Labeled Data Selection for Category Discovery – Supplementary Material

Bingchen Zhao[1] Nico Lang[2] Serge Belongie[2] Oisin Mac Aodha[1]

[1]University of Edinburgh [2]University of Copenhagen

A Soft weighting - Beta method

In this section, we aim to provide additional ablation results to justify the design choices we made in the main text.

A.1 Additional ablations

Ablation on α and β values. In Tab. A1, we present more results (*i.e.* in additional to the results in Tab. 5 in the main paper) where we use a wider range of α and β values. We observe that the default values of $\alpha = \beta = 5$ across all datasets offer a competitive performance among other choices, and that when we chose parameters that select labeled examples that are more similar to the unlabelled set (*i.e.* $\alpha > \beta$) results in worse performance.

Ablation of using reweighting *vs*. resampling. The weight we obtained from Eq.13 from the main paper can also be used to resample the dataset rather than weighting the loss value. The resampling of the dataset can be done by normalizing the weight associated with each of the categories to a probability value and then using this probability to sample data from the dataset to form training mini-batches. When using resampling, we use equal weighting in the loss function during training. In Tab. A2, we present the comparison of using the weight for resampling the dataset and weighting the loss. The difference between the two approaches is marginal, thus we choose to default the design of the beta method to do soft weighting on the losses.

Ablation of the similarity metric. In Eq. 12 of the main paper, we compute the similarity score as the cosine similarity of the labeled centroids to the farthest unlabeled data point. However, other ways of computing this similarity are also potentially valid. In Tab. A3, we evaluate using other choices, such as using the maximum similarity of the labeled centroids to the unlabeled data and the median of the all cosine similarity score between the labeled centroids to the unlabeled data points. We can see from the results that using the cosine similarity of the most similar unlabeled data (*i.e.* 'max') leads to a performance drop compared to the no selection baseline, while using the median value of all similarities and the minimum value of the similarities yield similar results.

Table A1: Ablations on more values of α and β. We default to $\alpha = \beta = 5$ for our experiments. Scores are performance on 'New' categories.

α	β	CUB	Aircraft	SDogs
1	1	52.3	49.3	54.2
7	3	42.4 (-9.9)	38.5 (-10.8)	46.0 (-7.2)
9	3	40.2 (-12.1)	36.7 (-12.6)	45.2 (-9.0)
3	7	57.4 (+5.1)	55.4 (+6.1)	57.2 (+3.0)
3	9	57.7 (+5.4)	55.8 (+6.5)	56.1 (+1.9)
5	5	56.8 (+4.5)	56.9 (+7.6)	56.6 (+2.4)

Table A2: Our main experiments use soft weighting. Here we ablate using the weight for resampling the dataset instead of weighting the loss. Scores are performance on 'New' categories.

	CUB	Aircraft	SDogs
baseline	52.3	49.3	54.2
resampling	56.5 (+4.2)	57.2 (+7.9)	56.9 (+2.7)
soft-weight	56.8 (+4.5)	56.9 (+7.6)	56.6 (+2.4)

Table A3: Ablations on the design choice of Eq. 12 in the main paper. We default to using the minimum similarity of the labeled centroids to the unlabeled data.

	CUB	Aircraft	SDogs
baseline	52.3	49.3	54.2
max	47.9 (-4.4)	45.1 (-4.2)	50.2 (-4.0)
median	57.0 (+4.7)	56.7 (+7.4)	56.2 (+2.0)
min	56.8 (+4.5)	56.9 (+7.6)	56.6 (+2.4)

B Binning method

B.1 Implementation details

In the main text we briefly outlined our hard selection-based binning method. In this section, we provide more details on how the binning method is implemented. The binning method for data selection simply 'chunks' the labeled source dataset into several equal-sized subsets based on the similarity to the target unlabeled set, and then selects the chunks that are not too similar or too dissimilar. The difficulty is that we cannot know how unrelated data is in the labeled data pool, thus we cannot determine how many chunks to use or which chunk to select. To address this, we design a selection method that first filters out unrelated data and then performs selection after. Our binning approach is illustrated in Fig. A1.

