HyperPrompt: Prompt-based Task-Conditioning of Transformers

Yun He, Huaixiu Steven Zheng, Yi Tay, Jai Gupta, Yu Du, Vamsi Aribandi, Zhe Zhao, YaGuang Li, Zhao Chen, Donald Metzler, Heng-Tze Cheng, Ed H. Chi

Proceedings of the 39th International Conference on Machine Learning

簡介

近年來，Prompt-Tuning在自然語言處理領域引起了廣泛關注，作為一種新的參數高效微調範式。這種方法允許對大型語言模型進行輕量級的調整，與Adapter層在效率上相似。然而，在多任務學習場景中，語言模型仍面臨著諸多挑戰，特別是在訓練和服務單一模型的同時，還需要在所有任務中實現帕累托效率。

本文介紹了一種新穎的方法：HyperPrompt。這種方法為Transformer模型引入了任務條件化的hyper-prompts，並將其注入到self-attention模塊中，作為全局任務記憶。HyperPrompt的關鍵創新在於使用HyperNetworks生成這些prompts，實現了參數和計算效率，同時允許任務間靈活的信息共享。

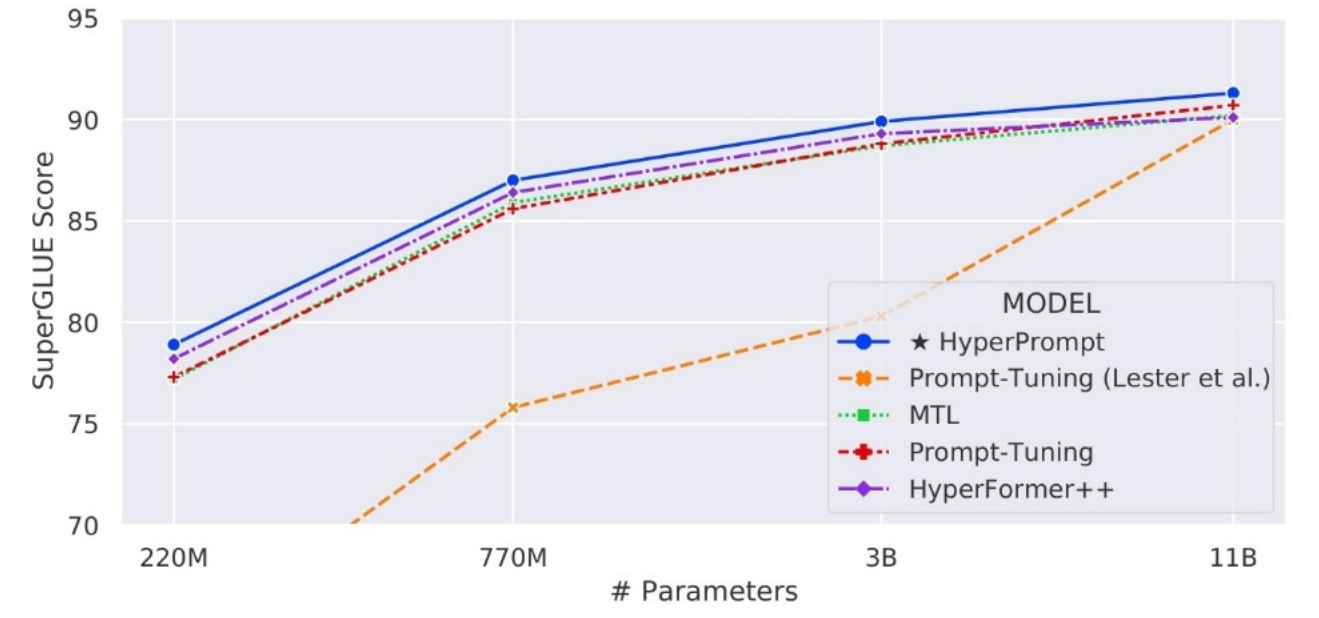


Figure 1. HyperPrompt achieves state-of-the-art performance on

如圖1所示，HyperPrompt在多種模型規模上都表現出色，特別是在SuperGLUE基準測試中，值得

如圖1所示，HyperPrompt在各種模型規模上都表現出色，特別是在SuperGLUE基準測試中。值得注意的是，即使在較小的模型上（如220M參數的模型），HyperPrompt也能顯著優於多任務學習（MTL）基線和其他參數高效的方法，如Prompt-Tuning和HyperFormer++。這一結果表明，HyperPrompt成功地解決了之前Prompt-Tuning方法在中等規模模型上性能不佳的問題。

