

# CoCoA-Mix: Confusion-and-Confidence-Aware Mixture Model for Context Optimization

Dasol Hong<sup>1</sup>   Wooju Lee<sup>1</sup>   Hyun Myung<sup>1</sup>

<sup>1</sup>Korea Advanced Institute of Science and Technology



GitHub



Project Page



ArXiv

## ● Vision-Language Models (VLMs)

- Pre-trained VLMs demonstrate remarkable performance in open-set scenarios, where model handles novel categories without predefined labels.
- Their versatility makes them highly useful across a wide range of robotics applications, especially in unpredictable environments.

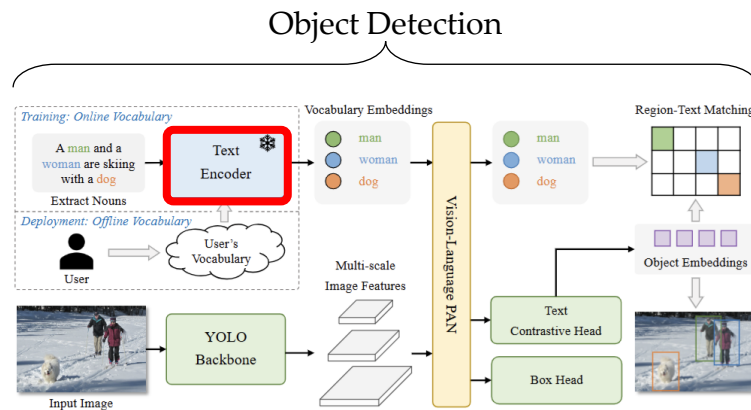


Fig 1. Architecture of YOLO-World<sup>[1]</sup>

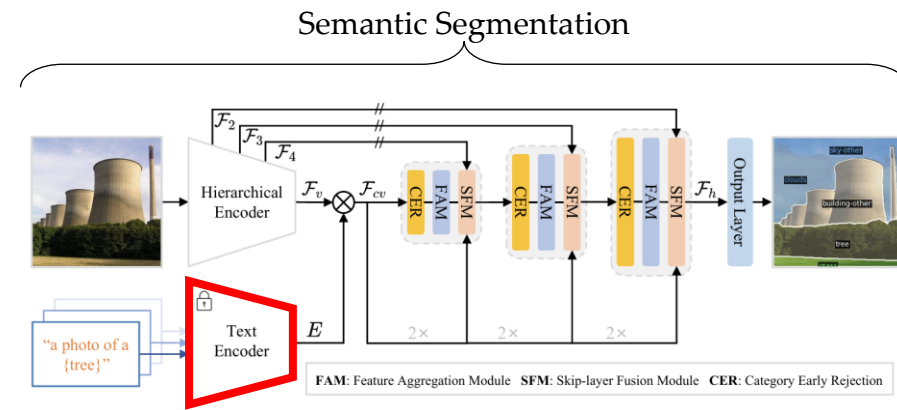


Fig 2. Architecture of SED<sup>[2]</sup>

## ● Need for Prompt Tuning

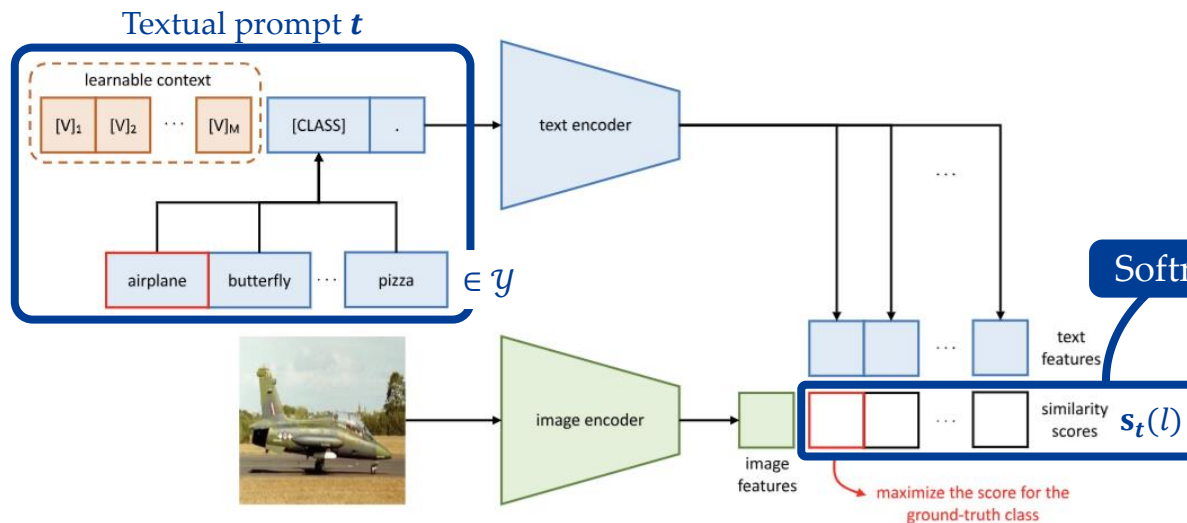
- Despite their strong zero-shot capabilities, pre-trained VLMs often require adaptation to perform well on specific downstream tasks due to their generic embeddings.
- Prompt tuning enables this adaptation by optimizing textual prompts while keeping the model frozen, making it an efficient and scalable solution.
- This approach enhances task performance without the need for full model fine-tuning.



Fig 3. Performance of YOLO-World<sup>[1]</sup> combined with MobileSAM<sup>[3]</sup> under different prompts.

## ● Preliminary: Prompt Tuning of CLIP model

- CLIP<sup>[4]</sup> maps images and texts into a shared embedding space using separate visual and textual encoders.
- In image classification, textual prompts like “a photo of a [CLASS]” are embedded and compared to the visual feature for prediction.



Probability of the image belonging to the class  $l$ :

$$\hat{p}_t(l) = \frac{\exp(s_t(l)/\tau)}{\sum_{l' \in \mathcal{Y}} \exp(s_t(l')/\tau)},$$

where  $t$  is the textual prompt;  $\tau$  is the temperature scale;  $\mathcal{Y}$  represents the set of classes, and  $s_t(l)$  denotes the cosine similarity between the visual and textual embeddings of the class  $l$  generated by  $t$ .

