Enhancing Vision-Language Few-Shot Adaptation with Negative Learning

Supplementary Material

In appendix, we provide additional details and experimental results to enhance understanding and insights into our proposed SimNL. This supplementary document is organized as follows:

- Full numerical results for the few-shot learning task are detailed in Section A.1.
- We compare our SimNL with other state-of-the-art methods in domain generalization tasks, utilizing an alternative ViT backbone, in Section A.2.
- More sensitivity analyses of the hyper-parameters are conducted in Section A.3.
- We differentiate our method from related work in Section B.
- Detailed statistics for all utilized datasets are provided in Section C.1.
- We present the specific positive and negative prompts we used for each dataset in Section C.2.
- We list the license information for all used assets in Section D.
- Finally, we explore potential future work and discuss the limitations and broader impacts of this work in Section E.

A. Additional Experimental Results

A.1. Full Numerical Results on Few-Shot Learning

In Figure 5 of the main text, we have evaluated our SimNL on few-shot learning task and compared with other state-of-the-art methods. In Table A1, we present the corresponding full numerical results on the few-shot learning task. We also report the 95% confidence interval over 3 random seeds of our SimNL to ensure reliability of our results. In the last column, we present the average recognition accuracy over 11 datasets. The results indicate that our SimNL consistently outperforms other state-of-the-art methods across various few-shot learning settings by substantial margins.

Our SimNL demonstrates superior recognition performance across nearly all tested scenarios, with certain exceptions in lower-shot settings for the Food101 [2] and OxfordPets [35] datasets. We attribute this to the prevalent challenge of overfitting, a common issue not exclusive to our approach but also affecting many existing methods, especially TaskRes [65]: While TaskRes consistently secures the second-best performance across the other 9 datasets, it underperforms significantly on these two. We hypothesize that this issue arises from the noisy training data with *intense colors and sometimes wrong labels* [2, 35]. However, in this work, we have designed a label refinement mechanism specifically to address this issue. As a result, our SimNL secures a relatively robust performance and achieves the second-best on these two datasets.

A.2. More Results on Domain Generalization

In Table 1 in Section 4.2 of the main text, we compare the generalization performance of our SimNL with other state-of-the-art methods in the presence of distribution shifts with ResNet-50 visual backbone. In Table A2, we present the performance comparison on domain generalization task using ViT-B/32 visual backbone. Consistently, our SimNL not only achieves state-of-the-art performance on the source dataset but also attains an average performance gain of 0.57% across 4 out-of-distribution (OOD) target datasets. This verifies that our SimNL demonstrates superior generalizability compared to other state-of-the-art methods, independent of the visual backbone utilized.

A.3. More Sensitivity Analyses of Hyper-Parameters

Building upon the sensitivity analyses of λ and τ detailed in Section 4.3 of the main text, this section extends our examination to include the sensitivity of parameters α and β on 16-shot ImageNet [6]. In our experiments on ImageNet [6], we set the hyper-parameters α and β defined in Section 3 to 1.2 and 2.0, respectively. To comprehensively investigate the effects of different hyper-parameters, we conducted a sensitivity experiment where we varied each hyper-parameter individually and evaluated the performance on 16-shot ImageNet [6] in Fig. A3. We can see that our choice of $\alpha = 1.2$ and $\beta = 2.0$ yields the highest performance. Moreover, our SimNL maintains robust performance when adjusting these two hyper-parameters, since our SimNL includes adapters from both textual and visual modalities and each of our four adapters can work effectively, as we presented in Table 4 in the main text.

B. More Discussions on Related Work

B.1. Differences between Our Proposed Method and Traditional Negative Learning

In this work, we apply the concept of negative learning to vision-language few-shot adaptation. However, our method is different with traditional negative learning approaches [20, 66]. Specifically, traditional negative learning approaches aim to learn a positive classifier using negatibe

Table A1. **Full numerical results on few-shot learning task**. For each dataset, we report the mean accuracy and 95% confidence interval over 3 random seeds of our SimNL on 1-/2-/4-/8-/16-shot settings. †We report the zero-shot performance of CLIP [38] for all settings. For TaskRes [65], we report the results using the enhanced base classifier (*i.e.*, TaskRes*). The best results are in **bold** and the second are underlined.

