Novel Class Discovery for Ultra-Fine-Grained Visual Categorization

Supplementary Material

6. More on Algorithm Procedure

In addition to the details elaborated in the main paper, we present the training procedure of our two-stage RAPL framework that includes three main designs for UFG-NCD, *i.e.*, channel-wise region alignment (CRA), proxy-guided supervised learning (PSL) and proxy-guided contrastive learning (PCL). The detailed training procedure of the pre-train phase and the discover phase is shown in Algorithms 1 and 2 respectively. Note that, we first pre-train the whole network on labeled split with the objective in Eq. (11), and then fine-tune the model on the combination of labeled and unlabeled splits with the objective in Eq. (12). In order to obtain more algorithmic details and reproduce the results reported in Tabs. 1 and 2, please refer to the source code included in the supplementary material.

Algorithm 1: Pre-train Phase of our RAPL

Input: Feature Extractor f, Projection Head (with Global Average Pooling) h, Labeled split $\mathcal{D}^l$, Old proxies $\mathcal{P}^{old}$.
Output: f, h, $\mathcal{P}^{old}$.

Randomly initialize f, h, $\mathcal{P}^{old}$;
for $n = 1$ *in* $[1, max\_epoch\_pre - train]$ **do**
  for $i = 1$ *in* $[1, max\_iteration]$ **do**
    Sample labeled mini-batches X^l from D^l;
    Extract feature maps Q^l;
    Group and assign region label for each feature matrix z in $Z \in Q^l$;
    Calculate $\mathcal{L}^{CRA}$ by Eq. (2);
    Generate feature vector V^l by $V^l = h(\text{flatten}(Q^l)$;
    Construct similarity matrix S between V^l and $\mathcal{P}^{old}$ by Eq. (3);
    Generate top-k negative mask M by Eq. (4);
    Perform mask scale softmax by Eq. (6) on the selected similarities by Eq. (5);
    Calculate $\mathcal{L}^{PC}$ by Eq. (7);
    Construct similarity matrix S^p within $\mathcal{P}^{old}$ by Eq. (8);
    Calculate $\mathcal{L}^{REG}$ by Eq. (9);
    Calculate overall optimization objective by Eq. (11);
    Update f, h and $\mathcal{P}^{old}$ by SGD [23];
  end
end

Algorithm 2: Discover Phase of our RAPL

Input: Feature Extractor f, Projection Head (with Global Average Pooling) h, Labeled and Unlabeled splits $\mathcal{D}^l, \mathcal{D}^u$, Old and New proxies $\mathcal{P}^{old}, \mathcal{P}^{new}$.
Output: f, h, $\mathcal{P}^{old}$ and $\mathcal{P}^{new}$.

Initialize $\mathcal{P}^{new}$ by k-means [18].
for $n = 1$ *in* $[1, max\_epoch\_discover]$ **do**
  for $i = 1$ *in* $[1, max\_iteration]$ **do**
    Sample labeled and unlabeled mini-batches $[X^l, X^u]$ from $D^l \cup D^u$;
    Extract feature maps $Q = [Q^l, Q^u]$;
    Group and assign region label for each feature matrix z in $Z \in Q$;
    Calculate $\mathcal{L}^{CRA}$ by Eq. (2);
    Generate feature vector $V = [V^l, V^u]$ by $V = h(\text{flatten}(Q)$;
    Construct similarity matrix S between V^l and $\mathcal{P}^{old}$ by Eq. (3);
    Generate top-k negative mask M by Eq. (4);
    Perform mask scale softmax by Eq. (6) on the selected similarities by Eq. (5);
    Calculate $\mathcal{L}^{PC}$ on by Eq. (7);
    Calculate $\mathcal{L}^{PCL}$ by Eq. (10);
    Calculate overall optimization objective by Eq. (12);
    Update f, h and $\mathcal{P}^{new}$ by SGD [23];
  end
end

