

Figure Explanation

依照圖表在論文內的章節分類

[Section 5.2：PCME單機版](#)

[Section 5.3：PCME FL版](#)

[Section 5.4：PCME在其他FL環境](#)

[Section 5.5：Compare w/ Non-opp. Training](#)

[Section 5.6：PCME w/ FreezeOut](#)

Section 5.2：PCME單機版

- 5.1：LeNet-5，在單機版本的訓練中，accuracy隨著訓練回合的改變
 - $I=0.1$ ：在訓練進行到總回合數的前10%時，就開始嘗試做freezing
 - $I=0.25$ ：在訓練進行到總回合數的前25%時，就開始嘗試做freezing
- 5.2：MobileNetV2，在單機版本的訓練中，accuracy隨著訓練回合的改變
- 5.3：ResNet-18，在單機版本的訓練中，accuracy隨著訓練回合的改變
- 5.4：三種模型在單機版本的訓練中，所需的總傳輸量
- 5.5：三種模型在單機版本的訓練中，所需的總訓練時間

Section 5.3：PCME FL版

- 5.6：兩種模型在FL的訓練環境裡，當只選一半的worker作為participants時 ($S=0.5$)，accuracy隨著訓練回合的改變
- 5.7：三種模型在FL的訓練環境裡，當全選worker作為participants時 ($S=1$)，accuracy隨著訓練回合的改變
- 5.8：三種模型在FL版本的訓練中，所需的總傳輸量
- 5.9：三種模型在FL版本的訓練中，所需的總訓練時間
- 5.10：兩種模型在FL版本的訓練過程中，到達各目標accuracy所需的傳輸量
 - 這很重要，因為我們發現只考慮總傳輸量會忽略了early stop的可能性
 - e.g. baseline 總傳輸量5G，可達到final acc. 0.85

- e.g. freeze 總傳輸量3G，可達到final acc. 0.8
 - ⇒ 乍看之下Freeze省了2G，也只降低5% acc
 - ⇒ 但忽略了訓練到一半時，說不定baseline只花2G就達到acc=0.8了
 - ⇒ 這樣反而做freeze還比較爛
- 5.11：兩種模型在FL版本的訓練中，每個回合所需的傳輸量

Section 5.4：PCME在其他FL環境

- 5.12：兩種模型在non-iid 的FL訓練中，accuracy隨著訓練回合的改變
- 5.13：兩種模型在non-iid 的FL訓練中，所需的總傳輸量
- 5.14：兩種模型在non-iid 的FL訓練中，所需的總訓練時間
- 5.15：在不同的上傳頻率裡，兩種模型所需的總傳輸量
- 5.16：在不同的上傳頻率裡，兩種模型所需的總訓練時間

Section 5.5：Compare w/ Non-opp. Training

- 5.17：與不同non opp. training `converged_threshold` 的組別比較，MobileNet與ResNet-18在訓練過程中所能達到的 `best acc.`
- 5.18：與不同non opp. training `converged_threshold` 的組別比較，MobileNet與ResNet-18在訓練過程中所需的總傳輸量
- 5.19：與不同non opp. training `converged_threshold` 的組別比較，MobileNet與ResNet-18到達各目標accuracy所需的傳輸量
- 5.20：與不同non opp. training `converged_threshold` 的組別比較，MobileNet與ResNet-18每個回合所需的傳輸量
- ⇒ 總結：凸顯pcme (opp. training)，透過使用額外的運算資源，能夠達到non opp. training best `converged_threshold` 的結果
 - 省去了自己瞎試各種 `converged_threshold` 的時間
 - 且同一組參數可以直接套到ResNet-18, MobileNet兩種模型，不須依模型再調整

Section 5.6：PCME w/ FreezeOut

- 5.21：在單機版本裡，將套了PCME的FreezeOut與原始FreezeOut比較，列出accuracy隨著訓練回合的改變