Supplementary Material for Semantic-guided Robustness Tuning for Few-Shot Transfer Across Extreme Domain Shift

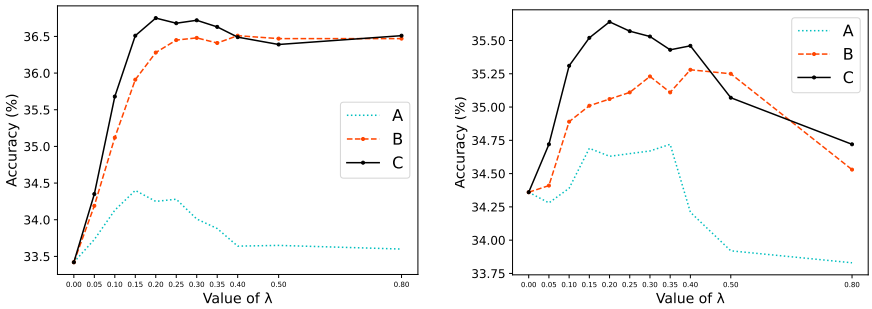
K. Xiao, Z. Wang, J. Li

6 More Ablation Study

In this section, we conduct more detailed analysis for our proposed SRT, including more ablation studies on MMIT-Mixup, RI-FT and some popular fine-tuning techniques (*e.g.*, LoRA), as well as a broader study of more LPMs.

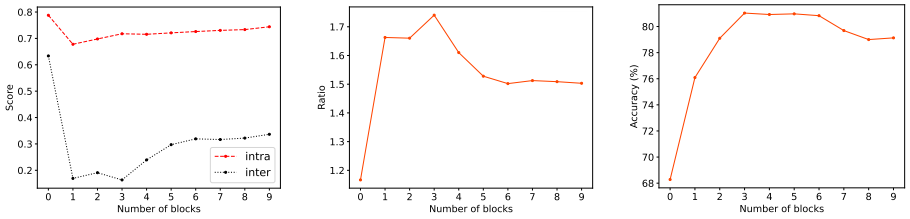
6.1 More analysis of MMIT-Mixup.

Analysis of λ and modulus scaling. To demonstrate the effect of modulus scaling as well as the hyper-parameter λ, we conduct experiments based on CLIP and DINO, as shown in Figure 6. Considering the L_{CLIP} does not participate in the computation graph of fused features, we remove it in order to more clearly demonstrate the role of modulus scaling and λ. We can see that, 1) for x^{m_1}, since it does not have any modulus scaling, it has minimal effect under a wide range of λ, and it even makes a negative gain when λ is large, because the large λ seriously damages the image features distribution. 2) for x^{m_2}, due to the modulus scaling of x_{text}, it has a much smoother selection of λ and successfully enables textual information to help the fine-tuning. 3) for x^{m_3}, since it maintains the input-output modulus stability, it enables a more stable forward computing chain, which can achieve a larger improvement.



(a) CLIP-basd **(b)** DINO-basd

Fig. 6: The analysis of λ and modulus scaling based on CLIP and DINO. Here A, B, C reveal using x^{m_1}, x^{m_2}, x^{m_3} as the new cls-token feature, respectively.



(a) Similarity score **(b)** Similarity ratio **(c)** 1-shot accuracy

Fig. 7: Analysis of blocks number of MMIT-Mixup. The blocks number means the last 'k' blocks, *e.g.* '3' reveals performing MMIT-Mixup before each of the last 3 blocks. The definition of similarity score and ratio is consistent with the w in Table 2.

