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

Abstract

Thanks to the large pre-trained vision-language models (VLMs) like CLIP [37], we can craft a zero-shot classifier by discrete prompt design, e.g.*, the confidence score of an image being "*`[CLASS]`*" can be obtained by using the VLM provided similarity between the image and the prompt sentence "*`a photo of a [CLASS]`*". Furthermore, prompting shows great potential for fast adaptation of VLMs to downstream tasks if we fine-tune the soft prompts with few samples. However, we find a common failure that improper fine-tuning or learning with extremely few-shot samples may even under-perform the zero-shot prediction. Existing methods still address this problem by using traditional anti-overfitting techniques such as early stopping and data augmentation, which lack a principled solution specific to prompting. In this paper, we present Prompt-aligned Gradient, dubbed* `ProGrad` *to prevent prompt tuning from forgetting the general knowledge learned from VLMs. In particular,* `ProGrad` *only updates the prompt whose gradient is aligned (or non-conflicting) to the general knowledge, which is represented as the optimization direction offered by the pre-defined prompt predictions. Extensive experiments under the few-shot learning, domain generalization, base-to-new generalization and cross-dataset transfer settings demonstrate the stronger few-shot generalization ability of* `ProGrad` *over state-of-the-art prompt tuning methods.*

1. Introduction

After seeing and reading countless image-text association pairs, large and deep vision-language models (VLM) [37, 18] can memorize the **general knowledge** (a.k.a. encyclopedic knowledge) about what visual patterns correspond to what textual sequence and vice versa. Thanks to the powerful language modeling of VLMs, we can establish a communication channel in human-readable natural language, *i.e.*, **prompt** [25, 51, 19], to query the general knowledge. Prompting bridges the interface gap between the pre-trained and downstream tasks (*e.g.*, regression *vs.* classification) without the need for additional fine-tuning adaptation. For example, we can craft a concrete prompt—"`a photo of a [CLASS]`"—for zero-shot image classification: by using the popular vision-language model CLIP [37], we input the image to the vision end and the prompt sentence to the language end, then obtain a vision-language similarity as the confidence score of classifying the image as "`[CLASS]`".

In practice, the prompt-based zero-shot image classification is not accurate because the hand-crafted prompt may not be the most machine-favorable (*e.g.*, "`this is a picture of`" could be more grammatically prevailing in VLM training), or not specific to the downstream domain (*e.g.*, "`a photo of a person doing`" is better in action recognition) [37]. Recently, prompt tuning or prefix tuning [23, 26, 54, 55] has been proposed to replace the hand-crafted prompt with a set of tunable word embedding vectors, which do not have to be translatable back to human-readable words. Yet, prompt tuning is still as tricky as conventional fine-tuning: as the training continues, the generalization ability may decrease and even under-perform the zero-shot baseline. As shown in Figure 1(a&b), the prompt tuning method CoOp [54] achieves the best results via early stopping, and its accuracies heavily drop by at most 4% when the training continues. Besides, Figure 1(c&d) show that CoOp underperforms zero-shot CLIP without augmentation or enough samples from downstream tasks. To the best of our knowledge, existing methods still rely on the conventional anti-overfitting techniques such as early stopping and data augmentation [54, 55, 11, 36], which lacks a principled solution to the nature of improper prompt tuning. Furthermore, the Grad-CAM visualization results indicate that the fine-tuned prompt misleads the VLM to forget the general knowledge that the classification should at least focus on the foreground object but not the background. Comparing CoOp (Figure 2(b)) with zero-shot CLIP (Figure 2(c)), we find that the CoOp model distracts its attention to the background, while CLIP mainly focuses on the foreground object. These results demonstrate the over-fitting risk of existing prompt

*Corresponding author.

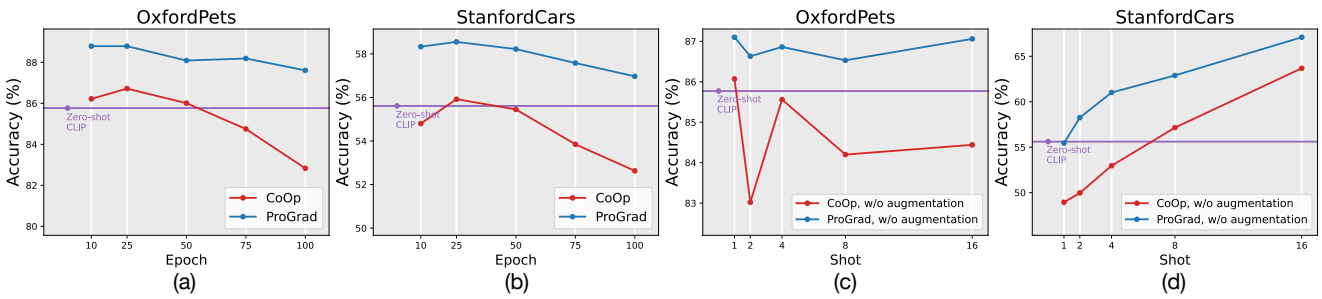

Figure 1: Comparison of Zero-shot CLIP, CoOp, and our `ProGrad` on Stanford Cars and OxfordPets datasets. (a)&(b): Given 1 shot training sample, CoOp's performance severely drops and under-performs zero-shot CLIP by large margins when the training continues. (c)&(d): CoOp may fail to improve CLIP without data augmentation or plenty of samples.

