PRE: Vision-Language Prompt Learning with Reparameterization Encoder

Thi Minh Anh Pham minhanh.pham151@gmail.com
Department of Computer Science
Queen Mary University of London

Duc-An Nguyen an.nguyen@cs.ox.ac.uk
Department of Computer Science
University of Oxford

Cephas Svosve cephas.svosve@maths.ox.ac.uk
Department of Mathematics
University of Oxford

Vasileios Argyriou vasileios.argyriou@kingston.ac.uk
School of Computer Science and Mathematics
Kingston University London

Georgios Tzimiropoulos g.tzimiropoulos@qmul.ac.uk
Department of Computer Science
Queen Mary University of London

Abstract

Large vision-language foundation models such as CLIP have demonstrated great potential in zero-shot transferability to downstream tasks. However, manual prompt engineering is the major challenge for deploying such models in practice since it requires domain expertise and extreme time. To avoid non-trivial prompt engineering, recent work Context Optimization (CoOp) introduced the concept of prompt learning to the vision domain using learnable textual tokens. While CoOp can achieve substantial improvements over manual prompts, its learned context is worse generalizable to wider unseen classes within the same dataset. In this work, we present Prompt Learning with Reparameterization Encoder (PRE) - a simple and efficient method that enhances the generalization ability of the learnable prompt to unseen classes in practical domains. Instead of directly optimizing the prompts, PRE employs a prompt encoder to reparameterize the input prompt embeddings, enhancing the exploration of domain-specific knowledge from few-shot data. Experiments and extensive ablation studies on 8 benchmarks demonstrate that our approach is an efficient method for prompt learning in vision-language foundation models. Specifically, PRE achieves a notable enhancement of 5.60% in average accuracy on *New* classes and 3% in *Harmonic mean* compared to CoOp in the 16-shot setting.

Keywords: prompt learning; domain specific data; few-shot learning; vision-language foundation models; CLIP.

© *The 12th International Conference on Learning Representations*, Vienna, Austria. DMLR@ICLR 2024. .

Table 1: Compared to existing methods on 8 datasets with 16-shot settings, PRE obtains a higher performance within good training time.

Methods	**Prompts**	**Accuracy (%)**			**Training Time**
		Base	**New**	**H**	
CLIP	hand-crafted	68.81	74.43	71.42	-
CoOp	textual	83.32	66.92	73.34	6ms/image
CoCoOp	textual+visual	80.89	70.99	74.47	160ms/image
ProGrad	textual	82.96	70.30	75.58	22ms/image
PRE	textual	82.14	**71.90**	**76.22**	6.2ms/image

1 Introduction

In recent years, vision-language foundation models (VLMs) have brought new light on leveraging natural language supervision in visual recognition systems, enabling a wide exploration of open-set visual concepts Li et al. (2021); Jia et al. (2021); Yao et al. (2021a). Notably, VLMs with contrastive learning, exemplified by Contrastive Language-Image Pretraining (CLIP) models Radford et al. (2021), have gained diverse visual concepts and rich cross-modal representations that hold great potential to be transferred to various tasks. As these VLMs evolve, a pivotal question arises: How can the valuable knowledge from pretraining be effectively adapted to downstream tasks with domain-specific data?

In the initial study Radford et al. (2021), prompt engineering has been utilized to add more meaningful context in textual class descriptions by using a set of manually selected prompts for the given task. For example, on the Oxford Pets dataset, employing a tuning prompt such as ”a photo of a class, a type of pet.” helps improve the accuracy performance Yao et al. (2021b) Jin et al. (2021). However, prompt engineering relies on trial and error, demanding significant human effort for word tuning - a slight change in wording could make a huge difference in performance, and does not guarantee the optimal prompts.

Following research in NLP Li and Liang (2021), many recent works, beginning with Context Optimization(CoOp) Zhou et al. (2022b) introduced the concept of prompt learning to replace the manual prompts with a sequence of prompt tokens. Then, these prompt tokens are learned by minimizing the distance between the visual features and prompt-based text features using a few training examples to provide more flexibility in text encoding. Despite significant improvements over CLIP, a noticeable problem with CoOp-based methods is the poor generalization to the unseen (*New*) classes within the same practical domain.

The soft prompt optimization in CoOp attempts to learn separate prompt tokens solely through the pre-trained knowledge embedded in the fixed parameters of CLIP's text encoder. Because of the fixed over-parameterization of CLIP and lack of training examples, naive prompt tuning would lead to overfitting the seen (*Base*) classes on specific datasets. Intuitively, we believe the values of prompt embeddings should be dependent on each other rather than independent, and their interdependence varies according to different domain-specific data. We need a trainable mechanism that jointly processes prompt embeddings. It could flexibly capture domain-relevant dependencies within prompt tokens beyond the constraints of the frozen text encoder. Incorporating these associate-learned prompt tokens into the text encoder would make the optimization easier to find a more contextually generalizable prompt specific to the particular dataset. Based on our hypothesis, we introduce Prompt Learning with a Reparameterization Encoder (PRE). Our main contributions are as follows:

- Instead of directly learning the prompts, PRE reparameterizes the original prompt embeddings utilizing a prompt encoder before feeding them into the Text Encoder. Our prompt encoder incorporates a Bidirectional long short-term memory network (BiLSTM). This BiLSTM not only serves as a parameterizing network but also exploits the domain-specific long-range dependencies in the prompt sequence. We further adopt a residual connection for the prompt encoder to avoid forgetting the original knowledge encoded by the pre-trained CLIP.
- We perform extensive ablation studies of PRE on eight classification datasets to analyze its characteristics. Specifically, several network architectures have been implemented in the prompt encoder, thus offering distinct benefits for different recognition tasks. The code is available at Github.

Main results: We assess the performance of PRE through extensive experimentation on the base-to-new generalization setting across eight image classification datasets. The evaluation results in Table 1 highlight the efficiency and effectiveness of PRE. Our method demonstrated substantial accuracy improvement for the *New* class compared to other methods while maintaining good training time.

2 Related Work

2.1 Vision-language Foundation Models

The field of Vision-Language Models has experienced significant progress in forming robust representations that can be effectively transferred to various downstream tasks Kamath et al. (2021); Li et al. (2019); Hong et al. (2020); Kim et al. (2021); Bao et al. (2022). A typical VLM comprises three components: an image encoder, a text encoder, and a loss function. Recent works such as CLIP Radford et al. (2021), ALIGN Jia et al. (2021), and DeCLIP Li et al. (2021) bridge the vision-language modalities by learning text encoder and image encoder jointly with a contrastive loss, using large image-caption pairs datasets. Notably, CLIP showcases an impressive ability for zero-shot image recognition. Similar to the previous work CoOp and CoCoOp, we apply the pre-trained CLIP for knowledge transfer, aiming to facilitate the adaptation of such models in downstream datasets.

