

ProLoG: Hybrid Prompt and LoRA Based Adaptation of Vision-Language Models for OOD Generalization (Technical Appendix)

Jungwuk Park¹, Dong-Jun Han², Jaekyun Moon¹

¹Korea Advanced Institute of Science and Technology (KAIST)

²Yonsei University

savertm9@gmail.com, djh@yonsei.ac.kr, jmoon@kaist.edu

A. Implementation Details

In this section, we present our implementation details for all the experiments. Our overall implementation follows the setup of (Zhou et al. 2022b,a), and the implementation of LoRA follows (Zanella and Ben Ayed 2024). We use CLIP with ViT-B/16 and ViT-B/32 as vision-language backbones: both are used for the base-to-new generalization setting, ViT-B/16 for cross-dataset generalization and domain generalization, and ViT-B/32 for Waterbirds. All experiments are conducted using an NVIDIA RTX 3090 GPU. The model is fine-tuned in a few-shot setting with 16 samples per class for each base task. For the base-to-novel generalization setup, the model is trained using an SGD optimizer for 12 epochs with a batch size of 4 and an initial learning rate of 0.0035. For the cross-dataset generalization setups of UCF101 and DTD, all methods, including ProLoG, MaPLE, PromptSRC, and CoPrompt, are trained for 5 epochs. For ImageNet in the cross-dataset generalization setup and the domain generalization setup, the training of ProLoG is conducted for 2 epochs, and for Waterbirds, the epoch is set to 6. In all cases, the batch size is set to 4, with an initial learning rate of 0.0026, and a cosine annealing scheduler is consistently used to adjust the learning rate across all setups.

For the implementation of the proposed method, the prompts are initialized with descriptions tailored to the content of each dataset, as detailed in . The rank of each LoRA module is consistently set to 4 across all experiments. In the base-to-novel generalization setup, LoRA modules are inserted into all transformer layers, except the first layer, in the image and text encoders of CLIP. In the cross-dataset generalization setting for UCF101 and DTD, LoRA modules are inserted into the second to fourth transformer layers. For the cross-dataset generalization and domain generalization settings on ImageNet, they are inserted into the eighth to tenth transformer layers. We use the same text augmentations as in (Roy and Etemad 2024), where the prompt “a photo of a [class]” is fed into GPT-3 to generate about 10 descriptive variations per class. For each class, multiple text augmentations are generated (about 10 variations), and the regularization loss is computed using the average of their features obtained from the pre-trained CLIP network in each

batch. The coefficient λ for the regularization loss is fixed as 30 across all experiments except for Waterbirds. For Waterbirds, λ is set to 5. Finally, regarding the inference strategy, δ_1 , δ_2 , and C_0 are consistently set to 0.98, 0.85, and 0.5, respectively, throughout all experiments.

Prompt Initialization

For training on the base task in the experimental section, we initialized the learnable prompts with descriptions tailored to the content of each dataset. These dataset-specific prompts are designed to align closely with the visual characteristics of the data, enabling stable training of the vision-language foundation model on the base task. The initial prompt descriptions for each dataset are summarized in Table 1.

| Dataset | Prompt Description |
|---------------------|---------------------|
| OxfordPets | animal portrait |
| OxfordFlowers | flower close-up |
| FGVCAircraft | airplane profile |
| DescribableTextures | close-up pattern |
| EuroSAT | aerial view |
| StanfordCars | automotive exterior |
| Food101 | culinary dish |
| SUN397 | photo image |
| Caltech101 | photo image |
| UCF101 | person action |
| ImageNet | photo image |

Table 1: Dataset-Specific Prompt Initialization

Comparison with random initialization. To see the effect of prompt initialization, we compare our initialization in Table 1 with random initialization using our ProLoG framework. For this experiment, we adopt Gaussian random initialization. Table 2 presents the results in the base-to-new generalization setting. The prompt initialization in Table 1 achieves better performance on new classes while slightly showing lower performance on base classes, resulting in a higher HM score.

B. More details on the Masking Strategy in Our Hybrid Structure

Detailed Explanation of the Masking Strategy

We provide a detailed motivation and explanation for the masking strategy. Standard LoRA is applied to all tokens in

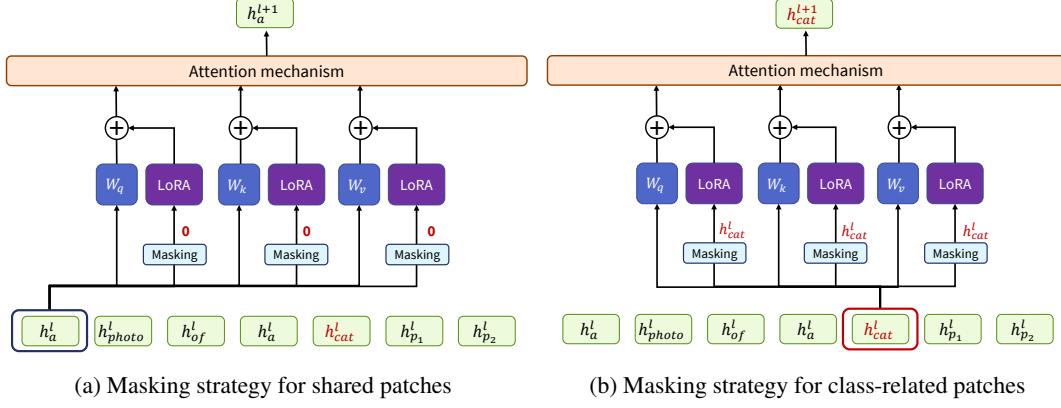


Figure 1: The proposed masking strategy in the hybrid structure.

| | Average on 11 datasets | | |
|--|------------------------|-------|--------------|
| | Base | New | HM |
| ProLoG (with prompt initialization in Table 1) | 84.77 | 76.89 | 80.64 |
| ProLoG (with random initialization) | 84.80 | 76.63 | 80.51 |

Table 2: Comparison with random initialization.

the text input. For example, consider a base class "car" with the common template "A photo of a." In this case, LoRA is applied and optimized on both the shared context ("A photo of a") and the base class token ("dog"). However, it is important to note that the template "A photo of a" is shared between base and target tasks. Therefore, LoRA modules optimized on the shared tokens during training can introduce bias for unseen target classes at test time. For instance, aligning the text embedding of "A photo of a dog" with dog image embeddings during training makes the shared tokens contribute to the alignment. As a result, for unseen target class prompts like "A photo of a car", the shared context may bias the output toward the base class at inference time.

To address this issue caused by the shared patches, our method ensures that LoRA is applied only to class-specific patches while excluding shared patches (including the templates, start/end tokens, and prompts), preventing its application to commonly shared parts, as depicted in Fig. 1. The results in Table 4 of the main paper demonstrate that our masking strategy effectively mitigates the overfitting of LoRA, enhancing generalization on both ID and OOD performance.