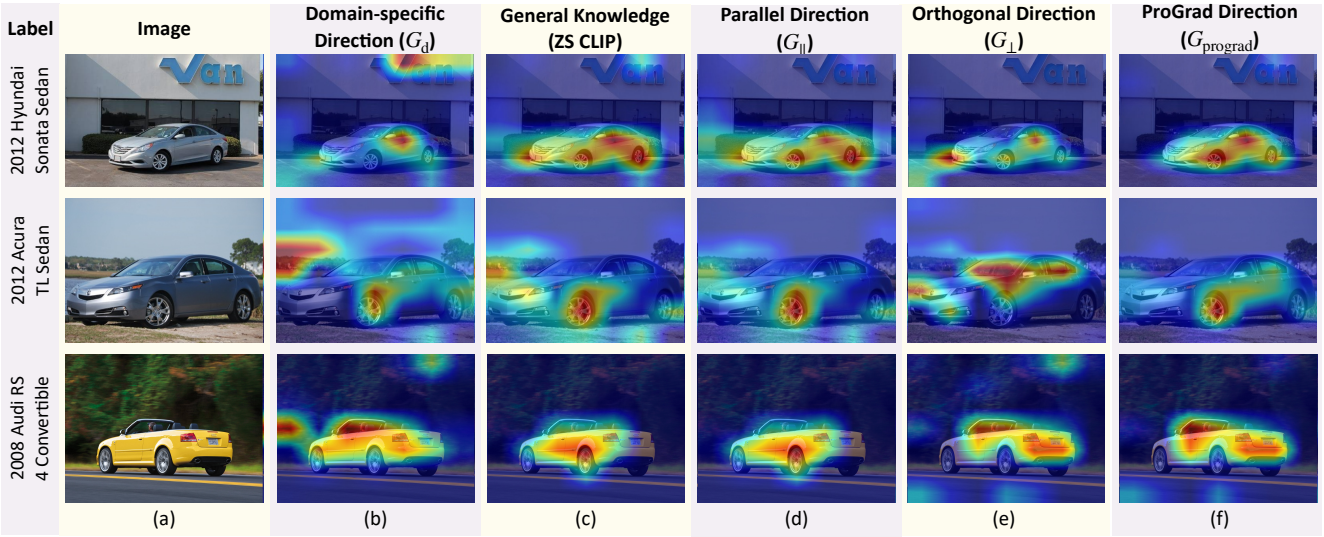

Figure 2: Comparisons of Grad-CAM [43] visualization for prompt tuning methods using different gradient strategies on Stanford Cars Datasets.

tuning strategies, especially when the number of training samples is extremely limited (*e.g.*, 1 or 2).

To this end, we present a novel prompt tuning method called Prompt-aligned Gradient (`ProGrad`) to overcome the improperly biased tuning for CLIP. The principle of `ProGrad` is to regularize each tuning step not to conflict with the general knowledge offered by the original prompt, *e.g.*, the zero-shot CLIP predictions. Specifically, we measure the general knowledge direction $\boldsymbol{G}_{\text{g}}$ using the gradient of Kullback–Leibler (KL) divergence between the predictions of the *zero-shot prompted CLIP* and the few-shot fine-tuned model, which we name as *general direction*. Similarly, we compute the domain-specific knowledge direction $\boldsymbol{G}_{\text{d}}$ using the gradient of cross-entropy between the *ground-truth* and the few-shot fine-tuned model, dubbed *domain-specific direction*. We decompose the domain-specific direction $\boldsymbol{G}_{\text{d}}$ into: 1) a vector $\boldsymbol{G}_{\perp}$ orthogonal to the general direction, which denotes the non-conflicting domain-specific knowledge; and 2) a vector $\boldsymbol{G}_{\parallel}$ parallel to the general direction, which denotes the general knowledge. Note that the first gradient component does NOT override the general direction as any two orthogonal vectors can be transformed into two non-conflicting base vectors. For the second component, it must be one of the two directions: 1) the same of the general direction, which indicates that the update is aligned to the general knowledge, and 2) the opposite of general direction, indicating a conflicting update that should be discarded to avoid forgetting. Overall, in each iteration, `ProGrad` only updates the parameters in the prompt-aligned direction that has an acute angle to the general direction. Compared to CoOp and CLIP, both $\boldsymbol{G}_{\text{g}}$ and $\boldsymbol{G}_{\perp}$ (Figure 2(d&e)) help to regularize the model to focus on the foreground, and `ProGrad` (Figure 2(f)) further improves the visual response.

Following CLIP, CoOp and CoCoOp [55], we evaluate `ProGrad` under the few-shot learning, domain generalization, base-to-new generalization and cross-dataset transfer

settings over 15 image classification benchmarks, covering generic object classification, fine-grained image recognition, action classification. In summary, our ProGrad achieves: 1) clear improvement compared to CoOp over all of the 11 datasets; 2) clear improvement on the harmonic mean of base and new classes accuracies on all 11 datasets compared to CoOp and CoCoOp, and 3) clear improvement on both the source and target datasets of the domain generalization.

2. Related Work

VLMs Adaptation. VLM can be adapted for various downstream tasks, *e.g.*, visual question answering [21, 48], visual grounding [51], image retrieval [28] and semantic segmentation [39]. We focus on image classification task. Conventional "pre-train then fine-tune" paradigm that plugs in a classifier to the visual backbone and trained on downstream data has been widely-adopted, *e.g.*, Linear Probe [37]. CLIP-Adapter [10] add feature adapters to boost fine-tuning results. Recently, NLP community has introduced a "prompt-based learning" paradigm that fine-tunes the prompt using a "fill-in-the-blank" cloze test to maximize the ground-truth token [23, 26]. The prompt-based learning has recently been applied in CV community [54, 29, 7, 46, 4, 20, 56]: CoOp [54] uses a continuous prompt optimization from downstream data instead of hand-craft design. CoCoOp [55] further extends CoOp by learning image conditional prompt to improve generalization. ProDA [29] learns a prompt distribution over the output embedding space. VPT [7] introduces variational prompt tuning by combining a base learned prompt with a residual vector sampled from a instance-specific underlying distribution. TPT [46] proposes a test-time prompt tuning, which does not require training data and optimizes the prompt to achieve consistent predictions across different augmented views. Our `ProGrad` follows the line of *prompt-based learning* to improve few-shot generalization ability by aligning the gradient to general direction, without model structure modification or tuning the pre-trained parameters.

Knowledge Transfer. Forgetting mitigation is widely used in incremental learning [27, 40, 35, 42, 17] through knowledge distillation or memory replay. However, prompt-based learning differs fundamentally from incremental learning in that it does not have access to pre-trained data, which is required for incremental learning to store old data from memory storage. For example, OGD [8] projects gradients from new classes to the orthogonal direction of previous task gradients, but the requirement to store old task gradients is not possible for prompt tuning since we lack access to the pre-training process. Moreover, OGD modifies gradients of downstream tasks in non-conflicting scenarios, potentially leading to sub-optimal performance. Another related field that leverages gradient matching to transfer knowledge is domain generalization [45, 38] and multi-task learning [44, 52]. However, these methods are not directly applicable in prompt tuning whose transfer direction is only from general to downstream. In Appendix, we will show how their methods fail in several ablative studies.

3. Methodology

In this section, we introduce the preliminary concepts of prompt-based zero-shot inference, prompt-based learning, and present our proposed Prompt-aligned Gradient solution to align the domain knowledge with general knowledge for few-shot generalization.

3.1. Preliminaries

Contrastive language-image pre-training (CLIP) [37] adopts a contrastive language-image pre-training paradigm on tremendous pairs of images with natural language descriptions. For contrastive learning, the associated image and sentences are taken as the positive samples, while the non-associated pairs are regarded as negative samples. The contrastive objective maximizes the similarity of positive pairs while minimize the similarity of negative pairs.

