
APPENDIX

A	Additional Implementation Details	2
B	Representation Visualization	3
C	Results on Additional Datasets	4
D	Experiments with the Stronger DINOv2 Representations	5
E	Category Discovery with Estimated Category Numbers	6
F	Utilization Ratio of Unlabelled Data	7
G	GCD Classifier <i>vs.</i> Debiased Classifier	8
H	Performance of the Semantic Distribution Detector	9
I	Analysis of Attention Maps	10
J	Ablation Studies on More Datasets	12
K	Impact of Hyperparameters	13
L	Stability Analysis	15
M	Prediction Error Analysis	16

A ADDITIONAL IMPLEMENTATION DETAILS

We adopt the class splits of labelled (‘Old’) and unlabelled (‘New’) categories in Vaze et al. (2022a) for generic object recognition datasets (including CIFAR-10 Krizhevsky et al. (2009) and CIFAR-100 Krizhevsky et al. (2009)) and the fine-grained Semantic Shift Benchmark Vaze et al. (2022b) (comprising CUB Wah et al. (2011), Stanford Cars Krause et al. (2013), and FGVC-Aircraft Maji et al. (2013)). Specifically, for all these datasets except CIFAR-100, 50% of all classes are selected as ‘Old’ classes (\mathcal{Y}_l), while the remaining classes are treated as ‘New’ classes ($\mathcal{Y}_u \setminus \mathcal{Y}_l$). For CIFAR-100, 80% of the classes are designated as ‘Old’ classes, while the remaining 20% as ‘New’ classes. Furthermore, for ImageNet-1K Deng et al. (2009), which is not covered in Vaze et al. (2022a), we follow Wen et al. (2023) to select the first 500 classes sorted by class ID as the labelled classes. For all the datasets, 50% of the images from the labelled classes are randomly sampled to form the labelled dataset \mathcal{D}_l , and all remaining images are regarded as the unlabelled dataset \mathcal{D}_u . Moreover, following Vaze et al. (2022a) and Wen et al. (2023), the model’s hyperparameters are chosen based on its performance on a hold-out validation set, formed by the original test splits of labelled classes in each dataset. All experiments utilize the PyTorch framework on a workstation with an Intel i7 CPU and eight Nvidia Tesla V100 GPUs. The models are trained with a batch size of 128 on a single GPU, except for the the model on CIFAR-100, ImageNet-100 and ImageNet-1K dataset, for which the training is performed with eight GPUs.

B REPRESENTATION VISUALIZATION

Here, we show the visual representation of the baseline and our method using t -SNE [Van der Maaten & Hinton \(2008\)](#). Specifically, we randomly select a set of 20 classes, including 10 from the ‘Old’ categories and 10 from the ‘New’ categories. The clearly distinguishable clusters depicted in Fig. 1 indicate that the features obtained within our framework form notably cohesive groupings compared to those of the baseline. This effectively demonstrates the optimization impacts induced by our method on the clustering feature space.

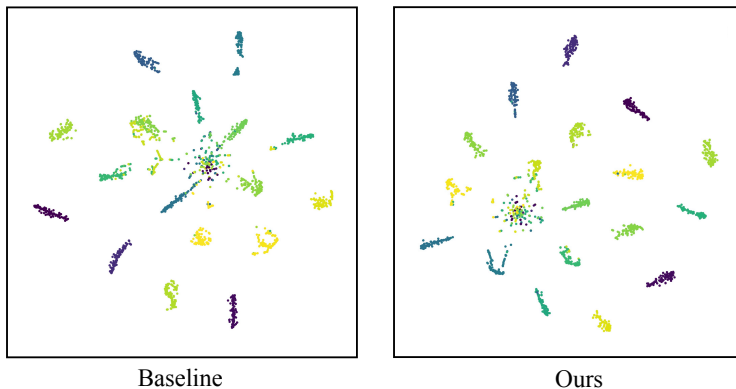


Figure 1: t -SNE visualization of 20 classes randomly sampled from the CIFAR-100 [Krizhevsky et al. \(2009\)](#) dataset.

C RESULTS ON ADDITIONAL DATASETS

To assess the performance of the proposed method comprehensively, we conducted evaluations on two more fine-grained datasets: Oxford-Pet [Parkhi et al. \(2012\)](#) and Herbarium 19 [Tan et al. \(2019\)](#). Oxford-Pet is a challenging dataset featuring various species of cats and dogs with limited data. Herbarium19, on the other hand, is a botanical research dataset encompassing diverse plant types, known for its long-tailed distribution and fine-grained categorization. The outcomes of our experiments on these datasets are detailed in Tab. 1. The results of SimGCD [Wen et al. \(2023\)](#) on Oxford-Pet are obtained through the execution of the officially released code. Our DebGCD model consistently demonstrates superior performance on both datasets.

Table 1: Comparison with state-of-the-art GCD methods on Herbarium19 [Tan et al. \(2019\)](#) and Oxford-Pet [Parkhi et al. \(2012\)](#).

