XtremeCLIP: Extremely Parameter-efficient Tuning for Low-resource Vision Language Understanding

Moming Tang[1], **Chengyu Wang**[2], **Jianing Wang**[1], **Chuanqi Tan**[2], **Songfang Huang**[2], **Cen Chen**[1]*, **Weining Qian**[1]
[1]East China Normal University, [2]Alibaba Group
mmtang@stu.ecnu.edu.cn,chengyu.wcy@alibaba-inc.com,
lygwjn@gmail.com, {chuanqi.tcq,songfang.hsf}@alibaba-inc.com,
{cenchen,wnqian}@dase.ecnu.edu.cn

Abstract

Recently, Contrastive Visual-Language Pre-training (CLIP) has demonstrated remarkable capability in various Visual Language Understanding (VLU) tasks. Yet, most CLIP-based methods require tasks-specific designs and sufficient training data. In this paper, we introduce a simple yet efficient paradigm for low-resource VLU named XtremeCLIP, which involves very few trainable parameters to improve the generalization ability of the trained models. In our XtremeCLIP framework, we reformulate a series of VLU tasks as a unified open-book affinity-matching problem. Furthermore, to handle the insufficient supervised signals in small datasets, we adopt contrastive learning to utilize the implicit sorting information of ground-truth labels to provide more supervised cues. Extensive experiments over multiple datasets on visual entailment, visual question answering, and image classification show that XtremeCLIP consistently outperforms existing baselines in low-resource settings. [1]

1 Introduction

Pre-trained Visual-Language models such as X-VLM (Zeng et al., 2021) and CLIP (Radford et al., 2021) have been proposed to unify visual and textual representations in the same embedding space and shown great potential for Visual Language Understanding (VLU). Conventional fine-tuning approaches (Clark et al., 2020; Lee et al., 2020; Wang et al., 2023) heavily depend on the time-consuming and labor-intensive process of data annotation, which are bothersome in low-resource scenarios. In the literature, Ben Zaken et al. (2022); Song et al. (2022) propose partial-parameter fine-tuning to preserve the pre-trained knowledge of these models. Yao et al. (2021); Song et al. (2022); Tsimpoukelli et al. (2021) reformulate visual grounding and visual question answering as a "fill-in-blank" problem by hand-crafted prompts. Gao et al. (2021); Zhang et al. (2022) utilize lightweight adapters (Houlsby et al., 2019) to retain the knowledge of CLIP. Besides, Zhou et al. (2022b,a); Zhu et al. (2022) address image classification tasks by utilizing textual representations describing image categories.

Despite the success, we suggest there are still some drawbacks in existing works. i) The discrete prompt paradigm requires labor-intensive prompt-engineering, while the soft template paradigm results in an unstable training process. ii) Adapters or partial-parameter fine-tuning methods may underperform due to their relatively large number of tunable parameters, requiring additional training data to achieve satisfactory results. iii) The aforementioned methods are task-specific in design, implying that their effectiveness may be derived from task-specific architectures. Hence, it is vital for us to design a more unified parameter-efficient tuning approach in order to solve various VLU tasks.

In this paper, we present XtremeCLIP, an extremely parameter-efficient tuning method for solving various VLU tasks based on CLIP (Radford et al., 2021). XtremeCLIP reformulates a series of VLU tasks uniformly into an open-book affinity-matching problem. Here, we adopt a knowledge-base prototype matrix to record the salient characteristics for each class by visual-textual fusion features, then perform affinity matching between image-text pairs and prototypes of each class. We further utilize the implicit sorting information of ground-truth labels by contrastive learning to provide more supervised cues from low-resource training sets. During model training, all parameters of textual and visual encoders in CLIP are fixed. Hence, XtremeCLIP is extremely parameter-efficient. We conduct extensive experiments on a visual entailment (VE) benchmark (i.e., SNLI-VE), a

*Corresponding author
[1]The source code is publicly available in the EasyNLP framework (Wang et al., 2022). URL: https://github.com/alibaba/EasyNLP.

