Learning Prompts for Transfer Learning along with Test-time Adaption

Giovanni Scialla (239181), Mattia Franzin (239930), Adnan Irshad (241763), Hira Afzal (241351)

***Abstract*—Large pre-trained vision-language models like CLIP have shown great potential in learning representations that are transferable across a wide range of downstream tasks. Inspired by the Natural Language Processing (NLP) literature, recent CLIP adaptation approaches learn prompts as the textual inputs to fine-tune CLIP for downstream tasks.**

I. Introduction

In this work, we show that a major challenge for deploying large vision-language models in practice is prompt engineering, which requires domain expertise and is extremely time-consuming. We propose a series of methods drawn from the literature that address the problem of adapting vision-language models for downstream image recognition by learning effective prompts.

First part of our project is centred around exploring approaches for learning to Prompt's context words with learnable vectors while maintaining the pre-trained parameters of the model unchanged. While these methods have demonstrated significant advancements, they present some limitations, particularly in their adaptability when encountering new or unseen data sets and tailored to the training data. This leads us to the second focus of our project: "Test-Time Adaption (TTA) for prompts tuning". In second part, we delve into the mechanisms of TTA, exploring how it allows for real-time adjustments and fine-tuning of prompts based on the test data. This adaptive approach significantly enhances the model's flexibility and accuracy, especially in zero-shot and few-shot learning scenarios, where the model must perform well on data distributions it was not explicitly trained on.

II. Large Pre-trained Vision-Language Model

Vision-language models such as *CLIP* [6] have recently demonstrated great potential in learning generic visual representations and allowing zero-shot transfer to a variety of downstream classification tasks via prompting. *CLIP* consists of two encoders. The image encoder aims to map high-dimensional images into a low-dimensional embedding space while the text encoder aims to generate text representations from natural language. The close text and image representations are in the embedding space, the more related they are. In order to achieve this relationship a contrastive loss has been used during training to maximize the cosine similarity for matched pairs while minimizing the cosine similarity for all other unmatched pairs.
The prediction probability is then computed as:

$$p(y=i|x) = \frac{exp(cos(w_i, f)/t)}{\sum_{j=1}^{K} exp(cos(w_j, f)/t)} \tag{1}$$

III. Prompt Learning Methods

A. Context Optimization (CoOp) [9]

In this work, they propose a simple approach called *Context Optimization (CoOp)* to automate prompt engineering. Concretely, *CoOp* models a prompt's context words with learnable vectors. Two implementations are provided to handle tasks of different natures: one is based on unified context, which shares the same context with all classes; while the other is based on class-specific context, which learns a specific set of context tokens for each class. By forwarding a prompt t to the text encoder $g(.)$, the prediction probability is computed as

$$p(y=i|x) = \frac{exp(cos(g(t_i), f)/t)}{\sum_{j=1}^{K} exp(cos(g(t_j), f)/t)} \tag{2}$$

The gradients can be back-propagated all the way through the text encoder $g(.)$, making use of the rich knowledge encoded in the parameters to optimize the context. The design of continuous representations also allows full exploration in the word embedding space.

B. Conditional Context Optimization (CoCoOp)[10]

In this work, they identify a critical problem of *CoOp*: the learned context is not generalizable to wider unseen classes within the same dataset. To address the problem, they propose *Conditional Context Optimization (CoCoOp)*, which extends *CoOp* by further learning a lightweight neural network, called *Meta-Net*, to generate for each image an input-conditional token vector, which is combined with the context vectors, in order to obtain dynamic prompts that are less sensitive to class shift. Let $h_\theta(.)$ denote the Meta-Net parameterized by θ, each context token is now obtained by $v_m(x) = v_m + h_\theta(x)$
The prompt for the i-th class is thus conditioned on the input. The prediction probability is computed as

$$p(y=i|x) = \frac{exp(cos(g(t_i(x), f)/t)}{\sum_{j=1}^{K} exp(cos(g(t_j(x), f)/t)} \tag{3}$$

During training, we update the context vectors together with the Meta-Net's parameters θ.

C. Multi-modal Prompt Learning (MaPLe) [5]

In this work they propose *Multi-modal Prompt Learning (MaPLe)* for both vision and language branches to improve alignment between the vision and language representations. their design promotes strong coupling between the vision-language prompts to ensure mutual synergy and discourages learning independent uni-modal solutions. They argue that due to the multi-modal nature of CLIP, where a text and image

encoder co-exist and both contribute towards properly aligning the V-L modalities, any prompting technique should adapt the model completely and therefore, learning prompts only for the text encoder in CLIP is not sufficient to model the adaptations needed for the image encoder. Concretely, *MaPLe* proposes a joint prompting approach where the context prompts are learned in both vision and language branches. The idea is to append learnable context tokens in the language branch and explicitly condition the vision prompts on the language prompts via a *coupling function* to establish interaction between them. To learn hierarchical contextual representations, they introduce *deep prompting* in both branches through separate learnable context prompts across different transformer blocks (up to a certain depth J).