Method	Oxford-Pet			Herbarium19		
	All	Old	New	All	Old	New
k -means MacQueen (1967)	77.1	70.1	80.7	13.0	12.2	13.4
RankStats+ Han et al. (2021)	-	-	-	27.9	55.8	12.8
UNO+ Fini et al. (2021)	-	-	-	28.3	53.7	14.7
ORCA Cao et al. (2022)	-	-	-	24.6	26.5	23.7
GCD Vaze et al. (2022a)	80.2	85.1	77.6	35.4	51.0	27.0
XCon Fei et al. (2022)	86.7	<u>91.5</u>	84.1	-	-	-
OpenCon Sun & Li (2022)	-	-	-	39.3	58.9	28.6
DCCL Pu et al. (2023)	88.1	88.2	88.0	-	-	-
SimGCD Wen et al. (2023)	<u>91.7</u>	83.6	<u>96.0</u>	44.0	58.0	36.4
μ GCD Vaze et al. (2023)	-	-	-	45.8	61.9	37.2
InfoSieve Rastegar et al. (2023)	90.7	95.2	88.4	40.3	59.0	30.2
DebGCD	93.0	86.4	96.5	<u>44.7</u>	<u>59.4</u>	<u>36.8</u>

D EXPERIMENTS WITH THE STRONGER DINOv2 REPRESENTATIONS

To further evaluate the robustness of the proposed method, we also evaluate the performance of DebGCD utilizing the stronger DINOv2 Oquab et al. (2023) pre-trained weights. Like in Vaze et al. (2023), in Tab. 2, we also compare our method with the k-means MacQueen (1967) baseline, and SimGCD Wen et al. (2023), μ GCD Vaze et al. (2023). Our method outperforms other methods on CUB Wah et al. (2011) and FGVC-Aircraft Maji et al. (2013) on ‘All’, ‘Old’ and ‘New’ classes consistently. On Stanford Cars Krause et al. (2013), our method outperforms other methods on ‘New’ classes, while performing the second-best on ‘All’ and ‘Old’ classes. Moreover, for the average performance of ‘All’ classes across the three datasets, DebGCD outperforms the SimGCD baseline by about 6% and μ GCD by about 3%. Additionally, we also evaluate our model on generic datasets and compare it with the SimGCD baseline in Tab. 3, demonstrating consistent improvement. The results on both fine-grained and generic datasets validate the robustness of our proposed method on the stronger DINOv2 representations, further showcasing its effectiveness.

Table 2: Comparison with state-of-the-art GCD methods on SSB leveraging DINOv2 Oquab et al. (2023) pre-trained weights.

Method	CUB			Stanford Cars			FGVC-Aircraft			Average
	All	Old	New	All	Old	New	All	Old	New	All
<i>k</i> -means MacQueen (1967)	67.6	60.6	71.1	29.4	24.5	31.8	18.9	16.9	19.9	38.6
GCD Vaze et al. (2022a)	71.9	71.2	72.3	65.7	67.8	64.7	55.4	47.9	59.2	64.3
CiPR Hao et al. (2024)	78.3	73.4	80.8	66.7	77.0	61.8	59.2	65.0	56.3	68.1
SimGCD Wen et al. (2023)	71.5	<u>78.1</u>	68.3	71.5	81.9	66.6	63.9	<u>69.9</u>	60.9	69.0
μ GCD Vaze et al. (2023)	74.0	75.9	73.1	76.1	91.0	<u>68.9</u>	<u>66.3</u>	68.7	<u>65.1</u>	<u>72.1</u>
SPTNet Wang et al. (2024)	76.3	79.5	74.6	-	-	-	-	-	-	-
DebGCD	<u>77.5</u>	80.8	<u>75.8</u>	<u>75.4</u>	<u>87.7</u>	69.5	71.9	76.0	69.8	74.9

Table 3: Comparison with state-of-the-art GCD methods on generic datasets leveraging DINOv2 Oquab et al. (2023) pre-trained weights.

Method	CIFAR-10			CIFAR-100			ImageNet-100			ImageNet-1K		
	All	Old	New	All	Old	New	All	Old	New	All	Old	New
GCD Vaze et al. (2022a)	97.8	99.0	97.1	79.6	84.5	69.9	78.5	89.5	73.0	-	-	-
CiPR Hao et al. (2024)	99.0	<u>98.7</u>	99.2	90.3	89.0	93.1	88.2	87.6	<u>88.5</u>	-	-	-
SimGCD Wen et al. (2023)	98.7	96.7	99.7	88.5	<u>89.2</u>	87.2	89.9	95.5	87.1	<u>58.0</u>	<u>66.9</u>	<u>53.2</u>
SPTNet Wang et al. (2024)	-	-	-	-	-	-	<u>90.1</u>	<u>96.1</u>	87.1	-	-	-
DebGCD	<u>98.9</u>	97.5	<u>99.6</u>	<u>90.1</u>	90.9	<u>88.6</u>	93.2	97.0	91.2	71.7	86.2	64.5

E CATEGORY DISCOVERY WITH ESTIMATED CATEGORY NUMBERS

Following the majority of the literature, we experiment mainly using the ground-truth category numbers. In this section, we report the results of DebGCD using the number of categories estimated utilizing an off-the-shelf method Vaze et al. (2022a), to showcase the performance with the ground-truth category numbers are not available. Tab. 4 reports the estimated numbers. We compare DebGCD with SimGCD Wen et al. (2023), μ GCD Vaze et al. (2023), and GCD Vaze et al. (2022a) in Tab. 5. For both CUB Wah et al. (2011) and Stanford Cars Krause et al. (2013), despite a discrepancy of approximately 15% between the ground-truth and estimated category numbers, our method exhibits a smaller decline in performance compared to GCD and SimGCD. The same trend is also observed on Imagenet-100 Deng et al. (2009). DebGCD remains the most competitive method on ‘All’ classes using the same estimated category numbers on all four datasets, which clearly demonstrates the robustness and effectiveness of our proposed method.

Table 4: Estimated class numbers in the unlabelled data using method proposed in Vaze et al. (2022a).