Findings of the Association for Computational Linguistics: ACL 2023, pages 6368–6376

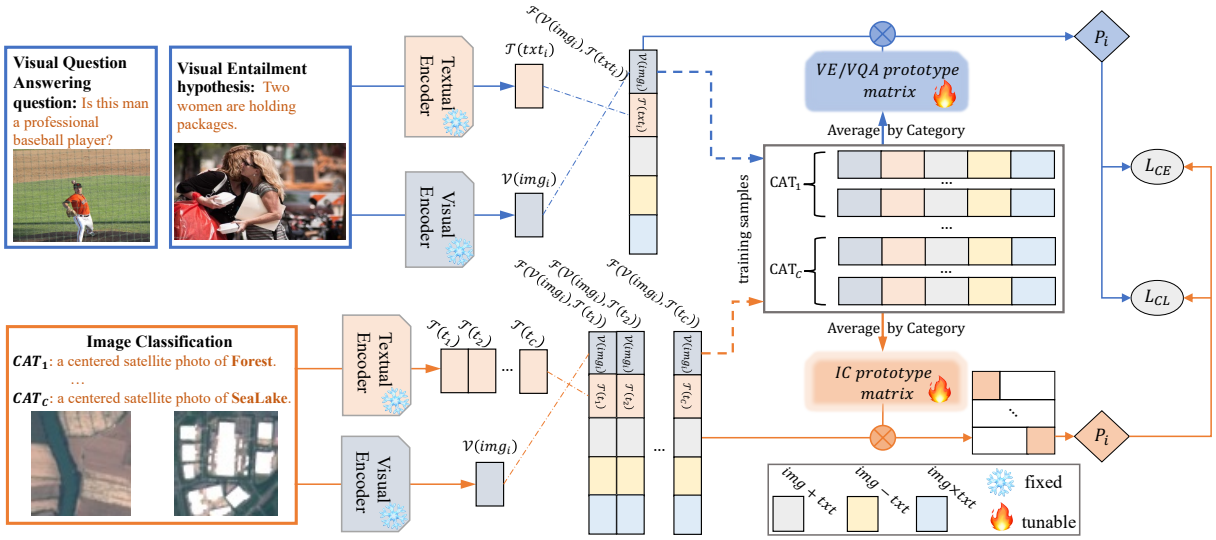

Figure 1: Model architecture and training procedure of XtremeCLIP.

visual question answering (VQA) benchmark (i.e., VQA v2), and three widely used image classification (IC) benchmarks (i.e., EuroSAT, DTD, and FGVC). Results show XtremeCLIP consistently outperforms baselines in low-resource scenarios.

2 XtremeCLIP: The Proposed Method

The model architecture and training procedure of XtremeCLIP are in Figure 1. First, a knowledge-base prototype matrix is constructed (Snell et al., 2017) by combining visual and textual features, designed to serve as a repository of the key characteristics for each class. Then, open-book affinity matching is performed between the image-text instance and the prototypes for each class.

2.1 Prototype Matrix Construction

Given a set of N image-text training instances: $D = \{(\text{img}_\text{i}, \text{txt}_\text{i}), l_i\}_{i=1}^{N}$, where l_i denotes the ground-truth label, txt_i denotes the corresponding textual description of the image img_i. Image-text pairs are encoded using visual $\mathcal{V}$ and textual $\mathcal{T}$ encoders of CLIP:

$$v_1 = \mathcal{V}(\text{img}_\text{i}), v_2 = \mathcal{T}(\text{txt}_\text{i})), v_1, v_2 \in R^d \quad (1)$$