Masking Strategy for the Image Encoder

Similar to the text encoder, the class token in the image encoder of CLIP is shared across all image samples in both the base and target tasks. Thus, we define an image binary mask S_I for each image patch as $S_I = [s_{cls}^I, s_1^I, \dots, s_N^I]$ where s_i^I is 0 for the class token patch and 1 otherwise.

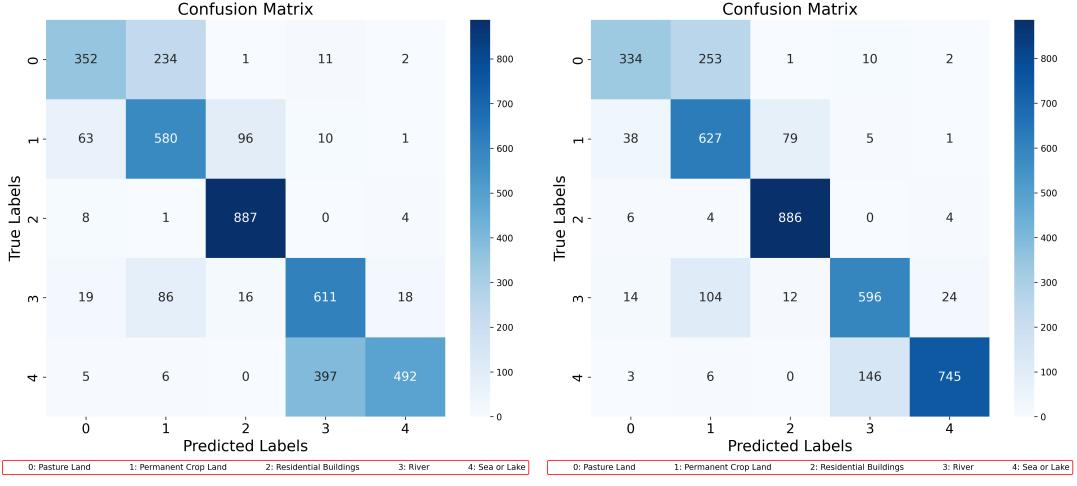
C. Detailed Insights on Scenario II in Our Inference Strategy

In this section, we provide more detailed insights on Scenario II of our inference strategy, as described in the Proposed Algorithm section of the main paper. As mentioned

in the main paper, the contextual knowledge of the base task learned by the prompts offers limited advantages for OOD samples that deviate significantly from the base context. To illustrate this clearly, we use the EuroSAT dataset, which is a satellite-image dataset. In the base-to-new generalization setting, the classes of EuroSAT are divided into two groups: base and new classes. The base classes include "Annual Crop Land", "Forest", "Herbaceous Vegetation Land", "Highway or Road", "Industrial Buildings". The new classes consist of "Pasture Land", "Permanent Crop Land", "Residential Buildings", "River", "Sea or Lake".

It is noteworthy that during training, the model is optimized to classify the base classes like Land, Forest, Road, and Buildings, whereas the new classes include OOD classes, such as River and Sea or Lake, that deviate significantly from the context of the base classes. Therefore, when the model uses the learned prompts to make predictions on a target task involving the new classes, the contextual knowledge embedded in the learned prompts from the base classes could potentially introduce biased predictions for the new classes, such as River and Sea or Lake. This is because the base classes do not share contextual relevance with the OOD classes, leading to suboptimal predictions in certain cases.

Fig. 2a shows the confusion matrix for predictions made by ProLoG without the inference strategy, while Fig. 2b presents the confusion matrix for predictions by ProLoG with the inference strategy. The predictions for the confusion matrix are measured on the new classes of EuroSAT after training on the base classes. The results in Table 2a confirm that without the inference strategy, ProLoG misclassifies many samples from the Sea or Lake (label 4) class as River (label 3). In contrast, ProLoG with the inference strategy effectively classifies these classes, as shown in Table 2b, demonstrating that our ensemble predictions enhance generalization on OOD classes by integrating the context knowledge of the base task with a context-agnostic perspective and considering diverse viewpoints.



(a) Confusion matrix of ProLog w/o the inference strategy (b) Confusion matrix of ProLog w/ the inference strategy

Figure 2: Proposed masking strategy in the hybrid structure.

| | Average on 11 datasets | | | Base | ImageNet New | HM | Base | Flowers102 | | | Base | OxfordPets New | HM |
|-----------------------------------|------------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|----------------|----|
| | Base | New | HM | | | | | New | HM | New | | | |
| CLIP† (Zhou et al. 2022b) | 69.34 | 74.22 | 71.70 | 72.43 | 68.14 | 70.22 | 72.08 | 77.80 | 74.83 | 91.17 | 97.26 | 94.12 | |
| CoOp† (Zhou et al. 2022b) | 82.69 | 63.22 | 71.66 | 76.47 | 67.88 | 71.92 | 97.60 | 59.67 | 74.06 | 93.67 | 95.29 | 94.47 | |
| CoCoOp† (Zhou et al. 2022a) | 80.47 | 71.69 | 75.83 | 75.98 | 70.43 | 73.10 | 94.87 | 71.75 | 81.71 | 95.20 | 97.69 | 96.43 | |
| KgCoOp† (Yao, Zhang, and Xu 2023) | 80.73 | 73.60 | 77.00 | 75.83 | 69.96 | 72.78 | 95.00 | 74.73 | 83.65 | 94.65 | 97.76 | 96.18 | |
| TCP† (Yao, Zhang, and Xu 2024) | 84.13 | 75.36 | 79.51 | 77.27 | 69.87 | 73.38 | 97.73 | 75.57 | 85.23 | 94.67 | 97.20 | 95.92 | |
| CLIPood† (Shu et al. 2023) | 83.90 | 74.50 | 78.90 | 77.50 | 70.30 | 73.70 | 93.50 | 74.50 | 82.90 | <u>95.70</u> | 96.40 | 96.00 | |
| MaPLE† (Khattak et al. 2023a) | 82.28 | 75.14 | 78.55 | 76.66 | 70.54 | 73.47 | 95.92 | 72.46 | 82.56 | 95.43 | 97.76 | 96.58 | |
| PromptSRC† (Khattak et al. 2023b) | 84.26 | 76.10 | 79.97 | 77.60 | 70.73 | 74.01 | <u>98.07</u> | 76.50 | <u>85.95</u> | 95.33 | 97.30 | 96.30 | |
| CoPrompt† (Roy and Etemad 2024) | 84.00 | 77.23 | 80.48 | <u>77.67</u> | 71.27 | 74.33 | 97.27 | 76.60 | 85.71 | 95.67 | 98.10 | 96.87 | |
| CoPrompt* (Roy and Etemad 2024) | 82.67 | 75.72 | 79.04 | 76.67 | <u>71.15</u> | 73.81 | 96.64 | 75.11 | 84.52 | 95.30 | 97.16 | 96.22 | |
| ProLoG (ours) | 84.77 | 76.89 | 80.64 | 78.25 | 70.51 | 74.18 | 98.10 | 77.48 | 86.58 | 95.92 | 98.01 | 96.95 | |

Table 3: Results in the base-to-new generalization setting using ViT-B/16. † indicates results reported in the original papers, while * denotes results reproduced using the official code.

