HyperPrompt: Prompt-based Task-Conditioning of Transformers

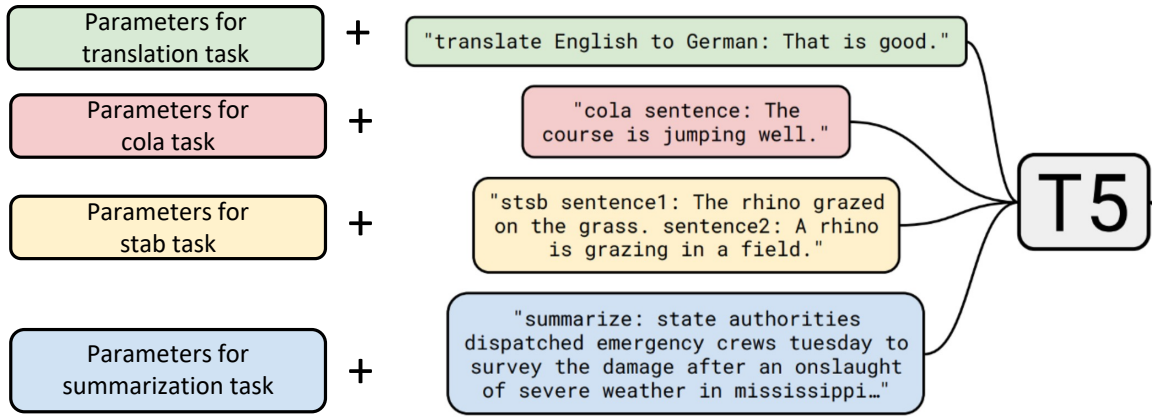
Yun He[*1] Huaixiu Steven Zheng[*2]
Yi Tay[2] Jai Gupta[2] Yu Du[2] Vamsi Aribandi[2]
Zhe Zhao[2] YaGuang Li[2] Zhao Chen[3]
Donald Metzler[2] Heng-Tze Cheng[2] Ed H. Chi[2]

*Equal contribution [1]Texas A&M University
[2]Google Research [3]Waymo LLC

Task-Conditioning of Transformers for MTL

- Multi-task learning (MTL) on Transformers
  - Pros: more parameter-efficient than single-task learning
  - Cons: the task interference is inevitable in fitting all task data sets within a single set of parameters.