Fig 4. Architecture of CLIP<sup>[4]</sup>

## ● Decomposing Specialization and Generalization

- The expected error  $\epsilon_T(\hat{p})$  of a predictive distribution  $\hat{p}$  in an arbitrary target domain  $\mathcal{D}_T$ :

$$\epsilon_T(\hat{p}) = \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}_T} [-\log \hat{p}(y)],$$

where  $y$  is the ground-truth label for the image  $\mathbf{x}$ .

- **Definition 3.1 (Mixture Model)** Let  $K + 1$  different prompts be given by  $\mathcal{T} = \{\mathbf{t}_0, \mathbf{t}_1, \dots, \mathbf{t}_K\}$ , and let  $\boldsymbol{\pi} = \{\pi_0, \pi_1, \dots, \pi_K\}$  denote a set of non-negative weights satisfying  $\sum_{i=0}^K \pi_i = 1$ . The mixture model  $\hat{p}_{\mathcal{T}}^{\boldsymbol{\pi}}$  is defined as a weighted combination of the individual prompts:

$$\hat{p}_{\mathcal{T}}^{\boldsymbol{\pi}}(l) = \frac{\exp\left(\sum_{i=0}^K \pi_i \mathbf{s}_{\mathbf{t}_i}(l)/\tau\right)}{\sum_{l' \in \mathcal{Y}} \exp\left(\sum_{i=0}^K \pi_i \mathbf{s}_{\mathbf{t}_i}(l')/\tau\right)}.$$

Ref: Probability of the model with prompt  $\mathbf{t}$

$$\hat{p}_{\mathbf{t}}(l) = \frac{\exp(\mathbf{s}_{\mathbf{t}}(l)/\tau)}{\sum_{l' \in \mathcal{Y}} \exp(\mathbf{s}_{\mathbf{t}}(l')/\tau)},$$

- **Theorem 3.2.** The expected error of the mixture model  $\hat{p}_{\mathcal{T}}^{\boldsymbol{\pi}}$  can be bounded as follows:

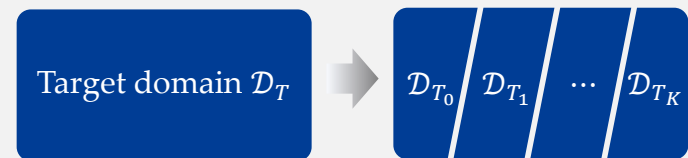
\* Please refer to [Appendix A](#) for details.

$$\epsilon_T(\hat{p}_{\mathcal{T}}^{\boldsymbol{\pi}}) \leq \sum_{i=0}^K \pi_i \epsilon_T(\hat{p}_{\mathbf{t}_i}).$$

## ● Decomposing Specialization and Generalization

- **Lemma 3.3.** The expected error of the mixture model  $\hat{p}_{\mathcal{T}}^{\pi}$  is given by:

$$\epsilon_T(\hat{p}_{\mathcal{T}}^{\pi}) = \sum_{i=0}^K \lambda_i \epsilon_{T_i}(\hat{p}_{\mathcal{T}}^{\pi}),$$



where  $\lambda_i = \Pr_{(\mathbf{x}, y) \sim \mathcal{D}_T}[(\mathbf{x}, y) \in \mathcal{D}_{T_i}]$ : the probability that a sample from  $\mathcal{D}_T$  belongs to the sub-domain  $\mathcal{D}_{T_i}$ .

- Based on Theorem 3.2, the error of the mixture model can be upper-bounded as follows:

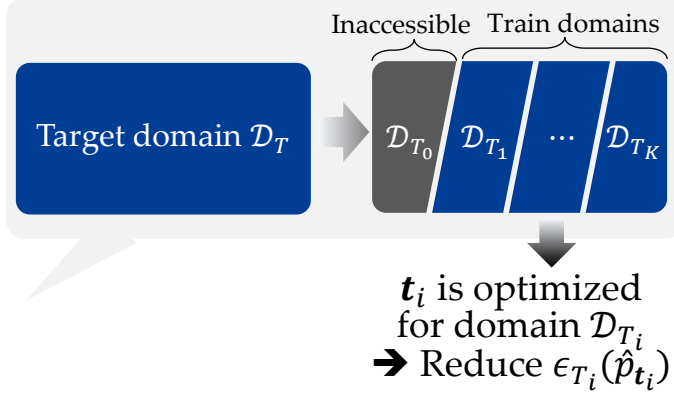
$$\epsilon_T(\hat{p}_{\mathcal{T}}^{\pi}) \leq \sum_{i=0}^K \lambda_i \left( \underbrace{\pi_i^{\text{in}} \epsilon_{T_i}(\hat{p}_{\mathbf{t}_i})}_{\text{specialization error}} + \underbrace{\sum_{\substack{j=0 \\ j \neq i}}^K \pi_j^{\text{out}} \epsilon_{T_i}(\hat{p}_{\mathbf{t}_j})}_{\text{generalization error}} \right),$$

Ref: Theorem 3.2

$$\epsilon_T(\hat{p}_{\mathcal{T}}^{\pi}) \leq \sum_{i=0}^K \pi_i \epsilon_{T_i}(\hat{p}_{\mathbf{t}_i}).$$

where  $\pi_i^{\text{in}}$ : the mixing weights of the prompt  $\mathbf{t}_i$  for its own domain  $\mathcal{D}_{T_i}$ ;  $\pi_j^{\text{out}}$ : the mixing weight of the prompt  $\mathbf{t}_j$  ( $j \neq i$ ) when applied to the domain  $\mathcal{D}_{T_i}$ .

## • Decomposing Specialization and Generalization

$$\epsilon_T(\hat{p}^{\pi}) \leq \sum_{i=0}^K \lambda_i \left( \underbrace{\pi_i^{in} \epsilon_{T_i}(\hat{p}_{t_i})}_{\text{specialization error}} + \underbrace{\sum_{\substack{j=0 \\ j \neq i}}^K \pi_j^{out} \epsilon_{T_i}(\hat{p}_{t_j})}_{\text{generalization error}} \right),$$


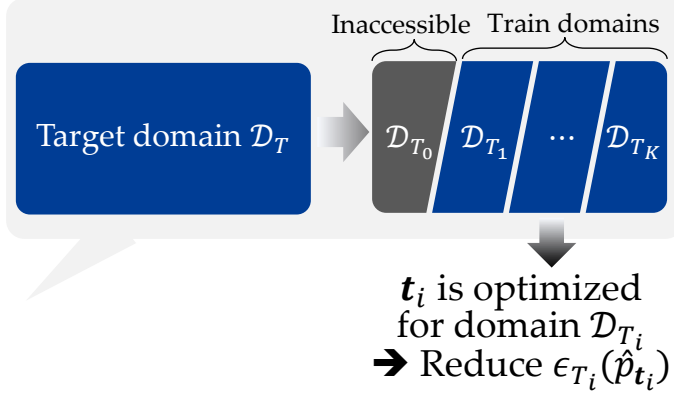
$t_i$  is optimized for domain  $\mathcal{D}_{T_i}$   
→ Reduce  $\epsilon_{T_i}(\hat{p}_{t_i})$