HyperPrompt的主要優勢包括：

1. 參數和計算效率高
2. 任務間信息共享靈活
3. 在各種模型規模上性能均有提升

這些特性使HyperPrompt成為一種有前景的多任務學習方法，特別適用於需要在多個任務上同時取得良好性能的場景。

問題陳述

在多任務學習的一般設定中，我們考慮一組任務　　，其中　是任務總數，　　表示第　個任務的訓練集，包含　　個樣本。給定一個預訓練的Transformer模型　　（例如T5），我們的目標是最小化以下目標函數：

$$L(\theta) = \sum_{\tau=1}^{T} \sum_{n=1}^{N_\tau} C(f_\theta(x_\tau^{(n)}), y_\tau^{(n)})$$

其中　　通常是交叉熵損失，　　是模型對訓練樣本　　的輸出。

然而，這種直接的多任務學習方法存在一些問題：

1. 任務無關的參數　導致性能不如單任務微調
2. 難以捕捉任務特定的信息，特別是對於低資源任務

為了解決這些問題，我們引入了一組任務條件化參數　　，更新後的目標函數為：

$$L(\theta, \delta_{\tau_{\tau=1}}^{T}) = \sum_{\tau=1}^{T} \sum_{n=1}^{N_\tau} C(f_{\theta,\delta_\tau}(x_\tau^{(n)}), y_\tau^{(n)})$$

其中　是第　個任務的特定參數化。

我們的主要目標是：

1. 通過引入任務條件化參數　　來提高大多數任務的微調性能
2. 保持參數效率，即

這種方法旨在實現多任務學習的參數效率和計算效率，同時在所有任務上達到帕累托效率。

方法

HyperPrompt的設計遵循兩個關鍵原則：

1. 將任務條件注入self-attention模塊，以提高計算效率並通過token級交互增強表達能力
2. 使用HyperNetworks生成prompts，同時提高參數效率並允許靈活的任務共享程度

Prompt-Based任務條件化Transformer

在標準的self-attention計算中，我們有：

$$K_\tau = X_\tau W_k, V_\tau = X_\tau W_v, Q_\tau = X_\tau W_q$$

其中　　是來自第　個任務的輸入序列，　是序列長度，　是模型維度。

HyperPrompt引入了hyper-prompts　　和　　，並將它們與原始的key和value拼接：

$$K'_\tau = \mathrm{concat}(P_{\tau,k}, K_\tau)$$

$$V'_\tau = \mathrm{concat}(P_{\tau,v}, V_\tau)$$

修改後的attention計算為：

$$O_\tau = \mathrm{Attention}(Q_\tau, K'_\tau, V'_\tau) = \mathrm{sof\,tmax}(Q_\tau K'^{T}_\tau) V'_\tau$$