Zero-shot transfer inference adapts the pre-trained CLIP model to downstream tasks without fine-tuning the model. Taking image classification as an example, zero-shot transfer is enabled by formulating the classification task as an image-text matching problem, where the text is obtained by extending the "`[CLASS]`" name using a template like "`a photo of a [CLASS].`". CLIP [37] finds that such a simple template narrows the distribution gap to pre-training text inputs. The image-class matching score is measured based on the cosine similarity $< \boldsymbol{w}_i, \boldsymbol{f} >$ between the image feature $\boldsymbol{f}$ and the class-extended text feature $\boldsymbol{w}_i$ for i-th class. The image feature $\boldsymbol{f}$ for image $\boldsymbol{x}$ is extracted by the image encoder, while the text feature $\boldsymbol{w}_i$ for i-th class is obtained by feeding the prompt description into the text encoder. The probability for i-th class is obtained as

$$p_{\text{zs}}(\boldsymbol{w}_i|\boldsymbol{x}) = \frac{\exp(< \boldsymbol{w}_i, \boldsymbol{f} > /\tau)}{\sum_{j=1}^{K} \exp(< \boldsymbol{w}_j, \boldsymbol{f} > /\tau)}, \tag{1}$$

where K denotes the number of classes, and τ is a temperature learned by CLIP.

Prompt-based learning further strengths the transferring ability of the CLIP model and avoids prompt engineering by automatically learning the prompt given few samples from the downstream task. Different from the zero-shot transfer that used a fixed hand-craft prompt, CoOp [54] constructs and fine-tunes a set of M continuous context vectors $\boldsymbol{v} = \{\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_M\}$ as the turnable prompt. Specifically, the prompt $\boldsymbol{t}_i = \{\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_M, \boldsymbol{c}_i\}$ combines the learnable context vectors $\boldsymbol{v}$ and the class token embedding $\boldsymbol{c}_i$, and is fed to the text encoder $g(\cdot)$. CoOp optimizes the static context vectors $\boldsymbol{v}$ by minimizing the negative log-likelihood

of the ground-truth token:

$$\begin{aligned}\mathcal{L}_{\text{ce}}(\boldsymbol{v}) &= -\sum_i \boldsymbol{y}_i \log p(\boldsymbol{t}_i|\boldsymbol{x}),\\ p(\boldsymbol{t}_i|\boldsymbol{x}) &= \frac{\exp(< g(\boldsymbol{t}_i), \boldsymbol{f} > /\tau)}{\sum_{j=1}^{K} \exp(< g(\boldsymbol{t}_j), \boldsymbol{f} > /\tau)},\end{aligned} \tag{2}$$

where $\boldsymbol{y}$ denotes the one-hot ground-truth annotation and K denotes the number of classes.

3.2. Prompt-aligned Gradient

CoOp faced a challenge that the transfer performance drops when the number of annotations is very limited (*e.g.*, one per class), even underperforms the zero-shot transfer. Also, CoOp heavily relies on anti-overfitting techniques such as early stopping and data augmentation. To overcome the over-fitting challenge, we propose an effective and efficient fine-tuning paradigm `ProGrad` to align the few-shot downstream knowledge with the large-scale general knowledge.

Motivated by the success of knowledge distillation [34, 16] in knowledge transfer, we leverage the zero-shot CLIP predictions as the general knowledge, and compare the fine-tuned predictions with the general knowledge to regularize the gradient direction. Specifically, we obtain the domain-specific direction by calculating the cross-entropy $\mathcal{L}_{\text{ce}}(\boldsymbol{v})$ between the model prediction $p(\boldsymbol{t}_i|\boldsymbol{x})$ and the ground-truth $\boldsymbol{y}$ according to Eq. (2), and the general knowledge direction based on the Kullback-Leibler (KL) divergence between $p(\boldsymbol{t}_i|\boldsymbol{x})$ and the zero-shot CLIP prediction $p_{\text{zs}}(\boldsymbol{w}_i|\boldsymbol{x})$:

$$\mathcal{L}_{\text{kl}}(\boldsymbol{v}) = -\sum_i p_{\text{zs}}(\boldsymbol{w}_i|\boldsymbol{x}) \log \frac{p(\boldsymbol{t}_i|\boldsymbol{x})}{p_{\text{zs}}(\boldsymbol{w}_i|\boldsymbol{x})}. \tag{3}$$

We denote the gradients of $\mathcal{L}_{\text{kl}}(\boldsymbol{v})$ and $\mathcal{L}_{\text{ce}}(\boldsymbol{v})$ as $\boldsymbol{G}_{\text{g}} = \nabla_{\boldsymbol{v}}\mathcal{L}_{\text{kl}}(\boldsymbol{v})$ and $\boldsymbol{G}_{\text{d}} = \nabla_{\boldsymbol{v}}\mathcal{L}_{\text{ce}}(\boldsymbol{v})$, respectively. The relations between $\boldsymbol{G}_{\text{g}}$ and $\boldsymbol{G}_{\text{d}}$ are two-fold. (1) Their angle is smaller than 90°(Figure 3(a)), which indicates that the optimization direction of few-shot downstream knowledge does not conflict with general knowledge. In this case, we safely set the updated gradient direction $\boldsymbol{G}_{\text{prograd}}$ as $\boldsymbol{G}_{\text{d}}$. (2) Their angle is larger than 90°(Figure 3(b)), which indicates that the few-shot downstream knowledge conflicts with general knowledge. In other words, optimizing the context vectors following $\boldsymbol{G}_{\text{d}}$ will lead to the forgetting of the pre-trained general knowledge. In this case, we project the $\boldsymbol{G}_{\text{d}}$ to the orthogonal direction of $\boldsymbol{G}_{\text{g}}$ to optimize the model for classification, which avoids increasing the KL loss. Our `ProGrad` strategy is mathematically formulated as:

$$\boldsymbol{G}_{\text{prograd}} = \begin{cases} \boldsymbol{G}_{\text{d}}, & \text{if } \boldsymbol{G}_{\text{d}} \cdot \boldsymbol{G}_{\text{g}} \geq 0 \\ \boldsymbol{G}_{\text{d}} - \lambda \cdot \frac{\boldsymbol{G}_{\text{d}} \cdot \boldsymbol{G}_{\text{g}}}{\|\boldsymbol{G}_{\text{g}}\|^2} \boldsymbol{G}_{\text{g}}, & \text{otherwise.} \end{cases} \tag{4}$$

Fig 3(c) illustrates the pipeline of our `ProGrad`. Instead of updating the context vectors using $\boldsymbol{G}_{\text{d}}$ in CoOp [54], we optimize the context vectors using $\boldsymbol{G}_{\text{prograd}}$, which prevent the gradient direction from overfitting to few-shot downstream samples. We further introduce λ in Eq. (4) to generalize the formulation, which can flexibly control the strength of general knowledge guidance in applications. In particular, $\lambda = 1$ denotes projecting $\boldsymbol{G}_{\text{d}}$ to the orthogonal direction of $\boldsymbol{G}_{\text{g}}$ (Figure 3(b)), while setting $\lambda = 0$ makes `ProGrad` degenerate to CoOp, *i.e.*, CoOp is a special case of our strategy. We include the detailed analysis of λ in Appendix.

