Parameter-Efficient Fine-Tuning for Vision-Language Models

Zhuoheng Li[1], Zhuosheng Liu[2], and Jiawei Zhang[1]
[1]IFM Lab, Department of Computer Science
[2]Department of Food Science and Technology
University Of California, Davis
{pipli,zsliu}@ucdavis.edu, jiawei@ifmlab.org

Abstract

Parameter-efficient fine-tuning (PEFT) methods have gained widespread adoption in large language models (LLMs) due to their efficiency and efficacy. Expanding on this concept, our research explores the application of PEFT methods to vision-language models, with a particular emphasis on CLIP. We introduce a comprehensive evaluation framework that examines these methods across diverse backbones and datasets. The study reveals that while PEFT methods show strong performance in standard classification tasks, they face limitations in more complex, multimodal scenarios. Through an empirical examination of prompt tuning and adapter techniques, we highlight their potential to alleviate data collection challenges in data-scarce environments. The study also underscores the need for enhanced PEFT approaches for detailed scene understanding and decision-making tasks. We open-sourced our code at https://github.com/Andy-LZH/peft4clip

1. Introduction

In recent years, the machine learning community has seen remarkable advances, especially in large language models (LLMs). Models like ChatGPT, including GPT-3.5 and GPT-4, have set new standards in NLP. They're known for their ability to follow human instructions and learn from feedback, as shown in studies [41, 42]. This progress has led to the development of open-source LLMs like LLaMA, PaLM, Alpaca, and Vicuna, diversifying NLP research [11, 37, 50, 59].

Similarly, in computer vision, the Visual Transformer [14] has sparked a shift towards combining language and images for a deeper understanding of visual contexts. Models like CLIP, trained on text-image pairs, are leading the integration of vision-language. However, they face challenges like catastrophic forgetting and aligning language with vision in downstream tasks, prompting the need for new fine-tuning methods .[18, 25, 45, 49, 57].

The NLP field has extensively explored challenges for fine-tuning large models, particularly with large models like GPT-3 and its billions of parameters. Research also points out the high costs and risks of catastrophic forgetting of full-modal fine-tuning. In response, there is growing interest in parameter-efficient tuning methods like prompt tuning and adapters [8, 20, 21, 58].

Our paper seeks to fill a critical gap in existing research by conducting a comprehensive analysis of fine-tuning methods for Vision-Language models, like CLIP. We specifically focus on their performance in downstream tasks where data collection is challenging, offering new insights into their adaptability in these contexts.

Key Contributions of Our Study:

1. We offer a detailed review of fine-tuning methods, including prompt engineering, tuning, adapters, and LoRA-like techniques.
2. Our evaluation framework shows that current PEFT methods are effective in data-scarce domains and highlights the need for advanced methods for complex scene understanding and decision-making tasks.

2. PEFT in Vision-Language Model

2.1. Preliminary

CLIP (Contrastive Language - Image Pre-training) integrates vision and language processing through a dual encoder framework, that is, Vision Encoder and Text Encoder, using a contrastive learning objective [45]. This framework aligns image and text representations in a shared embedding space, enabling efficient cross-modal understanding.

Image Encoder in CLIP adopts a ResNet [18] or a

Vision Transformer (ViT) [14] architecture. With ResNet, CLIP leverages deep layers and residual connections for robust image feature extraction. Alternatively, ViT treats images as sequences of flattened patches, applying self-attention to capture global dependencies.

Language Encoder in CLIP uses a transformer architecture [13, 51], optimized to process textual input such as captions and descriptions, encoding them in the same embedding space as the Vision Encoder.

Formalizing CLIP is trained to minimize the distance between embeddings of matching image-text pairs and maximize it for nonmatching pairs, formalized as:

- Vision Encoder: $f(I)$ where I is an input image. The function f processes I and outputs a feature vector in a shared embedding space, facilitating alignment with textual data.
- Language Encoder: $g(T)$ where T is a text input. The encoder g transforms T into a corresponding vector in the same embedding space.
- Let $S(I, T)$ represent the cosine similarity score between the image embedding and the text embedding, calculated as the dot product of $f(I)$ and $g(T)$.
- The contrast loss function, $\mathcal{L}$, is designed to optimize this similarity metric across a batch of image-text pairs.

