ProTeCt: Prompt Tuning for Taxonomic Open Set Classification

Supplementary Material

The appendix is organized as follows. Section 8 provides more training details for ProTeCt. Section 9 shows the complete proof of Lemma 4.1. Section 10 shows more examples for explaining the implementation of treecut sampler. Section 11, Section 12, and Section 13 shows the complete results conducted on Cifar100, Sun and ImageNet, respectively. Section 13 further shows the complete domain generalization results by applying the model trained on ImageNet to its 4 variants in a zero-shot fashion. We also test the robustness of ProTeCt on additional hierarchies in Section 14 with the FGVC Aircraft [29] dataset and the RSI-CB [22] satellite dataset. Ablations of different ProTeCt components are shown in Section 15 and more visualizations of incorrect predictions from existing prompt tuning methods are illustrated in Section 16.

8. Additional Training Details

In addition to the training details provided in the main paper, we list the url links that are used for training and evaluating ProTeCt. For CoOp and CoCoOp baselines, we adopt the code from https://github.com/KaiyangZhou/CoOp. For MaPLe, we adopt the code from https://github.com/muzairkhattak/multimodal-prompt-learning.

9. Treecut size of a balanced M-ary tree

Lemma 4.1. *For a balanced M-ary tree with depth L (root node is excluded and is at depth 0), the number of all valid treecut is* $L + \sum_{l=2}^{L} \sum_{k=1}^{N-1} \frac{N!}{k!(N-k)!}|_{N=M^{l-1}}$

Proof. This can be proved by induction. Given an M-ary tree with depth L, the number of treecuts is denoted as f_L. The idea is that when adding the depth L, we only need to recompute the additional possible treecuts between depth $L-1$ and L. Since there are $N = M^{L-1}$ nodes in layer $L-1$, the possible treecuts after adding layer L is $1 + \sum_{k=1}^{N-1} \frac{N!}{k!(N-k)!}|_{N=M^{L-1}}$, where 1 indicates the treecut that covers all nodes at depth L and $\sum_{k=1}^{N-1} \frac{N!}{k!(N-k)!}|_{N=M^{L-1}}$ means k nodes are covered in layer $L-1$. Below is the proof.

- When $L = 1$, $f_1 = 1$.
- When $L = 2$, $f_2 = 1 + f_1 + \sum_{k=1}^{N-1} \frac{N!}{k!(N-k)!}|_{N=M^{L-1}}$. Consider the binary case, where $M = 2$ and $N = M^{L-1} = 2$, then $f_2 = 1 + f_1 + \frac{2!}{1!(2-1)!} = 1 + 1 + 2 = 4$
- Similarly, $f_3 = 1 + f_2 + \sum_{k=1}^{N-1} \frac{N!}{k!(N-k)!}|_{N=M^{L-1}}$
- $$\begin{aligned} f_L &= 1 + f_{L-1} + \sum_{k=1}^{N-1} \frac{N!}{k!(N-k)!}|_{N=M^{L-1}} \\ &= 1 + 1 + f_{L-2} + \sum_{k=1}^{N-1} \frac{N!}{k!(N-k)!}|_{N=M^{L-2}} \\ &+ \sum_{k=1}^{N-1} \frac{N!}{k!(N-k)!}|_{N=M^{L-1}} \\ &= L + \sum_{l=2}^{L} \sum_{k=1}^{N-1} \frac{N!}{k!(N-k)!}|_{N=M^{l-1}} \end{aligned}$$

□

10. Additional Examples for Treecut Sampler

In this section, we provide more detailed examples of the proposed Treecut sampler (i.e. Algorithm 1 in the main paper). Given the class hierarchy $\mathcal{T}$ on the left of Figure 6, three possible treecuts can be sampled by $\mathcal{T}$, i.e., $\mathcal{Y}_{\mathcal{T}_c} = \{n_1, n_6\}$ (see Figure 6), $\mathcal{Y}_{\mathcal{T}_c} = \{n_2, n_3, n_6\}$ (see Figure 7), and $\mathcal{Y}_{\mathcal{T}_c} = \{n_3, n_4, n_5, n_6\}$ (see Figure 8), depending on the sampled values p_n at each internal node $n \in \mathcal{N}^{int} = \{n_0, n_1, n_2\}$. Note that p_{n_0} is always set to 1 to ensure that the tree is not entirely pruned. As described in the paper, we use a dependency matrix $\mathbf{D}$ to correct $\mathbf{p}$ as $\tilde{\mathbf{p}}$, which is aligned with the dependency relationship among the internal nodes. For example, in the example shown in Figure 6, $p_{n_2} = 1$ is corrected as $\tilde{p}_{n_2} = 0$, since n_2 depends on n_1 and $p_{n_1} = 0$. A mask $\mathbf{b}$, flagging the unavailable labels, is then computed according to the values of $\tilde{\mathbf{p}}$. More specifically, the corresponding row in $min(\mathbf{B}, 0)$ is fetched when $\tilde{p}_n = 1$, and that in $\bar{\mathbf{B}}$ is used when $\tilde{p}_n = 0$, for each internal node $n \in \mathcal{N}^{int}$. These masks are accumulated into the final mask $\mathbf{b}$, as shown on the right of each figure, where entries of 0 indicate the available labels for the sampled label set. For example, in Figure 7, the sampled label set contains $\{n_2, n_3, n_6\}$, because b_2, b_3 and b_6 are 0s. Note that since the proposed Treecut Sampler maintains the node dependency with pre-computed matrices defined by the given hierarchy $\mathcal{T}$, it does not require any recursive traversal over the tree, and thus it is very efficient for the *on-the-fly* treecut sampling.

11. Complete Table of Cifar100 Experiments

In this section, we report the complete experiment results conducted on Cifar100. Table 6, Table 7 and Table 8 shows the results of vanilla CoOp and its results after adding ProTeCt. The CLIP features from ViT B16, ViT B32 and ViT

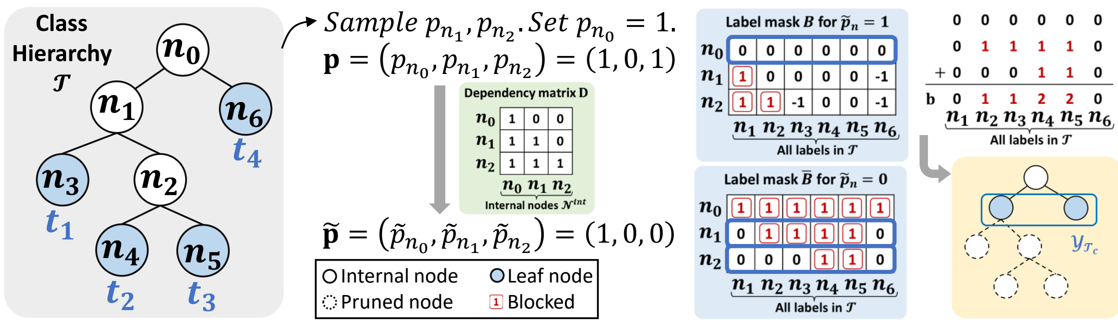

Figure 6. Treecut example of $\mathcal{Y}_{\mathcal{T}_c} = \{n_1, n_6\}$.

