SigLIP-Adapter

Tri Nguyen
Department of Computer Science
Indiana University
Bloomington, IN, USA
trihnguy@iu.edu

Carter Mullenix
Department of Data Science
Indiana University
Bloomington, IN, USA
cartmull@iu.edu

***Abstract*—Vision Language Models like CLIP (Contrastive Language-Image Pre-training) have gained significant attention in pattern recognition and computer vision. CLIP's capabilities for zero-shot and few-shot transfer learning are a hot topic in multimodal learning. To maximize accuracy and reduce the cost of retraining the model, recent research focuses on developing adapters to tune vision and text to take advantage of the extra information between image-text pairs. In this research, we propose using the SigLIP-Adapter to adapt SigLIP to zero-shot and few-shot predictions. This method will capture features using the textual features and visual features of images. In this paper, we propose two approaches: SigLIP-Adapter and Trainable SigLIP-Adapter, where the first one is training-free and the second one is trainable, depending on the number of epochs by adding two lightweight learnable components (i.e., a projector and a learnable latent space) to improve model performance.**

***Index Terms*—Vision–Language Model, Parameter-Efficient Fine-Tuning (PEFT), Image Captioning, SigLIP, Few-Shot Learning**

I. Introduction

Recent studies [1], [2] showed that incorporating text into visuals significantly increases the visual understanding of the model and hence improves its performance. SigLIP (Sigmoid loss for Language-Image Pre-training) [3], which also has a dual tower structure for the vision-language model that consists of a textual and visual encoder. Unlike standard contrastive learning with softmax normalization, sigmoid loss operates solely on image-text pairs and does not require a global view of the pairwise similarities for normalization. SigLIP performs better than CLIP when the batch size is less than 16k in ImageNet; therefore, we suggest using a smaller batch size, such as 2-4, for the dataset. In addition, we suggest using the smaller VLM model SmolVLM [4] to generate the description of the image, then summarize using T5 (Text-to-Transformer) [5].

II. Background and related work

A. Vision-Language Model

Modality allows humans to perceive the world through vision, touch, audio, and text/language. Vision and text are two main ways that humans can perceive the world, and there are many ongoing research interests from global researchers. Since the development of Transformer, it has paved the way for advanced research for machine learning and deep learning in general, which gave birth to Vision Transformer, which is mostly the backbone of popular computer vision models. Vision-Language model is the pretraining model that is mainly categorized into Image-Text Contrastive Learning and Pre-training with Generative Objectives.

Image-Text Contrastive Learning: Image-Text Contrastive Learning is the most popular method for the vision-language model. It utilizes contrastive learning to process input Image-Text to ensure that the image-text pairs with similar semantics are close in an embedding space, and image-text pairs with different semantics.

Contrastive Language–Image Pre-training (CLIP) [1] pairs a vision encoder and a text encoder and trains them on 400 million (image-text) pairs collected from the Internet and uses cosine similarity to match the image-text embeddings while minimizing mismatches in the same batch.

ALIGN [2] follows the same contrastive recipe but pushes scale even further: over 1.8B noisy Web pairs and larger backbones (EfficientNet-L2 for image encoder and a BERT-like text encoder). ALIGN uses softmax in favor of a single image-to-text contrastive loss, uses global batch normalization to stabilize massive batch sizes, and discards captions longer than 64 sub-words to reduce sequence length.

Pre-training with Generative Objectives: Pre-training with Generative Objectives with Vision-Language model by produce missing information for data rather than assign a class or a score: the text encoder reconstruct masked words while seeing the image, and the vision encoder predicts masked patches while reading captions. The model generate pixels or tokens rather than trying to match between the pairs hence it learn detailed, fine-grained across modalities. LLAVA [6] improves multimodal understanding by projecting CLIP image embeddings into the token space of a large language model; once the image is turned into a short sequence of pseudo-tokens, the frozen LLM can reason over visual content as if it were additional text.