這種設計帶來兩個主要好處：

1. 直接參與attention特徵圖的計算，允許tokens獲取任務特定的語義
2. 作為任務特定的記憶，供multihead attention檢索相關信息

HyperPrompt架構

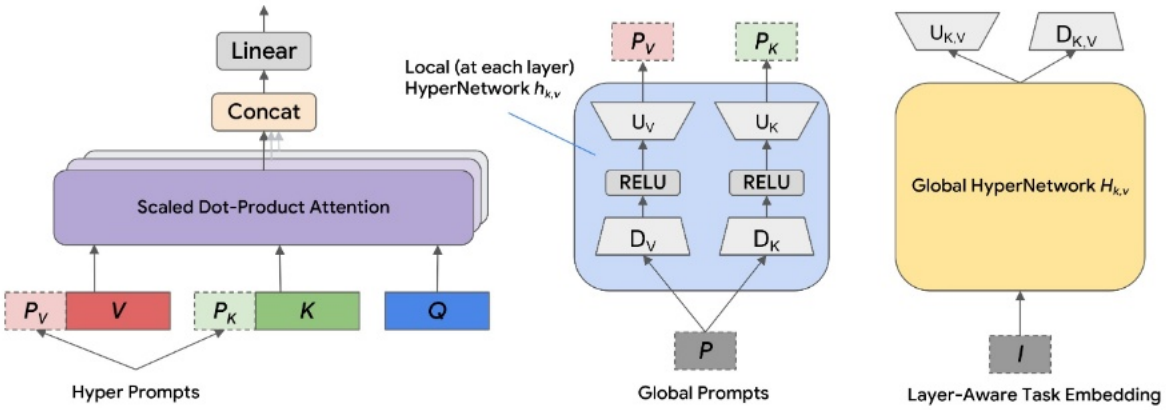


Figure 2. HyperPrompt framework: (a) in each Transformer block, task-specific hyper-prompts $P_{K,V}$ are prepended to the original key K and value V for the query Q to attend to, (b) in HyperPrompt-Share/Sep, global prompts P are used to generate the hyper-prompts $P_{K,V}$

如圖2所示，HyperPrompt架構包含以下關鍵組件：

1. **全局prompts**：　　，其中

2. **局部HyperNetworks**：在第　個Transformer塊中：

$$\mathrm{P}_{\tau,\mathrm{k}}^{\mathrm{m}} = \mathrm{h}_{\mathrm{k}}^{\mathrm{m}}(\mathrm{P}_{\tau}) = \mathrm{U}_{\mathrm{k}}^{\mathrm{m}}(\mathrm{Relu}(\mathrm{D}_{\mathrm{k}}^{\mathrm{m}}(\mathrm{P}_{\tau})))$$

$$\mathrm{P}_{\tau,\mathrm{v}}^{\mathrm{m}} = \mathrm{h}_{\mathrm{v}}^{\mathrm{m}}(\mathrm{P}_{\tau}) = \mathrm{U}_{\mathrm{v}}^{\mathrm{m}}(\mathrm{Relu}(\mathrm{D}_{\mathrm{v}}^{\mathrm{m}}(\mathrm{P}_{\tau})))$$

　其中　　和　　是下投影和上投影矩陣，　是瓶頸維度 (　　)。

HyperPrompt有三個變體：

1. **HyperPrompt-Share**：所有任務共享相同的局部HyperNetworks
2. **HyperPrompt-Sep**：每個任務有自己的局部HyperNetworks
3. **HyperPrompt-Global**：使用全局HyperNetwork生成所有任務和塊的局部HyperNetworks

HyperPrompt-Global詳解

HyperPrompt-Global引入了層感知任務嵌入：

$$\mathrm{I}_{\tau}^{\mathrm{m}} = \mathrm{h}_{\mathrm{t}}(\mathrm{k}_{\tau}, \mathrm{z}_{\mathrm{m}})$$