- Goal: Improve both specialization and generalization in prompt tuning
- CoA-loss handles class confusion and CoA-weights adapts confidence across domains
  - Domain  $\mathcal{D}_{T_0}$  : Unseen target domain, no labeled data available (hand-crafted prompt  $t_0$ )
  - Domain  $\mathcal{D}_{T_i}$  : Seen training domains with labeled data ( $i > 0$ )
    - \* Prompt  $t_i$  is optimized for each training domain  $\mathcal{D}_{T_i}$ .

→ “a photo of a [CLASS].”



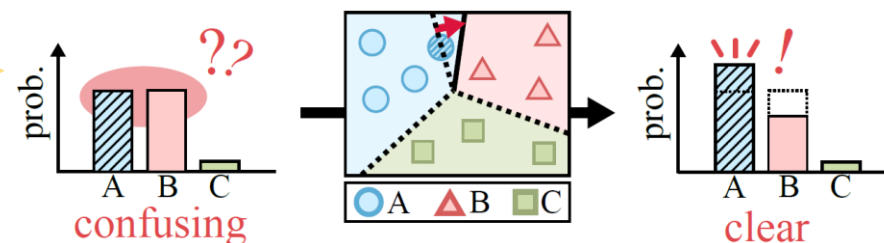
## ● Confusion-Aware Loss for Specialization

$$\epsilon_T(\hat{p}_T^\pi) \leq \sum_{i=0}^K \lambda_i \left( \underbrace{\pi_i^{in} \epsilon_{T_i}(\hat{p}_{t_i})}_{\text{specialization error}} + \underbrace{\sum_{\substack{j=0 \\ j \neq i}}^K \pi_j^{out} \epsilon_{T_i}(\hat{p}_{t_j})}_{\text{generalization error}} \right),$$


- Most existing methods use standard cross-entropy for the specialization in prompt tuning:

$$\mathcal{L}_{CE}(\mathbf{x}, y; \hat{p}_t) = -\log \hat{p}_t(y).$$

- The loss do not explicitly address confusing cases arising from the frozen visual encoder.
- Therefore, it limits the specialization of prompt tuning.





## ● Confusion-Aware Loss for Specialization

- We propose confusion-aware loss (CoA-Loss), defined as follows:

$$\mathcal{L}_{\text{CoA}}(\mathbf{x}, y; \hat{p}_t) = 1 - \hat{p}_t(y).$$

- The overall loss is  $\mathcal{L}_{\text{prompt}}(\mathbf{x}, y; \hat{p}_t) = \mathcal{L}_{\text{CE}} + w\mathcal{L}_{\text{CoA}}$ , where  $w$  is a hyperparameter.
- The gradients of  $\mathcal{L}_{\text{prompt}}$  with respect to the similarities  $\mathbf{s}_t(y)$  and  $\mathbf{s}_t(c \neq y)$  are as follows:
  - CoA-loss induces larger gradient updates for the confusing classes.

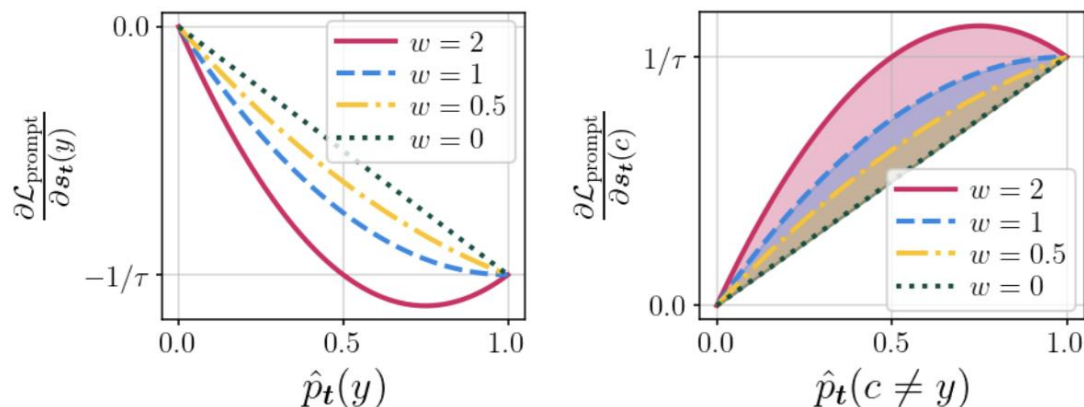


Fig 5. Gradient component of  $\mathcal{L}_{\text{prompt}}$  with respect to (a)  $\mathbf{s}_t(y)$  and (b)  $\mathbf{s}_t(c \neq y)$ , where  $w = 0$  represents standard cross-entropy.

$$\frac{\partial \mathcal{L}_{\text{prompt}}}{\partial \mathbf{s}_t(y)} = -\frac{1}{\tau} (1 - \hat{p}_t(y)) (1 - w\hat{p}_t(y)),$$

$$\frac{\partial \mathcal{L}_{\text{prompt}}}{\partial \mathbf{s}_t(c \neq y)} = \frac{1}{\tau} \hat{p}_t(c) (1 + w\hat{p}_t(y)).$$

- As  $w \rightarrow \infty$ , the largest gradient update occurs when  $\hat{p}_t(y) = 0.5$
- As  $w \rightarrow \infty$ , the largest gradient update occurs when  $\hat{p}_t(c) = \hat{p}_t(y)$

## ● Confidence-Aware Weights for Generalization without Trade-Offs

$$\epsilon_T(\hat{p}_T^\pi) \leq \sum_{i=0}^K \lambda_i \left( \underbrace{\pi_i^{\text{in}} \epsilon_{T_i}(\hat{p}_{t_i})}_{\text{specialization error}} + \underbrace{\sum_{\substack{j=0 \\ j \neq i}}^K \pi_j^{\text{out}} \epsilon_{T_i}(\hat{p}_{t_j})}_{\text{generalization error}} \right),$$

