Prompt Tuning for Unified Multimodal Pretrained Models

Hao Yang*, **Junyang Lin***, **An Yang, Peng Wang, Chang Zhou**
DAMO Academy, Alibaba Group
{yh351016, junyang.ljy, ya235025, zheluo.wp, ericzhou.zc}@alibaba-inc.com

Abstract

Prompt tuning has become a new paradigm for model tuning and it has demonstrated success in natural language pretraining and even vision pretraining. The parameter-efficient prompt tuning methods that optimize soft embeddings while keeping the pretrained model frozen demonstrate advantages in low computation costs and almost lossless performance. In this work, we explore the transfer of prompt tuning to multimodal pretrained models. Specifically, we implement prompt tuning to a unified sequence-to-sequence pretrained model by adding a sequence of learnable embeddings to each layer and finetuning the pretrained model on downstream task with only the learnable embeddings being optimized. Experimental results on a series of multimodal understanding and generation tasks demonstrate that our method OFA-PT can achieve comparable performance with finetuning across a series of multimodal generation and understanding tasks. Additionally, it significantly outperforms the unified multimodal pretrained model with other parameter-efficient tuning methods, e.g., Adapter, BitFit. etc. Besides, in comparison with finetuned models, the prompt-tuned models demonstrate improved robustness against adversarial attacks. We further figure out that experimental factors, including prompt length, prompt depth, and reparameteratization, have great impacts on the model performance, and thus we empirically provide a recommendation for the setups of prompt tuning. Codes and checkpoints are available at https://github.com/OFA-Sys/OFA

1 Introduction

Recent years have witnessed the great success of large-scale pretraining based on large models and big data in natural language processing (NLP) (Radford et al., 2018; Devlin et al., 2019; Yang et al., 2019; Liu et al., 2019; Raffel et al., 2020; Brown et al., 2020) and computer vision (Chen et al., 2020b,a,c; Chen and He, 2021; Bao et al., 2021; He et al., 2021b). Inspired by the success of BERT-like models (Devlin et al., 2019), researchers have found that pretraining can level up the downstream performance of cross-modal representation learning algorithms by a large margin (Chen et al., 2020d; Lu et al., 2019; Su et al., 2020; Tan and Bansal, 2019; Wang et al., 2021).

Following this line of research, unified multimodal pretrained models have gradually attracted much attention, and very recently, a series of such models based on the sequence-to-sequence learning framework have unified both cross-modal understanding and generation tasks and even achieved state-of-the-art performance (Li et al., 2022; Wang et al., 2022a; Yu et al., 2022; Alayrac et al., 2022; Wang et al., 2022b; Chen et al., 2022). Furthermore, note that the scale of unified multimodal pretrained models has been growing rapidly, showing a similar trend of developments in large language models (Raffel et al., 2020; Brown et al., 2020; Chowdhery et al., 2022).

Despite the great success of large-scale pretrained models across multiple domains, training such models requires a large amount of computation costs. The conventional finetuning is though effective in gaining high performance yet suffers from low training efficiency, especially when the pretrained model is of large scale in model size. There is a strong necessity for parameter-efficient transfer learning methods in the applications of large-scale foundation models. The most popular method in this field is **prompt tuning** (Liu et al., 2021a), which demonstrates success in natural language processing (Li and Liang, 2021; Liu et al., 2021c; Lester et al., 2021; Liu et al., 2021b; He et al., 2021a; Gu et al., 2022) and computer vision (Jia et al., 2022; Du et al., 2022; Zhou et al., 2021, 2022). In comparison with finetuning, prompt tuning only tunes pretrained models by a trivial amount of parameters (e.g., 1%). Prompt

Findings of the Association for Computational Linguistics: ACL 2023, pages 402–416
July 9-14, 2023 ©2023 Association for Computational Linguistics

tuning freezes most parameters of the pretrained model and only tunes several prompt embeddings, as well as the output layer if necessary. Recent advances have shown that prompt tuning can help pretrained models achieve comparable performance with finetuning across different downstream tasks, including natural language understanding and generation, image classification, etc. However, the studies on the parameter-efficient transfer methods for multimodal pretrained models, especially the unified multimodal pretrained models, are still scarce. Furthermore, along with the trend of model scaling in unified multimodal pretrained models, how to tune such models cost-effectively should be a significant topic of research in multimodal pretraining.

This work fills in the void and takes the lead to explore prompt tuning for the unified multimodal pretrained models. We propose OFA-PT, an implementation of prompt tuning based on the recently open-sourced unified multimodal pretrained model OFA (Wang et al., 2022a). To be more specific, in the stage of downstream transfer, we insert a sequence of learnable embeddings to each layer of the encoder and decoder, and only tune those embeddings while keeping the parameters of the pretrained model frozen. For the rest of the setups, we use the same finetuning procedures, which transform data to the format for sequence-to-sequence learning and train the model with maximum likelihood estimation for optimization. In comparison with finetuning, the number of tunable parameters (~1% of the total) for prompt tuning is much smaller than that of finetuning, leading to fewer computation costs, e.g., memory.

Through extensive experiments we observe that the parameter-efficient prompt tuning is able to help the pretrained model achieve comparable performance with finetuning across 4 multimodal downstream tasks, spanning from understanding to generation. To analyze the differences between finetuning and prompt tuning, we follow the assumption that prompt tuning with most parameters in the pretrained model frozen should induce model robustness. We experiment on the tuning methods with adversarial attack and observe phenomena consistent with the hypothesis. To take a step further, this study delves into the implementation details and investigate whether experimental factors, e.g., the prompt length, prompt depth, and reparameterization, could saliently influence the final downstream performance. We find that in general a longer prompt length (longer than 20 tokens) is a preferable choice, and our experiments show that 64 should be favored in most cases as a longer prompt sequence will not only increase the computation costs but also incur performance degradation. Also, we show that reparameterizaton with additional trainable parameters cannot introduce significant improvements in downstream performance.

2 Method

This section introduces the details of our proposed method. It provides the detailed implementation of prompt tuning on a unified multimodal pretrained model. The overall framework is illustrated in Figure 1.