Generalization Error Analysis. We further theoretically analyze the generalization error of our `ProGrad`. Here, we provide a sketch proof and include the detailed justification in Appendix. Our `ProGrad` keeps the optimal value $\mathcal{L}_{\text{kl}}$ of the pre-trained domain when optimizing the empirical risk on the downstream domain. The model $\hat{f}_{\text{prograd}}$ learned by such update rule can be viewed as optimizing the empirical risk on pre-trained and downstream domains [52]:

$$\hat{f}_{\text{prograd}} = \operatorname*{argmin}_{f\in\mathcal{F}} \hat{\mathcal{R}}_{(d+p)}(f) = \operatorname*{argmin}_{f\in\mathcal{F}} \hat{\mathcal{R}}_d(f) + \hat{\mathcal{R}}_p(f), \tag{5}$$

where $\mathcal{F}$ is the function class, and $\mathcal{R}(\cdot)$ and $\hat{\mathcal{R}}(\cdot)$ denote the expected risk and empirical risk. We bound the generalization error of `ProGrad` by virtue of *Rademacher Complexity* [1] and the Theorem 6.2 in [53]. The detailed proof is in Appendix.

Theorem 1 *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 $1-\epsilon$,*

$$\begin{aligned}\mathcal{R}_d(\hat{f}_{prograd}) \leq \hat{\mathcal{R}}_{(d+p)}(\hat{f}_{prograd}) &+ \frac{1}{2}\gamma_{\mathcal{F}}(D,P)\\ &+ \mathfrak{R}_p(\mathcal{F}) + \mathfrak{R}_d(\mathcal{F}) + \frac{3}{2}\sqrt{\frac{\ln(4/\epsilon)}{2N_d}}\\ &+ \frac{3}{2}\sqrt{\frac{\ln(4/\epsilon)}{2N_p}} + \frac{1}{2}\sqrt{\frac{\ln(4/\epsilon)}{2}\left(\frac{1}{N_d}+\frac{1}{N_p}\right)},\end{aligned} \tag{6}$$

where $\gamma_{\mathcal{F}}(D,P)$ is the integral probability metric [31] that measures the difference between the distribution of pre-trained domain and the downstream domain, $\mathfrak{R}_d(\mathcal{F})$ and $\mathfrak{R}_p(\mathcal{F})$ are the Rademacher complexity of $\mathcal{F}$.

Note that the bound of $\mathfrak{R}(\mathcal{F})$ is inversely proportional to the number of training samples. Theorem 1 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. The empirical training risk can be minimized to arbitrary small value when using deep models with high capacity. 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

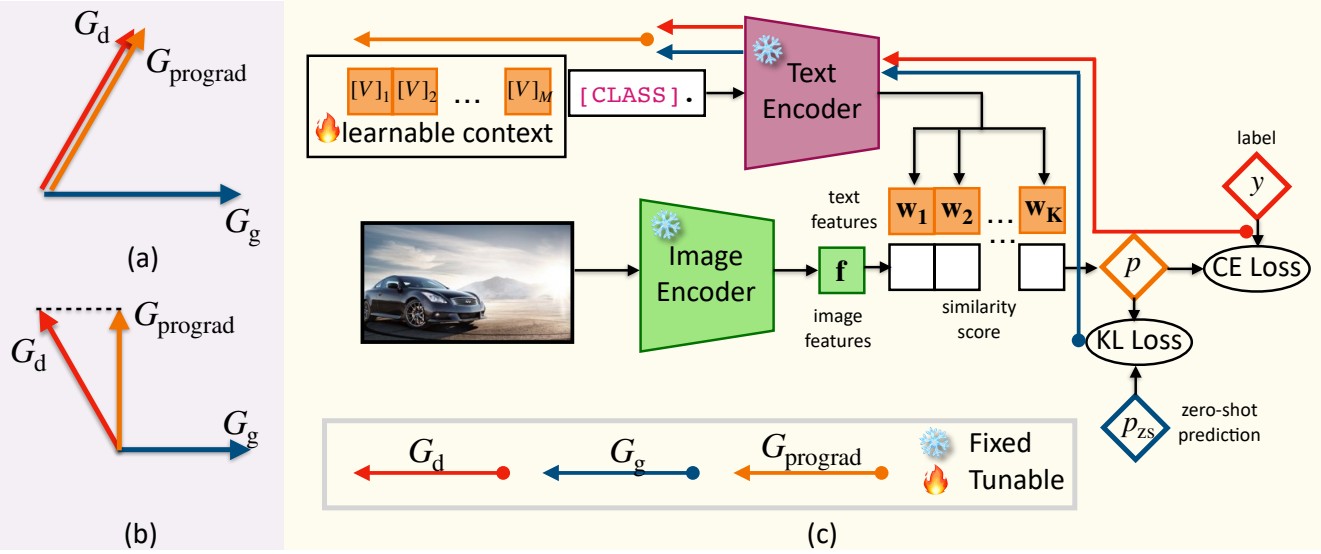

Figure 3: (a) If $\boldsymbol{G}_{\mathrm{d}}$ is aligned with $\boldsymbol{G}_{\mathrm{g}}$, we set $\boldsymbol{G}_{\mathrm{prograd}}$ as $\boldsymbol{G}_{\mathrm{d}}$. (b) If $\boldsymbol{G}_{\mathrm{d}}$ conflicts with $\boldsymbol{G}_{\mathrm{g}}$ (*i.e.*, their angle is larger than 90°), we set $\boldsymbol{G}_{\mathrm{prograd}}$ as the projection of $\boldsymbol{G}_{\mathrm{d}}$ on the orthogonal direction of $\boldsymbol{G}_{\mathrm{g}}$. (c) Training pipeline of our ProGrad. Only the context vectors are learnable.

N_p, we can approximate the generalization error bound as

$$\mathcal{R}_d(\hat{f}_{\mathrm{prograd}}) \leq \frac{1}{2}\gamma_{\mathcal{F}}(S,P) + \mathfrak{R}_d(\mathcal{F}) + \frac{3}{2}\sqrt{\frac{\ln(4/\epsilon)}{2N_d}} + \frac{1}{2}\sqrt{\frac{\ln(4/\epsilon)}{2}\frac{1}{N_d}}. \quad (7)$$

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

$$\mathcal{R}_d(\hat{f}_{\mathrm{coop}}) \leq 2\mathfrak{R}_d(\mathcal{F}) + 3\sqrt{\frac{\ln(4/\epsilon)}{2N_d}} + \sqrt{\frac{\ln(4/\epsilon)}{2}\frac{1}{N_d}}. \quad (8)$$

Under the assumption that the gap between pre-trained and downstream domains $\gamma(P,D)$ is small, the estimation error bound of $\mathcal{R}_d(\hat{f}_{\mathrm{coop}})$ is at least two times greater than $\mathcal{R}_d(\hat{f}_{\mathrm{prograd}})$. Considering that N_d is typically very small in few-shot setting, our ProGrad model $\hat{f}_{\mathrm{prograd}}$ achieves a much lower error bound than conventional fine-tuning model like CoOp $\hat{f}_{\mathrm{coop}}$.

