A Hidden Stumbling Block in Generalized Category Discovery: Distracted Attention

Supplementary Material

Contents

A. SimGCD

In this work, our primary experiment is based on SimGCD, a representative parametric GCD method that comprises two key components: (1) representation learning and (2) classifier learning.

1)Representation Learning employs supervised contrastive learning on labeled samples, and self-supervised contrastive learning on all samples. Specifically, given two augmented views $\boldsymbol{x}_i$ and $\boldsymbol{x}'_i$ of the same image in a batch B. The unsupervised contrastive loss is written as:

$$\mathcal{L}^u_{\text{rep}} = \frac{1}{|B|}\sum_{i\in B} -\log \frac{\exp\left(\boldsymbol{z}_i^\top \boldsymbol{z}'_i/\tau_u\right)}{\sum_i^{i\neq n}\exp\left(\boldsymbol{z}_i^\top \boldsymbol{z}'_n/\tau_u\right)}, \tag{1}$$

where $\boldsymbol{z} = g(f(\boldsymbol{x}))$ and is ℓ_2-normalized, g is a MLP projection head, f is the feature backbone, τ_u is a temperature value.

The supervised contrastive loss is employed to enhance feature representation by leveraging labeled data to pull samples from the same class closer in the feature space while pushing apart samples from different classes, formally written as:

$$\mathcal{L}^s_{\text{rep}} = \frac{1}{|B^l|}\sum_{i\in B^l}\frac{1}{|\mathcal{N}_i|}\sum_{q\in\mathcal{N}_i} -\log\left(\frac{\exp(\boldsymbol{z}_i^\top \boldsymbol{z}'_q/\tau_c)}{\sum_i^{i\neq n}\exp(\boldsymbol{z}_i^\top\boldsymbol{z}'_n/\tau_c)}\right), \tag{2}$$

where $\mathcal{N}_i$ represents the set of indices corresponding to images that share the same label as $\boldsymbol{x}_i$ within a batch B, and τ_c is a temperature parameter. Finally, the overall representation learning loss is:

$$\mathcal{L}_{\text{rep}} = (1-\lambda_{sim})\mathcal{L}^u_{\text{rep}} + \lambda\mathcal{L}^s_{\text{rep}} \tag{3}$$

2) Classifier Learning aims to train a classifier that assigns labels to unlabeled data. Within the SimGCD framework, this objective is achieved through a parametric classifier refined via a self-distillation strategy, where the number of categories, denoted as $|\mathcal{Y}_u|$, is predetermined. Letting $K = |\mathcal{Y}_u|$, SimGCD initializes a set of parametric prototypes for each category, represented as $\mathcal{C} = \{\boldsymbol{c}_1, \boldsymbol{c}_2, \boldsymbol{c}_3, \ldots, \boldsymbol{c}_K\}$. Given a backbone network $f(\cdot)$, a soft label is obtained by applying softmax classification over these parametric prototypes:

$$p_i^k = \frac{\exp\left(\frac{1}{\tau_s}\left(\boldsymbol{h}_i/\|\boldsymbol{h}_i\|_2\right)^\top\left(\boldsymbol{c}_k/\left\|\boldsymbol{c}_k\right\|_2\right)\right)}{\sum_j \exp\left(\frac{1}{\tau_s}\left(\boldsymbol{h}_i/\|\boldsymbol{h}_i\|_2\right)^\top\left(\boldsymbol{c}_j/\left\|\boldsymbol{c}_j\right\|_2\right)\right)}, \tag{4}$$

where $\boldsymbol{h}_i = f(\boldsymbol{x}_i)$ is the representation of $\boldsymbol{x}_i$ and τ_s is a temperature value. A soft label $\boldsymbol{q}'$ is similarly produced for $\boldsymbol{x}'_i$ with a sharper temperature τ_t. The classification objectives are simply cross-entropy loss $\mathcal{L}_{ce}(\boldsymbol{q}', \boldsymbol{p}) = -\sum_k \boldsymbol{q}'^{(k)}\log\boldsymbol{p}^{(k)}$ between the predictions and pseudo-labels or ground-truth labels. That is,