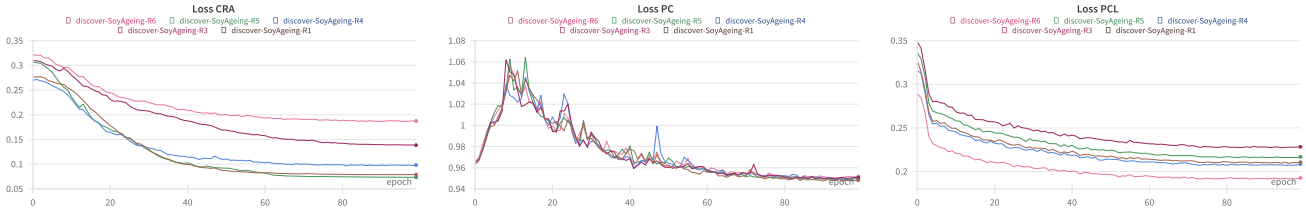
7. More on Experimental Deatils

7.1. Implementation Details on RAPL

To obtain more discriminative representation, we use a stronger feature extractor and higher resolution images than the general implementation of NCD. Specifically, we train all methods using a ResNet-50 [11] backbone with ImageNet [5] pre-trained weights on 448 $\times$ 448 resolution images. Furthermore, following [33], in the training process, the input images are cropped with a random scale within $\{0.67, 1.0\}$ and then resized to 448 $\times$ 448. After that, we perform random data augmentation to generate two views of each image, which includes horizontal flip, vertical flip, "ColorJitter", "Grayscale" and "GaussianBlur". While in the inference process, images are simply resized to 512 $\times$ 512 and center-cropped into 448 $\times$ 448. Due to the back-

(a) $\mathcal{L}^{CRA}$ (b) $\mathcal{L}^{PC}$ (c) $\mathcal{L}^{REG}$

Figure 7. We show the loss curves in the pre-train phase on five UFG-NCD datasets, where the x-axis indicates the training epochs and the y-axis is the loss cost for each loss as Eqs. (2), (7) and (9)



(a) $\mathcal{L}^{CRA}$ (b) $\mathcal{L}^{PC}$ (c) $\mathcal{L}^{PCL}$

Figure 8. We show the loss curves in the discover phase on five UFG-NCD datasets, where x-axis indicates the training epochs and y-axis is the loss cost for each loss as Eqs. (2), (7) and (10)

bone and image resolution, we used in UFG-NCD, the feature map $Z \in \mathbb{R}^{2048\times14\times14}$, and feature vector $v \in \mathbb{R}^{2048}$. Thus, we divide the feature maps into $H * W = 196$ groups to encode the features of all regions. To make the feature dimensions consistent, *i.e.*, $D = 2048$, the first 108 groups of feature maps each have 10 feature maps and the remaining groups each have 11 feature maps, denoted as:

$$\hat{Z}^i \in \begin{cases} \mathbb{R}^{10\times14\times14}, & \text{if } i < 108 \\ \mathbb{R}^{11\times14\times14}, & \text{if } i \geq 108 \text{ and } i < 196. \end{cases} \tag{13}$$

Following [25], we use a three-layer MLP projection head to facilitate representation learning of PSL and PCL in the embedding space, in which the input and output dimensions are both 2048, and the hidden dimension is set to 10240 for richer representation embedding.

Moreover, we illustrate the training loss curves on five UFG-NCD datasets in the pre-train and the discover phase of our RAPL in Figs. 7 and 8 respectively. Specifically, as shown in Fig. 7, the decline of each training loss in Eq. (11) is quite rapid in the early stage of the pre-train phase due to full supervision. Then, the decrease of these loss gradually slow down as training progress, and shows a stable convergence trend eventually. On the other hand, $\mathcal{L}^{CRA}$ and $\mathcal{L}^{PCL}$ show similar trendency in the discover phase. However, the proxy-guided classification loss $\mathcal{L}^{PC}$ shows a trend of first increasing and then decreasing due to the representation learning of the new incoming unlabeled data.

7.2. Implementation Details on Baselines

We adopt five representative methods from novel class discovery (NCD) for our ultra-fine-grained novel class discovery (UFG-NCD) task. RankStat IL [10] is widely used as a competitive baseline for NCD, while recent methods include state-of-the-art approaches are implemented based on UNO [7], *e.g.*, ComEx [28], IIC [16] and rKD [8].