Method	Setting	Caltech101 [9]	DTD [5]	EuroSAT [14]	FGVCAircraft [30]	Flowers102 [33]	Food101 [2]	ImageNet [6]	OxfordPets [35]	StanfordCars [24]	SUN397 [63]	UCF101 [48]	Avg.
†Zero-shot CLIP [38]	1-shot	84.52	40.33	41.80	16.98	65.46	77.31	60.33	85.51	54.26	58.56	61.44	58.77
CoOp [74]	1-shot	87.43	44.13	50.51	9.80	67.90	73.71	57.15	86.51	55.48	60.10	62.10	59.53
CoCoOp [73]	1-shot	86.01	45.14	35.08	17.81	67.52	77.42	60.84	**86.96**	57.22	62.28	62.84	59.92
CLIP-Adapter [10]	1-shot	88.70	45.66	61.51	17.21	73.43	76.77	61.20	85.99	55.14	61.28	62.29	62.65
Tip-Adapter-F [71]	1-shot	89.38	50.31	59.16	20.83	80.13	77.61	61.32	86.47	58.51	62.51	64.91	64.65
TaskRes [65]	1-shot	88.80	50.20	61.70	21.41	79.17	74.03	61.90	83.60	59.13	62.33	64.77	64.28
SimNL (Ours)	1-shot	**90.87**	**53.13**	**67.70**	**23.61**	**84.17**	**77.93**	**62.89**	86.90	**61.34**	**65.13**	**68.25**	**67.45**
		(±0.33)	(±0.48)	(±0.47)	(±0.18)	(±0.34)	(±0.22)	(±0.13)	(±0.31)	(±0.28)	(±0.08)	(±0.23)	(±0.28)
†Zero-shot CLIP [38]	2-shot	84.52	40.33	41.80	16.98	65.46	77.31	60.33	85.51	54.26	58.56	61.44	58.77
CoOp [74]	2-shot	87.92	45.04	60.43	18.25	77.47	72.26	55.88	82.36	58.10	59.82	64.13	61.97
CoCoOp [73]	2-shot	89.47	46.20	38.51	20.22	70.70	**78.81**	61.86	**88.81**	58.28	63.50	65.23	61.96
CLIP-Adapter [10]	2-shot	89.32	51.81	64.11	20.10	81.77	77.20	61.52	86.73	58.71	62.21	67.27	65.52
Tip-Adapter-F [71]	2-shot	89.81	54.00	65.82	23.47	82.50	77.83	61.69	87.10	62.05	63.55	66.23	66.73
TaskRes [65]	2-shot	90.27	55.13	65.83	24.13	86.57	75.17	62.63	84.63	63.70	64.97	70.00	67.54
SimNL (Ours)	2-shot	**91.19**	**59.17**	**74.40**	**26.22**	**88.43**	78.35	**63.47**	87.68	**64.77**	**66.73**	**70.25**	**70.06**
		(±0.26)	(±0.31)	(±0.87)	(±0.37)	(±0.18)	(±0.17)	(±0.06)	(±0.28)	(±0.21)	(±0.33)	(±0.48)	(±0.31)
†Zero-shot CLIP [38]	4-shot	84.52	40.33	41.80	16.98	65.46	77.31	60.33	85.51	54.26	58.56	61.44	58.77
CoOp [74]	4-shot	89.17	53.38	70.20	21.72	85.81	72.72	59.93	87.22	61.92	63.46	67.08	66.60
CoCoOp [73]	4-shot	90.31	47.90	63.56	20.56	72.72	**79.51**	62.52	**88.60**	59.90	64.90	67.90	65.31
CLIP-Adapter [10]	4-shot	89.98	57.02	73.18	22.99	87.30	77.93	61.84	87.36	62.26	65.90	68.90	68.61
Tip-Adapter-F [71]	4-shot	90.67	57.78	73.85	26.01	89.02	78.26	62.52	87.72	64.82	66.13	70.87	69.79
TaskRes [65]	4-shot	90.97	60.70	73.83	25.70	90.20	76.10	63.57	86.33	67.43	67.27	70.93	70.28
SimNL (Ours)	4-shot	**92.21**	**66.01**	**76.54**	**28.95**	**92.04**	78.74	**64.12**	88.13	**67.96**	**68.59**	**73.46**	**72.43**
		(±0.21)	(±0.49)	(±0.59)	(±0.29)	(±0.25)	(±0.11)	(±0.17)	(±0.26)	(±0.36)	(±0.29)	(±0.51)	(±0.32)
†Zero-shot CLIP [38]	8-shot	84.52	40.33	41.80	16.98	65.46	77.31	60.33	85.51	54.26	58.56	61.44	58.77
CoOp [74]	8-shot	90.15	59.88	76.51	25.93	90.84	71.52	60.91	86.40	68.49	65.63	71.81	69.82
CoCoOp [73]	8-shot	90.14	52.21	64.13	22.03	75.88	**79.59**	62.40	88.74	60.87	65.37	68.25	66.33
CLIP-Adapter [10]	8-shot	91.22	60.70	78.34	25.77	91.79	78.01	62.68	87.70	67.78	67.52	73.02	71.32
Tip-Adapter-F [71]	8-shot	91.54	62.67	77.83	30.21	91.85	78.71	64.00	88.07	69.53	68.80	74.50	72.52
TaskRes [65]	8-shot	92.40	64.77	79.33	31.48	94.73	76.40	64.67	87.17	71.83	68.73	75.33	73.35
SimNL (Ours)	8-shot	**93.40**	**67.78**	**81.62**	**33.90**	**95.23**	79.23	**65.37**	**89.29**	**72.08**	**70.93**	**76.84**	**75.06**
		(±0.23)	(±0.55)	(±0.54)	(±0.63)	(±0.26)	(±0.21)	(±0.09)	(±0.21)	(±0.51)	(±0.15)	(±0.39)	(±0.34)
†Zero-shot CLIP [38]	16-shot	84.52	40.33	41.80	16.98	65.46	77.31	60.33	85.51	54.26	58.56	61.44	58.77
CoOp [74]	16-shot	91.61	63.11	82.36	31.01	94.39	73.80	62.95	87.30	72.51	69.11	75.70	73.07
CoCoOp [73]	16-shot	90.90	57.53	70.77	22.40	79.14	79.68	62.71	89.93	62.22	67.21	70.81	68.48
CLIP-Adapter [10]	16-shot	92.44	66.14	82.76	31.83	93.91	78.21	63.59	87.91	74.12	69.59	76.80	74.30
Tip-Adapter-F [71]	16-shot	92.93	67.33	83.80	35.50	95.01	79.50	65.51	89.71	75.50	71.31	78.01	75.83
TaskRes [65]	16-shot	93.43	67.13	84.03	36.30	96.03	77.60	65.73	87.83	76.83	70.67	77.97	75.78
SimNL (Ours)	16-shot	**93.77**	**70.83**	**87.36**	**40.27**	**96.51**	**79.87**	**66.52**	**90.58**	**77.48**	**72.32**	**80.28**	**77.80**
		(±0.33)	(±0.77)	(±0.84)	(±0.53)	(±0.58)	(±0.34)	(±0.13)	(±0.38)	(±0.68)	(±0.24)	(±0.26)	(±0.50)

