GLCM-Adapter: Global-Local Content Matching for Few-shot CLIP Adaptation

Shuo Wang
shuowang.edu@gmail.com
Enlong Xie
xel@mail.ustc.edu.cn
Jinda Lu*
lujd@mail.ustc.edu.cn
Jinghan Li
lijh111@mail.ustc.edu.cn
Yanbin Hao
haoyanbin@hotmail.com

University of Science and Technology of China
Hefei, China

Abstract

Recent adaptations aim to boost the few-shot capability of Contrastive Vision Language Pre-training (CLIP) by transferring textual knowledge into an image recognition procedure. However, these adaptation methods are usually operated on the global view of an input image, and thus biased recognition of partial details of the image. To solve this issue, we propose a **G**lobal-**L**ocal **C**ontent **M**atching (GLCM) strategy, which focuses on both global and local views of the image. Specifically, we first extract global and local features from the input image using the CLIP visual encoder. Meanwhile, we embed the corresponding text knowledge into features by the CLIP textual encoder. Then, we construct local representation with the textual features by selectively combining discriminative local content. The local representation contains sufficient local details, and it can help the classifier to focus on the details of the image. Finally, we match the global and local content to construct a robust classifier, namely GLCM-Adapter. Our GLCM-Adapter pays attention to information from different views, and thus achieves robust recognition. We evaluate our method on the popular few-shot classification task with 11 benchmark datasets and achieve a significant improvement over state-of-the-art methods. For example, our method achieves more than 1% average gains over the Tip-Adapter-F, and obtains more than 76.5% average accuracy for the 16-shot setting.

1 Introduction

Recently, with the developments of the Vision-Language Models (VLMs), such as Contrastive Vision-Language Pre-training (CLIP) [28], several methods [39, 43] aim to adapt CLIP for few-shot tasks and have achieved significant improvements, where the few-shot

* Jinda Lu is the corresponding author.

task is designed to solve the classification task with limited training samples and categories. However, these methods only use global feature to fit the content of the image, thus ignoring the expression of the local content of the image, and then the recognition performance of such models is impaired when the training samples are insufficient.

To relieve these issues, recent works either extract the different parts of an image [20, 32] (*e.g.* foreground or background) or design global-local interaction architectures [41, 44] to help the model capture more content from the image. However, these methods introduce huge training parameters and thus require large-scale computational and storage costs. In this paper, we present a lightweight **G**lobal-**L**ocal **C**ontent **M**atching (GLCM) strategy to achieve a similar effect as in previous work [19, 20, 32, 33, 35, 41, 44]. Through a few fine-tuning steps, our method can focus on both local details and global structures and thus boost the adaptation methods. Specifically, our CLCM-Adapter consists of the following steps:

Local Construction (Section 3.2): We first extract various local features by utilizing CLIP's visual encoder, where each local feature focuses on specific local content. Next, we employ the CLIP classifier to filter the local features, where local content most relevant to the corresponding category is selected, and irrelevant local content is filtered out. Finally, we construct a new local representation by fusing the selected local features. We believe that the constructed representation contains discriminative local details, and it can help the model better analyze the detailed information of the image, and then help the model expand its perception of the target when the training samples are insufficient.

Content Matching (Section 3.3): We design a global-local matching strategy to help the classifier attend to both global and local information. Specifically, we devise a co-training mechanism by designing global and local classifiers with textual features to match the global features and local features of the image, respectively. In this process, the global features and local features are jointly fine-tuned to obtain better matching, while helping the model perceive the content of the sample in different fields of view. Moreover, we combine both classifiers to analyze the image content from multiple perspectives, thereby realizing sample recognition. Our contributions are summarized as threefold:

1. We pay attention to the local content of the image, and design a local construction by selectively combining local content to construct robust local representation, which helps the classifier focus on the details of the image.

2. We match both global and local content by jointly fine-tuning both global and local classifiers, and then analyze the image content from different views.

3. We conduct extensive experiments with 11 few-shot classification datasets, and our method achieves significant performance improvements over current methods.

2 Related Works

In this section, we first briefly introduce vision-language models and recent vision-language model adaptation methods, and then we list related global-local content learning methods. Finally, we enumerate the differences between our method and related methods.