2.1 Preliminaries

We select the unified sequence-to-sequence framework as it unifies understanding and generation tasks, and we specifically implement prompt tuning on the recently open-sourced state-of-the-art model OFA* (Wang et al., 2022a). In brief, it is built with a Transformer-based (Vaswani et al., 2017) encoder-decoder framework.

Both the encoder and decoder consist of Transformer layers. To be more specific, an encoder layer consists of a multi-head self attention and a point-wise Feed-Forward Network (FFN). To build a connection between the encoder and decoder, the Transformer decoder layer additionally contains a cross-attention module in comparison with the encoder layer. The cross-attention is essentially multi-head attention, where the keys K and values V are the transformation of the encoder output states, instead of the inputs. Such architecture can handle tasks that provide inputs of the sequence-to-sequence format.

In this work, we focus on prompt tuning for the transfer of the multimodal pretrained model. We leave the prompt learning in the stage of pretraining to the future work.

2.2 Prompt Tuning for Multimodal Pretrained Models

In the following, we introduce our implementation details of prompt tuning on the sequence-to-sequence multimodal pretrained model. Note that

*https://github.com/OFA-Sys/OFA License: Apache-2.0

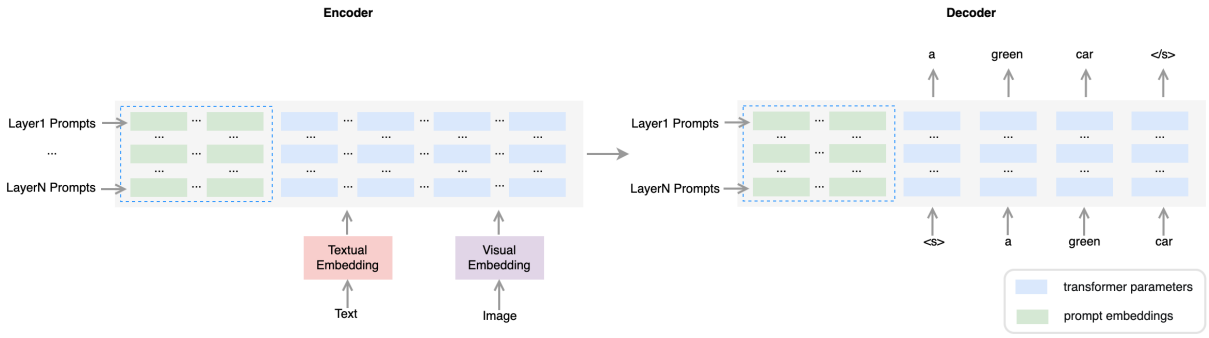

Figure 1: **Model overview.** An illustration of our multimodal prompt tuning architecture. Specifically, for the encoder and decoder, we add tunable prompt embeddings to each layer.

our method can extend to other generative multimodal pretrained models, e.g., BERT-like models.

Basic Implementation We focus on implementing prefix tuning (Li and Liang, 2021; Liu et al., 2021b) based on its outstanding performance in either natural language understanding or generation. In comparison with the other prompt tuning methods, e.g., P-Tuning (Liu et al., 2021c), Prompt Tuning (Lester et al., 2021), PPT (Gu et al., 2022), adding soft prompt embeddings to each layer demonstrates enhanced training stability and improved downstream task performance even on relatively small models. Specifically, for the encoder and decoder, we add tunable prompt embeddings to each layer. Formally, we refer the pretrained model to a function $\mathcal{M}(\cdot)$, and the generation function of the prompt embeddings to $\mathcal{G}(\cdot)$. The formulation is demonstrated below:

$$y = \mathcal{M}(\mathcal{G}(L, l), x), \tag{1}$$

where x refers to the multimodal inputs, L refers to the number of layers, and l refers to the prompt length, which should be predefined by a hyperparameter. At each layer, we prefix soft prompt embeddings $p^{(i)}$ to the input hidden states $h^{(i)}$ Note that we only prefix prompt embeddings at Transformer layers. In the simplest practice, the prompt generator $\mathcal{G}$ is a sparse embedding matrix of $\mathbb{R}^{L \times l \times h}$, and we select the corresponding embedding at the i-th index and the j-th layer as the prompt embedding. Below we provide an illustration of some more complex implementations, and we compare those methods in this study.

In the downstream tuning process, we only tune the newly added prompt embeddings at each layer and keep the parameters of the large pretrained model frozen. Therefore, while there are only a small amount of parameters that need to be updated, e.g., 1%, the computation costs are far fewer than those of finetuning.

Reparameterization Except for the simplest implementation of adding a sparse embedding matrix at each layer, a more complex one should be adding an encoder, e.g., an MLP layer, to reparameterize prompt embeddings. We also investigate the influence of reparameterization in this context.

Prompt Length Similar to previous studies (Li and Liang, 2021; Liu et al., 2021b), we find that the length of prompt embeddings make a great difference in different downstream tasks. In this study, we investigate how this factor imposes influence on model performance in different downstream tasks.

Prompt Depth To investigate the impacts of the place of prompt embedding insertion, we delve into the issue of prompt depth. Specifically, we simplify it to adding prompt embeddings to the encoder or decoder only, as well as to both modules.

3 Experiments

To validate the effectiveness of prompt tuning for multimodal pretrained models, we conduct experiments on 5 cross-modal tasks. Specifically, we experiment on cross-modal generation tasks, including referring expression comprehension and image captioning, and cross-modal understanding tasks, including visual entailment, image captioning, and visual question answering (VQA). We use the commonly used base-size and large-size models for the experiments, whose sizes are around 180M and 470M respectively. We provide more details about the experimental setups in the Appendix A.1.

