CLIP's Visual Embedding Projector is a Few-shot Cornucopia

Supplementary Material

Algorithm 1 PyTorch-like pseudo-code for ProLIP.

```
# target: Ground truth
# lmda: regularization loss weight
# Wo : Pretrained projection matrix
# bo : Pretrained bias term (only ResNet, 0 for ViT)
# xo:  output visual embeddings (N*K, Do)
# text_weights: normalized embeddings of classnames (K,D)

# Copy initial weights for use in the regularization loss
Wo_0 = copy.deepcopy(Wo)
# Set embedding projection matrix as trainable weights
Wo.requires_grad = True
bo.requires_grad = False

v = xo @ Wo + bo
v = l2_normalize(v,dim=-1)

#compute the cosine similarity scores
logits = 100. * v @ text_weights.T

#compute regularized loss
SE_loss = nn.MSELoss(reduction='sum')
loss = CE_loss(logits, target) + lmda * SE_loss(Wo, Wo_0)
```

This document provides:

- A PyTorch-like pseudo-code for ProLIP, shown in Algorithm 1.
- Per-dataset performance of few-shot classification with few-shot validation in Sec. 7, complementing Tab. 1.
- Grid search and hyperparameter sensitivity in Sec. 8, as well as the data of Tab. 2a, Tab. 2b and Fig. 3.
- Base-to-new generalization detailed performance in Sec. 9.
- Experiments on fine-tuning the text embedding projection matrix in Sec. 10.
- Test-time adaptation details in Sec. 11.
- Additional comparison of ProLIP$_\varnothing$ to architecture-specific methods in Sec. 12.
- Details about RLA, complementarity analysis, and additional experiments and ablations in Sec. 13.
- Training of ProLIP in Sec. 14.
- Preliminaries on CLIP in Sec. 15.

7. Details on few-shot classification with few-shot validation

In addition to the average across datasets in Tab. 1, Tabs. 23-24 provide the *per-dataset performance* of all methods, with for each the average accuracy over 10 seeds (*i.e.*, support sets). ProLIP performs particularly well on DTD, UCF101, StanfordCars, FGVCAircraft and EuroSAT. For some specific settings, *e.g.*, 1-shot DTD, 16-shot StanfordCars, 8 and 16-shot FGVCAircraft, the improvements over state-of-the-art are significant. On the other hand, for datasets like OxfordPets and Food101, where the zero-shot performance is already good, ProLIP and other baselines are outperformed by prompt learning methods (*e.g.*, ProGrad). This might be due to the relatively lower number of parameters in the latter, making them less prone to overfitting in very low-shot settings; when the number of shots increases, *e.g.*, 8-16 shots, ProLIP and prompt learning perform on par.

Future research may include the zero-shot accuracy on the few-shot training set in the parametric formulation of the regularization loss weight (*i.e.*, λ). That is, the higher the zero-shot accuracy, the smaller should be the distance between the fine-tuned projection matrix and the pretrained one (*i.e.*, higher λ).

8. ProLIP hyperparameters study

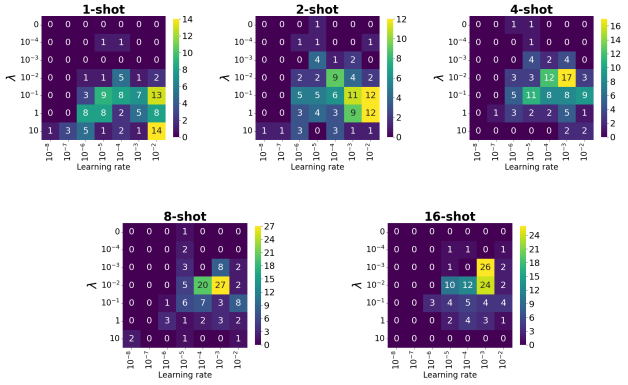

Figure 5. **Hyperparameters selected by grid search.** Learning rates and regularization loss weights λ found with grid search on the few-shot validation set. The distribution of these hyperparameters are shown for each few-shot setting ($N = 1, 2, 4, 8, 16$).

Grid search. Fig. 5 shows the distribution of hyperparameters found by grid search on the few-shot validation set (cf. Tab. 1). We draw two observations:

1. The learning rates span a wide range of values, and high values like 10^{-3} and 10^{-2} are selected several times, which would cause severe overfitting when no regularization is used (cf. Tab. 11 and Fig. 3).
2. $\lambda = 0$ is rarely selected, meaning that based on the few-shot validation set, regularized projection matrices generalize better.

Hyperparameter sensitivity. Tab. 10 complements Fig. 3, where ProLIP is trained for different fixed learning

rates, with fixed regularization loss weight λ. Looking at the values, we make the following observations:

1. For low learning rates (*i.e.*, 10^{-5}, 10^{-6}), unregularized ProLIP shows good performance for different values of N, demonstrating the effectiveness of simply fine-tuning the visual projection matrix. However, the performance drops significantly when the LR increases.
2. A higher value of λ works better for fewer training shots N, and vice versa. This effect is increasingly visible when the LR increases. Such observation is expected: with less data we need more regularization as overfitting risk is higher, and this is the base for formulating λ as a decreasing function of N (See Tab. 11, which shows the detailed numerical results of Tab. 2a).

9. Details on base-to-new generalization

Metrics details. Previous works [25, 64] calculate the *total harmonic mean* over datasets in two different ways.

To extend Tab. 5, in Tab. 12 we report for each architecture both ways of calculating the *total harmonic means*, renaming them H_{t1} and H_{t2} for disambiguation. It highlights the superiority of our method, regardless of the total harmonic mean used. We also detail the computation below.

In ProGrad [64], the total harmonic mean over the 11 datasets is computed as *the average harmonic means of individual datasets.* This writes:

$$\mathrm{H}_{t1} = \frac{1}{11}\sum_{i=1}^{11} \mathrm{HM}_i \,, \tag{9}$$

$\mathrm{HM}_i = 2 \times \frac{\mathrm{acc_{b}}_i \times \mathrm{acc_{n}}_i}{\mathrm{acc_{b}}_i + \mathrm{acc_{n}}_i}$ being the harmonic mean of dataset i. Here, $\mathrm{acc_{b}}_i$ and $\mathrm{acc_{n}}_i$ denote the accuracy on base and new classes for dataset i, respectively.

Instead in MaPLe [25], the total harmonic mean over the 11 datasets is calculated as *the harmonic mean of average base and average new classes accuracies*:

$$\mathrm{H}_{t2} = 2 \times \frac{\mathrm{acc_b} \times \mathrm{acc_n}}{\mathrm{acc_b} + \mathrm{acc_n}} \,, \tag{10}$$

where $\mathrm{acc_b} = \frac{1}{11}\sum_{i=1}^{11} \mathrm{acc_{b}}_i$ and $\mathrm{acc_n} = \frac{1}{11}\sum_{i=1}^{11} \mathrm{acc_{n}}_i$.