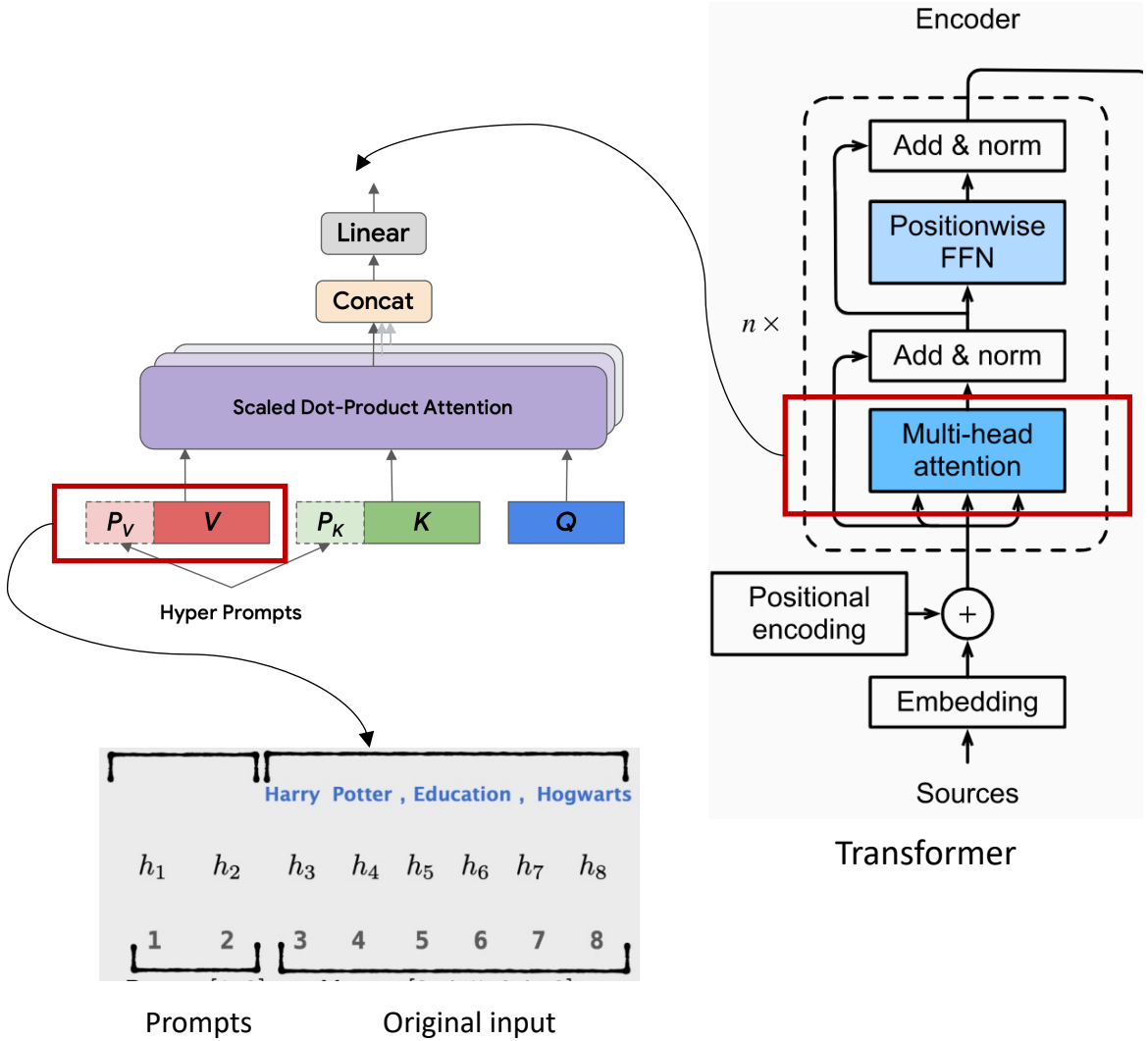
- Research Question: how to alleviate the task interference for Transformer-based MTL?
  - Each task has its own task-conditioned parameters, which are only updated by the corresponding task loss and hence will not be interfered by other tasks.



- Goal 1: inject task-conditioned parameters into Transformer.
- Goal 2: task-conditioned parameters should be space-efficient.