D. Additional Experimental Results

Results in the base-to-new generalization setting using ViT-B/16

In Table 3 using ViT-B/16, ProLoG again achieves the best average HM score, showing over 0.5% improvement in base-class accuracy over all baselines. While it performs slightly below the original CoPrompt (Roy and Etemad 2024) on new classes, it achieves a higher HM score while using only 7% of CoPrompt’s additional parameters (see Table 11). In our reproduced setting, ProLoG also outperforms the reproduced CoPrompt* with a 1.6% gain in HM, further validating its superiority. Moreover, ProLoG demonstrates substantial gains over CoPrompt in more challenging settings, including cross-dataset generalization (Table 2 in the main paper) and robustness to spurious correlations (Table 3 in the main paper).

Results on the ImageNet Dataset in the Cross-Dataset Generalization Setting

In this subsection, we present additional results on the ImageNet dataset in the cross-dataset generalization setting. When ImageNet is used as the base dataset, all target datasets fall under Scenario II based on the task similarity score, given the diverse and comprehensive range of object classes in ImageNet. As shown in Table 4 using

ViT-B/16, the results are consistent with previous experiments, further demonstrating the effectiveness of our method.

Results in the Domain Generalization Setting

In this setup, the model is trained on ImageNet and evaluated on four ImageNet variants that share the same classes but differ in image distributions. In this setting, Scenario I is applied to all target tasks. As shown in Table 5 using ViT-B/16, ProLoG performs on par with PromptSRC, with only a marginal 0.01% gap in the average score, which shows the best performance.

Additional Ablation Study on Hyperparameters

Effect of δ_2 in the inference strategy. In our inference strategy, we introduce the hyperparameter δ_2 , which is set to 0.85 for all experiments, to distinguish between Scenario II and Scenario III for a given target task. To evaluate the impact of varying δ_2 on performance, we conduct additional experiments in the cross-dataset generalization setting using the UCF101 dataset as the base task. The results in Table 6 demonstrate that ProLoG consistently outperforms competitive baselines (PromptSRC, MaPLE, and CoPrompt) across a wide range of δ_2 , confirming the robustness of our method to hyperparameter sensitivity.

Effect of rank in the LoRA module. Table 7a presents an

| Scenario | Base | | Target | | | | | | | | | Avg. |
|---------------|------------|---------------|--------------|--------------|---------------|--------------|----------------|--------------|--------------|--------------|--------------|--------------|
| | ImNet I | Caltech II | Pets II | Cars II | Flowers II | Food II | Aircraft II | SUN II | DTD II | EuSAT II | UCF II | |
| MaPLE | 70.72 | 93.53 | 90.49 | 65.57 | 72.23 | 86.20 | 24.74 | 67.01 | 46.49 | 48.06 | 68.69 | 66.30 |
| PromptsSRC | 71.27 | 93.60 | 90.25 | 65.70 | 70.25 | 86.15 | 23.90 | 67.10 | 46.87 | 45.50 | 68.75 | 65.81 |
| ProLoG | 71.06 | 94.35 | 90.44 | 66.21 | 72.41 | 86.46 | 23.69 | 67.51 | 45.39 | 48.16 | 69.67 | 66.43 |

Table 4: Results in a cross-dataset generalization setting using the ImageNet dataset as the base task.

| Scenario | Base | | Target | | | | Avg. |
|---------------|------------|--------------|--------------|--------------|--------------|--------------|------|
| | ImNet I | ImNetV2 I | ImNetS I | ImNetA I | ImNetR I | | |
| CLIP | 66.73 | 60.83 | 46.15 | 47.77 | 73.96 | 57.18 | |
| MaPLE | 70.72 | 64.07 | 49.15 | 50.90 | 76.98 | 60.28 | |
| PromptsSRC | 71.27 | 64.35 | 49.55 | 50.90 | 77.80 | 60.65 | |
| CoPrompt | 70.80 | 64.25 | 49.43 | 50.50 | <u>77.51</u> | 60.42 | |
| ProLoG | 71.06 | 64.36 | 49.64 | 51.05 | 77.50 | <u>60.64</u> | |

Table 5: Results in the domain generalization setting.

ablation study on the effect of varying the ranks (r) of LoRA on the average performance across 11 datasets in the base-to-new generalization setting without applying the inference strategy. The results indicate that increasing r improves the HM score up to $r = 5$, where the highest HM score (80.42) is achieved. However, further increasing r to $r = 8$ results in a slight performance drop. While $r = 5$ achieves the best overall performance, we chose $r = 4$ in the main paper to maintain a better balance between performance and parameter efficiency.

Effect of λ in the training strategy. In our main paper, we set λ (the weight for the proposed regularization loss in our training strategy) to 20. To examine its effect, we evaluate performance by varying λ . Table 7b presents the results on the average performance across 11 datasets in the base-to-new generalization setting without applying the inference strategy. The results show that increasing λ from 10 to 20 improves performance on both base and novel classes. Further increasing λ to 30 enhances performance on novel classes but slightly reduces performance on base classes. Therefore, we chose to set λ to 20, achieving the best performance on the HM score.

Effect of LoRA insertion position. In our main paper, LoRA modules are inserted into all transformer layers, except the first layer, in the image and text encoders of CLIP. To evaluate the impact of their placement on performance, we conduct additional experiments by inserting them at various positions within the transformer layers. Table 8 shows average performance in the base-to-novel generalization setup. The results in Table 8 indicate that applying LoRA only to the earlier layers (1st-6th) or later layers (7th-12th) results in generally lower performance. While applying LoRA to all layers achieves the best performance on base classes, it shows limited performance on new classes. Our chosen configuration (2nd-12th layers) achieves the highest HM score while ensuring effective generalization across tasks by keeping the first layer of each transformer fixed.

Impact of Text Augmentation Design

In our experiments, we adopt the text augmentation strategy as in CoPrompt. To examine how different text augmentation choices affect performance, we compare our method with two variants: (1) using 50% of the augmented texts per class, and (2) using handcrafted prompts (Yao, Zhang, and Xu 2023). As shown in Table 9, more diverse prompts, generated by LLMs to highlight class-relevant attributes, lead to improved performance. Moreover, ProLoG remains robust even with simpler augmentations (e.g., handcrafted), consistently outperforming prior works such as PromptSRC.

Impact of Image Augmentation Design

To evaluate the effect of ProLoG with different image augmentation strategies, we compare ProLoG using our proposed augmentation, random image augmentation (random resized crop and horizontal flip), and no augmentation. Table 10 reports the results on the FGVCAircraft and UCF101 datasets in the base-to-new generalization setting. As shown, ProLoG with our augmentation achieves the highest HM scores of 40.39 on FGVCAircraft and 83.27 on UCF101, confirming the effectiveness of our augmentation strategy.