This architecture and training methodology enable CLIP to achieve state-of-the-art performance in tasks such as zero-shot image classification, demonstrating its effectiveness in cross-modal learning. Furthermore, employing a simple linear-probe approach, which mounts a linear classifier head to the vision encoder $f(I)$, achieved comparable results on datasets like ImageNet [47] to its supervised learning counterparts like ResNet [18], EfficientNet [49], and ViT [14] [45]. However, the original paper on CLIP did not introduce a method for few-shot fine tuning, and previous research [15, 28] has shown that full model fine tuning can lead to poor performance in the Out-Of-Distribution task, underscoring the need for new fine-tuning methods to enhance CLIP's adaptability.

2.2. Prompt Engineering

Although not a direct application of parameter-efficient fine-tuning, we felt it is important to also discuss prompt engineering in the context. Prompt Engineering has gained substantial popularity in both the NLP and CV community since its intuitive approach and great performance, and its importance has been furthered with the widespread adoption of the LLM tools.

Prompt engineering in NLP involves a strategic input structure to guide large language models (LLMs) such as GPT-2 [44], GPT-3[8], GPT-3.5[42] and GPT-4[41]. This field has developed a variety of methodologies and best practices. Among them, Instruction Prompting and In-Context Learning are notable. **Instruction Prompting** is used to provide clear and explicit instructions to LLM, improving the precision and relevance of their responses [40]. This method has been used effectively in models such as GPT-3.5[42] and GPT-4[41], which are designed to follow instructions and integrate human feedback efficiently. **In-context Learning** is another crucial technique in prompt engineering. It enables LLMs to infer and generate responses based on the context provided within the prompt itself. This method is particularly effective for tasks where models need to interpret or continue a given narrative or pattern, allowing them to produce contextually coherent and relevant output. Furthermore, **Channel-Of-Thought (CoT) prompting** allows LLMs to address problems through intermediate reasoning steps, closely resembling human thought processes. Prompt Engineering largely improved the performance of LLMs, and were also very well stuided by the community; hence on top of what we summarized here, we gently introduce the reader to read those phrases. [9, 17, 26, 35, 40].

Prompt engineering in CLIP involves prompt engineering in the parts of the language and vision, and while the language inherits a similar method in the LLM part, vision was relatively less studied. The importance of prompts(text supervision) has been widely explored in the original CLIP's paper, which with simply a text template of *(a photo of a [CLS])* allows the model to adapt to different task sets without being bounded to a fixed class size[45]. Although vision prompt engineering were less studied, we found some recent publications that share really exciting results. Researchers have shown that by simply drawing a red circle on the image where interested, CLIP achived SOTA in expression composition and localization subtasks[48]. We also encourage our readers to read these recent work in prompt engineering in Vision-Language Models[5, 17]

2.3. Prompt Tuning

Due to the significant time required for crafting prompts in Prompt Engineering, prompt tuning has become an essential technique in Parameter Efficient Fine Tuning. Originally focused on improving NLP, prompt tuning is now important in both the NLP and CV communities.[22, 30, 32, 60]

NLP The original concept of prompt tuning involves modifying the input prompt to enhance model performance. This approach can be categorized into hard prompt tuning, where discrete input tokens are rearranged for better output, and soft prompt tuning, where input token embeddings are

concatenated with trainable tensors. **Soft prompt tuning** is optimized through backpropagation and is more parameter efficient than full model fine-tuning, although it may sometimes offer slightly reduced modeling performance [30]. An advanced form of prompt tuning is **Prefix tuning**, which extends the idea by adding trainable tensors to each transformer block, not just the input embeddings. This method integrates soft prompt embeddings via fully connected layers, enhancing the flexibility and adaptability of the model. Prefix tuning has been shown to achieve modeling performance comparable to full model fine tuning in all layers, requiring only training of a fraction (0.1%) of the parameters. In several instances, Prefix Tuning has even outperformed full-layer fine tuning, particularly in scenarios involving smaller target datasets, as it helps reduce overfitting [1, 3, 32].

CV/Vision-Language Prompt tuning's wide application in NLP has influenced its adoption in CV. Visual Prompt Tuning (VPT) [22], which takes inspiration from soft prompt tuning [30] and prefix tuning [32], applies these concepts to visual tasks. **VPT-Shallow** integrates trainable tensors with image patches and position embeddings at the input stage, while **VPT-Deep** involves adding trainable tensors to every transformer block, enhancing performance in complex visual tasks by allowing for more detailed adjustments within transformer layers. **CoOP**[60] adapts the principles of soft prompt tuning to the vision-language domain. It focuses on optimizing contextual tokens specifically for image recognition tasks, thus improving the model's ability to interpret and analyze visual data in conjunction with textual information. **Unified Prompt Tuning** (UPT)[24, 54] which created a concatenated shared trainable tensor in both image and text encoder, introduced an interesting regime in Vision-Language prompt tuning which we would encourage readers to explore.