2.1 Vision-Language Models

In recent years, with the development of data-driven networks [6, 8, 31], vision-language models [12, 28] have achieved significant zero-shot image recognition performance and strong generalization abilities by contrastively pre-training over large-scale image-text data. Subsequent works further improve the effectiveness of such methods by enhancing the vision-language pre-training process. Specifically, FLIP [17] incorporates masking strategies [10] over the visual encoder, the works in [16, 23, 37] introduces more self-supervision strategies, and BLIP [15] focuses on data cleaning, it employs data filter mechanism for effective pre-training.

2.2 Adaptation for Vision-Language Models

Current methods further demonstrate effectiveness by designing strategies to adapt Vision-Language Models to various downstream tasks [7, 39, 43]. In this work, we focus on the few-shot learning task, which recognize novel objects with very limited training samples. As a seminal work for this task, CoOp [43] proposes a prompt learning-based strategy, it designs learnable continuous tokens over the textual encoders to replace the handcrafted textual templates. Follow-up work in [42] further improves the domain generalization performance by proposing a lightweight network to generate image-conditional tokens. the works in [22, 26] achieves significant performance gains by designing multiple hand-crafted prompts. Moreover, the works in [13, 45] observe an over-fitting phenomenon in CoOp, and address this by introducing regularization mechanisms.

Different from prompt learning methods, CLIP-Adapter [5] proposes a feature adapter method, which designs lightweight architectures over the visual and textual backbones. Zhang *et al.* [39] proposes a cache module to cache the few training samples, which can be combined with the CLIP model for image recognition. Following works further enhance the cache module by introducing inter-modal distance [30], prior refinement [46], and knowledge from other foundation models [40].

2.3 Global-Local Content Learning

Our work draws inspiration from some global-local content learning methods. Therefore, we list some related works in this subsection, such as [18, 38, 41, 44]. Specifically, BML [44] proposes a meta-learning-based local branch to capture discriminative local information, and the works in [38, 41] extracts local features by cropping the inputted images.

Different from those related works [38, 41, 44], which focus on improving the effectiveness through either backbone pre-training or introducing a large amount of training parameters to large-scale fine-tuning, our method is efficient, we only require a few fine-tuning steps. Moreover, compared to existing adaptation methods [30, 34, 39, 47], which aim to boost performance over a single global branch, we focus on enhancing the adaptation model with both global information and local details.

3 Approach

In this section, we first overview our GLCM-Adapter. Next, we revisit some preliminaries and then illustrate our method in detail. Finally, we describe the training and inference process. The overview of our GLCM-Adapter is illustrated in Figure 1, given a training sample, we extract its global and local features, and we subsequently construct a robust local representation by averaging the selected top-K local features, and finally, we design an effective classifier by matching the global and local classifiers.

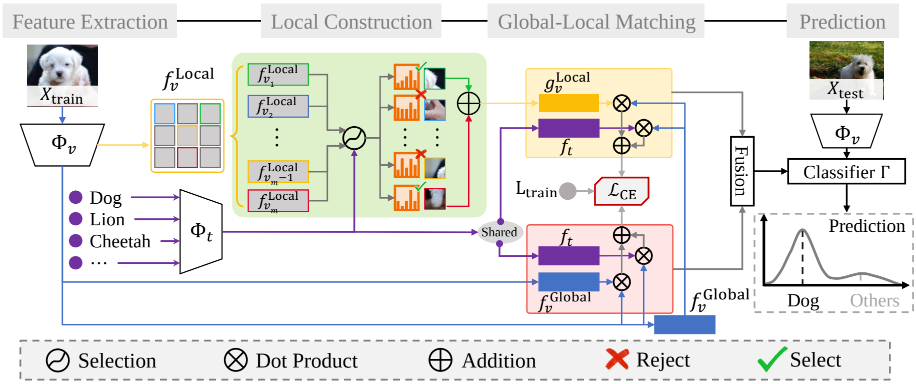

Figure 1: An overview of our GLCM-Adapter, given a training sample, we first extract its global and local features, and then we selectively construct a robust local representation, finally, we match the local and global classifiers to facilitate a robust classifier.

3.1 Preliminaries

Few-Shot Learning. The data for few-shot learning is divided into 2 parts: support set $\mathcal{D}_{\text{support}}$, and testing set $\mathcal{D}_{\text{test}}$, and they have the same categories. The goal of few-shot learning is to learn an image classification model that generalizes well to the N-way-K-shot task. Training samples for the N-way-K-shot task are sampled from $\mathcal{D}_{\text{support}}$ and the testing samples belong to $\mathcal{D}_{\text{test}}$, a N-way-K-shot task identifies N categories, and each category has K support samples.