A fusion function $\mathcal{F}$ is employed to obtain uniform image-text representations that capture the interactions between visual and textual information:

$$\mathcal{F}(v_1, v_2) = [v_1, v_2, v_1 + v_2, v_1 - v_2, v_1 \times v_2] \quad (2)$$

where $\mathcal{F}(v_1, v_2) \in R^{5d}$. These fusion features are used to construct the knowledge-base prototype matrix denoted as W_P by averaging them per their ground-truth labels:

$$M_c = \frac{\sum_{i=1}^{N} I(l_i = c) \cdot \mathcal{F}(\mathcal{V}(\text{img}_\text{i}), \mathcal{T}(\text{txt}_\text{i}))}{\sum_{i=1}^{N} I(l_i = c)} \quad (3)$$

$$W_P = [M_1, \cdots, M_C], (W_P \in R^{C \times 5d}) \quad (4)$$

where C denotes the number of classes, M_c denotes the prototype of the c-th class and $c \in 1 \cdots C$, $I(\cdot)$ denotes the indicator function, and $[\cdot]$ denotes the concatenation operator.

2.2 Open-book Matching

Prototype Matching for VE and VQA. In VE or VQA, affinity matching is performed between the fusion feature of a given image-text pair and the prototypes for each class: $P_i = \mathcal{F}(\mathcal{V}(\text{img}_\text{i}), \mathcal{T}(\text{txt}_\text{i})) \cdot W_P^\top$.

Prototype Matching for IC. In traditional IC tasks, only images are provided without corresponding textual descriptions. We obtain textual descriptions (prompts) for all classes, followin Radford et al. (2021). Given an image img_i and the textual descriptions of all image categories $\{t_c | c = 1 \cdots C\}$, the predicted probability (denoted as $P_{i,c}$) of img_i w.r.t. the c-th image category is as follows: $P_{i,c} = \mathcal{F}(\mathcal{V}(\text{img}_\text{i}), \mathcal{T}(t_c)) \cdot M_c^\top$. Thus, the entire probabilistic distribution P_i is: $P_i = [P_{i,c} | c = 1 \cdots C]$.

2.3 Training Paradigm

XtremeCLIP has only one set of tunable parameters, namely the Prototype Matrix denoted by W_p. Its fusion function, visual, and textual encoders are solely utilized for constructing the prototype

matrix, with all parameters frozen during the training phase. In XtremeCLIP, the model is trained using the Cross-Entropy (CE) loss given P_i. The sample-wise CE loss is defined as follows:

$$L_{CE} = -\sum_{c\in\{1\cdots C\}} l_{i,c} \cdot \log P_{i,c} \quad (5)$$

where $l_{i,c}$ denotes the ground-truth label w.r.t. the c-th class. However, the model can hardly achieve satisfactory performance with only supervised signals from CE in low-resource tasks. Given that instances' affinity with ground-truth classes should be ranked higher than other classes, this implicit sorting information can be utilized to guide the model to recognize instances' ground-truth classes via contrastive learning (Zhong et al., 2020). We define the affinity of the ground-truth category (i.e., the prototype matching probability, denoted as $P_{i,l}$) as positive samples and other affinities in P_i as negative samples. Following Liu et al. (2022); Liu and Liu (2021), the sample-wise Contrastive Learning (CL) loss is computed as:

$$L_{CL} = \sum_{c=1}^{C} \max(0, P_{i,l} - P_{i,c}). \quad (6)$$

The total loss function for XtremeCLIP, namely L, is defined as: $L = L_{CE} + L_{CL}$.

3 Experiments

3.1 Experimental Settings

We briefly describe the experimental settings and leave more details in Appendix.

Datasets. SNLI-VE (Xie et al., 2018) is utilized for visual entailment, consisting of image-text pairs whereby a premise is defined by an image. **VQA v2** (Goyal et al., 2017) is utilized for visual question answering, containing questions about images. Here, we only consider the yes/no samples. Questions with open answers require decoder models and are not the focus of this paper For IC, **EuroSAT** (Helber et al., 2019) contains satellite images consisting out of 10 categories. **DTD** (Cimpoi et al., 2014) contains describable textures images with 47 classes. **FGVC** (Maji et al., 2013) contains images of 102 aircraft model variants.