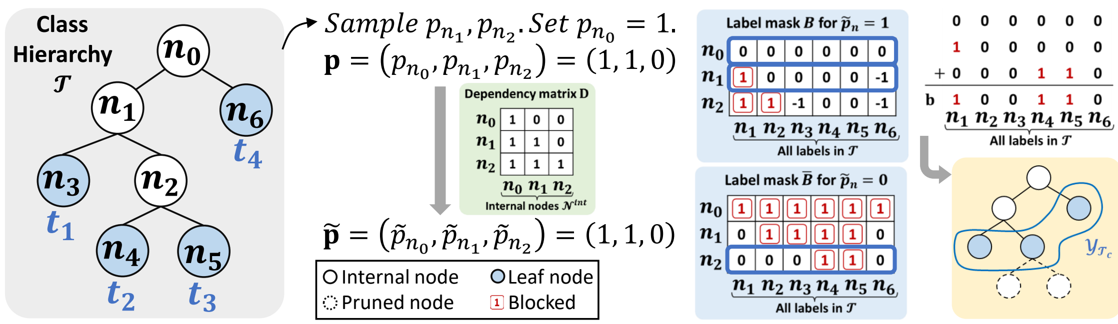

Figure 7. Treecut example of $\mathcal{Y}_{\mathcal{T}_c} = \{n_2, n_3, n_6\}$.

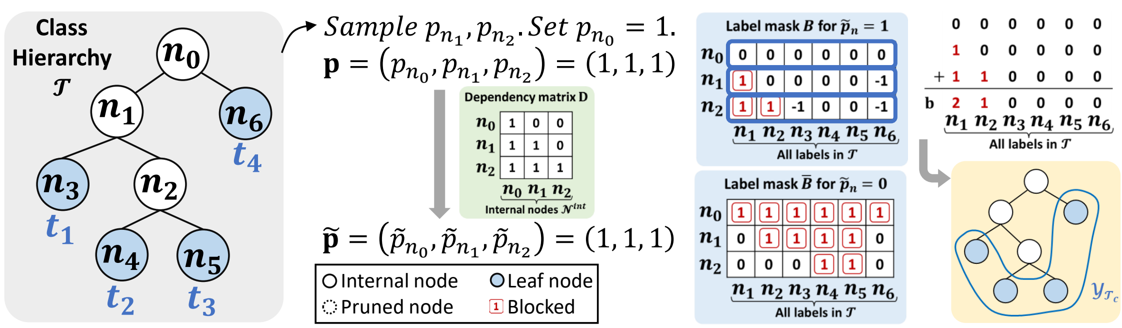

Figure 8. Treecut example of $\mathcal{Y}_{\mathcal{T}_c} = \{n_3, n_4, n_5, n_6\}$.

L14 are considered in Table 6, Table 7 and Table 8, respectively. While it is known that CLIP ViT L14 has a more powerful representation than ViT B32 and ViT B16 (also reflected in the leaf accuracy between three tables), all of them perform equally poor in terms of HCA (10.04/4.95/11.14 for 16-shot CoOp using CLIP B16/B32/L14 feature). This shows that simply using a stronger CLIP feature does not address the problem of hierarchical classification and does not improve hierarchical consistency. Furthermore, Table 6 contains the result of ProTeCt without using the treecut sampler ($\beta = 0$; Block 2 and Block 3) and without using NCL loss of (7) ($\lambda = 0$; Block 4) under multiple low-shot settings. For example, when 16-shot is considered, adding both NCL loss and treecut sampler ($\lambda = 0.5$ and $\beta = 0.1$) gives the result of 56.85 for HCA. Removing the tree dropout ($\lambda = 0.5$ and $\beta = 0$) yields 51.99 and removing the NCL loss ($\lambda = 0$ and $\beta = 0.1$) yields 47.97. This shows that both the NCL loss and the treecut sampler are important and lead to a significant gain over vanilla CoOp (HCA=10.04). Table 9 shows similar results when adding ProTeCt on MaPLe. Furthermore, we sampled T treecuts for each dropout rate $\beta = \{0.1, 0.3, 0.5, 0.7, 0.9\}$, where $T = 5$ and $T = 20$, resulting in 25 and 100 treecuts, respectively. Table 10 demonstrates ProTeCt can improve the MTA metric for both CoOp and MaPLe for both 25 and 100 randomly sampled treecuts. Table 11 further shows that ProTeCt can generalize to ResNet-based architectures.

Method	Encoder	K-shot	w/ ProTeCt	λ	β	Leaf Acc.	HCA
CoOp	ViT B16	16		N/A	N/A	72.88 ± 0.62	10.04 ± 1.11
CoOp	ViT B16	8		N/A	N/A	70.84 ± 0.85	6.03 ± 0.64
CoOp	ViT B16	4		N/A	N/A	69.47 ± 0.90	6.15 ± 1.04
CoOp	ViT B16	2		N/A	N/A	68.17 ± 0.57	4.19 ± 0.81
CoOp	ViT B16	1		N/A	N/A	65.03 ± 0.56	7.81 ± 0.14
CoOp	ViT B16	16	✓	0.5	0	72.08 ± 0.38	51.99 ± 0.24
CoOp	ViT B16	8	✓	0.5	0	68.94 ± 0.52	49.01 ± 0.54
CoOp	ViT B16	4	✓	0.5	0	66.38 ± 1.18	45.24 ± 0.93
CoOp	ViT B16	2	✓	0.5	0	63.96 ± 0.57	42.78 ± 1.49
CoOp	ViT B16	1	✓	0.5	0	62.01 ± 0.80	34.90 ± 1.08
CoOp	ViT B16	16	✓	1	0	70.86 ± 0.59	54.39 ± 0.68
CoOp	ViT B16	8	✓	1	0	68.76 ± 0.90	52.14 ± 0.32
CoOp	ViT B16	4	✓	1	0	66.92 ± 0.20	47.63 ± 0.54
CoOp	ViT B16	2	✓	1	0	64.87 ± 1.28	40.74 ± 0.87
CoOp	ViT B16	1	✓	1	0	62.57 ± 0.06	38.97 ± 1.29
CoOp	ViT B16	16	✓	0	0.1	72.81 ± 0.31	47.97 ± 0.70
CoOp	ViT B16	8	✓	0	0.1	70.94 ± 0.18	48.53 ± 0.02
CoOp	ViT B16	4	✓	0	0.1	69.10 ± 0.92	45.20 ± 0.25
CoOp	ViT B16	2	✓	0	0.1	68.85 ± 0.11	42.28 ± 1.57
CoOp	ViT B16	1	✓	0	0.1	64.77 ± 1.37	32.93 ± 0.42
CoOp	ViT B16	16	✓	0.5	0.1	72.94 ± 0.83	56.85 ± 1.60
CoOp	ViT B16	8	✓	0.5	0.1	71.10 ± 1.06	52.27 ± 0.62
CoOp	ViT B16	4	✓	0.5	0.1	69.46 ± 0.58	48.71 ± 0.13
CoOp	ViT B16	2	✓	0.5	0.1	68.63 ± 0.67	46.03 ± 0.24
CoOp	ViT B16	1	✓	0.5	0.1	66.88 ± 0.21	41.01 ± 1.18
CoOp	ViT B16	16	✓	1	0.1	73.26 ± 0.66	58.01 ± 0.43
CoOp	ViT B16	8	✓	1	0.1	70.10 ± 0.08	52.81 ± 0.05
CoOp	ViT B16	4	✓	1	0.1	68.41 ± 0.50	49.59 ± 0.89
CoOp	ViT B16	2	✓	1	0.1	67.73 ± 1.25	45.27 ± 0.28
CoOp	ViT B16	1	✓	1	0.1	63.84 ± 1.51	40.05 ± 1.48

