Appendix: Prompt-aligned Gradient for Prompt Tuning

Beier Zhu[1] Yulei Niu [2*] Yucheng Han[1] Yue Wu[3] Hanwang Zhang[1]
[1]Nanyang Technological University [2]Columbia University [3]Damo Academy, Alibaba Group
beier002@e.ntu.edu.sg, yn.yuleiniu@gmail.com yucheng002@e.ntu.edu.sg
matthew.wy@alibaba-inc.com hanwangzhang@ntu.edu.sg

A. Justification from Generalization Error

We further analyze the generalization error bound of our `ProGrad`. We define the expected risk $\mathcal{R}(\cdot)$ and empirical risk $\hat{\mathcal{R}}(\cdot)$ of a classifier f on domain $\mathcal{D}$ as

$$\mathcal{R}(f) = \mathbb{E}_{(X,Y)\sim\mathcal{D}}[\ell(f(X), Y)],\ \hat{\mathcal{R}}(f) = \frac{1}{N}\sum_{i=1}^{N}\ell(f(X_i), Y_i) \tag{1}$$

where $\ell(f(X), Y)$ denotes the cross-entropy and N is the volume of training data. We are interested in the downstream domain $\mathcal{D}_d$ and pre-trained domain $\mathcal{D}_p$, respectively. [1]

Let $\mathcal{F}$ be a function class, the conventional fine-tune model $\hat{f}_{\text{coop}}$ is trained on $\mathcal{D}_d$ by

$$\hat{f}_{\text{coop}} = \underset{f\in\mathcal{F}}{\operatorname{argmin}}\,\hat{\mathcal{R}}_d(f). \tag{2}$$

The zero-shot CLIP model $\hat{f}_{\text{p}}$ is considered to be trained on $\mathcal{D}_p$ by

$$\hat{f}_{\text{p}} = \underset{f\in\mathcal{F}}{\operatorname{argmin}}\,\hat{\mathcal{R}}_p(f). \tag{3}$$

For the implementation of `ProGrad`, we initialize the model $\hat{f}_{\text{prograd}}$ using the pre-trained model $\hat{f}_{\text{p}}$. We regularize each training step not to increase the KL divergence between the predictions of $\hat{f}_{\text{prograd}}$ and $\hat{f}_{\text{p}}$. In this way, $\hat{f}_{\text{prograd}}$ can keep the optimal value of the pre-trained domain $\mathcal{L}_{\text{kl}}$ when optimizing the empirical risk on the downstream domain. The model $\hat{f}_{\text{prograd}}$ learned by our `ProGrad` can be viewed as optimizing the empirical risk on both domains:

$$\hat{f}_{\text{prograd}} = \underset{f\in\mathcal{F}}{\operatorname{argmin}}\,\hat{\mathcal{R}}_{(d+p)}(f) = \underset{f\in\mathcal{F}}{\operatorname{argmin}}\,\hat{\mathcal{R}}_d(f) + \hat{\mathcal{R}}_p(f). \tag{4}$$

Based on Theorem 4.1 of [4], assuming that the neural network has L layers with parameters matrices $W_1, ..., W_L$, and their Frobenius norm are at most $M_1, ..., M_L$ and the activation functions are 1-Lispschitz continuous, positive-homogeneous, and applied element-wise. The output of the neural network is the softmax function that predicts c classes. Let $\mathcal{F}$ be a function class with the range $[a, b]$. Distribution is such that $\|\mathbf{x}\| \le B$. Let $\mathbf{X}_1^{N_d} = \{\mathbf{x}_n^{(d)}\}_{n=1}^{N_d}$ and $\mathbf{X}_1^{N_p} = \{\mathbf{x}_n^{(p)}\}_{n=1}^{N_p}$ be two set of i.i.d. samples drawn from the downstream domain $\mathcal{D}_d$ and the pre-trained domain $\mathcal{D}_p$. Then for any $\epsilon > 0$, we have with probability at least $1-\epsilon$,

$$\begin{aligned}\mathcal{R}_d(\hat{f}_{\text{prograd}}) \le\ & \hat{\mathcal{R}}_{(d+p)}(\hat{f}_{\text{prograd}}) + \frac{1}{2}\gamma_{\mathcal{F}}(D, P) + \\ & \frac{cB\left(\sqrt{2\log(2)L}+1\right)\prod_{j=1}^{L}M_j}{\sqrt{N_p}} \\ & + \frac{cB\left(\sqrt{2\log(2)L}+1\right)\prod_{j=1}^{L}M_j}{\sqrt{N_d}} \\ & + \frac{3}{2}\sqrt{\frac{(b-a)\ln(4/\epsilon)}{2N_d}} \\ & + \frac{3}{2}\sqrt{\frac{(b-a)\ln(4/\epsilon)}{2N_p}} \\ & + \frac{1}{2}\sqrt{\frac{(b-a)^2\ln(4/\epsilon)}{2}\left(\frac{1}{N_d}+\frac{1}{N_p}\right)},\end{aligned} \tag{5}$$