Additional Experiments Using a Different Model Backbone

We conduct additional experiments using OpenCLIP (Cherti et al. 2023) as the backbone, a more recent vision-language model than CLIP, which improves robustness and generalization through training on large-scale datasets like LAION. Table 11 shows the results (HM score) on the StandfordCars and EuroSAT datasets in the base-to-new generalization. As shown in the results, ProLoG outperforms baselines even with the different VLM backbone (OpenCLIP), confirming the robustness of ProLoG.

E. Comparison with CoPrompt

We provide a clear comparison between ProLoG and CoPrompt (Roy and Etemad 2024). First, *architecturally*, CoPrompt follows MaPLE by inserting prompts into both the text input and intermediate feature spaces, and generating image prompts from text features via neural networks. In contrast, our hybrid network inserts prompts only into the text input, while applying LoRA within the model, enabling our inference strategy by leveraging their complementary properties. Second, unlike CoPrompt’s simple random image augmentation (such as random resized crop, horizontal flip), ProLoG leverages semantic relationships between text embeddings to augment image features with key class attributes. Finally, at inference, we *adaptively apply prompts and LoRA*

| | Base | | | Target | | | | | | | | |
|--|-------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | UCF | Caltech | Pets | Cars | Flowers | Food | Aircraft | SUN | DTD | EuSAT | ImNet | Avg. |
| MaPLe* | 78.18 | 87.97 | 82.19 | 58.04 | 54.65 | 82.73 | 15.71 | 57.10 | 37.74 | 47.82 | 61.54 | 60.33 |
| PromptsSRC* | 76.20 | 91.22 | 82.49 | 63.88 | 66.41 | 85.30 | 20.70 | 64.18 | 42.65 | 50.75 | 66.63 | 63.42 |
| CoPrompt* | 77.80 | 90.39 | 84.57 | 60.19 | 54.53 | 82.04 | 21.27 | 62.19 | 38.95 | 43.43 | 65.04 | 60.26 |
| ProLoG (w/o inference strategy) | 77.68 | 88.90 | 76.08 | 61.72 | 58.20 | 83.23 | 19.40 | 59.19 | 38.08 | 44.88 | 62.09 | 59.18 |
| ProLoG ($\delta_2 = 0.75$) | 77.68 | 92.30 | 88.03 | 65.39 | 67.32 | 84.56 | 22.44 | 62.56 | 44.29 | 44.67 | 65.69 | 63.73 |
| ProLoG ($\delta_2 = 0.80$) | 77.68 | 92.30 | 88.03 | 65.39 | 67.32 | 84.56 | 22.44 | 62.56 | 44.29 | 44.67 | 65.69 | 63.73 |
| ProLoG ($\delta_2 = 0.85$) | 77.68 | 92.30 | 88.03 | 65.39 | 67.32 | 84.56 | 23.82 | 62.56 | 44.29 | 44.67 | 66.73 | 63.97 |
| ProLoG ($\delta_2 = 0.95$) | 77.68 | 92.90 | 88.03 | 65.39 | 67.32 | 85.41 | 23.82 | 62.56 | 44.50 | 41.30 | 66.73 | 63.80 |

Table 6: Effect of δ_2 in our inference strategy.

| Average on 11 datasets | | | |
|---------------------------|--------------|--------------|--------------|
| | Base | New | HM |
| ProLoG ($r = 3$) | 84.80 | 76.22 | 80.28 |
| ProLoG ($r = 4$) | 84.77 | 76.32 | 80.33 |
| ProLoG ($r = 5$) | 84.79 | 76.48 | 80.42 |
| ProLoG ($r = 8$) | 84.33 | 75.08 | 79.44 |

(a) Effect of rank in LoRA modules.

| Average on 11 datasets | | | |
|----------------------------------|--------------|--------------|--------------|
| | Base | New | HM |
| ProLoG ($\lambda = 10$) | 84.63 | 75.25 | 79.66 |
| ProLoG ($\lambda = 30$) | 84.77 | 76.32 | 80.33 |
| ProLoG ($\lambda = 40$) | 84.56 | 76.36 | 80.25 |

(b) Effect of λ in ProLoG w/o the inference strategy.

Table 7: Ablation studies on the hyperparameters (LoRA rank and λ).

| Average on 11 datasets | | | |
|---------------------------------|--------------|--------------|--------------|
| | Base | New | HM |
| ProLoG (1st-6th layers) | 83.44 | 74.79 | 78.88 |
| ProLoG (7th-12th layers) | 84.31 | 76.09 | 79.99 |
| ProLoG (1st-12th layers) | 85.03 | 75.97 | 80.24 |
| ProLoG (2nd-12th layers) | 84.77 | 76.89 | 80.64 |

Table 8: Effect of LoRA insertion position.

for ID/OOD handling, which is a novel approach not explored in CoPrompt.

F. Complexity Analysis

Parameter complexity of ProLoG.

Table 12 compares the total and additional parameters of baselines and ProLoG during inference on ViT-B/16 CLIP. ProLoG requires only 0.34M additional parameters in Scenarios I and II (fewer than MaPLe and CoPrompt) due to the introduced prompts and LoRA. In Scenario III, where the original CLIP is used, no additional parameters are required.

Computational Complexity of ProLoG

Table 13 compares the GFLOP of ProLoG with baselines like MaPLe, PromptSRC, and CoPrompt during inference on the Caltech101 dataset. In the image encoder, ProLoG requires slightly higher computational costs (approximately

| Method | Average HM |
|----------------------|--------------|
| ProLoG | 80.64 |
| ProLoG (50%) | 80.45 |
| ProLoG (Handcrafted) | 80.33 |
| PromptSRC | 79.97 |

Table 9: Comparison of average harmonic mean (HM) across different text augmentation strategies.

3.3 GFLOP) under Scenarios I and II due to the inclusion of prompts and LoRA modules, which enhance generalization performance. In Scenario III, ProLoG requires fewer computational costs as the model utilizes only the original pre-trained CLIP without additional modifications.

In the text encoder, the costs for Scenarios I and III remain at 146 GFLOP, aligning with the baselines. In Scenario II, the cost doubles due to the ensemble prediction requiring additional text embeddings to improve performance on OOD samples. However, note that *these text features can be precomputed once the target classes are defined*, and they remain fixed during inference while only the input images vary. Since the additional operations for the ensemble prediction in Scenario II involve only inner products and vector averaging—each requiring less than 0.00001 GFLOP—the extra computational cost is negligible during inference. This ensures that our method retains both computational efficiency and strong performance.

G. Details on Task similarity score (S_{TS})

We provide detailed values of the task similarity score computed for our inference strategy in the base-to-new generalization and cross-dataset generalization settings. The task similarity score plays a key role in categorizing target tasks into different scenarios, enabling the adaptive application of the learned prompts and LoRA. Table 14 presents the actual values of the task similarity score (S_{TS}) computed for each task in the base-to-new generalization. Since the base and novel classes are originally split from a single dataset, the similarity scores for new classes generally exhibit high values in this setting.