Model	RefCOCO			RefCOCO+			RefCOCOg		COCO Captions			
	val	testA	testB	val	testA	testB	val-u	test-u	B@4	M	C	S
Base-size Models												
OFA_{Base}	88.48	90.67	83.30	81.39	87.15	74.29	82.29	82.31	41.00	30.90	138.2	24.20
$\text{OFA-PT}_{\text{Base}}$	84.53	85.21	77.36	76.34	81.44	67.68	75.61	76.57	39.70	30.10	134.2	23.50
Large-size Models												
$\text{OFA}_{\text{Large}}$	90.05	92.93	85.26	85.80	89.87	79.22	85.89	86.55	42.40	31.50	142.2	24.50
$\text{OFA-PT}_{\text{Large}}$	90.05	92.31	85.59	84.54	89.40	77.77	85.27	85.89	41.81	31.51	141.4	24.42

Table 1: Experimental results on RefCOCO, RefCOCO+, RefCOCOg, and COCO Image Captioning. For the base-size model, OFA-PT significantly underperforms the finetuned OFA, but for the large-size model, OFA-PT is able to achieve comparable performance.

Model	SNLI-VE		VQA	
	dev	test	test-dev	test-std
Base-size Models				
OFA_{Base}	89.30	89.20	78.00	78.10
$\text{OFA-PT}_{\text{Base}}$	88.18	88.59	74.31	74.47
Large-size Models				
$\text{OFA}_{\text{Large}}$	90.30	90.20	80.40	80.70
$\text{OFA-PT}_{\text{Large}}$	90.04	90.12	78.30	78.53

Table 2: Experimental results of methods on multimodal understanding benchmark datasets, SNLI-VE and VQA.

3.1 Datasets & Metrics

Referring Expression Comprehension We conduct experiments on the 3 subtasks of referring expression comprehension, namely RefCOCO, RefCOCO+, and RefCOCOg (Yu et al., 2016; Mao et al., 2016). This task requires the model to generate a correct bounding box that answers the given text query on a provided image. We use Acc@0.5 as the evaluation metric.

Image Captioning We evaluate the image captioning capability of our method on the Microsoft COCO Image Captioning dataset (Chen et al., 2015). In this task, the model should generate a description that corresponds to the information of the given image. We use BLEU@4 (Papineni et al., 2002), METEOR (Lavie and Agarwal, 2007), CIDEr (Vedantam et al., 2015), and SPICE (Anderson et al., 2016) as the evaluation metrics.

Visual Entailment To evaluate the performance of entailment, we implement the experiments on SNLI-VE (Xie et al., 2019). Given an image and a text, the model should figure out their relations, whether they are entailment, contradiction, or neutrality. We follow the setups in Wang et al. (2022a) and add the given premise to the input. We use accuracy as the evaluation metric.

VQA We implement our experiments on VQA 2.0 (Antol et al., 2015; Goyal et al., 2017). This task requires the model to generate the correct answer based on an image and a question about certain information on the image. Following Wang et al. (2022a), we use the all-candidate evaluation, which requires the model to generate a probability for each candidate among the $3,129$ most frequent answers. We use accuracy as the evaluation metric.

3.2 Experimental Results

Below we provide the detailed experiment results, including the comparison of prompt tuning and finetuning, as well as prompt tuning and other parameter-efficient tuning methods.

Comparison with Finetuning We demonstrate the experimental results of the 4 tasks in Table 1 and Table 2. In general, for the base-size model, OFA-PT underperforms the original finetuned OFA by significant margins, but for the large-size model, OFA-PT is able to achieve comparable performance. To be more specific, in the evaluation of referring expression comprehension, for the base-size model, prompt tuning significantly underperforms finetuning by lagging behind a large margin of 5.64 on average across RefCOCO, RefCOCO+, and RefCOCOg, but for the large-size model, prompt tuning only slightly underperforms finetuning by a small margin of 0.59. In the evaluation of image captioning, for the base-size model, OFA-PT underperforms the finetuned OFA by a margin of 4.0, but for the large-size model, the performance gap is only 0.8. In the evaluation of visual entailment, the gap between the algorithms is closer, which is around 0.17. In the evaluation of VQA, for the base-size model the performance gap is 3.63 be-

Method	RefCOCO			RefCOCO+			RefCOCOg		COCO Captions			
	val	testA	testB	val	testA	testB	val-u	test-u	B@4	M	C	S
OFA-Bitfit	89.61	92.20	84.91	82.60	88.08	75.16	84.66	84.68	41.02	30.92	138.8	24.23
OFA-Adapter	90.01	92.30	85.02	83.79	88.93	76.09	85.10	85.45	41.38	31.16	139.5	24.30
OFA-PT	90.05	92.31	85.59	84.54	89.40	77.77	85.27	85.89	41.81	31.51	141.4	24.42

Table 3: Evaluation of different parameter-efficient tuning methods using large-size models on multimodal generation tasks. We find that OFA-PT can generally outperform OFA with Bitfit and Adapter.

Method	SNLI-VE		VQA	
	dev	test	test-dev	test-std
OFA-Bitfit	89.70	89.42	78.23	78.44
OFA-Adapter	89.84	89.78	78.27	78.47
OFA-PT	90.04	90.12	78.30	78.53

Table 4: Evaluation of different parameter-efficient tuning methods using large-size models on multimodal understanding tasks. OFA-PT outperforms the baselines significantly.

tween prompt tuning and finetuning, and for the large-size model the gap is 2.17 on the test-std set. Different from the other tasks, even in the experiments on the large-size model, the gap is still significant. We hypothesize that it is still necessary to search a better hyperparameter setup for this task due to the sensitivity of prompt tuning to hyperparameters.

Comparison with Other Parameter-Efficient Tuning Methods We additionally add a comparison with two parameter-efficient tuning methods, namely Adapter (Houlsby et al., 2019) and BitFit (Zaken et al., 2022) to test whether prompt tuning is the best solution of light-weight transfer. Table 3 and 4 demonstrate the results of different light-weight tuning methods implemented on the aforementioned datasets. In all the downstream tasks, OFA-PT surpasses the performance of OFA with Adapter or BitFit. The results reflect the simple but effective prompt tuning over other parameter-efficient tuning baselines. We suppose that changes in biases and adding intermediate layers might be conflicted with the complex architectural designs of the unified multimodal pretrained model, whereas the simple prepended learnable prefixes have separate components, e.g., weights, positional embeddings, etc., which can result in easier training with less human efforts on hyperparameter tuning.