where $\gamma_{\mathcal{F}}(D, P)$ is the integral probability metric [2] that measures the difference between the distribution of pre-trained domain and the downstream domain. The Eq. (5) shows that the generalization error $\mathcal{R}_d(\hat{f}_{\text{prograd}})$ is bounded by the empirical training risk $\hat{\mathcal{R}}_{(d+p)}(\hat{f}_{\text{prograd}})$, the two domain gap $\gamma_{\mathcal{F}}(D, P)$ and the estimation error that is inversely proportional to number of training samples, *i.e.*, N_d and N_p. The empirical training risk can be minimized to arbitrary small value and the estimation error that related to N_p asymptotically tends to 0 as the sample size N_p tends to infinity. Thanks to the large amount of pretrained samples N_p, we can approximate the generalization error bound for

*Corresponding author. Work started when at NTU.

[1]The pre-trained dataset includes samples from diverse classes. Here, we only consider the pre-trained data belonging to the classes of downstream task.

the model learned by ProGrad as

$$\mathcal{R}_d(\hat{f}_{\text{prograd}}) \leq \frac{1}{2}\gamma_{\mathcal{F}}(S,P) + \frac{cB\left(\sqrt{2\log(2)L}+1\right)\prod_{j=1}^{L}M_j}{\sqrt{N_d}} + \frac{3}{2}\sqrt{\frac{(b-a)\ln(4/\epsilon)}{2N_d}} + \frac{1}{2}\sqrt{\frac{(b-a)^2\ln(4/\epsilon)}{2}\frac{1}{N_d}}. \tag{6}$$

Similarly, we have the generalization error for $\hat{f}_{\text{coop}}$ as

$$\mathcal{R}_d(\hat{f}_{\text{coop}}) \leq 2\frac{cB\left(\sqrt{2\log(2)L}+1\right)\prod_{j=1}^{L}M_j}{\sqrt{N_d}} + 3\sqrt{\frac{(b-a)\ln(4/\epsilon)}{2N_d}} + \sqrt{\frac{(b-a)^2\ln(4/\epsilon)}{2}\frac{1}{N_d}}. \tag{7}$$

If the gap between the pre-trained domain $\mathcal{D}_p$ and the downstream domain $\mathcal{D}_d$ is very small, the $\gamma_{\mathcal{F}}(D,P)$ will tend to 0. Under this assumption, the estimation error bound of $\mathcal{R}_d(\hat{f}_{\text{coop}})$ is at least 2 times greater than $\mathcal{R}_d(\hat{f}_{\text{prograd}})$. Considering that in few-shot setting, N_d is typically very small, which makes our ProGrad model $\hat{f}_{\text{prograd}}$ a much lower error bound than conventional fine-tuning model $\hat{f}_{\text{coop}}$.

B. Additional Implementation Details

For ProGrad implementation, we first initialize the learnable context vector $\boldsymbol{v}$ with the word embeddings of the zero-shot hand-crafted prompt. Concretely, if the context length M is 16 and the hand-crafted prompt is "a photo of a", which only has 4 tokens, we initialize the former 12 context vectors with zeros and the last 4 context vectors with the word embedding of "a photo of a". We follow the training settings of CoOp [6]: All prompt-based models are trained by SGD with an initial learning rate of 0.002 which is decayed by the cosine annealing rule. During the first epoch, we use the warm-up trick by fixing the learning rate to 1×10^{-5} to alleviate the gradient explosion. The training epoch is set to 50 for all shots of experiments of ImageNet dataset. For the rest 10 datasets, the training epoch is set to 50 for 1 shot, 100 for 2/4 shots and 200 for 8/16 shots. We train all prompt-based model with batch size of 32 expect for CoCoOp. As described in [7], CoCoOp consumes a significant amount of GPU memory if the batch size is set larger than one. We set the batch size to 1, following their original setting. Our experiments are conducted on one 2080Ti GPU for all datasets except ImageNet where we train the models on one A100 GPU.

C. Hand-crafted Prompts

The hand-crafted prompts for 11 datasets as well as the ImageNet variants are listed in Table 1. We select the ensemble prompts from CLIP [3], examples for ImageNet are shown in Table 2.

Table 1: Hand-crafted Prompts.

Dataset	Hand-crafted prompt
OxfordPets	"a type of pet, a photo of a {}."
OxfordFlowers	"a type of flower, a photo of a {}."
FGVCAircraft	"a type of aircraft, a photo of a {}.
DescribableTextures	"a texture of {}."
EuroSAT	"a centered satellite photo of {}."
StanfordCars	"a photo of a {}."
Food101	"a type of food, a photo of {}."
SUN397	"a photo of a {}."
Caltech101	"a photo of a {}."
UCF101	"a photo of a person doing {}."
ImageNet	"a photo of a {}."
ImageNetSketch	"a photo of a {}."
ImageNetV2	"a photo of a {}."
ImageNetA	"a photo of a {}."
ImageNetR	"a photo of a {}."

