Visual Prompt Tuning

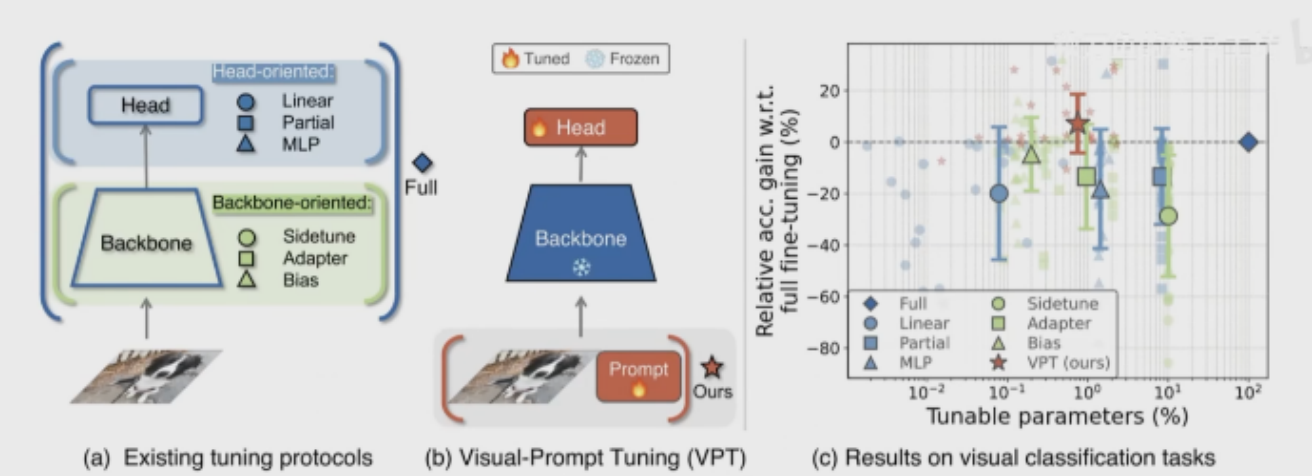
Visual Prompt Tuning

Menglin Jia*[1,2], Luming Tang*[1]
Bor-Chun Chen[2], Claire Cardie[1], Serge Belongie[3]
Bharath Hariharan[1], and Ser-Nam Lim[2]

[1]Cornell University [2]Meta AI [3]University of Copenhagen

Abstract. The current *modus operandi* in adapting pre-trained models involves updating all the backbone parameters, *i.e.*, full fine-tuning. This paper introduces Visual Prompt Tuning (VPT) as an efficient and effective alternative to full fine-tuning for large-scale Transformer models in vision. Taking inspiration from recent advances in efficiently tuning large language models, VPT introduces only a small amount (less than 1% of model parameters) of trainable parameters in the input space while keeping the model backbone frozen. Via extensive experiments on a wide variety of downstream recognition tasks, we show that VPT achieves significant performance gains compared to other parameter efficient tuning protocols. Most importantly, VPT even outperforms full fine-tuning in many cases across model capacities and training data scales, while reducing per-task storage cost. Code is available at github.com/kmnp/vpt.

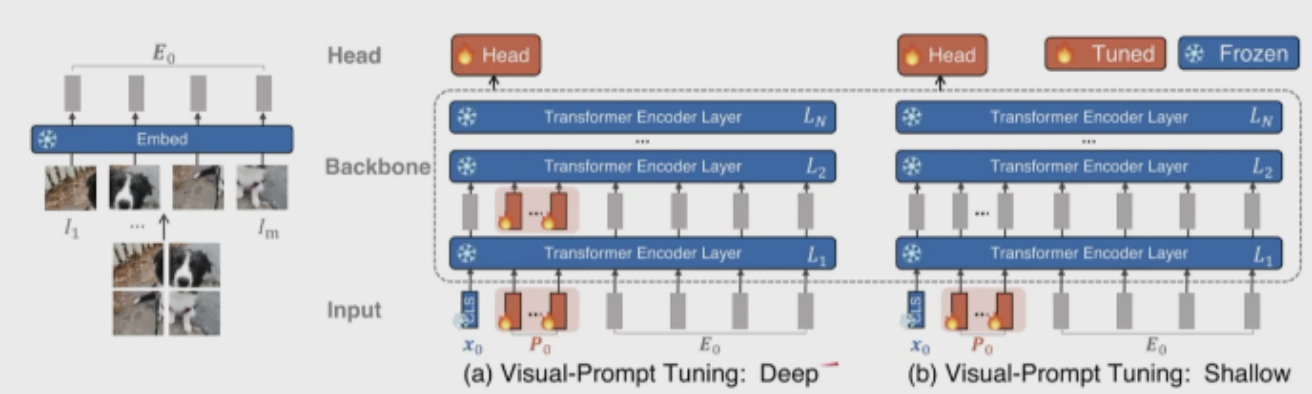
提出一个叫Visual prompt tuning的微调方法，主要作用于视觉模型，引入一小部分可训练的参数（不到1%模型参数量），在冻结模型backbone的情况下，达到比较好的效果。文章提到和别的方法比，在下游任务上，起到了显著性的效果。