2.4. Adapters

In LLMs, adapters provide an efficient alternative to full model fine-tuning. By adding lightweight layers within the transformer architecture, these modules fine-tune the model for specific tasks while keeping the majority of the model's parameters frozen. This approach significantly reduces the computational cost and the time required for training.[21]

LoRA (Low-Rank Adaptation) is a prominent example of an adapter in NLP. LoRA modifies the self-attention and feedforward layers of a transformer model by introducing low-rank matrices. These matrices are trained to adjust the model's behavior, providing task-specific adaptation while maintaining a low computational footprint. This method has been effective in adapting LLMs for various NLP tasks with minimal training costs [20].

Just as adapters streamline the fine-tuning process for language models, they are also transforming how we combine text and images in Vision-Language models. The CLIP-Adapter is a key example. Specifically, the **CLIP-Adapter** adds small, targeted adjustments to CLIP's existing set-up with extra feature learned from CLIP's output combining with CLIP's original output, allowing it to better select adaptability to new tasks. This process does not require much extra computing power and can significantly improve how the model performs on tasks that need insights from both what it sees and what it reads [15].

3. Experiments

Despite the extensive exploration of Parameter-Efficient Fine-Tuning (PEFT) methods, a significant gap in the current literature is the predominant focus on ImageNet-related datasets. While these datasets have been instrumental in advancing our understanding of PEFT methods, they offer a limited view of potential applications, especially in the context of vision-language models.

Most existing studies have concentrated on a set of 11 standard datasets[54, 60], primarily oriented around ImageNet[47]. This concentration raises concerns about the generalizability of these methods across a broader spectrum of tasks and domains. There is a pressing need for a more systematic empirical study that explores a diverse range of adaptation benchmarks. Such a study is crucial to unlocking the true potential of vision language models, extending their impact beyond traditional image-focused tasks to other fields.

Our experiment aims to fill this gap by conducting a comprehensive empirical analysis in various data sets that are not exclusively connected to ImageNet[47]. This approach will enable a deeper understanding of the adaptability and effectiveness of PEFT methods in various scenarios. We believe that such a systematic study will be crucial in guiding future adaptations of vision-language models, contributing significantly to their applicability in a wide range of real-world applications.

3.1. Dataset

We follow VPT[22] to run our experiments on the VTAB-1K benchmark[55] which Incorporated 19 different tasks ranging from 3 categories: • NATURAL which are most similar to what on ImageNet, • SPECIALIZED which encompass medical or satellite imagery[1], and • STRUC-

[1]Our report temporarily omits results from the Retinopathy dataset [2] due to maintenance issues encountered with our framework.