Proposed Methods: HyperPrompt

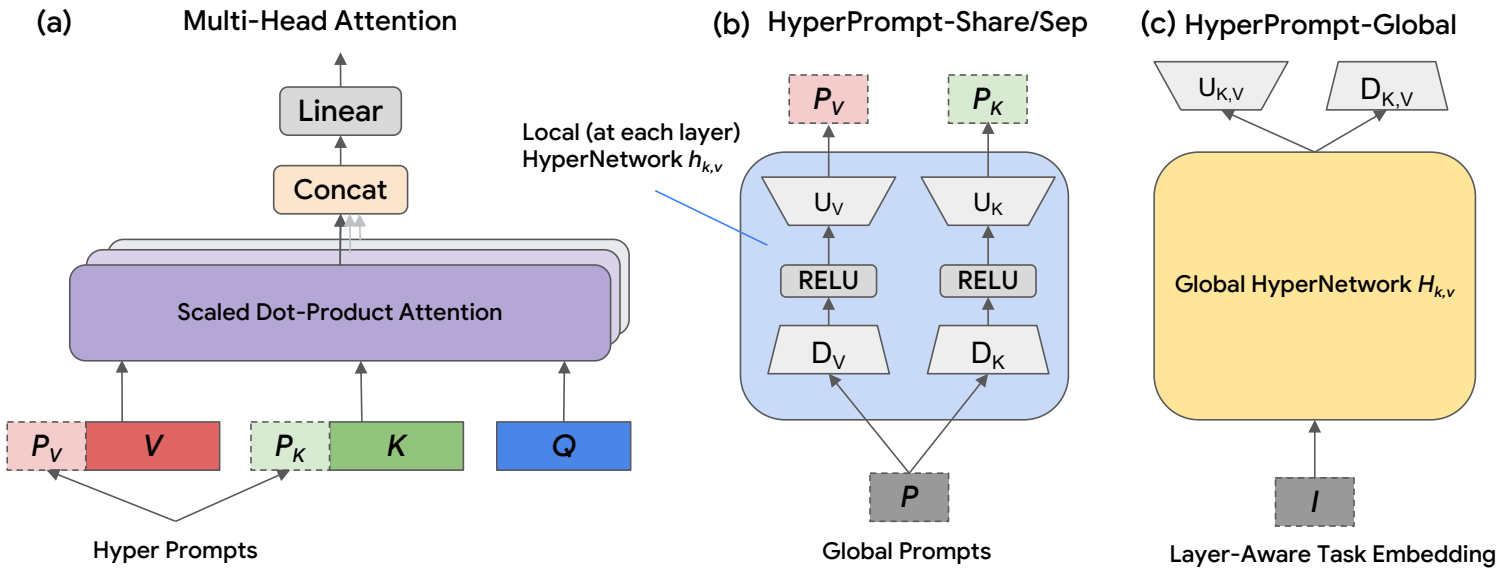
- Which module of a Transformer should we inject the task-conditioned parameters for MTL?
  - Previous work[1][2] focus on feed-forward network (FFN).
- Key difference from previous work:
  - Self-attention is better than FFN to have task-conditioned parameters.
- We inject hyper-prompts (learnable embeddings) into Transformer's self-attention modules.
  - Hyper-prompts serve as task-dependent global memories for the queries to attend to.



1. Hypergrid transformers: towards a single model for multiple tasks, ICLR 2021
2. Parameter-efficient Multi-task Fine-tuning for Transformers via Shared Hypernetworks, ACL 2021

HyperNetworks Generate Hyper-Prompts

- In prompt-tuning[1], prompts are directly initialized
- Key technical contribution:
  - Fig(b): at each layer of a Transformer, local HyperNetworks generate the hyper-prompts.
  - Fig(c): a global HyperNetwork generates the local HyperNetworks, which enables the flexible knowledge transfer between tasks and layers.
  - HyperNetworks also enables hyper-prompts to be parameter-efficient.