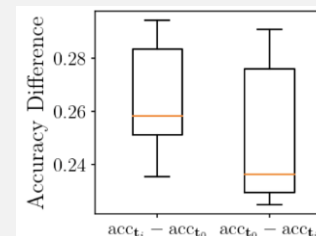
- **Assumption 3.4.** The specialized prediction  $\hat{p}_{t_i}$  for  $\mathcal{D}_{T_i}$  satisfies the following relationships:

$$\epsilon_{T_{t_i}}(\hat{p}_{t_i}) \leq \epsilon_{T_{t_i}}(\hat{p}_{t_{j \neq i}}) \quad \text{and} \quad \epsilon_{T_{t_{j \neq i}}}(\hat{p}_{t_0}) \leq \epsilon_{T_{t_{j \neq i}}}(\hat{p}_{t_i}).$$

1. A prediction  $\hat{p}_{t_i}$  optimized for a specific domain  $\mathcal{D}_{T_i}$  always performs better than predictions  $\hat{p}_{t_{j \neq i}}$  made by prompts optimized for other domains
2. The generalized prediction  $\hat{p}_{t_0}$  is more effective for unseen classes.

**Statistic experiment using the CIFAR-100 and the pre-trained CLIP model:**

- 100 classes are randomly splited into 50 in-class and 50 out-class domains.
- The prediction with prompt trained on the in-class subset is compared with the zero-shot on both domains.
- $p$ -value were  $p_{\text{in}} = 9.25 \times 10^{-12}$  and  $p_{\text{out}} = 2.06 \times 10^{-10}$   
 ➔ Inequalities in Assumption 3.4 hold with strong statistical significance.



## ● Confidence-Aware Weights for Generalization without Trade-Offs

- **Assumption 3.4.** The specialized prediction  $\hat{p}_{t_i}$  for  $\mathcal{D}_{T_i}$  satisfies the following relationships:

$$\epsilon_{T_{t_i}}(\hat{p}_{t_i}) \leq \epsilon_{T_{t_i}}(\hat{p}_{t_{j \neq i}}) \quad \text{and} \quad \epsilon_{T_{t_{j \neq i}}}(\hat{p}_{t_0}) \leq \epsilon_{T_{t_{j \neq i}}}(\hat{p}_{t_i}).$$

- Optimizing  $\pi_i^{\text{in}}$  for in-class domains

$$\pi_i^{\text{in}} = \arg \min_{\pi_i^{\text{in}}} \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}_{S_i}} [\mathcal{L}_{\text{CE}}(\mathbf{x}, y; \hat{p}_{\mathcal{T}}^{\pi})].$$

- If the specialized prediction  $\hat{p}_{t_i}$  outperforms the generalized prediction  $\hat{p}_{t_{j \neq i}}$   $\rightarrow \pi_i^{\text{in}}$  increases  
Otherwise  $\rightarrow \pi_i^{\text{in}}$  decreases
- Further details on the cross-entropy effect in the mixture model are provided in [Appendix B](#).

## ● Confidence-Aware Weights for Generalization without Trade-Offs

- **Assumption 3.4.** The specialized prediction  $\hat{p}_{t_i}$  for  $\mathcal{D}_{T_i}$  satisfies the following relationships:

$$\epsilon_{T_{t_i}}(\hat{p}_{t_i}) \leq \epsilon_{T_{t_i}}(\hat{p}_{t_{j \neq i}}) \quad \text{and} \quad \epsilon_{T_{t_{j \neq i}}}(\hat{p}_{t_0}) \leq \epsilon_{T_{t_{j \neq i}}}(\hat{p}_{t_i}).$$

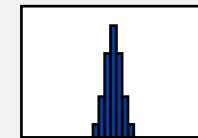
- Optimizing  $\pi_i^{\text{out}}$  for out-class domains

$$\pi_i^{\text{out}} = \arg \min_{\pi_i^{\text{out}}} \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}_{S_i}} [\mathcal{L}_{\text{Ent}}(\mathbf{x}; \hat{p}_{t_i}, \hat{p}_{t_0})],$$

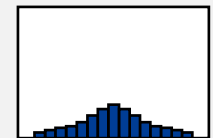
$$\mathcal{L}_{\text{Ent}} = \max(0, H(\hat{p}_{t_0}) - H(\hat{p}_{t_i}) + d),$$

where  $d$  is a margin and  $H(\hat{p})$  is the normalized entropy of  $\hat{p}$  over the out-class set, i.e.  $H(\hat{p}) = \sum_{c \sim \mathcal{Y}_i^{\text{out}}} -\hat{p}(c) \log \hat{p}(c) / \log |\mathcal{Y}_i^{\text{out}}|$ .

- It makes specialized predictions less confident than generalized ones.



High entropy  
→ Uncertain prediction



Low entropy  
→ Confident prediction

## ● Experiments: Base-to-New Generalization

☑ Is CoCoA-Mix effective at balancing specialization on base classes and generalization to new classes?

■ Each dataset is evenly split into two disjoint subsets: *Base* for tuning and unseen *New*

Tab 1. Performance comparison on 11 datasets in the base-to-new benchmark. H represents the harmonic mean.

METHOD	AVERAGE			IMAGENET			CALTECH101		
	BASE	NEW	H	BASE	NEW	H	BASE	NEW	H
CLIP	65.14	68.78	66.82	64.43	60.04	62.16	90.64	91.16	90.90
CoOp	77.23	68.56	71.33	73.72 ± 0.29	64.94 ± 0.87	69.05	97.16 ± 0.16	93.92 ± 0.80	95.51
ProGrAD	78.74	72.19	75.06	74.81 ± 0.29	66.68 ± 0.26	70.51	97.50 ± 0.08	95.49 ± 0.27	96.48
KgCoOp	78.67	74.62	76.38	75.44 ± 0.08	69.43 ± 0.29	72.31	97.61 ± 0.33	94.80 ± 0.45	96.18
MAPLE	77.14	72.91	74.69	75.40 ± 0.29	<b>70.43 ± 0.12</b>	<b>72.83</b>	97.47 ± 0.31	93.77 ± 1.11	95.57
DePT	79.20	66.36	71.78	73.50 ± 0.22	70.00 ± 0.16	71.71	97.83 ± 0.05	<b>95.83 ± 0.25</b>	<b>96.82</b>
CoA-LOSS	79.12	73.66	76.15	<b>75.68 ± 0.00</b>	67.98 ± 0.31	71.62	97.94 ± 0.14	94.54 ± 0.24	96.21
CoCoA-Mix	<b>79.31</b>	<b>75.10</b>	<b>77.03</b>	75.47 ± 0.09	68.92 ± 0.10	72.04	<b>98.02 ± 0.03</b>	94.39 ± 0.10	96.17