	CUB	Stanford Cars	CIFAR-100	ImageNet-100
Ground-truth K	200	196	100	100
Estimated K	231	230	100	109

Table 5: Results with the estimated number of categories. The estimated class numbers in Tab. 4 are adopted for all methods.

Method	CUB			Stanford Cars			CIFAR-100			ImageNet-100		
	All	Old	New	All	Old	New	All	Old	New	All	Old	New
GCD Vaze et al. (2022a)	47.1	55.1	44.8	35.0	56.0	24.8	73.0	76.2	66.5	72.7	91.8	63.8
SimGCD Wen et al. (2023)	61.5	66.4	59.1	49.1	65.1	41.3	80.1	81.2	77.8	81.7	91.2	76.8
μ GCD Vaze et al. (2023)	62.0	60.3	62.8	56.3	66.8	51.1	-	-	-	-	-	-
DebGCD	64.5	68.5	62.5	63.3	78.6	55.8	83.0	84.6	79.9	84.9	93.3	80.7

F UTILIZATION RATIO OF UNLABELLED DATA

The data utilization ratio is a notable index for pseudo-labeling methods, offering clear insights into the data efficiency. Our examination encompasses the utilization ratio of unlabelled data from both the ‘Old’ and ‘New’ classes during the training of the debiased classifier on FGVC-Aircraft [Maji et al. \(2013\)](#) and Stanford Cars [Krause et al. \(2013\)](#), as depicted in Fig. 2. Initially, the majority of data from the unknown categories remains untapped. Subsequently, after approximately 20 epochs, samples from unknown categories start to be incorporated. The utilization ratio keeps growing, reaching a ratio of around 40% at the 100th epoch. Ultimately, more than 60% of the known categories’ samples and nearly half of the unknown categories’ samples are utilized.

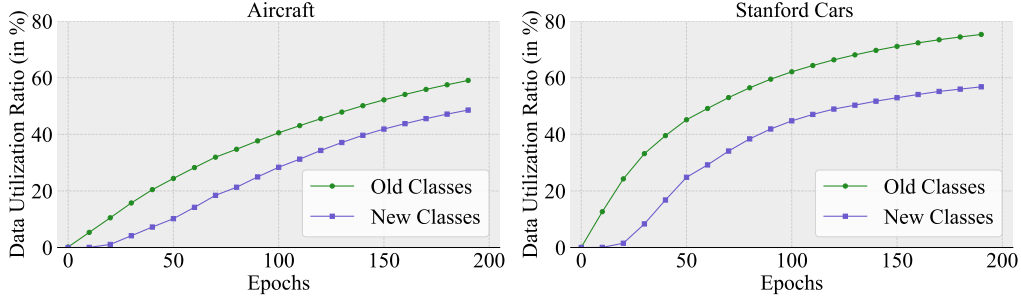


Figure 2: Unlabelled data utilization ratios for ‘Old’ and ‘New’ classes during training on FGVC-Aircraft [Maji et al. \(2013\)](#) (left) and Stanford Cars [Krause et al. \(2013\)](#) (right) datasets.

G GCD CLASSIFIER vs. DEBIASED CLASSIFIER

We compare the performance between the two classifiers, the GCD Classifier and the debiased classifier, in our framework. We report the ACC results across different epochs in Fig. 3 when training on Stanford Cars [Krause et al. \(2013\)](#), including unlabelled data from both training and the validation splits of the original dataset. Initially, the debiased classifier exhibits bias towards the ‘Old’ classes, given that the training data primarily comprises labelled data from known categories. However, as predicted scores of the unlabelled samples, particularly those from the unknown categories, progressively surpass the debiasing threshold, the performance on the unknown categories gradually improves and eventually matches with the labelled categories. Ultimately, upon convergence of the model, the performance on both known and unknown categories converges to that of the GCD classifier.

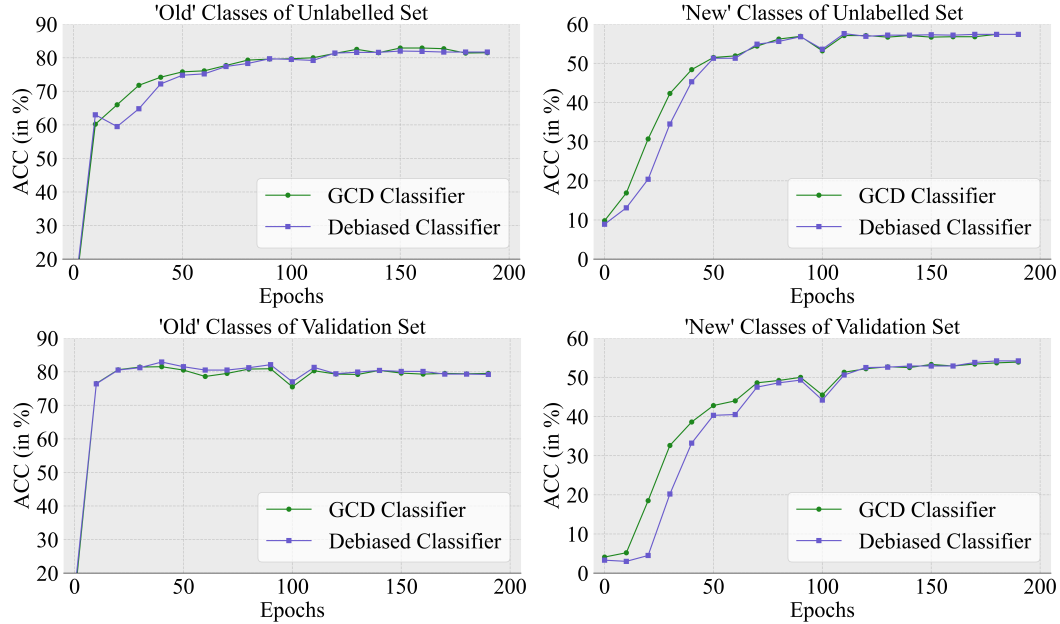


Figure 3: ACC evolution on both the ‘Old’ and ‘New’ classes of GCD Classifier and debiased classifier during training on Stanford Cars dataset [Krause et al. \(2013\)](#). The top two figures depict ACC on the unlabelled training set, while the bottom two illustrate ACC on the validation set.

