CLIP-driven Coarse-to-fine Semantic Guidance for Fine-grained Open-set Semi-supervised Learning

Supplementary Material

In this supplementary material, we provide further details on our proposed method, including the architecture of the adapter we used and the training algorithm. We also present additional experimental comparisons and ablations.

1. Architecture of Adapter

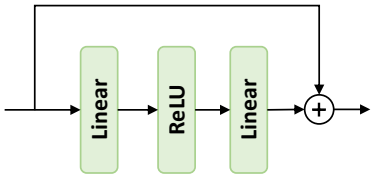

Figure 1. The architecture of adapter in the CFSG-CLIP.

Fig. 1 illustrates the architecture design of our adapter module used in the proposed CFSG-CLIP approach. By introducing two linear transformations and a non-linear activation, we enhance the model's representational capabilities. Moreover, we use a skip connection to retain the generalization ability of CLIP and apply two linear layers to better adapt to the requirements of the downstream task.

2. Optimization Procedure of CFSG-CLIP

In this section, we present the detailed optimization procedure for labeled and unlabeled samples shown in Algorithm 1.

Steps 1 to 10 illustrate the training process of coarse-guidance branch. Specifically, in step 1, we obtain the textual features of coarse prompts and fine prompts through textual encoder. In step 2, the labeled and unlabeled samples with different augmentations are fed into visual encoder with adapter to obtain the global and local visual features. Steps 3 to 7 are the computational process of the semantic filtering module for labeled and unlabeled samples with different augmentations. From step 3 to step 5, we calculate the similarity between patch-level visual features and coarse prompts' textual features to select the top-k patch-level local features. Then, we calculate the similarity weights between top-k patch-level features and global visual features in step 6. We employ the similarity weights to weight and aggregate the selected patch-level features in step 7. In step 8, we obtain the probability predictions of global and local visual features. We calculate the loss of labeled and unlabeled samples with different augmentations for coarse-guidance branch in step 10.

Table 1. Analysis of different finetuning strategies on the FGVCAircraft dataset. The results are reported based on a single run with seed 1.

	Prompt-only	Adapters-only	All
Parameters (M)	8.0	9.2	9.3
Training Time (h)	11.3	11.6	11.7

Table 2. Comparison of various backbones and pre-trained weights on the FGVCAircraft dataset. The results are reported based on a single run with seed 1.

Data	ViT-B/32	ViT-B/16	ViT-L/14
Maple	38.69	53.09	63.59
Ours	41.21	61.07	66.95

Building upon the training of coarse-guidance branch, we apply the visual-semantic injection strategy in step 11 to embed the category-related visual-semantic cues derived from the coarse-guidance branch into the visual encoder of fine-guidance branch, enabling the visual encoder to focus on more fine-grained clues. We then repeat steps 3 to 9 to obtain the global and local probability predictions of the fine-guidance branch, which are used to calculate the fine-guidance branch loss.

Finally, we optimize the losses of the two branches separately during model training.

3. More Ablations

3.1. Different Fine-tuning Strategies

To assess the efficiency of our proposed fine-tuning strategies in fine-grained OSSL task, we compare the number of parameters and training time across various fine-tuning strategies. We can see that the computational cost of using prompt and adapter fine-tuning simultaneously is negligible as shown in the Table 1.

3.2. Various Backbones

The various backbone experiments on the Aircraft dataset with 5 labeled samples are shown in the Table 2. As the model size increases, the performance of the model improves significantly. Additionally, to investigate the effect of patch size, we experiment with patch sizes of 32×32 and 16×16 on the FGVCAircraft dataset. The size of 16×16 captures finer details and boosts performance by 19.86%.

Algorithm 1 Optimization of CFSG-CLIP in labeled and unlabeled samples

Input: $\{(x^l, y^l)\}$ and $\{(x^u)\}$: Labeled and unlabeled samples. $\alpha(\cdot)$ and $\beta(\cdot)$: Weak and strong augmentation. p_m^c and p_m^f: Coarse prompts and fine prompts of the m-th class name. $\mathcal{V}$ and $\mathcal{V}_\mathcal{D}$: Visual encoder and visual encoder embedded with visual semantic cues. $\mathcal{T}$: Textual encoder. $\mathcal{A}_c$ and $\mathcal{A}_f$: Adapter. λ_c and λ_f: Weights of losses. δ: Confidence threshold.