B. Parameter-Efficient Fine-Tuning

Parameter-Efficient Fine-Tuning (PEFT) [7] is a fine-tuning method that freezes the parameters of the model's backbone and fine-tunes the incorporated learnable parameters on downstream datasets. PEFT is categorized into 2 types: Prompt Tuning and Adapter Tuning.

Prompt tuning: Prompt tuning adapts with downstream tasks by adding a learnable token in either input and learn only a tiny set of "soft" prompt vectors that are hidden layers of the model. Co-OP utilizes learnable tokens instead of design prompts as input for the text encoder, achieving great performance in a few-shot image classification tasks.

Adapter-Tuning: Adapter tuning will insert a small neural module (adapter block) inside each transformer layer and train those adapters. Clip-Adapter [8] uses a lightweight projection layer with two linear layers following the last layer of the vision encoder and text encoder to learn new features and is injected with pretrained features through residual networks. FLAVA [9] mask 40% of image's patches and 15% if caption's tokens, then use seperate MLP heads to reconstruc them then regenerate pieces of both modalities for cross-modal correspondence. Then model recovers parts of images and texts, and some models could recover the full image and text description from image-text pairs. Image description Enhanced CLIP-Adapter (IDEA) [10] uses a Multimodal adapter before fine-tuning, a captioning Llama [11] generates rich natural-language descriptions for each image; these synthetic sentences are fed through the text encoder and paired with the corresponding image features. The adapters are therefore trained with a contrastive loss on image–caption pairs that carry far more semantic detail than simple class names, yielding markedly better few-shot performance while still adding less than 2% extra parameters to CLIP.

III. Methods

A. SigLIP-adapter

Given a few-shot training set containing N classes and K examples per class, we enhance a frozen SigLIP backbone with an adapter that Leverages both visual features and auto-generated textual descriptions. Denote the image features of the few-shot set by $\mathbf{I}_{\text{train}} \in \mathbb{R}^{NK\times D}$ and their corresponding text features by $\mathbf{T}_{\text{train}} \in \mathbb{R}^{NK\times D}$, where D is the embedding dimension. For a test image $\mathbf{i}_{\text{test}}$ we compute:

$$\mathbf{i}_{\text{train}} = \text{VisionEncoder}(\text{Images}) \tag{1}$$

$$\mathbf{t}_{\text{train}} = \text{TextEncoder}(\text{Captions}) \tag{2}$$

Then we compute the visual and textual similarities between the test image and the few-shot examples:

$$\mathbf{s}^I = \mathbf{I}_{\text{train}}\,\mathbf{i}_{\text{test}}, \qquad \mathbf{s}^T = \mathbf{T}_{\text{train}}\,\mathbf{i}_{\text{test}} \tag{3}$$

where $\mathbf{s}^I, \mathbf{s}^T \in \mathbb{R}^{NK\times 1}$. We fuse them with a weighting hyper-parameter $\alpha \in [0, 1]$,

$$\mathbf{s} = (1-\alpha)\,\mathbf{s}^I + \alpha\,\mathbf{s}^T, \tag{4}$$

apply a sharpening activation $f(x) = \exp\big(\theta\,(x-1)\big)$ with scale $\theta > 0$, then aggregate the K examples of every class via

$$\text{FewShot}_c = g(\mathbf{s}) = \sum_{k=1}^{K} \mathbf{s}_{c,k}, \quad c = 1, \ldots, N. \quad (5)$$

Finally the overall logits combine few-shot and zero-shot knowledge:

$$\begin{aligned} \text{logits} \;&= \beta \underbrace{g\Big(f\big((1-\alpha)\,\text{Sim}_I + \alpha\,\text{Sim}_T\big)\Big)}_{\text{few-shot knowledge}} \\ &+ \underbrace{\mathbf{T}_{\text{class}}\;\mathbf{i}_{\text{test}}}_{\text{zero-shot knowledge}} \end{aligned} \quad (6)$$

where $\mathbf{T}_{\text{class}} \in \mathbb{R}^{N\times D}$ are the class-name text embeddings, while $\beta > 0$ trades off the two sources.