METHOD	OXFORDPETS			STANFORDCARS			FLOWERS102		
	BASE	NEW	H	BASE	NEW	H	BASE	NEW	H
CLIP	90.01	94.24	92.07	55.37	66.65	60.49	69.23	73.90	71.49
CoOp	94.10 ± 0.73	94.42 ± 4.17	94.16	69.54 ± 0.75	71.39 ± 1.28	70.44	90.60 ± 1.50	67.00 ± 1.04	77.01
ProGrAD	95.00 ± 0.31	97.36 ± 0.42	96.16	71.45 ± 0.39	73.16 ± 0.58	72.29	91.36 ± 0.63	74.92 ± 0.90	82.32
KgCoOp	94.65 ± 0.15	97.59 ± 0.08	96.10	68.64 ± 0.35	74.96 ± 0.53	71.66	90.09 ± 0.63	76.31 ± 0.42	82.63
MAPLE	94.80 ± 0.94	97.67 ± 0.21	96.21	67.97 ± 0.29	74.40 ± 0.45	71.04	88.03 ± 1.62	73.43 ± 0.49	80.06
DePT	94.00 ± 0.29	88.63 ± 0.78	91.23	71.83 ± 0.52	59.27 ± 0.76	64.94	<b>94.53 ± 0.53</b>	66.30 ± 1.42	77.92
CoA-LOSS	94.90 ± 0.49	<b>97.93 ± 0.08</b>	<b>96.39</b>	72.70 ± 0.11	73.07 ± 1.27	72.87	88.89 ± 1.75	75.58 ± 1.31	81.67
CoCoA-Mix	<b>95.16 ± 0.38</b>	97.60 ± 0.09	96.36	<b>73.09 ± 0.25</b>	<b>74.97 ± 0.08</b>	<b>74.01</b>	91.04 ± 1.79	<b>77.37 ± 0.38</b>	<b>83.64</b>

METHOD	FOOD101			FGVCAIRCRAFT			SUN397		
	BASE	NEW	H	BASE	NEW	H	BASE	NEW	H
CLIP	83.58	84.95	84.26	19.51	24.60	21.76	66.76	70.52	68.59
CoOp	89.19 ± 0.19	88.45 ± 0.89	88.81	26.17 ± 7.89	19.50 ± 11.94	11.46	77.37 ± 0.66	72.06 ± 1.56	74.60
ProGrAD	89.33 ± 0.08	89.93 ± 0.58	89.63	34.21 ± 1.99	28.53 ± 2.08	30.97	<b>79.16 ± 0.36</b>	74.34 ± 0.75	76.20
KgCoOp	<b>90.26 ± 0.11</b>	<b>91.25 ± 0.15</b>	<b>90.75</b>	33.43 ± 0.56	32.27 ± 1.19	32.81	79.07 ± 0.24	76.78 ± 0.24	77.91
MAPLE	89.37 ± 0.54	90.77 ± 0.54	90.06	31.67 ± 0.66	33.13 ± 2.38	32.29	78.33 ± 0.21	<b>77.67 ± 0.45</b>	<b>78.00</b>
DePT	89.80 ± 0.08	88.10 ± 0.16	88.94	<b>35.93 ± 0.93</b>	24.33 ± 0.09	29.01	79.10 ± 0.22	67.27 ± 0.46	72.70
CoA-LOSS	90.11 ± 0.18	90.87 ± 0.42	90.49	33.91 ± 0.68	32.47 ± 0.37	33.17	78.70 ± 0.25	75.43 ± 0.72	77.03
CoCoA-Mix	90.09 ± 0.16	90.93 ± 0.09	90.50	33.51 ± 0.28	<b>34.15 ± 0.14</b>	<b>33.83</b>	78.51 ± 0.17	76.60 ± 0.24	77.54

METHOD	DTD			EUROSAT			UCF101		
	BASE	NEW	H	BASE	NEW	H	BASE	NEW	H
CLIP	53.24	54.71	53.97	54.79	66.21	59.96	69.03	69.61	69.32
CoOp	71.22 ± 1.13	53.62 ± 3.45	61.03	79.93 ± 1.07	64.79 ± 6.36	71.19	80.58 ± 0.66	64.11 ± 2.84	71.32
ProGrAD	72.07 ± 0.29	50.56 ± 2.43	59.35	81.29 ± 3.36	69.81 ± 5.56	74.80	80.97 ± 0.29	73.32 ± 1.85	76.93
KgCoOp	72.92 ± 1.05	59.14 ± 1.53	65.28	83.20 ± 0.72	<b>70.51 ± 9.30</b>	<b>75.61</b>	80.09 ± 0.24	<b>77.75 ± 0.40</b>	78.90
MAPLE	70.40 ± 2.57	58.40 ± 3.00	63.71	76.50 ± 3.85	55.70 ± 3.19	64.27	78.57 ± 2.11	76.60 ± 1.56	77.53
DePT	<b>74.40 ± 0.83</b>	53.13 ± 1.07	61.98	78.70 ± 1.56	50.53 ± 5.71	61.08	<b>81.57 ± 0.84</b>	66.53 ± 0.87	73.28
CoA-LOSS	73.23 ± 2.02	58.09 ± 0.81	64.76	83.38 ± 0.49	70.07 ± 2.49	76.09	80.83 ± 0.80	74.22 ± 0.91	77.38
CoCoA-Mix	72.80 ± 1.89	<b>64.29 ± 1.25</b>	<b>68.25</b>	<b>83.49 ± 0.66</b>	69.11 ± 3.10	75.54	81.28 ± 0.95	<b>77.75 ± 0.24</b>	<b>79.47</b>



- Experiments: Cross-Dataset Transfer

- ☑ Is CoCoA-Mix capable of transferring learned knowledge effectively across different datasets?
- The prompt is trained on ImageNet with 1,000 classes and tested on 10 different datasets with non-overlapping classes

Tab 2. Performance comparison in cross-dataset transfer.

METHOD	SOURCE	TARGET	H
CLIP	66.73	64.89	63.97
CoOp	69.06 $\pm$ 0.43	59.88	61.52
ProGrad	70.21 $\pm$ 0.16	62.36	63.58
KGCoOp	70.52 $\pm$ 0.05	64.45	65.17
MAPLE	69.53 $\pm$ 0.39	65.24	65.26
DEPT	68.03 $\pm$ 0.09	65.06	64.42
CoCoA-MIX	70.85 $\pm$ 0.09	65.27	66.07

- Detailed results are provided in [Appendix C](#).