$t_c^m = \mathcal{T}(p_c^m), t_f^m = \mathcal{T}(p_f^m)$ ▷ *Obtain the textual features of coarse prompts and fine prompts.*

$\{\tilde{z}_c^l, z_{c_1}^l, z_{c_2}^l, ..., z_{c_P}^l\} = \mathcal{A}_c(\mathcal{V}(\alpha(x^l))),$
$\{\tilde{z}_c^{u_w}, z_{c_1}^{u_w}, z_{c_2}^{u_w}, ..., z_{c_P}^{u_w}\} = \mathcal{A}_c(\mathcal{V}(\alpha(x^{u_w}))),$
$\{\tilde{z}_c^{u_s}, z_{c_1}^{u_s}, z_{c_2}^{u_s}, ..., z_{c_P}^{u_s}\} = \mathcal{A}_c(\mathcal{V}(\beta(x^{u_s})))$ ▷ *Obtain the global and local features of the labeled and unlabeled samples for the coarse-guidance branch.*

for $e \in [l, u_w, u_s]$ **do**

  $s_{c_i}^e = \text{sim}(z_{c_i}^e, t_c^m)$ ▷ *Calculate the similarity between patch-level visual features and textual features.*

  $\mathcal{K} = \{i \in P : \text{rank}(s_{c_i}^e) \leq k\}$ ▷*Select the top-k patch-level local features.*

  $w_{c_i}^e = \frac{\exp(\text{sim}(z_{c_i}^e, \tilde{z}_c^e))}{\sum \exp(\text{sim}(z_{c_i}^e, \tilde{z}_c^e))}, i \in \mathcal{K}$ ▷ *Calculate the similarity weights between top-k local features and global features.*

  $z_c^e = \sum_{i \in \mathcal{K}} w_{c_i}^e z_{c_i}^e$ ▷ *Obtain the aggregated local features for coarse-guidance branch.*

  $\tilde{p}_c^e = \frac{\exp(\text{sim}(\tilde{z}_c^e, t_c^m)/\tau)}{\sum_{m'} \exp(\text{sim}(\tilde{z}_c^e, t_c^{m'})/\tau)}, p_c^e = \frac{\exp(\text{sim}(z_c^e, t_c^m)/\tau)}{\sum_{m'} \exp(\text{sim}(z_c^e, t_c^{m'})/\tau)}$ ▷ *Global and local predictions for coarse-guidance branch.*

end for

$L_c = H(y^l, \tilde{p}_c^l) + H(y^l, p_c^l) + \lambda_c(\mathbb{1}(\max_m(\tilde{p}_c^{u_w}) > \delta)H(\tilde{p}_c^{u_w}, \tilde{p}_c^{u_s}) + \mathbb{1}(\max_m(p_c^{u_w}) > \delta)H(p_c^{u_w}, p_c^{u_s}))$ ▷ *Calculate the coarse-guidance branch loss.*

$\{\tilde{z}_f^l, z_{f_1}^l, z_{f_2}^l, ..., z_{f_P}^l\} = \mathcal{A}_f(\mathcal{V}_D(\alpha(x^l), \text{proj}(z_c^l))),$
$\{\tilde{z}_f^{u_w}, z_{f_1}^{u_w}, z_{f_2}^{u_w}, ..., z_{f_P}^{u_w}\} = \mathcal{A}_f(\mathcal{V}_D(\alpha(x^{u_w}), \text{proj}(z_c^{u_w}))),$
$\{\tilde{z}_f^{u_s}, z_{f_1}^{u_s}, z_{f_2}^{u_s}, ..., z_{f_P}^{u_s}\} = \mathcal{A}_f(\mathcal{V}_D(\beta(x^{u_s}), \text{proj}(z_c^{u_s})))$ ▷ *Inject visual semantic cues and obtain the global and local features of the labeled and unlabeled samples for the fine-guidance branch.*

Repeat steps 3-9 ▷ *Global and local predictions for fine-guidance branch.*

$L_f = H(y^l, \tilde{p}_f^l) + H(y^l, p_f^l) + \lambda_f(\mathbb{1}(\max_m(\tilde{p}_f^{u_w}) > \delta)H(\tilde{p}_f^{u_w}, \tilde{p}_f^{u_s}) + \mathbb{1}(\max_m(p_f^{u_w}) > \delta)H(p_f^{u_w}, p_f^{u_s}))$ ▷ *Calculate the fine-guidance branch loss.*

Output: L_c and L_f to update the network parameters.