Table 6. Performance of few-shot CoOp on Cifar100 under ViT B16. Ablations cover both NCL strengths λ and tree dropout rate β.

Method	Encoder	K-shot	w/ ProTeCt	λ	β	Leaf Acc.	HCA
CoOp	ViT B32	16		N/A	N/A	68.13 ± 0.19	4.95 ± 0.61
CoOp	ViT B32	8		N/A	N/A	65.52 ± 0.15	5.82 ± 0.29
CoOp	ViT B32	4		N/A	N/A	63.42 ± 1.40	8.56 ± 0.72
CoOp	ViT B32	2		N/A	N/A	63.65 ± 0.60	10.25 ± 0.88
CoOp	ViT B32	1		N/A	N/A	59.53 ± 0.60	3.43 ± 0.86
CoOp	ViT B32	16	✓	0	0.1	68.42 ± 0.91	47.79 ± 0.54
CoOp	ViT B32	8	✓	0	0.1	66.39 ± 0.48	44.47 ± 0.98
CoOp	ViT B32	4	✓	0	0.1	64.73 ± 0.17	31.72 ± 0.33
CoOp	ViT B32	2	✓	0	0.1	64.55 ± 0.44	30.78 ± 0.66
CoOp	ViT B32	1	✓	0	0.1	60.91 ± 0.42	34.64 ± 0.55
CoOp	ViT B32	16	✓	0.5	0.1	68.87 ± 1.09	51.55 ± 0.65
CoOp	ViT B32	8	✓	0.5	0.1	66.85 ± 0.32	48.39 ± 1.35
CoOp	ViT B32	4	✓	0.5	0.1	65.41 ± 0.74	41.63 ± 0.39
CoOp	ViT B32	2	✓	0.5	0.1	62.86 ± 0.81	38.13 ± 0.61
CoOp	ViT B32	1	✓	0.5	0.1	61.59 ± 0.80	35.65 ± 0.19
CoOp	ViT B32	16	✓	1	0.1	68.93 ± 0.22	51.67 ± 0.58
CoOp	ViT B32	8	✓	1	0.1	65.54 ± 0.54	48.36 ± 0.63
CoOp	ViT B32	4	✓	1	0.1	64.28 ± 0.07	42.78 ± 1.04
CoOp	ViT B32	2	✓	1	0.1	61.68 ± 0.67	40.53 ± 0.42
CoOp	ViT B32	1	✓	1	0.1	58.98 ± 0.88	36.59 ± 0.76

Table 7. Performance of few-shot CoOp on Cifar100 under ViT B32. Ablations cover different NCL strengths λ.

Method	Encoder	K-shot	w/ ProTeCt	λ	β	Leaf Acc.	HCA
CoOp	ViT L14	16		N/A	N/A	79.98 ± 0.97	11.14 ± 0.47
CoOp	ViT L14	8		N/A	N/A	79.37 ± 0.90	6.91 ± 0.67
CoOp	ViT L14	4		N/A	N/A	77.34 ± 0.78	7.78 ± 0.82
CoOp	ViT L14	2		N/A	N/A	76.63 ± 0.65	5.21 ± 0.87
CoOp	ViT L14	1		N/A	N/A	73.26 ± 0.95	4.87 ± 0.15
CoOp	ViT L14	16	✓	0	0.1	81.17 ± 0.34	63.40 ± 0.30
CoOp	ViT L14	8	✓	0	0.1	80.00 ± 0.98	62.11 ± 0.81
CoOp	ViT L14	4	✓	0	0.1	79.05 ± 0.68	57.19 ± 0.26
CoOp	ViT L14	2	✓	0	0.1	78.53 ± 0.69	40.59 ± 0.68
CoOp	ViT L14	1	✓	0	0.1	76.48 ± 0.52	45.11 ± 0.68
CoOp	ViT L14	16	✓	0.5	0.1	80.95 ± 0.38	68.92 ± 0.77
CoOp	ViT L14	8	✓	0.5	0.1	79.87 ± 0.11	64.05 ± 0.57
CoOp	ViT L14	4	✓	0.5	0.1	79.18 ± 0.51	51.88 ± 0.45
CoOp	ViT L14	2	✓	0.5	0.1	76.76 ± 0.24	51.96 ± 0.06
CoOp	ViT L14	1	✓	0.5	0.1	73.89 ± 0.62	50.31 ± 1.02
CoOp	ViT L14	16	✓	1	0.1	80.45 ± 0.90	70.15 ± 0.98
CoOp	ViT L14	8	✓	1	0.1	79.25 ± 0.93	65.75 ± 0.69
CoOp	ViT L14	4	✓	1	0.1	78.37 ± 0.13	47.30 ± 0.20
CoOp	ViT L14	2	✓	1	0.1	75.21 ± 0.33	54.78 ± 0.66
CoOp	ViT L14	1	✓	1	0.1	74.93 ± 0.15	52.08 ± 1.04

Table 8. Performance of few-shot CoOp on Cifar100 under both ViT L14. Ablations cover different NCL strengths λ.