Constrastive Languange-Image Pre-training (CLIP). CLIP consists of an image encoder $\boldsymbol{\Phi}_v$ and a text encoder $\boldsymbol{\Phi}_t$. After pre-training, CLIP can be effectively generalized to downstream tasks with fixed encoders. Specifically, Given C hand-crafted prompts $\boldsymbol{T}$ (*e.g.* the word embedding of “a photo of a $\{class\}$”, $\{class\}$ is the category name), where C is the number of categories, CLIP first represents them into features, denoted as $\boldsymbol{f}_t = \boldsymbol{\Phi}_t(\boldsymbol{T})$, and $\boldsymbol{f}_t \in \mathbb{R}^{C\times d}$. Then the classification logits can be calculated as:

$$\text{logits}_{\text{CLIP}} = cos\langle \boldsymbol{f}_t, f_{\text{test}}\rangle/\tau, \tag{1}$$

where $f_{\text{test}} = \boldsymbol{\Phi}_v(X_{\text{test}})$, and $f_{\text{test}} \in \mathbb{R}^d$, it is the extracted feature of test sample, and τ is the temperature coefficient learned in the pre-training phase, $cos\langle\cdot,\cdot\rangle$ denotes the cosine similarity, and $\text{logits}_{\text{CLIP}} \in \mathbb{R}^C$, it aims to classify the input sample into C different categories.

3.2 Local Construction

Our local construction selects local features and then fuses the selected local features to construct a robust representation. For convenient illustration, we utilize 1 training sample to describe our local construction process. Specifically, we remove the last pooling layer of the CLIP visual encoder to extract the local features, and given a training sample with labels as $(X_{\text{train}}, L_{\text{train}})$, we firstly extract its features by:

$$\begin{aligned} L_{\text{onehot}} &= \text{Onehot}(L_{\text{train}}), \qquad (2)\\ f_v^{\text{Global}}, \boldsymbol{f}_v^{\text{Local}} &= \boldsymbol{\Phi}_v(X_{\text{train}}), \bar{\boldsymbol{\Phi}}_v(X_{\text{train}}), \qquad (3) \end{aligned}$$

where $\bar{\mathbf{\Phi}}_v(\bullet)$ is the CLIP visual encoder removing the last pooling layer, Onehot$(\bullet)$ transforms the input label into one-hot vector, and thus $L_{\text{onehot}} \in \mathbb{R}^C$. Moreover, we denote the global feature and the local features as f_v^{Global} and $\boldsymbol{f}_v^{\text{Local}}$, respectively, where $f_v^{\text{Global}} \in \mathbb{R}^d$, and $\boldsymbol{f}_v^{\text{Local}} \in \mathbb{R}^{m\times d}$, m is the number of local features.

For local construction, we utilize the pre-trained CLIP model to calculate the classification score S for the local features, which can be formalized as:

$$S = \text{Softmax}(cos\langle f_t, \boldsymbol{f}_v^{\text{Local}}\rangle/\tau), \tag{4}$$

where Softmax$(\cdot)$ denotes the Softmax function, and $S \in \mathbb{R}^{C\times m}$. Then, we calculate the KL divergence D_{KL} between the classification score with the one-hot label to select local features, which can be formalized as:

$$dis = D_{\text{KL}}(L_{\text{onehot}}|S), \tag{5}$$

where $dis \in \mathbb{R}^m$. Thus we select local features with the top-K dis values, the selected local features are denoted as $\boldsymbol{G}_v^{\text{Local}}$, and $\boldsymbol{G}_v^{\text{Local}} \in \mathbb{R}^{K\times d}$. To construct a robust local representation, we fuse the selected local features by averaging them, formalized as:

$$g_v^{\text{Local}} = Averaging(\boldsymbol{G}_v^{\text{Local}}), \tag{6}$$

where the constructed local representation is g_v^{Local}. Compared with the global feature f_v^{Global}, which presents the global content, the local representation g_v^{Local} contains more detailed local information, and then further facilitate the construction of a robust local classifier.