Algorithm 1 Pseudocode for SigLIP-Adapter

Input: test image $\mathbf{i}_{\text{test}}$
few-shot image feats $\mathbf{I}_{\text{train}} \in \mathbb{R}^{NK\times D}$
few-shot text feats $\mathbf{T}_{\text{train}} \in \mathbb{R}^{NK\times D}$
class text feats $\mathbf{T}_{\text{class}} \in \mathbb{R}^{N\times D}$
hyper-params α, β, θ

// Compute vision similarity
$\mathbf{s}^I \leftarrow \mathbf{I}_{\text{train}}\;\mathbf{i}_{\text{test}}$
// Compute text similarity
$\mathbf{s}^T \leftarrow \mathbf{T}_{\text{train}}\;\mathbf{i}_{\text{test}}$
// Compute multimodal similarity
$\mathbf{s} \leftarrow (1-\alpha)\,\mathbf{s}^I + \alpha\,\mathbf{s}^T$
// Compute activation function
$\mathbf{s} \leftarrow \exp\big(\theta\,(\mathbf{s}-1)\big)$
// Reshape to $N \times K$ and sum over K
fewShot $\leftarrow$ reshape$(\mathbf{s}, N, K)$
// Aggregate similarity to form few-shot
fewShot$\leftarrow$ sum(fewShot, dim $= 1$)
// Compute zero-shot knowlegdge
zeroShot $\leftarrow \mathbf{T}_{\text{class}}\;\mathbf{i}_{\text{test}}$
*// Compute logits N*1*
return $\ell = \beta$ fewShot + zeroShot

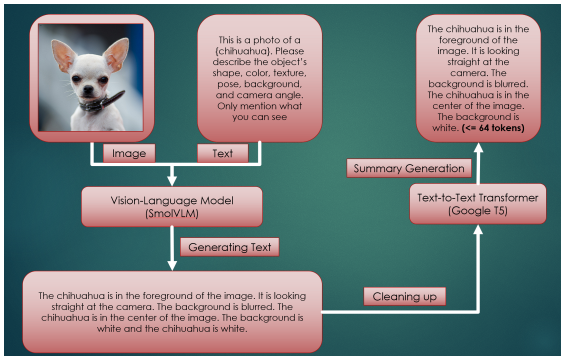

Fig. 1. Pipeline of generating image descriptions.

B. Image Description Generation

Current visual datasets generally lack textual descriptions for corresponding images and manually labeling these datasets is labor-intensive. Hence, we utilize SmolVLM [4] currently have 3 different sizes 2B, 500M, and 256M and we instruct them using SmoVLM to ask them about the prompt for generate description of the image. By customizing template for textual prompt for each dataset to generate descriptions. Then we clean the data by to reduce noisy text (Formatting from multimodal chat prompt, special symbol, etc.) Then we utilize using the Text-to-Text Transformer (T5) [5] to summarize text less than 64 tokens, which is maximum length of the SigLIP text encoder.

Fig. 2. Good examples of image descriptions of SmolVLM 256M for Oxford Pets, Caltech-101 and Food-101 datasets.

IV. Results

We trained, validated, and tested the model on the Oxford-IIIT Pet [12], Caltech 101 [13], Food-

101 [14], EuroSAT [15], and Describable Textures Dataset (DTD) [16] datasets, using the standard splits of 70% for training, 20% for validation, and 10% for testing.

For one-shot experiments, we sampled one image per class ten times with different random seeds and report the average top-1 accuracy. All methods use identical data preprocessing and SigLIP model weights; only the adapter components differ.

TABLE I
SIGLIP ADAPTER DATASET PERFORMANCE

Dataset	W/o SmolVLM+T5 **Zero-Shot**	With SmolVLM+T5 **Zero-Shot**	SigLIP Adapter **1-Shot**
Oxford Pets	63.59%	65.59%	69.39%
Caltech101	49.24%	50.66%	55.36%
Food-101	18.15%	20.45%	23.35%
EuroSat	19.19%	18.89%	49.19%
DTD	35.46%	35.28%	43.79%

In the baseline zero-shot setting without SmolVLM+T5, the models achieved 63.59% on Oxford Pets, 49.24% on Caltech101, 18.15% on Food-101, 19.19% on EuroSat, and 35.46% on DTD.