Per-dataset performance. We report in Tab. 17 and Tab. 18 the per-dataset accuracy for base and new classes, as well as the harmonic mean metrics.

10. Fine-tuning the text embedding projector

Can the text embedding projector work? As discussed in Sec. 3, CLIP also maps text embeddings to the shared space using a projection matrix. Here, instead of fine-tuning the visual projection matrix $\boldsymbol{W}_o$, we fine-tune its textual counterpart $\boldsymbol{W}_{ot}$, with the same strategy adopted in ProLIP$_\varnothing$. That is, the visual backbone, including $\boldsymbol{W}_o$, is frozen. Only $\boldsymbol{W}_{ot}$ is trained with:

$$L_{\text{ProLIP (text)}} = L(\boldsymbol{W}_{ot}) + \lambda \|\boldsymbol{W}_{ot} - \boldsymbol{W}_{ot}^{(0)}\|_{\mathrm{F}}^2, \tag{11}$$

where λ is set to $\frac{1}{N}$. Tab. 13 shows that this variant is also a strong baseline, though underperforming ProLIP$_\varnothing$ where the visual embedding projection is fine-tuned. This experiment gives a positive signal on the extendability of our method to other modalities.

Tab. 14 complements Tab. 13, showing the performance of this version, coined 'ProLIP$_\varnothing$ (text)', for different values of LR. We note that this baseline is strong, yet still underperforming ProLIP$_\varnothing$ and exhibiting more sensitivity to the choice of LR.

11. Details on test-time ProLIP

TPT [40] learns a single prompt for each test image using an unsupervised loss function. Given a test image $\mathbf{I}_{\text{test}}$, the image is augmented N_{views} times using a family of random augmentations $\mathcal{A}$. Predictions are made for each view, and the training consists of minimizing the entropy of the averaged probability distribution of these predictions:

$$\boldsymbol{p}^* = \mathrm{argmin}_{\boldsymbol{p}} - \sum_{i=1}^{K} \tilde{p}_{\boldsymbol{p}}(y_i|\mathbf{I}_{\text{test}}) \log \tilde{p}_{\boldsymbol{p}}(y_i|\mathbf{I}_{\text{test}}), \tag{12}$$

where

$$\tilde{p}_{\boldsymbol{p}}(y_i|\mathbf{I}_{\text{test}}) = \frac{1}{N_{\text{views}}} \sum_{i=1}^{N_{\text{views}}} p_{\boldsymbol{p}}(y_i|\mathcal{A}_i(\mathbf{I}_{\text{test}})). \tag{13}$$

In addition, *confidence selection* is used to filter out predictions with high entropy, which are considered as noisy. Self-entropy is computed for each of the N_{views}; a fixed cutoff percentile ρ keeps only predictions with lower entropy than τ. In Equation 12, $\tilde{p}_{\boldsymbol{p}}$ becomes:

$$\tilde{p}_{\boldsymbol{p}}(y|\mathbf{I}_{\text{test}}) = \frac{1}{\rho N} \sum_{i=1}^{N_{\text{views}}} 1_{\{H(p_i) \leq \tau\}} p_{\boldsymbol{p}}(y|\mathcal{A}_i(\mathbf{I}_{\text{test}})). \tag{14}$$

We apply the same framework (*i.e.*, loss function, confidence selection) with the only difference of minimizing Equation 12 over $\boldsymbol{W}_o$ instead of the prompt $\boldsymbol{p}$. For a fair comparison, we use the same number of steps for training (*i.e.*, 1 step) and the same value of the cutoff percentile $\rho = 0.1$. The learning rate is 10^{-4}. Note that, measured on ImageNet, ProLIP is ~ 13 times faster than TPT, as the latter requires backpropagation trough the whole text encoder, while in our case backpropagation is limited to the visual projection layer and is not applied on the text encoder. We also stress that since we perform only 1 step of training, the regularization loss cannot be used as the first value it takes is 0 (initially the fine-tuned projection matrix is equal to the pre-trained one).

Method		$N=1$	2	4	8	16
CLIP (0-shot)				58.89		
ProLIP (grid search)		64.21	67.43	**70.58**	**73.73**	**76.50**
ProLIP, LR=10^{-6}	$\lambda=1$	62.85	64.98	66.66	68.13	68.98
	$\lambda=10^{-1}$	63.69	66.51	68.87	71.07	72.50
	$\lambda=10^{-2}$	63.73	66.62	69.09	71.42	72.92
	$\lambda=0$	63.73	66.64	69.12	71.46	72.96
ProLIP, LR=10^{-5}	$\lambda=1$	64.28	66.59	68.30	69.67	70.49
	$\lambda=10^{-1}$	**64.60**	<u>67.49</u>	70.13	72.71	74.75
	$\lambda=10^{-2}$	63.54	66.87	70.03	73.06	75.69
	$\lambda=0$	62.84	66.35	69.69	72.89	75.65
ProLIP, LR=10^{-4}	$\lambda=1$	64.40	66.86	68.82	70.37	71.36
	$\lambda=10^{-1}$	<u>64.48</u>	**67.51**	<u>70.37</u>	73.08	75.25
	$\lambda=10^{-2}$	60.45	64.73	69.04	72.85	75.80
	$\lambda=0$	50.55	58.69	65.18	69.93	73.28
ProLIP, LR=10^{-3}	$\lambda=1$	64.39	66.82	68.78	70.42	71.45
	$\lambda=10^{-1}$	64.08	67.32	70.28	<u>73.17</u>	75.41
	$\lambda=10^{-2}$	58.42	64.43	69.16	72.94	<u>75.99</u>
	$\lambda=0$	40.05	49.60	56.35	60.33	61.79
ProLIP, LR=10^{-2}	$\lambda=1$	64.25	66.83	68.75	70.36	71.34
	$\lambda=10^{-1}$	63.04	67.03	70.05	72.75	74.73
	$\lambda=10^{-2}$	53.58	61.43	67.47	71.92	75.22
	$\lambda=0$	19.98	24.12	28.03	32.42	35.62

Table 10. **ProLIP sensitivity to hyperparameter choice.** Accuracy of ProLIP to the hyperparameters (learning rate LR and regularization weight λ) for $N \in \{1, 2, 4, 8, 16\}$-shot settings. Each number is an average over 11 datasets, 10 seeds for each.