3.3 Analyses

In this section, we move forward to analyzing prompt tuning in multimodal pretraining. Specifically, we examine the robustness of prompt tuning based on the assumption that keeping most parameters of the pretrained model frozen should lead to improved robustness to adversarial attack. Also, we evaluate how different setups of prompt tuning, say the prompt length, the depth of prompt, and reparameterization, influence the downstream performance, and try to provide a recommended setup for consistently better performance.

Robustness Analysis To test whether the multimodal pretrained model with prompt tuning for downstream transfer is robust, we conduct experiments of adversarial attack for the examination. Adversarial attack was first proposed in computer vision, which revealed the vulnerability of deep learning models. The most common adversarial attack methods in computer vision are gradient-based methods, such as FGSM (Goodfellow et al., 2014), PGD (Madry et al., 2017), MIM (Dong et al., 2017) and SI (Lin et al., 2019). Most of the typical unimodal adversarial attack on tasks are gradient-based methods. Among them, we select FGSM, which requires only one step of gradient computation on text and image embeddings. Experimental results are demonstrated in Figure 2. OFA-PT consistently demonstrates better robustness in comparison with the finetuned OFA across all tasks. This confirms our hypothesis and also shows one significant advantage of prompt tuning not reflected in the standard evaluation. In practice, if model vulnerability is a issue that matters, we recommend the application of prompt tuning or the robust prefix tuning framework (Yang and Liu, 2022) that demonstrates effectiveness in tuning pretrained language models for the enhanced robustness without significant performance degradation

Prompt Length To study the effects of the prompt length on the final downstream perfor-

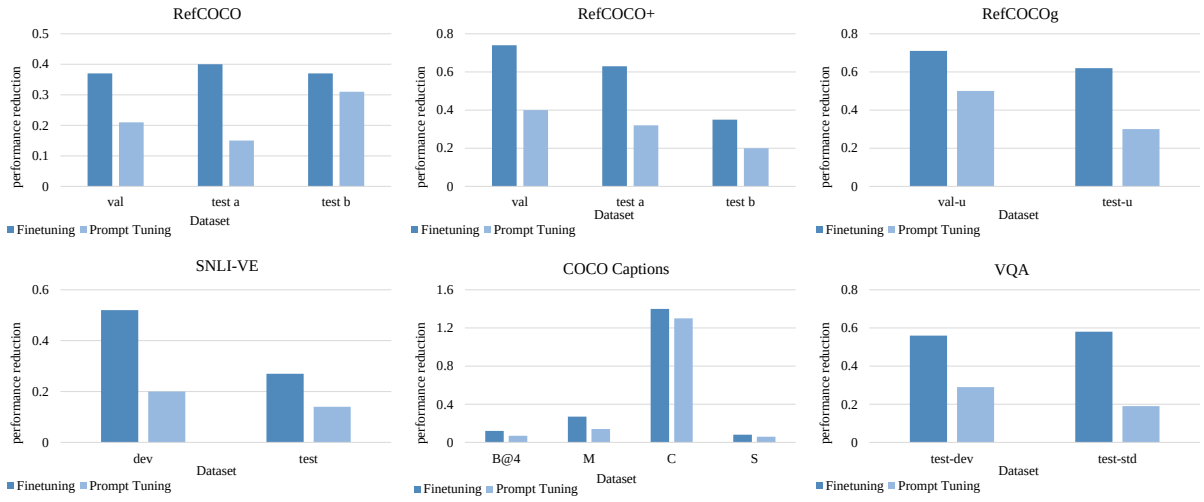

Figure 2: Experimental results on adversarial attack using large-size models. We discover that in the scenario of adversarial attack prompt tuning suffers from lower performance degradation across the tasks.

Method	RefCOCO			RefCOCO+			RefCOCOg		COCO Captions			
	val	testA	testB	val	testA	testB	val-u	test-u	B@4	M	C	S
Enc	89.48	91.71	84.98	84.50	89.22	77.71	85.07	85.58	41.39	31.08	141.1	24.34
Dec	88.90	91.28	84.32	83.46	88.24	76.82	84.54	85.02	40.08	30.43	140.8	24.06
EncDec	90.05	92.31	85.59	84.54	89.40	77.77	85.27	85.89	41.81	31.51	141.4	24.42

Table 5: Evaluation of different prompt insertion methods on multimodal understanding tasks. We specifically evaluate the performance of prompt tuning with prompts inserted to the encoder only, to the decoder only, or to both the encoder and decoder.

Method	SNLI-VE		VQA	
	dev	test	test-dev	test-std
Enc	89.64	89.70	78.10	78.26
Dec	88.56	88.71	77.84	78.03
EncDec	90.04	90.12	78.30	78.53

Table 6: Evaluation of different prompt insertion methods on multimodal understanding tasks. We specifically evaluate the performance of prompt tuning with prompts inserted to the encoder only, to the decoder only, or to both the encoder and decoder.

mance, we evaluate the prompt tuning performance on the downstream tasks with a prompt length selected from $\{10, 16, 30, 64, 100, 120\}$. As shown in Figure 3, a general tendency is that a longer prompt length with more parameters to tune can encourage improvements in downstream performance across the tasks. However, we observe diminishing marginal utility and a prompt too long may even negatively impact the performance. Although the best prompt length for tasks are different, we empirically advise that the length of 64 tokens can achieve a better performance on average. See Appendix A.2 for more details.

Prompt Depth As we base our implementation on the encoder-decoder model, we intuitively assume that where to insert prompt embeddings matters the performance. To simplify this issue, in our practice, we evaluate the performance of inserting prompts to the encoder only, to the decoder only, or to both the encoder and decoder. Experimental results are demonstrated in Table 5 and 6. We find that it is best to insert prompts to every layer of the whole Transformer model, though compared with the other alternatives it is less computation-efficient. In the comparison between insertion to the encoder only and to the decoder only, we observe that the former solution leads to a significantly better results across multiple downstream tasks. This suggests that the insertion of prompts to the bottom layers might contribute more to the success of downstream transfer.