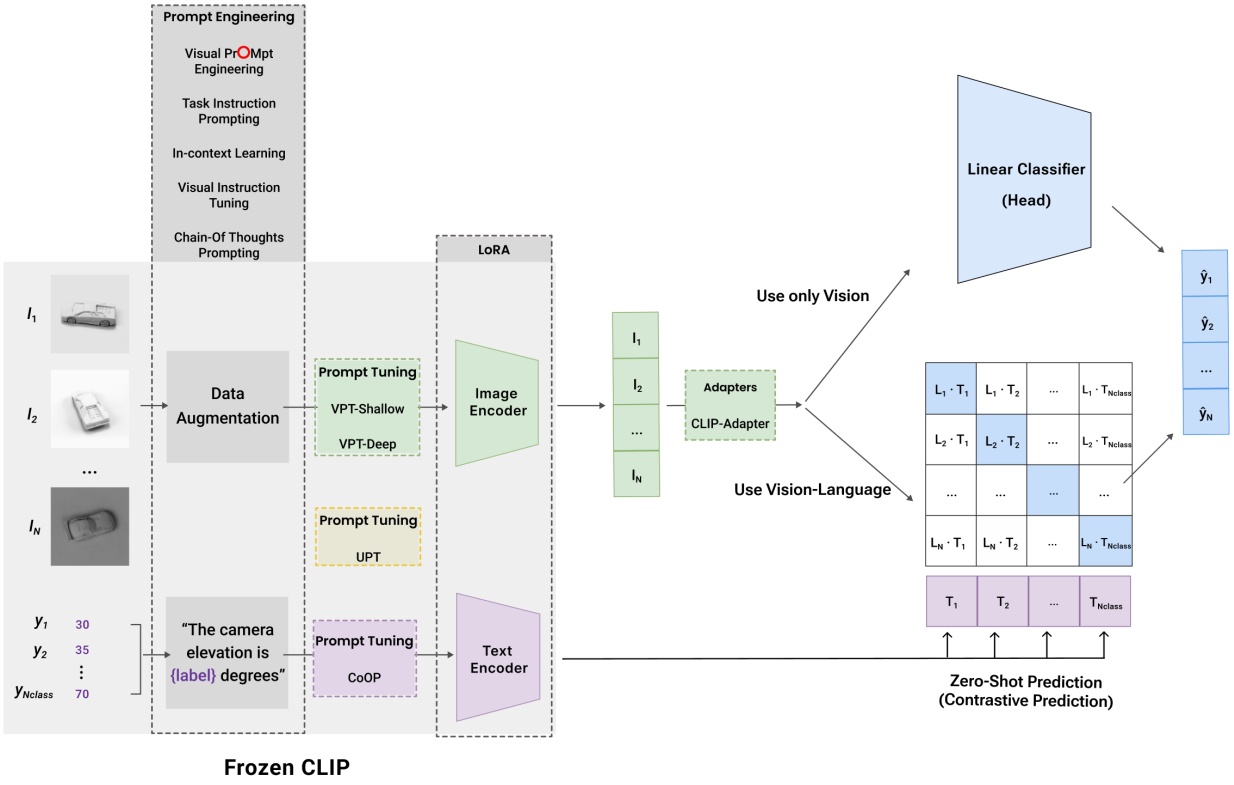

Figure 1. An overview of our evaluation pipelines

TURED which encompass complex scene understanding tasks like estimating distance, attitude, elevation, and game strategies. See Table.1 for a detailed attribute of the dataset.

• NATURAL							• SPECIALIZED				• STRUCTURED							
Caltech101[31]	CIFAR-100[27]	DTD[12]	Flowers102[39]	Oxford Pets[43]	SVHN[38]	Sun397[52]	PCam[7]	EuroSAT[19]	Resisc45[10]	Retinopathy[2]	Clevr/count[23]	Clevr/distance[23]	DMLab[6]	KITTI/distance[16]	dSprites/location[4]	dSprites/orientation[4]	SmallNORB/azimuth[29]	SmallNORB/elevation[29]

Table 1. Datasets in VTAB-1K[55].

3.2. Evaluation Pipelines

Refer to Figure 1 for an overview of our evaluation pipeline. This figure illustrates the specific optimization components utilized in each PEFT method and outlines our approach to test the robustness of adaptability.

Methods Our study focuses primarily on parameter-efficient fine-tuning methods related to image processing, selected on the basis of time and resource constraints. The main PEFT methods we examine are VPT-Shallow [22], VPT-Deep [22], and CLIP-Adapter [15].

Backbone In line with recent findings on the impact of the CLIP pre-training data scale [9], we extend our experiments to include new backbones from MetaCLIP [9], trained on both 400M and 2.5B datasets. Given the demonstrated success of these models in previous studies [15, 22, 60] and due to page limitations, we report results exclusively from the MetaCLIP-B16-2.5B backbone. For a comprehensive analysis of different backbones, please refer to our Supplementary material.

Inference At the output stage, we employ two types of evaluation, following CLIP [45]'s original design: the Linear Probe and Zero-Shot Prediction. The two output types in our pipeline are: 'Head', utilizing only visual features, and 'Contrastive Prediction', which leverages both textual and visual features. While VPT primarily used a linear probe approach due to its focus on addressing PEFT issues in Visual Transformers [14], which lack a textual component. The CLIP-Adapter was originally designed for CLIP; however, we also explore the robustness of CLIP-Adapter with only visual features in our experiments.