Method	Encoder	K-shot	w/ ProTeCt	λ	β	Leaf Acc.	HCA
MaPLe	ViT B16	16		N/A	N/A	75.01 ± 0.37	17.54 ± 0.83
MaPLe	ViT B16	8		N/A	N/A	73.93 ± 0.46	9.44 ± 1.13
MaPLe	ViT B16	4		N/A	N/A	72.68 ± 0.47	20.29 ± 1.07
MaPLe	ViT B16	2		N/A	N/A	71.37 ± 1.39	12.15 ± 0.25
MaPLe	ViT B16	1		N/A	N/A	68.75 ± 0.96	4.65 ± 1.52
MaPLe	ViT B16	16	✓	0	0.1	75.82 ± 0.10	58.63 ± 0.43
MaPLe	ViT B16	8	✓	0	0.1	74.29 ± 0.91	57.31 ± 0.79
MaPLe	ViT B16	4	✓	0	0.1	72.92 ± 0.42	54.12 ± 1.56
MaPLe	ViT B16	2	✓	0	0.1	71.09 ± 1.35	47.78 ± 0.64
MaPLe	ViT B16	1	✓	0	0.1	68.32 ± 0.20	39.43 ± 0.25
MaPLe	ViT B16	16	✓	0.5	0.1	75.34 ± 0.39	61.15 ± 0.53
MaPLe	ViT B16	8	✓	0.5	0.1	74.30 ± 0.29	60.24 ± 0.82
MaPLe	ViT B16	4	✓	0.5	0.1	71.35 ± 0.61	56.03 ± 0.35
MaPLe	ViT B16	2	✓	0.5	0.1	70.24 ± 1.01	52.56 ± 0.48
MaPLe	ViT B16	1	✓	0.5	0.1	69.33 ± 0.81	48.10 ± 0.26
MaPLe	ViT B16	16	✓	1	0.1	76.30 ± 0.56	62.04 ± 0.97
MaPLe	ViT B16	8	✓	1	0.1	73.60 ± 0.69	61.20 ± 0.77
MaPLe	ViT B16	4	✓	1	0.1	72.06 ± 0.34	56.51 ± 1.24
MaPLe	ViT B16	2	✓	1	0.1	69.95 ± 1.30	53.53 ± 0.67
MaPLe	ViT B16	1	✓	1	0.1	70.44 ± 0.10	46.94 ± 0.85

Table 9. Performance of few-shot MaPLe on Cifar100. Ablations cover different NCL strengths λ.

Method	Encoder	K-shot	w/ ProTeCt	λ	β	MTA (25)	MTA (100)
CoOp	ViT B32	16		N/A	N/A	52.33	54.58
CoOp	ViT B32	8		N/A	N/A	46.09	47.20
CoOp	ViT B32	4		N/A	N/A	53.35	54.30
CoOp	ViT B32	2		N/A	N/A	53.13	53.81
CoOp	ViT B32	1		N/A	N/A	38.80	40.16
CoOp	ViT B32	16	✓	0.5	0.1	86.26	85.73
CoOp	ViT B32	8	✓	0.5	0.1	85.05	84.57
CoOp	ViT B32	4	✓	0.5	0.1	81.01	80.61
CoOp	ViT B32	2	✓	0.5	0.1	79.95	79.98
CoOp	ViT B32	1	✓	0.5	0.1	78.08	76.95
CoOp	ViT B16	16		N/A	N/A	50.64	51.14
CoOp	ViT B16	8		N/A	N/A	47.95	50.41
CoOp	ViT B16	4		N/A	N/A	43.77	46.29
CoOp	ViT B16	2		N/A	N/A	40.81	42.95
CoOp	ViT B16	1		N/A	N/A	41.78	44.17
CoOp	ViT B16	16	✓	0.5	0.1	87.69	87.30
CoOp	ViT B16	8	✓	0.5	0.1	86.28	86.01
CoOp	ViT B16	4	✓	0.5	0.1	84.52	83.79
CoOp	ViT B16	2	✓	0.5	0.1	83.49	83.18
CoOp	ViT B16	1	✓	0.5	0.1	81.64	81.01
CoOp	ViT L14	16		N/A	N/A	58.81	60.89
CoOp	ViT L14	8		N/A	N/A	40.49	43.20
CoOp	ViT L14	4		N/A	N/A	44.71	47.39
CoOp	ViT L14	2		N/A	N/A	39.44	43.22
CoOp	ViT L14	1		N/A	N/A	52.32	54.90
CoOp	ViT L14	16	✓	0.5	0.1	90.83	90.48
CoOp	ViT L14	8	✓	0.5	0.1	89.39	89.16
CoOp	ViT L14	4	✓	0.5	0.1	84.48	84.79
CoOp	ViT L14	2	✓	0.5	0.1	85.57	85.29
CoOp	ViT L14	1	✓	0.5	0.1	83.65	83.52
MaPLe	ViT B16	16		N/A	N/A	52.21	50.82
MaPLe	ViT B16	8		N/A	N/A	58.56	61.48
MaPLe	ViT B16	4		N/A	N/A	66.14	67.06
MaPLe	ViT B16	2		N/A	N/A	55.98	57.59
MaPLe	ViT B16	1		N/A	N/A	50.60	54.99
MaPLe	ViT B16	16	✓	0.5	0.1	88.04	88.33
MaPLe	ViT B16	8	✓	0.5	0.1	87.65	88.13
MaPLe	ViT B16	4	✓	0.5	0.1	86.72	87.04
MaPLe	ViT B16	2	✓	0.5	0.1	85.03	85.39
MaPLe	ViT B16	1	✓	0.5	0.1	83.36	83.78

Table 10. Performance of MTA for both few-shot CoOp and MaPLe on Cifar100. 25 ($T = 5$) and 100 ($T = 20$) treecuts are sampled for MTA evaluation.

Method	Encoder	K-shot	w/ ProTeCt	Acc_{leaf}	HCA	MTA (25)
CoOp	ResNet-50	16		52.61	5.72	41.97
CoOp	ResNet-50	16	✓	52.83	33.34	79.06
CoOp	ResNet-101	16		56.97	5.58	53.43
CoOp	ResNet-101	16	✓	57.64	39.93	81.76

Table 11. CoOp 16-shot results on Cifar100 with ResNets.

12. Complete Table of SUN Experiments

In this section, we report the complete experiment result conducted on SUN. Table 12 and Table 13 show the results of vanilla CoOp and MaPLe, and their results after adding ProTeCt. When comparing the HCA results of the vanilla prompt tuning with that of Cifar100 and ImageNet, the HCA result on SUN is much higher and the gap between HCA and Acc_{leaf} is much smaller. This is due to the shallow hierarchy of SUN dataset, indicating SUN is a much simpler dataset for hierarchical classification. However, we still see that ProTeCt achieves consistent improvement over the vanilla prompt tuning methods. Table 14 further compares the MTA result of vanilla CoOp and MaPLe, and their ProTeCt counterpart.