How to determine the number of mixuped blocks? To demonstrate the impact of MMIT-Mixup blocks selection, we conduct experiments on cosine similarity and 1-shot accuracy, as shown in Figure 7. Obviously, by injecting class-specific textual information for fine-tuning, the inter-class feature similarity is greatly reduced, while the intra-class similarity remains almost unchanged, which successfully induces a robust class-specific representation, resulting in a great performance gain. For early-level features of the backbone, they have low semantic consistency with textual features (the modality gap between low-level feature and textual feature is large), this will make the mixed features having more ambiguity, so mixing will bring serious interference that is harmful for model training. Since RI-FT will increase the distance between domain-invariant prototypes, when there exists interference in prototypes, the distance optimizing will be hindered, leading to a weaker discrimination between different classes. For intra-class, it is equal to a data augmentation, so there is a slight increase in similarity between the same class samples. The similarity between high-level feature and text feature is higher, but there are less blocks that can utilize mixed feature, which limits the learning for bridging domain and modality gap. Therefore, textual injection should be balanced based on similarity score and ratio to obtain the most robust class-specific representation.

Why mixing on cls-token? In our framework, we just perform MMIT-Mixup on mid-level cls-token feature. This raises question: can we perform MMIT-Mixup on patch tokens? To investigate, we conduct experimental verification, as shown in Table 4. Obviously, mixing on patch-tokens indeed performs slightly better, because it can better leverage the coorelation between class text and local image features. But in order to fit each token, it needs a multiple dimension adapter, which greatly increases computing cost that limits its practical application. Therefore, by trade-off, we just mixup on cls-token, and it has already achieved good enough results.

	CLIP-based		DINO-based		adapter-dim
Method	1-shot	5-shot	1-shot	5-shot	
	50.93	62.22	53.67	65.17	-
cls-token	57.12	67.28	55.68	67.26	1x512x768
patch-token	57.36	67.31	55.24	67.65	196x512x768

Table 4: *avg* cls-I acc of mixing on cls-token and patch-tokens based on CLIP and DINO. Here 'adapter-dim' is the dimension of dimension transformation layer (*i.e.* '*linear*').

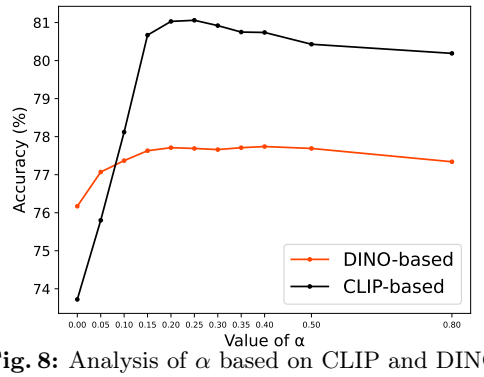


Fig. 8: Analysis of α based on CLIP and DINO.

Why the dimension transformation layers (*i.e.*, '*linear*') are trainable? In our framework, we set the '*linear*' layers before each MMIT-Mixup trainable. In order to compare the fixed layers with trainable layers, we conducted experiments on 1-shot acc and cosine similarity, as shown in Table 5. Considering the L_{CLIP} does not participate in the computation graph of fused features, we remove it in order to more clearly demonstrate the role of '*linear*'. Evidently, 'trainable' performs slightly better than 'fixed', because the optimized '*linear*' layers can map the textual features into a more discriminative space, allowing textual features to contain more accurate class-specific information, which has a similar effect to prompt tuning.

	CLIP-based			DINO-based			sim
Method	EuroSAT	CropDisease	ISIC	EuroSAT	CropDisease	ISIC	
	68.28	81.32	31.80	74.02	83.45	34.04	0.6694
fixed	73.53	84.20	36.27	76.05	84.52	35.03	0.6694
trainable	73.72	84.39	36.77	76.17	84.67	35.64	0.6258

Table 5: Analysis of the dimension transformation layer based on CLIP and DINO. We report the cls-I acc. 'fixed' and 'trainable' reveal the 'linear' is fixed or trainable, respectively. The *sim* means the average cosine similarity of textual features across different classes, *e.g.*, $sim(T_i, T_j), i \neq j$.