D. Additional Experiments

D.1. Additional Few-shot Classification Results

In this section, we further provide the detailed few-shot classification results of other learning-based fine-tuning methods with confidence interval at 95% in Table 3 and Table 4.

Cosine. As described in Section 4.5 of the main paper, we plug in an additional cosine classifier on top of the visual backbone and trained on downstream dataset.

CoOp learns the context prompt from data rather than hand-crafted design.

CLIP-Adapter learns additional feature adapter to boost conventional fine-tuning results.

Cosine + ProGrad employs ProGrad to the training process of cosine classifier.

CoOp + l_2 prompt reg. We further investigate whether simply using the l_2 distance between learned prompt vector $\boldsymbol{v}$ and the word embedding vector of hand-crafted prompt $\boldsymbol{v}_{\text{zs}}$ as the regularization can improve few-shot performance, *i.e.*, $\mathcal{L}_{\text{total}}(\boldsymbol{v}) = \mathcal{L}_{\text{ce}}(\boldsymbol{v}) + \alpha\|\boldsymbol{v} - \boldsymbol{v}_{\text{zs}}\|_2$, where we select $\alpha = 0.01$.

CoOp + GM applies gradient matching method [5] to CoOp, *i.e.*, we not only project the $\boldsymbol{G}_{\text{d}}$ to the perpendicular direction of $\boldsymbol{G}_{\text{g}}$ as the updated gradient, but also project the $\boldsymbol{G}_{\text{g}}$ to the perpendicular direction of $\boldsymbol{G}_{\text{d}}$ as the updated gradient to fine-tune the model alternately.

CoOp + KD. As described in Section 4.5 of the main paper, we apply knowledge distillation loss to CoOp, *i.e.*, $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{ce}} + \mathcal{L}_{\text{kl}}$

ProGrad Upper Bound first optimizes a prompt with plain cross-entropy loss on the full dataset to create $\boldsymbol{G}_{\text{g}}$ and then use such gradient to implement ProGrad.

Table 2: Prompt Ensembling Examples for ImageNet.

“a bad photo of a {}.” “a photo of many {}.” “a sculpture of a {}.”
“a photo of the hard to see {}.” “a low resolution photo of the {}.”
“a rendering of a {}.” “graffiti of a {}.” “a bad photo of the {}.”
“a cropped photo of the {}.” “a tattoo of a {}.” “the embroidered {}.”
“a photo of a hard to see {}.” “a bright photo of a {}.”
“a photo of a clean {}.” “a photo of a dirty {}.”
“a dark photo of the {}.” “a drawing of a {}.”
“a photo of my {}.” “the plastic {}.” “a photo of the cool {}.”
“a close-up photo of a {}.” “a black and white photo of the {}.”
“a painting of the {}.” “a painting of a {}.”
“a pixelated photo of the {}.” “a sculpture of the {}.”
“a bright photo of the {}.” “a cropped photo of a {}.” “a plastic {}.”
“a photo of the dirty {}.” “a jpeg corrupted photo of a {}.”
“a blurry photo of the {}.” “a photo of the {}.” “a good photo of the {}.”
“a rendering of the {}.” “a {} in a video game.’ “a photo of one {}.”
“a doodle of a {}.” “a close-up photo of the {}.” “a photo of a {}.”
“the origami {}.” “the {} in a video game.’ “a sketch of a {}.”
“a doodle of the {}.” “a origami {}.” “a low resolution photo of a {}.”
“the toy {}.” “a rendition of the {}.” “a photo of the clean {}.”
“a photo of a large {}.” “a rendition of a {}.” “a photo of a nice {}.”
“a photo of a weird {}.” “a blurry photo of a {}.” “a cartoon {}.”
“art of a {}.” “a sketch of the {}.” “a embroidered {}.”
“a pixelated photo of a {}.” “itap of the {}.”
“a jpeg corrupted photo of the {}.” “a good photo of a {}.”
“a plushie {}.” “a photo of the nice {}.” “a photo of the small {}.”
“a photo of the weird {}.” “the cartoon {}.” “art of the {}.”
“a drawing of the {}.” “a photo of the large {}.”
“a black and white photo of a {}.” “the plushie {}.”
“a dark photo of a {}.” “itap of a {}.”
“graffiti of the {}.” “a toy {}.” “itap of my {}.”
“a photo of a cool {}.” “a photo of a small {}.” “a tattoo of the {}.”