其中　是任務嵌入，　是層嵌入。

全局HyperNetworks定義如下：

$$(U_{\tau,k}^{m}, D_{\tau,k}^{m}) = H_k(I_\tau^m) = (W_{Uk}, W_{Dk})I_\tau^m$$

$$(U_{\tau,v}^{m}, D_{\tau,v}^{m}) = H_v(I_\tau^m) = (W_{Uv}, W_{Dv})I_\tau^m$$

HyperPrompt-Global的優勢包括：

1. 在任務和層之間實現靈活的信息共享
2. 參數高效的任務條件化

參數效率分析

HyperPrompt-Global的額外參數總數為：

$$dlT + 4(bdt) + Tt' + Mt' + (2t' + t)e$$

空間複雜度為 ，對 呈亞線性擴展。

在實際應用中，由於 ， ， ， ，通常有 。因此，簡化後的空間複雜度為 ，主要來自全局HyperNetworks，實際上獨立於 、 和 。

實驗

實驗設置

數據集：

- GLUE和SuperGLUE基準測試

模型：

- T5（Text-to-Text Transfer Transformer）
- 規模：Base（220M）到XXL（11B）

訓練細節：

- 300K步，批量大小128
- 學習率：1e-3，使用Adam優化器

評估：

- 為每個任務選擇最佳檢查點
- 計算每個任務的所有指標的平均值
- 計算GLUE和SuperGLUE所有任務的平均值

基線方法

1. **MTL**：用於多任務學習的原始T5
2. **Vanilla Adapter**：為每個任務添加Adapter模塊
3. **HyperFormer++**：使用HyperNetworks生成adapters
4. **Prompt-Tuning**：修改為多任務學習，為每個任務添加prompts並共同訓練所有任務

主要結果

![Figure 1]

如圖1所示，HyperPrompt-Global在所有模型規模上都優於所有基線方法：

- 在T5 Base上，SuperGLUE得分為78.9，比MTL基線的77.2有顯著提升
- 在T5 XXL上，SuperGLUE得分達到91.3，比MTL基線的90.2更高

值得注意的是，HyperPrompt-Global僅增加了0.14%的額外參數，就實現了這種性能提升。

全模型vs任務特定調整

在GLUE和SuperGLUE數據集上比較了全模型微調和僅調整任務特定參數的效果（使用T5 Large）：

1. GLUE結果：任務特定調整與MTL基線相當
2. SuperGLUE結果：任務特定調整存在較大的性能差距
   - HyperPrompt-Global：下降5.5個點
   - HyperFormer++：下降5.9個點

結論：對於困難的任務，全模型調整是必要的，以達到競爭性的結果。

計算效率

![Table 2]

如表2所示，HyperPrompt變體在前向傳播的操作數最少（9.8×10^12），而HyperFormer++最高（約為其他方法的3倍）。

在訓練時間方面：

- HyperPrompt-Share最快（8.0小時）
- HyperPrompt-Global與Vanilla Adapter相當
- HyperFormer++和Prompt-Tuning顯著更長

這表明HyperPrompt在訓練和推理方面都具有計算效率。

消融研究

T5 Base結果：

模型	GLUE	SuperGLUE	參數增加
HyperPrompt-Global	86.8	78.9	1.04x
MTL基線	85.5	77.2	1.0x
HyperFormer++	86.5	78.2	1.04x

T5 Large結果：

模型	GLUE	SuperGLUE	參數增加
HyperPrompt-Global	89.4	87.0	1.02x
MTL基線	88.3	85.9	1.0x
HyperFormer++	88.8	86.4	1.02x

這些結果表明，HyperPrompt-Global在保持參數效率的同時，在GLUE和SuperGLUE上都實現了最佳性能。

HyperPrompt變體比較

1. **HyperPrompt-Share**：
    - 在SuperGLUE上表現更好（Base和Large模型）
    - 在GLUE上比HyperPrompt-Sep差
2. **HyperPrompt-Sep**：
    - 在GLUE上表現更好（Base和Large模型）