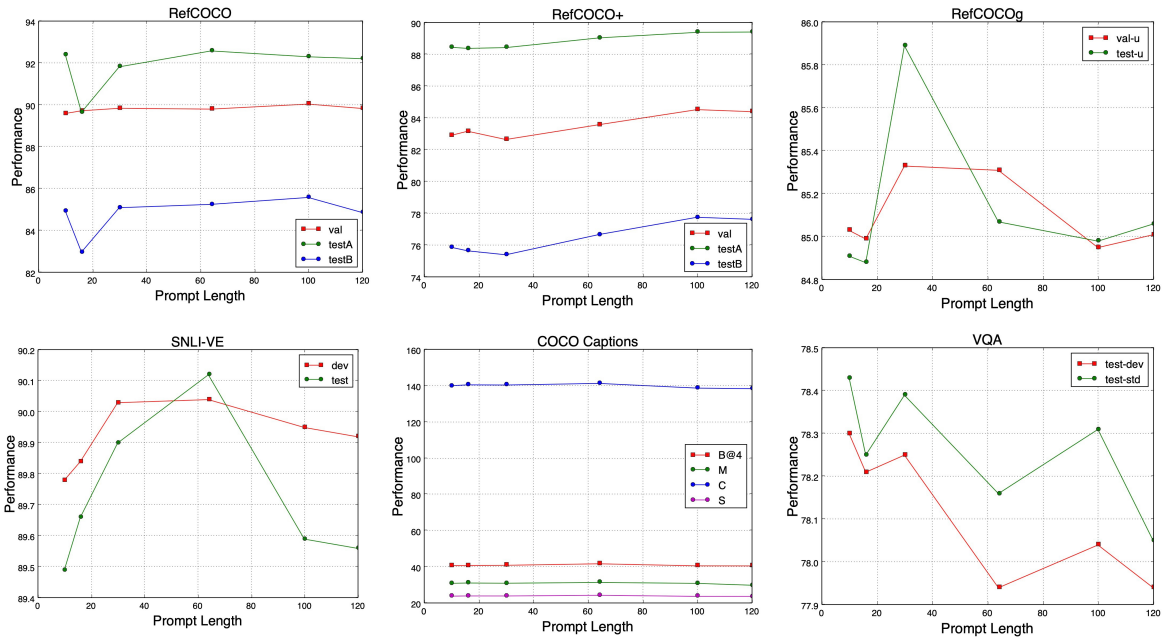

Figure 3: **Analysis of prompt lengths on multimodal downstream tasks.** We observe that increasing prompt lengths can generally bring performance improvements. Yet it cannot extend to all scenarios, and the increase might meet saturation. Based on the experimental results, we recommend 64 for the prompt length as it helps the model achieve the average best results across tasks.

Method	RefCOCO			RefCOCO+			RefCOCOg		COCO Captions			
	val	testA	testB	val	testA	testBxq	val-u	test-u	B@4	M	C	S
w/o MLP	90.05	92.31	85.59	84.54	89.40	77.77	85.27	85.89	41.81	31.51	141.4	24.42
w/ MLP	90.12	92.56	85.63	84.83	89.65	77.94	85.42	86.01	41.67	31.48	140.7	24.40

Table 7: Ablation study results of multimodal generation tasks on reparameterization using large-size models.

Method	SNLI-VE		VQA	
	dev	test	test-dev	test-std
w/o MLP	90.04	90.12	78.30	78.53
w/ MLP	89.98	90.02	78.26	78.48

Table 8: Ablation study results of multimodal understanding tasks on reparameterization using large-size models.

Reparameterization Empirically, directly updating the trainable embeddings leads to unstable optimization and a slight drop in performance. Prior work usually leveraged an encoder, e.g., an MLP (Li and Liang, 2021), to reparameterize the trainable embeddings. We evaluate the performance of reparameterization, and we demonstrate the experimental results in Table 7 and 8. For generation tasks, e.g., RefCOCO and RefCOCOg, MLP brings consistent improvements. For understanding tasks, e.g., SNLI-VE and VQA, MLP leads to relatively negative impacts. Thus we cannot come to a conclusion about which should be a preferable one. To achieve better performance on a specific dataset, it is still necessary to make an attempt on both methods.

4 Related Work

In this section, we include the review of multimodal pretraining as well as prompt tuning.

4.1 Multimodal Pretraining

The rise of vision & language pretraining started from the transfer of BERT (Devlin et al., 2019) to cross-modal representation learning. A series of studies (Lu et al., 2019; Su et al., 2020; Tan and Bansal, 2019; Chen et al., 2020d; Li et al., 2019) introduced BERT to multimodal pretraining.

The key idea of such transfer is that the powerful Transformer model can handle visual and linguistic information simultaneously. To take a step forward, recent studies have turned their focuses to the encoder-decoder framework, which is adaptive to both cross-modal understanding and generation, a series of encoder-decoder-based models or similar models that can perform sequence-to-sequence learning (Dong et al., 2019) have achieved new state-of-the-art performance across the downstream tasks (Wang et al., 2021; Li et al., 2022; Wang et al., 2022a; Yu et al., 2022; Wang et al., 2022b; Chen et al., 2022). Furthermore, these recent state-of-the-art models have unified different tasks concerning multiple modality combinations into a single framework and pretrained model. Also, we have witnessed similar trends in large language models that consistently scaling unified multimodal pretraiend model can lead to predictable performance improvement (Wang et al., 2022a,b; Chen et al., 2022). This indicates that prompt tuning should be a perfect combination with the recent unified multimodal pretrained model and it can unleash the power of large-scale pretrained models with fewer computation costs than the conventional finetuning.

4.2 Prompt-based Learning

Brown et al. (2020) illustrated that large-scale pretrained models can learn from the context and perform few-shot and zero-shot learning with the prompts of task instruction or a few task examples. This new paradigm raised attention of researchers in how to leverage pretrained models without tuning all the parameters, which is expensive in computation costs. Instead of using hard prompts by handcrafting, Li and Liang (2021) demonstrated that only tuning soft prompt embeddings at each layer is sufficient for the pretrained model to achieve competitive performance in natural language generation, and later a number of studies showed that prompt tuning can be essentially effective for low-resource scenarios (Liu et al., 2021c; Gu et al., 2022; Sun et al., 2022b) and it can even achieve comparable performance with finetuning (Lester et al., 2021; Liu et al., 2021b). Following this trend, a series of modification to prompts and adapters (Hu et al., 2022; He et al., 2021a; Jiang et al., 2022; Sun et al., 2022a) for improvements in performance or training efficiency have emerged and made prompt tuning a heated topic in the whole NLP community.