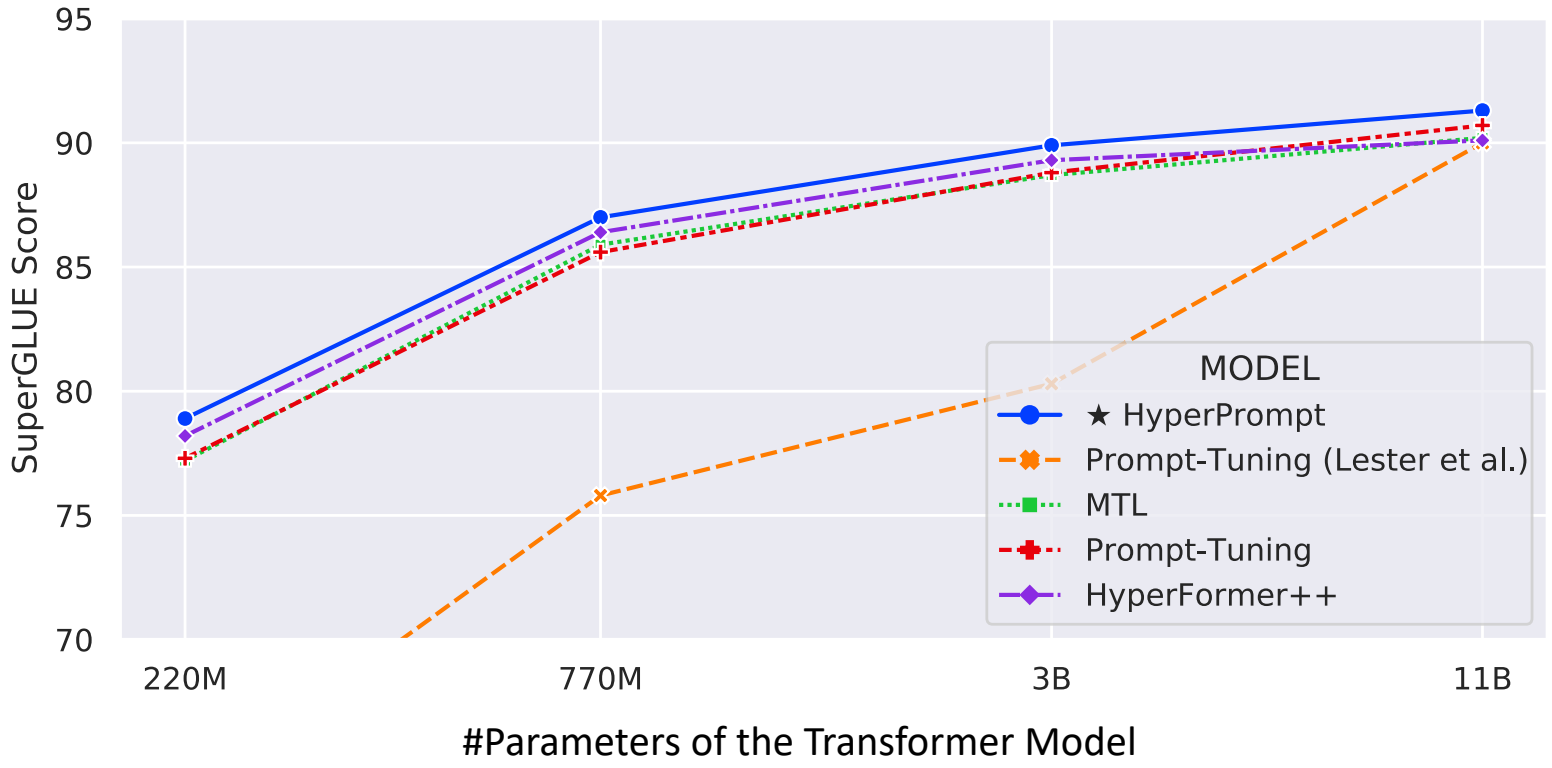


Framework of HyperPrompt

1. The Power of Scale for Parameter-Efficient Prompt Tuning, EMNLP 2021

Key Results

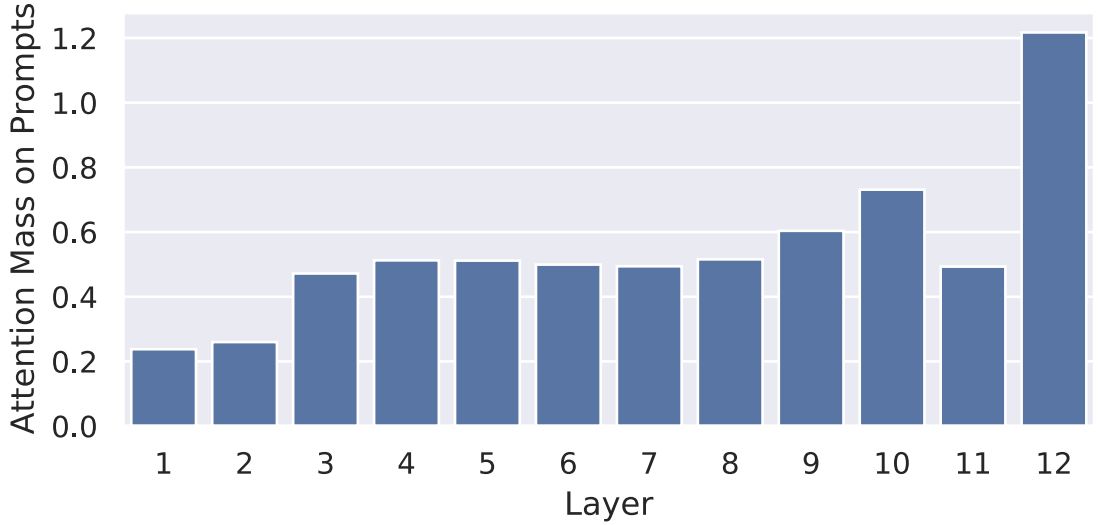
- T5 is the base Transformer model and HyperPrompt and baseline methods are applied on top of the base model. HyperPrompt achieves the SOTA performance on SuperGLUE across four different model sizes.