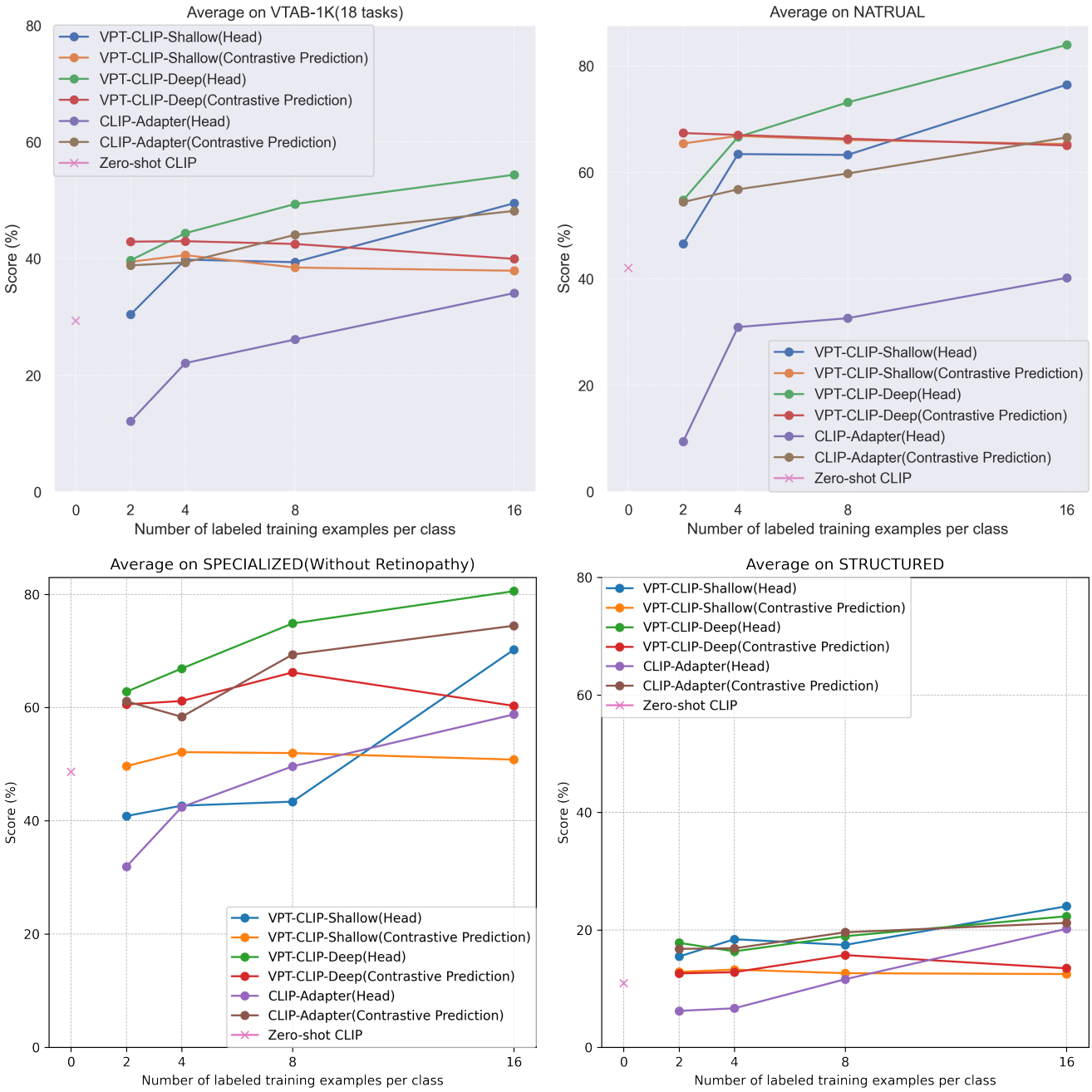

Figure 2. Comparative Analysis of Average Accuracy Across VTAB-1K[55] Subsets (Natural, Structured, and Specialized) and Overall Performance Versus Training Shots per Class for Various PEFT Methods using the MetaCLIP-B16-2.5B Backbone[53]

Formally, we defined our evluation method as follows: Let M be a set of PEFT methods where $M = \{\text{VPT-Shallow}[22], \text{VPT-Deep}[22], \text{CLIP-Adapter}[15]\}$. Let O be the set of output types where $O = \{\text{Vision (head)}, \text{Vision-Language (Contrastive Prediction)}\}$. Let S be the set of number of shots used in training where $S = \{2, 4, 8, 16\}$. Then, we define the function $f : M \times O \times S \rightarrow \mathbb{R}$, where $f(m, o, s)$ yields the performance metric.

3.3. Hyperparameter Tuning

Hyperparameter tuning is a critical stage in model optimization, aiming to find the most effective model settings.