3.3 Global-Local Matching

In this subsection, we elaborate on our global-local matching strategy. Specifically, we first illustrate the design of global and local classifiers, and then we describe the co-training strategy. For convenient description, we utilize the extracted test feature of the test sample X_{test} as f_{test}, and we only utilize the global features of the test samples with the purpose of reducing the computational and storage costs.

Specifically, given the global feature and the constructed local representation of the training sample X_{train}, we design classifiers for both local and global parts, respectively. We employ a cached module described in [39], the classification logits of the local cached module are calculated as:

$$\text{logits}_{\text{Cache}}^{\text{Local}} = \exp(-\beta^{\text{Local}}(1 - f_{\text{test}} \cdot g_v^{\text{Local}})) \bullet L_{\text{onehot}}, \tag{7}$$

where β^{Local} is the hyper-parameter for the local cached module. Then the local classifier can be written by fusing logits from the cached module with the pre-trained CLIP model, which can be written as:

$$\mathbf{\Gamma}^{\text{Local}} = \alpha^{\text{Local}}\text{logits}_{\text{Cache}}^{\text{Local}} + \text{logits}_{\text{CLIP}}, \tag{8}$$

where α^{Local} is the hyper-parameter to control the fusion process, and $\mathbf{\Gamma}^{\text{Local}}$ is the local classifier. Similarly, We calculate the global cached module and the classifier as:

$$\begin{aligned} \text{logits}_{\text{Cache}}^{\text{Global}} &= \exp(-\beta^{\text{Global}}(1 - f_{\text{test}} \cdot g_v^{\text{Global}})) \bullet L_{\text{onehot}}, && (9)\\ \mathbf{\Gamma}^{\text{Global}} &= \alpha^{\text{Global}}\text{logits}_{\text{Cache}}^{\text{Global}} + \text{logits}_{\text{CLIP}}, && (10)\end{aligned}$$

where the β^{Global} and α^{Global} are the hyperparameters of the global classifier, which has similar effects as described in the local classifier, and $\boldsymbol{\Gamma}^{\text{Global}}$ denotes the global classifier. To further match the local and global classifiers, we define our final classifier as the fusion of both classifiers, which can be formalized as:

$$\boldsymbol{\Gamma} = \gamma\boldsymbol{\Gamma}^{\text{Local}} + \boldsymbol{\Gamma}^{\text{Global}}, \tag{11}$$

where γ is the hyper-parameter to control the fusion ratio. Our co-training simultaneously fine-tunes the global and local classifiers with the fusion ratio γ, and thus the final classifier can focus on content from different scales.

3.4 Training and Inference

For the training stage, given N-way-K-shot support samples from $\mathcal{D}_{\text{support}}$, we firstly extract their local representations and global feaures, and then we construct the global and local classifiers. In our method, to keep the correspondence of the training and inference stages, we utilize the global features for classifier training, where we denote the global features of the support samples as $\boldsymbol{f}_v^{\text{Global}}$, and the labels as y_{train}, where $\boldsymbol{f}_v^{\text{Global}} \in \mathbb{R}^{NK\times d}$ and $y_{\text{train}} \in \mathbb{R}^{NK}$.

Specifically, despite setting the local representations and global features tuneable, we also apply a linear classifier over the CLIP textural features, which can be formalized as:

$$\bar{\boldsymbol{f}}_t = W * \boldsymbol{f}_t + b. \tag{12}$$

wher W and b are learnable parameters. Meanwhile, we use the cross-entropy(CE) loss with the support feature-label pairs to train the final classifier. The training loss can be formalized by the following equation:

$$\mathcal{L} = \frac{1}{NK}\sum_{i=1}^{NK} \text{CE}((\boldsymbol{\Gamma}(\boldsymbol{f}_v^{\text{Global}}), y_{\text{train}}), \tag{13}$$

With a few of optimization steps, the classifier can be generalized well for the few-shot tasks. For the inference stage, the classification is achieved by replacing the original parameters with the corresponding well-optimized ones. Compared to the original classifier, our GLCM-Adapter not only concentrates on the local details, but also attends to the global structural information, achieving a robust recognition of the object.

4 Experiments

In this section, we conduct experiments to validate the effectiveness of our **GLCMAdapter**. Specifically, we first introduce the experimental settings, then we analyze the ablation studies, then we describe the comparison results with the state-of-the-art methods. Our experiments aim to address the following research questions (**RQ**s):
RQ1: What are the effects of local construction?
RQ2: What are the influences of global local matching?
RQ3: How does our proposed GLCM-Adapter perform compared with the SOTA methods?