6.2 More analysis of RI-FT.

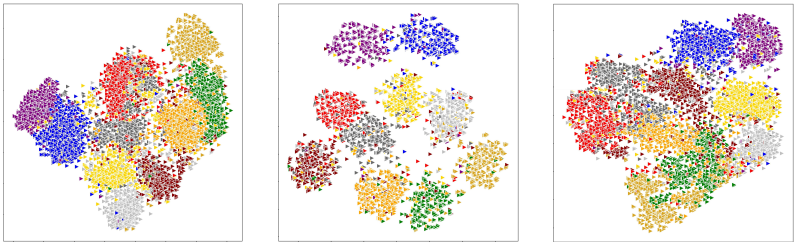
Analysis of α. To reveal the effect of hyperparameter α, we conduct experiments based on CLIP and DINO, as shown in Figure 8. It can be seen that when α is between 0.15 and 0.3, the performance is robust.

			1-shot			*sim*		
w-w	s-s	w-s	CLS	EuroSAT	CropDisease	ISIC	*intra*	*inter*	*w*
			i	68.28	81.32	31.80	0.78	0.63	1.16
			t	60.06	55.68	30.19			
√			i	80.66	84.17	38.64	0.50	0.13	1.44
			t	80.24	83.23	38.08			
	√		i	80.15	83.85	38.20	0.64	0.25	1.47
			t	77.44	78.89	37.21			
		√	i	81.03	85.22	39.27	0.53	0.10	1.54
			t	80.66	83.27	38.45			

Table 6: Augmented image pairs.

Why using weak-strong augmented pairs? Note that L_{RI} uses weak-strong augmented pairs. To analyze the role of weak-strong (w-s) pairs, we compare it with weak-weak (w-w) pairs and strong-strong (s-s) pairs, as shown in Table 6. Obviously, weak-strong pairs performs the best. In principle, the weak augmented image is equivalent to an invariant anchor used to provide an invariant prototype as

class weights. While strong augmented image is a significant transformation, which can be seen as a transformation on domain. The weak-weak pair lacks transformation, so it fails to achieve invariant representation for different transformations. The strong-strong pair lacks a robust anchor, resulting in significant deviations in the learned distribution. Contrary to them, the weak-strong pair requires a semantic consistency between transformations and invariant representation (*i.e.*, domain-invariant prototypes), which can bring excellent robustness.



(a) CLIP **(b)** L_{RI} **(c)** L_{con}

Fig. 9: The t-SNE plots of 10 classes from EuroSAT for comparing RI-FT with contrastive learning, with image features obtained from original CLIP, RI-FT (fine-tuning with L_{RI}) and contrastive learning (fine-tuning with L_{con}).

Comparing RI-FT with contrastive learning. Note that the contrastive learning [1, 7] also uses weak-strong pairs, especially the infoNCE loss [1] (as shown in Eq.15), and the recent CDFSC work ConFT [4] also employs infoNCE. Therefore, it is necessary to compare infoNCE with L_{RI} to explore who is more suitable for CDFSC.

				5-shot			*sim*		
Index	L_{RI}	L_{con}	CLS	EuroSAT	CropDisease	ISIC	*intra*	*inter*	*w*
A			i	84.45	94.31	45.24	0.78	0.63	1.16
			t	60.06	55.68	30.19			
B	√		i	91.07	95.33	53.16	0.89	0.08	2.25
			t	85.49	90.45	50.08			
C		√	i	87.84	92.37	48.56	0.34	0.18	1.17
			t	58.45	59.71	37.41			