	PEFT Methods Grid Search	
	CLIP-Adapter[15]	**VPT-Shallow, VPT-Deep[22]**
Optimizer	AdamW [36]	SGD[46]
Optimizer momentum	-	0.9
base_lr range	{0.001, 0.0001, 0.0005, 0.005}	{50, 25, 10, 5, 1}
Weight decay range	{0.01, 0.001, 0.0001, 0}	{0.01, 0.001, 0.0001, 0}
Num Tokens	-	{10-200}
alpha α	0.5	-
Total epochs	ViT-B[45, 56]{10-50}, ViT-L[45]{5-30}	

Table 2. Hyperparameters used for different method categories, adapted from [22]

We implemented a grid search method, systematically evaluating combinations of hyperparameters to ascertain the optimal configuration for our models.

In Visual Prompt Tuning introduced 'Num Token' as an additional hyperparameter, which represents the count of trainable tokens for the visual prompts. We followed Prompt Tuning's implementation details in choose Num Tokens [22] to run our experiments.

For CLIP-Adapter[15], the hyperparameter 'alpha' plays a crucial role, as mathematically defined by:

$f(I)' = \alpha * f_{adapter}(f(I)) + (1 - \alpha) * f(I)$ where f_{adapter} denotes the output embedding from CLIP-Adapter, $f(I)$ represents the original knowledge (embedding) from CLIP's image encoder, and $f(I')$ is the resulting embedding after tuning. The hyperparameter α ranges between 0 and 1, balancing the weight of the adapted parameters Θ' corresponding to the weights of f_{adapter} with the original parameters Θ of the original knowledge of CLIP $f(I)$, thus controlling the degree of adaptation.

Guided by the loss function used in CLIP and the hyperparameters outlined in Table 2, our tuning process also took inspiration from previously successful methods, where we adapt from VPT[22] in grid searching optimal hyperparameter.

3.4. Our Results

Following our evaluation pipeline, we conducted zero-shot and few-shot experiments on VTAB [55] as shown in Fig. 2

3.4.1 Observation

Across the different subsets of the VTAB-1K benchmark[55] (natural, structured, and specialized), we observe a distinct pattern in the performance of methods using Head-only prediction versus those employing Contrastive Prediction.

In the Head-only setup, VPT-Shallow[22] and CLIP-Adapter[15] show a steady increase in performance as the number of labeled training examples per class increases. This trend is indicative of the effectiveness of Head-only prediction in scenarios where visual features predominantly drive the task. VPT-Deep[22], while also benefiting from more labeled data, appears to produce more modest improvements, suggesting that the deeper integration of trainable tensors may not always translate to a linear performance increase with additional training examples.

Contrastive prediction, which uses text and images together, shows a surprising result with VPT-Shallow[22] and VPT-Deep[22]: their accuracy drops as we increase the number of training examples per class. This isn't well explained yet, but we think it might be because the model is fitting too closely to the specific training examples, especially in how input embedding from vision side were structured. This suggests that we need better PEFT methods that can handle text and images together without over-fitting.

The CLIP-Adapter[15]'s performance using Contrastive Prediction notably surpasses its Head-only configuration, underscoring the importance of aligning visual and textual representations, especially when adapting preexisting models like CLIP to new tasks.

Interestingly, across all methods and subsets, zero-shot CLIP serves as a baseline, with its performance being outpaced by most fine-tuned approaches. This reinforces the value of fine-tuning in specialized domains, where the zero-shot capabilities of models like CLIP may not fully capture the nuances required for optimal performance.

3.4.2 Implications of PEFT in Diverse Domains

From our observation, we concluded that current PEFT methods are quite effective for standard classification

tasks across different domains, showing promise especially where data is scarce. However, these methods seem to underperform in tasks involving complex scene understanding and decision making. This indicates a need for improved PEFT techniques that can handle the nuances of such complex scenarios more effectively.

- **Natural:** PEFT methods, including Zero-Shot learning, have shown remarkable effectiveness in natural data-rich environments. Such methods could lower barriers for researchers in domains where gathering large datasets is straightforward or can be synthetically generated.

- **Specialized:** In more specialized settings, PEFT methods like VPT-Deep have nearly achieved an 80% accuracy rate using only visual features, a noteworthy improvement over baseline methods such as Sup-Rotation-100% [55]. The CLIP Adapter, which uses both visual and textual information, also demonstrates comparable performance. This indicates that even with a domain gap, a slight amount of fine-tuning can significantly boost accuracy. This trend is particularly evident in domains such as medical and satellite imaging, suggesting that adapters or VPT can effectively learn new weights to emphasize relevant features for classification tasks. We encourage researchers, especially those in data-scarce fields such as medicine, to explore the use of PEFT with vision-language models such as CLIP for their classification challenges [15].