2.2 Prompt Learning

Prompt learning has emerged as a novel paradigm in NLP for exploiting pre-trained language models, gradually replacing the traditional fine-tuning transfer approach. The main idea of prompt learning is to formulate various NLP tasks as masked language modeling problems, adopting different templates (or prompts). In the context of Vision-Language Models like CLIP, human-crafted prompts based on class names are utilized to enable zero-shot visual recognition. Context Optimization (CoOp) Zhou et al. (2022b) extended soft prompt optimization to VLMs, where a set of prompts is learned and used as input to the text encoder alongside the class name. However, CoOp suffers from weak generalization, as the learned prompts tend not to capture domain-specific data well and perform poorly in novel classes. To address CoOp's generalization limitations, later work Conditional Context Optimization (CoCoOp) Zhou et al. (2022a) proposes a dynamic approach. It employs a small neural network to produce a visual feature from each input image, which is then combined with the learned prompts, making them input-specific. However, CoCoOp's Meta-Net demands additional computation, which can be limiting, especially when dealing with large datasets or resource-constrained environments.

Instead of directly learning the prompts like CoOp, PRE first passes the prompt tokens through a trainable encoder with a residual connection, enabling a flexible combination of the original prompt embeddings and embedding projections. This approach leads PRE to outperform CoOp, particularly in exploiting domain-specific data to handle unseen classes.

3 Methodolgy

3.1 Preliminaries

Our proposed method builds upon CLIP Radford et al. (2021), which is a well-known VLMl trained on an extensive dataset of 400 million image-text pairs. CLIP contains a visual encoder (ϕ) responsible for mapping images to visual embeddings and a textual encoder (θ) used for embedding corresponding textual information. During pretraining, CLIP LIP trains the image and text encoders by contrastive loss, which tries to maximize the similarity between matching pairs while minimizing the similarity with mismatching pairs.

Prompt Engineering: For downstream recognition tasks, CLIP performs the zero-shot inference by employing hand-engineered prompts to generate textual class embeddings. Given set $\mathcal{V}$ of $\mathcal{C}$ class names, the class descriptions $\{\mathbf{t}_c\}_{c=1}^{C}$ are generated with the manually designed prompt template, such as "`a class of a {class_name}`". Then the class descriptions are passed through the text encoder $\theta(\cdot)$ to compute the class-specific textual embeddings (weight): $\mathbf{w}_i^C = \theta(\mathbf{t}_i^C)$. Given an image $\mathbf{x}$ along with its label y, the image features are extracted with the visual encoder $\phi(\cdot)$: $\mathbf{f} = \phi(\mathbf{x})$.

$$P(y = i \mid \mathbf{x}) = \frac{\exp(cos(\mathbf{w}_i, \mathbf{f})/\tau)}{\sum_{j=1}^{C} \exp(cos(\mathbf{w}_j, \mathbf{f})/\tau)} \tag{1}$$

where $cos(\cdot)$ denotes the cosine similarity and τ is a learnable temperature parameter in CLIP. Finally, the class label predicted for image $\mathbf{x}$ is given by $\tilde{y} = arg_{max}\ P(y|\mathbf{x})$.

Soft prompt learning: CLIP's reliance on human-crafted prompt templates for generating textual embeddings results in limited adaptability to downstream tasks. Recently, CoOp has utilized soft prompt learning approach from a few samples on the target task. Specifically, CoOp introduces M context vectors $V = \{\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_M\}$ as the learnable prompt. The class embedding $\boldsymbol{c}_i$ of the i-th class is then concatenated with the learnable context vector V for generating the prompts $\mathbf{p}_i = \{\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_M, \boldsymbol{c}_i\}$.

$$P(y = i \mid \mathbf{x}) = \frac{\exp(cos(\theta(\mathbf{p}_i), \mathbf{f})/\tau)}{\sum_{j=1}^{C} \exp(cos(\theta(\mathbf{p}_j), \mathbf{f})/\tau)} \tag{2}$$

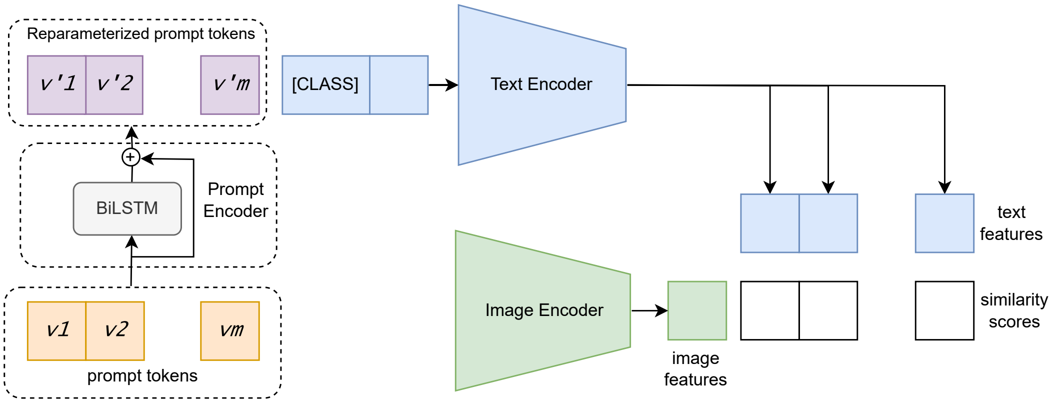


Figure 1: PRE is depicted in the following illustration: The original prompt embeddings V undergo projection using a trainable prompt encoder $\mathcal{F}(\cdot)$ equipped with a residual connection. This process enables the modeling of domain-specific sequential dependencies within the input prompt embeddings and acquires a new mapping for the soft prompts.

The prompts are learned by minimizing the cross-entropy loss:

$$\mathcal{L}_{VL} = -\sum_{c=1}^{C} \log\, P(c|\mathbf{x}) y_c \tag{3}$$

3.2 Prompt Learning with Reparameterization Encoder

While excelling in *Base* classes (68.81% CLIP vs 83.32% CoOp), soft prompt learning exhibits suboptimal results in novel classes (74.43% CLIP vs 66.92% CoOp), Table 1.