## ● Experiments: Few-Shot Class-Incremental Learning (FSCIL)

- ☑ Is CoCoA-Mix effective in mitigating forgetting and adapting to new tasks in few-shot class-incremental learning?
  - The number of prompts  $K + 1$  was increased incrementally, with each prompt  $t_i$  specializing in its session.

Tab 3. Performance comparison on CIFAR100 in the FSCIL benchmark. Mean represents the average accuracy across all sessions, and PD indicates the performance difference between the first and last sessions.

METHOD	ACC(%)↑									MEAN↑	PD↓
	0	1	2	3	4	5	6	7	8		
L2P	<b>89.9</b>	<b>86.0</b>	81.8	80.3	80.0	74.6	73.2	72.6	65.0	78.2	24.9
CLIP-ZSL	—	—	—	—	—	—	—	—	—	77.9	—
CoOp-FSCIL	88.6	78.9	77.5	76.0	76.8	78.3	79.2	79.8	79.3	79.4	9.3
FACT w/ CLIP	87.8	84.0	81.4	78.0	77.8	76.3	75.0	72.5	71.9	78.3	15.9
FSPT-FSCIL	86.9	83.1	81.9	80.7	80.4	79.9	80.1	79.9	79.4	81.4	7.5
CoCoA-MIX (OURS)	88.2	85.6	<b>84.6</b>	<b>82.7</b>	<b>82.8</b>	<b>82.5</b>	<b>82.3</b>	<b>81.8</b>	<b>80.8</b>	<b>83.5</b>	<b>7.4</b>



## Experiments: Ablation Studies

☑ Does CoA-loss improve specialization of prompt tuning?

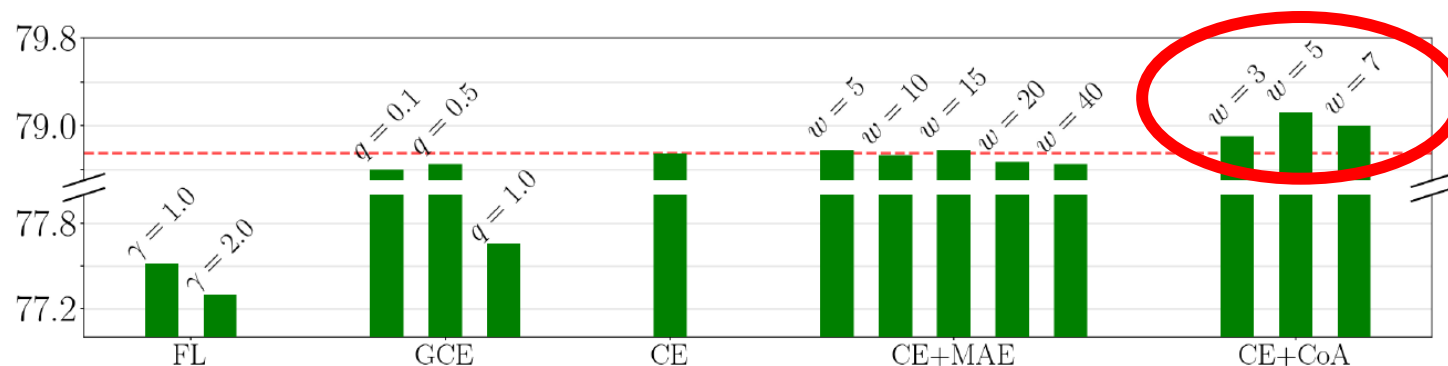


Fig 6. Performance comparison across various loss functions

☑ Is CoA-loss truly effective in handling confusing samples?

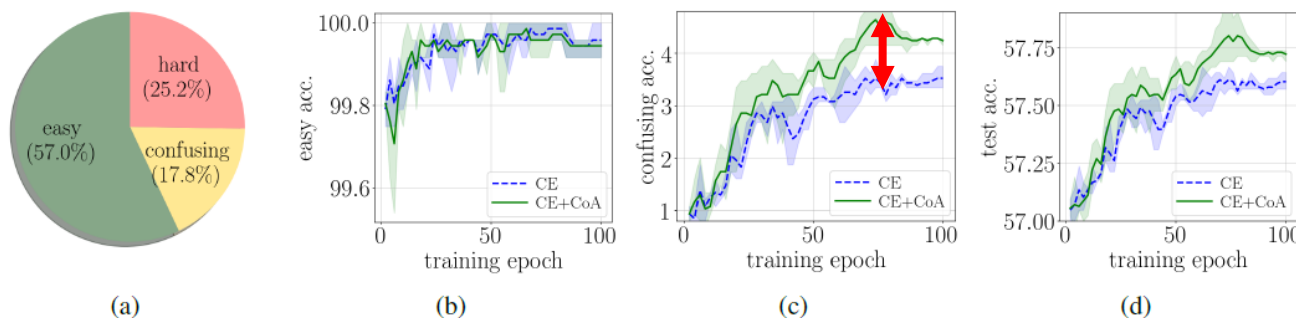


Fig 7. (a) Proportion of predictions by zero-shot CLIP on EuroSAT. (b) Accuracy on easy test samples correctly predicted by zero-shot CLIP. (c) Accuracy on confusing test samples misclassified by zero-shot CLIP with a probability gap below 0.2. (d) Accuracy on all test samples.

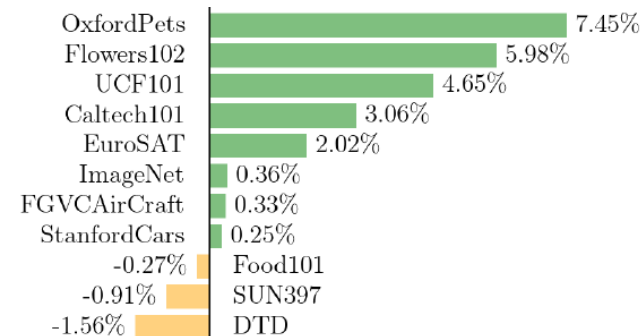


Fig 8. Performance improvement on confusing samples

## ● Experiments: Ablation Studies

- ☑ Does CoA-weights improve generalization of prompt tuning?

Tab 4. Effect of CoA-weights on *Base* and *New* classes.