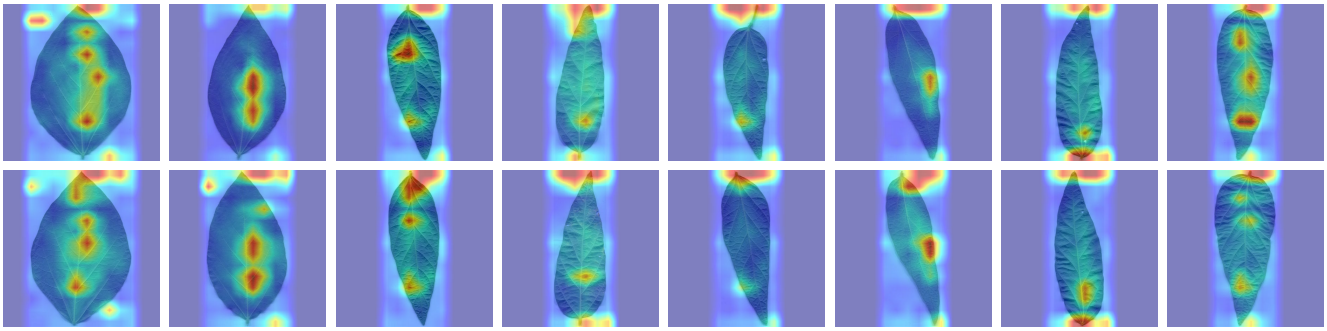
In our study, we apply these NCD methods to the UFG-NCD task, leveraging their open-source code. We set the batch size into 32 for these baselines, and we train UNO, ComEX and IIC for 200 epochs in both training phases, and train RankStat IL and rKD for 100 in the pre-train phase, and 200/500 epochs in the discover phase respectively. We fine-tuned the learning rates during each training phase to establish an optimized performance. Specifically, we set learning rates into $\{0.1, 0.01, 0.001\}$ in the pre-train phase and the discover phase respectively, and report the best performance on these combinations of learning rates. We report the optimal results of RankStat IL, UNO, ComEX and IIC when $lr_{pre-train} = 0.01$ and $lr_{discover} = 0.1$. We report the results of rKD when $lr_{pre-train} = 0.01$ and $lr_{discover} = 0.001$.

7.3. Details on Datasets.

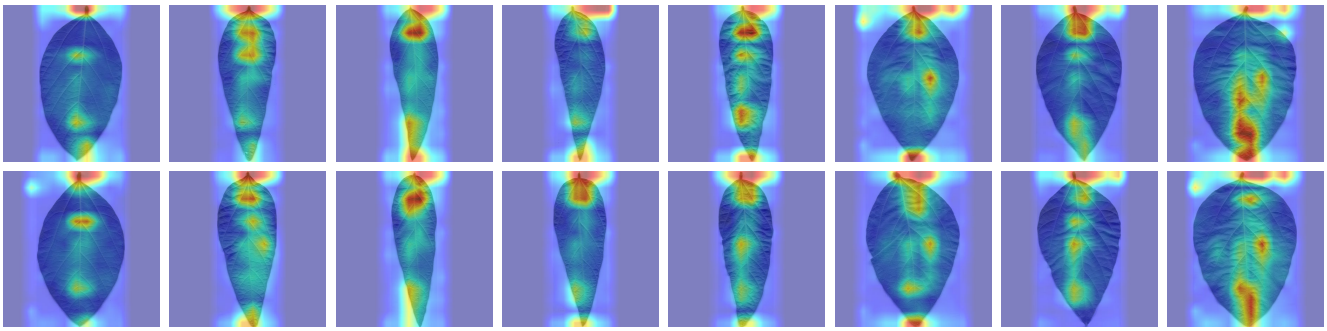
We choose five UFG-NCD datasets from [31] in our experiments, namely SoyAgeing-{R1, R3, R4, R5, R6}. For UFG-NCD, each class encompasses 5 images for both training and testing. Regarding GCD, there are 8 training im-