$$\mathcal{L}^u_{cls} = \frac{1}{|B|}\sum_{i\in B}\mathcal{L}_{ce}(\boldsymbol{q}'_i, \boldsymbol{p}_i) - \epsilon H(\bar{\boldsymbol{p}}), \tag{5}$$

$$\mathcal{L}^s_{cls} = \frac{1}{|B^l|}\sum_{i\in B^l}\mathcal{L}_{ce}(\boldsymbol{y}_i, \boldsymbol{p}_i), \tag{6}$$

where $\boldsymbol{y}_i$ denotes the one-hot label of $\boldsymbol{x}_i$. SimGCD employs a mean-entropy maximization regularizer as part of the unsupervised objective. Specifically, $\bar{\boldsymbol{p}} = \frac{1}{2|B|}\sum_{i\in B}(\boldsymbol{p}_i + \boldsymbol{p}'_i)$ represents the mean prediction of a batch, and the entropy is defined as $H(\bar{\boldsymbol{p}}) = -\sum_k \bar{\boldsymbol{p}}^{(k)}\log\bar{\boldsymbol{p}}^{(k)}$. The classification objective is:

$$\mathcal{L}_{cls} = (1-\lambda_{sim})\mathcal{L}^u_{cls} + \lambda\mathcal{L}^s_{cls}, \tag{7}$$

The overall objective of SimGCD is:

$$\mathcal{L}_{sim} = \mathcal{L}_{rep} + \mathcal{L}_{cls}. \tag{8}$$

B. Experimental Setup

B.1. The details of datasets

In this study, we validate the effectiveness of our method using three challenging fine-grained datasets from the Semantic Shift Benchmark [7]: `CUB` [9], `Stanford Cars` [4], and `FGVC-Aircraft` [6]. As illustrated

Dataset	All(classes/samples)	Old labeled	Old Unlabeled	New	λ	τ
CUB [9]	200/6k	100/1.5k	100/1.5k	100/3k	0.05	0.2
Stanford Cars [4]	196/8.1k	98/2.0k	98/2.0k	98/4.1k	0.05	0.01
FGVC-Aircraft [6]	100/6.7k	50/1.7k	50/1.7k	50/3.3k	0.05	0.01
CIFAR10 [5]	10/50.0k	5/12.5k	5/12.5k	5/25.0k	0.05	0.1
CIFAR100 [5]	100/50.0k	80/20.0k	80/12.5k	20/17.5k	0.05	0.1
ImageNet-100 [2]	100/127.2k	50/31.9k	50/31.9k	50/63.4k	0.05	0.05
Herbarium-19 [8]	683/34.2k	341/8.9k	341/8.9k	342/16.4k	0.05	1e-4

Table 1. Summary of datasets and training configurations.

CUB Stanford Cars FGVC-Aircraft CIFAR10 CIFAR100 ImageNet-100 Herbarium-19

Figure 1. Image examples from the used datasets.

in Figure 1, these datasets often contain complex background information. Following SimGCD [10], we partitioned each dataset into *Known* and *Unknown* categories, with each category representing 50% of the total number of classes. Notably, 50% of the samples in the *Known* classes are unlabeled. To further assess the robustness of our method, we applied it to three generic classification datasets (CIFAR10/100 [5] and ImageNet-100 [2]), as well as the challenging large-scale fine-grained dataset Herbarium-19 [8]. As shown in Figure 1, the background interference in these datasets is relatively minimal. We employed the same partitioning strategy for these datasets, except for CIFAR-100, where 80% of the classes were designated as *Known* categories. Detailed information of datasets can be found in Table 1.

B.2. Implementation details

Following SimGCD [10], we trained all methods with a ViT-B/16 backbone [3] pre-trained with DINO [1]. We use the output of AF with a dimension of 768 as the feature for an image and only fine-tune the last block of the backbone. We train with a batch size of 128 for 200 epochs with an initial learning rate of 0.1 decayed with a cosine schedule on each dataset. Aligning with [10], the balancing factor λ_{sim} is set to 0.35, and the temperature values τ_u, τ_c as 0.07, 1.0, respectively. For the classification objective, we set τ_s to 0.1, and τ_t is initialized to 0.07, then warmed up to 0.04 with a cosine schedule in the starting 30 epochs. For AF, the configurations of λ and τ are provided in Table 1. All experiments are done with an NVIDIA GeForce RTX 4090 GPU.