Method	Encoder	K-shot	w/ ProTeCt	λ	β	Leaf Acc.	HCA
CoOp	ViT B16	16		N/A	N/A	73.82 ± 0.12	38.28 ± 0.46
CoOp	ViT B16	8		N/A	N/A	71.77 ± 0.67	33.95 ± 0.08
CoOp	ViT B16	4		N/A	N/A	69.31 ± 0.51	30.51 ± 0.71
CoOp	ViT B16	2		N/A	N/A	66.34 ± 0.33	36.85 ± 0.67
CoOp	ViT B16	1		N/A	N/A	63.65 ± 1.42	33.36 ± 0.21
CoOp	ViT B16	16	✓	0	0.1	74.95 ± 0.69	60.95 ± 0.91
CoOp	ViT B16	8	✓	0	0.1	72.31 ± 0.18	57.61 ± 1.31
CoOp	ViT B16	4	✓	0	0.1	69.53 ± 0.77	54.79 ± 0.12
CoOp	ViT B16	2	✓	0	0.1	67.01 ± 1.10	50.78 ± 0.03
CoOp	ViT B16	1	✓	0	0.1	64.45 ± 0.96	47.75 ± 0.11
CoOp	ViT B16	8	✓	0.5	0.1	74.59 ± 0.41	62.94 ± 0.15
CoOp	ViT B16	4	✓	0.5	0.1	71.53 ± 0.67	58.17 ± 0.33
CoOp	ViT B16	2	✓	0.5	0.1	69.80 ± 0.98	56.85 ± 0.41
CoOp	ViT B16	16	✓	0.5	0.1	67.29 ± 1.32	51.82 ± 1.20
CoOp	ViT B16	1	✓	0.5	0.1	63.79 ± 1.16	49.62 ± 1.40
CoOp	ViT B16	16	✓	1	0.1	74.31 ± 0.23	62.96 ± 0.61
CoOp	ViT B16	8	✓	1	0.1	71.27 ± 0.42	58.74 ± 0.98
CoOp	ViT B16	4	✓	1	0.1	68.81 ± 0.71	55.90 ± 0.09
CoOp	ViT B16	2	✓	1	0.1	67.66 ± 0.51	50.94 ± 1.31
CoOp	ViT B16	1	✓	1	0.1	63.95 ± 1.19	50.99 ± 1.21

Table 12. Performance of few-shot CoOp on SUN. Ablations cover different NCL strengths λ.

Method	Encoder	K-shot	w/ ProTeCt	λ	β	Leaf Acc.	HCA
MaPLe	ViT B16	16		N/A	N/A	71.86 ± 0.11	33.25 ± 1.31
MaPLe	ViT B16	8		N/A	N/A	68.96 ± 0.51	29.63 ± 0.19
MaPLe	ViT B16	4		N/A	N/A	67.27 ± 0.45	25.97 ± 0.53
MaPLe	ViT B16	2		N/A	N/A	65.33 ± 1.21	29.79 ± 0.13
MaPLe	ViT B16	1		N/A	N/A	63.98 ± 0.99	25.15 ± 0.76
MaPLe	ViT B16	16	✓	0	0.1	72.89 ± 0.77	56.52 ± 0.88
MaPLe	ViT B16	8	✓	0	0.1	71.24 ± 0.76	55.49 ± 1.05
MaPLe	ViT B16	4	✓	0	0.1	69.24 ± 0.41	51.88 ± 1.22
MaPLe	ViT B16	2	✓	0	0.1	66.98 ± 0.44	51.60 ± 0.55
MaPLe	ViT B16	1	✓	0	0.1	63.80 ± 1.51	47.93 ± 0.31
MaPLe	ViT B16	16	✓	0.5	0.1	72.17 ± 1.20	59.71 ± 0.04
MaPLe	ViT B16	8	✓	0.5	0.1	71.04 ± 0.09	57.78 ± 1.22
MaPLe	ViT B16	4	✓	0.5	0.1	68.64 ± 0.61	54.86 ± 1.08
MaPLe	ViT B16	2	✓	0.5	0.1	66.37 ± 0.62	53.13 ± 0.39
MaPLe	ViT B16	1	✓	0.5	0.1	64.29 ± 1.23	50.45 ± 0.40
MaPLe	ViT B16	16	✓	1	0.1	71.03 ± 0.99	59.92 ± 0.06
MaPLe	ViT B16	8	✓	1	0.1	69.66 ± 0.16	57.60 ± 0.81
MaPLe	ViT B16	4	✓	1	0.1	66.96 ± 0.31	53.61 ± 0.55
MaPLe	ViT B16	2	✓	1	0.1	66.74 ± 0.36	53.54 ± 0.76
MaPLe	ViT B16	1	✓	1	0.1	63.46 ± 0.14	50.49 ± 1.01

Table 13. Performance of few-shot MaPLe on SUN. Ablations cover different NCL strengths λ.

Method	Encoder	K-shot	w/ ProTeCt	λ	β	MTA
CoOp	ViT B16	16		N/A	N/A	52.99
CoOp	ViT B16	8		N/A	N/A	55.24
CoOp	ViT B16	4		N/A	N/A	49.48
CoOp	ViT B16	2		N/A	N/A	51.94
CoOp	ViT B16	1		N/A	N/A	51.20
CoOp	ViT B16	16	✓	0.5	0.1	83.51
CoOp	ViT B16	8	✓	0.5	0.1	81.34
CoOp	ViT B16	4	✓	0.5	0.1	80.30
CoOp	ViT B16	2	✓	0.5	0.1	76.59
CoOp	ViT B16	1	✓	0.5	0.1	76.25
MaPLe	ViT B16	16		N/A	N/A	54.29
MaPLe	ViT B16	8		N/A	N/A	53.24
MaPLe	ViT B16	4		N/A	N/A	55.79
MaPLe	ViT B16	2		N/A	N/A	51.30
MaPLe	ViT B16	1		N/A	N/A	50.31
MaPLe	ViT B16	16	✓	0.5	0.1	82.27
MaPLe	ViT B16	8	✓	0.5	0.1	80.71
MaPLe	ViT B16	4	✓	0.5	0.1	79.10
MaPLe	ViT B16	2	✓	0.5	0.1	77.55
MaPLe	ViT B16	1	✓	0.5	0.1	76.73

Table 14. Performance of MTA for both few-shot CoOp and MaPLe on Sun.

13. Complete Table of ImageNet Experiments

In this section, we report the complete experiment result conducted on ImageNet. Table 15 first show the performance of CLIP and CoCoOp as a complement of Table 1 in the main paper. Note that none of the CLIP features (e.g. ViT B32, ViT B16, RN50, RN101) nor existing prompt tuning methods help the HCA metric. Table 16 and Table 17 show the results of vanilla CoOp and MaPLe, and their results after adding ProTeCt. Table 18 further compares the MTA result of vanilla CoOp and MaPLe, and their ProTeCt counterpart. Clearly, adding ProTeCt boosts both HCA and MTA. Furthermore, we apply the model trained on ImageNet to its four variants. Table 19, Table 20, Table 21 and Table 22 report the domain generalization results on ImageNetV2, ImageNet-sketch, ImageNet-A and ImageNet-R datasets for Acc_{leaf}, HCA and MTA. All four tables show that ProTeCt can not only improves the hierarchical consistency on the seen dataset, but also unseen datasets from other image domains.