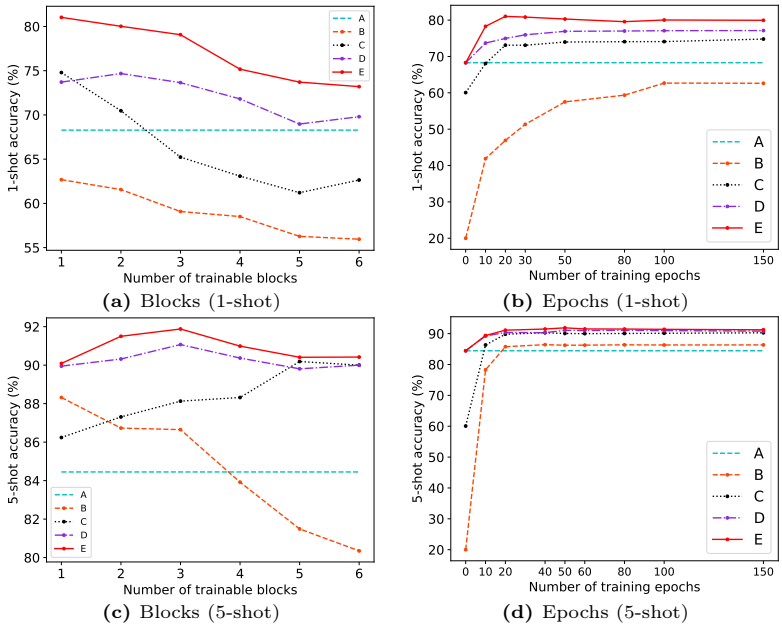
Table 7: Comparing RI-FT with contrastive learning on 5-shot setting. Here L_{con} means using a infoNCE.

$$L_{infoNCE} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp\left(\mathrm{sim}\left(q_i^w \cdot q_i^s\right)\right)}{\sum_z \exp\left(\mathrm{sim}\left(q_i^w \cdot q_z^s\right)\right)} \tag{15}$$

For 1-shot, infoNCE and L_{RI} have a similar form (only exchanges the positions of weak and strong samples), while for 5-shot, they are significantly different. We conduct experiments and t-SNE analysis on 5-shot setting, as shown in Table 7 and Figure 9. Both L_{RI} and infoNCE can bring improvement, but L_{RI} is clearly better. It can also be observed that, L_{RI} largely increases the distance between different classes (A *vs.* B), thus bringing better discrimination. InfoNCE increases the distance between different classes, but it also decreases the similarity of intra-class samples, which leads to poor discrimination of features.

In principle, though both of them use weak-strong pairs, but their purposes are different. InfoNCE regards the weak-strong pair of a sample as a positive

pair, while the other samples are all negative samples. It emphasizes the invariance of transformations, but it also regards the other intra-class samples as negative samples, so it will result in a significant decreases in the similarity of intra-class samples. Differently, L_{RI} regards all weak/strong intra-class samples and prototype of the same class as positive samples, while inter-class prototypes are considered as negative samples. Therefore, optimizing the distance between positive samples can bring invariance to transformations, while optimizing the distance between negative samples can increase the discrimination of inter-class features. In summary, for CDFSC, L_{RI} is more helpful in inducing more discriminative class-specific representations.



(a) Blocks (1-shot) **(b)** Epochs (1-shot)

(c) Blocks (5-shot) **(d)** Epochs (5-shot)

Fig. 10: Comparing SRT with different fine-tuning paradigms under 1-shot and 5-shot setting. Here A, B, C, D, E respectively reveal {A: the original CLIP, B: fine-tuning with Trad-FT, C: fine-tuning with L_{CLIP}, D: fine-tuning with L_{RI}, E: the combination of L_{RI} and L_{CLIP} (*i.e.*, L_{SRT}) }. The 'blocks' means the number of the last 'k' blocks, *e.g.*, '3' reveals the last 3 blocks are trainable.