Methods	ImageNet					Average of 11 datasets				
	1	2	4	8	16	1	2	4	8	16
$K=1$	62.31	62.86	63.44	64.70	66.02	66.61	69.06	72.07	74.66	77.91
$K=5$	62.39	62.93	63.45	64.70	66.08	66.73	69.33	72.39	74.94	78.06
$K=10$	62.41	62.98	63.49	64.76	66.09	66.73	69.71	72.46	75.00	78.11
$K=20$	62.42	63.05	63.50	64.81	66.13	66.77	69.62	72.52	75.06	78.15
$K=30$	**62.43**	63.07	63.52	64.82	66.14	66.80	69.69	72.57	75.07	78.17
$K=40$	62.42	63.10	63.54	64.87	66.15	66.81	69.81	72.59	75.08	78.19
$K=49$	62.42	**63.12**	**63.57**	**64.88**	**66.16**	**66.83**	**69.90**	**72.61**	**75.11**	**78.21**

Table 1: The accuracy (%) of the local classifier with different K for local construction.

4.1 Experimental Settings

Datasets. To validate the effectiveness of our method, we conduct experiments over the few-shot classification task with 11 widely used datasets. Specifically, the 11 few-shot datasets including ImageNet [3], Caltech101 [4], DTD [2], EuroSAT [11], FGVCAircraft [21], Food101 [1], OxfordFlowers [24], OxfordPets [25], StanfordCars [14], SUN397 [36], UCF101 [29].

Implementation Details. Our experiments follow the work in [39], specifically, we design our method with ResNet50 [9] and the modified transformer [27] as visual and textual encoder, respectively. Moreover, all parameters are optimized with 20 epochs, and we employ the pre-trained CLIP model with the hand-crafted textual prompts in [26] for local construction and few-shot comparison.

4.2 Ablation Studies

In this subsection, we use the validation sets of 11 few-shot datasets to evaluate the effectiveness of different components of our method. For convenience, we show the experimental results of the ImageNet dataset and an average of 11 datasets. Specifically, we conduct experiments with 1/2/4/8/16-shot training samples for evaluation, where the training sets are constructed by following [39].

4.2.1 The effects of local construction (RQ1).

In this ablation study, we conduct experiments to validate the effects of our local construction from 2 aspects: (1) sensitivity to different numbers of K and (2) comparison with different selection criteria. Meanwhile, all experiments are conducted with our local classifier.

Specifically, the experimental results for different numbers of K are illustrated in Table 1. For a clear illustration, we present the line chart with the performance results of different training samples, where the total number of K is 49, the top line shows the ImageNet results, and the bottom line describes the average results. We can observe that with K increases, the performance of the local classifier increases and gradually stabilizes as K= 10, where the variances between K=10 and the optimal results are less than 0.1%, which is marginal. Thus, to mitigate further computational and storage costs, we select K=10 for local construction.

Meanwhile, the experimental results with different selection criteria are shown in Table 2, where "Random-Select" denotes that we select local features randomly, "Max-Margin"

Methods	**ImageNet**					**Average of 11 datasets**				
	1	2	4	8	16	1	2	4	8	16
Random-Select	62.37	62.96	63.43	64.65	66.04	66.51	69.61	72.43	74.79	77.98
Max-Margin	62.37	62.94	63.45	64.73	65.99	66.61	69.33	72.29	74.89	77.95
Min-Margin	62.32	62.94	63.41	64.58	66.05	66.57	69.34	72.30	74.91	77.98
Ours	**62.41**	**62.98**	**63.49**	**64.76**	**66.09**	**66.73**	**69.71**	**72.46**	**75.00**	**78.11**

Table 2: The accuracy (%) of the local classifier with different selection criteria.

Methods	**ImageNet**					**Average of 11 datasets**				
	1	2	4	8	16	1	2	4	8	16
Local Branch	62.41	62.98	63.49	64.76	66.09	66.73	69.71	72.46	75.00	78.11
Global Branch	62.40	63.12	63.52	64.85	66.12	66.73	69.98	72.56	75.12	78.16
Global-Local	**62.44**	**63.15**	**63.56**	**64.89**	**66.18**	**66.85**	**70.16**	**72.70**	**75.26**	**78.24**

Table 3: The accuracy (%) of different parts of our method.

and "Min-Margin" represent that we select local features with maximum and minimum prediction margins, respectively, and "Ours" is the local branch with our selection criterion. Specifically, we can observe that our selection criterion achieves the best performance for all experimental settings, which demonstrates its effectiveness.