To successfully filter out unrelated data from the source, we would need a distance threshold. Setting a fixed value for the threshold would be hard, requiring a lot of trial and error. Thus we devised a method for automatically obtaining it. The high-level idea is that we can compute the distances between the data points *within* the target unlabeled data. This should give us a measure of how distant the categories in the target unlabeled data should be, and thus we can filter out data that has a larger distance than this threshold. To do this, we first randomly divide the target unlabeled data $\mathcal{D}^u$ into two sets, $\mathcal{D}_0^u$ and $\mathcal{D}_1^u$, each containing the same number of estimated clusters. Then with these sets we can compute:

$$\text{EMD}(\mathcal{D}^l \cup \mathcal{D}_0^u, \mathcal{D}_1^u) = \frac{\sum_{i \in I(\mathcal{D}^l \cup \mathcal{D}_0^u)} \sum_{j \in I(\mathcal{D}_1^u)} k_{i,j} d_{i,j}}{\sum_{i \in I(\mathcal{D}^l \cup \mathcal{D}_0^u)} \sum_{j \in I(\mathcal{D}_1^u)} k_{i,j}}, \tag{1}$$

where $I(\mathcal{D})$ stands for the indexes of the mean feature vectors of $\mathcal{D}$, $d_{i,j}$ is the distance between mean feature vectors, and $k_{i,j}$ is optimal flow.

Discarding distant unrelated data. $\mathcal{D}_0^u$ is used during the Earth Mover's Distance (EMD) computation to obtain a threshold value that can be used to filter out distant unrelated labeled categories that are dissimilar to the target data. We start by defining:

$$d_{i,:} = \frac{\sum_{j \in I(\mathcal{D}_1^u)} k_{i,j} d_{i,j}}{\sum_{j \in I(\mathcal{D}_1^u)} k_{i,j}}, \tag{2}$$

where $i \in I(\mathcal{D}^l \cup \mathcal{D}_0^u)$, $d_{i,:}$ denotes the distance of one source category/cluster i to the target unlabeled data, and $k_{i,j}$ is the flow from Eq. (1). By calculating $\bar{d}_{\mathcal{D}^u} = \frac{1}{|I(\mathcal{D}_0^u)|} \sum_{i \in I(\mathcal{D}_0^u)} d_{i,:}$, which can be understood as the distance within the target unlabeled data itself, we can filter out any entry in $I(\mathcal{D}^l)$ that is in $\{i | d_{i,:} > \bar{d}_{\mathcal{D}^u}, i \in I(\mathcal{D}^l)\}$. The rationale behind this is that if a labeled category has a distance larger than $\bar{d}_{\mathcal{D}^u}$, it would be too distant from the target unlabeled data to provide a useful learning signal for a category discovery method. After this filtering, the remaining dataset $\mathcal{D}^r$ will contain labeled categories that have a smaller distance than the threshold $\bar{d}_{\mathcal{D}^u}$.

Selecting relevant data. In the previous step, we filtered out data from the source that was too dissimilar to the target. Next, we need to remove data that is too *similar*. To do this, we rank the labeled categories in the remaining source data $\mathcal{D}^r$ based on their distance to the target data. We then divide $\mathcal{D}^r$ into L equal-sized subsets $\mathcal{D}^r = \bigcup_{l=1}^{L} \mathcal{D}_l^r$, where $\mathcal{D}_1^r$ contains the most similar labeled categories to the target data, and $\mathcal{D}_L^r$ contains the most dissimilar labeled categories in $\mathcal{D}^r$. Empirically, we found that selecting the distant subsets such as $\mathcal{D}_L^r$ gives a better performance compared to selecting the similar ones such as $\mathcal{D}_1^r$. Perhaps surprisingly, we later show that the performance of using $\mathcal{D}_L^r$ sometimes outperforms using the entire subset $\mathcal{D}^r$, and can even outperform models that use all the original labeled data $\mathcal{D}^l$.

In practice, we randomly split $\mathcal{D}^u$ into $\mathcal{D}_0^u$ and $\mathcal{D}_1^u$ multiple times to get a more robust estimate of $\mathcal{D}^r$. We then select $\mathcal{D}_L^r$ as our new labeled dataset for supervising the category discovery process, where $L = 2$. We choose $L = 2$, as it strikes a balance of not being too similar, but still relevant to the task. The selection is done by assigning the weight of 1 to examples in $\mathcal{D}_L^r$ and assigning a weight of 0 to all the other examples. For implementation, as the examples with 0 weight will not influence the training process, we can discard them and only sample from $\mathcal{D}_l^r$ for training.

B.2 Ablations of the binning method

In this section, we provide ablations of the binning data selection method. The evaluation is performed using the CUB [14], Stanford Dogs, and FGVC-Aircraft [6] datasets from SSB [12]. The selection of labeled data is done using the combination of SSB, Stanford Dogs, and iNat-Insect as the source set, and we report the clustering accuracy on the 'New' categories.