Incorporating SmolVLM+T5 led to modest improvements across most datasets. Performance on Oxford Pets increased to 65.59%, Caltech101 to 50.66%, and Food-101 to 20.45%. Minor decreases were observed for EuroSat, which declined slightly to 18.89%, and DTD, which decreased to 35.28%.

When applying the SigLIP Adapter with 1-shot training, substantial improvements were observed across all datasets. Oxford Pets performance rose to 69.39%, Caltech101 improved to 55.36%, Food-101 increased to 23.35%, EuroSat saw a significant jump to 49.19%, and DTD reached 43.79%.

These results demonstrate that the SigLIP Adapter, even under limited supervision, can provide considerable performance gains, particularly for datasets such as EuroSat and DTD, where the baseline zero-shot results were relatively low.

V. DISCUSSION

The results demonstrate the effectiveness of introducing adapter modules for improving performance under both zero-shot and few-shot conditions. The inclusion of SmolVLM+T5 yielded slight improvements in zero-shot performance for most datasets, suggesting that enhancing the textual features with a small language model can modestly benefit vision-language alignment. However, the limited size of the SmolVLM+T5 models may have constrained the extent of these gains, particularly for datasets such as EuroSat and DTD, which differ significantly from typical object-centric datasets.

The introduction of the SigLIP Adapter with 1-shot training produced much larger improvements across all datasets. Notably, datasets composed of textures and satellite imagery, such as EuroSat and DTD, benefited disproportionately. This suggests that the adapter was particularly effective at compensating for domain shifts and low semantic alignment between the visual features and textual prompts in the original SigLIP model.

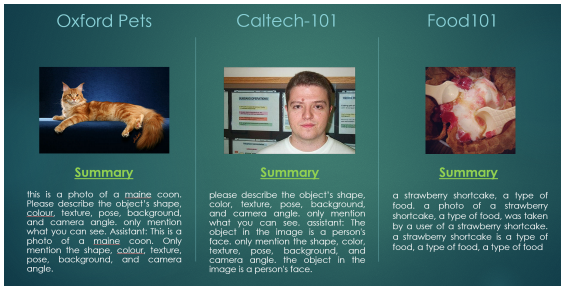

Fig. 3. Bad examples of image descriptions of SmolVLM 256M for Oxford Pets, Caltech-101, and Food-101 datasets.

Qualitative analysis of the generated summaries further supports these findings. As shown in the examples, the outputs for Oxford Pets and Caltech-101 contain noticeable prompt leakage, with significant portions of the original prompt text being repeated verbatim in the summary. Food-101, while less affected by prompt leakage, exhibited semantic repetition and collapse, redundantly emphasizing the object category (”a type of food”) without adding a meaningful new description. These issues suggest that the small SmolVLM+T5 model struggled to generate fully coherent and contextually rich summaries, particularly for visually complex or ambiguous datasets.

The relatively larger gains in EuroSat and DTD compared to object-centric datasets such as Oxford Pets and Food-101 indicate that the adapter modules can significantly enhance feature representations in non-standard vision tasks. These results emphasize the value of lightweight adaptation techniques in settings where full fine-tuning may be infeasible or where domain-specific data is limited.

Overall, the findings support the hypothesis that adapter tuning, even with minimal supervision, can substantially close the gap between zero-shot and one-shot performance while maintaining efficiency in model size and training requirements. These qualitative shortcomings highlight an important direction for future research: developing stronger multimodal adapters or enhanced prompt-tuning strategies that can better mitigate prompt contamination and semantic drift in low-data regimes.

VI. Conclusion

This work investigated the effectiveness of adapter tuning for improving zero-shot and few-shot performance in multimodal vision language models. Experiments were conducted on five diverse datasets: Oxford-IIIT Pet, Caltech 101, Food 101, EuroSAT, and DTD. While the incorporation of a small SmolVLM and T5 model yielded modest improvements in zero-shot settings, the application of the SigLIP Adapter with one-shot training resulted in substantial performance gains across all datasets.