H PERFORMANCE OF THE SEMANTIC DISTRIBUTION DETECTOR

We evaluate the OOD detection performance of our semantic distribution detector in DebGCD, using the threshold-free Area Under the Receiver-Operator curve (AUROC) as the evaluation metric, which is widely used in the OOD detection literature. A comparison of the OOD performance between training the entire framework and training solely the distribution detector is presented in Tab. 6. A significant improvement in OOD performance is obtained by training jointly the GCD classifier and debiased classifier. This aligns with the results presented in Tab. 4 of the main paper, which demonstrate the mutual benefits among the three branches (tasks) in our framework. Additionally, we visualize the distribution of the score s_i on the challenging SSB datasets in Fig. 4 which shows that our method can successfully distinguish samples from ‘Old’ and ‘New’ classes in the unlabelled data of both the training and validation splits of the original dataset.

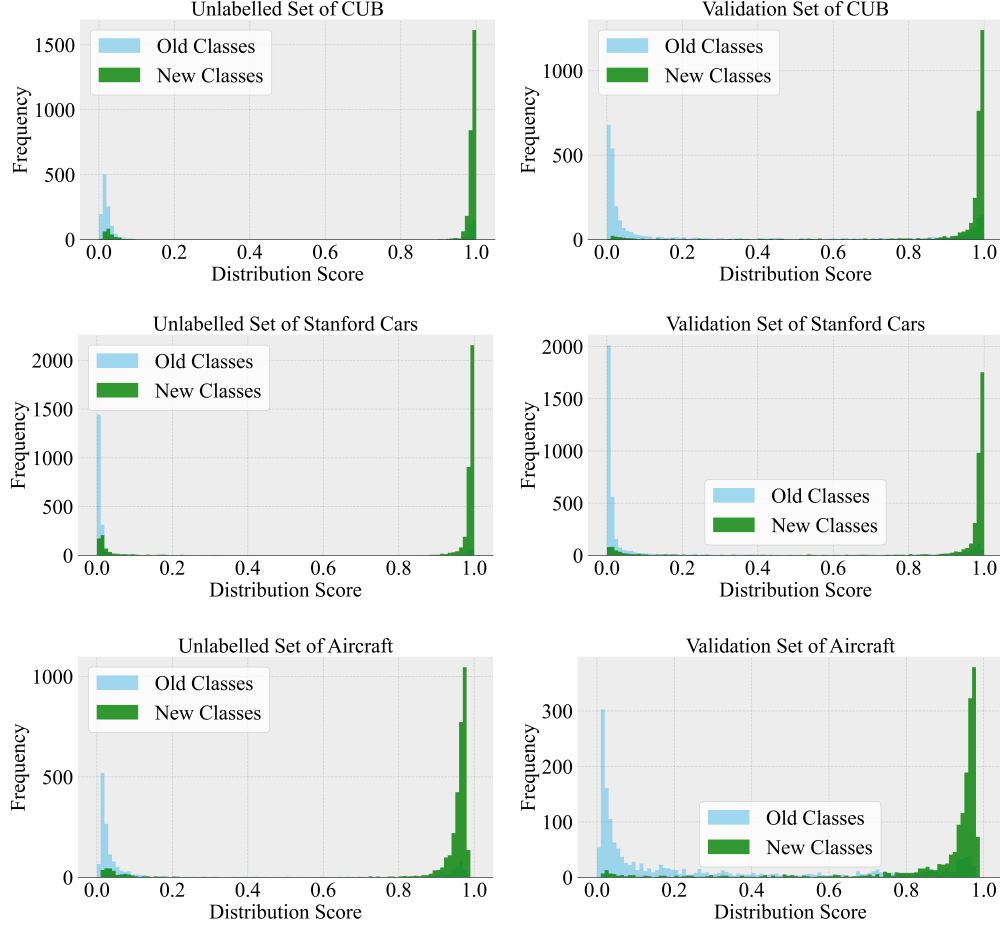


Figure 4: Histograms of the distribution scores s_i for datasets in SSB Vaze et al. (2022b).

Table 6: OOD performance in terms of AUROC on unlabelled data, including CIFAR-10 Krizhevsky et al. (2009), CIFAR-100 Krizhevsky et al. (2009), ImageNet-100 Deng et al. (2009), CUB Wah et al. (2011), Stanford Cars Krause et al. (2013), and FGVC-Aircraft Maji et al. (2013).