In this work, we propose a more adaptable parameterization of soft prompts, achieved through the utilization of a prompt encoder (Fig. 1). This encoder can be trained on the downstream task to enable domain-specific modifications to the prompt embeddings before forwarding them into the fixed text encoder. Specifically, we project the original prompt embeddings V consisting of M prompt tokens $\{\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_M\}$ through the prompt encoder into a reparameterized sequence $\tilde{\boldsymbol{V}}$ as follows:

$$\tilde{\boldsymbol{V}} = [\tilde{\boldsymbol{v}}_1, \tilde{\boldsymbol{v}}_2, ..., \tilde{\boldsymbol{v}}_M] = [\mathcal{F}(\boldsymbol{v}_1), \mathcal{F}(\boldsymbol{v}_2), ..., \mathcal{F}(\boldsymbol{v}_M)] \tag{4}$$

where $\mathcal{F}(\cdot)$ represents the reparameterization function of the prompt encoder, which consists of a network $\varphi(\cdot)$ with a residual connection. $\mathcal{F}(\cdot)$ is applied to each prompt token:

$$\mathcal{F}(\boldsymbol{v}_i) = \varphi(\boldsymbol{v}_i) + \boldsymbol{v}_i, i \in \{1...M\} \tag{5}$$

Network $\varphi(\cdot)$ acts as an adaptable mechanism that establishes associations between prompt embeddings and enables task-relevant reparameterization in the prompt embeddings. Different encoder network architectures exhibit varying effects on the model's adaptability (Ablation Studies). In PRE, we propose a one-layer BiLSTM architecture, as in Fig. 2. BiLSTM processes the input prompt sequence in both forward and backward directions simultaneously. By considering both backward and forward context, it enhances the modeling of the domain-specific sequential dependencies in the projected prompt tokens. The

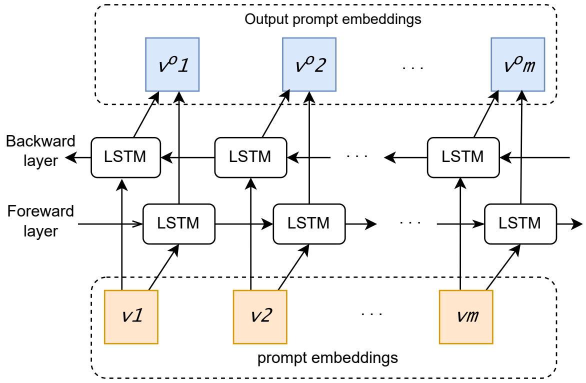


Figure 2: Illustration of one-layer Bidirectional LSTM architecture used in Prompt Encoder.

encoder itself has a skip connection. It enables the model to be more flexible in combining the original embedding of each prompt token with the mapping representation derived from the network $\varphi(\cdot)$ (Equation (5)). By employing this residual-style blending, PRE can flexibly combine the original knowledge encoded from CLIP and the newly learned knowledge acquired from the few-shot training examples through the BiLSTM network.

Once the projected context sequence $\tilde{\boldsymbol{V}}$ is obtained from the prompt encoder, it is merged with the class token embedding $\boldsymbol{c}_i$. This combined input is then fed into the text encoder $\theta(\cdot)$, resulting in the generation of a class-specific encoded adaptable prompt $\mathbf{t}_i^r = \theta(\mathbf{r}_i)$.

$$P(y = i \mid \mathbf{x}) = \frac{\exp(cos(\mathbf{t}_i^r, \mathbf{f})/\tau)}{\sum_{j=1}^{C} \exp(cos(\mathbf{t}_j^r, \mathbf{f})/\tau)} \tag{6}$$

Training: We train the prompt embeddings V and the encoder parameters $\mathcal{F}(\cdot)$ on the downstream task, while preserving all other parameters fixed. The training objective is to maximize the log-likelihood of correct output y given the encoded learnable prompt $\mathbf{t}_c^r$:

$$\mathcal{L} = -\sum_{c=1}^{C} \log\, p(c|\mathbf{x}; \mathbf{t}_c^r) y_c \tag{7}$$

The gradients can be back-propagated through the text encoder $\theta(\cdot)$ and the prompt encoder $\mathcal{F}(\cdot)$. This differential optimization not only utilizes the pre-trained knowledge stored in the fixed text encoder's parameters but also takes advantage of the prompt embeddings projected by the reparameterizing encoder. This combination guides the gradient towards convergence on more generalizable prompts to downstream tasks.

4 Experiments

Following CLIP, CoOp, and CoCoOp, we mainly evaluate the accuracy of our proposed method based on generalization from base-to-new classes within a dataset.

Datasets: The base-to-new generalization is conducted on 8 datasets, namely: Caltech101 for generic object classification; Oxford-Pets, Stanford Cars, Flowers102, Food101, FGVC Aircraft for fine-grained visual categorization; DTD for texture classification; and EuroSAT for satellite image classification.

Table 2: Comparison with existing methods in base-to-new generalization setting with ViT-B/16 as the backbone. The context length M is 4 with the 16-shot samples from the base classes. H: Harmonic mean (to highlight the generalization trade-off Xian et al. (2017)).

(a) Average over 8 datasets.

	Base	New	H
CLIP	68.81	**74.43**	71.42
CoOp	**83.32**	66.92	73.34
CoCoOp	80.89	70.99	74.47
ProGrad	82.96	70.30	75.58
PRE	82.02	71.90	**76.22**

(b) Caltech101.

	Base	New	H
CLIP	96.84	**94.00**	95.40
CoOp	**98.11**	93.02	95.50
CoCoOp	97.96	93.81	95.84
ProGrad	98.02	93.89	**95.91**
PRE	98.00	93.50	95.70

(c) OxfordPets.

	Base	New	H
CLIP	91.17	97.26	94.12
CoOp	94.24	96.66	95.43
CoCoOp	95.20	**97.69**	96.43
ProGrad	95.07	97.63	96.33
PRE	**95.27**	97.61	**96.43**

(d) StanfordCars.

	Base	New	H
CLIP	63.37	**74.89**	68.85
CoOp	**76.20**	67.14	71.38
CoCoOp	70.49	73.59	72.01
ProGrad	76.68	68.63	72.43
PRE	75.83	69.90	**72.74**

(e) Flowers102.

	Base	New	H
CLIP	72.08	**77.80**	74.83
CoOp	**97.63**	66.55	79.15
CoCoOp	94.87	71.75	81.71
ProGrad	95.54	71.87	**82.03**
PRE	96.04	70.80	81.64

(f) Food101.

	Base	New	H
CLIP	90.10	91.22	90.66
CoOp	89.44	86.50	87.95
CoCoOp	90.70	91.29	90.99
ProGrad	90.37	89.59	89.98
PRE	**90.96**	**91.46**	**91.21**

(g) FGVCAircraft.

	Base	New	H
CLIP	27.19	**36.29**	31.09
CoOp	39.24	23.49	29.39
CoCoOp	33.41	23.71	27.74
ProGrad	**40.54**	27.57	32.82
PRE	35.63	32.43	**34.53**

(h) DTD.

	Base	New	H
CLIP	53.24	**59.90**	56.37
CoOp	**80.17**	47.54	59.69
CoCoOp	77.01	56.00	**64.85**
ProGrad	77.35	52.35	62.44
PRE	77.84	53.93	63.70

(i) EuroSAT.

	Base	New	H
CLIP	56.48	64.05	60.03
CoOp	**91.54**	54.44	68.28
CoCoOp	87.49	60.04	71.21
ProGrad	90.11	60.89	72.67
PRE	86.23	**64.47**	**73.78**

Models: Our implementation is based on CoOp's code Zhou et al. (2022b) with the CLIP model. The experiments were conducted based on the vision backbone ViT-B/16 image encoder Dosovitskiy et al. (2020). Similar to CoCoOp, we set the prompt tokens to 4 and initialize the context vectors using the template "a photo of a []". The class names are inserted at the end of these random templates. The final performance is averaged over three random seeds.