- 在SuperGLUE上比HyperPrompt-Share差

3. **HyperPrompt-Global**：

- 在GLUE和SuperGLUE上都始終表現最佳
- 展示了調整信息共享程度的能力

Attention模式分析

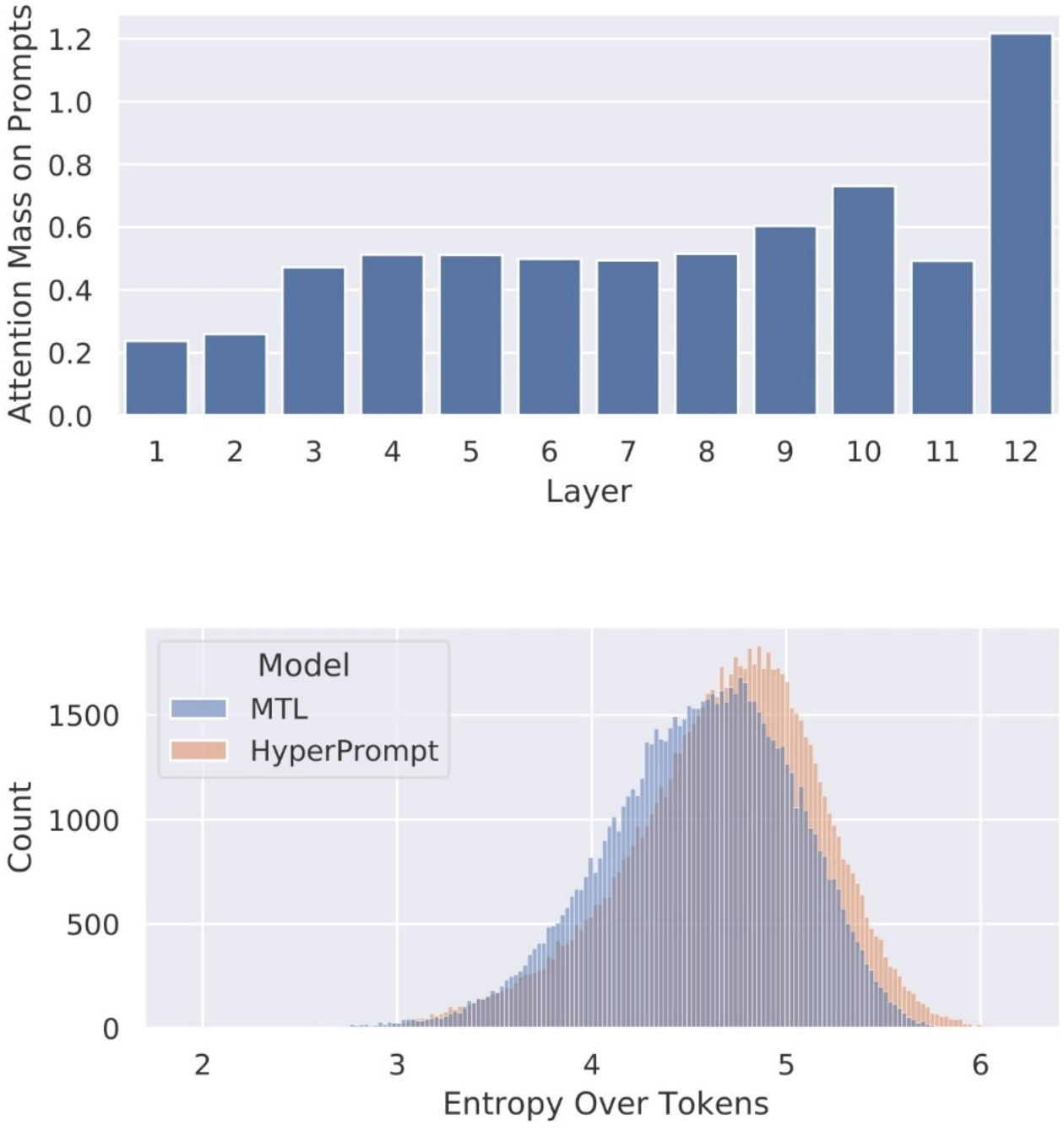


Figure 3. Visualization of attention mass and entropy distribution

Figure 3. Visualization of attention mass and entropy distribution.