Method		$N=1$	2	4	8	16
CLIP (0-shot)				58.89		
ProLIP$_\varnothing$, $\lambda=1/N$	LR=10^{-5}	64.28	67.07	69.68	72.57	75.20
	LR=10^{-4}	64.40	67.28	70.08	72.97	75.57
	LR=10^{-3}	64.39	67.20	70.01	73.02	75.73
	LR=10^{-2}	64.25	67.20	69.98	72.70	75.34
	Average	**64.33**	<u>67.19</u>	<u>69.94</u>	<u>72.82</u>	**75.46**
ProLIP$_\varnothing$, $\lambda=1/N^2$	LR=10^{-5}	64.28	67.32	70.22	73.10	75.68
	LR=10^{-4}	64.40	67.53	70.36	73.08	75.07
	LR=10^{-3}	64.39	67.40	70.25	73.10	75.80
	LR=10^{-2}	64.25	67.31	70.02	72.50	74.50
	Average	**64.33**	**67.39**	**70.21**	**72.95**	<u>75.26</u>
ProLIP$_\varnothing$, $\lambda=0$	LR=10^{-5}	62.84	66.35	69.69	72.89	75.65
	LR=10^{-4}	50.55	58.69	65.18	69.93	73.28
	LR=10^{-3}	40.05	49.60	56.35	60.33	61.79
	LR=10^{-2}	19.98	24.12	28.03	32.42	35.62
	Average	43.36	49.69	54.81	58.89	61.59

Table 11. **ProLIP$_\varnothing$ with a parametric λ.** Accuracy (%) of ProLIP$_\varnothing$ with fixed learning rate (LR) and λ as a function of N. For each λ value, we report performance for different LRs and averaged across LRs. Numbers are averages over 11 datasets and 10 seeds. We highlight **best** and <u>2nd best</u> for averages across LRs.

	Base	New	H_{t1}	H_{t2}
CLIP	61.72	65.91	63.64	63.75
CoOp	71.96	61.26	65.58	66.18
CoCoOp	72.23	60.77	65.35	66.01
ProGrad	73.29	65.96	69.06	69.43
ProLIP$_\varnothing$	**75.45**	**69.43**	**72.12**	**72.31**

(a) ResNet-50

	Base	New	H_{t1}	H_{t2}
CLIP	69.34	74.22	71.59	71.70
CoOp	82.69	63.22	70.83	71.66
CoCoOp	80.47	71.69	75.44	75.83
MaPLe	82.28	**75.14**	78.27	78.55
ProLIP$_\varnothing$	**83.85**	74.78	**78.85**	**79.06**

(b) ViT-B/16

Table 12. **Base-to-new.** Performance comparison of methods on ResNet-50 and ViT-B/16 architectures across 11 datasets with either H_{t1} (equation 9) or H_{t2} (equation 10). Numbers highlight the superiority of our method.

Method	$N=1$	2	4	8	16
CLIP (0-shot)			58.89		
ProLIP$_\varnothing$ (text)	64.05	66.93	69.71	72.56	75.01
ProLIP$_\varnothing$ (ours)	**64.33**	**67.19**	**69.94**	**72.82**	**75.46**

Table 13. **Comparison to fine-tuning the text embedding projection matrix.** We report the classification accuracy (%) averaged over 11 datasets, 10 seeds, and 4 learning rates LR $\in \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$ where we fine-tune the text projection matrix instead of the visual one, with the same regularization strategy. We call this variant 'ProLIP$_\varnothing$ (text)' and use $\lambda = 1/N$.

Method	LR	$N=1$	2	4	8	16
CLIP (0-shot)				58.89		
ProLIP$_\varnothing$ (text)	10^{-5}	64.25	67.10	69.91	72.82	75.34
	10^{-4}	64.13	67.14	70.01	72.80	75.20
	10^{-3}	63.99	66.74	69.52	72.41	75.00
	10^{-2}	63.81	66.72	69.39	72.21	74.51

Table 14. **Fine-tuning the text embedding projection matrix.** We report classification accuracy (%) of 'ProLIP$_\varnothing$ (text)' averaged over 11 datasets and 10 seeds, using different learning rates (LR).

Method	**Cls.**	**B2N**	**Arch. Agnostic**
CLIP-LoRA [55]	**77.74**	77.28	✗
ProLIP$_\varnothing$	76.26	**79.06**	✓

Table 15. **CLIP-LoRA vs. ProLIP$_\varnothing$.** Few-shot classification ('**Cls.**') accuracies are averages of 550 runs (11 datasets, 10 seeds, 5 few-shot settings), Base-to-new ('**B2N**') harmonic means are averages of 110 runs (11 datasets, 10 seeds, N=16-shot setting). For fair comparison, we adopt 'a photo of a {}.' as template, similarly to CLIP-LoRA.

12. Comparison to architecture-specific baselines

Comparison to low-rank adaptation (LoRA). Zanella *et al*. [55] recently showed that applying low-rank adaptation (LoRA) [22] to CLIP is a comptetive baseline to adapters and prompt learning. They apply LoRA on query, key and value matrices of the ViT, and show strong performance on the few-shot classification setting. We compare ProLIP$_\varnothing$ to CLIP-LoRA [55] for ViT-B/16 on few-shot classification and base-to-new generalization. Few-shot classification results are averaged across 5 settings (*i.e.*, N={1,2,4,8,16}), 11 datasets and 10 seeds, while base-to-new generalization results are averaged across 11 datasets and 10 seeds for $N = 16$. For a fair comparison, we use the template 'a photo of a {}' for the class names, similarly to CLIP-LoRA. The results are shown in Tab. 15. Slower than ProLIP, CLIP-LoRA works well on classification [55] but compromises base-to-new generalization (B2N) [11] and is specific to ViT.

Arch.	Method	**Cls.**	**Params**
RN50	BatchNorm [47]	71.66	**0.05M**
	ProLIP$_\varnothing$	**75.46**	2.10M
ViT-B/16	LayerNorm [60]	78.13	**0.07M**
	ProLIP$_\varnothing$	**81.00**	0.39M

Table 16. **Normalization parameters tuning vs. ProLIP$_\varnothing$.** Few-shot classification ('**Cls.**') accuracies are averages of 110 runs (11 datasets & 10 seeds). The number of samples per class is $N = 16$.

Comparison to normalization techniques. We compare here ProLIP$_\varnothing$ to methods that fine-tune only affine normalization parameters for BatchNorm [47] in CNN and LayerNorm [60] in ViT. Re-purposing these methods to few-shot settings requires full backpropagation and mini-batch training. Despite having more parameters, ProLIP$_\varnothing$ is much faster and largely outperforms both baselines. Results are shown in Tab. 16.