Table 3. Performance with only labeled data and a mix of labeled and unlabeled data. The results are reported based on a single run with seed 1.

Method	Dogs	Cars	CUB	Aircraft
Only labeled	80.24	80.40	80.88	49.31
Labeled + Unlabeled	85.38	84.90	90.28	61.07

3.3. Performance with Labeled Data

To investigate the effect of unlabeled data in fine-grained OSSL task, we report the results using only labeled data on the four datasets with 5 labeled samples shown in the Table 3. The large performance gap shows that the model can effectively utilize useful information in unlabeled data to improve the generalization ability of the model through semi-supervised learning.

3.4. Hyperparameter Analysis

To investigate the impact of different hyperparameters for unlabeled data on model training, we conduct experiments with different values. As shown in Table 4 the model achieves the best performance when the weight is set to 1.

Table 4. Performance with different hyperparameters on the FGVCAircraft dataset. The results are reported based on a single run with seed 1.

λ_c & λ_f	0.2	0.4	0.6	0.8	1.0
Acc	57.65	59.21	59.39	60.89	61.07

4. Additional Experiments

4.1. Comparison with Coarse OSSL Methods

As shown in Table 5, we evaluate the performance of OpenMatch [29], FixMatch [31] and IOMatch [22] on fine-grained datasets respectively. Those methods are mainly designed for coarse-grained OSSL task based on ResNet training from scratch. Therefore, their performance is significantly inferior to that of the CLIP-driven methods with few labeled samples.

4.2. Open-set OOD Detection Performance

All OSSL methods are trained on the dataset where unlabeled samples contain OOD samples. When testing, some methods [11, 23, 37, 42] focus on the restricted environ-

Table 5. Classification accuracy (%) for coarse OSSL methods on four fine-grained benchmark datasets with varying labeled set sizes under the fine-grained OSSL setting. The results are presented as the mean with standard deviation over three runs using different random seeds.

Method	Stanford Dogs		Stanford Cars		CUB-200-2011		FGVCAircraft	
	5	20	5	20	5	20	5	20
FixMatch [31]	6.49±0.12	17.53±0.81	7.49±0.19	63.80±0.61	34.20±0.80	68.90±0.22	9.70±0.20	53.27±2.12
OpenMatch [29]	9.20±1.77	21.41±0.36	10.72±0.65	57.82±8.47	33.50±3.65	66.20±0.92	12.46±1.21	52.13±4.35
IOMatch [22]	8.43±0.07	21.87±0.44	9.69±0.54	57.23±6.92	39.23±0.40	72.50±0.24	10.40±0.75	54.55±2.13
CLIP [28]	79.25±0.00	79.25±0.00	75.97±0.00	75.97±0.00	66.00±0.00	66.00±0.00	31.37±0.00	31.37±0.00
Ours	**85.48**±0.21	**89.42**±0.16	**90.38**±0.09	**93.08**±0.08	**84.73**±0.17	**90.53**±0.34	**61.09**±0.27	**72.29**±0.10

Table 6. AUROC (%) of OOD detection on the unlabeled training set for CLIP-based methods across four fine-grained benchmark datasets with varying labeled set sizes under the fine-grained OSSL setting.