Table 15 shows the task similarity score (S_{TS}) values for each task in the cross-dataset generalization setup. In this setting, as the base and target datasets are entirely different, task-specific knowledge learned from the base task is generally not shared with the target tasks. For instance, due to the specificity of the UCF101 dataset, which focuses on human actions, target datasets such as OxfordPets, StanfordCars,

| | FGVBAircraft | | | UCF101 | | |
|--------------------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | Base | New | HM | Base | New | HM |
| ProLoG (No Img. Aug.) | 44.12 | 34.31 | 38.60 | 86.97 | 77.88 | 82.17 |
| ProLoG (Rand. Img. Aug.) | 43.58 | 35.15 | 38.91 | 87.28 | 77.99 | 82.37 |
| ProLoG | 43.43 | 37.76 | 40.39 | 87.09 | 79.77 | 83.27 |

Table 10: Comparison of image augmentation strategies in the base-to-new generalization setting.

| Method | StanfordCars | EuroSAT |
|-------------------|--------------|--------------|
| CLIP | 68.65 | 61.76 |
| OpenCLIP | 84.73 | 84.46 |
| MaPLe (OpenCLIP) | 88.59 | 82.93 |
| ProLoG (OpenCLIP) | 89.55 | 86.11 |

Table 11: Comparison on StanfordCars and EuroSAT using OpenCLIP backbone.

| | Total | Additional param. | HM |
|-----------------|---|-------------------|--------------|
| CLIP (ViT-B/16) | 149.62M | - | 71.70 |
| MaPLe | 153.17M | 3.55M | 78.55 |
| CoPrompt* | 154.62M | 4.74M | 79.04 |
| ProLoG | S_I and S_{II} : 150.22M S_{III} : 149.62M | 0.34M - | 80.64 |

Table 12: Comparison of total and additional parameters.

and Flowers102 show relatively low similarity scores.

For each target task, our inference strategy, guided by predefined thresholds δ_1 and δ_2 , is applied to handle diverse practical scenarios, including both ID and OOD data. Note that even if scenarios are not ideally categorized by δ_1 and δ_2 , it does not significantly impact performance due to the strong generalization capabilities of prompts and LoRA trained with our regularization loss, as well as the inherent robustness of the original CLIP.

H. Detailed Description on Image Feature Augmentation

Recent works (Ramesh et al. 2022; Vidit, Engilberge, and Salzmann 2023) have shown that directional information in the text feature space can guide image features semantically. (Ramesh et al. 2022) demonstrates that moving along a textual direction can semantically alter the generated image, while (Vidit, Engilberge, and Salzmann 2023) extracts style information via algebraic operations on text features and injects it into image features by adding the resulting style vector to them, to improve DG performance.

Building on this insight, we strategically perform an algebraic operation between the original text feature and the augmented text feature with key attributes, i.e., $\hat{\mathbf{z}}_T^{ep} - \mathbf{z}_T^{ep}$, and add this vector to the image embedding. This results in an augmented image feature that reflects class-specific attributes. The results in Table 10 demonstrate the effectiveness of our image augmentation method compared to simple random image augmentation and no augmentation.

| | Image Encoder | Text Encoder (precomputed) | Total (actual inference) | HM |
|-----------|--|-------------------------------|-----------------------------|--------------|
| PromptSRC | 17.09 | 146 | 17.09 | 79.97 |
| MaPLe | 17.04 | 146 | 17.04 | 78.55 |
| CoPrompt* | 17.30 | 146 | 17.30 | 79.04 |
| ProLoG | S_I : 20.64 S_{II} : 20.64 S_{III} : 16.75 | 146 146×2 146 | 20.64 20.64 16.75 | 80.64 |

Table 13: Comparison of GFLOP. Note that GFLOP in the text encoder can be *precomputed* and prepared before inference. Therefore, **the extra computational cost for the ensemble prediction in Scenario II is negligible during inference.**

I. Detailed Results of Table 1 in the Main Paper

In this section, we present the detailed results corresponding to Table 1 in the main paper across 11 datasets. Tables 16 and 17 provide the detailed results of Table 1 (in the main paper) and Table 3 (Appendix), using ViT-B/32 and ViT-B/16, respectively. The results include the mean and 95% confidence intervals of the values reported in the main paper, where the confidence intervals are computed based on the standard deviation over three random runs. These results offer deeper insights into the reliability and consistency of the model’s performance across diverse datasets.

J. Projection Method for Computing Task Similarity Score

Let $(Z_b \in \mathbb{R}^{d \times N_b})$ be base class text embeddings and $(z_t \in \mathbb{R}^d)$ be a target class text embedding. We project z_t onto the subspace spanned by Z_b as:

$$\text{Proj}_{Z_b}(z_t) = Z_b(Z_b^\top Z_b)^+ Z_b^\top z_t, \quad (1)$$

where $(\cdot)^+$ is the Moore-Penrose pseudoinverse.

K. Additional Related Work

Full fine-tuning and linear probing (Kumar et al. 2022) are typical adaptation approaches that directly update the parameters of pre-trained models. However, simply fine-tuning VLMs or applying linear-probing for specific tasks results in a significant drop in generalization performance due to overfitting. To address this, (Shu et al. 2023) proposed fine-tuning VLMs with a new training loss, margin metric softmax, which considers semantic relations between classes to retain the original knowledge of CLIP. However, fine-tuning the entire network not only requires updating a substantial number of parameters but also yields low OOD performance, as the entire network tends to overfit to downstream tasks.

| | ImNet | | Caltech | | Pets | | Cars | | Flowers | | Food | |
|----------|-------|--------|---------|--------|------|--------|------|--------|---------|--------|------|--------|
| | Base | New | Base | New | Base | New | Base | New | Base | New | Base | New |
| S_{TS} | 1 | 0.9816 | 1 | 0.8798 | 1 | 0.8783 | 1 | 0.9457 | 1 | 0.8713 | 1 | 0.9041 |

| | Aircraft | | SUN | | DTD | | EuSAT | | UCF | |
|----------|----------|--------|------|--------|------|--------|-------|--------|------|--------|
| | Base | New | Base | New | Base | New | Base | New | Base | New |
| S_{TS} | 1 | 0.9161 | 1 | 0.9698 | 1 | 0.9117 | 1 | 0.9276 | 1 | 0.8993 |

Table 14: Task similarity score (S_{TS}) between the base and target tasks in the base-to-new generalization setting.

| | Base | | | | | | Target | | | | |
|----------|--------|---------|--------|--------|---------|--------|----------|--------|--------|--------|--------|
| | UCF101 | Caltech | Pets | Cars | Flowers | Food | Aircraft | SUN | DTD | EuSAT | ImNet |
| S_{TS} | 1 | 0.8848 | 0.7343 | 0.6865 | 0.7488 | 0.8653 | 0.8348 | 0.9003 | 0.9139 | 0.9268 | 0.8471 |

Table 15: Task similarity score (S_{TS}) between the base and target tasks in the cross-dataset generalization setting using the UCF101 dataset as the base task.