4. Experiments

4.1. Datasets and Implementation Details

We validate the effectiveness of ProGrad on four settings: (1) few-shot classification, (2) domain generalization, (3) base-to-new generalization, (4) cross-dataset transfer.

Datasets. For few-shot learning, base-to-new generalization and cross-dataset transfer, we use 11 datasets, *i.e.*, ImageNet [6] and Caltech101 [9] for generic object classification, OxfordPets [33], StanfordCars [22], Flowers102 [32], Food101 [2] and FGVCAircraft [30] for fine-grained image recognition, EuroSAT [13] for satellite image classification, UCF101 [47] for action classification, DTD [5] for texture classification, and SUN397 [50] for scene recognition. For domain generalization, we use ImageNet as the source dataset and select ImageNet-V2 [41], ImageNet-Sketch [49], ImageNet-A [15], ImageNet-R [14] as the target datasets.

Training Details. For few-shot learning, following CoOp and CLIP, all models are trained with {1, 2, 4, 8, 16} shots respectively then evaluated on the full test split. For domain generalization and base-to-new generalization, we evaluate 4-shot performance, which justifies the robustness under low-shots condition. All results of learning-based models are averaged over three random seeds. The standard deviation values can be found in the Appendix. Unless otherwise stated, we adhere to CoOp to use ResNet-50 [12] as the backbone of image encoder. Following [54] and [55], the length of context tokens M is set to 16 for few-shot classification and $M = 4$ for the other three settings. λ is set to 1 by default, except that λ is set to 0.8 for 16 shots. We adopt the training settings from CoOp, *e.g.*, training epochs, training schedule and the data augmentation settings, and refer readers to Appendix for further details

Baselines. We compare ProGrad against 4 methods: (1) Zero-shot CLIP (2) Linear probe (3) CoOp and (4) CoCoOp. Although our method can beat some other fine-tune methods, *e.g.*, CLIP-Adapter [10], we focus on *single prompt-based learning* methods. The results of other fine-tuning methods are in Appendix.

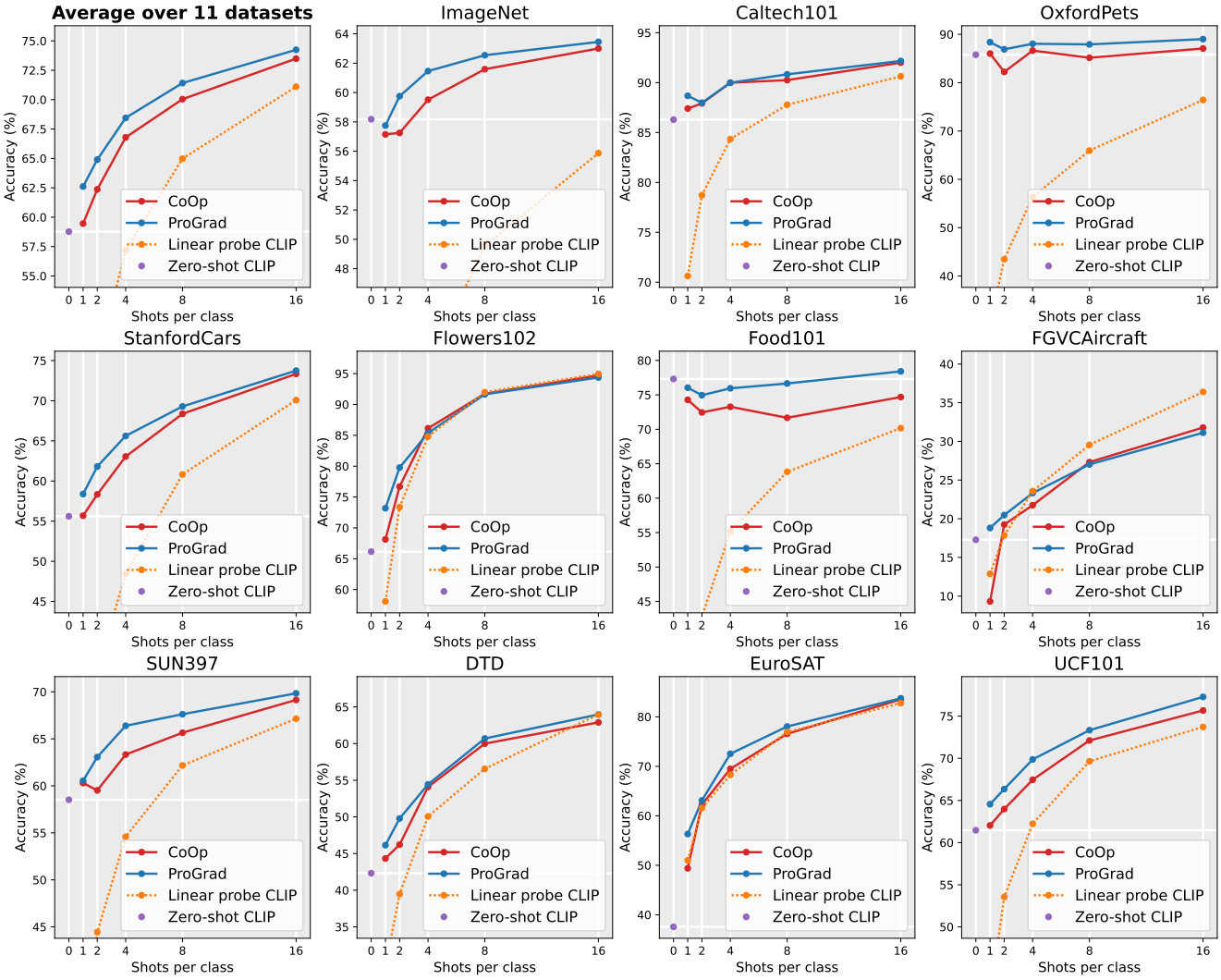

Figure 4: Accuracy (%) of few-shot learning on 11 datasets. The context length M is set to 16. Standard deviations are reported in Appendix.

4.2. Few-Shot Classification

Setup. We compare with two zero-shot CLIP models, *i.e.*, CLIP and CLIP++ stand for using single prompt and prompt ensembling respectively (Please refer to Appendix for the prompts templates). `ProGrad` and `ProGrad++` stand for using single prompt and prompt ensembling as general knowledge to implement ProGrad respectively. Note that we only use the hand-crafted prompt ensembling to generate $\boldsymbol{G}_{\text{g}}$, which provides a more accurate general direction. Therefore, `ProGrad++` still optimizes a *single* prompt with 16 learnable tokens, which is identical to the size of CoOp.

