Efficient Prompt Tuning for Vision and Language Models

Bing Li[1*][0009-0009-3858-173X], Feng Li[2*][0009-0005-9683-9461], Shaokun Gao[4][0009-0002-6350-5770]

Qile Fan[3*][0009-0003-0551-8490], Yuchen Lu[5][0009-0004-5563-4905] , Renyu Hu[6][0009-0005-7548-2475]

Zhiyuan Zhao[7][0009-0007-7982-3374]

[1-7] Nanjing University of Posts and Telecommunications, Nanjing, China

*These authors contributed to the work equally.

cq9067@gamil.com

Abstract. Recently, large-scale pre-trained visual language models have demonstrated excellent performance in many downstream tasks. A more efficient adaptation method for different downstream tasks is prompt tuning, which fixes the parameters of the visual language model and adjusts only prompt parameters in the process of adapting the downstream tasks, using the knowledge learned by the visual language model during pre-training to solve the problems in the downstream tasks. However, the loss of the downstream task and the original loss of the visual language model are not exactly same during model training. For example, CLIP uses contrast learning loss to train the model, while the downstream image classification task uses the cross-entropy loss commonly used in classification problems. Different loss has different guiding effects on the task. The trend of the accuracy of the visual language model task during training is also different from that with the downstream task. The choice of an appropriate loss function and a reasonable prompt tuning method have a great impact on the performance of the model. Therefore, we pro-pose a more efficient method of prompt tuning for CLIP, experiments on 11 datasets demonstrate that our method achieves better performance and faster convergence in the downstream task.

Keywords: Deep Learning , Visual Language Models, CLIP , Prompt tuning , Few-shot learning.

1 Introduction

The visual language pre-training model performs well in many downstream tasks, such as CLIP [1], ALIGN [2]. An important feature of the visual language pre-training model is to map text and images into a common vector space. For example, image encoder and text encoder of CLIP model are used to extract features of images and text respectively. CLIP model utilizes the idea of contrast learning to maximize the cosine similarity be-tween matched image text pairs and minimize the cosine similarity of unmatched image text pairs. In contrast, there are usually two methods for adapting visual language pre-training models to downstream tasks, fine tuning and prompt tuning

[3]. Fine tuning pre-training models need to consume a lot of storage and computational resources to adjust the parameters of the whole model, while prompt tuning adapts downstream tasks by fixing the pre-training model parameters and adding additional trainable parameters. So prompt tuning only needs to save the parameters of the pre-trained model and add a few additional parameters for different downstream tasks [3].

The visual language model usually consists of an image encoder and a text encoder to extract image features and text features, respectively. Therefore, there are three prompt tuning methods, namely visual prompt tuning, text prompt tuning, visual and text prompt tuning. For visual prompt tuning, such as VPT [4], a small number of learnable parameters will be added to the vision transformer. For text prompt tuning, such as CoOp [5], trainable parameters are added instead of manual fixed text prompts to find the optimal solution matching the current task in a continuous parameter space. For visual and text prompt tuning, such as UPT [6], unified prompt is input to the transformer for processing and then shunted to serve as separate prompt for the image and text encoders, respectively. The above approach has made significant progress and achievements in many downstream tasks, such as few-shot learning.

However, the above approaches to prompt tuning exploration for visual language model lack consideration for downstream tasks. For example, during our replication, we found that the trend of the accuracy of the visual language model task during training was different from that with the downstream task. We illustrate this problem with an example of CLIP model adapted to downstream task of few-shot learning. In Fig.1 (left), we showed the accuracy of training CLIP with contrast learning loss. We found that when the contrast learning accuracy leveled off, the classification accuracy with few-shot learning did not fully converge, which may be due to the different training difficulty of different tasks. In Fig.1(right), we showed the change in accuracy when training the CLIP model with classification loss of few-shot learning task. We found that the classification accuracy was high, but the contrast learning accuracy was low. However, a model with good performance should perform well in both.