D. Prompt Pre-Training with Twenty-Thousand Classes for Open-Vocabulary Visual Recognition (POMP) [8]

In this work, they propose an alternative for prompt learning, trying to improve *CoOp* and *CoCoOp* by reducing their **GPU memory** usage and **training time** (especially *CoCoOp*). To do that, they avoid comparing each selected class with all the negative others during training: so instead of picking **N** classes, just **K** are sampled, with $K \ll N$. This process has been called **local contrast**, because after picking the ground-truth (*positive*) class, it is compared with a random subset of *negative* classes, with all of them sharing a uniform probability distribution to be picked. Now, since the comparison involves fewer classes, the gradient is diminished to $\frac{K}{N}$, resulting in a less negative gradient. To fix that, a **local correction** term *m* is added to the logits of negative classes. The final prediction probability becomes:

$$\tilde{P}(y|\boldsymbol{x};\boldsymbol{\theta}) = \frac{exp(\boldsymbol{x}^\intercal \boldsymbol{w}_y^{(\boldsymbol{\theta})}/\tau)}{exp(\mathbf{x}^\intercal \boldsymbol{w}_y^{(\boldsymbol{\theta})}/\tau) + \sum_{i\sim N} exp(\mathbf{x}^\intercal \boldsymbol{w}_i^{(\boldsymbol{\theta})}/\tau + m)} \tag{4}$$

with **m** being:

$$m = -\log((K-1)/(N-1)) \tag{5}$$

encouraging the positive logits to be larger than negative ones, resulting in a more stringent decision boundary.

The setup is similar to *CoOp*, and in order to improve **generalization** using few classes per training step, the total number of classes has been raised to 21 thousands, showing better GPU consumption and less training time (per step).

E. Visual-Language Prompt Tuning with Knowledge-guided Context Optimization [3]

In this work, they kept the *CoOp* model, but tried another approach to improve **unseen classes accuracy**. The reason for that is the fact *CoOp* performs even worse than *zero-shot CLIP* after it has been fine-tuned: the reason is obvious, the model gets specialized for **seen classes**, while forgetting about **general knowledge**. They propose to add another loss, alongside the *cross-entropy* one. Since they want to retain general knowledge, the way-to-go is by not using the learnable prompts: basically the new loss is a measure of the embedding distance between the **learned prompts** and the **hand-crafted prompts** (*"a photo of a [class]"*). The loss is computed as follows:

$$\mathcal{L}_{kg} = \frac{1}{N_c}\sum_{i=1}^{N_c} \|\boldsymbol{w}_i - \boldsymbol{w}_i^{clip}\|_2^2 \tag{6}$$

with $\|\cdot\|$ being the euclidean distance, and N_c the seen classes. The final loss is just a combination of the two:

$$\mathcal{L} = \mathcal{L}_{ce} + \lambda \mathcal{L}_{kg} \tag{7}$$

where λ is used balance the effect of $\mathcal{L}_{kg}$ in the final objective. $\lambda = 8.0$ seems to work good.

F. Prompt-aligned Gradient for Prompt Tuning [1]

In this work, as the previous one, they kept the *CoOp* model as a starting point, and added a custom loss: the difference here is how the two are combined. The other loss involved, in addition to the cross-entropy loss $\mathcal{L}_{ce}$, also called **domain-specific knowledge**, a **general knowledge** loss is computed based on the Kullback-Leibler (KL) divergence between $p(\boldsymbol{t}_i|\boldsymbol{x})$ (learned prompt) and the zero-shot CLIP prediction $p_{zs}(\boldsymbol{w}_i|\boldsymbol{x})$:

$$\mathcal{L}_{kl}(\boldsymbol{v}) = -\sum_i p_{zs}(\boldsymbol{w}_i|\boldsymbol{x}) \log \frac{p(\boldsymbol{t}_i|\boldsymbol{x})}{p_{zs}(\boldsymbol{w}_i|\boldsymbol{x})} \tag{8}$$