Table 1 provides averaged accuracy over 11 datasets, and Figure 4 illustrates the detailed comparisons. Overall, `ProGrad` clearly outperforms over baselines on average performance. Specifically, `ProGrad` outperforms CoOp by 9.5%, 6.9% and 5.1% on FGVCAircraft, EuroSAT and Flowers102 given 1 shot, and the average improvement is 3.2%. These results demonstrate the anti-overfitting ability of our `ProGrad` when the training samples are extremely

Table 1: Averaged accuracy (%) of few-shot learning on 11 datasets. $M = 16$.

#shots	0	1	2	4	8	16
CLIP	58.77	-	-	-	-	-
CLIP++	**59.38**	-	-	-	-	-
LP	-	37.32	48.02	57.27	64.88	70.57
CoOp	-	59.46	62.37	66.79	70.04	73.49
`ProGrad`	-	62.61	64.90	68.45	71.41	74.28
`ProGrad++`	-	**63.06**	**65.28**	**68.71**	**71.80**	**75.03**

Table 2: Evaluation on robustness to distribution shift with different visual backbones. Standard deviations are reported in Appendix.

(a) ResNet50

	Source	Target			
	ImageNet	-V2	-Sketch	-A	-R
CLIP	58.18	51.34	33.32	21.65	56.00
LP	41.29	33.65	13.09	11.18	26.82
CoOp	61.34	53.81	32.83	22.08	54.62
CoCoOp	61.04	53.71	32.30	22.07	53.60
ProGrad	**62.17**	**54.70**	**34.40**	**23.05**	**56.77**

(b) ResNet101

	Source	Target			
	ImageNet	-V2	-Sketch	-A	-R
CLIP	61.24	54.82	38.66	28.03	64.34
LP	47.01	38.46	19.09	16.33	39.43
CoOp	63.99	56.99	39.40	29.50	64.04
CoCoOp	63.59	56.98	39.16	29.09	64.14
ProGrad	**64.98**	**57.86**	**40.53**	**30.13**	**65.61**

(c) ViT-B/32

	Source	Target			
	ImageNet	-V2	-Sketch	-A	-R
CLIP	62.00	54.75	40.82	29.59	66.01
LP	46.77	39.12	20.32	16.32	39.48
CoOp	64.74	56.59	40.03	31.10	64.54
CoCoOp	64.63	56.59	40.74	30.27	64.12
ProGrad	**65.36**	**57.42**	**41.73**	**31.89**	**66.53**

(d) ViT-B/16

	Source	Target			
	ImageNet	-V2	-Sketch	-A	-R
CLIP	66.73	60.84	46.13	47.80	74.01
LP	54.70	45.57	28.20	22.47	44.12
CoOp	69.86	62.83	46.90	48.98	74.55
CoCoOp	70.13	63.05	46.48	49.36	73.80
ProGrad	**70.45**	**63.35**	**48.17**	**49.45**	**75.21**

limited. Furthermore, leveraging prompt ensembling can further explore the potential of ProGrad. From Table 1, with more accurate general knowledge offered by prompt ensembling, CLIP++ improves the zero-shot CLIP from 58.77% to 59.38%; ProGrad++ increases the accuracy of ProGrad from 74.28% to 75.03% at 16 shots.

4.3. Domain Generalization

We train our ProGrad on the source dataset (ImageNet) for three different random seeds, and assess it on ImageNet-V2, ImageNet-Sketch, ImageNet-A, and ImageNet-R. This setting evaluates the generalizability on a target domain which differs from the source domain. Fine-tuning on limited data from a specific domain may mislead the model to learn spurious correlations or in-distribution patterns, resulting in biased models with poor performance in unseen domains. In contrast, zero-shot CLIP avoids exploiting such spurious correlations or patterns, as it is not learned on that distribution. By leveraging knowledge from the pre-trained domain to regularize fine-tuning on a specific distribution, our ProGrad method is expected to be robust to distribution shifts. As shown in Table 2, despite the exposure to the source dataset, ProGrad clearly outperforms other methods on all target datasets as well as the source dataset with ResNet-based and ViT-based backbones.

4.4. Base-to-New Generalization

We follow CoCoOp [55] to evaluate the generalization performance from seen classes to unseen classes. All the classes are equally divided into two groups, *i.e.*, base classes and new classes, and all methods are only trained on base classes and tested on both base classes and new classes. The harmonic mean of base classes and new classes accuracies is reported to evaluate the trade-off. As illustrated in Table 3, ProGrad yields the highest average performance across all metrics, whereas CoOp and CoCoOp exhibit poor performance for new classes, consistently underperforming zero-shot CLIP. These results highlight that ProGrad's superior generalizability to both base and new classes. The detailed results of 11 datasets are in Appendix.

Table 3: Averaged accuracy (%) over 11 datasets for base-to-new generalization.

	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**

4.5. Cross-Dataset Transfer

All models are trained on ImageNet as source dataset and evaluated on the rest 10 target datasets. The goal of this setting is to demonstrate the potential to transfer beyond a single dataset. The results are presented in Table 4. As shown, our ProGrad not only achieves the highest performance on source datasets but also outperforms other baselines 9 out of 10 target datasets.

4.6. Further Analysis

Comparison with Learning without Forgetting (LwF). ProGrad employs the gradient direction of knowledge distillation loss as a form of regularization. To determine whether our approach is equivalent to conventional knowledge distillation, we compared its performance against a simple knowledge distillation method, *i.e.*, $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{ce}} + \alpha \cdot \mathcal{L}_{\text{kd}}$, which is identical to the implementation of Learning without

Table 4: **Comparison of prompt learning methods in the cross-dataset transfer setting**. Prompts are learned from 4-shots ImageNet.

	Source	Target										
	ImageNet	Caltech101	OxfordPets	StanfordCars	Flowers102	Food101	FGVCAircraft	SUN397	DTD	EuroSAT	UCF101	Average
CoOp	61.34	84.48	85.99	54.16	60.10	75.48	14.09	57.48	35.32	**26.72**	57.56	55.70
CoCoOp	61.04	84.73	86.42	52.34	61.24	73.79	13.74	55.94	36.60	23.46	57.97	55.21
ProGrad	**62.17**	**88.30**	**86.43**	**55.61**	**62.69**	**76.76**	**15.76**	**60.16**	**39.48**	24.87	**58.70**	**57.36**

Table 5: Comparison with LwF (knowledge distillation). Average accuracy (%) over 11 datasets.