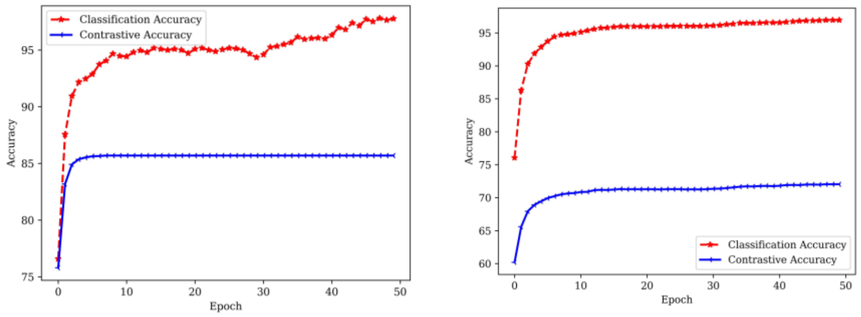

Fig. 1. The trend of accuracy during training CLIP with comparative learning loss(left). The trend of accuracy during training CLIP with classification loss(right).

Therefore, to solve the above problem, we propose a more efficient method of prompt tuning called **E**fficient **P**rompt **T**uning (**EPT**). With this approach, EPT can be better adapted to different downstream tasks and improve the performance on these tasks. The main contributions of our paper are as follows: 1) We propose a new prompt tuning method, called Efficient Prompt Tuning (EPT), for downstream task adaptation of visual language models; 2) We firstly propose incorporating downstream task loss into the prompt tuning process of visual language model; 3)We perform EPT method on 11 datasets extensive experiments to demonstrate that it outperforms all other existing prompt tuning methods.We hope that our work will stimulate more in-depth research in the field of multimodal prompt tuning.

2 Related work

At present, prompt tuning methods for visual language models are still a major challenge. In general, deep learning-based approaches can be divided into two categories: **2.1. Single-modal prompt tuning** and **2.2. Muti-modal prompt tuning**. In this section, related work from both perspectives is presented in detail.

2.1 Single-modal prompt tuning

Large-scale pre-trained models can be adapted to downstream tasks by prompt tuning. For different downstream tasks, only different prompts need to be designed [3]. Compared with fine-tuning pre-trained models, prompt engineering has a higher accuracy with less data and does not need to adjust the parameters of the whole model, saving computational resources. While the setting of prompts can greatly affect the model performance and it is a time and effort consuming task to design the prompt templates manually [5]. The current unimodal prompt tuning methods can be broadly classified into two categories: text prompt tuning and visual prompt tuning.

Prompt tuning originated from natural language processing techniques [3]. Excellent prompt tuning methods allow large-scale pre-trained models to effectively adapt to downstream tasks, such as text classification. To this end, Shin et al. proposed auto prompt based on gradient descent to find the prompt that adapts to the downstream task in a discrete space [7]. Soft prompt method proposed by Qin et al. used continuous optimizable vector space instead of the traditional hard prompt which were always fixed manual templates with single structure, circumventing the problem of poor performance on a particular corpus [8].

A common approach to image recognition problems in computer vision is to use pre-trained convolutional models to fine-tune a subset of parameters, such as classifier heads or bias terms, in order to achieve an improvement in the accuracy of the model for downstream tasks [4]. However, fine-tuning pre-trained model suffered from the problem of low accuracy. Moreover, fine-tuning the whole pre-trained model required a lot of storage resources and computational resources. There also exist some researchers in computer vision who draw inspiration from prompt tuning in NLP. For example, visual prompt tuning (VPT) proposed by Jia et al. introduced a small number of task-specific learnable parameters into the input space and froze the entire pre-trained Transformer backbone during training in the downstream tasks [4]. This approach reduced the utilization of computational resources because only a few prompt parameters need to be tuned. The current experiments demonstrated that VPT performed well in the field of few-shot learning.

2.2 Muti-modal prompt tuning

The multimodal prompt tuning technique originated from the popularity of large-scale pre-trained multimodal models. The current mainstream visual language models usually contain dual-stream and single-stream structured Transformer models, such as LXMERT [9], Oscar [10], ViLBERT [11], etc. Oscar proposed by Li et al. improved the performance of cross-modal models by increasing the recognition of picture objects and text connection between them. However, cross-modal models based on contrast learning also performed well in many tasks, such as CLIP [1] and ALIGN [2], which extracted features of different modalities by image encoder and text encoder respectively and mapped them to the same vector space and then computed the cosine similarity of different texts and images, showing excellent performance in downstream tasks, such as few-shot learning.