如圖3所示，我們分析了HyperPrompt-Global的attention模式：

1. **Hyper-prompts上的attention質量**：
   - 在較低層較低，在較高層較高
   - 表明較高層更專注於任務特化
2. **Tokens上的attention分數熵**：
   - HyperPrompt-Global顯示更高的熵
   - 暗示更多樣化的attention分佈

這些觀察結果表明：

- 較低層學習任務無關的表示
- 較高的熵可能有助於更好的泛化

Hyper-Prompt長度的影響

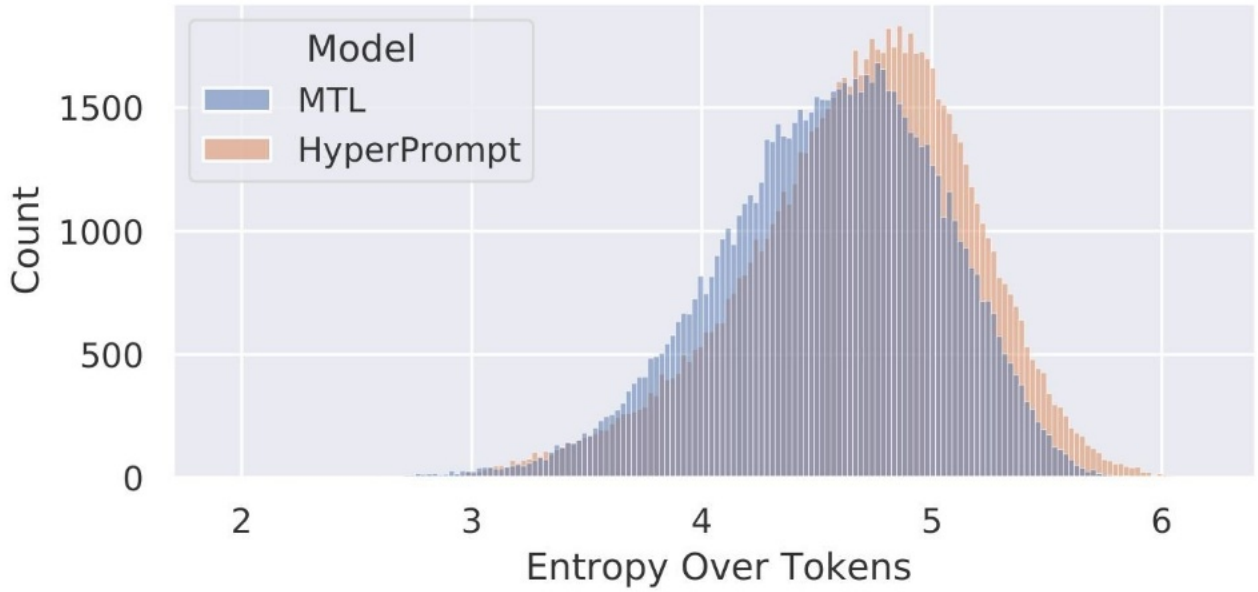
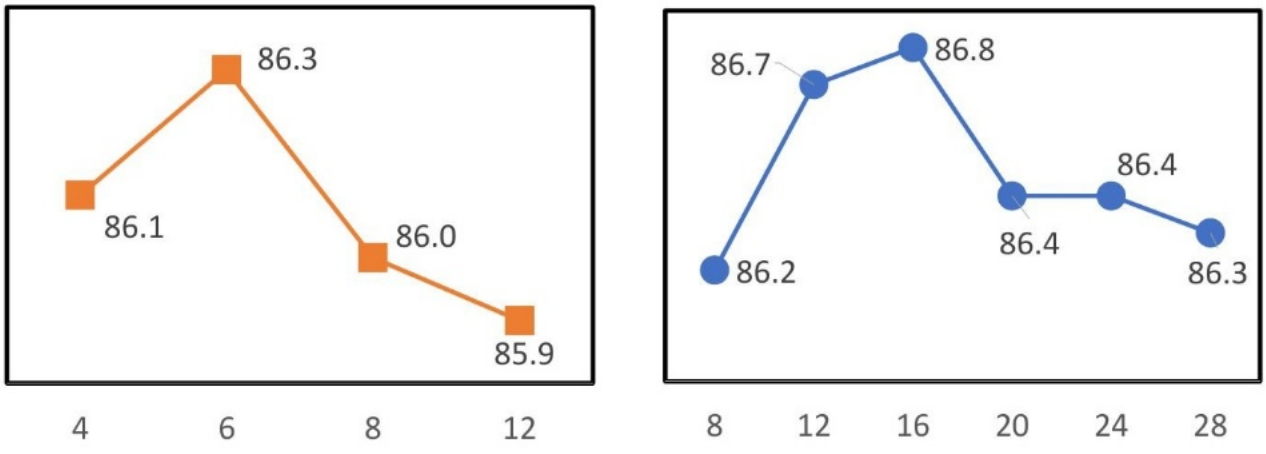


Figure 3. Visualization of attention mass and entropy distribution.

distribution towards higher values for HyperPrompt-Global. This signifies that injecting hyper-prompts encourages a more diverse attention distribution, which seems to be beneficial to model generalization.