#shots	1	2	4	8	16
CoOp	59.46	62.37	66.79	70.04	73.49
LwF(KD), $\alpha = 0.25$	61.09	63.01	67.74	70.90	73.39
LwF(KD), $\alpha = 0.5$	61.13	63.36	67.14	70.34	72.68
LwF(KD), $\alpha = 1$	61.52	64.07	66.52	70.01	72.01
LwF(KD), $\alpha = 2$	60.98	62.66	64.92	67.78	68.98
LwF(KD), $\alpha = 4$	59.58	61.76	62.92	65.01	65.42
ProGrad	**62.61**	**64.90**	**68.45**	**71.41**	**74.28**

Table 6: Applying ProGrad to cosine classifier. Average accuracy (%) over 11 datasets.

#shots	1	2	4	8	16
Cosine	30.50	43.74	53.33	61.26	65.00
+ ProGrad	**32.29**	**46.14**	**55.18**	**62.05**	**66.47**

Forgetting (LwF)[24]. We repeated few-shot experiments on 11 datasets using a range of α values and report the average results in Table 5. The results demonstrate that ProGrad outperforms KD for various few-shot settings. Although KD (LwF) with small $\alpha \leq 1$ improves the performance of CoOp in low-shot scenarios (*e.g.*, 1, 2, and 4 shots), its performance drops when the shots is large (*i.e.*, 8 and 16 shots). These findings suggest that ProGrad operates differently from KD (LwF) and is more robust to the sample number.

Failure cases. We analyze cases where ProGrad models fail while CoOp succeeds. In specific, we count the percentage of the failure cases that zero-shot CLIP models also fails in Figure 5. We found that a high proportion of the failure cases are also mis-classified by zero-shot CLIP model (red bar in Figure 5), implying that the imprecision of zero-shot general knowledge represented by $\boldsymbol{G}_{\text{g}}$ is detrimental to model generalization. As sample size increases, $\boldsymbol{G}_{\text{d}}$ represents downstream knowledge more accurately and with less bias. As expected, the red bar becomes larger.

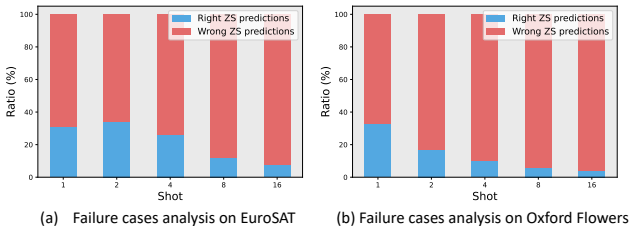

Figure 5: Distribution of samples that are mis-classified by ProGrad but correctly classified by CoOp.

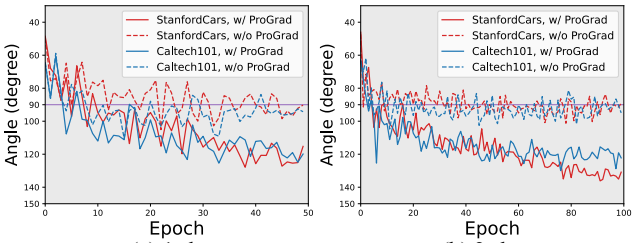

Figure 6: Angles between $\boldsymbol{G}_{\text{d}}$ and $\boldsymbol{G}_{\text{g}}$ during training on StanfordCars and Caltech101.

Conflict of knowledge. ProGrad requires the updated gradient direction be acute to the general knowledge gradient directions. We explore how this constraint helps to defuse the conflicts of domain-specific and general knowledge by visualizing the angle between their representative gradients during training (angle between $\boldsymbol{G}_{\text{d}}$ and $\boldsymbol{G}_{\text{g}}$). As depicted in Figure 6, without $\boldsymbol{G}_{\text{prograd}}$, the angle between $\boldsymbol{G}_{\text{d}}$ and $\boldsymbol{G}_{\text{g}}$ converges to 90 degree due to the fact that "all high-dimensional random vectors are almost always orthogonal to each other" [3]. Intuitively, without any constraint, the optimization direction $\boldsymbol{G}_{\text{d}}$ is independent to the general direction, and the average angle would be orthogonal. In contrast, utilizing $\boldsymbol{G}_{\text{prograd}}$ results in the convergence of the angle to an obtuse angle. The reason is that $\boldsymbol{G}_{\text{prograd}}$ intervenes the model to learn the downstream knowledge aligned with the

general knowledge and leads to the insufficient learning of downstream knowledge that is incompatible with the general knowledge. As training stabilizes, $\boldsymbol{G}_{\mathrm{d}}$ struggles to learn the conflicting knowledge, reflecting an obtuse angle to the $\boldsymbol{G}_{\mathrm{g}}$. Thanks to ProGrad, we discard such conflicting knowledge to avoid forgetting.

Upper Bound of Performance. As the general gradient direction $\boldsymbol{G}_{\mathrm{g}}$ is the key for improvement, we are interested in the upper-bound of performance if we can find an oracle general direction $\boldsymbol{G}_g^{\mathrm{full}}$ instead of the one offered by hand-crafted prompt. To achieve this, we first optimize a prompt with plain cross-entropy loss on the full dataset to create $\boldsymbol{G}_g^{\mathrm{full}}$ and then use such gradient to implement ProGrad. The averaged performance over 11 datasets, as shown in Figure 7, indicate a more accurate regularization direction $\boldsymbol{G}_g^{\mathrm{full}}$ elicits a stronger ProGrad model. The detailed results of 11 datasets are in Appendix.

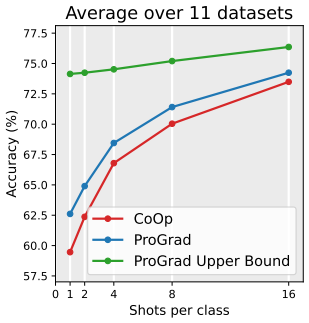

Figure 7: The upper bound of performance (averaged accuracy) of GradPrad.

Applying ProGrad to conventional fine-tuning. We are also interested in the effectiveness of ProGrad for the conventional "pre-train then fine-tune" paradigm. Specifically, we plug in an additional cosine classifier on top of the visual backbone and compare the averaged performance over 11 datasets of few-shot classification. Table 6 shows that conventional fine-tuning can benefit from our ProGrad. The implementation details and the result of each dataset are provided in Appendix.

Effect of hyper-parameter λ. We further analyze the effect of the hyper-parameter λ described in Eq. (4) in the main paper. Results are shown in Table. 7. As discussed in Section 3.2 in the main paper, a smaller λ weakens the general knowledge regularization, which results in a inferior performance under low-shot setting for most datasets. However, for DTD in Table 7, using a smaller $\lambda = 0.9$ to reduce the general knowledge regularization can improve the 16 shots results. One possible reason is that texture images of DTD has large gap with the CLIP pre-trained images that collected from the Internet, stronger regularization from pre-trained knowledge might be detrimental to the fine-tune performance if downstream data is sufficient.