$\pi_i^{\text{in}}$	$\pi_i^{\text{out}}$	BASE	NEW	H
✗	✗	79.12	73.66	76.15
✓	✗	79.30	73.81	76.32
✓	✓	<b>79.31</b>	<b>75.10</b>	<b>77.03</b>

- ☑ Are CoA-weights sensitive to the way the out-class set is generated?

Tab 5. Ablation study comparing different strategies for generating unseen classes. The table reports accuracy on *New* classes.

	NONE	RANDOM STRING	RANDOM STRING AND WORD	RANDOM WORD
ACCURACY	74.12	75.00	75.04	<b>75.10</b>

**Thanks for your kind attention**  
ds.hong@kaist.ac.kr



@



## ● Proof of Theorem 3.2

Consider  $K + 1$  individual prompts  $\mathcal{T} = \{t_0, t_1, \dots, t_K\}$  and a mixture model  $\hat{p}_{\mathcal{T}}^{\pi}$  with non-negative weights  $\pi = \{\pi_0, \pi_1, \dots, \pi_K\}$ , where  $\sum_{i=0}^K \pi_i = 1$ . Let  $\mathcal{D}_T$  be an arbitrary target domain. The expected error  $\epsilon_T(\hat{p}_{\mathcal{T}}^{\pi})$  of the mixture model on the target domain is defined as follows in terms of the Kullback-Leibler (KL) divergence:

$$\epsilon_T(\hat{p}_{\mathcal{T}}^{\pi}) = \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}_T} [-\log \hat{p}_{\mathcal{T}}^{\pi}(y)],$$

where  $y$  is the ground-truth label for the image  $\mathbf{x}$ .

Using the definition of the mixture model,  $\hat{p}_{\mathcal{T}}^{\pi}(y) = \frac{\exp(\sum_{i=0}^K \pi_i s_{t_i}(y)/\tau)}{\sum_{l' \in \mathcal{Y}} \exp(\sum_{i=0}^K \pi_i s_{t_i}(l')/\tau)}$ , the expected error can be decomposed into two terms as follows:

$$\begin{aligned} \epsilon_T(\hat{p}_{\mathcal{T}}^{\pi}) &= \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}_T} [-\log \hat{p}_{\mathcal{T}}^{\pi}(y)] \\ &= \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}_T} \left[ -\log \frac{\exp(\sum_{i=0}^K \pi_i s_{t_i}(y)/\tau)}{\sum_{l' \in \mathcal{Y}} \exp(\sum_{i=0}^K \pi_i s_{t_i}(l')/\tau)} \right] \\ &= \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}_T} \left[ -\sum_{i=0}^K \pi_i s_{t_i}(y)/\tau + \log \sum_{l' \in \mathcal{Y}} \exp\left(\sum_{i=0}^K \pi_i s_{t_i}(l')/\tau\right) \right] \\ &= \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}_T} \left[ -\sum_{i=0}^K \pi_i s_{t_i}(y)/\tau + \sum_{i=0}^K \pi_i \log \sum_{l' \in \mathcal{Y}} \exp(s_{t_i}(l')/\tau) \right] \\ &\quad + \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}_T} \left[ -\sum_{i=0}^K \pi_i \log \sum_{l' \in \mathcal{Y}} \exp(s_{t_i}(l')/\tau) + \log \sum_{l' \in \mathcal{Y}} \exp\left(\sum_{i=0}^K \pi_i s_{t_i}(l')/\tau\right) \right]. \end{aligned}$$

[◀ Go Back to Main](#)

The first term is rewritten using the definition of the individual predictive distribution  $\hat{p}_{t_i}$  for the prompt  $t_i$ , given as

$\hat{p}_{t_i}(y) = \frac{\exp(s_{t_i}(y)/\tau)}{\sum_{l' \in \mathcal{Y}} \exp(s_{t_i}(l')/\tau)}$ , as follows:

$$\begin{aligned} &\mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}_T} \left[ -\sum_{i=0}^K \pi_i s_{t_i}(y)/\tau + \sum_{i=0}^K \pi_i \log \sum_{l' \in \mathcal{Y}} \exp(s_{t_i}(l')/\tau) \right] \\ &= \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}_T} \left[ -\sum_{i=0}^K \pi_i \left( s_{t_i}(y)/\tau - \log \sum_{l' \in \mathcal{Y}} \exp(s_{t_i}(l')/\tau) \right) \right] \\ &= \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}_T} \left[ -\sum_{i=0}^K \pi_i \left( \log \exp(s_{t_i}(y)/\tau) - \log \sum_{l' \in \mathcal{Y}} \exp(s_{t_i}(l')/\tau) \right) \right] \\ &= \sum_{i=0}^K \pi_i \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}_T} \left[ -\log \frac{\exp(s_{t_i}(y)/\tau)}{\sum_{l' \in \mathcal{Y}} \exp(s_{t_i}(l')/\tau)} \right] \\ &= \sum_{i=0}^K \pi_i \epsilon_T(\hat{p}_{t_i}). \end{aligned}$$

As a result, the first term is equivalent to a convex combination of the expected errors of the individual predictive distributions with weights  $\pi$ .

For the second term, Jensen's inequality (Jensen, 1906) can be applied to bound it, as  $\log \sum \exp$  is a convex function:

$$\begin{aligned} &\mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}_T} \left[ -\sum_{i=0}^K \pi_i \log \sum_{l' \in \mathcal{Y}} \exp(s_{t_i}(l')/\tau) + \log \sum_{l' \in \mathcal{Y}} \exp\left(\sum_{i=0}^K \pi_i s_{t_i}(l')/\tau\right) \right] \\ &\leq \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}_T} \left[ -\sum_{i=0}^K \pi_i \log \sum_{l' \in \mathcal{Y}} \exp(s_{t_i}(l')/\tau) + \sum_{i=0}^K \pi_i \left( \log \sum_{l' \in \mathcal{Y}} \exp(s_{t_i}(l')/\tau) \right) \right] \\ &\leq 0. \end{aligned}$$

By combining the results from the first and second terms, we conclude that the expected error of the mixture model on the target domain is bounded as follows:

$$\epsilon_T(\hat{p}_{\mathcal{T}}^{\pi}) \leq \sum_i \pi_i \epsilon_T(\hat{p}_{t_i}).$$

## ● Effect of Cross-Entropy in the Mixture Model

[◀ Go Back to Main](#)