Training Details: We maintain consistency with CoOp Zhou et al. (2022b) and CoCoOp Zhou et al. (2022a) in terms of training epochs and training procedures which adopted the SGD optimizer with 0.002 initial learning rate, CosineAnnealingLR schedule. We conducted all training and testing on two NVIDIA GeForce RTX 3080 Ti GPUs.

Baselines: We present the results of the PRE method to compare its performance with CLIP (hand-crafted prompts) and three soft prompt learning methods including CoOp, CoCoOp, and ProGrad.

4.1 Generalization From Base-to-New Classes

On each of the 8 datasets, we split the classes equally into two groups, one as *Base* classes and the other as *New* classes. Learning-based models, i.e., CoOp, CoCoOp and PRE, are trained using only the *Base* classes while evaluation is conducted on the *Base* and *New* classes separately to assess the generalization capabilities. The detailed results on 16-shot settings, $M = 4$ learnable prompts on ViT-B/16 are shown in Table 2.

PRE Significantly Narrows Generalization Gap. Compared with existing methods, our proposed PRE method achieves the highest *New* performance on three out of eight datasets. In direct comparison with CoOp, PRE demonstrates significant improvements in accuracy for unseen classes. Specifically, the accuracy increases more than 5% from 67.14% to 71.90%, substantially narrowing the gap between soft prompt learning with manual prompts. The results confirm that our reparameterizing encoder $\mathcal{F}(\cdot)$ coupled with a residual connection, proficiently captures task-relevant dependencies within the initial prompt embeddings. This associate-learned projected prompt sequence navigates a more efficient optimization process to find a contextually generalizable prompt specific to the downstream domain. Furthermore, when compared to both CoCoOp and ProGrad, PRE showcases a relatively improved performance in novel classes - averaging 71.90% against CoCoOp's 70.99% and ProGrad's 70.30%. This consistent enhancement in novel classes' performance across various classes demonstrates PRE's remarkable ability to balance performance across a wide spectrum of domains.

Regarding the Harmonic mean (represents the generalization trade-off Xian et al. (2017)), PRE outperforms all other methods. Our proposed method consistently achieves the highest Harmonic mean across five out of eight datasets. On average, it surpasses CLIP by 5%, CoOp by 3%, CoCoOp by nearly 2%, and ProGrad by 1%. Besides the performance improvements in novel classes, this is partly attributed to the fact that PRE maintains a good *Base* classes' performance. Specifically, PRE achieves a remarkable 6.7% higher Base accuracy than CLIP and surpasses CoCoOp on six out of eight datasets with an average accuracy of 82.02% vs 80.89%. This performance improvement in seen classes comes from the efficacy of the residual connection, which dynamically balances and blends knowledge from pretrained CLIP and the newly acquired insights from the prompt encoder based on few-shot samples.

Various K-shot samples: Table 3 summarizes the average performance across all 8 datasets, considering various K-shot samples on ViT-B/16. Similar to the observations

Table 3: Comparison in the base-to-new setting with different K-shot samples in terms of the average performance among all 8 datasets with backbones ViT-B/16.

Methods	Prompts	$K = 4$			$K = 8$			$K = 16$		
		Base	New	H	Base	New	H	Base	New	H
CoOp	textual	**79.33**	66.02	71.47	**81.67**	66.29	71.89	**83.32**	66.92	73.34
CoCoOp	textual+visual	76.51	**71.48**	73.68	78.67	70.78	74.14	80.89	70.99	74.47
ProGrad	textual	79.15	70.40	74.11	80.55	70.84	75.00	80.96	70.30	75.58
PRE	textual	78.62	71.31	**74.46**	79.84	**71.49**	**75.07**	82.02	**71.90**	**76.22**

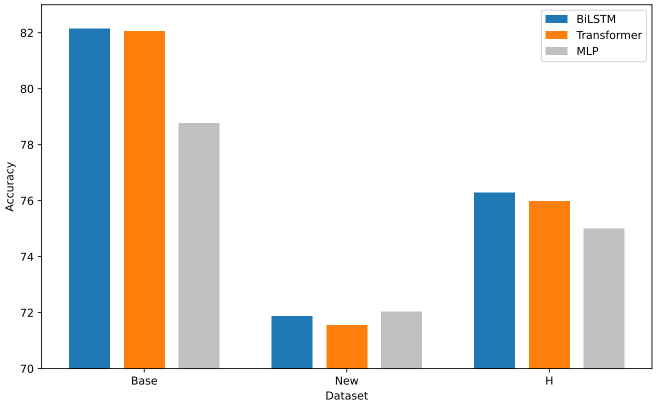


Figure 3: The impact of different reparameterizing encoder network architectures on average Accuracy on *Base*, *New* and *H* over 8 datasets on the 16-shot setting.

in the 16-shot settings, PRE consistently achieves a higher average Harmonic mean than existing methods. CoOp still achieves the best performance regarding *Base* classes while obtaining the worst *New* class performance in all K-shot samples. The performance gap between PRE and others is most significant when K = 16 shots and tends to diminish as K decreases. This phenomenon is because the reparameterizing encoder $\mathcal{F}(\cdot)$ in PRE requires a certain number of data samples to fully explore its potential in learning the optimal prompt embeddings.

4.2 Ablation Studies

In this subsection, we conducted ablation studies to investigate the effectiveness of different components in PRE.

Parameter-efficiency of PRE: The total number of trainable parameters in PRE consists of 1) trainable prompt embeddings, and 2) reparameterizing encoder network. Our encoder network $\mathcal{F}(\cdot)$ contains a one-layer Bidirectional LSTM. Let M be the number of prompt tokens and d be the dimensionality of model embeddings, the BiLSTM network has an input size of d and a hidden size of $d/2$, and it incorporates two separate LSTM units for the forward and backward direction. For each LSTM unit, the weight matrix for input-to-hidden connections has a shape of $4 \times d \times d/2$, and the weight matrix for hidden-to-hidden connections has a shape of $4 \times d/2 \times d/2$. Taking into account both the forward and backward LSTM units, we have $3 \times d \times d$ parameters of the reparameterizing encoder network and $d \times M$ soft prompt parameters. As the number of prompt tokens M varies from 1 to 16, the number of trainable parameters in PRE remains relatively low, resulting in efficient training time compared to other methods (Table 1).

Effects of different reparameterization encoder network architectures: To gain a deeper understanding of the reparameterizing mechanism and how its choice of network architecture impacts the model's performance, we have implemented a range of network architectures in the prompt encoder including a bottle-neck MLP, Transformer Encoder network, and BiLSTM, all equipped with a residual connection. All three networks lead to improvements in novel classes and the Harmonic mean compared to the CoOp

Table 4: Comparison in different encoder network architectures with vs. without Residual connection in terms of the average *Base*, *New*, and *H* performance in the 16-shot setting.