- **Structured:** Despite the success in natural and specialized tasks, PEFT methods have struggled in structured domains. Datasets that require complex scene understanding, such as estimating distances in RGB images without LiDAR as in KITTI [16], or categorizing objects in a simulated environment, as in DMLab [6], have proven challenging. Yet, with vision-language models showing promise in VQA problems [33, 34], there is potential in exploiting CLIP's dual-modality for complex reasoning. We call upon the community to direct more focus towards PEFT methods in challenging domains that require intricate scene understanding and decision-making.

4. Conclusion

Our emprical study on Parameter Efficient Fine Tuning (PEFT) methods for Vision-Language models, particularly focusing on the CLIP model, underscores the potential and challenges in this rapidly evolving field. We have demonstrated that while current state-of-the-art methods exhibit robust performance in standard image classification tasks, particularly those akin to ImageNet datasets, they exhibit limitations when applied to more complex scene understanding and decision-making tasks.

The exploration of various PEFT techniques, including prompt tuning, adapters, and LoRA-like methods, reveals that these approaches can significantly mitigate the challenges of data scarcity. This is particularly pertinent for domains where collecting large, annotated datasets is impractical or unfeasible. Our findings suggest that the application of PEFT methods can enable effective model adaptation with minimal additional training, thereby reducing computational costs and time.

However, the results also highlight a critical gap in the capability of these methods in dealing with complex multimodal scenarios. This gap points to the need for more advanced PEFT strategies that can better integrate and balance textual and visual information for nuanced scene comprehension. Such advancements could pave the way for more generalized and versatile applications of vision-language models across a broader spectrum of tasks, beyond standard image classifications.

4.1. Limitations

Our empirical study, while extensive in its scope and depth, encountered certain limitations that are important to acknowledge. Firstly, our ambition to test a wide array of backbones, including OpenAI's ViT-B32, ViT-B16, and MetaCLIP's B32-400M, B16-400M, B32-2.5B, and B16-2.5B, meant that the sheer volume of tests and datasets we aimed to cover was substantial. This ambition, while valuable for comprehensive analysis, posed practical challenges.[45, 56]

Due to constraints in computational resources and people power, our study focused on a limited set of parameter-efficient fine-tuning (PEFT) methods. While this focus allowed a detailed exploration of specific techniques, it also meant that other potentially valuable methods, such as LoRA, CoOP, UPT, and other notable methods, were not included in this phase of our research.[20]

4.2. Future Directions

To address these limitations, we are planning follow-up works and supplemental studies. These will expand our analysis to include a broader range of PEFT methods, such as LoRA and advanced prompt engineering techniques. By doing so, we aim to provide a more comprehensive understanding of the effectiveness of these methods across different model architectures and datasets.

Furthermore, we intend to publish detailed ablation studies on the various backbones used in our research. This additional analysis will be included in supplementary

materials, offering deeper insights into the performance and characteristics of each backbone. These efforts are aligned with our goal to continually contribute to the field and support the machine learning community with more thorough and diverse research findings.

References

[1] Peft: Parameter-efficient fine-tuning of billion-scale models on low-resource hardware. 3

[2] Diabetic retinopathy detection. https://www.kaggle.com/c/diabetic-retinopathy-detection. Accessed: [Insert date here]. 4

[3] Understanding parameter-efficient llm finetuning: Prompt tuning and prefix tuning. 3

[4] dsprites: Disentanglement testing sprites dataset. https://github.com/deepmind/dsprites-dataset/, 2017. 4

[5] Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola. Exploring visual prompts for adapting large-scale models, 2022. 2

[6] Charles Beattie, Joel Z Leibo, Denis Teplyashin, Tom Ward, Marcus Wainwright, Heinrich Küttler, Andrew Lefrancq, Simon Green, Víctor Valdés, Amir Sadik, et al. Deepmind lab. *arXiv preprint arXiv:1612.03801*, 2016. 4, 7

[7] Babak Ehteshami Bejnordi, Mitko Veta, Paul Johannes van Diest, Bram van Ginneken, Nico Karssemeijer, Geert Litjens, Jeroen AWM van der Laak, and the CAMELYON16 Consortium. Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. *JAMA*, 318(22):2199–2210, 2017. 4