For all prompt-based methods, we set the context length M to 16 except for CoOp + l_2 prompt reg. The learned length for CoOp + l_2 prompt reg needs to be equal to the hand-crafted prompt length to compute the l_2 norm, *e.g.*, the M has to be 4 if the hand-crafted prompt is “`a photo of a` ”. According to the average results in Table 3, we observe that our CoOp + ProGrad still achieves the best average performance. By comparing the results of 1) Cosine and Cosine + ProGrad; and 2) CoOp and CoOp + ProGrad, we demonstrates both conventional “pre-train then fine-tune” paradigm and prompt tuning paradigm can benefit from our `ProGrad`. The gap between CoOp and CoOp + l_2 prompt reg demonstrates that directly regularize the learned prompt to be not far away from the hand-crafted prompt has limited improvement. By digging into CoOp + KD and CoOp + GM, we find performance improvement by introducing the general knowledge. However, their performance still underperforms our CoOp + ProGrad. This is because 1) CoOp + KD learns the average knowledge from two domains which still allows the fine-tuned model to learn from the downstream knowledge that conflicts with the general knowledge; 2) CoOp + MD additional requires the fine-tuned model to discards the general knowledge that is not aligned with the downstream knowledge, as the downstream data is limited, the inaccurate estimation of $\boldsymbol{G}_{\mathrm{d}}$ will lead the model focus on biased general knowledge.

D.2. Additional Results for Base-to-New Generalization

Table 5 further presents the results for base-to-new generalization on each of the 11 datasets.

D.3. Additional Results for Domain Generalization

Table 6 further provides the averaged accuracies with standard deviations for domain generalization setting.

References

[1] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *arXiv preprint arXiv:2110.04544*, 2021. 5, 6

[2] Alfred Müller. Integral probability metrics and their generating classes of functions. *Advances in Applied Probability*, 29(2):429–443, 1997. 1

[3] 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 *ICML*, 2021. 2

[4] Shuo Yang, Songhua Wu, Tongliang Liu, and Min Xu. Bridging the gap between few-shot and many-shot learning via distribution calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 1

[5] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *NeurIPS*, 2020. 2

[6] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *arXiv preprint arXiv:2109.01134*, 2021. 2

[7] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *CVPR*, 2022. 2

Table 3: Accuracy (%) with standard deviation of few-shot learning on 11 datasets (**Part I**). The context length M is set 16 for prompt-based methods. * indicates results copied from [1].