	CIFAR-10	CIFAR-100	ImageNet-100	CUB	Stanford Cars	FGVC-Aircraft
\mathcal{L}_{sdl}	66.1	90.8	96.5	77.5	78.6	76.2
$\mathcal{L}_{sdl} + \mathcal{L}_{gcd} + \mathcal{L}_{adl}$	97.5	94.8	99.5	86.8	89.6	86.3

I ANALYSIS OF ATTENTION MAPS

In our DebGCD framework, both the backbone embedding space and the GCD classifier are optimized. Thus, the `CLS` token is indirectly optimized. We can glean insights from its attention with the patch embeddings. In Fig. 5, we visualize the attention maps from the final transformer block in the DINO backbone Caron et al. (2021) on the three fine-grained datasets in SSB benchmark Vaze et al. (2022b). Within this final block, a multi-head self-attention layer with 12 attention heads attends to the input features, producing 12 attention maps between the `CLS` token and patch embeddings at a resolution of 14×14 . Following Caron et al. (2021), we compute the mean value of these attention maps and upsample them to the image size to visualize the most prominent regions. The visualization demonstrates that the attention maps generated by our model predominantly focus on the object of interest, effectively ignoring spurious factors and background clutter, while those of the DINO baseline are more scattered over the entire image.

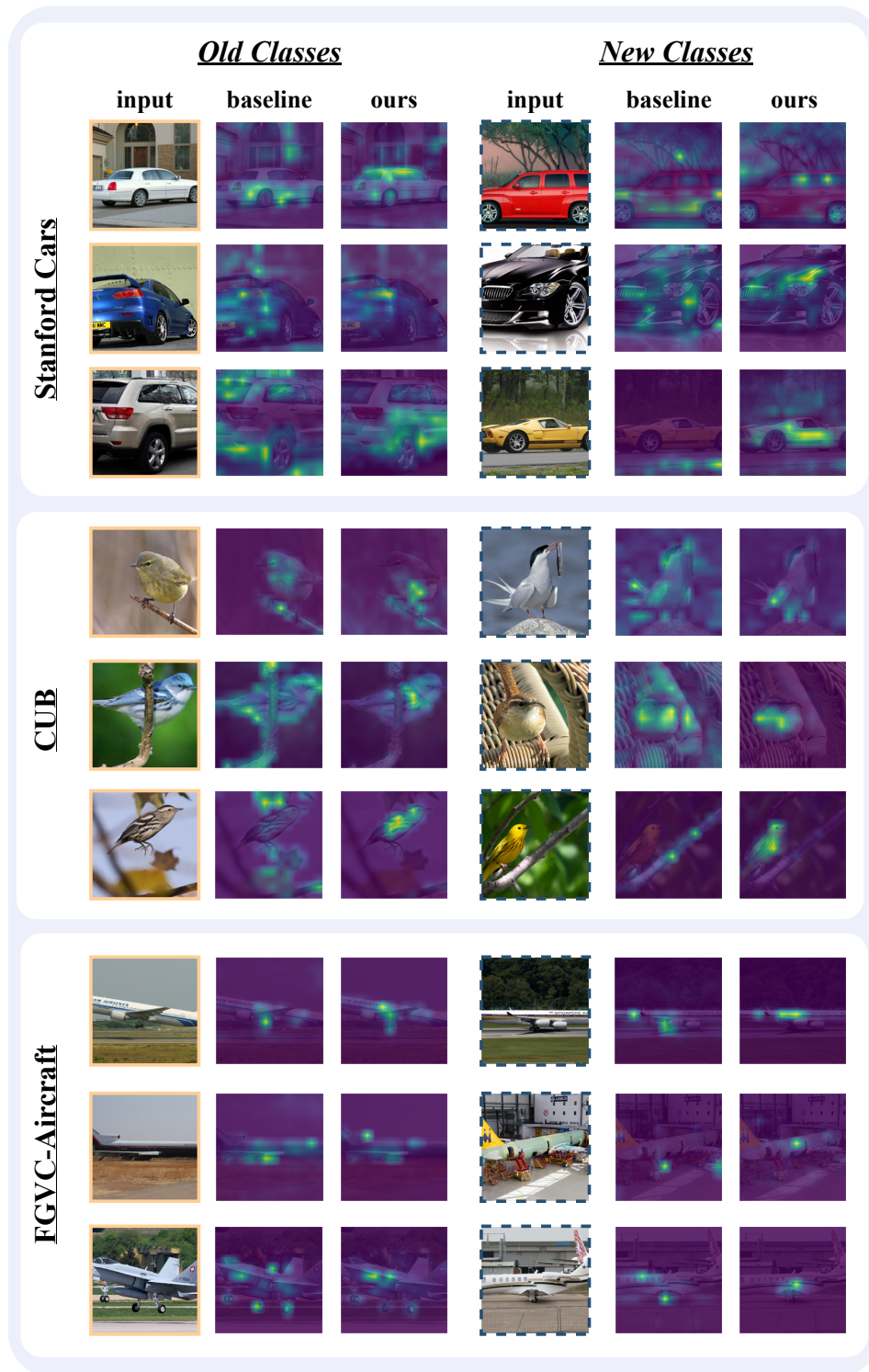


Figure 5: Visualization of attention maps. Our method successfully directs its attention towards foreground objects, irrespective of whether they belong to the ‘Old’ or ‘New’ classes. The baseline denotes the pre-trained DINO.