L. Limitations and Future Work

While our method demonstrates strong performance on image classification tasks, it has not been extensively explored for multimodal generative tasks such as VQA. Nevertheless, our prompt-LoRA hybrid approach is generally applicable to transformer-based architectures, as both learnable prompts and LoRA modules can be directly inserted. In addition, our text-guided image augmentation method enables the generation of meaningful image feature augmentations corresponding to diverse text captions, potentially enhancing generalization. Regarding the inference strategy, a potential extension could involve defining a task similarity score based on the similarity between the text features of image captions from the base and target tasks. This score could be used to adaptively apply prompts and LoRA modules, enabling effective handling of both ID and OOD inputs. We leave this direction for future work, as it may offer a promising path toward extending our framework to more complex multimodal tasks.

M. Broader Impact

This work proposes ProLoG, a hybrid adaptation method for improving OOD generalization in vision-language models. By enhancing adaptability to specific domains, our approach can benefit applications like medical diagnostics, disaster response, and accessibility technologies. As our method focuses on improving generalization without introducing task-specific biases, it does not inherently pose risks of misuse or ethical concerns, provided it is deployed in responsibly designed systems.

References

- Cherti, M.; Beaumont, R.; Wightman, R.; Wortsman, M.; Ilharco, G.; Gordon, C.; Schuhmann, C.; Schmidt, L.; and Jitsev, J. 2023. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2818–2829.
- Khattak, M. U.; Rasheed, H.; Maaz, M.; Khan, S.; and Khan, F. S. 2023a. Maple: Multi-modal prompt learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19113–19122.
- Khattak, M. U.; Wasim, S. T.; Naseer, M.; Khan, S.; Yang, M.-H.; and Khan, F. S. 2023b. Self-regulating prompts: Foundational model adaptation without forgetting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 15190–15200.
- Kumar, A.; Raghunathan, A.; Jones, R.; Ma, T.; and Liang, P. 2022. Fine-tuning can distort pretrained features and underperform out-of-distribution. *arXiv preprint arXiv:2202.10054*.
- Ramesh, A.; Dhariwal, P.; Nichol, A.; Chu, C.; and Chen, M. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2): 3.
- Roy, S.; and Etemad, A. 2024. Consistency-guided Prompt Learning for Vision-Language Models. In *The Twelfth International Conference on Learning Representations*.
- Shu, Y.; Guo, X.; Wu, J.; Wang, X.; Wang, J.; and Long, M. 2023. Clipood: Generalizing clip to out-of-distributions. In *International Conference on Machine Learning*, 31716–31731. PMLR.
- Vidit, V.; Engilberge, M.; and Salzmann, M. 2023. Clip the gap: A single domain generalization approach for object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 3219–3229.
- Yao, H.; Zhang, R.; and Xu, C. 2023. Visual-language prompt tuning with knowledge-guided context optimization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 6757–6767.
- Yao, H.; Zhang, R.; and Xu, C. 2024. TCP: Textual-based Class-aware Prompt tuning for Visual-Language Model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 23438–23448.
- Zanella, M.; and Ben Ayed, I. 2024. Low-Rank Few-Shot Adaptation of Vision-Language Models. In *Proceedings of*