Method	Encoder	K-shot	w/ ProTeCt	λ	β	Leaf Acc.	HCA
CLIP	ViT-B32	0		N/A	N/A	63.31	4.29
CLIP	ViT-B16	0		N/A	N/A	68.36	3.32
CLIP	RN50	0		N/A	N/A	59.81	4.16
CLIP	RN101	0		N/A	N/A	62.30	2.03
CoCoOp	ViT-B16	16		N/A	N/A	71.20 ± 0.13	2.92 ± 1.23

Table 15. Performance of zero-shot CLIP and 16-shot CoCoOp on ImageNet.

Method	Encoder	K-shot	w/ ProTeCt	λ	β	Leaf Acc.	HCA
CoOp	ViT B16	16		N/A	N/A	71.23 ± 0.67	2.99 ± 1.04
CoOp	ViT B16	8		N/A	N/A	69.40 ± 0.52	3.00 ± 0.58
CoOp	ViT B16	4		N/A	N/A	68.06 ± 0.42	2.95 ± 0.62
CoOp	ViT B16	2		N/A	N/A	65.46 ± 0.77	1.56 ± 0.17
CoOp	ViT B16	1		N/A	N/A	63.67 ± 0.85	1.59 ± 0.43
CoOp	ViT B16	16	✓	0	0.1	70.47 ± 0.22	27.81 ± 0.71
CoOp	ViT B16	8	✓	0	0.1	70.03 ± 0.14	26.17 ± 0.52
CoOp	ViT B16	4	✓	0	0.1	69.32 ± 0.11	21.99 ± 0.10
CoOp	ViT B16	2	✓	0	0.1	68.09 ± 0.23	20.92 ± 1.02
CoOp	ViT B16	1	✓	0	0.1	67.26 ± 0.65	18.69 ± 1.12
CoOp	ViT B16	8	✓	0.5	0.1	70.27 ± 0.36	34.63 ± 0.33
CoOp	ViT B16	4	✓	0.5	0.1	69.65 ± 0.41	31.84 ± 0.35
CoOp	ViT B16	2	✓	0.5	0.1	68.09 ± 0.16	27.05 ± 0.27
CoOp	ViT B16	16	✓	0.5	0.1	67.24 ± 0.24	26.09 ± 0.53
CoOp	ViT B16	1	✓	0.5	0.1	66.69 ± 0.15	23.79 ± 0.15
CoOp	ViT B16	16	✓	1	0.1	69.92 ± 0.21	37.74 ± 0.12
CoOp	ViT B16	8	✓	1	0.1	69.34 ± 0.17	34.66 ± 0.55
CoOp	ViT B16	4	✓	1	0.1	68.06 ± 0.44	30.87 ± 0.32
CoOp	ViT B16	2	✓	1	0.1	67.12 ± 0.35	26.34 ± 1.1
CoOp	ViT B16	1	✓	1	0.1	66.11 ± 0.50	25.79 ± 0.06

Table 16. Performance of few-shot CoOp on ImageNet. Ablations cover different NCL strengths λ.

Method	Encoder	K-shot	w/ ProTeCt	λ	β	Leaf Acc.	HCA
MaPLe	ViT B16	16		N/A	N/A	70.70 ± 0.11	4.15 ± 1.05
MaPLe	ViT B16	8		N/A	N/A	70.44 ± 0.06	4.32 ± 0.90
MaPLe	ViT B16	4		N/A	N/A	70.20 ± 0.06	2.95 ± 0.87
MaPLe	ViT B16	2		N/A	N/A	69.74 ± 0.25	4.27 ± 1.32
MaPLe	ViT B16	1		N/A	N/A	68.91 ± 0.13	2.97 ± 1.08
MaPLe	ViT B16	16	✓	0	0.1	70.08 ± 0.26	23.38 ± 1.43
MaPLe	ViT B16	8	✓	0	0.1	69.00 ± 0.26	21.71 ± 0.64
MaPLe	ViT B16	4	✓	0	0.1	68.50 ± 0.41	19.03 ± 0.21
MaPLe	ViT B16	2	✓	0	0.1	67.45 ± 0.32	17.54 ± 0.52
MaPLe	ViT B16	1	✓	0	0.1	67.03 ± 0.11	16.54 ± 0.32
MaPLe	ViT B16	16	✓	0.5	0.1	69.59 ± 0.25	27.74 ± 1.31
MaPLe	ViT B16	8	✓	0.5	0.1	69.06 ± 0.49	25.25 ± 0.52
MaPLe	ViT B16	4	✓	0.5	0.1	68.13 ± 0.01	25.25 ± 0.12
MaPLe	ViT B16	2	✓	0.5	0.1	67.45 ± 0.43	20.14 ± 1.07
MaPLe	ViT B16	1	✓	0.5	0.1	66.80 ± 0.26	20.62 ± 0.65
MaPLe	ViT B16	16	✓	1	0.1	69.52 ± 0.71	31.24 ± 1.02
MaPLe	ViT B16	8	✓	1	0.1	68.48 ± 0.06	26.92 ± 0.42
MaPLe	ViT B16	4	✓	1	0.1	68.59 ± 0.17	26.28 ± 0.31
MaPLe	ViT B16	2	✓	1	0.1	67.12 ± 0.11	22.96 ± 0.05
MaPLe	ViT B16	1	✓	1	0.1	66.16 ± 0.88	20.44 ± 0.77

Table 17. Performance of few-shot MaPLe on ImageNet. Ablations cover different NCL strengths λ.

Method	Encoder	K-shot	w/ ProTeCt	λ	β	MTA
CoOp	ViT B16	16		N/A	N/A	46.98
CoOp	ViT B16	8		N/A	N/A	46.04
CoOp	ViT B16	4		N/A	N/A	42.57
CoOp	ViT B16	2		N/A	N/A	44.89
CoOp	ViT B16	1		N/A	N/A	40.52
CoOp	ViT B16	16	✓	0.5	0.1	88.61
CoOp	ViT B16	8	✓	0.5	0.1	87.86
CoOp	ViT B16	4	✓	0.5	0.1	87.37
CoOp	ViT B16	2	✓	0.5	0.1	86.14
CoOp	ViT B16	1	✓	0.5	0.1	86.14
MaPLe	ViT B16	16		N/A	N/A	48.29
MaPLe	ViT B16	8		N/A	N/A	45.84
MaPLe	ViT B16	4		N/A	N/A	51.84
MaPLe	ViT B16	2		N/A	N/A	48.17
MaPLe	ViT B16	1		N/A	N/A	48.16
MaPLe	ViT B16	16	✓	0.5	0.1	87.87
MaPLe	ViT B16	8	✓	0.5	0.1	87.26
MaPLe	ViT B16	4	✓	0.5	0.1	86.85
MaPLe	ViT B16	2	✓	0.5	0.1	85.93
MaPLe	ViT B16	1	✓	0.5	0.1	85.18

Table 18. Performance of MTA for both few-shot CoOp and MaPLe on ImageNet.