	Component					SoyAgeing-R1			SoyAgeing-R3			SoyAgeing-R4			SoyAgeing-R5			SoyAgeing-R6			Average		
	$\mathcal{L}^{PC}$	$\mathcal{L}^{REG}$	$\mathcal{L}^{CRA}$	$\mathcal{L}^{PCL}$	$\mathcal{L}^{VCL}$	All	Old	New	All	Old	New	All	Old	New	All	Old	New	All	Old	New	All	Old	New
(1)	✓	✓	✓		✓	56.36	74.14	38.59	54.34	74.75	33.94	56.46	74.14	38.79	55.86	72.53	39.19	48.28	62.63	33.94	54.26	71.64	36.89
(2)	✓	✓		✓		56.97	75.56	38.38	**58.99**	**78.79**	39.19	56.46	72.73	40.20	55.96	72.32	39.60	47.98	60.81	35.15	55.27	72.04	38.50
(3)	✓		✓	✓		58.48	76.97	**40.00**	57.37	76.16	38.59	**58.69**	**74.34**	**43.03**	57.78	**74.75**	40.81	**52.12**	**66.87**	**37.37**	56.89	**73.82**	39.96
(4)	✓	✓	✓	✓		**58.99**	**79.19**	38.79	**58.99**	78.59	**39.39**	57.07	71.92	42.22	**61.01**	**74.75**	**47.27**	50.20	64.65	35.76	**57.25**	**73.82**	**40.69**

Table 5. Ablation study on all UFG-NCD datasets. Each component of our method is removed in isolation, where $\mathcal{L}^{VCL}$ indicates vanilla contrastive learning without the guidance of proxies.



(a) Visualization on 8 cultivars from SoyAgeing-R1.



(b) Visualization on 8 cultivars from SoyAgeing-R3.

Figure 9. Visualization of top-15 groups of feature maps.

ages and 2 testing images per class, where unlabeled training data consists of 50% of the images from C^l, in addition to all images from C^u, *i.e.*, 1188 unlabeled training images contain 396 images from C^l and 796 images from C^u. We present the detailed statistics of the dataset in Tab. 6.

8. More Results and Analysis

8.1. More on Ablation Results

In addition to results on SoyAgeing-R1 (See in Tab. 3), we present more ablation results on all UFG-NCD datasets by individually removing each component of RAPL in Tab. 5. Moreover, we report the average accuracy in terms of "All", "Old" and "New" in Tab. 5, and the results demonstrate the effectiveness of each component of our RAPL.

8.2. Feature Maps with CRA

We provide the visualization of extracted feature maps in the test process after training by our RAPL in Fig. 9,

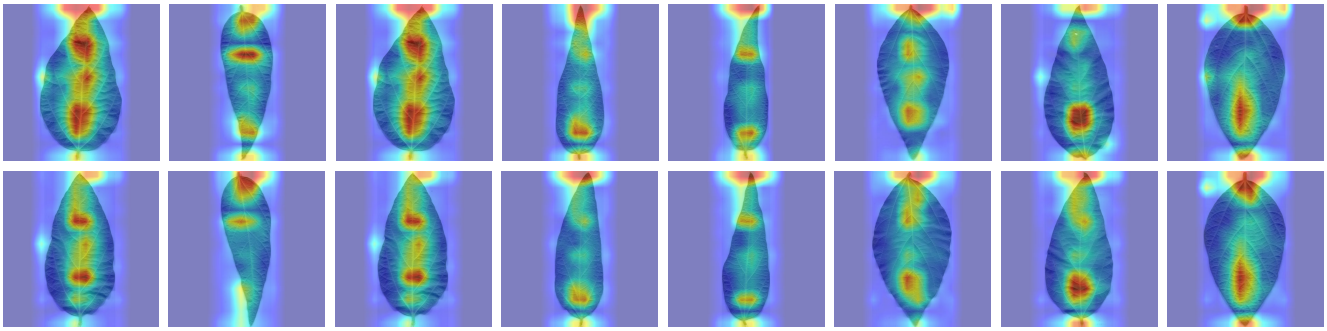
Task	C^u	C^l	Train		Test	
			N^u	N^l	N^u	N^l
UFG-NCD	99	99	495	495	495	495
GCD	99	99	1188	396	196	196

Table 6. Statistics of the dataset splits of SoyAgeing-{R1, R3, R4, R5, R6} for UFG-NCD and GCD.