J ABLATION STUDIES ON MORE DATASETS

In addition to the Stanford Cars dataset, we present ablation results on additional datasets to validate the effectiveness of the proposed components. These include the other two datasets from the SSB benchmark: CUB [Wah et al. \(2011\)](#) and FGVC-Aircraft [Maji et al. \(2013\)](#), as well as the generic dataset ImageNet-100 [Deng et al. \(2009\)](#), detailed in Tab. 7. The results indicate that directly applying debiased learning to the original GCD classifier results in a performance decline across all three datasets (Row (1) vs. Row (2)). In contrast, utilizing an auxiliary classifier leads to performance improvements of 3.3%, 3.5%, and 1.7% on the three datasets, respectively, as observed in Row (1) vs. Row (3). This underscores the importance of the auxiliary classifier in achieving effective debiased learning. Moreover, the joint training of the debiased classifier and the OOD detector provides further enhancements (Row (3) vs. Row (5)). Lastly, the incorporation of distribution guidance results in additional performance improvements. These findings align with those observed on the Stanford Cars dataset, as demonstrated in manuscript.

Table 7: Ablations on more datasets, including CUB [Wah et al. \(2011\)](#), FGVC-Aircraft [Maji et al. \(2013\)](#) and ImageNet-100 [Deng et al. \(2009\)](#). ACC of ‘All’, ‘Old’ and ‘New’ categories are listed.

	Debiased Learning	Auxiliary Classifier	Semantic Dist. Learning	Dist. Guidance	CUB			FGVC-Aircraft			ImageNet-100		
					All	Old	New	All	Old	New	All	Old	New
(1)	✗	✗	✗	✗	60.3	65.6	57.7	54.2	59.1	51.8	83.0	93.1	77.9
(2)	✓	✗	✗	✗	58.6	72.3	51.7	53.7	62.9	49.1	82.8	94.1	77.2
(3)	✓	✓	✗	✗	63.8	69.3	61.1	57.7	59.8	56.5	84.7	94.0	80.0
(4)	✗	✗	✓	✗	61.3	69.4	57.3	56.6	64.8	52.5	83.5	92.4	78.9
(5)	✓	✓	✓	✗	64.9	70.9	61.9	59.4	64.4	56.9	85.0	93.8	80.3
(6)	✓	✓	✓	✓	66.3	71.8	63.5	61.7	63.9	60.6	85.9	94.3	81.6

K IMPACT OF HYPERPARAMETERS

In this section, we analyze the impact of hyperparameters in our DebGCD framework, including the depth of the projection network ρ_s , loss weights, and the number of tuned blocks.

Depth of projection network ρ_s . As discussed in the paper, it is essential to disentangle the OOD and GCD feature spaces due to the differing learning objectives of these two tasks. To assess the impact of the depth of the projection network ρ_s , we conduct an experiment on the SSB benchmark, focusing on the number of layers in this MLP network. Here, a depth of 0 denotes the absence of a projection network, meaning that the two tasks are optimized within the same feature space. As shown in Tab. 8, incorporating a 1-layer ρ_s results in performance improvements by 1.3%, 1.6% and 1.1% on CUB, Stanford Cars, and FGVC-Aircraft, respectively. The average GCD performance across all categories of DebGCD gradually improves as the number of MLP layers increases from 0 to 5. However, extending the MLP to 7 layers yields little to no further improvement in performance. In our implementation, we therefore adopt a 5-layer MLP for ρ_s in our framework.

Loss weights λ_{sdl} and λ_{adl} . For these two loss weights, we first intuitively set the default value based on existing literature and our hypothesis. Our rationale for selecting values for the loss weights is as follows: For λ_{sdl} , we take inspiration from the previous literature using OVA classifier [Saito & Saenko \(2021\)](#). In the paper, the model is fine-tuned with a learning rate of 10^{-3} , while the learning rate in the SimGCD baseline is 0.1 (which is 100 times larger than 10^{-3}). To achieve a similar learning effect, as validated in [Saito & Saenko \(2021\)](#), we scale our λ_{sdl} value from 1.0 down to 1/100. Therefore, we set $\lambda_{sdl} = 0.01$ by default. For λ_{adl} , the weight of the debiased classifier, we expect it to play an important role similar to that of the original GCD classifier (where the loss weight is set to 1.0). Thus, we have defaulted this value to 1.0. After determining the default values, we conducted experiments on the SSB benchmark regarding the two loss weights by exploring values around the defaults. For λ_{sdl} , the range was (0.005, 0.01, 0.02). As for λ_{adl} , the range was (0.5, 1.0, 2.0). The impact of λ_{sdl} is detailed below in Tab. 9, with λ_{adl} set to 1.0. The impact of λ_{adl} is illustrated below in Tab. 10, with λ_{sdl} set to 0.01. The results are in line with our hypothesis, indicating that our selected hyperparameters are indeed reasonable.

Table 8: GCD performance on SSB [Vaze et al. \(2022b\)](#) using different number of layers in ρ_s .

MLP layer	CUB			Stanford Cars			FGVC-Aircraft			Average
	All	Old	New	All	Old	New	All	Old	New	All
0	63.6	75.2	57.8	62.3	76.2	54.1	59.6	62.2	58.3	61.8
1	64.9	71.6	61.6	63.9	80.2	56.0	60.7	63.7	59.2	63.1
3	66.0	73.5	62.3	64.7	82.2	56.2	61.1	64.2	59.5	63.9
5	66.3	71.8	63.5	65.3	81.6	57.4	61.7	63.9	60.6	64.4
7	65.8	72.0	62.7	64.8	80.5	57.3	61.9	65.2	60.3	64.1

Table 9: GCD performance on SSB [Vaze et al. \(2022b\)](#) using different values of λ_{sdl} .