Baselines. In our work, we compare XtremeCLIP with zero-shot CLIP (Radford et al., 2021); fine-tuning paradigms including standard fine-tuning, mixout (Lee et al., 2020), pre-trained weight decay (weight decay) (Lee et al., 2020) and Layer-wise Learning Rate Decay (LLRD) (Clark et al., 2020); partial-parameter fine-tuning paradigms including BitFit (Ben Zaken et al., 2022) and BiNor (Song et al., 2022); and adapter-based methods including CLIP-Adapter (Gao et al., 2021) and Tip-Adapter (Zhang et al., 2022).

Backbone. For fair comparison, all baselines and our approach adopt the ViT-B/16 (ViT-Base with the patch size 16 × 16) version of CLIP. Other versions of CLIP are also experimented with.

3.2 Experimental Results

VE&VQA results in low-resource settings. Table 1 presents the results of XtremeCLIP and baselines, in low-resource VE and VQA. The fine-tuning paradigms perform worse than partial fine-tuning paradigms in all settings, which demonstrates conventional fine-tuning paradigms are data-hungry and not suitable for low-resource VLU tasks. XtremeCLIP consistently outperforms partial fine-tuning and adapter-based methods, showing that reformulating VLU tasks as prototype affinity matching can efficiently utilize visual-textual information with much fewer trainable parameters.

Few-shot IC. Table 2 presents the performance of XtremeCLIP and baselines, in few-shot IC. Fine-tuning paradigms are still not suitable for few-shot image classification. Unlike BiNor and BitFit, CLIP-Adapter and Tip-Adapter specifically utilize adapters to learn from low-resource datasets meanwhile preserving the knowledge of CLIP, thus performing the best among baselines. Although XtremeCLIP has fewer trainable parameters than baselines, it still performs the best thanks to the supervised cues provided by contrastive learning and our task modeling approach.

Ablation study. We replace the prototype matrix of XtremeCLIP with a randomly initialized matrix i.e., XtremeCLIP w/o. proto). We also detach the contrastive loss from XtremeCLIP (i.e., XtremeCLIP w/o. cl), or replace the fusion feature with the concatenation of visual and textual features (i.e., XtremeCLIP w/o. fusion). Table 3 presents the results of XtremeCLIP and its ablations. Detaching contrastive loss drops the performance, as contrastive learning provides more supervised cues. Reformulating VLU tasks as prototype affinity matching is somehow an open-book retrieval problem, which can augment model performance (Chen et al., 2022). Replacing the fusion feature drastically drops performance for VLU, which demonstrates the importance of interaction

Method	# Params.	SNLI-VE			VQA v2			Avg
		2k	5k	10k	2k	5k	10k	
Zero-shot learning	0		33.74			52.03		42.89
Full fine-tuning	149M	47.31	48.12	51.10	52.79	53.29	54.10	51.12
LLRD (Clark et al., 2020)	149M	50.18	55.35	57.23	52.06	52.90	53.88	53.60
mixout (Lee et al., 2020)	149M	50.19	53.97	55.16	53.17	53.86	53.83	53.36
weight decay (Lee et al., 2020)	149M	50.68	54.07	55.09	53.18	53.92	53.81	53.46
BitFit (Ben Zaken et al., 2022)	176-178K	54.88	58.02	59.56	52.96	53.84	54.72	55.66
BiNor (Song et al., 2022)	208-210K	54.91	58.03	59.54	52.93	53.83	54.75	55.67
CLIP-Adapter (Gao et al., 2021)	131K-262K	54.77	57.83	59.21	53.21	53.45	54.21	55.45
Tip-Adapter (Zhang et al., 2022)	5-10M	54.65	58.11	59.67	52.94	53.63	54.70	55.62
XtremeCLIP	5-7K	**55.61**	**59.53**	**62.06**	**53.51**	**56.44**	**59.21**	**57.73**

Table 1: Accuracy (%) on **V**isual **E**ntailment and **V**isual **Q**uestion **A**nswering tasks with 2000, 5000, 10000 training samples. Here, #Params. denotes the number of tunable parameters. Best results are in bold.