Recent prompt tuning methods for multimodal pretrained models mostly serve for CLIP-like models (Zhou et al., 2021, 2022; Rao et al., 2021). Similarly, researchers tried to incorporate adapters to CLIP and also achieved satisfactory performance (Gao et al., 2021; Zhang et al., 2021). Except for prompt tuning for CLIP-like models, another line of work explored visual prompts for frozen language models. Tsimpoukelli et al. (2021) showed that when there is a powerful large pretrained language model, a visual encoder for prompt tuning is sufficient for multimodal few-shot learning. To take a step forward, Alayrac et al. (2022) proposed Flamingo, a colossal multimodal model that enables in-context learning. It could achieve state-of-the-art performance in a series of cross-modal downstream tasks in either few-shot or full-shot learning scenarios. Such tremendous success indicates the strong potential of prompt tuning in multimodal pretraining.

5 Conclusion

In this work, we explore prompt tuning for unified multimodal pretrained models. Specifically, we propose OFA-PT, which is an implementation of prefix tuning, a simple but effective prompt tuning method, on the recently open-sourced SoTA model OFA. Through extensive experiments, we demonstrate that the unfiied multimodal pretrained model with the parameter-efficient prompt tuning can achieve comparable performance with the fine-tuned model, but with fewer parameters to tune (e.g., 1%), and prompt tuning can surpass other light-weight tuning methods, e.g., Adapter and BitFit. Through our analysis, we figure out a significant advantage of prompt tuning about its robustness against adversarial attack. Furthermore, we provide a comprehensive analysis about the influence of prompt tuning setups, including the prompt length, prompt depth, and reparameterization. Potentially prompt tuning can be an alternative to finetuning, but still, there are some salient limitations in this method, e.g., slow convergence and training instabilities. We hope that future studies in this field can alleviate the aforementioned problems and thus promote the application of prompt tuning.

Limitations

This section disccuses the limitations of prompt tuning for the unified multimodal pretrained mod-

els, and point out some directions for future work.

One limitation of prompt tuning in this setup is the sensitivity to hyperparameter tuning. It is difficult to search for a suitable hyperparamter setup. The hyperparameter tuning experience in finetuning is not suitable for prompt tuning. Fortunately, we find that prompt tuning for generative multimodal pretrained models is not as sensitive to hyperparameters as prompt tuning for pretrained language models. We provide details of hyperparameter setups in Appendix A.1.

Another limitation of prompt tuning in this setup is slow convergence. Though prompt tuning has noticeable advantages in training efficiency, it costs at least 40 epochs for prompt tuning to achieve the nearly best performance on some datasets (e.g., RefCOCO). A larger number of training epochs may incur more computation costs though prompt tuning has an advantage in training efficiency compared with finetuning. We demonstrate more details in Appendix A.2. This indicates that finding a better solution for fast and stable convergence is also important besides reaching comparable or even improved performance over the conventional finetuning.

Despite the aforementioned limitations, prompt tuning demonstrates significantly better robustness against adversarial attack. In the future, we should pay more attention to this merit and find ways to leverage it.

Ethics Statement

We base our method on an existing multimodal pretrained model, which is capable of vision-language understanding and generation. Thus, there exist potential risks in AI-generated contents. Additionally, as our method only finetunes only a small amount of parameters of the pretrained models, we lack control of the output model, which may generate harmful contents. These results may possibly be attributed to the noise in the pretraining data. In the future research, it is essential to study how to increase the controllability on the generation while most parameters of the output model are originated from the pretrained model.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. 2022. Flamingo: a visual language model for few-shot learning. CoRR, abs/2204.14198.

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. SPICE: semantic propositional image caption evaluation. In ECCV 2016, volume 9909 of Lecture Notes in Computer Science, pages 382–398. Springer.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015. VQA: visual question answering. In ICCV 2015, pages 2425–2433. IEEE Computer Society.

Hangbo Bao, Li Dong, and Furu Wei. 2021. Beit: Bert pre-training of image transformers. arXiv preprint arXiv:2106.08254.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In NeurIPS 2020.

Ting Chen, Simon Kornblith, Kevin Norouzi, Mohammad Swersky, and Geoffrey Hinton. 2020a. Big self-supervised models are strong semi-supervised learners. In NeurIPS 2020, pages 10466–10478. PMLR.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020b. A simple framework for contrastive learning of visual representations. In ICML 2020, pages 1597–1607. PMLR.

Xi Chen, Xiao Wang, Soravit Changpinyo, A. J. Piergiovanni, Piotr Padlewski, Daniel M. Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Nan Ding, Keran Rong, Hassan Akbari, Gaurav Mishra, Linting Xue, Ashish V. Thapliyal, James Bradbury, Weicheng Kuo, Mojtaba Seyedhosseini, Chao Jia, Burcu Karagol Ayan, Carlos Riquelme, Andreas Steiner, Anelia Angelova, Xiaohua Zhai, Neil Houlsby, and Radu Soricut. 2022. Pali: A jointly-scaled multilingual language-image model. ArXiv, abs/2209.06794.

Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. 2020c. Improved baselines with momentum contrastive learning. arXiv preprint arXiv:2003.04297.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C. Lawrence Zitnick. 2015. Microsoft COCO captions: Data collection and evaluation server. CoRR, abs/1504.00325.

Xinlei Chen and Kaiming He. 2021. Exploring simple siamese representation learning. In CVPR 2021, pages 15750–15758.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020d. UNITER: universal image-text representation learning. In ECCV 2020, volume 12375 of Lecture Notes in Computer Science, pages 104–120. Springer.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways. CoRR, abs/2204.02311.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In NAACL-HLT 2019, pages 4171–4186. Association for Computational Linguistics.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In NeurIPS 2019, pages 13042–13054.

Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. 2017. Boosting adversarial attacks with momentum. CoRR, abs/1710.06081.

Yu Du, Fangyun Wei, Zihe Zhang, Miaojing Shi, Yue Gao, and Guoqi Li. 2022. Learning to prompt for open-vocabulary object detection with vision-language model. CoRR, abs/2203.14940.

Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. 2021. Clip-adapter: Better vision-language models with feature adapters. CoRR, abs/2110.04544.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. CoRR, abs/1412.6572.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In CVPR 2017, pages 6325–6334. IEEE Computer Society.

Yuxian Gu, Xu Han, Zhiyuan Liu, and Minlie Huang. 2022. PPT: pre-trained prompt tuning for few-shot learning. In ACL 2022, pages 8410–8423. Association for Computational Linguistics.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2021a. Towards a unified view of parameter-efficient transfer learning. CoRR, abs/2110.04366.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. 2021b. Masked autoencoders are scalable vision learners. arXiv preprint arXiv:2111.06377.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In ICML 2019, volume 97 of Proceedings of Machine Learning Research, pages 2790–2799. PMLR.

Shengding Hu, Ning Ding, Huadong Wang, Zhiyuan Liu, Jingang Wang, Juanzi Li, Wei Wu, and Maosong Sun. 2022. Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification. In ACL 2022, pages 2225–2240. Association for Computational Linguistics.

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge J. Belongie, Bharath Hariharan, and Ser-Nam Lim. 2022. Visual prompt tuning. CoRR, abs/2203.12119.

Yuezihan Jiang, Hao Yang, Junyang Lin, Hanyu Zhao, An Yang, Chang Zhou, Hongxia Yang, Zhi Yang, and Bin Cui. 2022. Instance-wise prompt tuning for pretrained language models. CoRR, abs/2206.01958.

Alon Lavie and Abhaya Agarwal. 2007. METEOR: an automatic metric for MT evaluation with high levels of correlation with human judgments. In WMT@ACL 2007, pages 228–231. Association for Computational Linguistics.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In EMNLP 2021, pages 3045–3059. Association for Computational Linguistics.

Gen Li, Nan Duan, Yuejian Fang, Daxin Jiang, and Ming Zhou. 2019. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. CoRR, abs/1908.06066.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven C. H. Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In International Conference on Machine Learning.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In ACL/IJCNLP 2021, pages 4582–4597. Association for Computational Linguistics.

Jiadong Lin, Chuanbiao Song, Kun He, Liwei Wang, and John E. Hopcroft. 2019. Nesterov accelerated gradient and scale invariance for adversarial attacks. CoRR, abs/1908.06281.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021a. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. CoRR, abs/2107.13586.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2021b. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. CoRR, abs/2110.07602.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021c. GPT understands, too. CoRR, abs/2103.10385.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. CoRR, abs/1907.11692.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In NeurIPS 2019, pages 13–23.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2017. Towards deep learning models resistant to adversarial attacks. CoRR, abs/1706.06083.

Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L. Yuille, and Kevin Murphy. 2016. Generation and comprehension of unambiguous object descriptions. In CVPR 2016, pages 11–20. IEEE Computer Society.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In ACL 2002, pages 311–318.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res., 21:140:1–140:67.

Yongming Rao, Wenliang Zhao, Guangyi Chen, Yansong Tang, Zheng Zhu, Guan Huang, Jie Zhou, and Jiwen Lu. 2021. Denseclip: Language-guided dense prediction with context-aware prompting. CoRR, abs/2112.01518.

Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. 2020. VL-BERT: pre-training of generic visual-linguistic representations. In ICLR 2020. OpenReview.net.

Tianxiang Sun, Zhengfu He, Hong Qian, Xuanjing Huang, and Xipeng Qiu. 2022a. Bbtv2: Pure black-box optimization can be comparable to gradient descent for few-shot learning. CoRR, abs/2205.11200.

Tianxiang Sun, Yunfan Shao, Hong Qian, Xuanjing Huang, and Xipeng Qiu. 2022b. Black-box tuning for language-model-as-a-service. In ICML 2022, volume 162 of Proceedings of Machine Learning Research, pages 20841–20855. PMLR.

Hao Tan and Mohit Bansal. 2019. LXMERT: learning cross-modality encoder representations from transformers. In EMNLP-IJCNLP 2019, pages 5099–5110. Association for Computational Linguistics.

Maria Tsimpoukelli, Jacob Menick, Serkan Cabi, S. M. Ali Eslami, Oriol Vinyals, and Felix Hill. 2021. Multimodal few-shot learning with frozen language models. In NeurIPS 2021, pages 200–212.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In NeurIPS 2017, pages 5998–6008.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In CVPR 2015, pages 4566–4575. IEEE Computer Society.

Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. 2022a. Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. CoRR, abs/2202.03052.

Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Mohammed, Saksham Singhal, Subhojit Som, and Furu Wei. 2022b. Image as a foreign language: Beit pretraining for all vision and vision-language tasks. ArXiv, abs/2208.10442.

Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. 2021. Simvlm: Simple visual language model pretraining with weak supervision. CoRR, abs/2108.10904.

Ning Xie, Farley Lai, Derek Doran, and Asim Kadav. 2019. Visual entailment: A novel task for fine-grained image understanding. CoRR, abs/1901.06706.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In NeurIPS 2019, pages 5754–5764.

Zonghan Yang and Yang Liu. 2022. On robust prefix-tuning for text classification. ArXiv, abs/2203.10378.

Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. 2022. Coca: Contrastive captioners are image-text foundation models. ArXiv, abs/2205.01917.

Licheng Yu, Patrick Poirson, Shan Yang, Alexander C. Berg, and Tamara L. Berg. 2016. Modeling context in referring expressions. In ECCV 2016, volume 9906 of Lecture Notes in Computer Science, pages 69–85. Springer.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In ACL 2022, pages 1–9. Association for Computational Linguistics.