4.2.2 The influences of global-local matching (RQ2).

In this ablation study, we conduct experiments to evaluate the performance of our global local matching strategy. The experimental results are shown in Table 3, where "Local Branch" and "Global Branch" denote our local classifier and global classifier, respectively, and "Global-Local" is our final classifier. Specifically, we can find that (1) our global-local matching achieves the best performances for all settings, and the most average gains over the local branch is around 0.7%. (2) the performance of "Global Branch" is slightly better than "Local Branch", we believe the reason is that the "Local Branch" selectively utilizes local features, while missing some relative information.

4.3 Comparison with other methods (RQ3)

In this subsection, we compare our method with state-of-the-art methods for few-shot classification. Specifically, we follow existing methods [30, 39] to conduct few-shot classification with 1/2/4/8/16-shot training samples for comparison. The experimental results are shown in Figure 2, where the compared methods include Zero-Shot CLIP [28], Linear Probing CLIP [28], CoOp [43], training-free and training-need TIP-Adapter [39] (denoted as TIP-Adapter and TIP-Adapter-F, respectively), and TIP-X [30]. Following are detailed illustrations, where "GLCM-Adapter" represents our proposed GLCM-Adapter. From the average results, we can find that our GLCM-Adapter yields remarkable performance improvement over all compared methods. Specifically, it brings more than 1% performance improvement over the Tip-Adapter-F method for all shot settings, and the best performance gain is more than 2.5% for 2-shot, which is significant. Meanwhile, we can observe that on the large-scale ImageNet dataset, our method obtains significant results, it achieves more than 1% accuracy gain over current methods with about 64.90% and 66.15% accuracy for 8-shot, and