Method	# Params.	EuroSat (10)		DTD (47)		FGVC (102)		Avg
		8 shot	16 shot	8 shot	16 shot	8 shot	16 shot	
Zero-shot learning	0	48.43		44.27		24.8		39.17
Full fine-tuning	149M	62.99	67.75	62.06	64.78	27.72	28.14	52.24
LLRD (Clark et al., 2020)	149M	70.91	75.58	64.30	69.39	30.18	31.36	56.95
mixout (Lee et al., 2020)	149M	70.85	72.23	64.07	68.97	28.98	30.24	55.89
weight decay (Lee et al., 2020)	149M	70.93	72.17	64.01	69.09	29.04	30.03	55.88
BitFit (Ben Zaken et al., 2022)	196∼427K	74.15	83.59	64.36	66.43	38.52	41.61	61.44
BiNor (Song et al., 2022)	228∼459K	78.63	86.59	65.07	70.04	38.43	41.73	63.42
CLIP-Adapter (Gao et al., 2021)	131∼262K	81.85	88.37	65.07	71.10	40.17	44.88	65.24
Tip-Adapter (Zhang et al., 2022)	84∼979K	82.02	87.49	67.32	71.81	39.51	45.12	65.55
XtremeCLIP	25∼256K	**82.57**	**89.19**	**67.61**	**72.81**	**42.66**	**48.30**	**67.19**

Table 2: Accuracy (%) on **I**mage **C**lassification tasks with 8-shot and 16-shot images of EuroSat, DTD and FGVC. Here, (·) stands for the number of image categories. Best results are in bold.

Method	VE (10k)	VQA (10k)	FGVC (16)
Full	62.06	59.21	48.30
w/o. cl	60.94 (-1.12)	54.90 (-4.31)	48.21 (-0.09)
w/o. proto	61.98 (-0.08)	55.45 (-3.76)	48.15 (-0.15)
w/o. fusion	58.25 (-3.81)	54.62 (-4.59)	48.09 (-0.21)

Table 3: Accuracy (%) of XtremeCLIP and its ablations.

Backbone	Method	VE	VQA	FGVC
ViT-B/16	Full FT	51.10	54.12	28.14
	XtremeCLIP	62.06	59.21	48.30
ViT-B/32	Full FT	52.88	57.13	21.54
	XtremeCLIP	61.09	58.71	40.29
ViT-L/14	Full FT	54.59	56.05	28.86
	XtremeCLIP	61.79	59.12	58.93

Table 4: Accuracy (%) of XtremeCLIP and full fine-tuning (FT) utilizing various CLIP backbones.

between visual and textual information.

Model-scale study. We test various CLIP versions with results in Table 4. The settings are the same as in ablation study. It shows that XtremeCLIP can be effectively adapted to different CLIPs and consistently has good performance.

Data-scale study. Figure 2 presents the influence of the number of training instances on XtremeCLIP. As the number of training data increases, the accuracy of XtremeCLIP on VE and FGVC significantly increases while full fine-tuning only slightly increases, which demonstrates XtremeCLIP has higher data-using efficiency.

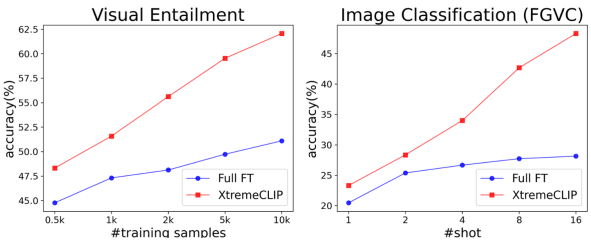

Figure 2: Accuracy (%) of XtremeCLIP and full fine-tuning with various training samples on VE and FGVC.

4 Conclusion

We propose XtremeCLIP, a simple and efficient paradigm that reformulates VLU tasks as a prototype affinity matching problem. We adopt contrastive learning to leverage implicit sorting information from ground-truth labels, providing more supervised cues to handle insufficient supervised signals in small datasets. Experimental results demonstrate that XtremeCLIP consistently outperforms all baselines in low-resource scenarios.