13. Further analysis and discussion

Complementarity to other methods. We showed in Tab. 8 that ProLIP is complementary to other methods that learn different components for few-shot adaptation. Recently, Tang et al. [45] proposed interpreting CLIP few-shot adaptation methods from a unified perspective of logit bias. That is, every method learns a bias on top of the zero-shot CLIP logits. We detail here the bias learned by each the two methods ProLIP was shown to be complementary to: TaskRes and Tip-Adapter-F, as well as the bias learned by ProLIP. TaskRes learns an element-wise adapter on top of $\boldsymbol{t}$, the text-based frozen classifier. It writes:

$$\text{Logits}_{\text{TaskRes}} = \boldsymbol{v}^\intercal(\boldsymbol{t} + \alpha \boldsymbol{r}) = \underbrace{\boldsymbol{v}^\intercal \boldsymbol{t}}_{\text{zero-shot logits}} + \alpha \boldsymbol{v}^\intercal \boldsymbol{r}. \quad (15)$$

The bias learned by TaskRes is thus a new linear probe trained on top of frozen visual features $\boldsymbol{v}$.

Tip-Adapter-F builds a cache model from the training features $\boldsymbol{F}_{\text{train}}$ and their labels $\boldsymbol{L}_{\text{train}}$. It writes:

$$\text{Logits}_{\text{Tip-Adapter-F}} = \underbrace{\boldsymbol{v}^\intercal \boldsymbol{t}}_{\text{zero-shot logits}} + \alpha \phi(\boldsymbol{v}^\intercal \boldsymbol{F}_{\text{train}}^\intercal) \boldsymbol{L}_{\text{train}}. \quad (16)$$

	Base	New	H_{t1}	H_{t2}
CLIP	61.72	65.91	63.64	63.75
CoOp	71.96	61.26	65.58	66.18
CoCoOp	72.23	60.77	65.35	66.01
ProGrad	73.29	65.96	69.06	69.43
ProLIP$_\varnothing$	**75.45**	**69.43**	**72.12**	**72.31**

(a) **Average over 11 datasets**.

Base	New	HM
64.46	59.99	62.14
65.49	57.70	61.35
66.21	58.01	61.84
66.96	60.04	63.23
67.39	**62.24**	**64.71**

(b) ImageNet

Base	New	HM
90.90	90.72	90.81
94.38	87.48	90.80
94.43	87.81	91.00
94.47	90.84	92.46
95.39	**91.15**	**93.22**

(c) Caltech101

Base	New	HM
85.86	93.85	89.68
90.31	94.03	92.13
89.07	91.00	90.02
91.78	**94.86**	**93.29**
90.86	93.13	91.98

(d) OxfordPets

	Base	New	HM
CLIP	55.55	66.35	60.47
CoOp	61.77	62.51	62.14
CoCoOp	61.68	59.98	60.82
ProGrad	63.01	64.32	63.66
ProLIP$_\varnothing$	**64.61**	**65.93**	**65.26**

(e) StanfordCars

Base	New	HM
64.10	70.92	67.34
89.33	62.77	73.73
88.07	66.26	75.62
88.19	69.38	77.66
89.42	**72.34**	**79.98**

(f) Flowers102

Base	New	HM
81.48	82.15	81.81
80.40	81.09	80.74
79.77	77.68	78.71
83.10	83.57	83.33
82.39	**84.47**	**83.42**

(g) Food101

Base	New	HM
17.89	25.13	20.90
22.53	20.40	21.41
22.73	19.40	20.93
22.77	24.24	23.48
26.67	**26.92**	**26.79**

(h) FGVCAircraft

	Base	New	HM
CLIP	66.45	70.17	68.26
CoOp	71.48	65.57	68.40
CoCoOp	71.88	67.10	69.41
ProGrad	73.71	69.78	71.69
ProLIP$_\varnothing$	**75.20**	**72.69**	**73.92**

(i) SUN397

Base	New	HM
49.31	54.35	51.71
67.71	43.92	53.28
63.54	40.78	49.68
66.90	53.06	59.18
71.00	**57.09**	**63.29**

(j) DTD

Base	New	HM
39.26	43.62	41.33
73.53	40.19	51.97
83.63	40.95	54.98
79.67	49.99	61.43
88.16	**66.69**	**75.94**

(k) EuroSAT

Base	New	HM
63.70	67.71	65.64
74.59	58.23	65.40
73.51	59.55	65.80
75.66	65.52	70.23
78.89	**71.13**	**74.81**

(l) UCF101

Table 17. **Base-to-new generalization with ResNet-50.** Per-dataset base, new, and harmonic mean accuracy of ProLIP$_\varnothing$ with $N = 4$ (except 'CLIP' which is zero-shot); cf. Tab. 5(a).

	Base	New	H_{t1}	H_{t2}
CLIP	69.34	74.22	71.59	71.70
CoOp	82.69	63.22	70.83	71.66
CoCoOp	80.47	71.69	75.44	75.83
MaPLe	82.28	**75.14**	78.27	78.55
ProLIP$_\varnothing$	**83.85**	74.78	**78.85**	**79.06**

(a) **Average over 11 datasets**.

Base	New	HM
72.43	68.14	70.22
76.47	67.88	71.92
75.98	70.43	73.10
76.66	**70.54**	**73.47**
76.56	68.63	72.38

(b) ImageNet

Base	New	HM
96.84	94.00	95.40
98.00	89.81	93.73
97.96	93.81	95.84
97.74	94.36	96.02
98.55	**94.39**	**96.43**

(c) Caltech101

Base	New	HM
91.17	97.26	94.12
93.67	95.29	94.47
95.20	97.69	96.43
95.43	**97.76**	**96.58**
94.96	96.64	95.79

(d) OxfordPets

	Base	New	HM
CLIP	63.37	74.89	68.65
CoOp	78.12	60.40	68.13
CoCoOp	70.49	73.59	72.01
MaPLe	72.94	**74.00**	73.47
ProLIP$_\varnothing$	**79.30**	70.64	**74.72**

(e) StanfordCars

Base	New	HM
72.08	77.80	74.83
97.60	59.67	74.06
94.87	71.75	81.71
95.92	72.46	82.56
96.14	**74.09**	**83.69**

(f) Flowers102

Base	New	HM
90.10	91.22	90.66
88.33	82.26	85.19
90.70	91.29	90.99
90.71	**92.05**	**91.38**
90.25	90.58	90.41

(g) Food101

Base	New	HM
27.19	36.29	31.09
40.44	22.30	28.75
33.41	23.71	27.74
37.44	**35.61**	36.50
44.01	33.94	**38.32**

(h) FGVCAircraft

	Base	New	HM
CLIP	69.36	75.35	72.23
CoOp	80.60	65.89	72.51
CoCoOp	79.74	76.86	78.27
MaPLe	80.82	**78.70**	**79.75**
ProLIP$_\varnothing$	**82.22**	77.29	79.68

(i) SUN397

Base	New	HM
53.24	59.90	56.37
79.44	41.18	54.24
77.01	56.00	64.85
80.36	59.18	68.16
81.44	**60.23**	**69.25**

(j) DTD

Base	New	HM
56.48	64.05	60.03
92.19	54.74	68.69
87.49	60.04	71.21
94.07	73.23	82.35
92.35	**77.23**	**84.12**

(k) EuroSAT

Base	New	HM
70.53	77.50	73.85
84.69	56.05	67.46
82.33	73.45	77.64
83.00	78.66	80.77
86.57	**78.91**	**82.56**

(l) UCF101

Table 18. **Base-to-new generalization with ViT-B/16.** Per-dataset base, new, and harmonic mean accuracy of ProLIP$_\varnothing$ with $N = 16$ (except 'CLIP' which is zero-shot); cf. Tab. 5(b).