| | Average on 11 datasets | | | ImageNet | | | Caltech101 | | | OxfordPets | | |
|-----------------------------------|------------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | Base | New | HM | Base | New | HM | Base | New | HM | Base | New | HM |
| CLIP* (Zhou et al. 2022b) | 66.95 | 71.54 | 69.17 | 67.46 | 64.04 | 65.71 | 93.74 | 94.00 | 93.87 | 90.75 | 96.70 | 93.63 |
| CoOp* (Zhou et al. 2022b) | 79.33 | 63.48 | 70.52 | 71.05 | 62.10 | 66.27 | 97.71 | 91.71 | 94.60 | 92.53 | 95.47 | 93.97 |
| CoCoOp* (Zhou et al. 2022a) | 77.25 | 69.35 | 73.09 | 71.21 | 66.80 | 68.93 | 96.64 | 93.56 | 95.08 | 93.09 | 96.70 | 94.86 |
| KgCoOp (Yao, Zhang, and Xu 2023) | 77.31 | 70.74 | 73.88 | 70.64 | 65.62 | 68.04 | 97.22 | 94.76 | 95.97 | 93.67 | 96.42 | 95.03 |
| TCP* (Yao, Zhang, and Xu 2024) | 80.69 | 71.84 | 76.01 | 71.79 | 65.79 | 68.66 | 97.55 | 94.87 | 96.19 | 92.88 | 96.67 | 94.74 |
| CLIPoold* (Shu et al. 2023) | 80.34 | 71.78 | 75.82 | 71.91 | 65.59 | 68.60 | 97.55 | 92.47 | 94.94 | 94.47 | 95.53 | 95.00 |
| MaPLe* (Khattak et al. 2023a) | 78.99 | 70.69 | 74.61 | 71.50 | 65.63 | 68.44 | 97.00 | 92.91 | 94.91 | 93.44 | 93.54 | 93.46 |
| PromptSRC* (Khattak et al. 2023b) | 81.28 | 71.29 | 75.95 | 72.58 | 65.73 | 68.99 | 97.52 | 94.71 | 96.09 | 93.78 | 96.20 | 94.97 |
| CoPrompt* (Roy and Etemad 2024) | 79.15 | 70.14 | 74.38 | 71.21 | 66.45 | 68.75 | 97.49 | 94.43 | 95.93 | 93.54 | 94.66 | 94.08 |
| ProLoG (ours) | 81.05 | 73.05 | 76.84 | 73.02 | 66.48 | 69.60 | 97.23 | 94.98 | 96.09 | 94.77 | 97.40 | 96.06 |
| 95% confidence interval | ± 0.08 | ± 0.27 | ± 0.11 | ± 0.29 | ± 0.49 | ± 0.40 | ± 0.06 | ± 0.43 | ± 0.25 | ± 0.02 | ± 0.25 | ± 0.13 |
| | StandfordCars | | | Flowers102 | | | Food101 | | | FGVCAircraft | | |
| | Base | New | HM | Base | New | HM | Base | New | HM | Base | New | HM |
| CLIP* (Zhou et al. 2022b) | 61.07 | 69.77 | 65.13 | 72.27 | 74.18 | 73.21 | 85.14 | 87.13 | 86.12 | 21.43 | 29.51 | 24.83 |
| CoOp* (Zhou et al. 2022b) | 72.89 | 57.90 | 64.54 | 94.83 | 56.56 | 70.85 | 85.44 | 83.02 | 84.21 | 30.76 | 22.59 | 25.80 |
| CoCoOp* (Zhou et al. 2022a) | 67.29 | 67.02 | 67.15 | 91.07 | 68.65 | 78.29 | 86.58 | 87.47 | 87.02 | 26.89 | 26.99 | 26.94 |
| KgCoOp* (Yao, Zhang, and Xu 2023) | 67.62 | 69.87 | 68.73 | 91.07 | 70.35 | 79.38 | 85.94 | 87.44 | 86.68 | 29.17 | 21.60 | 24.82 |
| TCP* (Yao, Zhang, and Xu 2024) | 74.61 | 69.06 | 71.73 | 95.92 | 71.74 | 82.09 | 86.15 | 87.43 | 86.78 | 33.61 | 29.87 | 31.63 |
| CLIPoold* (Shu et al. 2023) | 71.89 | 67.96 | 69.87 | 90.79 | 71.92 | 80.26 | 86.80 | 87.29 | 87.04 | 35.83 | 31.43 | 33.49 |
| MaPLe* (Khattak et al. 2023a) | 67.86 | 68.82 | 68.33 | 93.21 | 72.16 | 81.34 | 86.41 | 87.69 | 87.05 | 28.93 | 26.72 | 27.78 |
| PromptSRC* (Khattak et al. 2023b) | 74.04 | 68.77 | 71.30 | 95.73 | 71.85 | 82.07 | 86.40 | 87.34 | 86.87 | 34.18 | 27.74 | 30.57 |
| CoPrompt* (Roy and Etemad 2024) | 65.29 | 64.46 | 64.87 | 93.21 | 69.12 | 79.37 | 86.04 | 87.16 | 86.59 | 28.27 | 28.52 | 28.38 |
| ProLoG (ours) | 74.44 | 70.99 | 72.67 | 96.02 | 71.88 | 82.21 | 86.72 | 87.69 | 87.20 | 31.90 | 32.00 | 31.95 |
| 95% confidence interval | ± 0.47 | ± 0.26 | ± 0.09 | ± 0.09 | ± 0.25 | ± 0.19 | ± 0.06 | ± 0.04 | ± 0.01 | ± 0.26 | ± 0.03 | ± 0.12 |
| | SUN397 | | | DTD | | | EuroSAT | | | UCF101 | | |
| | Base | New | HM | Base | New | HM | Base | New | HM | Base | New | HM |
| CLIP* (Zhou et al. 2022b) | 69.75 | 73.05 | 71.36 | 53.82 | 58.45 | 56.04 | 53.69 | 68.82 | 60.32 | 67.37 | 71.34 | 69.30 |
| CoOp* (Zhou et al. 2022b) | 79.17 | 68.31 | 73.33 | 76.10 | 46.26 | 57.53 | 88.52 | 55.13 | 67.32 | 83.61 | 59.25 | 69.30 |
| CoCoOp* (Zhou et al. 2022a) | 77.65 | 76.14 | 76.89 | 72.45 | 51.93 | 60.50 | 85.62 | 56.18 | 67.84 | 81.23 | 71.44 | 76.02 |
| KgCoOp* (Yao, Zhang, and Xu 2023) | 77.92 | 74.84 | 76.35 | 76.16 | 55.68 | 64.33 | 79.19 | 68.97 | 73.73 | 81.85 | 72.63 | 76.96 |
| TCP* (Yao, Zhang, and Xu 2024) | 80.55 | 76.02 | 78.22 | 78.07 | 52.24 | 62.59 | 91.60 | 74.32 | 82.06 | 84.90 | 72.26 | 78.07 |
| CLIPoold* (Shu et al. 2023) | 79.15 | 75.57 | 77.32 | 74.19 | 57.61 | 64.86 | 97.67 | 69.28 | 81.06 | 83.50 | 74.91 | 78.97 |
| MaPLe* (Khattak et al. 2023a) | 78.88 | 76.12 | 77.47 | 76.91 | 52.48 | 62.38 | 93.42 | 70.91 | 80.62 | 81.36 | 70.58 | 75.57 |
| PromptSRC* (Khattak et al. 2023b) | 80.67 | 76.52 | 78.53 | 79.92 | 57.31 | 66.75 | 94.16 | 67.09 | 78.19 | 85.09 | 70.90 | 77.35 |
| CoPrompt* (Roy and Etemad 2024) | 80.29 | 71.94 | 75.77 | 78.53 | 54.47 | 64.32 | 93.26 | 68.86 | 78.43 | 83.53 | 71.53 | 77.07 |
| ProLoG (ours) | 81.04 | 77.00 | 78.96 | 78.59 | 60.02 | 68.05 | 93.82 | 70.70 | 80.60 | 84.00 | 74.45 | 78.93 |
| 95% confidence interval | ± 0.07 | ± 0.11 | ± 0.02 | ± 0.68 | ± 1.54 | ± 1.24 | ± 0.90 | ± 2.21 | ± 1.07 | ± 0.02 | ± 0.02 | ± 0.02 |

Table 16: Detailed results of Table 1 in the main paper on 11 datasets using ViT-B/32. The reported results indicate (mean \pm 95% confidence interval) over 3 random runs. * denotes results reproduced using the official code.

the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 1593–1603.

Zhou, K.; Yang, J.; Loy, C. C.; and Liu, Z. 2022a. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 16816–16825.

Zhou, K.; Yang, J.; Loy, C. C.; and Liu, Z. 2022b. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9): 2337–2348.

