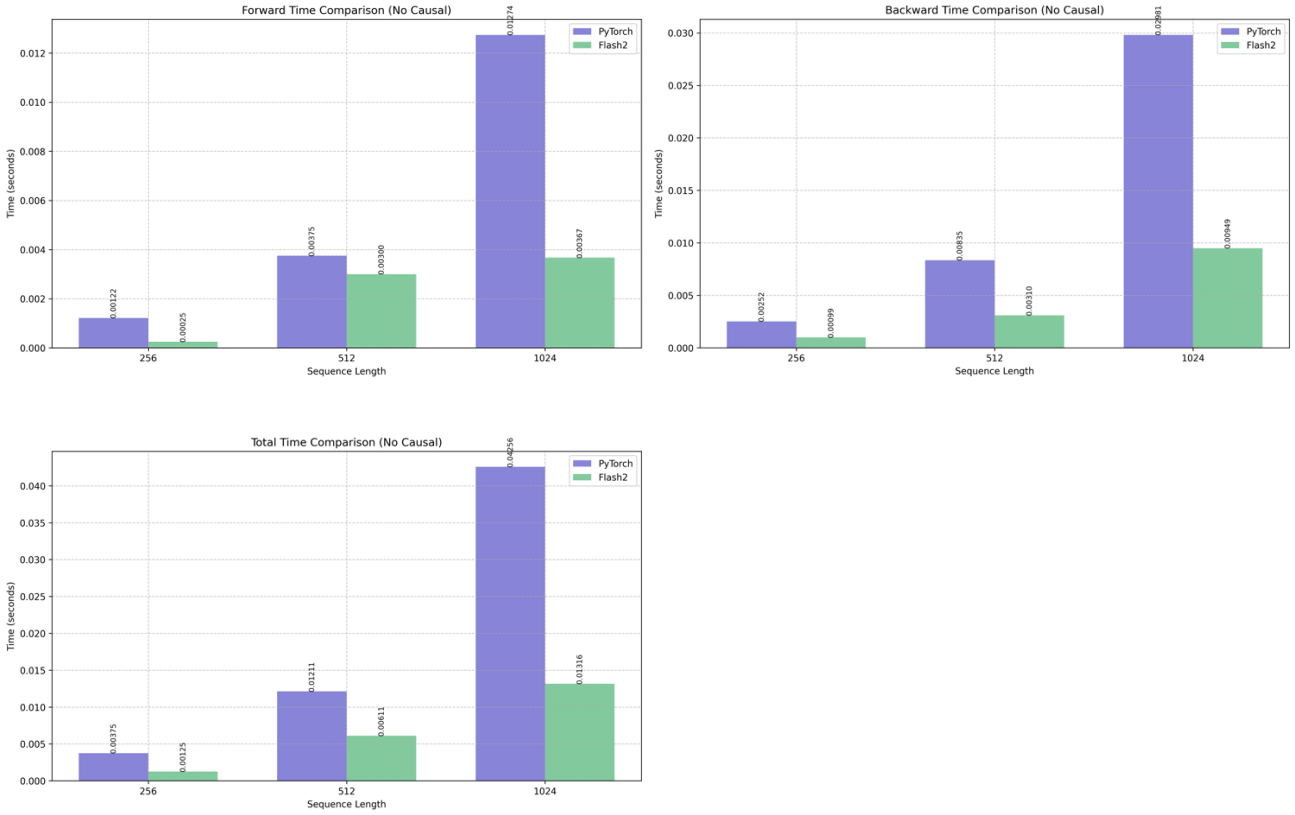
Lab5
112062520 戴維恩

1. 必較不同 sequence length 之下，Pytorch 與 Flash2 的執行時間差異。

   設定：batch_size = 16 / num_heads = 32 / emb_dim = 2048 / repeats = 30

   結論：我將結果分成比較 forward time（左）/ backward time（右） / total time（下）三種。可以發現無論是 forward time 還是 backward time，Flash2 版本的速度都快上許多，由此可見優化資料讀取方式確實可以讓程式效能得到很大的提升；尤其當 sequence length 數值增加，這樣的效能差距會更加明顯，在 sequence length = 1024 時，Flash2 比 Pytorch 快約 3 倍，這樣的加速相當驚人。