Encoder Network	Residual connection	Accuracy (%)		
		Base	New	H
BiLSTM	**Yes**	82.02	71.90	76.22
BiLSTM	No	80.51	70.57	74.67
		-1.64	-1.31	-1.61
Transformer Encoders	**Yes**	82.05	71.55	75.98
Transformer Encoders	No	82.01	69.34	74.54
		-0.04	-2.21	-1.44
Bottleneck MLP	**Yes**	78.77	72.04	75.00
Bottleneck MLP	No	74.54	68.88	71.16
		-4.23	-3.16	-3.84

method (Table 4). These results underscore the efficacy of the prompt reparameterization technique for adapting VLMs, although the outcomes vary based on the chosen network architecture. We observe that the Transformer Encoder network follows a similar trend to the BiLSTM, albeit with slightly lower performance. While MLP-based reparameterization demonstrates an improvement in *New* class accuracy, it comes with a significant decline in performance for the *Base* class. This is due to bottleneck MLPs' inherent limitations in capturing long-range sequential dependencies and contextual information within prompt embeddings. As feedforward networks, MLPs process tokens independently, neglecting their relationships. This lack of contextual understanding may result in less contextually rich prompt embeddings. Among three architectures, BiLSTM has the most promising results and isn't affected by specific hyperparameter choices, making them easier to train and less prone to overfitting in few-shot training examples.

Residual connection is an important component in PRE: According to Table 4, the performance of both the *Base*, *New*, and *H* decreases when the skip connection is removed from the prompt encoder in all network architectures. Notably, MLP exhibits a significant drop of up to 4.23% in accuracy for *Base* classes and 3.16% in the *New* class performance, demonstrating a significant vulnerability in handling input prompt embeddings. This is because the MLPs are inherently shallow networks without inherent memory mechanisms to capture sequential dependencies. As a result, when the residual connection is removed, without the complementary from the original prompt embeddings, the MLP lacks the ability to effectively propagate contextual information through the layers, leading to a more significant performance degradation.

Limitations and Future Works: In terms of *New* class performance, PRE's performance lags behind that of CLIP in 5 out of the 8 datasets (as seen in Table 2). In terms of future work, one direction is to further develop the reparameterizing prompt encoder network with potentially a more efficient architecture that can enhance the model's generalizability. Furthermore, a promising avenue is to investigate the influence of our reparameterizing encoder within the context of other VLMs tuning methods such as adapter-based. We believe that with targeted modifications, our approach could potentially be integrated to accelerate certain aspects of these models, and this will be explored in future research.

5 Conclusions

In this paper, we presented PRE - a practical soft prompt learning method for Vision Language adaptation with limited data and resource constraints. Our method demonstrates its efficiency in achieving improved generalization performance over prior works while maintaining performance on the seen classes. We hope that our approach could be explored in collaboration with other well-established methods in the future, thereby contributing to the overall enhancement of adaptability within the realm of Vision Language models.

Acknowledgement

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No. 101135800 (RAIDO).

References

Hangbo Bao, Wenhui Wang, Li Dong, Qiang Liu, Owais Khan Mohammed, Kriti Aggarwal, Subhojit Som, Songhao Piao, and Furu Wei. Vlmo: Unified vision-language pre-training with mixture-of-modality-experts. *Advances in Neural Information Processing Systems*, 35:32897–32912, 2022.

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.

Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez-Opazo, and Stephen Gould. A recurrent vision-and-language bert for navigation. *arXiv preprint arXiv:2011.13922*, 2020.

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.

Woojeong Jin, Yu Cheng, Yelong Shen, Weizhu Chen, and Xiang Ren. A good prompt is worth millions of parameters: Low-resource prompt-based learning for vision-language models. *arXiv preprint arXiv:2110.08484*, 2021.

Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. Mdetr-modulated detection for end-to-end multi-modal understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1780–1790, 2021.

Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In *International Conference on Machine Learning*, pages 5583–5594. PMLR, 2021.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.

LH Li, M Yatskar, D Yin, CJ Hsieh, and KW Chang. A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021.

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.

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.

Anastasia Razdaibiedina, Yuning Mao, Rui Hou, Madian Khabsa, Mike Lewis, Jimmy Ba, and Amjad Almahairi. Residual prompt tuning: Improving prompt tuning with residual reparameterization. *arXiv preprint arXiv:2305.03937*, 2023.

Yongqin Xian, Bernt Schiele, and Zeynep Akata. Zero-shot learning-the good, the bad and the ugly. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4582–4591, 2017.

Lewei Yao, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu. Filip: Fine-grained interactive language-image pre-training. *arXiv preprint arXiv:2111.07783*, 2021a.

Yuan Yao, Ao Zhang, Zhengyan Zhang, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. Cpt: Colorful prompt tuning for pre-trained vision-language models. *arXiv preprint arXiv:2109.11797*, 2021b.

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, 2022a.

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, 2022b.

This appendix is organized as follows:

- Section A provides the experimental and dataset details for PRE.

- Section B provides extended Ablation studies about the effect of the context length, prompt initialization, and parameter sharing on PRE of the base-to-new generalization experiments. It also studies different residual network architectures in PRE.

- Section C provides the interpretation of the learned textual prompts using the nearest words in the embedding space.

- Section D gives additional detailed results for each dataset of the base-to-new generalization experiments in different K-shot settings and the detailed experiment results to see the impact of the residual network as well as parameter sharing settings on PRE.

.1 EXPERIMENTAL DETAILS

The detailed instructions on how to run the PRE method to reproduce the results are available at the GitHub link: github.com/minhanh151/PRE.

.1.1 Datasets Details

The datasets utilized in our experiments align with the ones employed in the CoOp Zhou et al. (2022b). These datasets encompass 8 different benchmarks designed for few-shot visual recognition. For comprehensive reference, Table A1 provides detailed information about each dataset, such as the number of classes, sizes of the training and testing sets, and the original image recognition tasks associated with each dataset.

.1.2 Training Details:

Adopting the training settings from CoOp Zhou et al. (2022b), we maintain a consistent training epoch of 50 for all the experiments conducted with various shots. For prompt-based models, we employ a batch size of 32, except in the case of CoCoOp. As reported by (Zhou et al., 2022) Zhou et al. (2022a), CoCoOp exhibits a considerable GPU memory consumption when the batch size is set larger than one. Hence, we follow their original configuration and set the batch size to 1 for CoCoOp in our experiments. We conducted our experiments with two NVIDIA GeForce RTX 3080 Ti GPUs, with 8 GB of memory each. On each task, training took between 4 minutes and 2 hours.

.1.3 Hyperparameters

The original CoOp method has different versions with different class token positions and parameter initialization strategies. To maintain consistency, we selected a specific configuration for our baseline, where the token position is ”end,” the parameter initialization strategy is ”a photo of a,” and the length of learnable context tokens is 4, similar to the CoOp and CoCoOp settings.