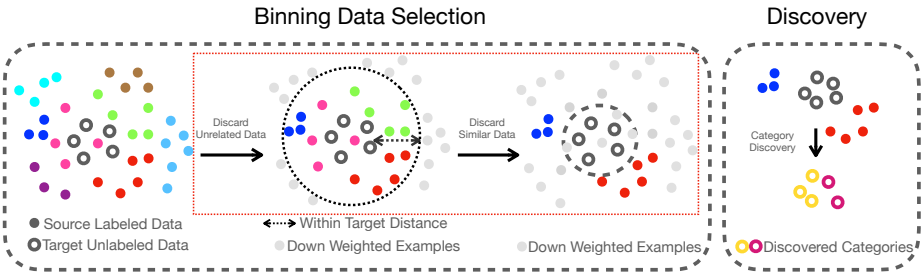

Fig. A1: Overview of our binning labeled data selection process for category discovery. We first discard labeled source data based on a threshold calculated from within the unlabeled target data. Next we discard data that is too similar to the unlabeled target data. The remaining labeled source data, along with the unlabeled target data, is then fed to a category discovery method.

Table A4: Here we ablate the major parts of the binning data selection method. We report accuracy on 'New' categories and use the 'all dataset' as the source pool. 'Ours' stands for applying the binning method on SimGCD.

	CUB	Aircraft	SDogs
Ours w/o Discard	12.1	8.9	10.0
Ours w/o Selection	45.7	41.4	47.2
Ours	56.7	56.8	55.6

Table A5: Here we estimate the number of clusters using off-the-shelf estimation methods. We report accuracy on 'New' categories where models use the 'all dataset' source pool. 'Ours' stands for applying the binning method on SimGCD. The final row of this table is using the ground truth number of categories.

	CUB	Aircraft	SDogs
Ours w/ [11]	55.7	53.7	54.7
Ours w/ [16]	54.2	52.9	53.1
Ours w/ GT	56.7	56.8	55.6

Directly partition the entire labeled source. Our proposed binning selection method has two steps. The first is to obtain a threshold $\bar{d}_{\mathcal{D}^u}$ for discarding irrelevant data from the labeled source. Then we chunk the remaining data into two equal-sized subsets based on the distance to the target data. The second subset is then selected for supervising the category discovery models. In Tab. A4, we present the results of ablating these two parts. We can see that if we remove the 'discard unrelated data' step (*i.e.* Ours w/o Discard), the performance drops significantly. This is because we end up selecting unrelated data to the target for supervising the category discovery model. Next, we remove the second selection step, and instead directly use all remaining data after the removal of unrelated data. Again, the performance also drops, as similar data is selected and the model gets confused.

Using estimated category numbers for clustering. To compute the mean feature vectors $\bar{\mathbf{h}}_c^u$ on the unlabeled target data we need to have the knowledge of how many clusters to use for a clustering algorithm like k-means. The common assumption of category discovery models is that the number of clusters in the target data is known *a priori* [2,11,15]. Each of our baseline methods also receives

this information. In this section, we study the performance of our binning selection method when the number of clusters is unknown and has to be estimated using off-the-shelf cluster number estimation methods [11, 16]. In Tab. A5, we present a performance comparison using an estimated number of clusters. Our binning selection method is robust to the number of clusters used.

Sensitivity to hyperparameters. Here we provide additional ablations on the impact of the hyperparameters of the binning method. Specifically, we investigate the role of the number of chunks L, the distance metric used, and the number of clusters.

Tab. A6 presents the ablation of the value of L. For the results in the main paper we select $L = 2$. For the different values of L explored in this ablation, we always select the L-th chunk for supervising the model, *i.e.* if $L = 3$, we discard the first two and use the 3rd chunk for supervising the category discovery model. We can see that our binning method is robust to a range of L, and note that higher values of L will result in smaller numbers of images per chunk, thus when the value of L is higher, the performance starts to decrease.

Table A6: Ablation of the number of chunks L. 'Ours' stands for applying the binning method on SimGCD.

Method	CUB			FGVC-Aircraft		
	All	Old	New	All	Old	New
Ours w. $L = 2$	58.2	64.6	56.7	55.7	57.9	56.8
Ours w. $L = 3$	56.1	62.3	55.0	53.4	55.2	53.4
Ours w. $L = 4$	57.1	62.0	55.6	56.1	58.2	55.7
Ours w. $L = 5$	53.0	57.8	50.1	48.7	51.4	49.1

Table A7: Ablation of distance metric used. The results here are based on applying binning on SimGCD.