Compared with different fine-tuning paradigms. In order to explore the performance of different fine-tuning paradigms in CDFSC fine-tuning, we compare them in Figure 10, with a focus on overfitting problem, convergence speed, and classification accuracy. Among them, **B** needs to randomly initialize a linear classifier, while **C** uses textual features as class weights, which is equivalent to a linear classifier given initialized weights. And **D** does not need any given

class weights. Particularly, Trad-FT (line **B**) introduces randomly initialized parameters that need to be learned from scratch, so the fine-tuning is slower to converge (about 100 epochs for 1-shot), and is easier to overfit (more severe performance damage when given more trainable blocks). L_{CLIP} (line **C**) is given initialized classifier weights, so it is faster to converge, but its simple alignment with class weights can easily lead to overfitting on support images, especially when the low-data problem is severe (*i.e.*, 1-shot). L_{RI} (line **D**) does not introduce any new parameters, and it adaptively optimizes the distance between features and prototypes to achieve better discrimination. Obviously, using L_{RI} for fine-tuning has an extremely fast convergence speed and is more resistant to overfitting problems when there are more trainable blocks. In addition, the combination of L_{RI} and L_{CLIP} integrates two different optimization perspectives, which exhibits extremely fast convergence speed and high accuracy.

			1-shot					5-shot				
Method	Backbone	FT	EuroSAT	CropDisease	ISIC	*avg.*	*imp.*-I	EuroSAT	CropDisease	ISIC	*avg.*	*imp.*-I
StyleAdv	res10	F	70.94	74.13	33.96	59.68	-	86.58	93.65	45.77	75.33	-
StyleAdv	ViT-S	F	72.15	81.22	33.05	62.14	-	88.57	94.85	47.73	77.05	-
CLIP	ViT-B	F	68.28	81.32	31.80	60.47	-	84.45	94.31	45.24	74.67	-
DINO	ViT-B	F	74.02	83.45	34.04	63.84	-	90.55	95.23	48.42	78.07	-
DINOv2	ViT-B	F	63.65	89.04	31.20	61.30	-	81.72	95.82	41.89	73.14	-
StyleAdv-FT	res10	T	72.92	80.69	35.76	63.12	3.44	91.64	96.51	53.05	80.40	5.07
StyleAdv-FT	ViT-S	T	74.93	84.11	33.99	64.34	2.20	90.12	95.99	51.23	79.11	2.06
CLIP+SRT	ViT-B	T	81.03	85.22	39.27	68.35	7.88	91.88	95.99	54.21	80.69	6.02
DINO+SRT	ViT-B	T	77.95	85.64	35.51	66.37	2.53	92.62	96.30	52.96	80.63	2.56
DINOv2+SRT	ViT-B	T	73.51	90.73	33.38	65.87	4.57	92.11	97.36	52.49	80.65	7.51
CLIP+SRT+FT	ViT-B	T	80.77	84.35	38.64	67.92	7.45	91.32	95.20	53.51	80.01	5.34
DINO+SRT+FT	ViT-B	T	75.68	83.73	34.20	64.54	0.70	92.09	95.82	52.44	80.12	2.05
DINOv2+SRT+FT	ViT-B	T	73.33	89.60	32.71	65.21	3.91	92.31	96.62	51.78	80.24	7.10

Table 8: The combination of RI-FT and Trad-FT. We report the accuracy of cls-I.

Is RI-FT and Trad-FT compatible? Since SRT has obtained a robust visual-encoder, we can consider whether RI-FT and Trad-FT can be combined, for example, we can first fine-tune with SRT, then fix the visual-encoder and use Trad-FT to train a linear classifier. To verify this idea, we conduct experiments, as shown in Table 8. It can be observed that their combination sometimes perform better, but most perform worse. This is because, when the representations are already robust enough, a classifier based on the mean-centroid (*i.e.*, cls-I) can be a more accurate class weights than a new-trained linear classifier. Therefore, we believe that using SRT is already sufficient.

Compared with popular LPM fine-tuning techniques. Note that the currently most popular LPM fine-tuning technique is LoRA [6], which adds few trainable parameters and keeps the parameters of the original network fixed. While SRT allows the last few blocks to be trainable, for example, the last 1 block is trainable for 1-shot. In order to compare whether using trainable blocks or using LoRA is better for CDFSC, we conducted experimental verification.

Moreover, for the discussion with attention *vs*. MMIT-Mixup, we additionally compared the recent attention-based FSC method SP [2], as shown in Table 9.