In the cross-modal domain, Zhou et al. proposed to use contextual optimization (CoOp) on text modalities to achieve prompt tuning of CLIP, and obtained excellent performance in the field of few-shot learning [5]. Other methods such as CoCoOp [12], DualCoOp [13] and ProGrad [14] emerged subsequently after this. However, this method does not use prompt parameters on image modality. Unified prompt tuning (UPT) [6] proposed by Zang et al. adapted the unified prompt parameters to multimodal features using Transformer. And then shunted them and embedded them into text encoder and image encoder of CLIP model respectively. The problem with this approach is that Transformer structure is huge compared to the prompt parameters. On the other hand, the initial aim of prompt tuning was to efficiently adapt pre-trained models to downstream tasks using a small number of prompt parameters.

However, a common problem with the above methods is that the design of the prompt tuning method does not adequately consider the impact on downstream tasks. Different loss guides the task differently. During training, accuracy trend for the visual language model task also differs from downstream tasks. For example, the trend of accuracy when the CLIP model is trained under contrast learning loss is not the same as that in a few-shot learning task. To solve the above problem, we propose an efficient method of prompt tuning called EPT, and we will present our work in detail in Section 3.

3 Approach

After extensive experimental and reproduction work, we propose an efficient prompt tuning method for visual language models, called Efficient Prompt Tuning (EPT). Our prompt tuning method is based on the CLIP model, so we first introduce the CLIP visual language model in section 3.1 Visual and language pre-training. We will then introduce prompt tuning method on image encoders in section 3.2 Visual prompt tuning and prompt tuning method on text encoders in section 3.3 Text prompt tuning. Finally, to solve the problem mentioned above, we will introduce loss fusion methods specific to downstream tasks in 3.4 Downstream task-related loss fusion.

3.1 Visual and language pre-training

CLIP [1] consists of an image encoder and a text encoder. The image encoder is usually built with ResNet50 [15] or ViT [16] as the backbone, while the text encoder is usually built on Transformer [17]. A pair of image-text data (image, text) is input to the image encoder and text encoder respectively to extract the corresponding features. For the encoded image features and text features, CLIP is applied to maximize the cosine similarity of matched image-text data pairs and minimize the cosine similarity of other mismatched image-text data pairs.

To construct the text description, the label of the image is introduced into the manual template "a photo of [class]" and then, the encoded features are extracted by the text encoder. For the extracted visual features and text features, the final predicted class probabilities are expressed as follows:

$$p(y=i \mid \boldsymbol{x})=\frac{exp\ (cos\ (\boldsymbol{\omega_i},\boldsymbol{z})/\tau)}{\sum_{j=1}^{N} exp\ (cos\ (\boldsymbol{\omega_j},\boldsymbol{z})/\tau)} \tag{1}$$

For a given image x and a text set y consisting of N image categories, $\boldsymbol{\omega_i}$ denote the text features extracted by the text encoder, $\boldsymbol{z}$ denote the visual features of the image extracted by the image encoder. $cos(\cdot,\cdot)$ is used to calculate the cosine similarity between the text features and the visual features. τ refers to a fixed temperature coefficient.

3.2 Visual prompt tuning

VPT [4] was the first means to introduce prompt engineering as a large-scale pre-trained model, such as ViT [16] for image processing. Simple trainable prompt parameters that were simply added were difficult to adapt to complex image information and realize the potential of pre-trained visual models. To expand the space of input prompt parameters, we apply a fully connected neural network [18] to encode high-dimensional prompt parameters, which are subsequently combined with image features as the input to the image encoder. After extensive experiments, we found that simply adding parameters may cause the model to overfit the training data, so we added the dropout layer to the fully connected neural network. The architecture is shown in Fig.2. Thus, original prompt parameter of dimension d_1 (d_1 can be a large value) is first encoded and downscaled by the fully connected neural network to output a prompt parameter of dimension d_2 (d_2 can be a value that matches the image encoder). Our approach takes ViT as the reference model. The prompt tuning method for the visual part is represented as follows, where the green color 🟩indicates the parameters that can be tuned during the training of the model. The rest of the parameters in ViT are fixed.