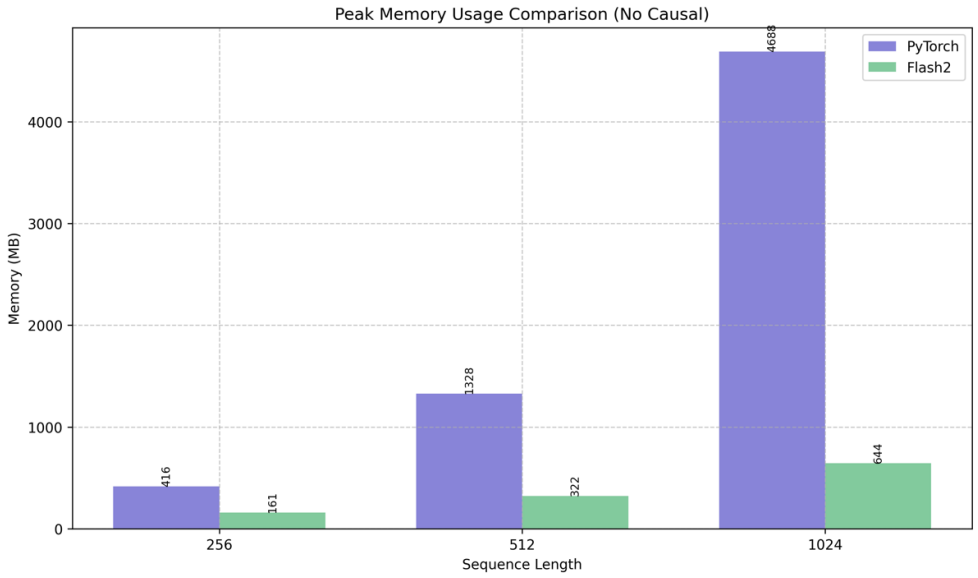


2. 必較不同 sequence length 之下，Pytorch 與 Flash2 的 peak_mem_usage 差異。

   設定：batch_size = 16 / num_heads = 32 / emb_dim = 2048 / repeats = 30

   結論：在 memory usage 的差異上面就明顯更大了，在 sequence length = 1024 時，Flash2 僅使用 644MB，而 Pytorch 需要 4688MB，而隨著 sequence length 增加，Flash2 的成長曲線也較為平緩。

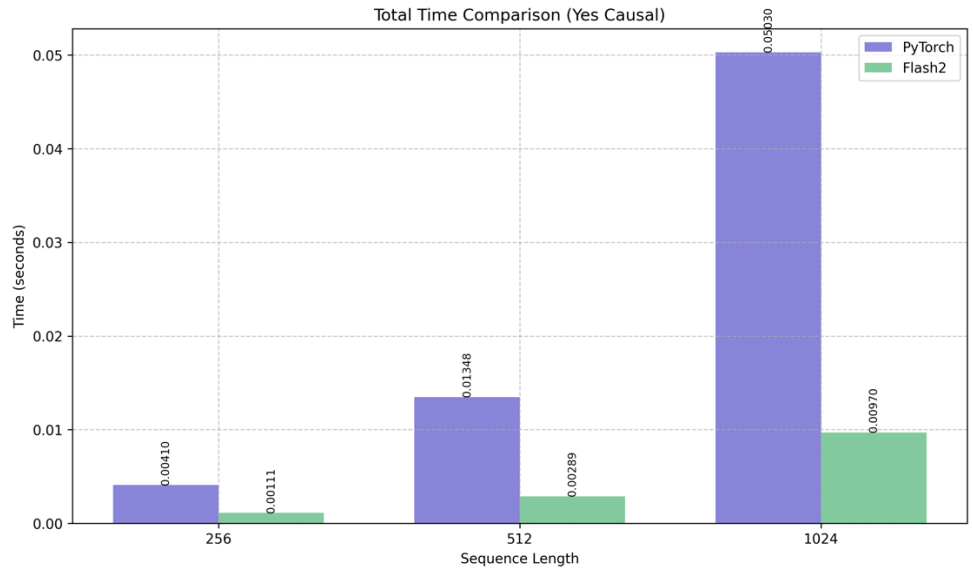
Lab5
112062520 戴維恩



3. 必較不同 sequence length 之下，且為 causal 計算模式，Pytorch 與 Flash2 的執行時間差異。

   設定：batch_size = 16 / num_heads = 32 / emb_dim = 2048 / repeats = 30

   結論：可以發現在 causal 模式下，會需要額外的計算時間，因此不管是 Pytorch 或是 Flash2 在執行時間總量方面都有些微增加；而以相對的結果來看，依舊是 Flash2 的效能比起 Pytorch 版本優化許多，成長曲線也較為平緩。



4. 必較不同 sequence length 之下，且為 causal 計算模式，Pytorch 與 Flash2 的 peak_mem_usage 差異。

   設定：batch_size = 16 / num_heads = 32 / emb_dim = 2048 / repeats = 30

Lab5
112062520 戴維恩

結論：在 memory usage 上，是否為 causal 模式並沒有差異，因此 memory 效能維持相同。