- HyperPrompt vs. Prompt-Tuning
  - Hyper-prompts are generated by HyperNetworks, allowing flexible knowledge transfer between tasks and layers.

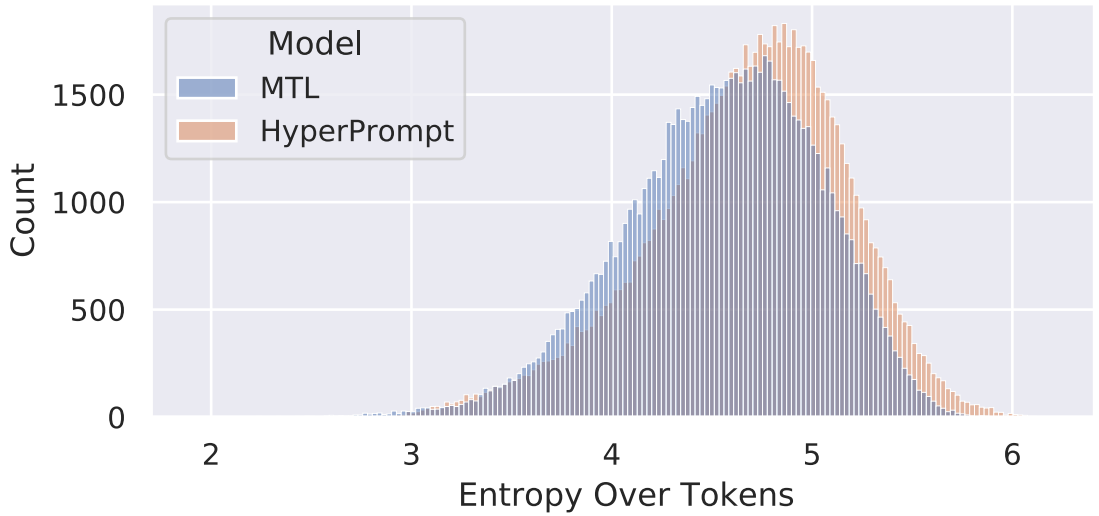
Peeking into HyperPrompt

Fig 1. The attention mass on hyper-prompts for each encoder layer



Higher-levels of Transformer becomes more task-specialized while it is beneficial for the lower-levels to learn task-agnostic representation.

Fig 2. The entropy of the attention scores on the tokens



A shift of entropy distribution towards higher values for HyperPrompt, showing that injecting hyper-prompts encourages a more diverse attention distribution.

Highlights

- We introduce hyper-prompts as task-conditioned parameters to alleviate the task interference and conflicts for multi-task learning on Transformers.

- Key differences between HyperPrompt and previous work:
  1. Hyper-prompts are injected into the self-attention module, which is a better place for task-conditioning than feed-forward module.
  2. Tuning all parameters is better than freezing backbone model.

- Key technical contribution: the hyper-prompts are end-to-end learnable via generation by HyperNetworks, enabling flexible knowledge sharing among tasks and layers.

- HyperPrompt outperforms the strong MTL baseline by a large margin on SuperGLUE score (78.9 vs 77.2 for T5 Base). Such a performance gain continues all the way to model size as big as XXL with 11B parameters (91.3 vs 90.2) with only 0.14% additional task-conditioned parameters.