$$\begin{gathered}\boldsymbol{P_1}=FCN(\boldsymbol{P_0})\\ [x_1,Z_1,E_1]=L_1([x_0,\boldsymbol{P_1},E_0])\\ [x_i,Z_i,E_i]=L_i([x_{i-1},Z_{i-1},E_{i-1}])\\ y=Head(x_k)\end{gathered} \tag{2}$$

In Equation (2), $\boldsymbol{P_0}$ represents the initial soft prompt parameters, $\boldsymbol{P_1}$ denotes the soft prompt parameters encoded by the fully connected neural network (FCN). Z_i represents the feature characteristics computed by the i^{th} transformer layer. In the context of ViT , these parameters are integrated prior to the position encoding. Thus, the relative localization of x_k to the prompt is preserved.

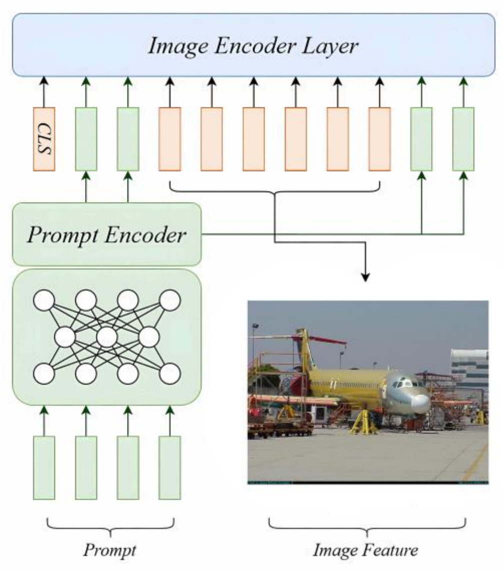

Fig. 2. The trend of accuracy during training CLIP with classification loss.

3.3 Text prompt tuning

In this section, we introduce text prompt tuning part of EPT. Since the text prompt tuning method proposed by CoOp [5] has made great progress, we still use the method in CoOp, which use trainable continuous parameters instead of discrete words as "prompts". Prompt parameters and image labels are stitched together and fed into the text encoder, so that the corresponding text is described as "[soft] [soft] [soft] [soft] [soft] [class]" . Fig.3 shows the detailed architecture of the text prompt tuning, where the soft tokens represent the optimizable prompt parameters.

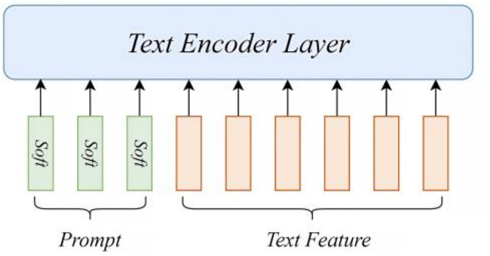

Fig. 3. The architecture of text prompt tuning method practiced in CLIP image encoder.

Thus, a given text description is fed into the text encoder to generate the probability of a visual feature falling into a category i , as shown in Equation (3). The ***[class]*** token in the prompt t_i is replaced with the corresponding category of the image i^{th} , such as "airplane" and "dog". $g(t_i)$ denotes the features extracted by the text encoder from text description consisting of optimizable prompt parameters and the label of i^{th} image.

$$p(y = i|\, \boldsymbol{x}) = \frac{exp\ (cos\ (\boldsymbol{g}(\boldsymbol{t_i}), \boldsymbol{z})/\tau}{\sum_{j=1}^{N} exp\ (cos\ (\boldsymbol{g}(\boldsymbol{t_j}), \boldsymbol{z})/\tau} \tag{3}$$

3.4 Downstream task-related loss fusion

In this section, we will detail the implementation of loss specific to downstream tasks. In CLIP model, the image and text pairs are trained with the goal of contrast learning, which is to maximize the cosine similarity of N matched image text pairs at diagonal positions and minimize the cosine similarity of $N^2 - N$ mismatched image text pairs at other positions in image text pairs of batch size N. InfoNCE loss is used in CLIP [1]. The loss for image encoder is as follows:

$$L_I = -\frac{1}{N}\sum_{i=1}^{N} log \frac{exp\ (cos\ (\boldsymbol{\omega_i}, \boldsymbol{z})/\tau)}{\sum_{j=1}^{N} exp\ (cos\ (\boldsymbol{\omega_j}, \boldsymbol{z})/\tau)} \tag{4}$$