Method	CUB			FGVC-Aircraft		
	All	Old	New	All	Old	New
Euclidean	58.2	64.6	56.7	55.7	57.9	56.8
Cosine	57.2	64.0	56.4	55.4	57.3	56.2
ℓ_2 Norm	59.0	64.7	56.9	55.8	57.4	57.2

Tab. A7 presents the ablation on different distance metrics to use when calculating the Earth Mover's Distance (EMD). The Euclidean distance is used by default in the main paper. The other distance metrics we tested include the cosine distance and an alternative where we first ℓ_2 normalize the features and

then compute the Euclidean distance. We are robust to these choices, and we can see that the performance is slightly better when using the ℓ_2 normalized Euclidean distance.

Table A8: Ablation where we vary the number of clusters. The results are based on applying binning on SimGCD.

Method	CUB			FGVC-Aircraft		
	All	Old	New	All	Old	New
GT	58.2	64.6	56.7	55.7	57.9	56.8
$K = 50$	55.4	60.3	53.1	54.2	57.7	55.8
$K = 100$	57.2	63.5	56.4	53.2	56.2	53.1
$K = 500$	52.3	57.2	49.3	50.1	51.2	47.6
$K = 1000$	49.6	54.2	46.2	45.7	48.9	44.3

Another important ablation is the number of clusters to use when clustering the unlabeled data. Note that all the baseline category discovery methods we have used in the main paper assume the number of clusters is known. As noted earlier, this is a common assumption in the category discovery literature. In Tab. A5, we presented the results of using different cluster number estimation methods for our binning selection methods. In Tab. A8 we present the results of setting the number of clusters to a fixed value. We can see that when the fixed number is close to the true number of clusters (*i.e.* 100 for CUB and 50 for Aircraft), the model gets the best performance. When the number is larger, the performance degrades gradually.

C Datasets

In Tab. A9, we present the dataset statistics for datasets we used in our experiments.

D Discovery methods

For completeness, in this section we briefly describe the GCD [11], XCon [2], and μGCD [13] methods that we have used for some of the comparisons in the main paper. GCD and XCon focus on the representation learning for the GCD tasks, while the classifier is directly implemented as a semi-supervised k-means classifier. μGCD utilizes loss for both representation learning and classifier learning, with the classifier implemented similarly to the parametric classifier in SimGCD.

For all four methods we evaluated (including SimGCD from the main paper), we assume that the number of categories in the unlabeled dataset is known. In practice where the number of categories is not known, it can be estimated

Table A9: The statistics of the datasets we used for our experiments.

	Labeled		Unlabeled	
Dataset	#Image	#Class	#Image	#Class
CUB [14]	1.5K	100	4.5K	200
Stanford Cars [5]	2.0K	98	6.1K	196
FGVC-Aircraft [6]	1.7K	50	5.0K	50
NABirds [9]	12K	200	36K	400
iNat-Insect [10]	31.5K	1263	94.8K	2526
Stanford Dogs [4]	3K	60	9K	120
Herbarium-19 [8]	8.9K	341	25.4K	683
ImageNet-100 [7]	31.9K	50	95.3K	100
ImageNet-1k-SSB [7]	1284K	2000	1920K	2000

using off-the-shelf category number estimation methods like semi-supervised k-means [11]. In Tabs. A5 and A8, we present the results of using off-the-shelf categories number estimation methods and the results of using fixed number of categories.

GCD. For representation learning, GCD [11] utilizes contrastive learning. Specifically, for the unlabeled data $\mathbf{x}_i \in B$, a self-supervised contrastive loss is employed:

$$\mathcal{L}_{\text{rep}}^{u} = \frac{1}{|B|} \sum_{\mathbf{x}_i \in B} -\log \frac{\exp(\hat{\mathbf{z}}_i^{\top} \tilde{\mathbf{z}}_i / \tau_u)}{\sum_{\mathbf{x}_j \in B} \exp(\hat{\mathbf{z}}_j^{\top} \tilde{\mathbf{z}}_i / \tau_u)}, \tag{3}$$

where $\mathbf{z}_i = m(f(\mathbf{x}_i))$ is the projected feature for contrastive learning, $\hat{\mathbf{z}}_i$ and $\tilde{\mathbf{z}}_i$ are the features of two augmentations of the same image $\mathbf{x}_i$, τ_u is a temperature value, and B is a mini-batch of unlabeled images. For the labeled data $\mathbf{x}_i \in B^l$, a supervised contrastive loss is employed to learn a discriminative representation:

$$\mathcal{L}_{\text{rep}}^{s} = \frac{1}{|B^l|} \sum_{\mathbf{x}_i \in B^l} \omega_i \frac{1}{|\mathcal{N}_i|} \sum_{p \in \mathcal{N}_i} -\log \frac{\exp(\hat{\mathbf{z}}_i^{\top} \tilde{\mathbf{z}}_p / \tau_s)}{\sum_{n \neq i} \exp(\hat{\mathbf{z}}_i^{\top} \tilde{\mathbf{z}}_n / \tau_s)}, \tag{4}$$

where the indices of images with the same label as $\mathbf{x}_i$ is stored in $\mathcal{N}_i$, ω_i is the weight for example i, and τ_s is another temperature value. The representation used for category discovery is learned using these two losses with a weighting factor λ, $\mathcal{L}_{\text{rep}} = (1-\lambda)\mathcal{L}_{\text{rep}}^{u} + \lambda \mathcal{L}_{\text{rep}}^{s}$.

