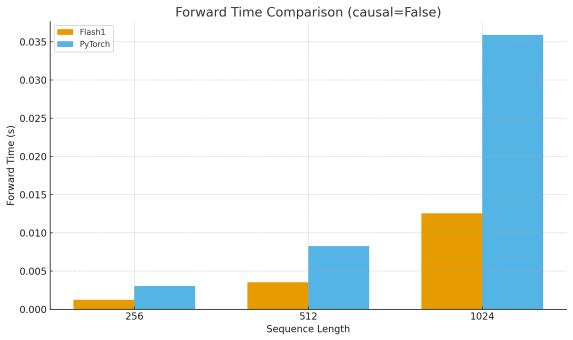
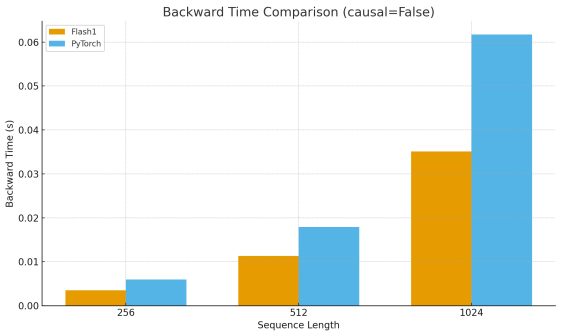
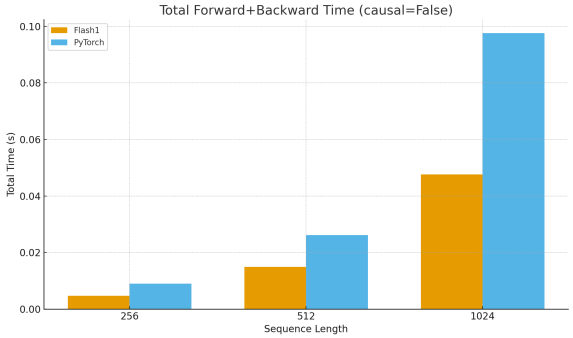
Lab6 FlashAttention
112062619 陳航希

1. causal=false, seqlen = 256, 512, 1024



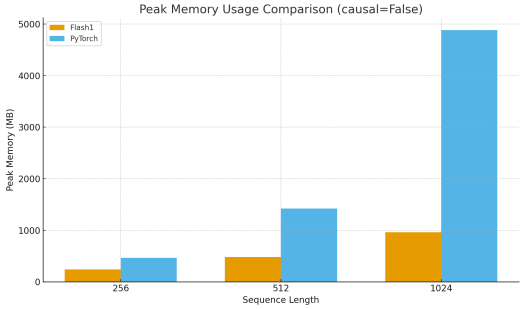




在這個實驗中,我比較了不同 sequence length（256 / 512 / 1024) causal=false的情況下, PyTorch 與 Flash1 在 forward time、backward time 以及 total time 上的效能差異。實驗設定如下: batch_size = 32、num_heads = 16、emb_dim = 1024、repeats = 30。
我將結果分別以柱狀圖呈現在 forward,backward及 forward+backward total time。
從圖中可以明顯觀察到以下現象:

1. Flash1的執行速度在所有階段都優於PyTorch版本。在forward time上面特別明顯。可以看到在seqlen=1024下, Flash1版本有3倍左右的速度優勢。backward time相對於forwad就沒有這麼明顯, 但還是有2倍左右的差距, 整體total time的差距可以看到還是以backward bound為主。