	Method	#shots per class				
		1	2	4	8	16
Average	Cosine	30.50 ± 1.24	43.74 ± 1.37	53.33 ± 1.57	61.26 ± 1.45	65.00 ± 2.87
	CoOp	59.44 ± 1.88	62.31 ± 1.40	66.72 ± 0.93	70.06 ± 0.53	73.48 ± 0.39
	CLIP-Adapter*	61.45	64.32	67.51	70.78	74.35
	Cosine + ProGrad	32.29 ± 1.12	46.14 ± 1.49	55.18 ± 1.99	62.05 ± 0.93	66.47 ± 1.69
	CoOp + l_2 prompt reg	60.84 ± 1.16	62.75 ± 1.18	66.85 ± 0.76	70.08 ± 0.58	72.92 ± 0.46
	CoOp + GM	61.27 ± 0.96	63.23 ± 0.50	64.59 ± 0.63	66.40 ± 0.49	67.12 ± 0.29
	CoOp + KD	61.52 ± 0.99	64.07 ± 0.52	66.52 ± 0.38	70.01 ± 0.31	72.01 ± 0.37
	ProGrad Upper Bound	74.14 ± 0.51	74.24 ± 0.29	74.52 ± 0.52	75.20 ± 0.35	76.36 ± 0.25
	ProGrad	62.61 ± 0.80	64.90 ± 0.86	68.45 ± 0.52	71.41 ± 0.49	74.28 ± 0.40
ImageNet	Cosine	15.95 ± 0.07	26.56 ± 0.30	37.08 ± 0.29	46.18 ± 0.19	53.36 ± 0.39
	CoOp	57.15 ± 1.03	57.25 ± 0.43	59.51 ± 0.25	61.59 ± 0.17	63.00 ± 0.18
	CLIP-Adapter*	58.14	58.55	59.41	60.36	61.27
	Cosine + ProGrad	19.21 ± 0.28	31.18 ± 0.18	42.59 ± 0.29	51.73 ± 0.18	57.65 ± 0.33
	CoOp + l_2 prompt reg	57.51 ± 0.22	61.27 ± 0.49	62.49 ± 0.12	62.71 ± 0.01	62.88 ± 0.09
	CoOp + GM	60.41 ± 0.17	60.51 ± 0.13	60.75 ± 0.06	61.01 ± 0.14	61.44 ± 0.03
	CoOp + KD	60.85 ± 0.22	61.08 ± 0.10	61.51 ± 0.07	61.67 ± 0.12	62.05 ± 0.09
	ProGrad Upper Bound	61.28 ± 0.19	61.60 ± 0.14	62.42 ± 0.16	63.5 ± 0.15	64.34 ± 0.11
	ProGrad	57.75 ± 0.24	59.75 ± 0.33	61.46 ± 0.07	62.54 ± 0.03	63.45 ± 0.08
Caltech101	Cosine	60.76 ± 1.71	73.10 ± 1.01	81.43 ± 0.65	87.02 ± 0.60	90.60 ± 0.05
	CoOp	87.40 ± 0.98	87.92 ± 1.12	89.48 ± 0.47	90.25 ± 0.18	92.00 ± 0.02
	CLIP-Adapter*	88.52	89.19	91.04	91.71	93.42
	Cosine + ProGrad	61.95 ± 0.12	75.24 ± 0.88	82.98 ± 0.38	88.59 ± 0.21	91.31 ± 0.19
	CoOp + l_2 prompt reg	87.04 ± 0.61	87.37 ± 0.78	88.82 ± 0.40	89.62 ± 0.29	91.67 ± 0.26
	CoOp + GM	89.14 ± 0.15	89.37 ± 0.26	89.64 ± 0.33	89.36 ± 0.31	89.42 ± 0.13
	CoOp + KD	89.06 ± 0.29	89.71 ± 0.20	90.13 ± 0.16	90.09 ± 0.30	91.39 ± 0.05
	ProGrad Upper Bound	91.08 ± 0.11	91.70 ± 0.30	91.76 ± 0.52	91.84 ± 0.16	92.86 ± 0.07
	ProGrad	88.68 ± 0.34	87.98 ± 0.69	89.99 ± 0.26	90.83 ± 0.07	92.17 ± 0.17
OxfordPets	Cosine	26.33 ± 0.75	41.60 ± 1.93	55.29 ± 1.97	66.60 ± 0.82	66.84 ± 16.24
	CoOp	86.01 ± 0.47	82.21 ± 2.12	86.63 ± 1.02	85.15 ± 1.12	87.06 ± 0.88
	CLIP-Adapter*	81.44	81.57	82.69	84.13	85.31
	Cosine + ProGrad	26.08 ± 0.73	40.58 ± 2.01	55.23 ± 1.44	66.78 ± 1.58	68.96 ± 14.35
	CoOp + l_2 prompt reg	87.55 ± 0.15	82.12 ± 2.61	84.93 ± 1.77	84.38 ± 0.75	86.28 ± 0.45
	CoOp + GM	87.05 ± 0.65	87.06 ± 0.67	88.45 ± 0.45	88.35 ± 0.15	88.38 ± 0.27
	CoOp + KD	87.10 ± 1.47	87.40 ± 0.60	88.56 ± 0.19	88.77 ± 0.24	89.16 ± 0.16
	ProGrad Upper Bound	88.56 ± 0.30	87.82 ± 0.78	87.99 ± 0.80	88.02 ± 0.40	88.88 ± 0.31
	ProGrad	88.36 ± 0.73	86.89 ± 0.42	88.04 ± 0.50	87.91 ± 0.54	89.00 ± 0.32
StanfordCars	Cosine	18.96 ± 0.34	33.37 ± 0.38	47.75 ± 0.38	61.30 ± 0.25	71.94 ± 0.31
	CoOp	55.68 ± 1.23	58.33 ± 0.60	63.05 ± 0.09	68.37 ± 0.25	73.34 ± 0.49
	CLIP-Adapter*	56.02	58.24	63.07	67.00	72.83
	Cosine + ProGrad	21.13 ± 0.50	39.44 ± 0.83	54.54 ± 0.57	66.47 ± 0.14	73.41 ± 0.11
	CoOp + l_2 prompt reg	55.86 ± 0.66	57.69 ± 0.51	62.82 ± 0.07	66.63 ± 0.25	69.86 ± 0.44
	CoOp + GM	57.37 ± 0.36	58.46 ± 0.24	59.72 ± 0.66	62.32 ± 0.59	63.87 ± 0.37
	CoOp + KD	57.48 ± 1.47	59.09 ± 0.60	61.47 ± 0.19	67.73 ± 0.24	70.48 ± 0.16
	ProGrad Upper Bound	70.54 ± 0.14	71.57 ± 0.16	71.66 ± 0.50	72.73 ± 0.15	75.27 ± 0.36
	ProGrad	58.38 ± 0.23	61.81 ± 0.45	65.62 ± 0.43	69.29 ± 0.11	73.46 ± 0.29
Flowers102	Cosine	51.33 ± 2.77	70.06 ± 2.29	82.43 ± 1.65	91.74 ± 0.73	95.68 ± 0.22
	CoOp	68.13 ± 1.74	76.68 ± 1.82	86.13 ± 0.75	91.74 ± 0.49	94.72 ± 0.34
	CLIP-Adapter*	71.97	78.80	85.31	90.69	94.30
	Cosine + ProGrad	52.08 ± 2.31	70.13 ± 1.90	81.09 ± 2.06	91.62 ± 0.41	93.94 ± 0.02
	CoOp + l_2 prompt reg	71.12 ± 0.55	80.36 ± 0.54	86.42 ± 0.33	91.58 ± 0.59	94.25 ± 0.38
	CoOp + GM	67.87 ± 0.31	69.09 ± 0.49	71.69 ± 0.68	75.76 ± 0.79	78.36 ± 0.34
	CoOp + KD	68.11 ± 1.47	71.02 ± 0.60	76.06 ± 0.19	84.53 ± 0.24	88.05 ± 0.16
	ProGrad Upper Bound	95.29 ± 0.28	95.29 ± 0.38	95.70 ± 0.38	95.86 ± 0.37	96.52 ± 0.02
	ProGrad	73.18 ± 0.73	79.77 ± 0.65	85.37 ± 0.96	91.64 ± 0.24	94.37 ± 0.24