Method	Stanford Dogs		Stanford Cars		CUB-200-2011		FGVCAircraft	
	5	20	5	20	5	20	5	20
CLIP [28]	69.48±0.00	69.60±0.00	63.67±0.00	63.78±0.00	65.40±0.00	65.44±0.00	47.44±0.00	47.38±0.00
CLIP-LORA [43]	68.12±1.42	70.56±0.14	63.71±0.49	64.58±1.11	63.67±1.04	64.97±0.72	49.23±0.33	50.37±0.80
CLIP-Adapter [7]	69.93±0.15	64.50±1.01	58.85±1.28	60.89±0.94	61.07±0.59	64.04±0.12	49.60±0.11	51.45±0.49
CoOp [47]	60.13±0.35	59.56±0.96	56.23±0.95	57.20±0.50	58.86±0.26	59.16±0.13	50.85±0.25	51.48±0.11
LoCoOp [25]	60.59±0.38	60.43±0.14	59.66±0.20	60.33±0.14	59.86±0.25	61.07±0.07	50.98±0.62	51.19±0.41
PLOT [3]	67.34±1.49	65.57±1.31	58.58±1.19	60.09±0.64	63.24±2.56	65.48±0.45	50.33±0.19	52.03±0.21
MaPLe [14]	67.70±2.06	64.47±2.22	56.28±0.71	56.16±1.11	61.66±2.62	64.94±0.79	50.85±0.71	52.33±1.09
Ours	68.68±0.55	69.63±0.84	63.68±0.52	65.68±1.00	63.86±0.40	68.58±0.61	50.60±0.18	54.85±0.14

Table 7. AUROC (%) of OOD detection on the open-set test set for CLIP-based methods across four fine-grained benchmark datasets with varying labeled set sizes under the fine-grained OSSL setting.

Method	Stanford Dogs		Stanford Cars		CUB-200-2011		FGVCAircraft	
	5	20	5	20	5	20	5	20
CLIP [28]	70.21±0.00	70.21±0.00	62.25±0.00	62.25±0.00	64.49±0.00	64.49±0.00	47.48±0.00	47.48±0.00
CLIP-LORA [43]	68.40±1.57	70.77±0.16	62.86±0.29	63.77±0.97	63.52±1.21	65.10±0.81	50.26±0.38	50.25±0.48
CLIP-Adapter [7]	70.04±0.08	64.01±0.75	58.99±0.99	60.68±1.10	62.15±0.66	65.70±0.09	49.08±0.24	52.27±2.21
CoOp [47]	60.14±0.16	59.32±0.89	56.37±0.69	57.27±0.24	59.10±0.10	59.46±0.18	50.28±0.14	50.52±0.20
LoCoOp [25]	60.61±0.29	60.56±0.10	59.17±0.16	59.85±0.24	59.96±0.05	61.22±0.19	50.60±0.59	50.74±0.37
PLOT [3]	67.46±2.05	65.38±1.16	58.45±0.59	59.12±0.34	63.78±2.45	67.16±0.53	50.51±0.27	51.86±0.24
MaPLe [14]	67.64±1.93	64.25±1.89	55.78±0.87	54.64±1.11	62.98±2.49	67.30±1.01	50.70±0.55	51.69±0.73
Ours	67.35±1.56	69.29±0.84	63.01±0.72	64.29±0.69	65.78±0.69	71.84±0.66	50.16±0.04	54.29±0.17

ments where test samples are guaranteed to only contain ID classes, while some methods [22, 29] also consider the test samples containing OOD samples. In the main paper, we follow the former setting to mainly focus on the test samples containing only ID samples in fine-grained OSSL task. In addition, we report the AUROC of OOD detection with other methods on unlabeled training sets and open-set test sets as shown in Table 6 and Table 7.

5. Details of Datasets

- **CUB-200-2011** includes 11,788 bird images from 200 species, officially divided into 5,994 training images and 5,794 test images.
- **Stanford Dogs** consists of 20,580 images depicting 120 dog variants, with 12,000 images allocated for training and 8,580 images designated for testing.

- **Stanford Cars** comprises 16,185 images of cars, divided into 196 categories, with 8,144 images allocated for training and 8,041 images designated for testing.
- **FGVCAircraft** contains 10,000 images of aircraft across 100 categories, with 6,667 images used for training and 3,333 images for testing.
- **Semi-Aves** includes a subset of bird species from the Aves kingdom of iNaturalist 2018 dataset. There are 200 ID class and 800 OOD class categories. The training and validation set comprises 5,959 labeled images, 26,640 and 122,208 ID class and OOD class unlabeled images, respectively. And the test set has 8,000 test images.