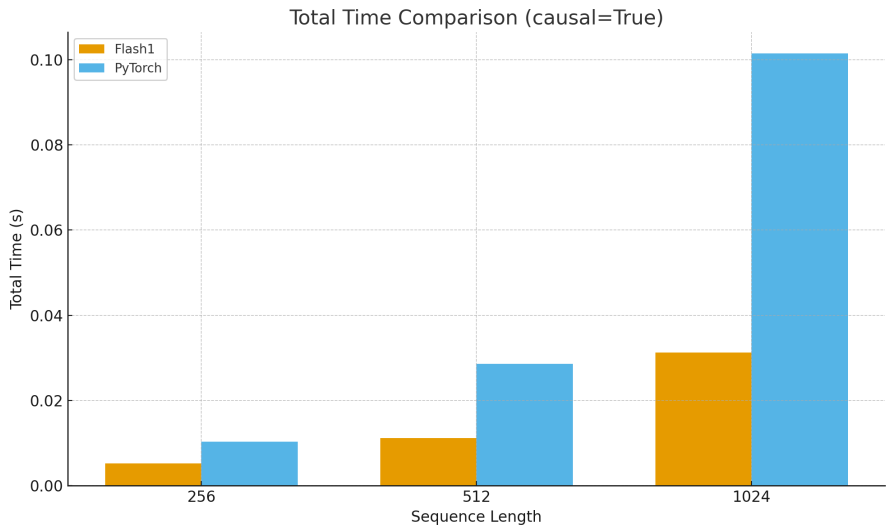
2. 除此之外, 隨著 seq len的增加,兩者的效能差距也會快速擴大。當 seq len= 256 時, Flash1 與 PyTorch 的差距不算大,但當 sequence length 提升至 1024 時,差異變得明顯, Flash1 幾乎能將執行時間壓到 PyTorch 的一半以下。



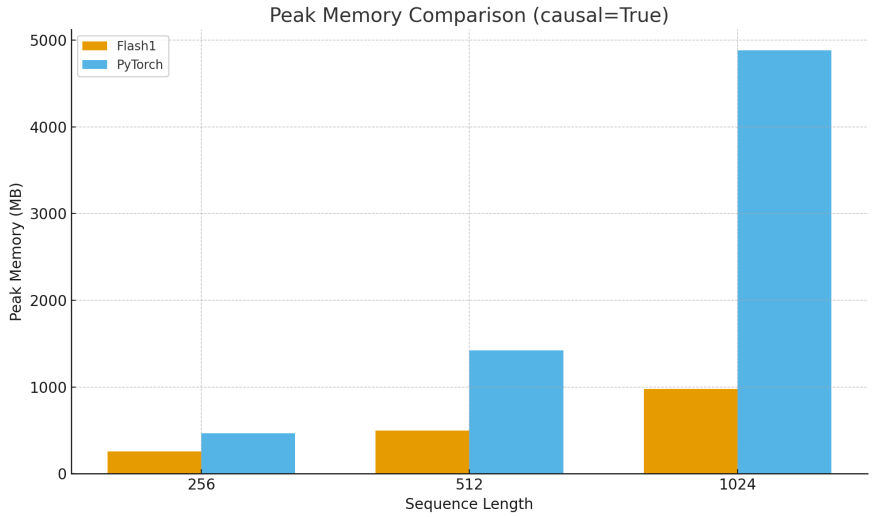
上圖為兩個實作方式在peak memory usage的差異, 可以看到隨著seq len增加, 兩個的差距相比速度有更明顯的差距, 當 sequence length = 256 時,兩者的差距大約落在 2 倍左右, 隨著 sequence length 增加到 512、1024, 這樣的差異變得更加明顯。尤其是在 sequence

length = 1024 的情況下，PyTorch 的記憶體需求接近 5GB，而 Flash1 僅約 1GB，記憶體使用量差距達到約 5 倍。

2. causal=true, seqlen = 256, 512, 1024



在 causal = True 的設定下，我同樣比較了不同 sequence length（256 / 512 / 1024）下PyTorch 與 Flash1 在整體 forward+backward（total time）上的效能差異。整體的趨勢與 non-causal 的情況相似: Flash1 在所有條件下仍然保持明顯優勢，而且Flash1 causal=true相較於 causal=false 時間上面又更短，而pytorch attention則沒有明顯的變化 導致差異進一步擴大。



最後causal=true的情況下peak memory使用的false的情況下相同 都是Flash1壓倒性的低。