Then, both the gradients are computed for both losses ($\boldsymbol{G}_g = \nabla\mathcal{L}_{kl}(\boldsymbol{v}), \boldsymbol{G}_d = \nabla\mathcal{L}_{ce}(\boldsymbol{v})$). Based on the **cross-product** between the two gradients, the final gradient $\boldsymbol{G}_{prograd}$ can differ:

$$\boldsymbol{G}_{pg} = \begin{cases} \boldsymbol{G}_d, & \text{if } \boldsymbol{G}_d \cdot \boldsymbol{G_g} \geq 0 \\ \boldsymbol{G}_d - \lambda \cdot \frac{\boldsymbol{G}_d \cdot \boldsymbol{G}_g}{\|\boldsymbol{G}_g\|^2}\boldsymbol{G}_g, & \text{otherwise.} \end{cases} \tag{9}$$

Basically, if the **domain-specific** gradient is in conflict with the **general** one (their angle is greater than 90°), the new gradient is the projection of G_d on the orthogonal direction of G_g. λ controls the strength of general knowledge guidance, with $\lambda = 1$ meaning *orthogonal projection*, and $\lambda = 0$ meaning *CoOp*.

IV. Test-time Adaption

A. Test-Time Prompts Tunining (TPT) [7]

Despite being effective, above-mentioned work requires access to downstream annotated data and restricts the ZL knowledge transfer of foundation models especiall CoOp. CoCoOp attenuates this, but it ends up slowing the training by adding the Meta-Net and performance drops in the base classes.

TPT addresses these challenges by tuning adaptive prompts on the fly with a single test to adapt prompts dynamically to new, unseen data, enhancing the model's flexibility. TPT optimizes the prompt **p** by minimizing the entropy across different augmented views of a test sample $\mathbf{X_{test}}$, ensuring consistent model predictions. In general, TPT is defined as:

$$p^* = argmin_p L(F, p, X_{test}) \tag{10}$$

Where **F** is defined as CLIP model with $\mathbf{E_{text}}$ and $\mathbf{E_{visual}}$ encoders for text inputs and augmented views of a single test sample $Fp(X) = sim(Etext(p; Y), Evisual(X))$

Image Classification: TPT enhance CLIP model in image classification tasks by tuning prompts. Since TPT uses unlabelled target data, they used unsupervised loss to compute average probability distribution as the mean of class probabilities over 'N' augmented views **A** of $\mathbf{X_{test}}$

$$p^* = argmin_p - \sum_{i=1}^{K} P_p(y_i|X_{test}) log P_p(y_i|X_{test}) \tag{11}$$

Where $P_p(y|A_i(X_{test})) = 1f\frac{1}{N}\sum_{i=1}^{N} P_p(y_i|A_i(X_{test}))$
TPT introduces a novel component called 'confidence selection' to improve prompts focusing on high-confidence views, enhancing the overall prediction accuracy. The final prediction probability becomes:

$$P_p(y|X_{test}) = 1f\frac{1}{N}\sum_{i=1}^{N}[H(pi) \leq T] P_p(y|A_i(X_{test})) \tag{12}$$

B. Diverse Data Augmentation with Diffusions for Effective Test-time Prompt Tuning (DiffTPT) [2]

TPT relies heavily on the generation of multiple augmented views of the test sample which introduces the risk of including 'noisy' augmentations. To address these issues, they propose DiffTPT which leverages diffusion models specifically **Stable Diffusion-V2** to generate diverse augmented views $\mathbf{D}_n$ from a single test image $\mathbf{X_{test}}$, first extract its latent features $\mathbf{z}_0$ from the pre-trained CLIP encoder $\mathbf{f(X_{test})}$ and then use the stable diffusion as the decoder to generate augmented images:

$$\mathbf{D}_n(\mathbf{X_{test}}) = G(\mathbf{f(X_{test})}, n_n) \tag{13}$$

Taking the augmented images $\mathbf{D}_n(\mathbf{X_{test}})$ into account, TPT equation (12) becomes:

$$P_v = 1f\frac{1}{H^N}\sum_{i=1}^{N}[H(pi) \leq T] P_n(y|\mathbf{D}_n(\mathbf{X_{test}})) \tag{14}$$

To overcome the shortcomings of entropy-based confidence selection, DiffTPT introduces a cosine similarity-based filtration technique. This technique selects the generated data that has a higher similarity to the single test sample, ensuring better prediction fidelity. By incorporating these innovative techniques, DiffTPT has been shown to improve zero-shot accuracy significantly on test datasets featuring distribution shifts and unseen categories demonstrating an average improvement of 5.13% in zero-shot accuracy compared to state-of-the-art TPT methods.