[8] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. 1, 2

[9] Banghao Chen, Zhaofeng Zhang, Nicolas Langrené, and Shengxin Zhu. Unleashing the potential of prompt engineering in large language models: a comprehensive review, 2023. 2, 4

[10] Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017. 4

[11] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways, 2022. 1

[12] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014. 4

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 2

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1, 2, 4

[15] 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, 2021. 2, 3, 4, 5, 6, 7

[16] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3354–3361, 2012. 4, 7

[17] Jindong Gu, Zhen Han, Shuo Chen, Ahmad Beirami, Bailan He, Gengyuan Zhang, Ruotong Liao, Yao Qin, Volker Tresp, and Philip Torr. A systematic survey of prompt engineering on vision-language foundation models, 2023. 2

[18] 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. 1, 2

[19] 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*, 2019. 4

[20] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. 1, 3, 7

[21] Zhiqiang Hu, Lei Wang, Yihuai Lan, Wanyu Xu, Ee-Peng Lim, Lidong Bing, Xing Xu, Soujanya Poria, and Roy Ka-Wei Lee. Llm-adapters: An adapter family for parameter-efficient fine-tuning of large language models, 2023. 1, 3

[22] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning, 2022. 2, 3, 4, 5, 6

[23] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2901–2910, 2017. 4

[24] Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Maple: Multi-modal prompt learning, 2023. 3

[25] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. 1

[26] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners, 2023. 2

[27] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009. 4

[28] Ananya Kumar, Aditi Raghunathan, Robbie Matthew Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *International Conference on Learning Representations*, 2022. 2

[29] Yann LeCun, Fu Jie Huang, and Léon Bottou. Learning methods for generic object recognition with invariance to pose and lighting. Technical report, CITATION, Courant Institute of Mathematical Sciences, 2004. 4

[30] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning, 2021. 2, 3

[31] Fei-Fei Li, Marco Andreeto, Marc'Aurelio Ranzato, and Pietro Perona. Caltech 101, 2022. 4

[32] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online, 2021. Association for Computational Linguistics. 2, 3

[33] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023. 7

[34] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. 7

[35] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing, 2021. 2

[36] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. 6

[37] Kiwan Maeng, Alexei Colin, and Brandon Lucia. Alpaca: Intermittent execution without checkpoints, 2019. 1

[38] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011. 4

[39] 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*. IEEE, 2008. 4

[40] OpenAI. How to work with large language models, 2023. 2

[41] OpenAI. Gpt-4 technical report, 2023. 1, 2

[42] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. 1, 2

[43] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012. 4

[44] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018. 2

[45] 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. 1, 2, 4, 6, 7

[46] Sebastian Ruder. An overview of gradient descent optimization algorithms, 2017. 6

[47] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015. 2, 3

[48] Aleksandar Shtedritski, Christian Rupprecht, and Andrea Vedaldi. What does clip know about a red circle? visual prompt engineering for vlms, 2023. 2

[49] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 1, 2

[50] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023. 1

[51] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 2

[52] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. *Computer vision and pattern recognition*, 43(3):3485–3492, 2010. 4

[53] Hu Xu, Saining Xie, Xiaoqing Ellen Tan, Po-Yao Huang, Russell Howes, Vasu Sharma, Shang-Wen Li, Gargi Ghosh, Luke Zettlemoyer, and Christoph Feichtenhofer. Demystifying clip data, 2023. 5

[54] Yuhang Zang, Wei Li, Kaiyang Zhou, Chen Huang, and Chen Change Loy. Unified vision and language prompt learning. *arXiv preprint arXiv:2210.07225*, 2022. 3

[55] Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, Lucas Beyer, Olivier Bachem, Michael Tschannen, Marcin Michalski, Olivier Bousquet, Sylvain Gelly, and Neil Houlsby. A large-scale study of representation learning with the visual task adaptation benchmark, 2020. 3, 4, 5, 6, 7

[56] Yuexiang Zhai, Shengbang Tong, Xiao Li, Mu Cai, Qing Qu, Yong Jae Lee, and Yi Ma. Investigating the catastrophic forgetting in multimodal large language models, 2023. 6, 7

[57] Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. Vision-language models for vision tasks: A survey, 2023. 1

[58] Renrui Zhang, Jiaming Han, Chris Liu, Peng Gao, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, and Yu Qiao. Llama-adapter: Efficient fine-tuning of language models with zero-init attention, 2023. 1

[59] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023. 1

[60] 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. 2, 3, 4