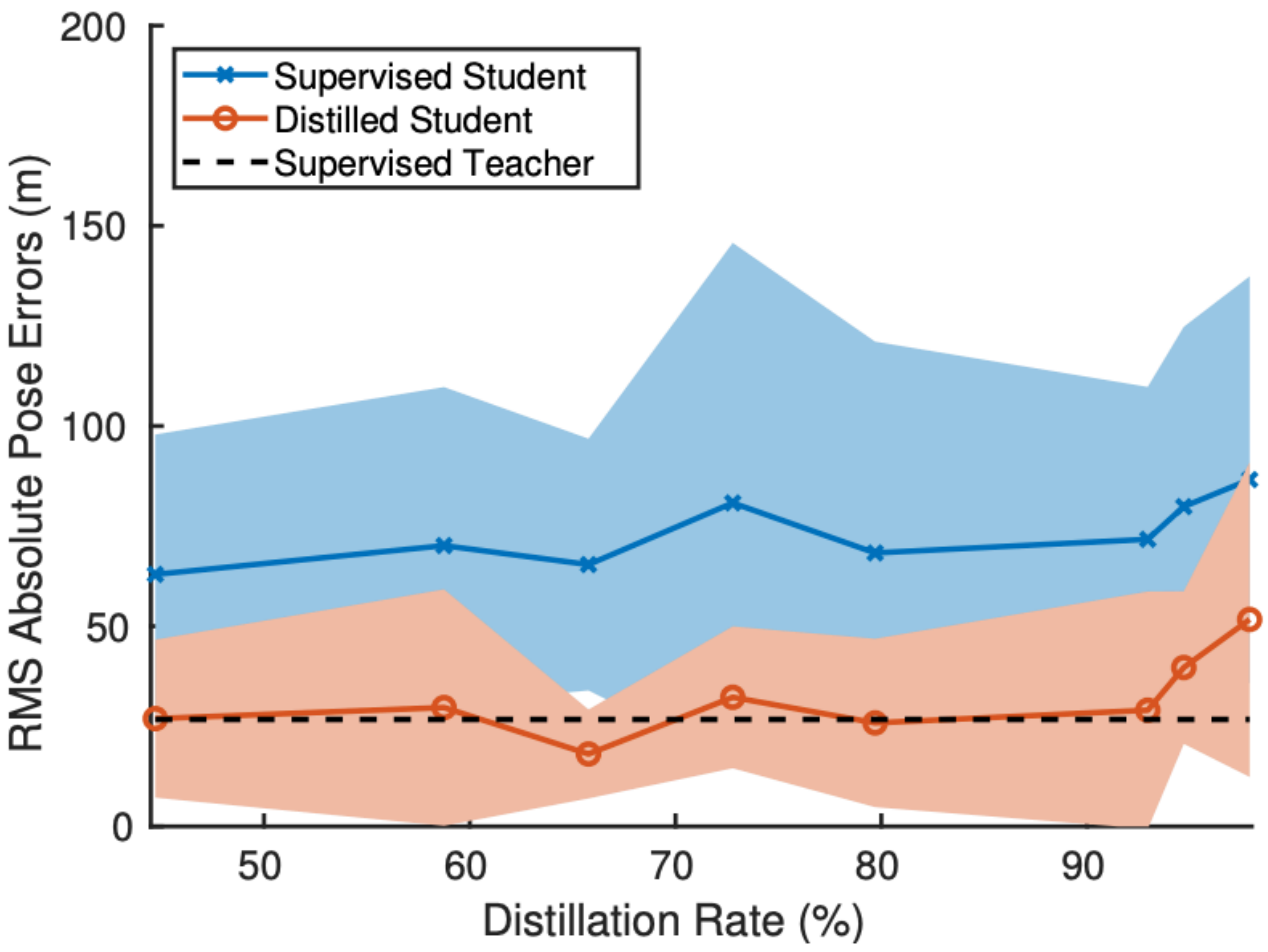
8. Exercise 02-08

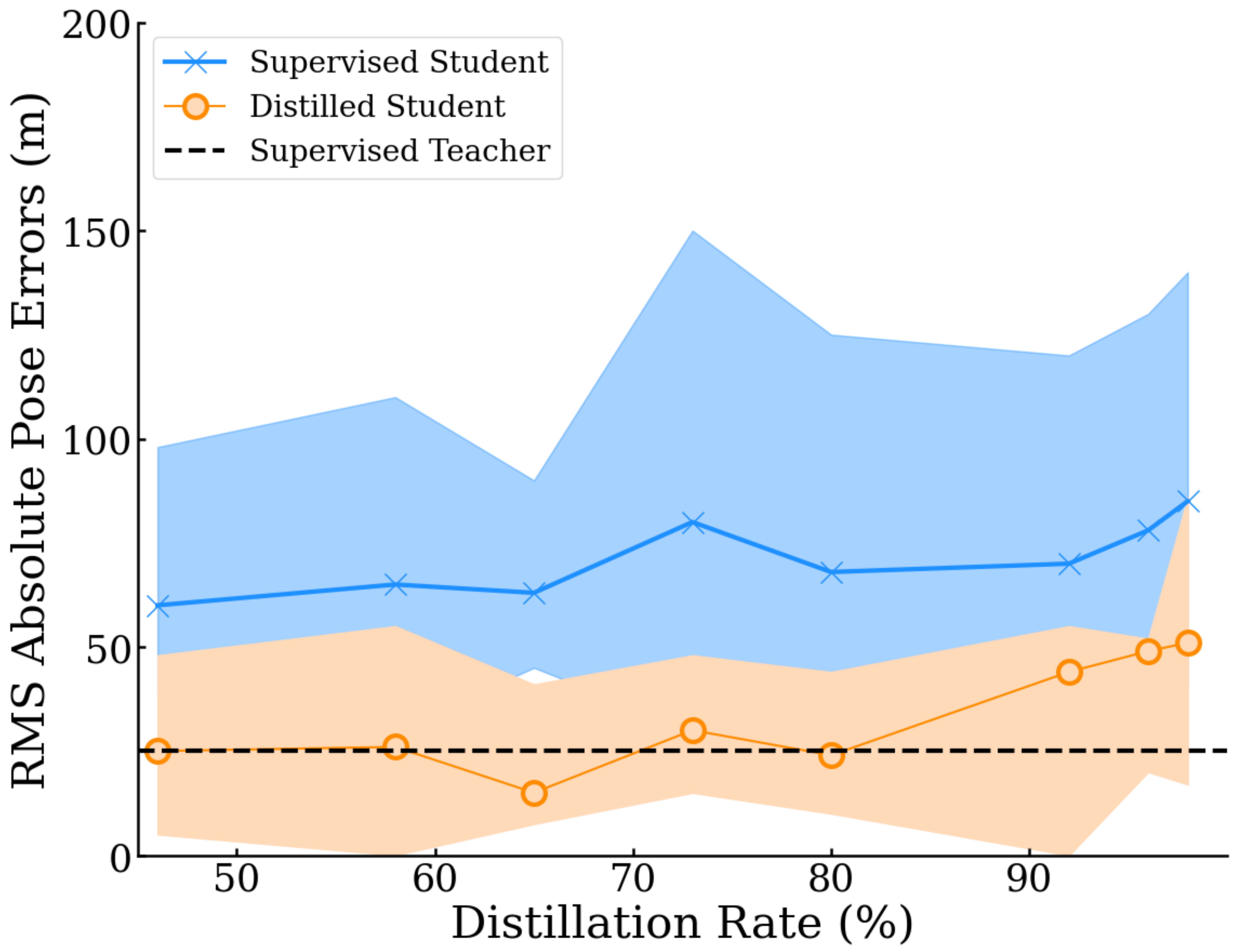
```
import matplotlib.pyplot as plt

### data setting
data_loc = [46, 58, 65, 73, 80, 92, 96, 98]
supervised_data = [60, 65, 63, 80, 68, 70, 78, 85]
supervised_top_data = [98, 110, 90, 150, 125, 120, 130, 140]
supervised_bottom_data = [40, 30, 45, 30, 40, 40, 40, 40]

distilled_data = [25, 26, 15, 30, 24, 44, 49, 51]
distilled_top_data = [48, 55, 41, 48, 44, 55, 52, 86]
distilled_bottom_data = [5, 0, 7.5, 15, 10, 0, 20, 17]
```

```
fig, ax = plt.subplots(figsize=(13, 10))
### plotting
ax.fill_between(data_loc,
                y1=supervised_top_data,
                y2=supervised_bottom_data,
                color='dodgerblue',
                alpha=0.4)

ax.fill_between(data_loc,
                y1=distilled_top_data,
                y2=distilled_bottom_data,
                color='peachpuff',
                alpha=1)

ax.plot(data_loc, supervised_data,
        marker='x',
        color='dodgerblue',
        linewidth=3,
        markersize=15,
        label='Supervised Student')
ax.plot(data_loc, distilled_data,
        marker='o',
        markersize=15,
        markerfacecolor='peachpuff',
        markeredgewidth=3,