$\boldsymbol{F}_{\text{train}}$ is fine-tuned, thus the bias is based on intra-modal similarity measures (*i.e.*, similarities in the visual space).

For ProLIP, we fine-tune the projection matrix $\boldsymbol{W}_o$. Omitting b_o for simplicity, the logits can be written as:

$$\text{Logits}_{\text{ProLIP}} = \boldsymbol{x}_o^\intercal \boldsymbol{W}_o^\intercal \boldsymbol{t} = \underbrace{\boldsymbol{x}_o^\intercal {\boldsymbol{W}_o^{(0)}}^\intercal \boldsymbol{t}}_{\text{zero-shot logits}} + \boldsymbol{x}_o^\intercal \boldsymbol{B}^\intercal \boldsymbol{t}. \quad (17)$$

That is, fine-tuning $\boldsymbol{W}_o$ is equivalent to learning a matrix $\boldsymbol{B}$, initialized with $\boldsymbol{0}_{D \times D_o}$. Thus, the bias learned by ProLIP is a linear combination of the pre-projected features, trained to match the fixed text-based probe $\boldsymbol{t}$. In short, each of the three methods learn a different bias, and we hypothesis that the results of Tab. 8 reflect that these biases contain orthogonal knowledge learned during few-shot adaptation.

It is worth noting that we fixed the LR to 10^{-4} for all the datasets in these experiments. While the complementarity was shown for fixed hyperparameters across all datasets, ($\alpha = \beta = 1$ for Tip-Adapter-F and $\alpha = 0.1$ for TaskRes), increasing the LR to 10^{-2} lead to overfitting since the biases of TaskRes and Tip-Adapter-F are not regularized, which highlight again the advantage of ProLIP in stability across LRs.

Revisiting CLIP-Adapter [13] with ProLIP's principles. Tab. 19 reports detailed results of CLIP-Adapter when varying its residual weight (α) and the learning rate (LR). Not only are the averaged results significantly worse than those of the Regularized Linear Adapter (RLA) variant and ProLIP$_\varnothing$, but CLIP-Adapter also exhibits high variance, especially in low-shot settings. Incorporating the ProLIP's principles, RLA consistently improves performance while being much more stable. Our ProLIP$_\varnothing$ still achieves the best results.

Number of augmented views. Following the literature [23, 58], we apply `RandomResizedCrop` and `RandomHorizontalFlip` augmentations during training. As mentionned earlier, ProLIP can be applied on pre-computed visual embeddings (before the projection layer). We ablate the number of views in which the features are saved. Fig. 6 shows that average accuracy over 11 datasets increases with more views. Interestingly, $\sim$ 10 views are sufficient to get results close to those with 300 views. In contrast, Lin et al. [31] showed that the gain saturates after more than two views for their cross-modal linear probe.

Visualization. We use UMAP to visualize EuroSAT test set feature manifolds, before and after 16-shot training (*i.e.*, zero-shot *vs.* ProLIP). The results are illustrated in Figs. 7a and 7b. We observe that the features are generally better clustered for ProLIP. Confusing categories like *Highway or Road, Permanent Crop Land and Pasture Land* exhibit remarkably better separation for our few-shot adapted model compared to zero-shot. This visualization hints that ProLIP learns better feature manifolds in the few-shot classification setting.

Method		$N=1$	2	4	8	16	Average
$\alpha = 0$	LR=10^{-5}	17.92	30.80	44.39	55.41	63.02	42.31
	LR=10^{-4}	39.17	50.45	59.91	66.78	71.74	57.61
	LR=10^{-3}	41.79	51.78	60.04	66.41	71.14	58.23
	LR=10^{-2}	39.36	45.45	49.47	52.34	53.28	47.98
	Average	34.56	44.62	53.45	60.24	64.80	51.53
$\alpha = 0.1$	LR=10^{-5}	57.65	62.21	66.46	70.30	73.12	65.95
	LR=10^{-4}	57.40	62.26	66.84	70.97	74.39	66.37
	LR=10^{-3}	44.81	53.50	61.25	67.13	71.49	59.64
	LR=10^{-2}	38.42	44.90	49.05	51.76	53.07	47.44
	Average	49.57	55.72	60.90	65.04	68.02	59.85
$\alpha = 0.3$	LR=10^{-5}	63.39	66.62	69.39	71.88	73.69	68.99
	LR=10^{-4}	60.26	64.28	68.23	71.96	75.12	67.97
	LR=10^{-3}	50.77	58.05	63.86	68.77	72.74	62.84
	LR=10^{-2}	37.55	44.37	49.22	52.07	54.69	47.58
	Average	52.99	58.33	62.68	66.17	69.06	61.85
$\alpha = 0.5$	LR=10^{-5}	63.79	66.61	68.72	70.40	71.48	68.20
	LR=10^{-4}	60.78	64.75	68.48	72.03	74.94	68.20
	LR=10^{-3}	55.47	60.73	65.62	69.90	73.54	65.05
	LR=10^{-2}	35.07	41.79	45.50	47.73	50.72	44.16
	Average	53.78	58.47	62.08	65.02	67.67	61.40
$\alpha = 0.7$	LR=10^{-5}	63.18	64.96	66.01	66.57	66.88	65.52
	LR=10^{-4}	61.32	65.19	68.69	71.75	74.02	68.19
	LR=10^{-3}	56.98	61.63	66.15	70.41	74.05	65.84
	LR=10^{-2}	36.73	41.80	43.91	46.28	47.87	43.32
	Average	54.55	58.40	61.19	63.75	65.71	60.72
$\alpha = 0.9$	LR=10^{-5}	60.74	60.99	61.04	61.12	61.13	61.00
	LR=10^{-4}	62.42	65.40	67.38	68.69	69.40	66.66
	LR=10^{-3}	58.55	63.13	67.47	71.23	74.14	66.90
	LR=10^{-2}	51.42	56.48	61.59	66.71	70.91	61.42
	Average	58.28	61.50	64.37	66.94	68.90	64.00
RLA, $\lambda = 1/N$	LR=10^{-5}	63.23	65.56	68.08	70.90	73.44	68.24
	LR=10^{-4}	63.41	65.80	68.27	70.91	73.35	68.35
	LR=10^{-3}	63.38	65.82	68.30	70.99	73.55	68.41
	LR=10^{-2}	63.35	65.78	68.29	70.99	73.57	68.40
	Average	63.34	65.74	68.24	70.95	73.48	68.35
ProLIP$_\varnothing$, $\lambda = 1/N$	LR=10^{-5}	64.28	67.07	69.68	72.57	75.20	69.76
	LR=10^{-4}	64.40	67.28	70.08	72.97	75.57	70.06
	LR=10^{-3}	64.39	67.20	70.01	73.02	75.73	70.07
	LR=10^{-2}	64.25	67.20	69.98	72.70	75.34	69.89
	Average	**64.33**	**67.19**	**69.94**	**72.82**	**75.46**	**69.95**

