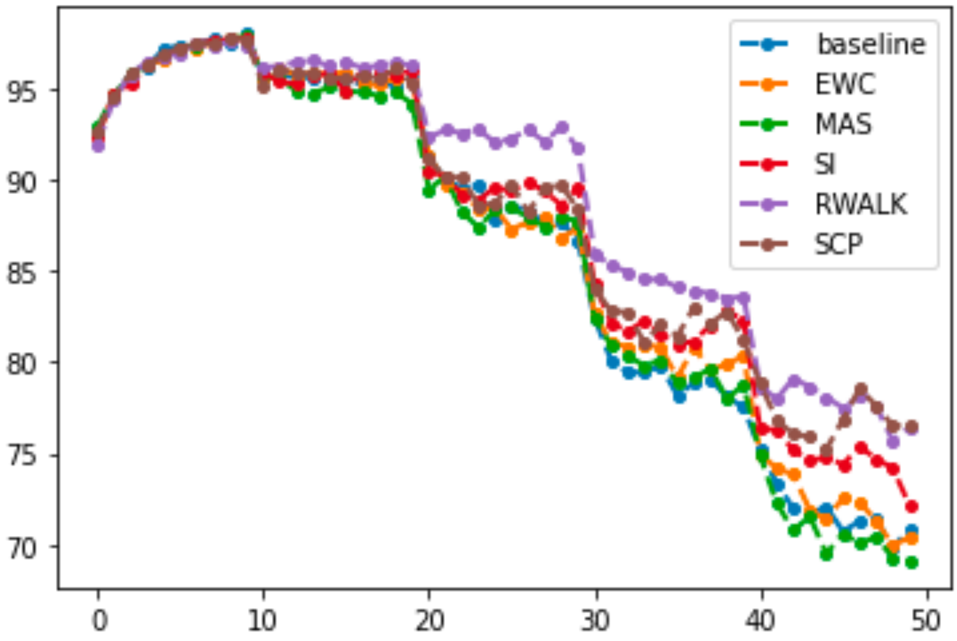
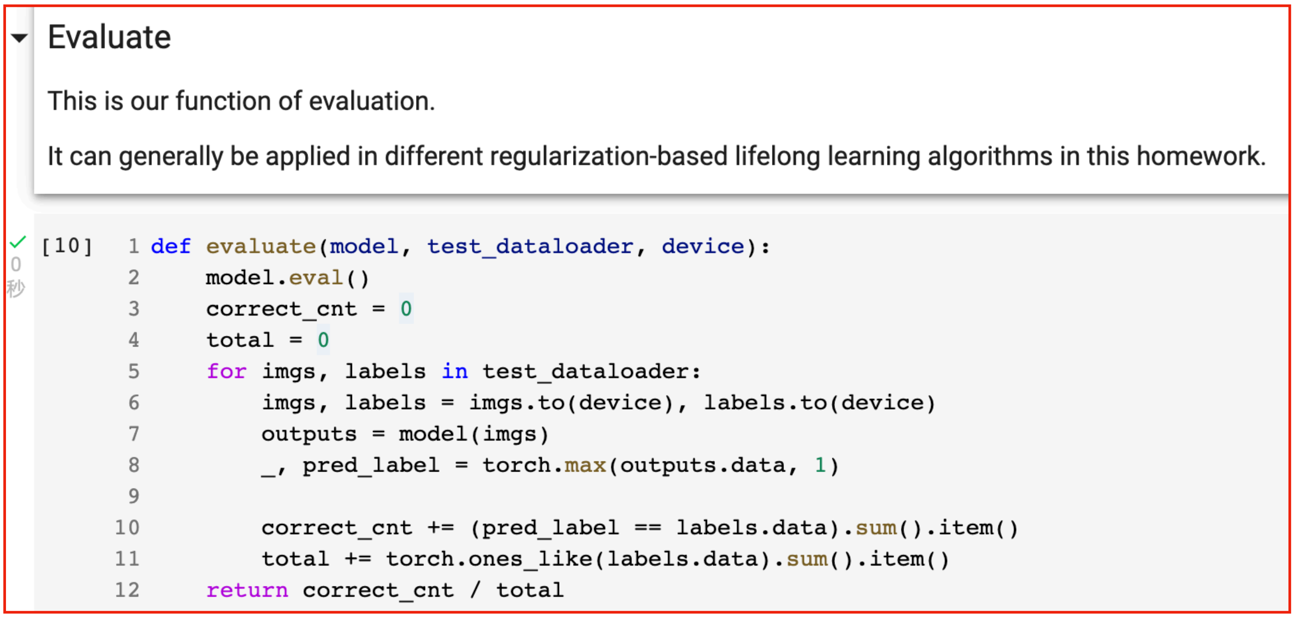
Q1: Plot the learning curve of the metric with every method. (The Plotting function is provided in the sample code.)



Q2: Describe the metric.



Evaluate

This is our function of evaluation.

It can generally be applied in different regularization-based lifelong learning algorithms in this homework.

```
[10]  1 def evaluate(model, test_dataloader, device):
      2     model.eval()
      3     correct_cnt = 0
      4     total = 0
      5     for imgs, labels in test_dataloader:
      6         imgs, labels = imgs.to(device), labels.to(device)
      7         outputs = model(imgs)
      8         _, pred_label = torch.max(outputs.data, 1)
      9 
     10         correct_cnt += (pred_label == labels.data).sum().item()
     11         total += torch.ones_like(labels.data).sum().item()
     12     return correct_cnt / total
```

The screenshot is the evaluation step with special metrics. From line 9, it takes the max of each data; As a result, we know that “L–inf norm” is used in this part.
Ans: special metric = L–infinity norm

Q3: Paste the code that you implement Omega Matrix for MAS.
Ans:

```
    output = torch.pow(output, 2)
    loss = torch.sum(output, dim=1)
    loss = loss.mean()
    loss.backward()
```