The loss of the text encoder L_T and the loss of the image encoder L_I are symmetric[19]. Loss of CLIP model L_{CLIP} is the arithmetic average of the loss of the text encoder and the loss of the image encoder, so L_{CLIP} can be expressed as：

$$L_{CLIP} = Average(L_T + L_I) \tag{5}$$

As far as we know, the downstream task loss and the original training loss have different effects on the results when adapting the visual language to the downstream task. Therefore, in our approach, we integrate the loss of the downstream task into the training task of the visual language model. We choose the common classification task in few-shot learning as the reference downstream task. The cross entropy loss [20] of the classification task L_{down} with few shot learning is shown as follows:

$$L_{down} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} \boldsymbol{f_{i,j}} log\ (\boldsymbol{p_{i,j}}) \tag{6}$$

In an image classification problem with batch size N and number of classes M. For image i , $\boldsymbol{f_{i,j}}$ denotes the binary indicator (0 or 1) if class label j is the correct classification for image i . log denotes the natural logarithm. $\boldsymbol{p_{i,j}}$ denotes the probability that image i is predicted to be class j **.** Therefore, in order to integrate the loss of downstream task into prompt tuning of the visual language model, we define the loss function of EPT as follows. The parameter α in the formula is preset to 0.5.

$$L_{EPT} = (1 - \alpha)L_{CLIP} + \alpha L_{down} \tag{7}$$

4 Experiments and discussions

In this section, we first test few-shot learning performance of our method in Section 4.1 Few-shot learning. To verify the improvement of the model performance by the fused loss function, we test the performance of different loss functions on the model performance in Section 4.2 Performance of the model with different loss functions.

4.1 Few-shot learning

Baselines. We compare our approach with **1) Zero-shot CLIP**, which utilizes manually constructed prompts and does not use new training data. **2) The single-modal prompt tuning approach.** This approach used prompt tuning on the text or image modality of CLIP model for text and image, respectively. For visual prompt tuning, we chose VPT-deep [4] as the comparison model. For text prompt tuning, we chose CoOp [5] as the comparison model. **3) Multimodal prompt tuning approach.** This approach applied the prompt parameter on both image and text modalities of the visual language model at the same time. We choose UPT [6] as the comparison model.

Datasets. We follow Zhou et al. [5] to test the model's few-shot learning performance using 11 datasets (ImageNet [21], Caltech101 [22], OxfordPets [23], StanfordCars [24], Flowers102 [25], Food101 [26],FGVC-Aircraft [27], SUN397 [28], UCF101 [29], DTD [30], EuroSAT [31]) as our benchmarks. For image feature extraction, we used ViT-B/16 as part of visual prompt tuning. Following Zhou et al. we samely used 1/2/4/8/16 samples as training data and test data from the entire dataset as evaluation data. We recorded the average results of different random seeds as the final results. The results of all experiments are shown in Fig.4. All the details of the training follow Zhou et al.

EPT vs Single-modal prompt tuning approach. From the average results, our method beats VPT by 0.59%, 2.03%, 2.43%, and 3.82% at 2/4/8/16 training shots, respectively. Our method outperforms CoOp 1.31%, 2.53%, and 4.84% at 4/8/16 training shots, respectively. In general, our method has more obvious advantages over CoOp, VPT and other unimodal prompt tuning methods. In particular,on the datasets of Food101, FGVCAircraft, DTD, EuroSAT, and UCF101, our method has made great progress compared with the unimodal prompt tuning method. However, we observe that the performance of our method decreases compared to the previous method when the sample size is 1. This may be due to the loss of the downstream task addition that causes overfitting to some of the data. Also, on some datasets, such as OxfordPet, Flowsers102,StanfordCars, the improvement of EPT is less, which may be caused by the excessive noise of the data.

EPT vs Multimodal prompt tuning approach. From Fig.4, we observe that EPT achieves approximately the same excellent performance as UPT in most cases, such as Caltech101,OxfordPets, EuroSAT . It is worth noting that EPT outperforms UPT on a few datasets, such as Food101,FGVCAircraft,DTD,UCF101. From the average results, EPT performs essentially the same as UPT at 1/2/4/8 training samples. At 16 training shots, EPT outperformed UPT by 1.55% on average on 11 data sets. In addition, EPT only needs to adjust the image encoder and text encoder prompting parameters during prompt tuning. In contrast, UPT needs to adjust the whole Transformer parameters in addition to the image and text modalities in order to achieve consistent performance. In general, EPT performs well in the adaptation of few-shot learning due to the addition of downstream tasks to guide the visual language model.