Method	Encoder	K-shot	w/ ProTeCt	λ	β	Leaf Acc.	HCA	MTA
CoOp	ViT B16	16		N/A	N/A	64.01	2.31	43.74
CoOp	ViT B16	8		N/A	N/A	62.20	2.62	43.30
CoOp	ViT B16	4		N/A	N/A	61.51	2.48	40.68
CoOp	ViT B16	2		N/A	N/A	58.68	1.35	42.84
CoOp	ViT B16	1		N/A	N/A	56.43	1.51	38.27
CoOp	ViT B16	16	✓	1	0.1	62.60	32.84	86.66
CoOp	ViT B16	8	✓	1	0.1	62.15	30.65	85.84
CoOp	ViT B16	4	✓	1	0.1	61.24	26.85	85.52
CoOp	ViT B16	2	✓	1	0.1	60.42	23.22	84.38
CoOp	ViT B16	1	✓	1	0.1	60.16	22.95	84.38
MaPLe	ViT B16	16		N/A	N/A	64.15	1.97	45.93
MaPLe	ViT B16	8		N/A	N/A	62.76	1.99	43.98
MaPLe	ViT B16	4		N/A	N/A	63.45	2.51	49.41
MaPLe	ViT B16	2		N/A	N/A	61.75	2.81	45.92
MaPLe	ViT B16	1		N/A	N/A	61.78	2.18	45.50
MaPLe	ViT B16	16	✓	1	0.1	62.77	27.86	86.14
MaPLe	ViT B16	8	✓	1	0.1	61.42	23.45	85.51
MaPLe	ViT B16	4	✓	1	0.1	61.89	22.92	85.17
MaPLe	ViT B16	2	✓	1	0.1	60.43	20.10	84.23
MaPLe	ViT B16	1	✓	1	0.1	59.14	17.89	83.27

Table 19. Domain generalization on ImageNetv2 dataset using CoOp and MaPLe.

Method	Encoder	K-shot	w/ ProTeCt	λ	β	Leaf Acc.	HCA	MTA
CoOp	ViT B16	16		N/A	N/A	47.82	1.39	38.58
CoOp	ViT B16	8		N/A	N/A	45.93	2.10	42.56
CoOp	ViT B16	4		N/A	N/A	44.60	1.41	36.52
CoOp	ViT B16	2		N/A	N/A	42.17	0.96	36.01
CoOp	ViT B16	1		N/A	N/A	41.38	1.11	33.61
CoOp	ViT B16	16	✓	1	0.1	46.80	20.73	82.60
CoOp	ViT B16	8	✓	1	0.1	46.91	19.71	82.11
CoOp	ViT B16	4	✓	1	0.1	46.53	17.69	82.07
CoOp	ViT B16	2	✓	1	0.1	45.40	15.49	80.82
CoOp	ViT B16	1	✓	1	0.1	44.75	13.88	80.64
MaPLe	ViT B16	16		N/A	N/A	48.97	1.58	43.37
MaPLe	ViT B16	8		N/A	N/A	47.55	1.66	45.26
MaPLe	ViT B16	4		N/A	N/A	48.20	2.45	53.31
MaPLe	ViT B16	2		N/A	N/A	46.86	1.01	42.55
MaPLe	ViT B16	1		N/A	N/A	46.79	1.70	45.26
MaPLe	ViT B16	16	✓	1	0.1	47.47	17.77	82.52
MaPLe	ViT B16	8	✓	1	0.1	46.60	15.31	82.04
MaPLe	ViT B16	4	✓	1	0.1	47.23	14.95	81.67
MaPLe	ViT B16	2	✓	1	0.1	45.95	13.32	80.87
MaPLe	ViT B16	1	✓	1	0.1	44.92	11.24	79.94

Table 20. Domain generalization on ImageNet-sketch dataset using CoOp and MaPLe.

Method	Encoder	K-shot	w/ ProTeCt	λ	β	Leaf Acc.	HCA	MTA
CoOp	ViT B16	16		N/A	N/A	50.28	2.97	52.56
CoOp	ViT B16	8		N/A	N/A	48.08	4.19	45.05
CoOp	ViT B16	4		N/A	N/A	48.43	2.97	41.20
CoOp	ViT B16	2		N/A	N/A	46.56	1.95	52.47
CoOp	ViT B16	1		N/A	N/A	45.92	1.76	47.54
CoOp	ViT B16	16	✓	1	0.1	49.08	22.45	78.21
CoOp	ViT B16	8	✓	1	0.1	49.29	24.00	79.47
CoOp	ViT B16	4	✓	1	0.1	48.39	18.11	76.95
CoOp	ViT B16	2	✓	1	0.1	48.81	20.00	78.11
CoOp	ViT B16	1	✓	1	0.1	48.95	20.52	76.95
MaPLe	ViT B16	16		N/A	N/A	50.61	2.31	54.88
MaPLe	ViT B16	8		N/A	N/A	48.41	5.31	55.97
MaPLe	ViT B16	4		N/A	N/A	50.23	4.95	57.07
MaPLe	ViT B16	2		N/A	N/A	48.49	9.80	59.90
MaPLe	ViT B16	1		N/A	N/A	47.55	3.52	55.48
MaPLe	ViT B16	16	✓	1	0.1	47.41	19.75	77.46
MaPLe	ViT B16	8	✓	1	0.1	46.15	16.49	75.88
MaPLe	ViT B16	4	✓	1	0.1	47.35	17.39	77.64
MaPLe	ViT B16	2	✓	1	0.1	49.15	16.23	77.71
MaPLe	ViT B16	1	✓	1	0.1	47.15	16.03	76.81

Table 21. Domain generalization on ImageNet-A dataset using CoOp and MaPLe.

Method	Encoder	K-shot	w/ ProTeCt	λ	β	Leaf Acc.	HCA	MTA
CoOp	ViT B16	16		N/A	N/A	75.83	18.49	64.13
CoOp	ViT B16	8		N/A	N/A	74.79	5.91	49.56
CoOp	ViT B16	4		N/A	N/A	73.99	14.85	61.40
CoOp	ViT B16	2		N/A	N/A	70.94	16.32	56.67
CoOp	ViT B16	1		N/A	N/A	69.84	11.74	55.31
CoOp	ViT B16	16	✓	1	0.1	74.94	31.18	75.59
CoOp	ViT B16	8	✓	1	0.1	75.51	37.96	81.11
CoOp	ViT B16	4	✓	1	0.1	74.23	29.69	75.54
CoOp	ViT B16	2	✓	1	0.1	74.86	28.67	78.17
CoOp	ViT B16	1	✓	1	0.1	74.26	27.46	76.48
MaPLe	ViT B16	16		N/A	N/A	76.61	20.67	63.06
MaPLe	ViT B16	8		N/A	N/A	76.48	18.92	67.30
MaPLe	ViT B16	4		N/A	N/A	76.83	21.06	64.30
MaPLe	ViT B16	2		N/A	N/A	75.85	19.84	60.86
MaPLe	ViT B16	1		N/A	N/A	74.55	18.85	62.48
MaPLe	ViT B16	16	✓	1	0.1	75.70	32.58	77.99
MaPLe	ViT B16	8	✓	1	0.1	75.98	30.97	77.57
MaPLe	ViT B16	4	✓	1	0.1	76.31	29.28	78.52
MaPLe	ViT B16	2	✓	1	0.1	75.01	23.94	72.73
MaPLe	ViT B16	1	✓	1	0.1	74.60	25.20	75.72

Table 22. Domain generalization on ImageNet-R dataset using CoOp and MaPLe.