4.7. Impact of Hyper-Prompt Length

HyperPrompt prepends l trainable hyper-prompts to the keys and values of self-attention layer at every Transformer layer. In Figure 4, we present the results of tuning the prompt length l on GLUE using T5 Base as the example for HyperPrompt-Global (similar patterns are observed on T5 Large and SuperGLUE). We first add hyper-prompts on the decoder and search the best l and then search the best l for the encoder with the fixed best decoder hyper-prompt length. As shown in Figure 4(a), $l = 6$ is the best for the decoder. As shown in Figure 4(b), HyperPrompt-Global



(a) Decoder (b) Encoder

Figure 4. Impact of hyper-prompt length in HyperPrompt-Global (GLUE score on T5 Base).

如圖4所示，我們研究了hyper-prompt長度對性能的影響：

1. **解碼器hyper-prompt長度**：
    - 最佳長度：l = 6
2. **編碼器hyper-prompt長度**：
    - 最佳長度：l = 16（解碼器固定為l = 6）

性能與長度的關係：

- 峰值性能出現在 l ~ O(10)
- 原始序列長度：512（編碼器），32（解碼器）
- Hyper-prompts不會顯著增加時間複雜度

編碼器vs解碼器分析

模型部分	GLUE性能	SuperGLUE性能
僅編碼器	更好	顯著較差
僅解碼器	較差	顯著更好

可能的解釋：

- 編碼器prompts可能存在可訓練性問題
- 可能需要不同的學習率

建議：在SuperGLUE實驗中使用僅解碼器的設置。

結論

本研究提出了HyperPrompt，這是一種新穎的prompt-based任務條件化Transformer架構。HyperPrompt的主要貢獻包括：

1. 將hyper-prompts注入self-attention模塊，作為全局任務記憶
2. 使用HyperNetwork生成hyper-prompts，實現參數效率和靈活的任務信息共享

3. 在各種模型規模上都優於強基線方法，包括MTL、Prompt-Tuning和HyperFormer++

HyperPrompt的主要優勢：

- 參數和計算效率高
- 任務間信息共享靈活
- 在GLUE和SuperGLUE基準測試上表現出色

實驗結果表明，HyperPrompt在中等規模和大規模模型上都能有效提升性能，解決了之前Prompt-Tuning方法在較小模型上性能不佳的問題。

未來研究方向：

1. 探索HyperPrompt在零樣本和少樣本學習中的應用
2. 擴展到更大的模型和更多樣的任務
3. 研究編碼器prompts的可訓練性問題
4. 對HyperPrompt有效性進行理論分析

總的來說，HyperPrompt為多任務學習和參數高效微調提供了一個有前景的新方向，有潛力在各種NLP任務中得到廣泛應用。

論文總結

本堂課我們深入探討了HyperPrompt這一創新的多任務學習方法。我們學習了如何將prompt-based學習與HyperNetworks結合，以實現高效且靈活的任務條件化。HyperPrompt在GLUE和SuperGLUE等具有挑戰性的基準測試上的出色表現，證明了它在各種模型規模上的有效性。我們還討論了全模型微調vs任務特定參數調整的重要性，以及編碼器和解碼器在不同任務上的表現差異。這些見解為未來的研究提供了寶貴的方向，特別是在零樣本和少樣本學習、跨語言遷移等領域。總的來說，HyperPrompt為解決NLP中的多任務學習挑戰提供了一個富有前景的新方法。