Limitations

In this paper, the proposed XtremeCLIP framework is mainly focused on CLIP-based deterministic VLU tasks. In future work, we will extend XtremeCLIP to other Pre-trained Vision-Language models and apply XtremeCLIP to generative tasks such as image captioning, visual grounding or visual relation extraction.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant No. 62202170 and Alibaba Group through the Alibaba Innovation Research Program.

References

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. BitFit: Simple Parameter-efficient Fine-tuning for Transformer-based Masked Language-models. In *ACL*, pages 1–9.

Xiang Chen, Lei Li, Ningyu Zhang, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2022. Relation Extraction as Open-Book Examination: Retrieval-enhanced prompt tuning. In *SIGIR*, pages 2443–2448.

M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. 2014. Describing Textures in the Wild. In *CVPR*, pages 3606–3613.

Kevin Clark, MinhThang Luong, Quoc Le, and Christopher D. Manning. 2020. Pre-Training Transformers as Energy-Based Cloze Models. In *EMNLP*, pages 285–294.

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. *ArXiv*.

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*, pages 398–414.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. 2019. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, pages 2217–2226.

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*, page 2790–2799.

Cheolhyoung Lee, Kyunghyun Cho, and Wanmo Kang. 2020. Mixout: Effective Regularization to Finetune Large-scale Pretrained Language Models. In *ICLR*.

Yixin Liu and Pengfei Liu. 2021. SimCLS: A Simple Framework for Contrastive Learning of Abstractive Summarization. In *ACL*, pages 1065–1072.

Yixin Liu, Pengfei Liu, Dragomir Radev, and Graham Neubig. 2022. BRIO: Bringing Order to Abstractive Summarization. In *ACL*, pages 2890–2903.

S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi. 2013. Fine-Grained Visual Classification of Aircraft. Technical report.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language supervision. In *ICML*, pages 8748–8763.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical Networks for Few-shot Learning. In *NeurIPS*, pages 4080–4090.

Haoyu Song, Li Dong, Weinan Zhang, Ting Liu, and Furu Wei. 2022. CLIP Models are Few-Shot Learners: Empirical Studies on VQA and Visual Entailment. In *ACL*, pages 6088–6100.

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*, pages 200–212.

Chengyu Wang, Minghui Qiu, Taolin Zhang, Tingting Liu, Lei Li, Jianing Wang, Ming Wang, Jun Huang, and Wei Lin. 2022. Easynlp: A comprehensive and easy-to-use toolkit for natural language processing. In *EMNLP (System Demonstrations)*, pages 22–29.

Xiaodan Wang, Lei Li, Zhixu Li, Xuwu Wang, Xiangru Zhu, Chengyu Wang, Jun Huang, and Yanghua Xiao. 2023. AGREE: aligning cross-modal entities for image-text retrieval upon vision-language pre-trained models. In *WSDM*, pages 456–464.

Ning Xie, Farley Lai, Derek Doran, and Asim Kadav. 2018. Visual Entailment Task for Visually-Grounded Language Learning. In *NeurIPS*.

Yuan Yao, Ao Zhang, Zhengyan Zhang, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. 2021. CPT: Colorful Prompt Tuning for Pre-trained Vision-Language Models. *ArXiv*.

Yan Zeng, Xinsong Zhang, and Hang Li. 2021. Multi-Grained Vision Language Pre-Training: Aligning Texts with Visual Concepts. *ArXiv*.

Renrui Zhang, Wei Zhang, Rongyao Fang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. 2022. Tip-Adapter: Training-Free Adaption of CLIP for Few-Shot Classification. In *ECCV*, page 493–510.

Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. Extractive Summarization as Text Matching. In *ACL*, pages 6197–6208.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. 2022a. Conditional prompt learning for vision-language models. In *CVPR*, pages 16816–16825.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. 2022b. Learning to Prompt for Vision-Language Models. *IJCV*, pages 2337–2348.

Beier Zhu, Yulei Niu, Yucheng Han, Yuehua Wu, and Hanwang Zhang. 2022. Prompt-aligned Gradient for Prompt Tuning. *ArXiv*.