λ_{sdl}	CUB			Stanford Cars			FGVC-Aircraft			Average
	All	Old	New	All	Old	New	All	Old	New	All
0.02	65.5	73.2	61.6	64.3	79.2	57.1	60.6	63.5	59.1	63.5
0.01	66.3	71.8	63.5	65.3	81.6	57.4	61.7	63.9	60.6	64.4
0.005	65.8	72.4	62.5	64.9	81.2	57.0	62.1	65.4	60.3	64.3

Table 10: GCD performance on SSB [Vaze et al. \(2022b\)](#) using different values of λ_{adl} .

λ_{adl}	CUB			Stanford Cars			FGVC-Aircraft			Average
	All	Old	New	All	Old	New	All	Old	New	All
0.5	64.3	72.2	60.3	63.6	79.3	56.1	60.2	63.5	58.6	62.7
1.0	66.3	71.8	63.5	65.3	81.6	57.4	61.7	63.9	60.6	64.4
2.0	65.5	70.8	62.8	64.1	83.0	55.0	60.4	63.5	58.8	63.3

Number of tuned blocks. In the baseline configuration [Wen et al. \(2023\)](#), only the last transformer block of the ViT-B/16 backbone is fine-tuned during training. In contrast, our framework incorporates additional tasks, including OOD detection and debiased learning, which would require different embedding spaces, thus calling for the need of more trainable parameters. In our experiments on both fine-grained and generic datasets, we explore tuning the last two blocks, and we note that tuning more than two blocks may lead to instability during training. Furthermore, we observe that increasing the number of tuned blocks can improve performance on specific datasets, particularly those that are fine-grained. As shown in Table 11, tuning one additional transformer block leads to a performance improvement of over 1% on the fine-grained datasets. In contrast, the performance enhancement on the generic datasets is more modest, at no more than 0.6%. Similar strategies have also been employed in previous methods, such as Infosieve [Rastegar et al. \(2023\)](#).

Table 11: GCD performance of SimGCD and DebGCD by tuning different numbers of transformer blocks.

Method	# of tuned blocks	CUB			Stanford Cars			FGVC-Aircraft			ImageNet-100			CIFAR-100		
		All	Old	New	All	Old	New	All	Old	New	All	Old	New	All	Old	New
SimGCD	1	60.3	65.6	57.7	53.8	71.9	45.0	54.2	59.1	51.8	83.0	93.1	77.9	80.1	81.2	77.8
SimGCD	2	60.8	65.8	58.4	53.6	67.6	49.8	52.8	56.8	50.8	83.2	92.9	78.3	79.4	80.1	77.3
DebGCD	1	65.1	70.9	62.2	63.0	80.2	54.7	60.4	65.0	58.1	85.7	94.0	81.5	82.4	83.6	79.5
DebGCD	2	66.3	71.8	63.5	65.3	81.6	57.4	61.7	63.9	60.6	85.9	94.3	81.6	83.0	84.6	79.9

L STABILITY ANALYSIS

Following the baseline established in [Wen et al. \(2023\)](#), we also assess the stability of the proposed method across all datasets utilized in our experiments. Tab. 12 reports the average results together with the standard deviations, over three independent runs. Compared to the baseline results reported in [Wen et al. \(2023\)](#), we observe that the variance of DebGCD is even smaller, despite achieving significantly higher performance.

Table 12: Complete results of DebGCD and SimGCD over three independent runs.

Dataset	SimGCD			DebGCD		
	All	Old	New	All	Old	New
CUB	60.3±0.1	65.6±0.9	57.7±0.4	66.4±0.4	72.9±0.6	63.2±0.4
Stanford Cars	53.8±2.2	71.9±1.7	45.0±2.4	65.2±0.7	81.7±1.2	57.3±0.6
FGVC-Aircraft	54.2±1.9	59.1±1.2	51.8±2.3	61.7±0.5	65.9±1.2	59.5±1.1
CIFAR-10	97.1±0.0	95.1±0.1	98.1±0.1	97.3±0.1	95.0±0.2	98.4±0.1
CIFAR-100	80.1±0.9	81.2±0.4	77.8±2.0	83.1±0.7	84.7±0.7	80.0±0.9
ImageNet-100	83.0±1.2	93.1±0.2	77.9±1.9	86.1±0.6	94.5±0.5	81.8±0.6
ImageNet-1K	57.1±0.1	77.3±0.1	46.9±0.2	64.9±0.3	82.1±0.2	56.4±0.4
Oxford-Pet	-	-	-	93.2±0.2	86.3±0.1	96.8±0.3
Herbarium19	44.0±0.4	58.0±0.4	36.4±0.8	44.9±0.3	59.3±0.3	37.1±0.5

M PREDICTION ERROR ANALYSIS

In this section, we provide quantitative analysis on the improvements brought by our method from the perspective of prediction errors. Particularly, we examine the baseline model’s prediction by categorizing the errors into four types based on the relationship between the predicted (‘Pred’) and ground-truth (‘GT’) classes: ‘True Old’, ‘False New’, ‘False Old’, and ‘True New’. ‘True Old’ refers to incorrectly predicting an ‘Old’ class sample as another ‘Old’ class. ‘False New’ indicates incorrectly predicting an ‘Old’ class sample as a ‘New’ class. Conversely, ‘False Old’ means incorrectly predicting a ‘New’ class sample as an ‘Old’ class, and ‘True New’ refers to incorrectly predicting a ‘New’ class sample as another ‘New’ class. From this perspective, our debiased learning method primarily aims to mitigate the label bias between ‘Old’ and ‘New’ classes, thereby reducing the likelihood of ‘New’ class samples being predicted as ‘Old’. Consequently, this reduction in bias leads to a decrease in ‘False Old’ predictions while reducing the errors of all the other three types.