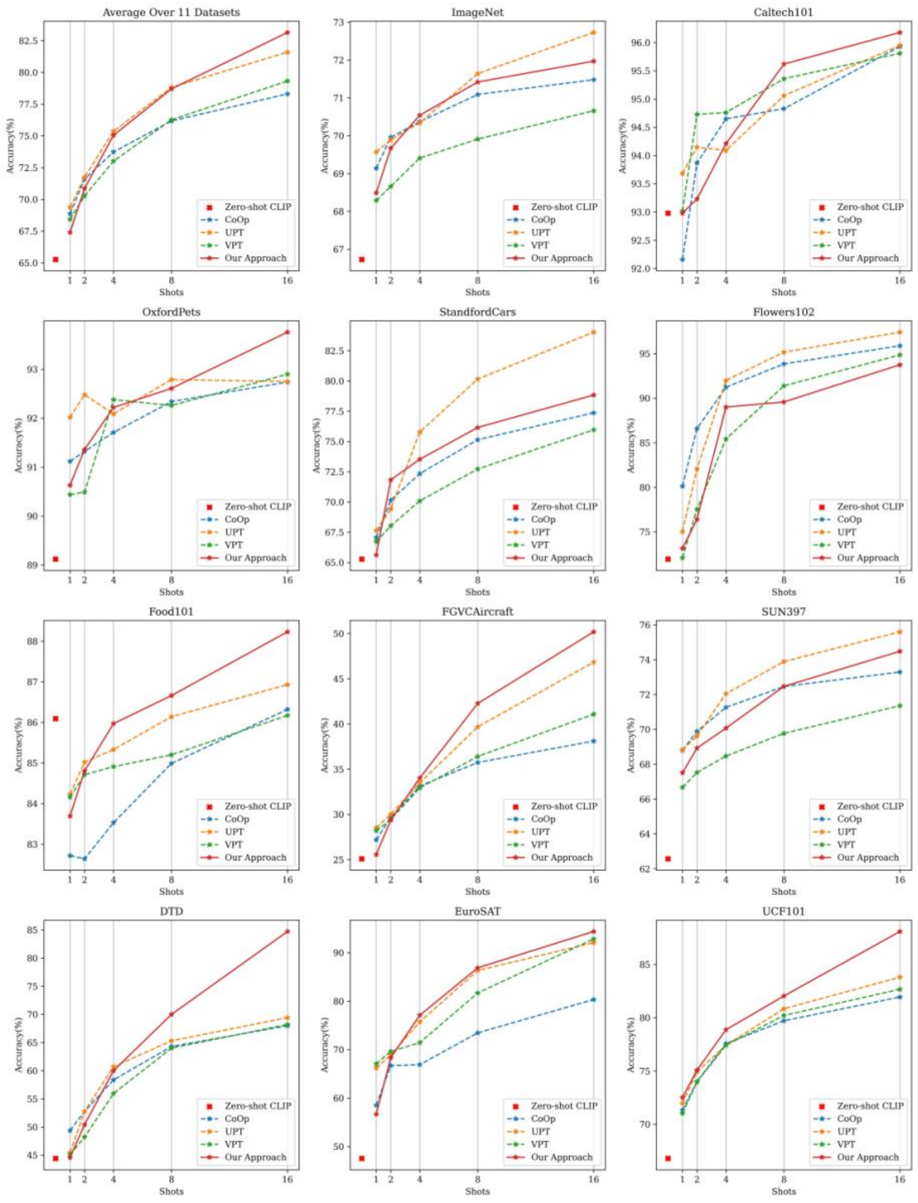

Fig. 4. Main results over 11 datasets under the few-shot learning setting.

4.2 Performance of the model with different loss functions

In this section, to test the performance of the new loss in downstream tasks and exclude the effect of the number of dataset categories on the experimental results, we set up two different scenarios (**datasets with few categories** and **datasets with large categories**) separately.