Table 4: Accuracy (%) with confidence interval at 95% of few-shot learning on 11 datasets (**Part II**). The context length M is set 16 for prompt-based methods. * indicates results copied from [1].

	Method	#shots per class				
		1	2	4	8	16
Food101	Cosine	25.32 ± 0.29	41.06 ± 0.29	54.10 ± 1.06	61.88 ± 0.33	68.50 ± 0.24
	CoOp	74.28 ± 1.40	72.45 ± 1.29	73.27 ± 2.07	71.67 ± 0.30	74.68 ± 0.03
	CLIP-Adapter*	75.09	75.59	75.92	76.53	76.97
	Cosine + ProGrad	27.19 ± 0.15	45.28 ± 0.36	58.57 ± 1.01	71.25 ± 0.29	75.61 ± 0.15
	CoOp + l_2 prompt reg	73.58 ± 2.20	68.89 ± 1.30	71.30 ± 0.49	72.42 ± 0.26	75.64 ± 0.33
	CoOp + GM	76.23 ± 1.51	77.97 ± 0.51	78.89 ± 0.10	78.90 ± 0.15	79.07 ± 0.06
	CoOp + KD	76.06 ± 1.47	77.59 ± 0.60	78.72 ± 0.19	78.38 ± 0.24	78.90 ± 0.16
	ProGrad Upper Bound	79.35 ± 0.15	78.19 ± 0.25	77.67 ± 0.51	78.05 ± 0.42	78.99 ± 0.14
	ProGrad	76.04 ± 0.54	74.95 ± 0.57	75.95 ± 0.27	76.65 ± 0.23	78.41 ± 0.08
FGVCAircraft	Cosine	12.47 ± 1.00	17.75 ± 1.35	22.00 ± 1.50	29.14 ± 0.54	36.47 ± 0.18
	CoOp	9.71 ± 6.09	18.74 ± 0.48	21.78 ± 0.50	27.55 ± 0.06	31.37 ± 0.53
	CLIP-Adapter*	19.63	22.27	25.62	30.48	38.72
	Cosine + ProGrad	12.83 ± 0.48	17.59 ± 1.59	19.70 ± 1.62	26.34 ± 0.51	31.98 ± 0.68
	CoOp + l_2 prompt reg	18.01 ± 0.44	19.78 ± 0.23	22.51 ± 0.94	27.24 ± 0.38	30.55 ± 0.54
	CoOp + GM	17.08 ± 0.37	19.34 ± 0.24	19.62 ± 0.40	21.07 ± 0.08	22.52 ± 0.19
	CoOp + KD	17.67 ± 0.45	19.29 ± 0.15	21.21 ± 0.60	25.55 ± 0.30	28.58 ± 0.42
	ProGrad Upper Bound	28.83 ± 0.09	28.97 ± 0.12	30.44 ± 0.79	31.92 ± 0.9	34.32 ± 0.16
	ProGrad	18.81 ± 0.50	20.47 ± 0.90	23.32 ± 0.36	27.02 ± 0.67	31.12 ± 0.62
SUN397	Cosine	25.32 ± 0.18	38.13 ± 0.37	49.83 ± 0.45	56.97 ± 0.21	62.84 ± 0.16
	CoOp	60.30 ± 0.64	59.52 ± 0.60	63.33 ± 0.39	65.65 ± 0.10	69.14 ± 0.11
	CLIP-Adapter*	61.16	62.08	64.74	66.88	69.20
	Cosine + ProGrad	29.66 ± 0.08	45.81 ± 0.39	55.92 ± 0.35	63.61 ± 0.16	67.33 ± 0.25
	CoOp + l_2 prompt reg	57.64 ± 0.33	59.81 ± 0.33	64.88 ± 0.45	67.66 ± 0.16	69.56 ± 0.11
	CoOp + GM	62.73 ± 0.35	62.85 ± 0.10	63.32 ± 0.21	63.77 ± 0.04	64.47 ± 0.27
	CoOp + KD	62.89 ± 0.40	64.10 ± 0.29	65.83 ± 0.26	67.02 ± 0.05	68.32 ± 0.19
	ProGrad Upper Bound	67.65 ± 0.33	66.89 ± 0.44	68.33 ± 0.36	68.46 ± 0.24	70.18 ± 0.33
	ProGrad	60.54 ± 0.24	63.06 ± 0.11	66.39 ± 0.43	67.62 ± 0.28	69.84 ± 0.18
DTD	Cosine	27.05 ± 0.83	38.42 ± 0.48	48.44 ± 2.29	58.47 ± 0.51	61.88 ± 0.38
	CoOp	43.77 ± 2.12	46.06 ± 1.05	53.82 ± 0.77	60.06 ± 1.18	63.26 ± 0.22