C. Extended Discussions

C.1. The impact of AF on model attention

To further investigate *Distracted Attention* in the model across various data sets, we used the self-attention scores of the final ViT block to generate patch masks on both the Stanford Cars and FGVC-Aircraft datasets. As depicted in Figure 2, while the [CLS] tokens for labeled data consistently focus on key objects, those for unlabeled data, particularly from unknown category, exhibit pronounced associations with background regions. This unintended capture of extraneous information negatively impacts the quality of feature representations and, consequently, model performance. As can be observed from the comparison between different methods, AF significantly ameliorates the model's attention, enabling it to more effectively concentrate on the critical target regions. However, it is noteworthy that the extent of improvement varies across datasets due to differences in background complexity. As shown, FGVC-Aircraft predominantly features backgrounds such as airports or skies, which introduce minimal interference compared to the more cluttered and diverse backgrounds present in the CUB and Stanford Cars. This inherent characteristic of FGVC-Aircraft explains why the performance gains achieved through AF are less pronounced, compared to CUB and Stanford Cars (**Table 1 of Section 4.2**).

C.2. Single-view TAP or Multi-view TAP?

During the training process of SimGCD+AF, each data point is augmented with two distinct views. And, TAP is applied to only one of these views. To further assess the

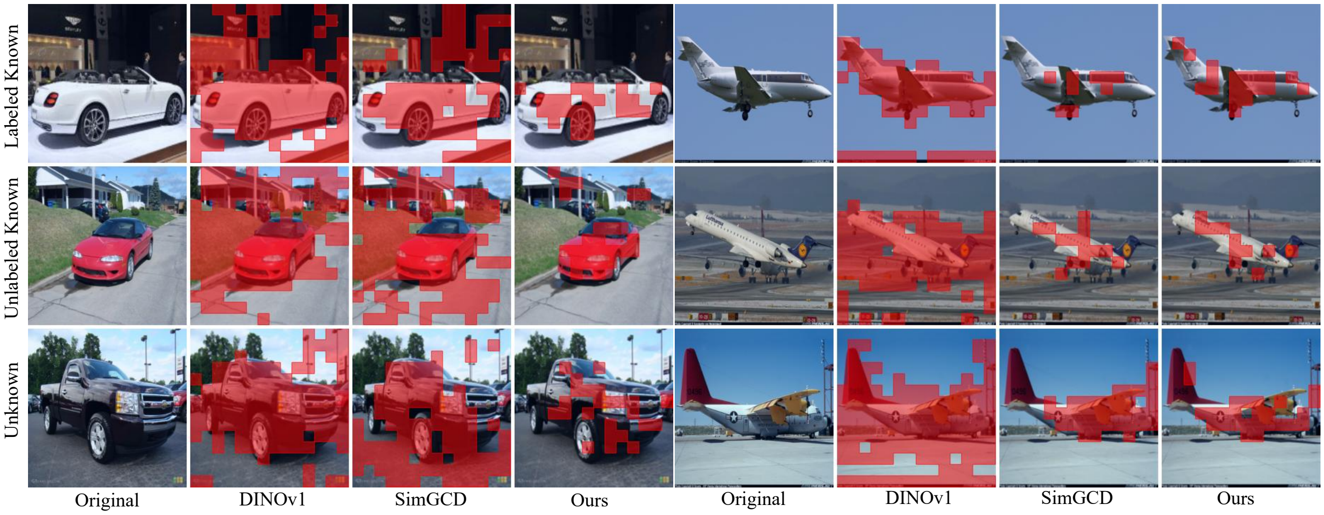

Figure 2. The masks obtained by thresholding the self-attention maps to retain same percent of the total mass cross different methods.

potential benefits of a more comprehensive approach, we experimented with multi-view TAP, where TAP is applied to both augmented views simultaneously. As shown in Table 2, while multi-view TAP does offer some performance improvements, it also leads to a noticeable degradation in comparison to single-view TAP. We believe that this can be attributed to two primary factors. First, TAP can be viewed as a form of non-regular image cropping augmentation, where single-view TAP is particularly effective in helping the model focus on key objects or regions of interest. By pruning unnecessary tokens in a single view, the model is able to maintain critical information, thus improving its ability to extract meaningful features from the image. Second, multi-view TAP essentially forces the model to train without the potential interference of background information across both views. While this might seem beneficial in theory by reducing noise, it can inadvertently reduce the model's ability to generalize.