Table A2. **Performance comparison on robustness to distribution shifts**. All the models are trained on 16-shot ImageNet [6] and directed tested on the OOD target datasets. The best results are in **bold** and the second best are underlined.

Method	Source	Target				
	ImageNet	-V2	-Sketch	-A	-R	Avg.
Zero-Shot CLIP [38]	62.05	54.79	40.82	29.57	65.99	47.79
Linear Probe CLIP [38]	59.58	49.73	28.06	19.67	47.20	36.17
CoOp [74]	66.85	58.08	40.44	30.62	64.45	48.40
TaskRes [65]	68.20	59.20	42.50	31.43	69.33	50.62
GraphAdapter [25]	68.47	59.10	42.70	31.73	69.43	50.74
SimNL (Ours)	**69.63**	**59.76**	**43.41**	**32.48**	**69.60**	**51.31**

Table A3. **Sensitivity of hyper-parameters**. All the results are reported on 16-shot ImageNet [6].

α	0.0	0.5	1.0	**1.2**	1.5	2.0
	66.14	66.30	66.41	**66.52**	66.44	66.36
β	1.0	1.5	**2.0**	2.5	3.0	3.5
	66.38	66.44	**66.52**	66.50	66.48	66.40

labels, while our method aim to learn a negative classifier using the positive label.