Qualitative analysis of the generated image-text summaries revealed limitations associated with prompt leakage and semantic drift, particularly in datasets characterized by less visually distinct categories. These observations suggest that the use of small language models may constrain the richness and coherence of generated textual features. Although the utilization of larger SmolVLM and T5 models could potentially alleviate these issues, computational resource constraints precluded their exploration in this study.

Overall, the findings underscore the efficacy of lightweight adapter modules in enhancing multimodal representations under limited supervision. Future work will focus on scaling to larger language models, improving prompt generation strategies, and developing more advanced adapter architectures to further strengthen generalization across both object-centric and domain-shifted datasets.

VII. Acknowledgments

The authors would like to thank Kaggle for providing free cloud-based computational resources, which made the experiments and model training possible.

References

[1] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," *arXiv preprint arXiv:2103.00020*, 2021.

[2] C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. V. Le, Y. Sung, Z. Li, and T. Duerig, "Scaling up visual and vision-language representation learning with noisy text supervision," 2021. [Online]. Available: https://arxiv.org/abs/2102.05918

[3] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer, "Sigmoid loss for language image pre-training," 2023. [Online]. Available: https://arxiv.org/abs/2303.15343

[4] Hugging Face Research Team, "Smolvlm: Smaller, faster, cheaper vision-language models," 2024. [Online]. Available: https://huggingface.co/blog/smolvlm

[5] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," 2023. [Online]. Available: https://arxiv.org/abs/1910.10683

[6] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," 2023. [Online]. Available: https://arxiv.org/abs/2304.08485

[7] N. Ding, Y. Qin, G. Yang, F. Wei, Z. Yang, Y. Su, S. Hu, Y. Chen, C.-M. Chan, W. Chen, J. Yi, W. Zhao, X. Wang, Z. Liu, H.-T. Zheng, J. Chen, Y. Liu, J. Tang, J. Li, and M. Sun, "Delta tuning: A comprehensive study of parameter efficient methods for pre-trained language models," 2022. [Online]. Available: https://arxiv.org/abs/2203.06904

[8] P. Gao, S. Geng, R. Zhang, T. Ma, R. Fang, Y. Zhang, H. Li, and Y. Qiao, "Clip-adapter: Better vision-language models with feature adapters," 2025. [Online]. Available: https://arxiv.org/abs/2110.04544

[9] A. Singh, R. Hu, V. Goswami, G. Couairon, W. Galuba, M. Rohrbach, and D. Kiela, "Flava: A foundational language and vision alignment model," 2022. [Online]. Available: https://arxiv.org/abs/2112.04482

[10] Z. Ye, F. Jiang, Q. Wang, K. Huang, and J. Huang, "Idea: Image description enhanced clip-adapter," 2025. [Online]. Available: https://arxiv.org/abs/2501.08816

[11] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "Llama: Open and efficient foundation language models," arXiv preprint arXiv:2302.13971, 2023.

[12] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "The oxford-iiit pet dataset," 2012. [Online]. Available: https://www.robots.ox.ac.uk/ vgg/data/pets/

[13] L. Fei-Fei, R. Fergus, and P. Perona, "Caltech 101 dataset," 2004. [Online]. Available: http://www.vision.caltech.edu/Image_Datasets/Caltech101/

[14] L. Bossard, M. Guillaumin, and L. Van Gool, "Food-101 – mining discriminative components with random forests," 2014. [Online]. Available: https://data.vision.ee.ethz.ch/cvl/datasets_extra/food-101/

[15] P. Helber, B. Bischke, A. Dengel, and D. Borth, "Eurosat dataset," 2023. [Online]. Available: https://www.kaggle.com/datasets/apollo2506/eurosat-dataset

[16] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi, "Describable textures dataset (dtd)," 2022. [Online]. Available: https://www.kaggle.com/datasets/jmexpert/describable-textures-dataset-dtd