Datasets	CUB			Stanford Cars			FGVC-Aircraft		
	All	Old	New	All	Old	New	All	Old	New
SimGCD	60.1	69.7	55.4	55.7	73.3	47.1	53.7	64.8	48.2
+AF(M-TAP)	66.8	73.1	63.6	63.2	79.9	55.1	57.4	65.7	53.3
+AF(S-TAP)	69.0	74.3	66.3	67.0	80.7	60.4	59.4	68.1	55.0

Table 2. Investigation of *Single-view Token Adaptive Pruning*. ’AF(M-TAP)’ refers to a setting where TAP is applied to both augmented views simultaneously.

C.3. [CLS] token attention vs. AF

To further demonstrate the effectiveness of AF, we utilize the attention weights between the [CLS] token and individual patches as the scores in AF, while employing the same strategy for pruning. The experimental results, as presented in Table 5, reveal that constraining the interaction between the [CLS] token and the irrelevant patches to a certain extent indeed enhances model performance. This improvement underscores the utility of refining the model's attention by mitigating the influence of task-irrelevant regions. However, it is particularly noteworthy that accuracy for *Old* category on `FGVC-Aircraft` exhibits a pronounced decline. This phenomenon suggests that the attention weights derived solely from the internal interactions between the [CLS] token and other patches are inadequate to guarantee that the model consistently attends to the correct key target regions. Such an outcome highlights the limitations of relying exclusively on intrinsic attention mechanism without additional guidance or constraints. Collectively, these findings not only underscore the generalizability and robustness of AF in diverse datasets, but also emphasize the necessity of incorporating more sophisticated strategies to ensure precise attention allocation in complex visual recognition tasks.

Datasets	CUB			Stanford Cars			FGVC-Aircraft		
	All	Old	New	All	Old	New	All	Old	New
SimGCD	60.1	69.7	55.4	55.7	73.3	47.1	53.7	64.8	48.2
+([CLS] Atten)	63.9	72.2	59.8	62.3	77.4	55.1	54.9	58.2	53.3
+AF	69.0	74.3	66.3	67.0	80.7	60.4	59.4	68.1	55.0

Table 3. Investigation of *[CLS] Token Attention*. ’AF([CLS] Atten)’ refers to using the attention weights between the [CLS] Token and patches as patch scores.

C.4. The impact of resolution

Our empirical evaluations reveal that Attention Focusing (AF) demonstrates limited performance improvements on `CIFAR10/100`, prompting a systematic investigation into its constraints. To this end, we conducted controlled experiments involving resolution scaling of input images.