Datasets. For datasets with few categories, we choose RAF-DB dataset [32], which is a face expression recognition dataset contains 7 basic expressions (i.e., neutral, happy, sad, surprised, fearful, disgusted, and angry). The training data consisted of 12,271 images and the test data consisted of 3,068 images. For datasets with large categories， we choose the Food101 dataset. This dataset includes 101 food categories with 101,000 images. For each category, 250 manually reviewed test images are presented along with 750 training images. In the two different experimental scenarios, all training images are used as training data. All test images are used as test data.

Loss functions. For the loss function, we choose ***1)Contrast learning loss***, which is usually InfoNCE loss in the CLIP model, which is to maximize the cosine similarity of N image text pairs at diagonal positions and minimize the cosine similarity of $N^2 - N$ image text pairs at other positions. The specific implementation is shown in Equation 5. ***2) Loss of downstream task***, in CLIP adaptation to downstream task of few-shot learning, which is to calculate the cross entropy loss of predicted image labels and real image labels. The specific implementation is shown in Equation 6. ***3)Loss fusion***, which is the loss associated with the downstream task used in the EPT. The specific implementation is shown in Equation 7.

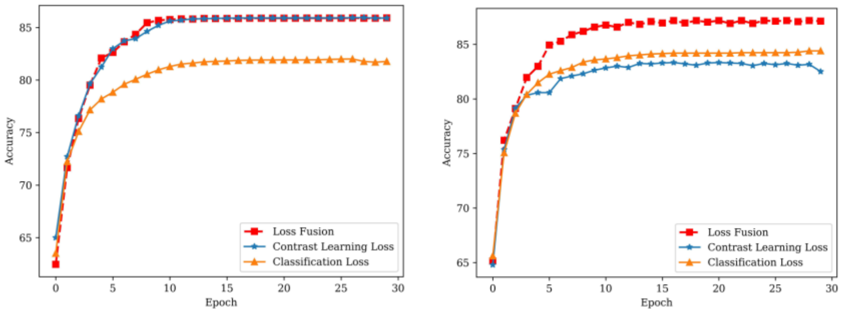

Fig.5. Classification accuracy of the model trained on the RAF-DB dataset with different loss(left). Classification accuracy of the model trained on the Food101 dataset with different loss (right).

Datasets with few categories. Fig.5(left) shows the influence of different loss in the RAF-DB dataset on the classification accuracy during training. We found that the loss of contrast learning and the loss of fusion performed similarly. The classification loss for the downstream task is slightly worse than the former. It is worth acknowledging that the fusion loss of the downstream task used in EPT outperforms. While classification loss is significantly weaker than contrast learning loss and loss fusion. This suggests that loss fusion can combine the properties of CLIP itself and improve the weakness of classification loss.

Datasets with large categories. Fig.5(right) shows the effect of different loss in the Food101 dataset on the classification accuracy during training. In the face of dataset containing 101 categories, we found that the loss after fusion outperformed the loss from comparison learning and the classification loss from downstream task. At the 5th epoch, both the contrast learning loss and the classification loss converged, however the fused loss did not converge. This may be the main reason why EPT outperformed

UPT, CoOp and VPT. Thus, the fusion loss of the downstream task utilized in EPT still performed well in the face of datasets with large number of categories.

5 Conclusion

With the rapid expansion and growth of the number of visual language model parameters, efficient and computationally efficient adaptation methods are critical for pre-trained models for downstream tasks. Our paper provides a novel solution to the problem of adapting large visual language models like CLIP from the perspective of model structure and design of loss functions. Our study sheds light on the problem of loss in downstream tasks that has been overlooked in previous studies and gives a solution called EPT. Performance comparable to the effect of previous studies can be achieved in EPT by simply adjusting the loss function and adding simple prompt parameters. The results show that the fused loss achieve excellent performance in both the CLIP model itself and in downstream tasks. Overall, we believe that multimodal prompt learning is a promising area of research. We hope that our study will stimulate more lively discussions and deeper research.