In Fig. 6, we present the ratios of the four types of *prediction errors* as a proportion of the total number of samples in the new or old categories across three datasets in the SSB benchmark. As shown in Fig. 6 (a), the error distributions vary significantly across datasets. Notably, the Stanford Cars dataset exhibits the highest number (16.5%) of ‘False Old’ samples, explaining why our method demonstrates the most substantial performance improvement on this dataset. In contrast, the CUB dataset shows the fewest (8.0%) ‘False Old’ samples, indicating relatively limited potential for performance enhancement. Comparing Fig. 6 (a) and Fig. 6 (b), we can see a significant reduction on the ratio of ‘False Old’ as well as other three types of errors on all the three datasets.

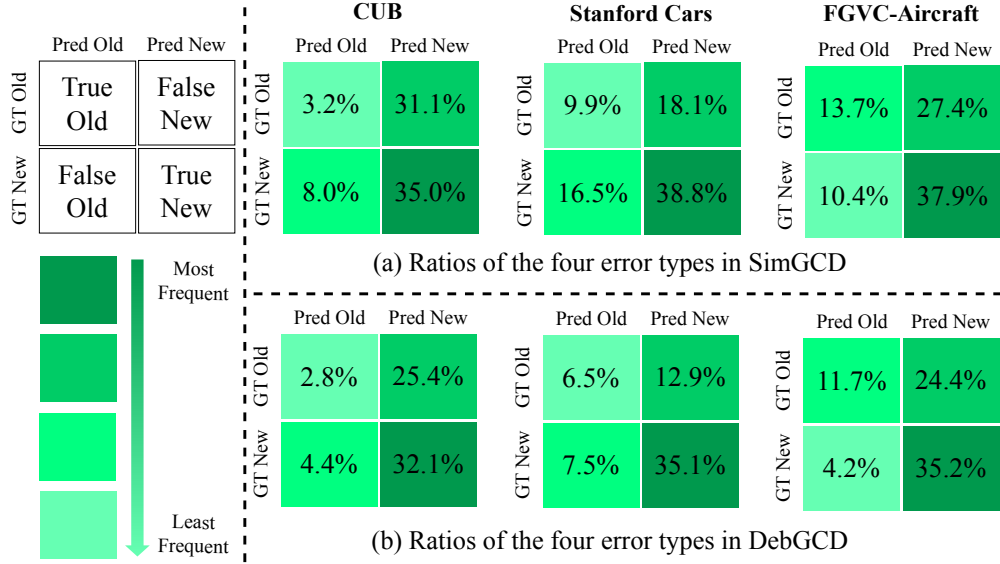


Figure 6: Ratios of the four types of prediction errors in GCD on SSB benchmark using SimGCD and DebGCD with DINO [Caron et al. \(2021\)](#) pre-trained backbone. ‘Pred’ and ‘GT’ refer to the predicted and ground-truth results, respectively.

REFERENCES

- Kaidi Cao, Maria Brbic, and Jure Leskovec. Open-world semi-supervised learning. In *ICLR*, 2022.
- Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021.
- Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.
- Yixin Fei, Zhongkai Zhao, Siwei Yang, and Bingchen Zhao. Xcon: Learning with experts for fine-grained category discovery. In *BMVC*, 2022.
- Enrico Fini, Enver Sangineto, Stéphane Lathuiliere, Zhun Zhong, Moin Nabi, and Elisa Ricci. A unified objective for novel class discovery. In *ICCV*, 2021.
- Kai Han, Sylvestre-Alvise Rebuffi, Sebastien Ehrhardt, Andrea Vedaldi, and Andrew Zisserman. Autonovel: Automatically discovering and learning novel visual categories. *IEEE TPAMI*, 2021.
- Shaozhe Hao, Kai Han, and Kwan-Yee K Wong. Cipr: An efficient framework with cross-instance positive relations for generalized category discovery. *TMLR*, 2024.
- Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCV workshop*, 2013.
- Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.
- James MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, volume 5, pp. 281–298. University of California press, 1967.
- Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.
- Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.
- Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *CVPR*, 2012.
- Nan Pu, Zhun Zhong, and Nicu Sebe. Dynamic conceptional contrastive learning for generalized category discovery. In *CVPR*, 2023.
- Sarah Rastegar, Hazel Doughty, and Cees Snoek. Learn to categorize or categorize to learn? self-coding for generalized category discovery. In *NeurIPS*, 2023.
- Kuniaki Saito and Kate Saenko. Ovanet: One-vs-all network for universal domain adaptation. In *ICCV*, 2021.
- Yiyu Sun and Yixuan Li. Opencon: Open-world contrastive learning. *TMLR*, 2022.
- Kiat Chuan Tan, Yulong Liu, Barbara Ambrose, Melissa Tulig, and Serge Belongie. The herbarium challenge 2019 dataset. *arXiv preprint arXiv:1906.05372*, 2019.
- Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 2008.
- Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Generalized category discovery. In *CVPR*, 2022a.
- Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. The semantic shift benchmark. In *ICML workshop*, 2022b.
- Sagar Vaze, Andrea Vedaldi, and Andrew Zisserman. No representation rules them all in category discovery. In *NeurIPS*, 2023.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

Hongjun Wang, Sagar Vaze, and Kai Han. Sptnet: An efficient alternative framework for generalized category discovery with spatial prompt tuning. In *ICLR*, 2024.

Xin Wen, Bingchen Zhao, and Xiaojuan Qi. Parametric classification for generalized category discovery: A baseline study. In *ICCV*, 2023.