Obviously, since LoRA introduces many learnable parameters initialized from 0, learning such new parameters is difficult and will bring a significant risk of overfitting, especially in extreme low-data situation (*e.g.*, 1-shot), resulting in its poor performance. For SP, it pre-trains cross-attention layers on the source domain for semantic injection. But for novel domains, since the extreme domain shift prevents robust image features, it is very difficult to calculate a correct attention map, which leads to serious performance damage. This is consistent with the analysis in section 4.3

Method	EuroSAT	CropDisease	ISIC
	68.28	81.32	31.80
LoRA [6]	75.81	82.23	32.97
SP [2]	66.14	80.07	32.56
SRT	81.03	85.22	39.27

Table 9: comparison with popular LPM fine-tuning techniques. We report the 5-way-1-shot cls-I acc.

6.3 More details during fine-tuning.

Comparison of different prompts. To demonstrate the impact of fixed prefixes and suffixes, we compare different prompts, as shown in Table 10. We can observe that, before fine-tuning, cls-T is sensitive to different prompts, for example, CLIP's cls-T of 'a photo of a *' (57.32%) *vs*. 'a photo of * by satellite' (63.49%). While after fine-tuning with SRT, cls-T is robust for different prompts, because the fine-tuned visual-encoder can obtain robust class-specific representations, which have high discrimination for different types of class weights (textual features), and the optimization of 'linear-proj' also helps the alignment.

		CLIP-based		DINO-based		DINOv2-based	
Prompt	CLS	Origin	+SRT	Origin	+SRT	Origin	+SRT
a photo of *	i	68.28	81.03	74.02	77.95	63.65	73.51
	t	60.06	80.66	31.77	76.22	17.02	73.52
a *	i	68.28	80.71	74.02	77.44	63.65	73.71
	t	58.20	80.95	27.53	75.00	16.69	73.40
a photo of a *	i	68.28	80.44	74.02	77.49	63.65	73.04
	t	57.32	80.08	24.18	75.38	17.47	73.44
a photo of * by satellite	i	68.28	80.57	74.02	78.02	63.65	73.33
	t	63.49	79.91	25.98	76.16	16.98	73.20

Table 10: Comparison of prompts on EuroSAT. * reveals the *class name*.

Details of EuroSAT 10-way-k-shot setting. More details of EuroSAT 10-way-k-shot setting is shown in Table 11. Obviously, when given more training data (*i.e.*, more shots), the problem of overfitting is less serious, so it is allowed to have more trainable blocks, and it also needs more epochs for fully capturing task-specific knowledge. In addition, comparing 10-way with the 5-way setting,

Shot	1	2	4	8	16
Train-num	1	2	3	4	5
Mixup-num	7	7	7	7	7
Training-epochs	15	20	30	50	80
Accuracy	76.82	83.67	86.89	90.22	93.75

Table 11: Details of EuroSAT 10-way-k-shot. 'Train-num' reveals the last blocks number of trainable blocks, 'Mixup-num' means performing MMIT-Mixup before each of the last n blocks.

since it has more classes, there is a need for earlier MMIT-Mixup (*i.e.*, injecting textual information into more blocks) to emphasize learning class-specific knowledge.