References

1. Radford, Alec, et al. "Learning transferable visual models from natural language supervision." International conference on machine learning. PMLR, 2021.
2. Jia, Chao, et al. "Scaling up visual and vision-language representation learning with noisy text supervision." International conference on machine learning. PMLR, 2021.
3. Liu, P. et al., 2023. Pre-train, prompt, and predict: A survey of prompting methods in NLP. ACM Computing Surveys, 55(9), pp. 1-35.
4. Jia, M. et al., 2022. Visual prompt tuning. In Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIII (pp. 709-727). Cham: Springer Nature Switzerland.
5. Zhou, K. et al., 2022. Learning to prompt for vision-language models. International Journal of Computer Vision, 130(9), pp. 2337-2348.
6. Zang, Y. et al., 2022. Unified vision and language prompt learning. arXiv preprint arXiv:2210.07225.
7. Shin, T. et al., 2020. Autoprompt: Eliciting knowledge from language models. arXiv preprint arXiv:2010.15980.
8. Qin, G. and Eisner, J., 2021. Learning how to ask: Querying LMs with soft prompts. arXiv preprint arXiv:2104.06599.
9. Tan, H. and Bansal, M., 2019. Lxmert: Learning cross-modality encoder representations from transformers. arXiv preprint arXiv:1908.07490.
10. Li, X. et al., 2020. Oscar: Object-semantics aligned pre-training for vision-language tasks. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX 16 (pp. 121-137). Springer International Publishing.
11. 11. Lu, J. et al., 2019. Vilbert: Pretraining visiolinguistic representations for vision-language tasks. Advances in neural information processing systems, 32.
12. 12. Zhou, K. et al., 2022. Conditional prompt learning for vision-language models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 16816-16825).
13. 13. Sun, X. et al., 2022. Dualcoop: Fast adaptation to multi-label recognition with limited annotations. arXiv preprint arXiv:2206.09541.

14. Xing, Y. et al., 2022. Class-aware visual prompt tuning for vision-language pre-trained model. arXiv preprint arXiv:2208.08340.
15. He, K. et al., 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
16. Dosovitskiy, A. et al., 2020. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.
17. Vaswani, A. et al., 2017. Attention is all you need. Advances in neural information processing systems, 30.
18. Farhat, N., 1989. Optoelectronic neural networks and learning machines. IEEE Circuits and Devices Magazine, 5(5), pp. 32-41.
19. Li, Y. et al., 2021. Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm. arXiv preprint arXiv:2110.05208.
20. Shannon, C.E., 2001. A mathematical theory of communication. ACM SIGMOBILE mobile computing and communications review, 5(1), pp. 3-55.
21. Deng, J. et al., 2009. Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition (pp. 248-255). IEEE.
22. Fei-Fei, L. et al., 2004. Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories. In 2004 conference on computer vision and pattern recognition workshop (pp. 178-178). IEEE.
23. Parkhi, O.M. et al., 2012. Cats and dogs. In 2012 IEEE conference on computer vision and pattern recognition (pp. 3498-3505). IEEE.
24. Krause, J. et al., 2013. 3D object representations for fine-grained categorization. In Proceedings of the IEEE international conference on computer vision workshops (pp. 554-561).
25. Nilsback, M.E. and Zisserman, A., 2008. Automated flower classification over a large number of classes. In 2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing (pp. 722-729). IEEE.
26. Bossard, L. et al., 2014. Food-101–mining discriminative components with random forests. In Proc. Computer Vision–ECCV 2014, Zurich, Switzerland, Sept. 6-12, 2014, Part VI (pp. 446-461). Springer Int. Pub.
27. Maji, S. et al., 2013. Fine-grained visual classification of aircraft. arXiv preprint arXiv:1306.5151.
28. Xiao, J. et al., 2010. Sun database: Large-scale scene recognition from abbey to zoo. In Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2010 (pp. 3485-3492). IEEE.
29. Soomro, K. et al., 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402.
30. Cimpoi, M. et al., 2014. Describing textures in the wild. In Proc. IEEE Conf. Computer Vision and Pattern Recognition (pp. 3606-3613).
31. Helber, P. et al., 2019. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. IEEE J. Sel. Topics in Appl. Earth Observ. and Remote Sensing, 12(7), pp. 2217-2226.
32. Li, S. and Deng, W., 2019. Blended emotion in-the-wild: Multi-label facial expression recognition using crowdsourced annotations and deep locality feature learning. Int. J. Computer Vision, 127(6-7), pp. 884-906.