A Case Study

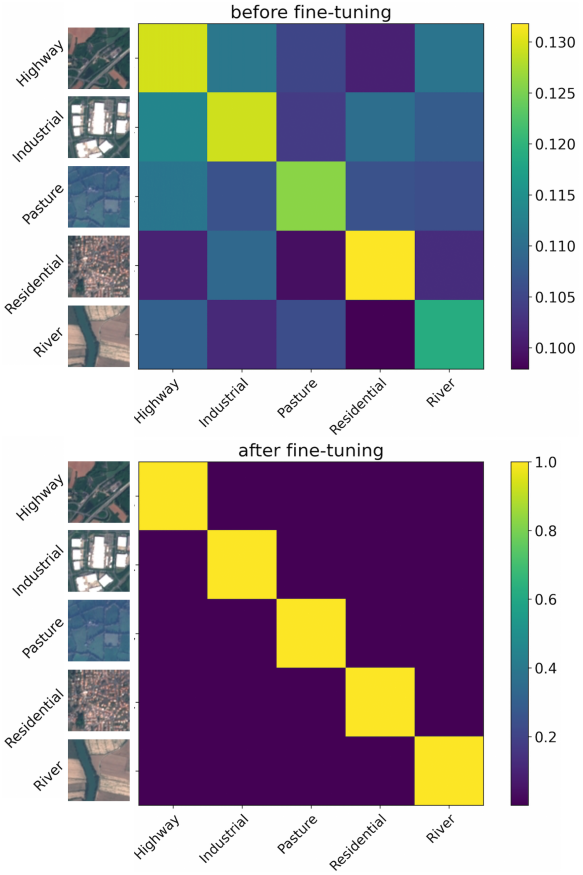

Figure 3: Probability distributions before and after fine-tuning for the few-shot IC task.

Figure 3 presents the probability distributions of several images before and after fine-tuning of our approach. The constructed knowledge-base prototype matrix indeed captures the salient characteristics of categories. Based on the knowledge, images can be correctly classified even in zero-shot learning. After fine-tuning, the performance of XtremeCLIP is further boosted.

Dataset	Prompt template
EuroSAT	a centered satellite photo of {}.
DTD	{} texture.
FGVC	a photo of a {}, a type of aircraft.

Table 5: The hard prompt templates for image classification datasets. {} denotes the position of the category names to be filled in.

Task	Dataset	# Class	# Test
IC	EuroSAT (Helber et al., 2019)	10	8100
	DTD (Cimpoi et al., 2014)	47	1692
	FGVC (Maji et al., 2013)	102	3333
VE	DNLI-VE (Xie et al., 2018)	3	17901
VQA	VQA V2 (Goyal et al., 2017)	2	80541

Table 6: Statistics of experimental datasets. #Class: the number of task categories. #Test: the number of test instances.

B Experimental Details

B.1 Training Corpora

We collect the pre-processed IC training corpora (i.e. FGVC (Maji et al., 2013), EuroSAT (Helber et al., 2019) and DTD (Cimpoi et al., 2014)) from the open-sourced project of (Zhang et al., 2022) on Github [2]. The hand-crafted prompt templates that describe the category names for EuroSAT, DTD, and FGVC are listed in Table 5. During model training, we randomly select 8 and 16 images of each category for few-shot IC.

For visual entailment and visual question-answering tasks, we download the pre-processed SNLI-VE (Xie et al., 2018) and VQA v2 (Goyal et al., 2017) from the open-sourced project X-VLM (Zeng et al., 2021) on Github [3] and randomly select 2000, 5000, and 10000 samples from each dataset for low-resource VLU tasks.

The statistics are listed in Table 6.

B.2 Experimental Details of Our Approach

We employ ViT-B/16 from OpenAI CLIP [4] as the default underlying model. We train XtremeCLIP by AdamW algorithm with $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 1e-4$. The training is processed on an NVIDIA Tesla A100 GPU. We run XtremeCLIP 20 epochs for VE and VQA with a batch size of 16 and it takes around 20 minutes; and 100 epochs for IC with a batch size of 16 and it takes around 60 minutes.