The derivative of the cross-entropy  $\mathcal{L}_{CE}$  for the mixture model  $\hat{p}_{\mathcal{T}}^{\pi}$  with respect to  $\pi_i^{\text{in}}$  is as follows:

$$\begin{aligned} \frac{\partial \mathcal{L}_{CE}(\mathbf{x}, y; \hat{p}_{\mathcal{T}}^{\pi})}{\partial \pi_i} &= \frac{\partial (-\log \hat{p}_{\mathcal{T}}^{\pi}(y))}{\partial \pi_i} \\ &= \frac{-1}{\hat{p}_{\mathcal{T}}^{\pi}(y)} \frac{\partial \hat{p}_{\mathcal{T}}^{\pi}(y)}{\partial \pi_i} \\ &= \frac{-1}{\hat{p}_{\mathcal{T}}^{\pi}(y)} \frac{\partial}{\partial \pi_i} \left( \frac{\exp \left( \sum_{i=0}^K \pi_i s_{t_i}(y) / \tau \right)}{\sum_{l' \in \mathcal{Y}} \exp \left( \sum_{i=0}^K \pi_i s_{t_i}(l') / \tau \right)} \right) \\ &= \frac{-1}{\hat{p}_{\mathcal{T}}^{\pi}(y)} \left( s_{t_i}(y) \hat{p}_{\mathcal{T}}^{\pi}(y) - \hat{p}_{\mathcal{T}}^{\pi}(y) \sum_{l \in \mathcal{Y}} \hat{p}_{\mathcal{T}}^{\pi}(l) s_{t_i}(l) \right) / \tau \\ &= - \left( s_{t_i}(y) - \sum_{l \in \mathcal{Y}} \hat{p}_{\mathcal{T}}^{\pi}(l) s_{t_i}(l) \right) / \tau \\ &= - (\hat{s}_{t_i}(y) - \hat{s}_{t_i}) / \tau, \end{aligned}$$

where  $\hat{s}_{t_i}$  is the importance-weighted similarity defined as a weighted sum of the predicted probability of the mixture model and the similarity derived from the prompt  $t_i$ , i.e.  $\hat{s}_{t_i} = \sum_{l \in \mathcal{Y}} \hat{p}_{\mathcal{T}}^{\pi}(l) s_{t_i}(l)$ . For example, if the mixture model predicts class  $l^*$  with the highest probability,  $\hat{s}_{t_i}$  approximates the similarity  $s_{t_i}(l^*)$  for class  $l^*$  derived from prompt  $t_i$ . Here, we explain how the CoA-weights  $\pi_i$  for in-classes is optimized through the cross-entropy of the mixture model. For simplicity, we assume  $\hat{s}_{t_i} \approx s_{t_i}(l^*)$ , where  $l^* = \arg \max_l \hat{p}_{\mathcal{T}}^{\pi}(l)$ .

In the case  $s_{t_i}(y) > \hat{s}_{t_i}$ , the prompt  $t_i$  predicts the correct class  $y$  with high similarity. Therefore, when the mixture model misclassifies, i.e.,  $l^* \neq y$ , the other prompts  $t_{j \neq i}$  provide low similarities for the correct class  $y$ . This case results in an increase in  $\pi_i$  through gradient updates, encouraging the mixture model to rely more on  $t_i$ .

Conversely, if  $s_{t_i}(y) < \hat{s}_{t_i}$ , the prompt  $t_i$  predicts the correct class  $y$  with low similarity. When the mixture model correctly classifies, i.e.  $l^* = y$ , it suggests that the other prompts  $t_{j \neq i}$  provide high similarities for the correct class  $y$ , while the prompt  $t_i$  underperforms. This case decreases  $\pi_i$ , allowing the mixture model to trust the other prompts  $t_{j \neq i}$  more.

## ● Cross-Dataset Transfer

[◀ Go Back to Main](#)

Table 6. Performance comparison on 11 datasets in cross-dataset transfer.

Method	SOURCE IMAGENET	TARGET AVERAGE	TARGET			
			CALTECH101	OXFORDPETS	STANFORDCARS	FLOWERS102
CLIP	66.73	64.89	93.27	89.18	65.56	68.05
CoOP	69.06	59.88 (−5.01)	91.06 (−2.21)	86.74 (−2.44)	59.84 (−5.72)	62.38 (−5.67)
ProGRAD	70.21	62.36 (−2.53)	92.41 (−0.86)	87.90 (−1.28)	62.94 (−2.62)	66.98 (−1.07)
KGCoOP	70.52	64.45 (−0.43)	93.55 (+0.28)	<b>89.86 (+0.68)</b>	65.61 (+0.05)	68.33 (+0.28)
MAPLE	69.53	65.24 (+0.35)	93.43 (+0.16)	89.77 (+0.59)	65.70 (+0.14)	<b>71.17 (+3.12)</b>
DEPT	68.03	65.06 (+0.17)	<b>94.07 (+0.80)</b>	89.43 (+0.25)	<b>65.87 (+0.31)</b>	69.93 (+1.88)
CoCoA-MIX	<b>70.85</b>	<b>65.27 (+0.38)</b>	93.46 (+0.19)	89.07 (−0.11)	65.59 (+0.03)	68.72 (+0.67)

Method	Target					
	FOOD101	FGVCAIRCRAFT	SUN397	DTD	EUROSAT	UCF101
CLIP	85.43	<b>24.81</b>	62.61	44.09	<b>48.36</b>	67.51
CoOP	83.29 (−2.14)	16.71 (−8.10)	59.40 (−3.21)	38.44 (−5.65)	39.24 (−9.12)	61.66 (−5.85)
ProGRAD	84.37 (−1.06)	17.10 (−7.71)	62.67 (+0.06)	39.87 (−4.22)	45.39 (−2.97)	63.98 (−3.53)
KGCoOP	85.83 (+0.40)	21.18 (−3.63)	64.84 (+2.23)	44.30 (+0.21)	44.64 (−3.72)	66.39 (−1.12)
MAPLE	86.13 (+0.70)	23.27 (−1.54)	<b>66.43 (+3.82)</b>	44.83 (+0.74)	43.73 (−4.63)	<b>67.93 (+0.42)</b>
DEPT	<b>86.27 (+0.84)</b>	22.10 (−2.71)	65.77 (+3.16)	45.53 (+1.44)	44.00 (−4.36)	67.60 (+0.09)
CoCoA-MIX	85.78 (+0.35)	24.10 (−0.71)	63.61 (+1.00)	<b>46.41 (+2.32)</b>	48.18 (−0.18)	67.78 (+0.27)