        color='darkorange',
        label='Distilled Student')

ax.axhline(y=25,
           color='black',
           linestyle='--',
           linewidth=3,
           label='Supervised Teacher')
```

```
### axis limit setting
ax.set_xlim([45, 100])
ax.set_ylim([0, 200])

### spine customizing
for spine_loc, spine in ax.spines.items():
    if spine_loc in ['right', 'top']:
        spine.set_visible(False)

    if spine_loc in ['left', 'bottom']:
        spine.set_linewidth(1.8)

### tick & ticklabel setting
x_ticks = np.arange(50, 91, 10)
y_ticks = np.arange(0, 201, 50)
ax.set_xticks(x_ticks)
ax.set_yticks(y_ticks)
ax.tick_params(direction='in',
               length=5,
               width=1.8,
               labelsize=25)

### legend customizing
ax.legend(loc='upper left',
          fontsize=20)

### label customizing
label_font_dict = {'size':30}
ax.set_xlabel("Distillation Rate (%)",
              fontdict=label_font_dict)
ax.set_ylabel("RMS Absolute Pose Errors (m)",
              fontdict=label_font_dict)
```

8. Exercise 02-08





Python for Data Visualization

2-05. Exercises

1. Exercise 02-01
2. Exercise 02-02
3. Exercise 02-03
4. Exercise 02-04
5. Exercise 02-05
6. Exercise 02-06
7. Exercise 02-07
8. Exercise 02-08