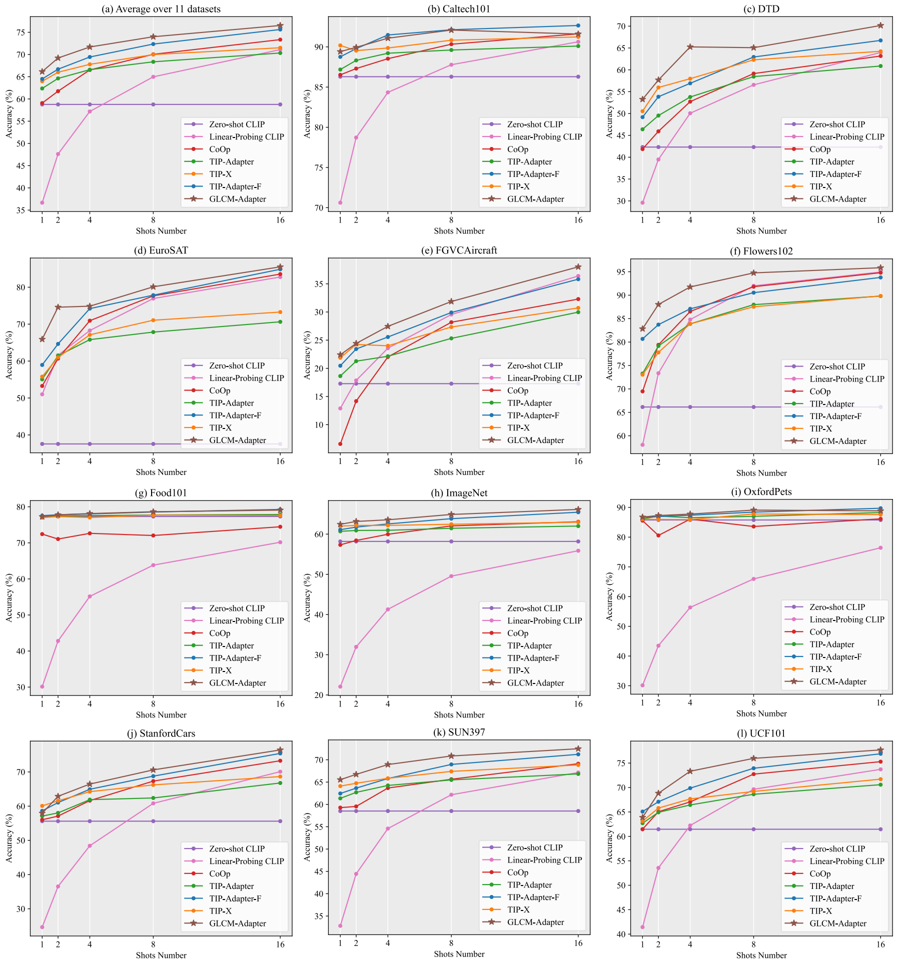

Figure 2: Few-shot performance with different methods on 11 datasets, we first show the average results, and the following are organized in the order of dataset names.

16-shot settings, respectively. Moreover, our GLCM-Adapter achieves a new state-of-the-art on DTD dataset for all shot settings, with 65.25% and 70.15% performance for 4-shot and 16-shot, respectively.

5 Conclusion

In this paper, we discuss the utilization of local content of CLIP adaptation methods and propose a **G**local-**L**ocal **C**ontent **M**atching (GLCM-Adapter) method to address this issue. Specifically, (1) The local construction is proposed to select discriminative local details and then construct a robust local representation. (2) The global-local matching is designed to enhance the robustness of the classifier for capturing multi-scale content. (3) Extensive experiments on 11 few-shot datasets demonstrate the effectiveness of our proposed methods.

We notice that our method utilizes local information from the CLIP model. But we believe that introducing local content detectors can further improve the model's capacity and generalization ability. And we focus on exploring some local content detectors, and then combine them with the CLIP model to further enhance the global-local content learning process in our future work.

6 Acknowledgments

The work was supported by the National Natural Science Foundation of China (Grants No. 62202439). This work was also supported by the advanced computing resources provided by the Supercomputing Center of the USTC.

References

[1] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI 13*, pages 446–461. Springer, 2014.

[2] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3606–3613, 2014.

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 248–255, 2009.

[4] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshop*, pages 178–178, 2004.

[5] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *International Journal of Computer Vision*, 132(2):581–595, 2024.

[6] Yuan Gao, Jinghan Li, Xiang Wang, Xiangnan He, Huamin Feng, and Yongdong Zhang. Revisiting attack-caused structural distribution shift in graph anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[7] Dan Guo, Shuo Wang, Qi Tian, and Meng Wang. Dense temporal convolution network for sign language translation. In *IJCAI*, pages 744–750, 2019.

[8] Dan Guo, Kun Li, Bin Hu, Yan Zhang, and Meng Wang. Benchmarking micro-action recognition: Dataset, method, and application. *IEEE Transactions on Circuits and Systems for Video Technology*, 2024.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[10] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[11] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. 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*, 12(7):2217–2226, 2019.

[12] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pages 4904–4916. PMLR, 2021.

[13] Muhammad Uzair Khattak, Syed Talal Wasim, Muzammal Naseer, Salman Khan, Ming-Hsuan Yang, and Fahad Shahbaz Khan. Self-regulating prompts: Foundational model adaptation without forgetting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15190–15200, 2023.

[14] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013.

[15] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pages 12888–12900. PMLR, 2022.

[16] Yangguang Li, Feng Liang, Lichen Zhao, Yufeng Cui, Wanli Ouyang, Jing Shao, Fengwei Yu, and Junjie Yan. Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm. *arXiv preprint arXiv:2110.05208*, 2021.

[17] Yanghao Li, Haoqi Fan, Ronghang Hu, Christoph Feichtenhofer, and Kaiming He. Scaling language-image pre-training via masking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23390–23400, 2023.

[18] Jinda Lu, Shuo Wang*, Xinyu Zhang, Yanbin Hao, and Xiangnan He*. Semantic-based selection, synthesis, and supervision for few-shot learning. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 3569–3578, 2023.

[19] Jinda Lu, Shuo Wang, Yanbin Hao, Haifeng Liu, Xiang Wang, and Meng Wang. Rethinking visual content refinement in low-shot clip adaptation. *arXiv preprint arXiv:2407.14117*, 2024.

[20] Xu Luo, Longhui Wei, Liangjian Wen, Jinrong Yang, Lingxi Xie, Zenglin Xu, and Qi Tian. Rectifying the shortcut learning of background for few-shot learning. *Advances in Neural Information Processing Systems*, 34:13073–13085, 2021.