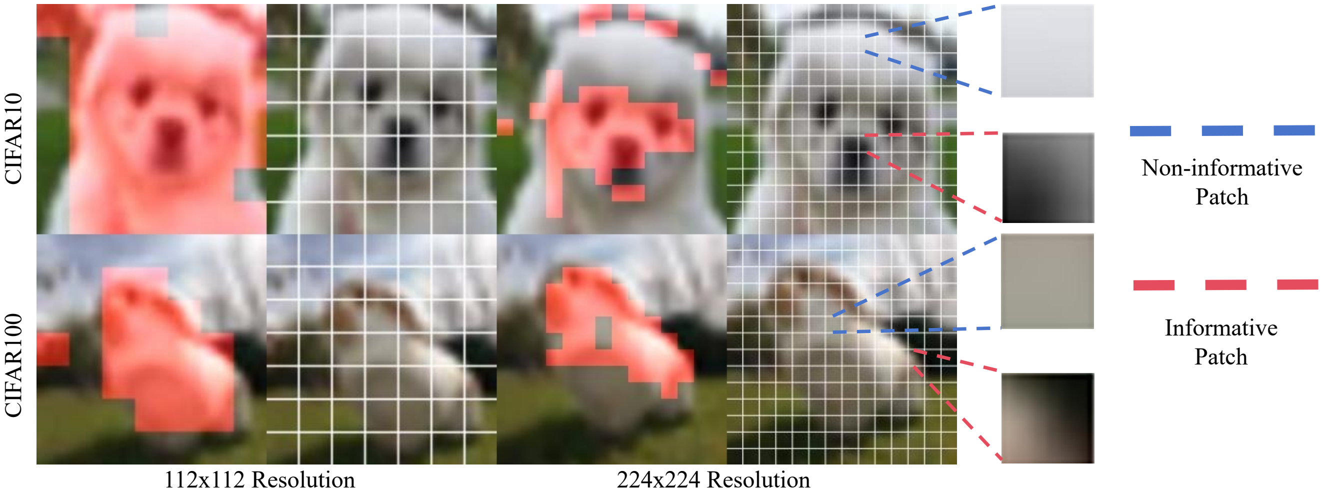

Figure 3. The partitions of input images with the same patch size under different resolutions.

Datasets	CIFAR10			CIFAR100			FLOPs
	All	Old	New	All	Old	New	
SimGCD [10]	97.1	95.1	98.1	80.1	81.2	77.8	16.87G
SimGCD+AF(224x224)	97.4	95.7	98.3	79.8	83.5	72.4	18.32G
SimGCD+AF(112x112)	97.8	95.9	98.8	82.2	85.0	76.5	4.7G

Table 4. Comparison with different resolutions.

As illustrated in Figure 3, original 32×32 pixel images were upsampled to target resolutions of 112×112 and 224×224, followed by uniform patch selection strategies under AF. Notably, a critical phenomenon emerged when maintaining consistent patch size across resolutions: Some internal patches of the target contain less information in high-resolution input images. For instance, the blue-dashed area in Figure 3 highlights a region devoid of meaningful texture, which the TIME module assigns a low significance score due to insufficient structural information. This selection bias induces cascading effects, including (1) loss of global object-related information during representation reconstruction and (2) suboptimal feature extraction due to discarding foundational constituent patches. Quantitative experiments in Table 4 corroborates these observations: 224×224 resolution fails to achieve remarkable performance improvements, even exhibiting performance degradation on CIFAR100, whereas adopting 112×112 resolution not only yields significant performance gains but also substantially reduces computational cost by over 70%, with FLOPs decreasing from 16.87G to 4.7G.

This finding establishes a critical implementation protocol for AF: Processing original low-resolution images through moderate resolution scaling achieves synergistic optimization of model performance and computational efficiency by balancing information integrality with operational cost constraints.

C.5. Class Token or Aggregation Token?

In AF, we compute the average of all remaining tokens, including the [CLS] token, to represent the image feature, which serves as the output of the backbone. The rationale behind this approach is that the remaining tokens are considered key patches that contain critical information about the object. In contrast, a common practice is to use only the [CLS] token as the image representation. As shown in Table 5, this approach results in a significant drop in performance. We believe the primary cause of this decline is that applying the self-attention mechanism solely in the final block prevents the [CLS] token from effectively aggregating information from the diverse patches throughout the image.

Datasets	CUB			Stanford Cars			FGVC-Aircraft		
	All	Old	New	All	Old	New	All	Old	New
SimGCD	60.1	69.7	55.4	55.7	73.3	47.1	53.7	64.8	48.2
+AF([CLS])	65.2	69.5	63.1	56.2	75.9	46.6	54.6	65.6	49.1
+AF	69.0	74.3	66.3	67.0	80.7	60.4	59.4	68.1	55.0

Table 5. Investigation of *Token Aggregation*. 'AF([CLS])' refers to a setting where the [CLS] token is used as the output of the backbone.

C.6. Computational efficiency of AF

To further validate the lightweight characteristics of AF module, we conducted quantitative comparisons during both training and inference phases. As illustrated in Table 6, while the parameter exhibits a more substantial increase during the training phase, the increase becomes negligible during inference —- each TIME module requires only a single vector for computation. Notably, despite the increased training parameters, the additional computational overhead remains marginal, with only a modest prolongation in training time consumption. Similarly, the testing

Method	Parameter quantity		Time consumption	
	Training	Testing	Training	Testing
SimGCD	81.82M	81.82M	18.875s	8s
SimGCD+AF	132.21M	81.83M	21.125s	10s

Table 6. Quantitative comparison of parameter quantities and time consumption for training and testing phases.

