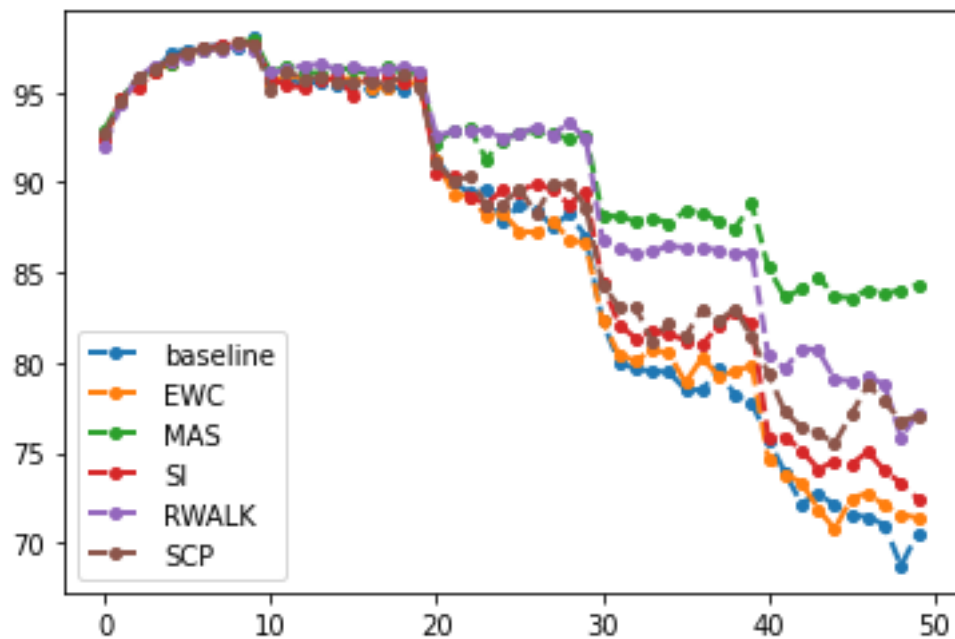


## Question 1

1-1 Plot the learning curve of the metric with every method. (The Plotting function is provided in the sample code.) (0.5pt)



1-2 Describe the metric. (0.5pt)

```
def evaluate(model, test_dataloader, device):
    model.eval()
    correct_cnt = 0
    total = 0
    for imgs, labels in test_dataloader:
        imgs, labels = imgs.to(device), labels.to(device)
        outputs = model(imgs)
        _, pred_label = torch.max(outputs.data, 1)

        correct_cnt += (pred_label == labels.data).sum().item()
        total += torch.ones_like(labels.data).sum().item()
    return correct_cnt / total
```

根據 sample code 中的 evaluate，這邊用來評估的方式是：

若預測出的 label(pred\_label)與正確的 label 相符合，也就是預測正確的數，除上總共有幾個 label，是很直觀的求正確率。

## Question 2

Paste the code that you implement Omega Matrix for MAS. (1pt)

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