14. Additional Taxonomies

To investigate the robustness of ProTeCt across hierarchies, we consider the FGVC Aircraft [29] dataset and the RSI-CB [22] satellite dataset. These datasets have their built-in hierarchies, which beyond differing from those of SUN [42] and WordNet [11], are a technical hierarchy of fine-grained aircraft classes and satellite image classes, respectively. Table 23 and Table 24 summarize the CoOp results for these experiments, showing that ProTeCt improves performance under all metrics. This illustrates its taxonomy robustness.

K-shot	w/ ProTeCt	Acc_{leaf}	HCA	MTA (25)
16		41.88	17.82	21.11
16	✓	42.00	29.94	32.95
		(+0.12)	(+12.12)	(+11.84)
1		23.61	11.55	16.77
1	✓	27.30	16.47	24.67
		(+3.69)	(+4.92)	(+7.90)

Table 23. Comparison of CoOp with/without ProTeCt on FGVC Aircraft [29] dataset.

K-shot	w/ ProTeCt	Acc_{leaf}	HCA	MTA (25)
16		91.79	43.50	64.49
16	✓	93.21	85.21	91.44
		(+1.42)	(+41.71)	(+26.95)
1		63.93	32.29	52.17
1	✓	65.00	48.36	67.05
		(+1.07)	(+16.07)	(+14.88)

Table 24. Comparison of CoOp with/without ProTeCt on RSI-CB [22] satellite dataset.

15. Complete Ablation Results

This section complements Figure 4(a) and 4(b) in the main paper with the error bar. Table 25 and Table 26 show how the NCL strength λ and tree dropout rates β affect the Acc_{leaf} and HCA. Please refer to Section 6.3 of the main paper for more discussion.

Method	Encoder	K-shot	w/ ProTeCt	λ	β	Leaf Acc.	HCA
CoOp	ViT B16	1	✓	0	0.1	64.77 ± 1.37	32.93 ± 0.42
CoOp	ViT B16	1	✓	0.1	0.1	66.18 ± 0.51	39.41 ± 0.78
CoOp	ViT B16	1	✓	0.3	0.1	67.13 ± 0.66	40.82 ± 0.33
CoOp	ViT B16	1	✓	0.5	0.1	66.88 ± 0.21	41.01 ± 1.18
CoOp	ViT B16	1	✓	0.7	0.1	66.50 ± 0.71	40.39 ± 0.37
CoOp	ViT B16	1	✓	1	0.1	63.84 ± 1.51	40.05 ± 1.48

Table 25. Ablation of different NCL strength λ on Cifar100 using CoOp 1 shot setting.

Method	Encoder	K-shot	w/ ProTeCt	λ	β	Leaf Acc.	HCA
CoOp	ViT B16	16	✓	1	0	70.86 ± 0.59	54.39 ± 0.68
CoOp	ViT B16	16	✓	1	0.1	73.26 ± 0.66	58.01 ± 0.43
CoOp	ViT B16	16	✓	1	0.2	72.48 ± 0.57	59.32 ± 0.21
CoOp	ViT B16	16	✓	1	0.3	71.49 ± 0.36	58.82 ± 0.12
CoOp	ViT B16	16	✓	1	0.4	70.15 ± 0.75	57.93 ± 0.50
CoOp	ViT B16	16	✓	1	0.5	69.22 ± 0.32	56.66 ± 0.41
CoOp	ViT B16	16	✓	1	0.6	68.35 ± 0.78	53.75 ± 0.85
CoOp	ViT B16	16	✓	1	0.7	66.58 ± 0.45	53.27 ± 0.42
CoOp	ViT B16	16	✓	1	0.8	66.62 ± 0.38	50.74 ± 1.05
CoOp	ViT B16	16	✓	1	0.9	66.77 ± 0.38	49.76 ± 1.02

Table 26. Ablation of different tree dropout rates β on Cifar100 using CoOp 16 shot setting.

16. Visualizations

This section illustrates some misclassified examples of prior prompt tuning methods in ImageNet and its variants (i.e. ImageNetv2, ImageNet-S, ImageNet-A, ImageNet-R). Note that the hierarchy of these variants may differ from the one of ImageNet. The misclassification can occur in both coarse or fine-grained levels of the hierarchy. Note that ProTeCt can successfully classify all the illustrated examples at **every** hierarchy level in the examples shown in Figure 9-13. Figure 9 presents the correct/incorrect predictions of CoOp and its ProTeCt counterpart at multiple tree levels on ImageNet. CoOp [48] fails to generate consistent predictions at different hierarchy levels, and even predicts incorrectly at coarser hierarchy levels when the predictions at the leaf level are correct. More examples of the predictions on ImageNet variants are shown in Figure 10-13, where [GT, Prediction] shows the groundtruth and incorrect prediction by vanilla prompt tuning.

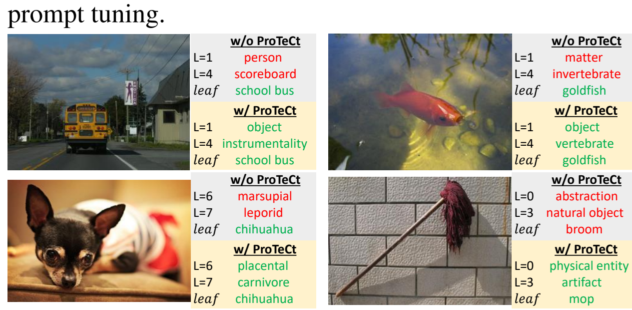

Figure 9. ImageNet visual examples at multiple hierarchy levels. Correct/incorrect model predictions (green/red) of CoOp w/ and w/o ProTeCt, respectively. L denotes the tree level.

(a) (b) (c)

Figure 10. ImageNetv2 visual examples: (a): [Taxicab, Teddy bear], (b): [Washing machine, Bath towel], (c):[Grey fox, Marsupial].

(a) (b) (c) (d)

Figure 11. ImageNet-S visual examples: (a): [Water bottle, Soap dispenser], (b): [Umbrella, Lampshade], (c):[Folding chair, Baby bed], (d):[Pembroke Welsh Corgi, Marsupial].

(a) (b) (c)

Figure 12. ImageNet-A visual examples: (a): [Chihuahua, Cottontail rabbit], (b): [Fox squirrel, Bird], (c):[American black bear, Koala].

(a) (b) (c)

Figure 13. ImageNet-R visual examples: (a): [Killer whale, Person], (b): [Wine bottle, Fruit], (c):[Cheeseburger, Ice cream].