	CLIP-Adapter*	45.65	50.54	56.43	61.59	66.03
	Cosine + ProGrad	26.95 ± 1.38	38.87 ± 1.02	48.05 ± 3.02	56.24 ± 2.81	63.40 ± 0.58
	CoOp + l_2 prompt reg	43.74 ± 1.45	45.98 ± 2.76	53.25 ± 1.55	59.08 ± 0.58	62.31 ± 1.05
	CoOp + GM	43.81 ± 2.15	47.64 ± 0.63	49.17 ± 1.52	53.17 ± 0.63	54.06 ± 0.45
	CoOp + KD	43.01 ± 2.18	49.31 ± 1.10	53.03 ± 1.49	60.26 ± 0.34	63.14 ± 0.39
	ProGrad Upper Bound	66.11 ± 0.77	67.04 ± 0.72	67.24 ± 0.34	68.46 ± 0.41	69.27 ± 0.64
	ProGrad	46.14 ± 1.74	49.78 ± 1.37	54.43 ± 0.86	60.69 ± 0.10	63.97 ± 0.61
EuroSAT	Cosine	37.55 ± 5.27	52.93 ± 5.66	49.81 ± 6.23	46.08 ± 11.13	33.30 ± 13.04
	CoOp	49.40 ± 3.86	62.23 ± 4.94	69.49 ± 3.23	76.56 ± 1.73	84.05 ± 1.05
	CLIP-Adapter*	54.53	63.73	68.33	75.81	82.81
	Cosine + ProGrad	41.55 ± 6.19	51.35 ± 5.76	47.64 ± 9.68	30.03 ± 2.99	33.30 ± 1.67
	CoOp + l_2 prompt reg	54.28 ± 5.38	62.60 ± 2.77	70.43 ± 1.81	77.32 ± 2.20	83.30 ± 1.11
	CoOp + GM	48.02 ± 4.04	57.12 ± 2.03	62.88 ± 2.28	68.74 ± 2.18	67.72 ± 1.00
	CoOp + KD	49.51 ± 1.12	58.89 ± 1.06	66.79 ± 0.76	74.37 ± 0.91	77.87 ± 1.74
	ProGrad Upper Bound	90.97 ± 0.36	89.81 ± 1.89	89.57 ± 0.70	90.19 ± 0.47	90.92 ± 0.35
	ProGrad	56.32 ± 3.04	63.10 ± 3.77	72.53 ± 1.29	78.04 ± 2.45	83.74 ± 0.70
UCF101	Cosine	34.41 ± 0.40	48.21 ± 1.00	58.47 ± 0.81	68.46 ± 0.66	73.64 ± 0.32
	CoOp	62.03 ± 1.13	63.98 ± 0.91	67.45 ± 0.74	72.11 ± 0.29	75.67 ± 0.49
	CLIP-Adapter*	63.80	66.98	70.07	73.45	76.99
	Cosine + ProGrad	36.61 ± 0.14	52.11 ± 1.43	60.66 ± 1.50	69.85 ± 0.94	74.27 ± 0.30
	CoOp + l_2 prompt reg	62.88 ± 0.74	64.43 ± 0.71	67.46 ± 0.40	72.28 ± 0.88	75.77 ± 0.29
	CoOp + GM	64.27 ± 0.48	66.14 ± 0.25	66.37 ± 0.27	67.91 ± 0.29	68.96 ± 0.04
	CoOp + KD	64.99 ± 0.35	67.29 ± 0.46	68.44 ± 0.13	71.77 ± 0.41	74.15 ± 0.55
	ProGrad Upper Bound	76.99 ± 0.42	77.24 ± 0.47	76.95 ± 0.66	78.16 ± 0.17	78.44 ± 0.26
	ProGrad	64.55 ± 0.50	66.35 ± 0.18	69.86 ± 0.30	73.33 ± 0.65	77.28 ± 0.96

Table 5: Accuracy (%) for the base-to-new generalization evaluation. The context length M is 4 for prompt-based methods which are learned from the base classes with 4 shots. H: Harmonic mean.

(a) **Average over 11 datasets**.

	Base	New	H
CLIP	61.72	65.91	63.64
CoOp	71.96	61.26	65.58
CoCoOp	72.23	60.77	65.35
ProGrad	**73.29**	**65.96**	**69.06**

(b) ImageNet.

	Base	New	H
CLIP	64.46	59.99	62.14
CoOp	65.49	57.70	61.35
CoCoOp	66.21	58.01	61.84
ProGrad	**66.96**	**60.04**	**63.23**

(c) Caltech101.