[2] https://github.com/gaopengcuhk/Tip-Adapter/DATASET.md

[3] https://github.com/zengyan-97/x-vlm

[4] https://github.com/openai/CLIP

Fusion Function	VE (10K)	VQA (10K)	FGVC (16)
XtremeCLIP	**62.06**	**59.21**	**48.30**
Quadratic	58.98	53.64	45.27
Exponential	57.45	52.41	42.52

Table 7: Accuracy (%) of XtremeCLIP utilizing various fusion fuctions. Quadratic for quadratic combination, Exponential for elementwise exponential operation.

Model	EuroSAT	DTD	FGVC
XtremeCLIP	**89.19**	**72.81**	**48.21**
CoOp (Zhou et al., 2022b)	84.87	62.57	37.48
Linear Probe (Gao et al., 2021)	82.76	63.97	36.39

Table 8: Accuracy (%) of XtremeCLIP,CoOp and Linear Probe on image classification tasks with 16-shot images of EuroSat, DTD and FGVC.

B.3 Experimental Details of Baselines

For full fine-tuning paradigms (i.e. Mixout (Lee et al., 2020), pre-trained weight decay (weight decay) (Lee et al., 2020), layerwise Learning rate decay (LLRD) (Clark et al., 2020)) and partial parameter fine-tuning paradigms (i.e. BiNor (Song et al., 2022), BitFit (Ben Zaken et al., 2022), Linear Probe (Radford et al., 2021)), we set the learning rate for CLIP parameters as $5e-7$ and the learning rate for classification head as $2e-3$ after the grid search. We train full fine-tuning baselines and partial fine-tuning baselines by AdamW algorithm with $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 1e-4$. We run the aforementioned baselines 20 epochs for VE and VQA with a batch size of 16, and 100 epochs for IC with a batch size of 16.

For CLIP-Adapter (Gao et al., 2021) [5] and Tip-Adapter (Zhang et al., 2022) [6], we directly take their open-sourced codes on GitHub. Though Tip-Adapter is proposed for few-shot IC only, by replacing the image features with the visual-textual fusion features of the input image-text pairs when constructing the instance retrieval matrix, it can be directly utilized for other VLU tasks as well.

To adapt CLIP-Adapter to VE and VQA, we respectively apply visual and textual adapter to the visual $\mathcal{V}$ and textual encoder $\mathcal{T}$ of CLIP to learn adaptive visual and textual features, then weight sum the adaptive visual and textual features with the original visual and textual feature from CLIP, following the original paper. Thereafter, we get the visual-textual fusion representations by the fusion function $\mathcal{F}$. Finally, we perform image-text pair classification with the classification head as in Gao et al. (2021).

B.4 Fusion Function Ablation

Table 7 shows the results of XtremeCLIP with various fusion functions, including traditional higher order and element-wise exponential operations. The results indicate that the selected fusion function, namely $\mathcal{F}$ in Eq. 2, is both simple and highly effective, outperforming the others.

B.5 Additional Comparasion

Table 8 presents the image classification results of XtremeCLIP and the baseline methods, namely CoOp (Zhou et al., 2022b), and Linear Probe (Gao et al., 2021), which are solely utilized for image classification. The results demonstrate that reformulating image classification as an open-book matching paradigm indeed helps XtremeCLIP consistently outperform CoOp and Linear Probe.

[5] https://github.com/gaopengcuhk/CLIP-Adapter
[6] https://github.com/gaopengcuhk/Tip-Adapter

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
The fifth section "Limitation"

☑ A2. Did you discuss any potential risks of your work?
The fifth section "Limitation"

☑ A3. Do the abstract and introduction summarize the paper's main claims?
The first section "Introduction"

☒ 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?

The third section "Experiments"

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
The third section "Experiments" and the Appendix B2

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?
the Appendix B2

☑ 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?
The third section "Experiments"

☑ 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.)?
The third section "Experiments"

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.