After the representation is learned, a semi-supervised k-means algorithm is applied for assigning labels to the unlabeled data. This algorithm is modified from the original unconstrained k-means algorithm by forcing the assignment of the labeled data to be the ground truth label. Specifically, this algorithm first obtains the initial centroids using k-means++ [1], updates the centroids, and then assigns labels like the k-means algorithm except it forces the labels of the labeled data to be the ground truth.

XCon. XCon [2] extends GCD by proposing dataset partitioning to learn more discriminative representations. Specifically, it first partitions the whole dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}$ into K sub-dataset $\{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_K\}$ using a k-means clustering of a self-supervised representation, and then uses the same supervised and self-supervised contrastive losses as GCD on each of the sub-datasets (*i.e.* by sampling negative and positive samples only within the sub-dataset). XCon claims that this partitioning can help the sampling of hard negative examples, and thus help representation learning. Apart from these contrastive learning losses on the partitioned dataset, XCon also employs the same contrastive losses over the whole dataset as GCD. The label assignment is performed similarly to GCD, using a semi-supervised k-means algorithm.

μGCD. μGCD [13] extends SimGCD [15] method, which we introduced in the main paper. μGCD adopts a 'cosine' classifier [3] with the features being 'L2'-normalized to help the model learn a more balanced classifier across old and new categories. The loss functions for learning are the same as the SimGCD learning losses. Different from SimGCD, μGCD makes use of a teacher-student architecture, where the student is fed with a strongly augmented image and is trained with gradient descent, and the teacher uses a weakly augmented image and is updated via moving average from the student model.

References

1. Arthur, D., Vassilvitskii, S., et al.: k-means++: The advantages of careful seeding. In: Symposium on Discrete Algorithms (2007) 7
2. Fei, Y., Zhao, Z., Yang, S., Zhao, B.: Xcon: Learning with experts for fine-grained category discovery. In: BMVC (2022) 4, 6, 8
3. Gidaris, S., Komodakis, N.: Dynamic few-shot visual learning without forgetting. In: CVPR (2018) 8
4. Khosla, A., Jayadevaprakash, N., Yao, B., Fei-Fei, L.: Novel dataset for fine-grained image categorization. In: FGVC Workshop at CVPR (2011) 7
5. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3d object representations for fine-grained categorization. In: ICCV Workshops (2013) 7
6. Maji, S., Rahtu, E., Kannala, J., Blaschko, M., Vedaldi, A.: Fine-grained visual classification of aircraft. arXiv:1306.5151 (2013) 3, 7
7. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. IJCV (2015) 7
8. Tan, K.C., Liu, Y., Ambrose, B., Tulig, M., Belongie, S.: The herbarium challenge 2019 dataset. In: FGVC Workshop at CVPR (2019) 7
9. Van Horn, G., Branson, S., Farrell, R., Haber, S., Barry, J., Ipeirotis, P., Perona, P., Belongie, S.: Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In: CVPR (2015) 7
10. Van Horn, G., Mac Aodha, O., Song, Y., Cui, Y., Sun, C., Shepard, A., Adam, H., Perona, P., Belongie, S.: The inaturalist species classification and detection dataset. In: CVPR (2018) 7

11. Vaze, S., Han, K., Vedaldi, A., Zisserman, A.: Generalized category discovery. In: CVPR (2022) 4, 5, 6, 7
12. Vaze, S., Han, K., Vedaldi, A., Zisserman, A.: Open-set recognition: A good closed-set classifier is all you need? In: ICLR (2022) 3
13. Vaze, S., Vedaldi, A., Zisserman, A.: No representation rules them all in category discovery. In: NeurIPS (2023) 6, 8
14. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: Caltech-UCSD Birds 200. Computation & Neural Systems Technical Report (2010) 3, 7
15. Wen, X., Zhao, B., Qi, X.: Parametric classification for generalized category discovery: A baseline study. In: ICCV (2023) 4, 8
16. Zhao, B., Wen, X., Han, K.: Learning semi-supervised gaussian mixture models for generalized category discovery. In: ICCV (2023) 4, 5