	Base	New	H
CLIP	90.90	90.72	90.81
CoOp	94.38	87.48	90.80
CoCoOp	94.43	87.81	91.00
ProGrad	**94.47**	**90.84**	**92.46**

(d) OxfordPets.

	Base	New	H
CLIP	85.86	93.85	89.68
CoOp	90.31	94.03	92.13
CoCoOp	89.07	91.00	90.02
ProGrad	**91.78**	**94.86**	**93.29**

(e) StanfordCars.

	Base	New	H
CLIP	55.55	**66.35**	60.47
CoOp	61.77	62.51	62.14
CoCoOp	61.68	59.98	60.82
ProGrad	**63.01**	64.32	**63.66**

(f) Flowers102.

	Base	New	H
CLIP	64.10	**70.92**	67.34
CoOp	**89.33**	62.77	73.73
CoCoOp	88.07	66.26	75.62
ProGrad	88.19	69.38	**77.66**

(g) Food101.

	Base	New	H
CLIP	81.48	82.15	81.81
CoOp	80.40	81.09	80.74
CoCoOp	79.77	77.68	78.71
ProGrad	**83.10**	**83.57**	**83.33**

(h) FGVCAircraft.

	Base	New	H
CLIP	17.89	**25.13**	20.90
CoOp	22.53	20.40	21.41
CoCoOp	22.73	19.40	20.93
ProGrad	**22.77**	24.24	**23.48**

(i) SUN397.

	Base	New	H
CLIP	66.45	**70.17**	68.26
CoOp	71.48	65.57	68.40
CoCoOp	71.88	67.10	69.41
ProGrad	**73.71**	69.78	**71.69**

(j) DTD.

	Base	New	H
CLIP	49.31	**54.35**	51.71
CoOp	67.71	43.92	53.28
CoCoOp	63.54	40.78	49.68
ProGrad	**66.90**	53.06	**59.18**

(k) EuroSAT.

	Base	New	H
CLIP	39.26	43.62	41.33
CoOp	73.53	40.19	51.97
CoCoOp	83.63	40.95	54.98
ProGrad	**79.67**	**49.99**	**61.43**

(l) UCF101.

	Base	New	H
CLIP	63.70	**67.71**	65.64
CoOp	74.59	58.23	65.40
CoCoOp	73.51	59.55	65.80
ProGrad	**75.66**	65.52	**70.23**

Table 6: Domain generalization results with standard deviation.

(a) ResNet50

	Source	Target			
	ImageNet	ImageNet-V2	ImageNet-Sketch	ImageNet-A	ImageNet-R
CoOp	61.34 ± 0.11	53.81 ± 0.10	32.83 ± 0.30	22.08 ± 0.59	54.62 ± 0.74
CoCoOp	61.04 ± 0.18	53.71 ± 0.26	32.30 ± 0.34	22.07 ± 0.34	53.60 ± 0.27
ProGrad	**62.17** ± 0.06	**54.70** ± 0.18	**34.40** ± 0.18	**23.05** ± 0.13	**56.77** ± 0.33

(b) ResNet101

	Source	Target			
	ImageNet	ImageNet-V2	ImageNet-Sketch	ImageNet-A	ImageNet-R
CoOp	63.99 ± 0.13	56.99 ± 0.21	39.40 ± 0.29	29.50 ± 0.56	64.04 ± 0.27
CoCoOp	63.59 ± 0.22	56.98 ± 0.25	39.16 ± 0.36	29.09 ± 0.18	64.14 ± 0.01
ProGrad	**64.98** ± 0.15	**57.86** ± 0.04	**40.53** ± 0.17	**30.13** ± 0.09	**65.61** ± 0.08

(c) ViT-B/32

	Source	Target			
	ImageNet	ImageNet-V2	ImageNet-Sketch	ImageNet-A	ImageNet-R
CoOp	64.74 ± 0.14	56.59 ± 0.27	40.03 ± 0.64	31.10 ± 0.06	64.54 ± 0.52
CoCoOp	64.63 ± 0.15	56.59 ± 0.04	40.74 ± 0.24	30.27 ± 0.05	64.12 ± 0.10
ProGrad	**65.36** ± 0.23	**57.42** ± 0.33	**41.73** ± 0.25	**31.89** ± 0.26	**66.53** ± 0.08

(d) ViT-B/16

	Source	Target			
	ImageNet	ImageNet-V2	ImageNet-Sketch	ImageNet-A	ImageNet-R
CoOp	69.86 ± 0.22	62.83 ± 0.37	46.90 ± 0.59	48.98 ± 0.33	74.55 ± 0.46
CoCoOp	70.13 ± 0.23	63.05 ± 0.06	46.48 ± 0.17	49.36 ± 0.26	73.80 ± 0.08
ProGrad	**70.45** ± 0.16	**63.35** ± 0.08	**48.17** ± 0.10	**49.45** ± 0.08	**75.21** ± 0.32