Negative Learning. In the negative learning setting [20, 66], we are given complementary labels $\bar{y} \in \mathcal{Y}\backslash\{y\}$, which represents the image does not belong to a specific class. Similarly, the encoded one-hot labels are given by $\bar{\boldsymbol{y}} \in \{0,1\}^C$. This leads to a complementary cross-entropy loss for optimizing the classifier parameters θ. The training objective for $\mathcal{F}_\theta$ can thus be expressed as

$$\min_\theta \mathcal{R}(\mathcal{F}_\theta) = \mathbb{E}_{(\boldsymbol{x},\bar{y})\sim P(\boldsymbol{x},\bar{y})}\left[\mathcal{L}\left(\mathcal{F}_\theta(\boldsymbol{x}),\bar{y}\right)\right],$$
$$\text{where } \mathcal{L}\left(\mathcal{F}_\theta(\boldsymbol{x}),\bar{y}\right) = -\sum_{k=1}^{C} \bar{\boldsymbol{y}}_k \log(1-\boldsymbol{p}_k). \tag{13}$$

Here, the goal is to minimize the probability corresponding to the complementary label, *i.e*., $\boldsymbol{p}_{\bar{y}} \to 0$.

Our Method. In this work, we apply the concept of negative learning to vision-language few-shot adaptation by employing a distinct CLIP-based negative classifier $\mathcal{G}_\varphi : \mathcal{X} \to \mathbb{R}^C$ with parameters φ. This classifier predicts the negative probability $\bar{\boldsymbol{p}} = \texttt{Softmax}(\mathcal{G}_\varphi(\boldsymbol{x}))$ that the image does not belong to specific classes. The expected classification risk for $\mathcal{G}_\varphi$ can be written as

$$\min_\varphi \mathcal{R}(\mathcal{G}_\varphi) = \mathbb{E}_{(\boldsymbol{x},y)\sim P(\boldsymbol{x},y)}\left[\mathcal{L}\left(\mathcal{G}_\varphi(\boldsymbol{x}),y\right)\right],$$
$$\text{where } \mathcal{L}\left(\mathcal{G}_\varphi(\boldsymbol{x}),y\right) = -\sum_{k=1}^{C} \boldsymbol{y}_k \log(1-\bar{\boldsymbol{p}}_k). \tag{14}$$

By optimizing this risk, we aim to reduce the negative probability $\bar{\boldsymbol{p}}_y \to 0$ for the true label.

B.2. Differences between Our Proposed Method and Contrastive Learning

In this work, we propose a negative learning-based approach for vision-language few-shot adaptation. However, this "negative" refers to a negative classifier, not the negative samples pairs in contrastive learning literature. In the following discussions, we show that our negative learning is fundamentally different with contrastive learning.

Specifically, CLIP operates as a similarity-based classifier where classification logits are derived from the similarities between image features f_v and class-specific text features $\{f_{t_c}\}_{c=1}^C$:

$$\boldsymbol{p}_y = \frac{\exp\left(\cos\left(f_{t_y}, f_v\right)/t\right)}{\sum_{t'}\exp\left(\cos\left(f_{t'}, f_v\right)/t\right)}, \tag{15}$$

In the context of few-shot adaptation of CLIP, both prompt-based learning methods and adapter-style fine-tuning methods seek to optimize $\{f_{t_c}\}_{c=1}^C$, whether by tuning the input prompt or directly modulating these features. As discussed in Section 3.1, these methods primarily employ positive learning, updating parameters via cross-entropy loss:

$$\begin{aligned}\mathcal{L}\left(\mathcal{F}_\theta(\boldsymbol{x}),y\right) &= -\sum_{k=1}^{C} \boldsymbol{y}_k \log \boldsymbol{p}_k \\ &= -\log\frac{\exp\left(\cos\left(f_{t_y}, f_v\right)/t\right)}{\sum_{t'}\exp\left(\cos\left(f_{t'}, f_v\right)/t\right)},\end{aligned} \tag{16}$$

where y is the ground-truth label for sample $\boldsymbol{x}$. This update strategy aims to optimize $p_y = 1$. By considering f_{t_y} and f_v as a positive pair and $f_{t_{\neq y}}$ and f_v as negative pairs, the positive learning process essentially becomes contrastive in nature. Specifically, Eq. (16) has the same form with the contrastive InfoNCE loss [34], which seeks to minimize the distance between the positive pair f_{t_y} and f_v while maximizing the distance between the negative pairs $f_{t_{\neq y}}$ and f_v.