| | Average on 11 datasets | | | ImageNet | | | Caltech101 | | | OxfordPets | | |
|-----------------------------------|------------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | Base | New | HM | Base | New | HM | Base | New | HM | Base | New | HM |
| CLIP† (Zhou et al. 2022b) | 69.34 | 74.22 | 71.70 | 72.43 | 68.14 | 70.22 | 96.84 | 94.00 | 95.40 | 91.17 | 97.26 | 94.12 |
| CoOp† (Zhou et al. 2022b) | 82.69 | 63.22 | 71.66 | 76.47 | 67.88 | 71.92 | 98.00 | 89.81 | 93.73 | 93.67 | 95.29 | 94.47 |
| CoCoOp† (Zhou et al. 2022a) | 80.47 | 71.69 | 75.83 | 75.98 | 70.43 | 73.10 | 97.96 | 93.81 | 95.84 | 95.20 | 97.69 | 96.43 |
| KgCoOp† (Yao, Zhang, and Xu 2023) | 80.73 | 73.60 | 77.00 | 75.83 | 69.96 | 72.78 | 97.72 | 94.39 | 96.03 | 94.65 | 97.76 | 96.18 |
| TCP† (Yao, Zhang, and Xu 2024) | 84.13 | 75.36 | 79.51 | 77.27 | 69.87 | 73.38 | 98.23 | 94.67 | 96.42 | 94.67 | 97.20 | 95.92 |
| CLIPoP† (Shu et al. 2023) | 83.90 | 74.50 | 78.90 | 77.50 | 70.30 | 73.70 | 98.70 | 94.60 | 96.60 | 95.70 | 96.40 | 96.00 |
| MaPLe† (Khattak et al. 2023a) | 82.28 | 75.14 | 78.55 | 76.66 | 70.54 | 73.47 | 97.74 | 94.36 | 96.02 | 95.43 | 97.76 | 96.58 |
| PromptSRC† (Khattak et al. 2023b) | 84.26 | 76.10 | 79.97 | 77.60 | 70.73 | 74.01 | 98.10 | 94.03 | 96.02 | 95.33 | 97.30 | 96.30 |
| CoPrompt† (Roy and Etemad 2024) | 84.00 | 77.23 | 80.48 | 77.67 | 71.27 | 74.33 | 98.27 | 94.90 | 96.55 | 95.67 | 98.10 | 96.87 |
| CoPrompt* (Roy and Etemad 2024) | 82.67 | 75.72 | 79.04 | 76.67 | 71.15 | 73.81 | 98.68 | 95.00 | 96.80 | 95.30 | 97.16 | 96.22 |
| ProLoG (ours) | 84.77 | 76.89 | 80.64 | 78.25 | 70.51 | 74.18 | 98.18 | 94.65 | 96.38 | 95.92 | 98.01 | 96.95 |
| 95% confidence interval | ± 0.19 | ± 0.36 | ± 0.16 | ± 0.19 | ± 0.07 | ± 0.07 | ± 0.17 | ± 0.39 | ± 0.22 | ± 0.17 | ± 0.08 | ± 0.10 |
| | StandfordCars | | | Flowers102 | | | Food101 | | | FGVCAircraft | | |
| | Base | New | HM | Base | New | HM | Base | New | HM | Base | New | HM |
| CLIP† (Zhou et al. 2022b) | 63.37 | 74.89 | 68.65 | 72.08 | 77.80 | 74.83 | 90.10 | 91.22 | 90.66 | 27.19 | 36.29 | 31.09 |
| CoOp† (Zhou et al. 2022b) | 78.12 | 60.40 | 68.13 | 97.60 | 59.67 | 74.06 | 88.33 | 82.26 | 85.19 | 40.44 | 22.30 | 28.75 |
| CoCoOp† (Zhou et al. 2022a) | 70.49 | 73.59 | 72.01 | 94.87 | 71.75 | 81.71 | 90.70 | 91.29 | 90.77 | 33.41 | 23.71 | 27.74 |
| KgCoOp† (Yao, Zhang, and Xu 2023) | 71.76 | 75.04 | 73.36 | 95.00 | 74.73 | 83.65 | 90.50 | 91.70 | 91.09 | 36.21 | 33.55 | 34.83 |
| TCP† (Yao, Zhang, and Xu 2024) | 80.80 | 74.13 | 77.32 | 97.73 | 75.57 | 85.23 | 90.57 | 91.37 | 90.97 | 41.97 | 34.43 | 37.83 |
| CLIPoP† (Shu et al. 2023) | 78.60 | 73.50 | 75.90 | 93.50 | 74.50 | 82.90 | 90.70 | 91.70 | 91.20 | 43.30 | 37.20 | 40.00 |
| MaPLe† (Khattak et al. 2023a) | 72.94 | 74.00 | 73.47 | 95.92 | 72.46 | 82.56 | 90.71 | 92.05 | 91.38 | 37.44 | 35.61 | 36.50 |
| PromptSRC† (Khattak et al. 2023b) | 78.27 | 74.97 | 76.58 | 98.07 | 76.50 | 85.95 | 90.67 | 91.53 | 91.10 | 42.73 | 37.87 | 40.15 |
| CoPrompt† (Roy and Etemad 2024) | 76.97 | 74.40 | 75.66 | 97.27 | 76.60 | 85.71 | 90.73 | 92.07 | 91.40 | 40.20 | 39.33 | 39.76 |
| CoPrompt* (Roy and Etemad 2024) | 73.21 | 70.48 | 71.82 | 96.64 | 75.11 | 84.52 | 90.24 | 91.62 | 90.92 | 37.13 | 36.35 | 36.71 |
| ProLoG (ours) | 80.78 | 73.77 | 77.12 | 98.10 | 77.48 | 86.58 | 90.68 | 91.86 | 91.27 | 43.43 | 37.76 | 40.39 |
| 95% confidence interval | ± 0.32 | ± 0.15 | ± 0.11 | ± 0.39 | ± 0.20 | ± 0.15 | ± 0.17 | ± 0.05 | ± 0.08 | ± 0.61 | ± 0.85 | ± 0.44 |
| | SUN397 | | | DTD | | | EuroSAT | | | UCF101 | | |
| | Base | New | HM | Base | New | HM | Base | New | HM | Base | New | HM |
| CLIP† (Zhou et al. 2022b) | 69.36 | 75.35 | 72.23 | 53.24 | 59.90 | 56.37 | 56.48 | 64.05 | 60.03 | 70.53 | 77.50 | 73.85 |
| CoOp† (Zhou et al. 2022b) | 80.60 | 65.89 | 72.51 | 79.44 | 41.18 | 54.24 | 92.19 | 54.74 | 68.69 | 84.69 | 56.05 | 67.46 |
| CoCoOp† (Zhou et al. 2022a) | 79.74 | 76.86 | 78.27 | 77.01 | 56.00 | 64.85 | 87.49 | 60.04 | 71.21 | 82.33 | 73.45 | 77.64 |
| KgCoOp† (Yao, Zhang, and Xu 2023) | 80.29 | 76.53 | 78.36 | 77.55 | 54.99 | 64.35 | 85.64 | 64.34 | 73.48 | 82.89 | 76.67 | 79.65 |
| TCP† (Yao, Zhang, and Xu 2024) | 82.63 | 78.20 | 80.35 | 82.77 | 58.07 | 68.25 | 91.63 | 74.73 | 82.32 | 87.13 | 80.77 | 83.83 |
| CLIPoP† (Shu et al. 2023) | 81.00 | 79.30 | 80.20 | 80.80 | 58.60 | 67.90 | 97.50 | 64.10 | 77.30 | 85.70 | 79.30 | 82.40 |
| MaPLe† (Khattak et al. 2023a) | 80.82 | 78.70 | 79.75 | 80.36 | 59.18 | 68.16 | 94.07 | 73.23 | 82.35 | 83.00 | 78.66 | 80.77 |
| PromptSRC† (Khattak et al. 2023b) | 82.67 | 78.47 | 80.52 | 83.37 | 62.97 | 71.72 | 92.90 | 73.90 | 82.32 | 87.10 | 78.80 | 82.74 |
| CoPrompt† (Roy and Etemad 2024) | 82.63 | 80.03 | 81.31 | 83.13 | 64.73 | 72.79 | 94.60 | 78.57 | 85.84 | 86.90 | 79.57 | 83.07 |
| CoPrompt* (Roy and Etemad 2024) | 82.48 | 79.31 | 80.86 | 81.31 | 61.24 | 69.83 | 92.39 | 75.61 | 83.02 | 85.33 | 79.87 | 82.51 |
| ProLoG (ours) | 83.02 | 78.76 | 80.83 | 83.20 | 64.40 | 72.59 | 93.87 | 78.86 | 85.67 | 87.09 | 79.77 | 83.27 |
| 95% confidence interval | ± 0.10 | ± 0.17 | ± 0.07 | ± 0.51 | ± 1.13 | ± 0.59 | ± 0.79 | ± 2.80 | ± 1.44 | ± 0.51 | ± 0.86 | ± 0.54 |

Table 17: Detailed results of Table 3 in Appendix on 11 datasets using ViT-B/16. The reported results indicate (mean ± 95% confidence interval) over 3 random runs. † indicates results reported in the original papers, while * denotes results reproduced using the official code.