Table 1: Details of 8 Datasets for Few-Shot Visual Recognition and Base-to-New Generalization Image Recognition Evaluation.

Dataset	Classes	Training Size	Testing Size	Task
Caltech101 (Fei-Fei et al., 2004)	100	4,128	2,465	Object Recognition
DTD (Cimpoi et al., 2014)	47	2,820	1,692	Texture Recognition
EuroSAT (Helber et al., 2019)	10	13,500	8,100	Satellite Image Recognition
FGVCAircraft (Maji et al., 2013)	100	3,334	3,333	Fine-Grained Aircraft Recognition
Flowers102 (Nilsback & Zisserman, 2008)	102	4,093	2,463	Fine-Grained Flowers Recognition
Food101 (Bossard et al., 2014)	101	50,500	30,300	Fine-Grained Food Recognition
OxfordPets (Parkhi et al., 2012)	37	2,944	3,669	Fine-Grained Pets Recognition
StanfordCars (Krause et al., 2013)	196	6,509	8,041	Fine-Grained Car Recognition

.2 EXTENDED ABLATION STUDIES

.2.1 Effect of context length

In this study, we examine the significance of the context length for the learnable prompts. To analyze its impact on base-to-new generalization, we conduct experiments using the ViT-16/B backbone with PRE models. Following a similar approach to CoOp Zhou et al. (2022b), we investigate three context lengths: 4, 8, and 16 context tokens. For context lengths of 8 and 16, the prompt is initialized as ”X X ... X a photo of a [Class]”. The performance using PRE across 8 datasets is then averaged and summarized in Fig. A1. Notably, we observe that setting the context length to 8 consistently yields superior performance compared to the other two settings across all three evaluation metrics. Learning prompts with context lengths of 4 and 16 exhibit similar performance levels on PRE. To ensure a fair comparison with CoOp and CoCoOp, we ultimately opt to set the context length to 4 in our final model. This decision ensures consistency in the experimental setup and facilitates a proper evaluation against the previous state-of-the-art methods.

.2.2 Effect of prompt initialization

Lester et al. (2021) Lester et al. (2021) find that the initialization of prompt parameters plays a major role in the final performance. To assess the influence of prompt initialization on prompt tuning, we carry out a comparative analysis employing two different methods: word embeddings-based initialization (’w/ init’) and random initialization (’w/o init’) on PRE. For random initialization, we utilize a zero-mean Gaussian distribution with a standard deviation of 0.02 to initialize the prompt tokens. On the other hand, word embeddings-based initialization involves initializing the prompt tokens with the phrase ”a photo of a.” with a context length of 4. After conducting our experiments and evaluating the performance across 8 datasets, we have summarized the results in Fig. A2. Notably, we have observed that employing word embeddings-based initialization yields slightly higher performance in all three evaluation metrics when compared to random initialization in PRE. This finding highlights the significance of the initialization strategy in prompt tuning and its substantial impact on the overall performance of the model.

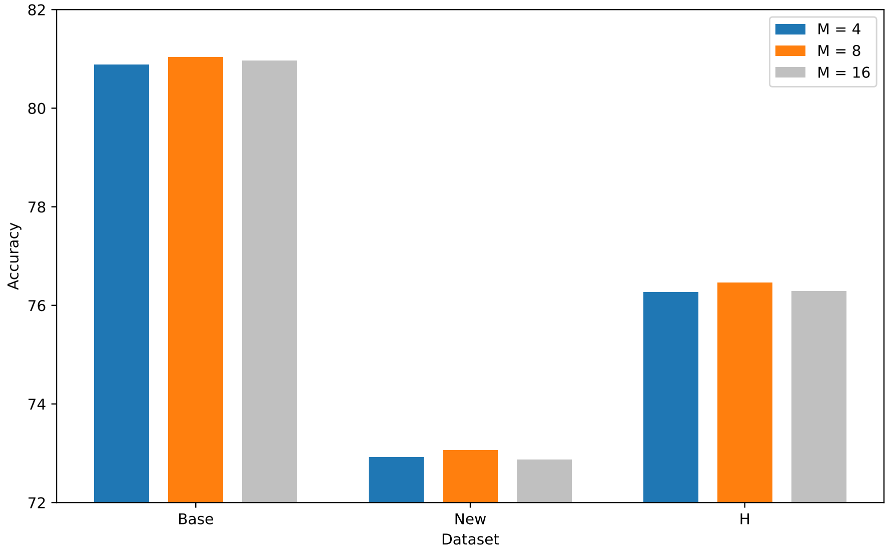


Figure 1: The impact of different context length on average Accuracy on *Base*, *New* and *H* over 8 datasets using PRE model.

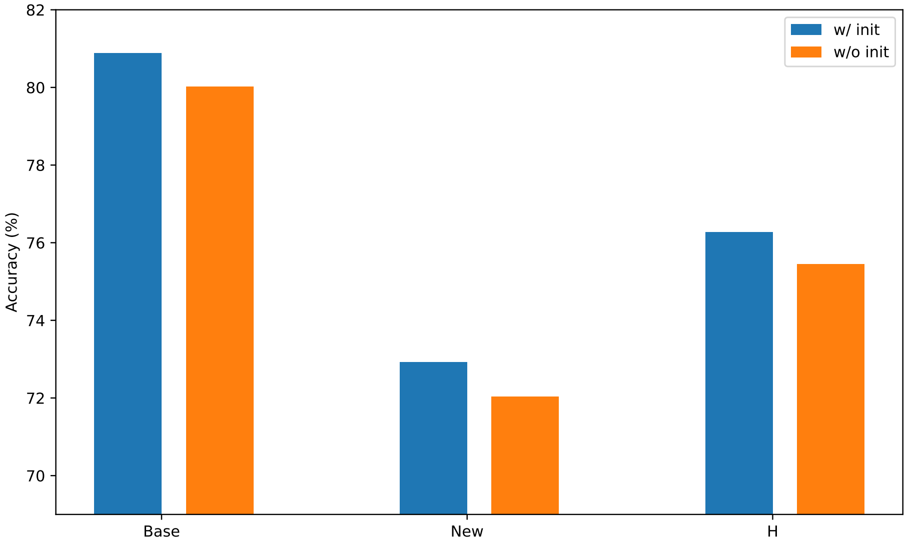
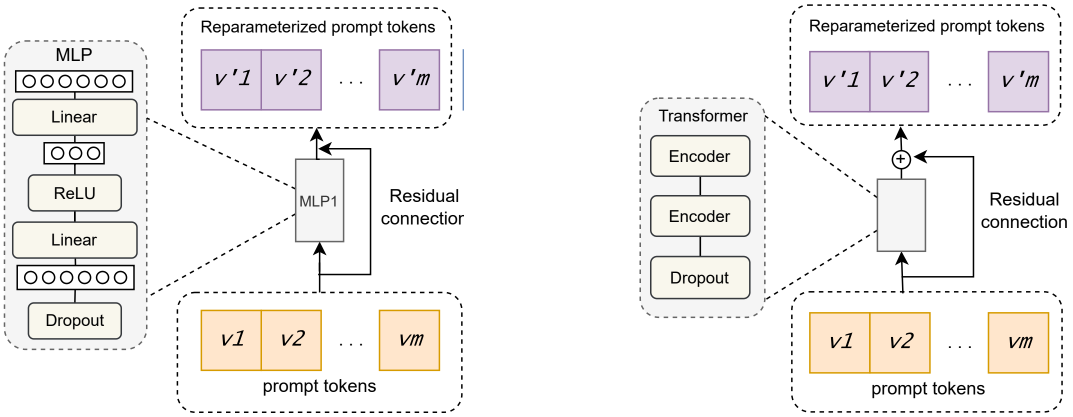


Figure 2: Comparison with vs without prompt initialization on PRE on average Accuracy on *Base*, *New* and *H* over 8 datasets using PRE model.