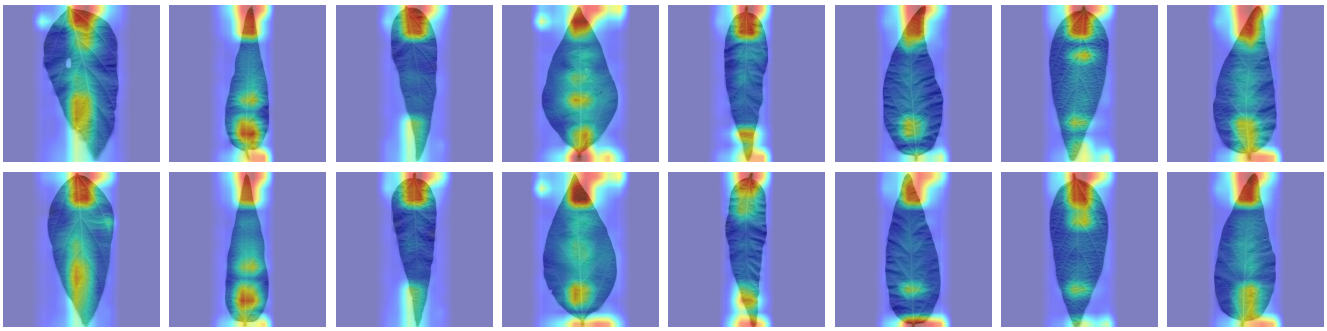
where each column contains two visualizations of images from the same class. First, we extract feature maps $Z \in \mathbb{R}^{2048\times14\times14}$ with the model trained by our RAPL, and we generate the region representation $v^r \in \mathbb{R}^{196}$ with the maximum element after performing global max pooling on each group of feature maps Z^i, where each element v_i^r is calculated by:

$$v_i^r = \text{MAX}(\text{GMP}(\hat{Z}^i)). \tag{14}$$

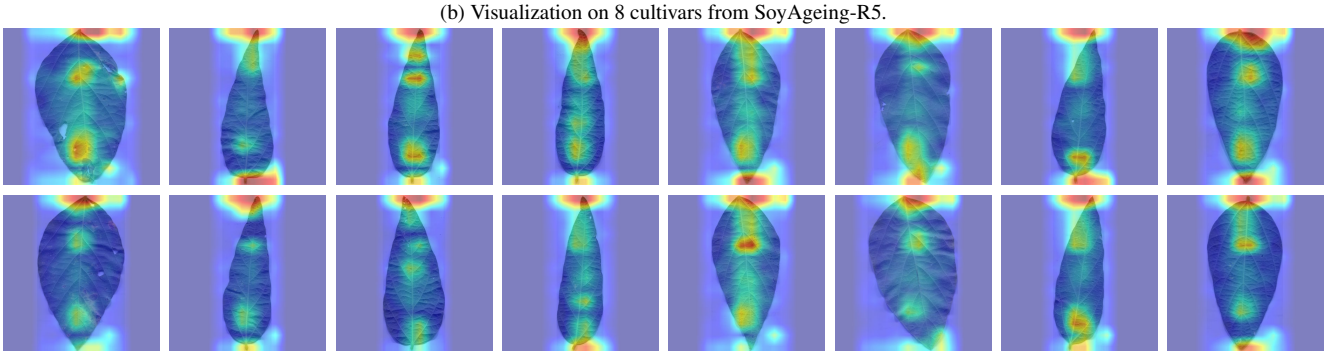
Since we perform CRA on the final feature maps, *i.e.*,



(a) Visualization on 8 cultivars from SoyAgeing-R4.



(b) Visualization on 8 cultivars from SoyAgeing-R5.



(c) Visualization on 8 cultivars from SoyAgeing-R6.

Figure 10. Visualization of top-15 groups of feature maps.