在fine-tune中有很多的思路，frozen整个网络在后面加隐藏层，又或者只frozen backbone，然后作前后处理（加head layer），也可能是把所有参数都激活，一起来fine-tune。但是如果

要fine-tune整个backbone的时候，需要把所有参数都load到GPU里面，训练耗时耗力还耗钱。

prompt是从NLP里传出来的，是通过一个很小的Token或者说Embedding，把它加入到网络的计算中，本文是加入到了Vision Transformer中，加了一个prompt token，然后fine-tune的时候只是训练prompt token和关系下游任务的Head layer的参数。



Shallow: Prompts are inserted into the first Transformer layer only

$$[\mathbf{x}_1, \mathbf{Z}_1, \mathbf{E}_1] = L_1([\mathbf{x}_0, \mathbf{P}, \mathbf{E}_0])$$
$$[\mathbf{x}_i, \mathbf{Z}_i, \mathbf{E}_i] = L_i([\mathbf{x}_{i-1}, \mathbf{Z}_{i-1}, \mathbf{E}_{i-1}])$$
$$\mathbf{y} = \texttt{Head}(\mathbf{x}_N) \ ,$$

Deep: Prompts are introduced at every Transformer layer's input space.

$$[\mathbf{x}_i, \_, \mathbf{E}_i] = L_i([\mathbf{x}_{i-1}, \mathbf{P}_{i-1}, \mathbf{E}_{i-1}])$$
$$\mathbf{y} = \texttt{Head}(\mathbf{x}_N) \ .$$

Deep：transformer里面的每一层都加入prompt token去做fine-tune
Shallow：只在transformer的第一层加prompt token

Results

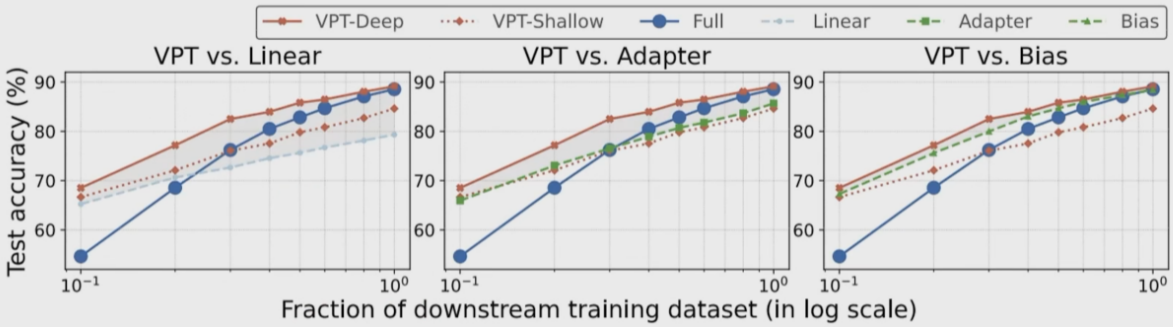


Fig. 3. Performance comparison on different downstream data scales, averaged across 5 FGVC tasks. VPT-DEEP is compared with LINEAR (left), ADAPTER (middle) and BIAS (right). Highlighted region shows the accuracy difference between VPT-DEEP and the compared method. Results of VPT-SHALLOW are FULL presented in all plots for easy reference. The size of markers are proportional to the percentage of tunable parameters in log scale