[21] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

[22] Sachit Menon and Carl Vondrick. Visual classification via description from large language models. *arXiv preprint arXiv:2210.07183*, 2022.

[23] Norman Mu, Alexander Kirillov, David Wagner, and Saining Xie. Slip: Self-supervision meets language-image pre-training. In *European Conference on Computer Vision*, pages 529–544. Springer, 2022.

[24] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pages 722–729. IEEE, 2008.

[25] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3498–3505, 2012.

[26] Sarah Pratt, Ian Covert, Rosanne Liu, and Ali Farhadi. What does a platypus look like? generating customized prompts for zero-shot image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15691–15701, 2023.

[27] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[28] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[29] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[30] Vishaal Udandarao, Ankush Gupta, and Samuel Albanie. Sus-x: Training-free name-only transfer of vision-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2725–2736, 2023.

[31] Shuo Wang, Dan Guo, Wen gang Zhou, Zheng jun Zha, and Meng Wang. Connectionist temporal fusion for sign language translation. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 1483–1491, 2018.

[32] Shuo Wang, Jun Yue, Jianzhuang Liu, Qi Tian, and Meng Wang. Large-scale few-shot learning via multi-modal knowledge discovery. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*, pages 718–734. Springer International Publishing, 2020.

[33] Shuo Wang, Xinyu Zhang, Yanbin Hao, Chengbing Wang, and Xiangnan He. Multi-directional knowledge transfer for few-shot learning. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 3993–4002, 2022.

[34] Shuo Wang, Jinda Lu, Haiyang Xu, Yanbin Hao, and Xiangnan He. Feature mixture on pre-trained model for few-shot learning. *IEEE Transactions on Image Processing*, 2024.

[35] Zhicai Wang, Yanbin Hao, Tingting Mu, Ouxiang Li, Shuo Wang, and Xiangnan He. Bi-directional distribution alignment for transductive zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19893–19902, 2023.

[36] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *Proceedings of the IEEE computer society conference on computer vision and pattern recognition*, pages 3485–3492, 2010.

[37] Jiarui Yu, Haoran Li, Yanbin Hao, Jinmeng Wu, Tong Xu, Shuo Wang, and Xiangnan He. How can contrastive pre-training benefit audio-visual segmentation? a study from supervised and zero-shot perspectives. In *Proceedings of the 34th British Machine Vision Conference (BMVC)*, 2023.

[38] Seonghoon Yu, Paul Hongsuck Seo, and Jeany Son. Zero-shot referring image segmentation with global-local context features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19456–19465, 2023.

[39] Renrui Zhang, Wei Zhang, Rongyao Fang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free adaption of clip for few-shot classification. In *European Conference on Computer Vision*, pages 493–510. Springer, 2022.

[40] Renrui Zhang, Xiangfei Hu, Bohao Li, Siyuan Huang, Hanqiu Deng, Yu Qiao, Peng Gao, and Hongsheng Li. Prompt, generate, then cache: Cascade of foundation models makes strong few-shot learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15211–15222, 2023.

[41] Fei Zhou, Peng Wang, Lei Zhang, Wei Wei, and Yanning Zhang. Revisiting prototypical network for cross domain few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20061–20070, 2023.

[42] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16816–16825, 2022.

[43] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022.

[44] Ziqi Zhou, Xi Qiu, Jiangtao Xie, Jianan Wu, and Chi Zhang. Binocular mutual learning for improving few-shot classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8402–8411, 2021.

[45] Beier Zhu, Yulei Niu, Yucheng Han, Yue Wu, and Hanwang Zhang. Prompt-aligned gradient for prompt tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15659–15669, 2023.

[46] Xiangyang Zhu, Renrui Zhang, Bowei He, Aojun Zhou, Dong Wang, Bin Zhao, and Peng Gao. Not all features matter: Enhancing few-shot clip with adaptive prior refinement. *arXiv preprint arXiv:2304.01195*, 2023.

[47] Xingyu Zhu, Shuo Wang*, Jinda Lu, Yanbin Hao, Haifeng Liu, and Xiangnan He. Boosting few-shot learning via attentive feature regularization. In *The 38th Annual AAAI Conference on Artificial Intelligence*, 2024.