Method	Cotton80			SoyGene			CUB-200-2011			Stanford Cars		
	All	Old	New	All	Old	New	All	Old	New	All	Old	New
UNO [7]	28.25	17.50	39.00	28.68	44.21	13.15	43.79	68.30	19.27	51.81	81.17	22.45
IIC [16]	32.19	30.00	34.38	30.16	51.60	8.91	41.57	65.05	18.09	52.87	84.35	21.38
RAPL	**42.92**	**46.67**	**39.17**	**40.93**	**64.65**	**17.41**	**56.42**	**81.11**	**32.29**	**64.41**	**85.61**	**43.40**

Table 7. Compared result of Cotton80, SoyGene and CUB.

$z \in \mathbb{R}^{14\times14}$, then each dimension of the feature map z represents a 32×32 sized region of the input image. However, as shown in Fig. 9, UFG images have white edges on both sides, which will lead to exactly the same features in these regions. Thus we set the region representation of these blank regions to 0, and the improved representation is denoted as $\overline{v_i^r}$ Then, we visualize feature maps $\overline{Z}$ from the top 15 significant groups of feature maps excluding those blank regions, *i.e.*

$$\overline{Z} = \{z^j \in \hat{Z}^i\} \quad \text{s.t.}, i \in \{i \mid \overline{v_i^r} \in \text{top-k}(\overline{v_i^r})\}, \tag{15}$$

where $k = 15$. As shown in Figs. 9 and 10, we can observe that the petiole and leaf tip are highlighted in most images. This is because these two parts contain rich tex-

Method	R1	R3	R4	R5	R6	AVG
MaskCOV [30]	79.80	74.65	**79.60**	**78.28**	**66.97**	75.86
SPARE [32]	78.28	**79.90**	78.69	77.27	64.44	75.72
RAPL	**80.10**	79.70	78.18	76.16	66.57	**76.14**

Table 8. Supervised classification for SoyAgeing series.

Splits		SoyAgeing-R1		
C^l	C^u	All	Old	New
33	165	40.61	61.21	36.48
66	132	50.71	66.36	42.88
99	99	58.99	79.19	38.79
132	66	59.90	72.27	35.15
165	33	64.34	69.82	36.97

Table 9. Illustrating the trend of performance on SoyAgeing-R1, when varying the numbers of labeled and unlabeled classes.

ture information on leaf veins and shapes. Additionally, many highlighted areas are distributed around the main leaf veins, which indicates that our model automatically focuses on vein parts that are rich in texture information. Note that, with the constraints of our RAPL, the dominant features of instances that from the same class are distributed in similar regions of the images.

Overall, the success of CRA can be summarized as follows: firstly, CRA aligns the feature channels in the same group into the assigned region. Then, RAPL compares different samples and proxies channel-wise, where the discriminative region features are preserved by the weights of aligned feature channels. Since CRA learns different weights of channels in the group within the assigned region for different classes, thanks to the consistent region-channel alignment, these channels focus on the same region for both labeled and unlabeled classes. Thus, the knowledge learned from labeled classes can be easily shared and transferred to discover novel classes of unlabeled samples.

8.3. More on Other Datasets

The results in the Tab. 7 show that our RAPL consistently outperforms NCD methods by a large margin on other Ultra-FGVC (Cotton80 and SoyGene) and FGVC (CUB and S-Cars) datasets. We follow the same details to implement RAPL, UNO and IIC. Note that, for a fair comparison with RAPL on all the datasets, we reproduce UNO and IIC with ResNet-50 rather than ViT-B/16.

8.4. More on Supervised Learning

As the Ultra-FGVC methods without the specific design for discovering new categories are not compatible with NCD, for a fair comparison, we adapt our RAPL into the conventional Ultra-FGVC image classification setting instead of UFG-NCD. In this case, all the samples are completely labeled, and we use them to train RAPL via the pre-train phase only. The results in the Tab. 8 show that our RAPL is able to achieve competitive performance with state-of-the-art Ultra-FGVC methods as well.

8.5. More on Different Split Protocols

To explore the effects of RAPL under strict annotation limitation, we train the network on varying splits of SoyAgeing-R1. The results in Tab. 9 indicate that RAPL has strong robustness on unlabeled data despite the percentage of labeled classes.