5. Conclusion

In this paper, we pointed out the over-fitting issues of existing prompt tuning methods for few-shot generalization, which heavily relies on early stopping and data augmentation. We proposed a prompt tuning method ProGrad that regularize each tuning step not to conflict with the general knowledge of the hand-crafted prompt. Experiments on few-shot classification, base-to-new generalization, domain generalization and cross-dataset transfer over 11 datasets demonstrate the effectiveness and efficiency of our ProGrad. In the future, we will explore how to apply ProGrad on other tasks like object detection and segmentation.

Table 7: Accuracy (%) of 1, 2, 4, 8, and 16 shots training with different λ on DTD and OxfordPets.

(a) **OxfordPets**.

λ	1 shot	2 shots	4 shots	8 shots	16 shots
0	86.01	82.21	86.63	85.15	87.06
0.2	87.12	83.16	84.87	84.00	86.15
0.4	88.09	83.56	85.50	84.04	86.67
0.7	87.74	84.70	86.93	86.30	87.90
0.9	88.26	86.47	87.52	87.38	88.52
1.0	88.36	86.89	88.04	87.91	89.00

(b) **DTD**.

λ	1 shot	2 shots	4 shots	8 shots	16 shots
0	44.33	46.22	54.08	59.99	62.89
0.2	43.80	45.21	54.02	60.14	63.61
0.4	44.17	47.44	54.32	59.65	63.16
0.7	44.93	47.77	54.92	59.28	63.10
0.9	45.78	48.46	55.44	60.46	64.28
1.0	46.14	49.78	54.43	57.98	61.15

Acknowledgments

This research is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG2-PhD-2021-01-002). The authors would like to thank the reviewers for their comments that help improve the manuscript.

References

[1] Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002. 4

[2] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *ECCV*, 2014. 5

[3] Tony Cai, Jianqing Fan, and Tiefeng Jiang. Distributions of angles in random packing on spheres. *Journal of Machine Learning Research*, 14(21):1837–1864, 2013. 8

[4] Guangyi Chen, Weiran Yao, Xiangchen Song, Xinyue Li, Yongming Rao, and Kun Zhang. Prompt learning with optimal transport for vision-language models. *arXiv preprint arXiv:2210.01253*, 2022. 3

[5] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *CVPR*, 2014. 5

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 5

[7] Mohammad Mahdi Derakhshani, Enrique Sanchez, Adrian Bulat, Victor Guilherme Turrisi da Costa, Cees GM Snoek, Georgios Tzimiropoulos, and Brais Martinez. Variational prompt tuning improves generalization of vision-language models. *arXiv preprint arXiv:2210.02390*, 2022. 3

[8] Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. Orthogonal gradient descent for continual learning. In *International Conference on Artificial Intelligence and Statistics*, 2020. 3

[9] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *CVPRW*, 2004. 5

[10] 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. 3, 5

[11] Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. In *ACL*, 2021. 1

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5

[13] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2019. 5

[14] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *ICCV*, 2021. 5

[15] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *CVPR*, 2021. 5

[16] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015. 4

[17] Xinting Hu, Kaihua Tang, Chunyan Miao, Xian-Sheng Hua, and Hanwang Zhang. Distilling causal effect of data in class-incremental learning. In *CVPR*, 2021. 3

[18] 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 *ICML*, 2021. 1

[19] 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. In *ACL*, 2022. 1

[20] Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Maple: Multi-modal prompt learning. *arXiv preprint arXiv:2210.03117*, 2022. 3

[21] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In *ICML*, 2021. 3

[22] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCVW*, 2013. 5

[23] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *EMNLP*, 2021. 1, 3

[24] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017. 8

[25] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*, 2021. 1

[26] Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. Gpt understands, too. *arXiv preprint arXiv:2103.10385*, 2021. 1, 3

[27] Yaoyao Liu, Yuting Su, An-An Liu, Bernt Schiele, and Qianru Sun. Mnemonics training: Multi-class incremental learning without forgetting. In *CVPR*, 2020. 3

[28] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *NeurIPS*, 2019. 3

[29] Yuning Lu, Jianzhuang Liu, Yonggang Zhang, Yajing Liu, and Xinmei Tian. Prompt distribution learning. In *CVPR*, 2022. 3

[30] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 5

[31] Alfred Müller. Integral probability metrics and their generating classes of functions. *Advances in Applied Probability*, 29(2):429–443, 1997. 4

[32] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, 2008. 5

[33] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *CVPR*, 2012. 5

[34] Mary Phuong and Christoph Lampert. Towards understanding knowledge distillation. In *ICML*, 2019. 4

[35] Chengwei Qin and Shafiq Joty. Continual few-shot relation learning via embedding space regularization and data augmentation. In *ACL*, 2022. 3

[36] Chengwei Qin and Shafiq Joty. Lfpt5: A unified framework for lifelong few-shot language learning based on prompt tuning of t5. In *ICLR*, 2022. 1

[37] 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. 1, 3

[38] Alexandre Rame, Corentin Dancette, and Matthieu Cord. Fishr: Invariant gradient variances for out-of-distribution generalization. *arXiv preprint arXiv:2109.02934*, 2021. 3

[39] Yongming Rao, Wenliang Zhao, Guangyi Chen, Yansong Tang, Zheng Zhu, Guan Huang, Jie Zhou, and Jiwen Lu. Denseclip: Language-guided dense prediction with context-aware prompting. *arXiv preprint arXiv:2112.01518*, 2021. 3

[40] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *CVPR*, 2017. 3

[41] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *ICML*, 2019. 5

[42] Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. In *ICLR*, 2018. 3

[43] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *CVPR*, 2017. 2

[44] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *NeurIPS*, 2018. 3

[45] Yuge Shi, Jeffrey Seely, Philip Torr, Siddharth N, Awni Hannun, Nicolas Usunier, and Gabriel Synnaeve. Gradient matching for domain generalization. In *ICLR*, 2022. 3

[46] Manli Shu, Weili Nie, De-An Huang, Zhiding Yu, Tom Goldstein, Anima Anandkumar, and Chaowei Xiao. Test-time prompt tuning for zero-shot generalization in vision-language models. *arXiv preprint arXiv:2209.07511*, 2022. 3

[47] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 5

[48] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *EMNLP-IJCNLP*, 2019. 3

[49] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. *NeurIPS*, 2019. 5

[50] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010. 5

[51] 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*, 2021. 1, 3

[52] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *NeurIPS*, 2020. 3, 4

[53] Chao Zhang, Lei Zhang, and Jieping Ye. Generalization bounds for domain adaptation. *NeurIPS*, 2012. 4

[54] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *arXiv preprint arXiv:2109.01134*, 2021. 1, 3, 4, 5

[55] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *CVPR*, 2022. 1, 2, 3, 5, 7

[56] Beier Zhu, Yulei Niu, Saeil Lee, Minhoe Hur, and Hanwang Zhang. Debiased fine-tuning for vision-language models by prompt regularization. *AAAI*, 2023. 3