Table 19. **Improving CLIP-Adapter with ProLIP's principles** results in the Regularized Linear Adapter (RLA) variant. We report classification accuracy (%) averaged over 11 datasets, 10 seeds, and 4 learning rates LR$\in\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$ for CLIP-Adapter with different α values, ProLIP$_\varnothing$ and RLA with $\lambda = 1/N$.

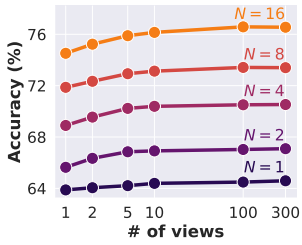

Figure 6. **Effect of augmented views.** Ablation of ProLIP using varying number of views and shots.

(a) Zero-shot CLIP (b) ProLIP

Figure 7. **Ablation and UMAP Visualization.** (a) and (b) UMAP of Zero-shot CLIP vs. ProLIP on EuroSAT, showing that some classes (*e.g.*, 'Pasture Land', 'Permanent Crop Land', 'Sea or Lake', etc.) are better clustered with our method.

temp	$N=1$	2	4	8	16
50	**64.47**	**67.37**	**70.25**	**73.13**	**75.72**
100	64.40	67.28	70.08	72.97	75.57
150	64.15	67.03	69.85	72.70	75.24

Table 20. **Effect of temperature (temp).** We report classification accuracy (%) of ProLIP$_\varnothing$ averaged over 11 datasets and 10 seeds for different temperature values. LR=10^{-4} and $\lambda = 1/N$ for all datasets.

More few-shot settings & Full data training. Training on 32 shots, ProLIP$_\varnothing$ yields 77.79% average accuracy over 11 datasets and 10 seeds, better than lower-shot results (cf. Tab. 2b). Using full data for training, ProLIP$_\varnothing$ improves to 81.03% compared to 79.97% for TaskRes. Of note, Tip-adapter-F trains $N_{\text{data}} \times D$ parameters, thus 1.3B parameters for full ImageNet, which is not feasible. This also highlights the benefit of ProLIP for which the number of trainable parameters is not a function of the dataset size.

Effect of temperature. In all the experiments of the paper, we use the pretrained temperature value temp$= 1/\tau = 100$ (cf. Eq. (2)). Here we ablate this choice and show in Tab. 20 the performance of ProLIP$_\varnothing$ for temp$= 50$ and temp$= 150$. We observe that the performance is not highly affected, and that temp$= 50$ even outperforms the pretrained value. Studying in depth the effect of this parameter is left for future research.

14. ProLIP training details

The text encoder is fully frozen during training of ProLIP. The templates are similar to previous works [23, 58] for fair comparison, and are detailed in Tab. 21 for each dataset.

For training, only the weight matrix $\boldsymbol{W}_o$ in Eq. (1) is fine-tuned. Note that for ResNets, a bias term $\boldsymbol{b}_o$ exists while for ViTs no bias is added in pretraining. We stress that fine-tuning also the bias term for ResNets does not change the results, as most of the parameters as concentrated in the weight matrix. In detail for ResNet-50, $\boldsymbol{W}_o \in \mathbb{R}^{D \times D_o}$, where $D = 1024$ and $D_o = 2048$, this makes a total of $\sim$2M parameters, while $b_o \in \mathbb{R}^D$ has only 1024 parameters. Tab. 22 shows the number of trainable parameters in ProLIP for different backbones.

Dataset	Template
Caltech101	"a photo of a {class}."
StanfordCars	"a photo of a {class}."
SUN397	"a photo of a {class}."
DTD	"{class} texture."
Eurosat	"a centered satellite photo of {class}."
FGVCAircraft	"a photo of a {class}, a type of aircraft."
Food101	"a photo of {class}, a type of food."
Flowers102	"a photo of a {class}, a type of flower."
OxfordPets	"a photo of a {class}, a type of pet."
UCF101	"a photo of a person doing {class}."
ImageNet	Ensemble of 7 templates:
ImageNet-A	{"itap of a {class}.", "a bad photo of the {class}.",
ImageNet-V2	"a origami {}.", "a photo of the large {class}.",
ImageNet-R	"a {class} in a video game.", "art of the {class}.",
ImageNet-Sketch	"a photo of the small {class}."}

Table 21. **Dataset-specific templates.** Following the literature, all but ImageNet dataset and its variants use a single template.

Backbone	$D \times D_o$	Parameters in W_o
ResNet-50	1024×2048	2.097M
ResNet-101	512×2048	1.049M
ViT-B/32	512×768	0.393M
ViT-B/16	512×768	0.393M

Table 22. **Number of trainable parameters per backbone.** It is the number of elements in the projection matrix $W_o \in \mathbb{R}^{D \times D_o}$.

15. Preliminaries

15.1. Zero-shot classification

We denote f and g the vision and text encoders of CLIP, respectively. During pretraining, CLIP learns a joint embedding space that pulls corresponding image-text representations closer together and pushes away dissimilar ones. At inference, given an image $\mathbf{I}$, one only needs the names of K candidate classes to perform *zero-shot classification*:

$$\hat{k} = \text{argmax}_k \boldsymbol{v}^\intercal \boldsymbol{t}_k, \tag{18}$$

where $\boldsymbol{v} = \dfrac{f(\mathbf{I};\theta_f)}{\|f(\mathbf{I};\theta_f)\|_2}$, $\boldsymbol{t}_k = \dfrac{g(\boldsymbol{T}_k;\theta_g)}{\|g(\boldsymbol{T}_k;\theta_g)\|_2}$; θ_f and θ_g are the frozen parameters of f and g, respectively; $\boldsymbol{T}_k$ is a text prompt describing the class k, *e.g.*, "a photo of $\{\text{class}_k\}$".