C. Align Your Prompts: Test-Time Prompting with Distribution Alignment for Zero-Shot Generalization (PromptAlign) [4]

A key limitation in TPT methods is their susceptibility to performance degradation due to distribution shifts. The PromptAlign, addresses by introducing distribution alignment strategy which aligns the statistics of OOD test samples with the source data, effectively minimizing the feature distribution shift at test time.

$$L(final) = L(entropy) + L(align) \tag{15}$$

By utilizing offline computed source data statistics, the method harmonizes token distribution alignment with entropy minimization, significantly improving ZS top-1 accuracy and generalization across diverse datasets.

V. Experiments and Results

Here we report the most salient results in adapting CLIP-like vision-language models using learnable prompts.
Datasets: 11 publicly available image classification datasets has been used in order to have a comprehensive benchmark, which covers a diverse set of vision tasks.

	Base Classes	Novel Classes	H
Zero-shot CLIP	69.34	74.22	71.70
CLIP + CoOp	**82.69**	63.22	71.66
CLIP + CoCoOp	80.47	71.69	75.83
CLIP + MaPLe	82.28	**75.14**	**78.55**
CLIP + POMP	-	-	-
CLIP + KgCoOp	80.73	73.60	77.00
CLIP + ProGrad*	73.29	65.96	69.06

TABLE I: Results of image classification on base classes seen during training and on OOD novel classes, the Harmonic Mean (H) has been used to highlight the generalization trade-off (average over 11 dataset). POMP has no Base/New/H comparison, it has just been cross-dataset evaluation as showed in Figure 1. Prograd comparison in the paper shows lower scores wrt. above CoOp and CoCoOp

	Target (cross-dataset)										
	Caltech101	*OxfordPets*	*StanfordCars*	*Flowers102*	*Food101*	*Aircraft*	*SUN397*	*DTD*	*EuroSAT*	*UCF101*	***Average***
hard prompt	93.3	88.2	65.6	67.4	85.3	23.7	62.6	44.3	42.0	65.1	63.7
CoOp [65]	93.7	89.1	64.5	68.7	85.3	18.5	64.2	41.9	46.4	66.6	63.9
CoCoOp [66]	94.4	90.1	65.3	71.9	86.1	22.9	67.4	45.7	45.4	68.2	65.7
LASP [5]	94.5	89.4	64.8	70.5	86.3	23.0	67.0	45.5	48.3	68.2	65.8
VPT [12]	93.7	**90.6**	65.0	70.9	86.3	24.9	67.5	46.1	45.9	68.7	66.0
MaPLe [29]	93.5	90.5	65.6	72.2	86.2	24.7	67.0	**46.5**	48.1	**68.7**	66.3
POMP (Ours)	**95.0**	89.5	**66.8**	**72.4**	**86.3**	**25.6**	**67.7**	46.2	**52.1**	68.5	**67.0**

Fig. 1: Cross-dataset evaluation

VI. Conclusions and Future Works

As reported on section V, prompt-learning techniques with **few-shot** learning improved **Zero-Shot CLIP**, while keeping **competitive results** in terms of generalization. Next, we will be trying to merge together the techniques explored for prompt learning along with test-time adaption and see if they actually could improve even better the generalization on OOD data, maybe, leading to greater results and deepening the exploration about this blended approach.

REFERENCES

[1] Yucheng Han Yue Wu Hanwang Zhang Beier Zhu, Yulei Niu. Prompt-aligned gradient for prompt tuning, 2023.

[2] Chun-Mei Feng, Kai Yu, Yong Liu, Salman Khan, and Wangmeng Zuo. Diverse data augmentation with diffusions for effective test-time prompt tuning, 2023.

[3] Changsheng Xu Hantao Yao, Rui Zhang. Visual-language prompt tuning with knowledge-guided context optimization, 2023.

[4] Jameel Hassan, Hanan Gani, Noor Hussein, Muhammad Uzair Khattak, Muzammal Naseer, Fahad Shahbaz Khan, and Salman Khan. Align your prompts: Test-time prompting with distribution alignment for zero-shot generalization, 2023.

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

[6] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020, 2021.

[7] Manli Shu, Weili Nie, De-An Huang, Zhiding Yu, Tom Goldstein, Anima Anandkumar, and Chaowei Xiao. Test-time prompt tuning for zero-shot generalization in vision-language models, 2022.

[8] Yi Zhu Shuai Zhang Shuai Zheng Mu Li Alex Smola Xu Sun Shuhuai Ren, Aston Zhang. Prompt pre-training with twenty-thousand classes for open-vocabulary visual recognition, 2023.

[9] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *CoRR*, abs/2109.01134, 2021.

[10] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models, 2022.