However, in this work, we apply negative learning and introduce another negative classifier $\mathcal{G}_\varphi$. Specifically, we mine a general negative feature $f_{t_c}^-$ for each class c, which is absent in samples from class c but present in samples from all other classes. Then the probability that the image not belongs to class y can be written as

$$\bar{\boldsymbol{p}}_y = \frac{\exp\left(\cos\left(f_{t_y}^-, f_v\right)/t\right)}{\sum_{t'}\exp\left(\cos\left(f_{t'}^-, f_v\right)/t\right)} \tag{17}$$

As discussed in Section 3.1, we update the set of negative features by

$$\mathcal{L}\left(\mathcal{G}_{\varphi}(\boldsymbol{x}), y\right) = -\sum_{k=1}^{C} \boldsymbol{y}_k \log(1 - \bar{\boldsymbol{p}}_k)$$
$$= -\log\left(1 - \frac{\exp\left(\cos\left(f_{t_y}^{-}, f_v\right)/t\right)}{\sum_{t'} \exp\left(\cos\left(f_{t'}^{-}, f_v\right)/t\right)}\right). \quad (18)$$

In the language of contrastive learning, we consider all $f_{t_{\neq y}}^{-}$ and f_v as positive pairs and $f_{t_y}^{-}$ and f_v as the only negative pair.

In summary, the concept of "negative" is different in our negative learning and contrastive learning: (1) In our negative learning, "negative" specifically refers to a negative classifier. We explicitly train another negative classifier $\mathcal{G}_{\varphi}$ to ensemble with the positive classifier; (2) In contrastive learning, "negative" refers to the negative sample pairs that are constructed and utilized during training. Specifically, by constructing positive and negative sample pairs, the objective of each classifier can be formulated as a contrastive objective as discussed above.

B.3. Differences with Other Similar Approaches

To the best of our knowledge, the following works adopt similar negative learning methods to open-set problems:

- **RPL** [3] is the seminal work that learns a set of reciprocal points as negative representations of target classes to enhance model's recognition capabilities when handling unseen samples.
- **CLIPN** [56] leverages additional large-scale datasets to a fine-tune a "no" text encoder to enhance CLIP's out-of-distribution (OOD) detection capability.
- **LSN** [32] also focus on learning some negative prompts for OOD detection task using CLIP.

However, we formulate negative learning from a different and more fundamental perspective, *i.e.*, using negative prediction to directly improve the accuracy of positive prediction in a closed-set problem. We also empirically validate that leveraging negative cues from CLIP can effectively improve both few-shot classification performance and generalization capability.

C. Additional Technical Details

C.1. Dataset Details

In Table C4, we present the detailed statistics of each dataset we used in our experiments, including the number of classes, the sizes of training, validation and testing sets, and their original tasks.

Table C4. **Detailed statistics of datasets used in experiments**. Note that the last 4 ImageNet variant datasets are designed for evaluation and only contain the test sets.

Dataset	Classes	Training	Validation	Testing	Task
Caltech101 [9]	100	4,128	1,649	2,465	Object recognition
DTD [5]	47	2,820	1,128	1,692	Texture recognition
EuroSAT [14]	10	13,500	5,400	8,100	Satellite image recognition
FGVCAircraft [30]	100	3,334	3,333	3,333	Fine-grained aircraft recognition
Flowers102 [33]	102	4,093	1,633	2,463	Fine-grained flowers recognition
Food101 [2]	101	50,500	20,200	30,300	Fine-grained food recognition
ImageNet [6]	1,000	1.28M	-	50,000	Object recognition
OxfordPets [35]	37	2,944	736	3,669	Fine-grained pets recognition
StanfordCars [24]	196	6,509	1,635	8,041	Fine-grained car recognition
SUN397 [63]	397	15,880	3,970	19,850	Scene recognition
UCF101 [48]	101	7,639	1,898	3,783	Action recognition
ImageNet-V2 [40]	1,000	-	-	10,000	Robustness of collocation
ImageNet-Sketch [55]	1,000	-	-	50,889	Robustness of sketch domain
ImageNet-A [16]	200	-	-	7,500	Robustness of adversarial attack
ImageNet-R [15]	200	-	-	30,000	Robustness of multi-domains

Table C5. **Positive and negative prompts used in experiments**. In addition to these prompts, we also employ CuPL [36] prompts to further enhance performance.