15.2. Few-shot classification

Given a set of N labeled samples from each of the K classes, research has been carried out to efficiently adapt CLIP using this set. All existing research in this direction can be gathered in three main avenues (see Fig. 1).

Prompt Tuning. It parameterizes the prompt template, *i.e.*, $\boldsymbol{T}_k = [\boldsymbol{w}]_1[\boldsymbol{w}]_2 ... [\boldsymbol{w}]_M[\text{class}_k]$, where $[\boldsymbol{w}]_1$, $[\boldsymbol{w}]_2, \ldots,$ and $[\boldsymbol{w}]_M$ are learned while keeping f and g frozen. Prompt tuning adapts CLIP "indirectly" on the classifier side, *i.e.*, the text embeddings are derived from the learned prompts.

Adapters. They learn a multi-layer perceptron (MLP) h_θ with a residual connection α on top of the frozen visual features $\boldsymbol{v}$, *i.e.*, $\boldsymbol{v} := \alpha\boldsymbol{v} + (1-\alpha)h_\theta(\boldsymbol{v})$, or on top of the frozen text features $\boldsymbol{t}$, *i.e.*, $\boldsymbol{t} := \alpha\boldsymbol{t} + (1-\alpha)h_\theta(\boldsymbol{t})$, or both.

Linear probing. It trains a linear classifier $\boldsymbol{W} \in \mathbb{R}^{K \times D}$ on top of the frozen visual features, D being the embedding space dimension. Matrix $\boldsymbol{W}$ can be initialized with text embeddings $\boldsymbol{t}_k$. Since the classifier is directly tuned, linear probing restricts CLIP to K classes after adaptation and cannot be applied in open-class setting.

15.3. CLIP architecture

CLIP adopts a transformer architecture [46] for the text encoder, but the vision encoder may be either a ResNet [17] or a Vision Transformer (ViT) [10]. We detail both architectures below and later elaborate on our unified method applicable to both architectures regardless of their intrinsic differences.

ResNet. CLIP replaces the global average pooling layer in ResNet with an attention pooling layer. The output of the multi-head attention layer is then projected to the shared latent space using a linear layer. Thus, f can be written as $f = f_2 \circ f_1$, where f_1 represents all the layers up to the attention pooling (included), and f_2 represents the linear projection head. Given an image $\mathbf{I}$:

$$\boldsymbol{x}_o = f_1(\mathbf{I}), \quad \boldsymbol{v} = f_2(\boldsymbol{x}_o) = \boldsymbol{W}_o\boldsymbol{x}_o + \boldsymbol{b}_o, \tag{19}$$

with $\boldsymbol{x}_o \in \mathbb{R}^{D_o}$ the output of the attention pooling layer, $\boldsymbol{W}_o \in \mathbb{R}^{D \times D_o}$ the projection matrix and $\boldsymbol{b}_o$ a bias term.

ViT. The transformer encoder consists of multiple residual attention blocks. Each block has two main components: a multi-head self-attention and a feed-forward neural network (MLP), with residual connections. The output of the last residual attention block is projected to the latent space using a trainable matrix. Thus, f can be written as $f = f_2 \circ f_1$, where f_1 represents all the layers up to the last residual attention block (included), and f_2 represents the projection matrix. Given an image $\mathbf{I}$:

$$\boldsymbol{x}_o = f_1(\mathbf{I}), \quad \boldsymbol{v} = f_2(\boldsymbol{x}_o) = \boldsymbol{W}_o\boldsymbol{x}_o, \tag{20}$$

where no bias term is included, unlike Eq. (19).

Similarly on the text side, the embeddings are projected into the shared latent space using a linear layer.

Dataset	Method	$N=1$	2	4	8	16
ImageNet	CLIP (0-shot)			60.35		
	CoOp [63]	61.19	61.58	62.22	62.87	63.70
	PLOT [6]	60.46	60.73	61.79	62.48	63.08
	KgCoOp [52]	60.90	61.44	62.00	62.20	62.43
	ProGrad [64]	**61.58**	**62.14**	**62.59**	63.04	63.54
	CLIP-Adapter [13]	59.82	59.94	59.97	59.98	61.31
	Tip-Adapter-F [58]	60.59	61.42	62.12	63.41	**65.06**
	Tip-Adapter-F* [58]	60.98	61.23	61.72	62.84	64.03
	Standard LP [36]	22.21	31.96	41.48	49.49	56.04
	LP++ [23]	61.18	61.56	62.55	**63.76**	64.73
	ProLIP	61.28	61.79	62.38	63.30	64.31
SUN397	CLIP (0-shot)			58.85		
	CoOp [63]	61.79	63.32	65.79	67.89	70.15
	PLOT [6]	62.53	63.87	65.85	67.83	69.90
	KgCoOp [52]	62.91	64.38	66.06	66.66	67.68
	ProGrad [64]	62.79	64.12	66.32	68.33	70.18
	CLIP-Adapter [13]	60.78	61.79	63.84	66.26	67.66
	Tip-Adapter-F [58]	61.02	62.15	63.86	67.25	70.94
	Tip-Adapter-F* [58]	62.58	63.79	65.49	67.43	69.25
	Standard LP [36]	32.56	43.77	54.49	61.83	67.03
	LP++ [23]	62.47	64.65	67.28	**69.34**	71.23
	ProLIP	**63.44**	**65.16**	**67.39**	69.31	**71.31**
DTD	CLIP (0-shot)			42.69		
	CoOp [63]	42.31	47.13	54.06	59.21	63.67
	PLOT [6]	45.82	51.32	55.67	61.38	65.29
	KgCoOp [52]	45.46	50.01	53.37	58.38	62.71
	ProGrad [64]	44.19	50.41	54.82	60.31	63.89
	CLIP-Adapter [13]	43.49	44.49	48.95	57.52	62.97
	Tip-Adapter-F [58]	46.92	48.50	57.16	62.38	65.23
	Tip-Adapter-F* [58]	47.68	52.24	56.09	61.05	65.04
	Standard LP [36]	29.63	41.19	51.72	58.78	64.56
	LP++ [23]	46.97	52.44	57.75	62.42	66.40
	ProLIP	**50.21**	**54.75**	**59.30**	**64.19**	**68.02**
Caltech101	CLIP (0-shot)			85.84		
	CoOp [63]	87.06	89.14	90.00	91.00	91.77
	PLOT [6]	**89.41**	**90.22**	90.69	91.55	92.17
	KgCoOp [52]	88.24	88.85	89.89	90.32	90.93
	ProGrad [64]	88.34	89.01	90.13	90.76	91.67
	CLIP-Adapter [13]	87.69	89.37	90.21	91.33	92.10
	Tip-Adapter-F [58]	87.35	88.17	89.49	90.54	92.10
	Tip-Adapter-F* [58]	88.68	89.36	90.40	91.62	92.63
	Standard LP [36]	68.88	78.41	84.91	88.70	91.14
	LP++ [23]	88.56	89.53	90.87	91.84	92.73
	ProLIP	89.25	89.80	**91.47**	**92.37**	**93.44**
UCF101	CLIP (0-shot)			61.80		
	CoOp [63]	62.80	65.62	68.69	72.57	76.39
	PLOT [6]	63.22	66.49	70.12	74.63	77.39
	KgCoOp [52]	64.37	64.91	68.41	69.86	71.73
	ProGrad [64]	65.13	66.57	69.80	73.01	75.76
	CLIP-Adapter [13]	64.25	66.68	69.77	73.90	77.26
	Tip-Adapter-F [58]	64.28	65.48	67.61	72.05	77.30
	Tip-Adapter-F* [58]	65.50	68.55	70.55	74.25	76.83
	Standard LP [36]	40.80	51.71	61.64	68.47	73.38
	LP++ [23]	65.41	69.20	71.68	74.86	77.46
	ProLIP	**67.88**	**70.07**	**73.51**	**77.06**	**79.79**
Flowers102	CLIP (0-shot)			65.98		
	CoOp [63]	69.00	78.47	85.34	91.68	94.47
	PLOT [6]	71.09	81.22	87.61	92.60	**95.18**
	KgCoOp [52]	68.73	69.63	76.51	80.71	84.48
	ProGrad [64]	72.16	79.55	84.56	91.73	94.10
	CLIP-Adapter [13]	66.86	69.71	77.42	87.20	91.16
	Tip-Adapter-F [58]	67.73	68.18	71.17	84.11	93.02
	Tip-Adapter-F* [58]	**78.46**	**85.14**	88.53	92.33	94.26
	Standard LP [36]	56.98	73.40	84.38	91.81	95.05
	LP++ [23]	78.21	84.69	**89.56**	92.61	94.26
	ProLIP	75.33	81.95	88.34	**92.68**	94.92