6.4 More LPMs.

			1-shot					5-shot				
Method	Backbone	FT	EuroSAT	CropDisease	ISIC	*avg.*	*imp.*-I	EuroSAT	CropDisease	ISIC	*avg.*	*imp.*-I
StyleAdv	Res10	F	70.94	74.13	33.96	59.68	-	86.58	93.65	45.77	75.33	-
	ViT-S	F	72.15	81.22	33.05	62.14	-	88.57	94.85	47.73	77.05	-
CLIP	ViT-B	F	68.28	81.32	31.80	60.47	-	84.45	94.31	45.24	74.67	-
	ViT-L	F	66.77	83.39	34.00	61.38	-	84.00	93.57	47.76	75.11	-
DINO	ViT-S	F	75.51	78.27	34.76	62.85	-	90.84	93.20	46.71	76.92	-
	ViT-B	F	74.02	83.45	34.04	63.84	-	90.55	95.23	48.42	78.07	-
DINOv2	ViT-S	F	63.73	86.22	32.49	60.81	-	80.09	95.16	44.36	73.20	-
	ViT-B	F	63.65	89.04	31.20	61.30	-	81.72	95.82	41.89	73.14	-
	ViT-L	F	70.16	90.40	30.71	63.76	-	86.44	95.60	43.42	75.15	-
MAE	ViT-B	F	68.18	81.68	31.26	60.37	-	82.46	92.92	45.72	73.70	-
MoCo-v3	ViT-B	F	69.22	79.40	33.44	60.69	-	88.07	93.13	47.17	76.12	-
StyleAdv-FT	Res10	T	72.92	80.69	35.76	63.12	3.44	91.64	96.51	53.05	80.40	5.07
	ViT-S	T	74.93	84.11	33.99	64.34	2.20	90.12	95.99	51.23	79.11	2.06
CLIP + SRT	ViT-B	T	81.03	85.22	39.27	68.35	7.88	91.88	95.99	54.21	80.69	6.02
	ViT-L	T	78.77	87.28	38.00	68.02	6.64	91.24	95.20	53.37	79.94	4.83
DINO + SRT	ViT-S	T	79.56	82.98	35.33	65.96	3.11	93.29	95.94	56.08	81.77	4.85
	ViT-B	T	77.95	85.64	35.51	66.37	2.53	92.62	96.30	52.96	80.63	2.56
DINOv2 + SRT	ViT-S	T	72.76	86.98	33.82	64.52	3.71	91.42	96.40	52.00	79.94	6.74
	ViT-B	T	73.51	90.73	33.38	65.87	4.57	92.11	97.36	52.49	80.65	7.51
	ViT-L	T	77.64	91.64	32.62	67.30	3.54	92.53	97.11	51.73	80.46	5.31
MAE + SRT	ViT-B	T	75.70	85.32	34.58	65.20	4.83	90.54	95.39	52.33	79.42	5.72
MoCo-v3 + SRT	ViT-B	T	75.08	84.59	35.68	65.12	4.43	91.07	95.24	52.80	79.70	3.58

Table 12: A broader study of more LPMs. We report the accuracy of cls-I. Obviously, SRT can bring stable and significant gains for various LPMs.

We conduct experiments on more types of LPMs, including CLIP, DINO, DINOv2, MAE [5] and MoCo-v3 [3], whose scales are ranged from ViT-S to ViT-L, as shown in Table 12. Obviously, these models do not show competitive performance before fine-tuning. While with the boost of SRT, they can achieve significant improvement with the help of textual guidance as well as utilizing their rich prior knowledge. Especially, for ViT-S, SRT brings greatly higher gains than Trad-FT (*e.g.*, the 5-shot *imp.*-I, DINO+SRT 4.85% *vs.* StyleAdv-FT 2.06%).

It indeed indicates that SRT is a general solution for LPM's CDFSC challenge and outperforms the existing methods with a large margin.

References

1. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International conference on machine learning. pp. 1597–1607. PMLR (2020)
2. Chen, W., Si, C., Zhang, Z., Wang, L., Wang, Z., Tan, T.: Semantic prompt for few-shot image recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 23581–23591 (2023)
3. Chen, X., Xie, S., He, K.: An empirical study of training self-supervised vision transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9640–9649 (2021)
4. Das, R., Wang, Y.X., Moura, J.M.: On the importance of distractors for few-shot classification. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9030–9040 (2021)
5. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 16000–16009 (2022)
6. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685 (2021)
7. Li, J., Zhang, Y., Wang, Z., Tu, K.: Semantic-aware representation learning via probability contrastive loss. arXiv preprint arXiv:2111.06021 (2021)