.2.3 Studies of different network architectures for Prompt Encoder

In this study, we examine other different network architectures for reparameterizing soft prompt embeddings in PRE. Following Anastasia et al. (2023) Razdaibiedina et al. (2023), we use a bottleneck MLP network (as shown in Fig. A3a). Specifically, this MLP network consists of a down-projection linear layer followed by a ReLU layer, and an up-projection linear layer, with a dropout layer at the end. The bottleneck size hyperparameter has been



(a) A shared bottleneck MLP for every prompt token

(b) A shared Transformer Encoder network for every prompt token

Figure 3: Two other network architectures for the prompt encoder in PRE.

Table 2: Comparison in PRE using different reparameterizing network architectures in terms of the average *Base*, *New*, and *H* performance in the base-to-new generalization setting.

Encoder Network	Parameter Sharing	Accuracy (%)		
		Base	New	H
BiLSTM	**Share**	82.02	71.90	76.22
BiLSTM	Separate	82.72	69.02	74.60
Bottle-neck MLP	**Share**	78.77	72.04	75.00
Bottle-neck MLP	Separate	80.26	72.07	75.54
Transformer Encoders	Share	82.05	71.55	75.66
Transformer Encoders	**Separate**	81.49	70.21	74.97

thoroughly studied to get the optimal performance. We also explored a Transformer Encoder network consisting of two Transformer encoder layers followed by a dropout layer, each layer equipped with two attention heads (as in Fig. A3b). The residual connection is used in all three network architectures (BiLSTM, MLP, and Transformer Encoders network) to combine the input prompt embeddings and the output of the reparameterization network. The reparameterizing network can be shared among every prompt token or separately process each prompt token using separate parameters. Table A2 shows the results of these networks on the base-to-new generalization setting. These results have been discussed in the Ablation Studies section. Following that, we expand the ablation study to investigate the impact of parameter sharing on these networks in the next part.

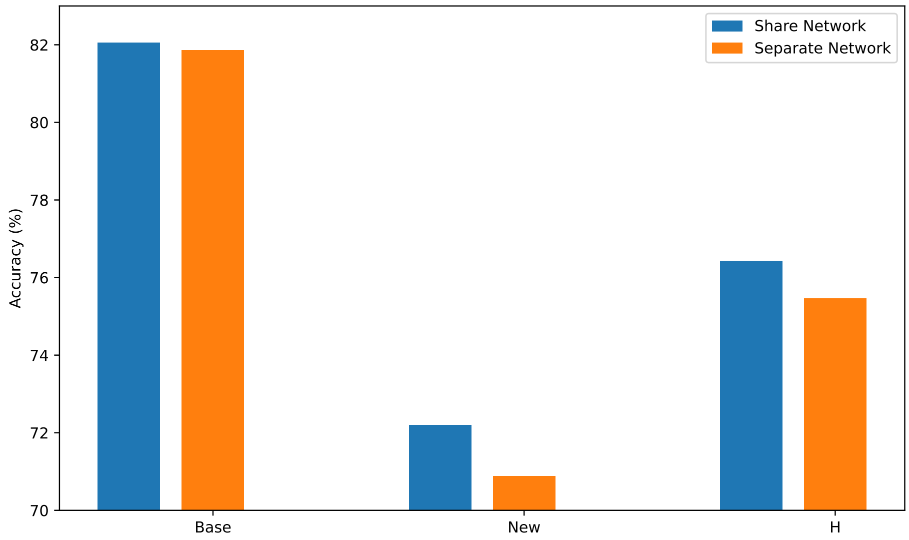


Figure 4: Performance of PRE with shared and separate BiLSTM networks on average Accuracy on *Base*, *New* and *H* over 8 datasets.

.2.4 Effect of processing prompt tokens independently by separate networks vs. jointly by a shared network in Prompt Encode

Fig. A3 illustrates the performance comparison of PRE when employing a shared BiLSTM network for every prompt token versus separate BiLSTM networks for each prompt token while Fig. A4 shows the impact of a shared MLP network vs. separate MLP networks on the prompt encoder. The detailed results of parameter sharing on three kinds of network architecture BiLSTM, Transformer Encoders, and MLP are shown in Table 2. We can see that the shared network exhibits significantly better performance on the *Base* classes, while still maintaining strong generalization capabilities. This performance improvement indicates that parameter sharing in the BiLSTM Encoder network of PRE is highly beneficial, as it allows for the effective learning of dependencies between prompt tokens, outperforming the separate network setting.

Table A3 shows the detailed results on all three network architectures. The downward trend is observed in the case of the Transformer Encoder network as well. This trend is consistent across both network architectures because they both utilize the benefits of processing prompt tokens collectively, which allows them to capture the extended dependencies present within the sequence of prompts.

In contrast, for the MLP network, the separate network configuration yields relatively higher performance on the *Base* classes while maintaining the performance in novel classes. MLP is not affected by the absence of parameter sharing as it does not primarily aim to capture interactions between prompt tokens. In contrast, separate MLP networks with increased parameters can enhance the encoder's adaptability in exploring various mappings of prompt embeddings.

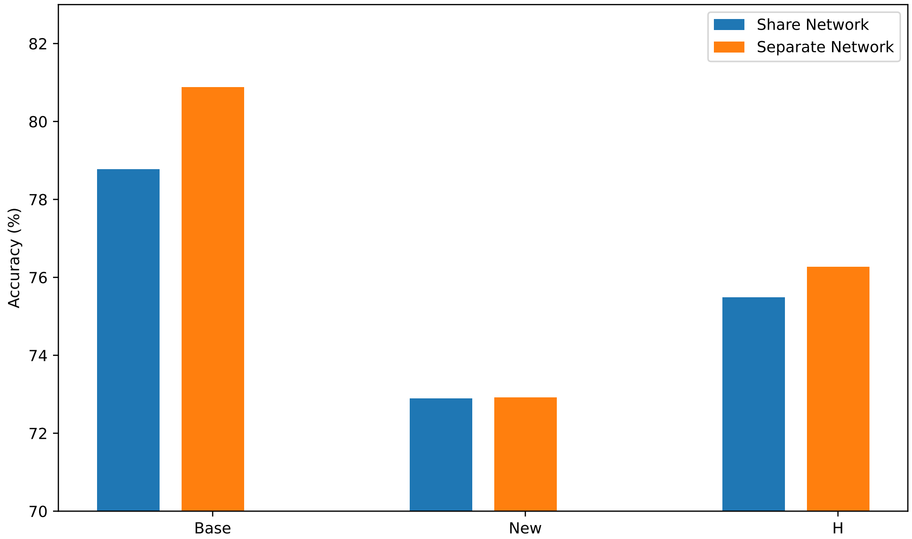


Figure 5: Performance of PRE with shared and separate MLP networks on average Accuracy on *Base*, *New* and *H* over 8 datasets.