Table 23. **Comparison to state-of-the-art methods**. Average classification accuracy (%) and standard deviation over 10 tasks for 11 benchmarks. Best values are highlighted in bold.

Dataset	Method	$N = 1$	2	4	8	16
StanfordCars	CLIP (0-shot)			55.78		
	CoOp [63]	57.00	58.96	62.81	68.40	72.87
	PLOT [6]	57.47	59.89	63.49	68.75	73.86
	KgCoOp [52]	57.19	58.94	59.85	61.42	62.99
	ProGrad [64]	58.63	61.23	65.02	69.43	72.76
	CLIP-Adapter [13]	56.67	57.94	61.13	65.43	70.24
	Tip-Adapter-F [58]	57.24	58.12	59.34	64.25	71.38
	Tip-Adapter-F* [58]	57.85	60.55	64.22	68.75	74.19
	Standard LP [36]	22.94	35.48	47.49	59.34	69.11
	LP++ [23]	57.20	59.95	63.44	67.81	72.33
	ProLIP	**58.72**	**61.71**	**65.68**	**70.64**	**75.64**
FGVCAircraft	CLIP (0-shot)			17.07		
	CoOp [63]	12.50	17.59	21.27	26.85	31.20
	PLOT [6]	17.75	19.55	22.26	26.70	32.09
	KgCoOp [52]	18.61	18.93	21.16	22.80	24.10
	ProGrad [64]	18.41	20.51	23.65	26.98	30.47
	CLIP-Adapter [13]	18.56	19.18	21.00	23.76	33.37
	Tip-Adapter-F [58]	18.23	19.12	20.55	23.60	30.37
	Tip-Adapter-F* [58]	19.08	20.79	23.99	30.58	36.16
	Standard LP [36]	12.66	16.92	21.11	26.53	32.42
	LP++ [23]	19.69	21.58	24.22	27.73	31.73
	ProLIP	**19.74**	**22.68**	**27.08**	**33.20**	**39.90**
EuroSAT	CLIP (0-shot)			36.22		
	CoOp [63]	40.36	56.15	66.13	77.02	82.59
	PLOT [6]	44.22	64.19	69.37	78.84	81.76
	KgCoOp [52]	43.86	52.92	59.51	63.23	64.04
	ProGrad [64]	49.37	65.22	69.57	78.44	82.17
	CLIP-Adapter [13]	43.00	48.60	59.15	69.92	75.38
	Tip-Adapter-F [58]	47.63	57.62	69.30	75.22	78.59
	Tip-Adapter-F* [58]	49.27	65.66	70.72	74.66	78.73
	Standard LP [36]	48.29	56.81	64.99	74.56	80.29
	LP++ [23]	57.23	61.65	68.67	75.86	80.53
	ProLIP	**57.95**	**70.03**	**76.48**	**81.81**	**85.81**
OxfordPets	CLIP (0-shot)			85.75		
	CoOp [63]	86.27	86.33	85.34	87.85	88.68
	PLOT [6]	87.15	87.23	88.03	88.38	88.23
	KgCoOp [52]	87.51	87.51	88.04	88.59	89.28
	ProGrad [64]	**88.34**	**87.88**	**88.59**	**88.87**	**89.39**
	CLIP-Adapter [13]	85.46	86.37	87.21	87.95	88.33
	Tip-Adapter-F [58]	85.70	86.05	86.40	87.66	89.08
	Tip-Adapter-F* [58]	86.05	86.49	87.19	87.89	88.26
	Standard LP [36]	30.62	42.64	55.60	67.32	76.23
	LP++ [23]	84.24	85.74	86.94	87.71	88.38
	ProLIP	85.46	86.17	87.05	88.15	89.17
Food101	CLIP (0-shot)			77.35		
	CoOp [63]	75.58	77.49	77.93	78.92	79.21
	PLOT [6]	77.46	77.72	78.23	78.40	78.86
	KgCoOp [52]	77.20	**78.04**	77.97	78.39	78.73
	ProGrad [64]	**78.36**	78.01	**78.38**	**79.11**	**79.51**
	CLIP-Adapter [13]	76.93	77.22	77.64	77.97	78.45
	Tip-Adapter-F [58]	77.53	77.53	77.82	78.26	78.99
	Tip-Adapter-F* [58]	77.58	77.36	77.78	78.17	78.72
	Standard LP [36]	31.59	44.60	56.13	64.45	70.97
	LP++ [23]	76.61	77.22	77.79	78.53	78.88
	ProLIP	77.06	77.61	77.74	78.37	79.21

Table 24. **Comparison to state-of-the-art methods** (Continued). Average classification accuracy (%) and standard deviation over 10 tasks for 11 benchmarks. Best values are highlighted in bold.