Dataset	Positive Prompts	Negative Prompts
	"itap of a {CLASS}."	"itap without any {CLASS}."
ImageNet [6]	"a bad photo of the {CLASS}."	"a bad photo with no {CLASS} in it."
ImageNet-V2 [40]	"a origami {CLASS}."	"a origami that isn't a {CLASS}."
ImageNet-Sketch [55]	"a photo of the large {CLASS}."	"a photo with no large {CLASS}."
ImageNet-A [16]	"a {CLASS} in a video game."	"a video game scene without a {CLASS}."
ImageNet-R [15]	"art of the {CLASS}."	"art that doesn't include a {CLASS}."
	"a photo of the small {CLASS}."	"a photo with no small {CLASS}."
Caltech101 [9]	"a photo of a {CLASS}."	"a photo without {CLASS}."
DTD [5]	"{CLASS} texture."	"not {CLASS} texture."
EuroSAT [14]	"a centered satellite photo of {CLASS}."	"a centered satellite photo without {CLASS}."
FGVCAircraft [30]	"a photo of a {CLASS}, a type of aircraft."	"a photo without {CLASS}, a type of aircraft."
Flowers102 [33]	"a photo of a {CLASS}, a type of flower."	"a photo without {CLASS}, a type of flower."
Food101 [2]	"a photo of {CLASS}, a type of food."	"a photo without {CLASS}, a type of food."
OxfordPets [35]	"a photo of a {CLASS}, a type of pet."	"a photo without {CLASS}, a type of pet."
StanfordCars [24]	"a photo of a {CLASS}."	"a photo of no {CLASS}."
SUN397 [63]	"a photo of a {CLASS}."	"a photo without {CLASS}."
UCF101 [48]	"a photo of a person doing {CLASS}."	"a photo of a person not doing {CLASS}."

C.2. Positive and Negative Prompts

In Table C5, we detail the specific positive and negative prompts utilized for each dataset. Additionally, as mentioned in Section 4.1, we incorporate prompts from CuPL [36] to further enhance model performance.

D. License Information

Datasets. We list the known license information for the datasets below:

- MIT License: ImageNet-A [16], ImageNet-V2 [40], ImageNet-R [15], and ImageNet-Sketch [55].
- CC BY-SA 4.0 License: OxfordPets [35].
- Research purposes only: ImageNet [6], StandfordCars [24], DTD [5], FGVCAircraft [30], SUN397 [63].

Code. In this work, we also use some code implementations from existing codebase: CLIP [38], CoOp [74], APE [76], and CuPL [36]. The code used in this paper are all under the MIT License.

E. Further Discussions

Future Work. In this work, we introduce negative learning for adapting VLMs to downstream tasks. We believe that the similar concept of *negative* learning can also be applied to prompt-based learning methods, and be extended to fine-tune other foundational models (*e.g*., other VLMs [21, 64] and LLMs [7, 51]). Besides, we also notice that there are a lot of research efforts dedicated to design better prompts (*e.g*., using LLM [31, 36, 62]) to fully exploit the capabilities of CLIP. We hope that with our work, future research endeavors can also be directed to investigate the utilization of negative prompts to better activate the negative inference capabilities, further broadening the scope and effectiveness of CLIP.

Limitations. We identify two potential limitations of our SimNL: (1) Its efficacy in zero-shot situations is constrained due to the scarcity of original negative descriptions in the CLIP training corpus; (2) Similar to other adapter-style fine-tuning approaches, our SimNL fine-tuned on a specific task cannot be directly applied to another task without additional adaptation. However, recently, Wang *et al*. [59] shows that adapter-style fine-tuning methods can be extended for these scenarios using the kNN algorithm.

Broader Impacts. In this work, we aim to build more reliable machine learning systems by leveraging the extensive knowledge of current foundational models. Specifically, we introduce negative learning to more efficiently transfer pre-trained VLMs to downstream tasks, enhancing both the task-specific performance of CLIP and its robustness to natural distribution shifts. Additionally, we explore the few-shot adaptation of VLMs in a noisy setting, which better aligns with real-world scenarios where mislabeled samples may exist in the support set. We hope this work inspires future studies to focus on the generalization and robustness of pre-trained large-scale foundation models.