Table 3: The nearest words for each of the 4 context vectors learned by PRE with BiLSTM network as prompt encoder, with their distances shown in parentheses.

#	Word 1	Word 2	Word 3	Word 4
Caltech101	borderlands (0.6706)	oth (0.6446)	pose (0.6732)	mark (0.7446)
OxfordPets	ol (0.5578)	moves (0.6612)	**wild** (0.6559)	mountain (0.5700)
Flowers102	screen (1.1097)	photoo (1.1657)	battles (1.4090)	sey (1.2732)
FGVC Aircraft	can (1.4438)	independent (1.8402)	**tail** (1.7982)	campaigner (1.0018)
DTD	aster (0.9711)	**consecutive** (0.8758)	**line** (1.0512)	stones (1.0743)
EuroSAT	three (0.4983)	report (0.5567)	rain (0.4577)	pose (0.5220)
Stanford Cars	salt (0.9016)	riot (1.1503)	N/A (0.9737)	toby (1.1054)
Food101	tur (1.1864)	color (0.7646)	lh (1.0958)	water (0.9743)

.3 INTERPRETATION OF PROMPTS

The prompts learned through optimization in the continuous space are difficult for humans to understand Zhou et al. (2022b). To address this issue, CoOp introduces a method that utilizes the nearest words in the embedding space to represent and visualize the learned prompts. In line with this approach, we present the nearest words corresponding to our learned prompts across 8 datasets in Table A3 and Table A4. Like CoOp's findings, most of these words remain difficult to interpret directly through human logic. However, we can still see some connections between these prompts and the corresponding images in certain datasets. For instance, in the OxfordPets dataset, the prompts learned by PRE seem to emphasize elements related to ”wild” and ”mountain” environments in each image. In the

Table 4: The nearest words for each of the 4 context vectors learned by PRE with Transformer Encoder network as prompt encoder. N/A means non-Latin characters.

#	Word 1	Word 2	Word 3	Word 4
Caltech101	draws (0.5176)	ability (0.4955)	mirrors (0.6053)	pre (0.4966)
OxfordPets	sep (0.5221)	line (0.6307)	living (0.4938)	marked (0.5395)
Flowers102	bag (0.7056)	paint (0.5929)	pup (0.5806)	pray (0.8519)
FGVC Aircraft	nail (0.7897)	cranl (1.2032)	mor (0.7339)	wound (0.8062)
DTD	aster (0.9711)	**consecutive** (0.8758)	**line** (1.0512)	stones (1.0743)
EuroSAT	ma (0.4567)	kt (0.5447)	wong (0.4650)	N/A (0.4693)
Stanford Cars	mines (0.4830)	line (0.4581)	**heights** (0.4842)	sheet (0.4345)
Food101	buddies (0.4452)	losing (0.7646)	voter (0.5958)	marching (0.4493)

FGVC Aircraft dataset, the learned prompts from PRE show a preference for the tail region to distinguish between different types of airplanes. Moreover, in the DTD dataset, PRE with both types of encoder network focuses on prompts associated with ”consecutive” and ”line”, which appear to be indicative of specific image texture characteristics. It demonstrates that the learned multiple prompts focus on particular attributes of categories.

.4 DETAILED RESULTS

This section presents a detailed comparison of the proposed PRE approach with existing methods in various aspects. Table A5 shows the results in the base-to-new generalization setting of PRE using a shared network and separate network as the reparameterizing mechanism for input prompt embeddings.

Table 5: Detailed results on 8 datasets to see the impact of residual connection on different encoder architectures of PRE in the base-to-new generalization setting with ViT-B/16 as the backbone. The context length M is 4 with the 16-shot samples from the base classes.

Dataset	**Set**	**CoOp**	**BiLSTM**		**Transformer Encoders**		**MLP**	
			Residual	W/o Residual	**Residual**	W/o Residual	**Residual**	W/o Residual
Average	Base	83.32	82.02	80.51	82.05	82.01	78.77	74.54
	New	66.92	**71.90**	70.57	**71.55**	69.34	**72.04**	68.88
	H	73.34	**76.22**	74.67	**75.98**	74.54	**75.00**	71.16
Caltech101	Base	98.11	98.00	97.77	98.00	98.00	97.63	97.10
	New	93.02	93.50	93.90	93.40	92.10	94.17	94.200
	H	95.50	95.70	95.80	95.64	94.96	95.87	95.63
OxfordPets	Base	94.24	95.27	94.90	95.37	96.50	95.37	95.30
	New	96.66	97.60	97.13	97.00	95.30	97.59	97.50
	H	95.43	96.42	96.00	96.18	95.90	96.47	96.39
Flowers102	Base	97.63	96.40	96.73	96.00	96.62	89.23	86.60
	New	66.55	70.80	68.43	70.90	67.93	72.53	74.60
	H	79.15	81.64	80.16	81.56	79.77	80.02	80.15
FGVC Aircraft	Base	39.24	35.63	34.40	36.70	37.10	33.82	23.66
	New	23.49	32.43	33.30	31.40	30.33	32.03	11.24
	H	29.39	34.53	33.84	33.84	33.38	32.90	15.24
DTD	Base	80.17	77.80	74.70	78.60	79.20	72.37	50.87
	New	47.54	53.93	54.50	52.75	51.30	54.60	52.10
	H	59.69	63.70	63.02	63.13	62.27	62.24	51.48
EuroSAT	Base	91.54	86.23	85.10	87.40	87.00	81.60	82.33
	New	54.44	64.47	56.50	63.73	57.10	62.00	60.20
	H	68.28	73.78	67.91	73.71	68.95	70.46	69.55
Stanford Cars	Base	76.20	75.83	71.47	73.34	72.74	70.20	71.50
	New	67.14	69.90	72.77	71.87	72.53	72.93	72.20
	H	71.38	72.74	72.11	72.60	72.86	71.54	71.85
Food101	Base	71.38	73.20	72.11	72.60	72.86	71.54	71.85
	New	86.5	91.46	88.02	91.36	88.12	91.02	89.02
	H	87.95	91.21	88.52	91.16	88.27	90.47	88.97