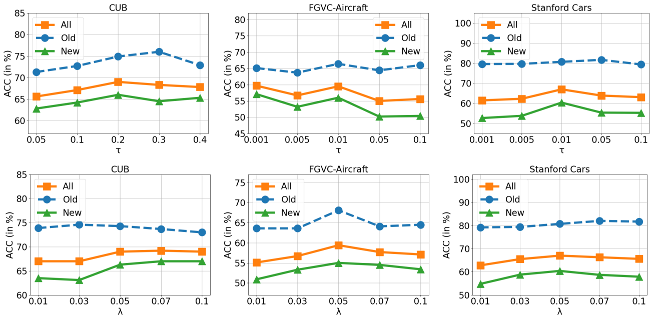

Figure 4. Investigation of the parameter λ and τ.

time demonstrates merely a slight increment. These results underscore that the AF module achieves enhanced functionality without substantially compromising computational efficiency. The minimal impact on inference phase makes it particularly suitable for deployment in resource-constrained environments.

C.7. Parameter analysis

1) Hyperparameter τ

For τ, we maintain $\lambda = 0.05$, while varying τ with a same interval. As shown in Figure 4, it is evident that τ can yield significant performance improvements within a specific range. However, the influence of τ on model performance is particularly pronounced, as it directly governs the extent of redundant information pruning. When τ is excessively large or small, it leads to over-pruning and under-pruning, respectively. Over-pruning results in the loss of global information, while under-pruning retains excessive redundancy, both of which adversely affect the model's performance. Furthermore, the inherent variability of key target regions across images, influenced by differences in object scale, spatial distribution, and background complexity, makes a fixed pruning amount suboptimal. This limitation is empirically demonstrated in **Table 7** of **Section 4.3**, where fixed pruning strategies underperform compared to adaptive approaches. Such variability highlights the need for a more flexible pruning framework that can dynamically adjust to the unique image.

2) Hyperparameter λ

For λ, we maintain τ as the pre-set value for the corresponding dataset, while varying within the set $\lambda = \{0.01, 0.03, 0.05, 0.07, 0.1\}$. As shown in Figure 4, it can be observed that the performance of AF declines when $\lambda \leq 0.03$. We attribute this phenomenon to the excessively low auxiliary loss, which diminishes the model's ability to prune redundant information. This reduction in pruning capacity leads to a lower pruning rate, resulting in the retention of excessive irrelevant features and, consequently, a degradation in representation. Conversely, when the loss is excessively high, the pruning rate of AF becomes overly aggressive, leading to incomplete image representations due to the excessive removal of critical information. These observations reveal a clear relationship between the auxiliary loss and the pruning rate: the loss function directly influences the model's pruning behavior by controlling the trade-off between retaining relevant features and eliminating redundancy. Despite these variations, AF consistently achieves significant performance improvements across different λ, demonstrating its robustness and effectiveness in enhancing image representation.

References

[1] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021. 2

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 248–255. Ieee, 2009. 2

[3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Matthias Minderer Mostafa Dehghani, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, , and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 2

[4] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 554–561, 2013. 1, 2

[5] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 2

[6] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 1, 2

[7] Andrea Vedaldi Sagar Vaze, Kai Han and Andrew Zisserman. Open-set recognition: A good closed-set classifier is all you need? In *arXiv preprint arXiv:2110.06207*, 2021. 1

[8] Kiat Chuan Tan, Yulong Liu, Barbara Ambrose, Melissa Tulig, and Serge Belongie. The herbarium challenge 2019 dataset. *arXiv preprint arXiv:1906.05372*, 2019. 2

[9] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-ucsd birds 200. 2010. 1, 2

[10] Xin Wen, Bingchen Zhao, and Xiaojuan Qi. Parametric classification for generalized category discovery: A baseline study. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16590–16600, 2023. 2, 4