Renrui Zhang, Rongyao Fang, Wei Zhang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. 2021. Tip-adapter: Training-free clip-adapter for better vision-language modeling. CoRR, abs/2111.03930.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. 2021. Learning to prompt for vision-language models. CoRR, abs/2109.01134.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. 2022. Conditional prompt learning for vision-language models. CoRR, abs/2203.05557.

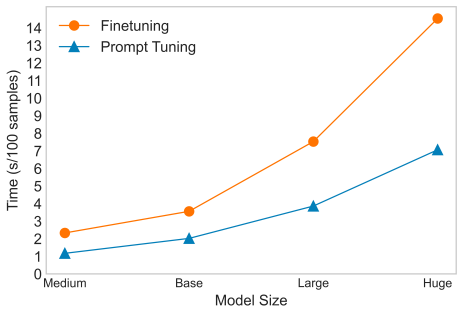

Figure 4: **Efficiency of different tuning methods.** We report the spent time per 100 samples of finetuning and prompt tuning on RefCOCO.

A Appendix

A.1 Experimental Setups

Referring Expression Comprehension Referring expression comprehension requires models to locate an image region described by a language query. We perform experiments on RefCOCO (Yu et al., 2016), RefCOCO+ (Yu et al., 2016), and RefCOCOg (Mao et al., 2016). We report the standard metric Acc@0.5 on the validation and test sets. For finetuning, the batch size is set to 128, the learning rate is set to 0.03, and the prompt length varies from 10–120. For Adapter, the batch size is set to 128 and the learning rate is set to $5e-5$. For Bitfit, the batch size is set to 128 and the learning rate is set to 0.001.

Visual Entailment Visual entailment requires the model to evaluate the semantic relation between the given image and text, i.e., entailment, neutrality, or contradiction. We perform experiments on the SNLI-VE (Xie et al., 2019) dataset. We report accuracy on both dev and test sets. The model is finetuned with a learning rate of 0.03 and a batch size of 128. The prompt length varies from 10–120. For Adapter, the batch size is set to 128 and the learning rate is set to $5e-5$. For Bitfit, the batch size is set to 128 and the learning rate is set to 0.001.

Image Captioning Image captioning is a standard vision & language task that requires models to generate an appropriate and fluent caption for an image. We report BLEU@4 (Papineni et al., 2002), METEOR (Lavie and Agarwal, 2007), CIDEr (Vedantam et al., 2015), and SPICE (Anderson et al., 2016) scores on the Karpathy test split. We finetune the model with a learning rate of 0.03, a batch size of 256, and a prompt length varying from 10–120. For Adapter, the batch size is set to 128 and the learn-

Method	Fintuning	Prompt Tuning
RefCOCO	40.00	77.44
SNLI-VE	80.96	164.48
COCO Captions	29.60	16.16
VQA	616.16	455.52

Table 9: Computation resource consumption of different tasks. We specifically compute the GPU-hours of both finetuning and prompt tuning on large-size models

Length	10	16	32	64	100	120
Score	91.84	91.29	91.94	92.29	92.10	91.93

Table 10: Evaluation average performance of prompt tuning on the downstream tasks with different prompt lengths.

ing rate is set to $5e-5$. For Bitfit, the batch size is set to 128 and the learning rate is set to 0.001. We only finetune the model with cross-entropy loss, without further CIDEr optimization.

Visual Question Answering Visual question answering (Antol et al., 2015; Goyal et al., 2017) is a cross-modal task that requires the models to answer the question given an image. We conduct experiments on VQA 2.0 and report the score on the test-std set. For finetuning, the batch size is set to 256 and the learning rate is set to 0.03. Exponential Moving Average (EMA) with a decay rate of 0.9999 is employed in finetuning. The prompt length varies from 10–120. For Adapter, the batch size is set to 128 and the learning rate is set to $5e-5$. For Bitfit, the batch size is set to 128 and the learning rate is set to 0.001.

A.2 Additional Experimental Results

In this section, we provide more experimental results for comprehensive understanding of the performance of prompt tuning.

Below we summarize the detailed performance of prompt tuning on the downstream tasks in the conditions of different prompt lengths. See Table 10. On average, a prompt length of 64 helps achieve the best average performance in the downstream tasks.

To evaluate the training efficiency of different methods, we experiment on the base model OFA of different sizes, spanning from 93M to 930M paramters. Figure 4 demonstrates their performance in efficiency by evaluating their used time of processing 100 samples. We find that prompt tuning consistently performs better than finetuning in training efficiency. For the huge-size model, it can perform around 2 times faster than finetuning. However, based on our observation, the advantage in training efficiency does not lead to less required computation resource. Table 9 lists the detailed computation resource consumption of both finetuning and prompt tuning. Specifically, we compute the computation resource consumption by calculating the GPU-hours of finetuning and prompt tuning on different tasks. We find that for image captioning and VQA, prompt tuning consumes less resource, but for the other tasks prompt tuning adversely consumes more. It reflects that for tasks similar to pretraining tasks, especially those with more data in the pretraining stage, prompt tuning is able to outperform finetuning, but for others, prompt tuning even incurs more carbon footprints. This indicates that the real computation resource consumption for downstream transfer should be an important issue in the field of prompt tuning and the solution to this problem can further the developments of the application.

A.3 Experimental Configuration

The experiments are conducted on Linux servers equipped with an Intel(R) Xeon(R) Platinum CPU @2.90GHz, 1024GB RAM and 8 NIVDIA A100-80GB GPUs. We run our experiments on 32 A100 GPUs. All models are implemented in Pytorch version 1.8.1 and Python 3.7.4.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Subsection Limitations

☑ A2. Did you discuss any potential risks of your work?
Subsection Ethics Statement

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Subsection 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☒ Did you use or create scientific artifacts?

Left blank.

☐ B1. Did you cite the creators of artifacts you used?
No response.

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
No response.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
No response.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
No response.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
No response.

☐ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
No response.

C ☑ Did you run computational experiments?

Subsection 4

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Subsection A.2 A.3

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Subsection 4.3

☒ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Run once, the variance is small and negligible.

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Subsection 3 Subsection A.1

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.