Pattern Recognition Letters
journal homepage: www.elsevier.com

Rethinking Class Orders and Transferability in Class Incremental Learning
Supplementary Material

Chen He[a,b], Ruiping Wang[a,b,**], Xilin Chen[a,b]

[a] *Key Laboratory of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing, 100190, China*
[b] *University of Chinese Academy of Sciences, Beijing, 100049, China*

ABSTRACT

This is a supplementary material for "Rethinking Class Orders and Transferability in Class Incremental Learning", which offers additional implementation details, experimental results, and discussions due to the space limitation in the main paper.

1. Introduction

In this supplementary material, we show the implementation details of the datasets and the chosen Class Incremental Learning (CIL) methods (Sec. 4.1)[1], more quantitative results including the results on Group ImageNet (Sec. 4.2, Sec. 4.3, Sec. 4.4) etc., the visualization of the class orders (Sec. 4.4), and the potential applications (Sec. 4.6).

2. Implementation Details

2.1. Dataset

In this work, we choose Group ImageNet and Group iNaturalist as our datasets. The reason for choosing them is that the number of superclasses and the number of subclasses in each superclass is the same. The benefit is that under the 10×10 and 9×9 protocol of these datasets respectively, *even* and *group* have the same number of incremental phases. Since more incremental phases generally yield lower accuracy [1] for CILs, these two datasets offer a chance that *even* and *group* can be fairly compared with an equal number of incremental phases. The experiments of incremental protocols 20×5 of *even* or *group* mentioned in the main paper are performed by making the class 1-20, ..., 81-100 in *even* or *group* as new classes to be learned at each class increment (similar for 50×2).

Group ImageNet. Group ImageNet has 10 superclasses and each superclass has 10 subclasses. The implementation of this dataset has been already elaborated in this paper [2]. The class order *even* is generated by randomly shuffling a predefined class order to ensure that class 1-10, 11-20,..., 91-100 each covers 10 superclasses. Similarly, the class order *group* is generated by properly shuffling a predefined class order to ensure that class 1-10, 11-20,..., 91-100 come from the same superclass in each group of classes. For incremental protocols 20×5, we still use the above-mentioned version of *even* but simply make the CIL method learn 20 classes at each incremental phase (similar for 50×2). We use 5 different *even* class orders and 5 different *group* class orders in our experiments mentioned in Sec. 4.

Group iNaturalist. Group iNaturalist is a subset of iNaturalist [3], which has 9 superclasses and each superclass has 9 subclasses. Since the foreground in the original iNaturalist dataset may only occupy a small portion of the whole image, we use the ground-truth bounding boxes provided in the iNaturalist detection track and crop the foreground to make it more oriented for incremental recognition instead of detection.

Specifically, to ensure that the cropped images have the same image resolution, we find the smallest square that surrounds the ground-truth bounding box and resize the cropped image inside the square to 64×64. There are only 9 superclasses with bounding box annotations and for each superclass we simply choose the 9 subclasses in each superclass with the most images. The 9 superclasses are: Aquatic animals* (* means that most of the classes in this superclass are aquatic animals), Mollusca, Actinopterygii, Amphibia, Aves, Mammalia, Reptilia, Arachnida, Insecta. The subclasses in these superclasses are:

- **Aquatic Animals***: Anthopleura sola, Aurelia aurita,

**Corresponding author
e-mail: chen.he@vipl.ict.ac.cn (Chen He), wangruiping@ict.ac.cn (Ruiping Wang), xlchen@ict.ac.cn (Xilin Chen)

[1] We use the blue color to denote sections, tables in the main paper.

Callinectes sapidus, Cancer productus, Dermasterias imbricata, Physalia physalis, Pisaster ochraceus, Pollicipes polymerus, Scolopendra polymorpha.

- **Mollusca**: Ariolimax columbianus, Cryptochiton stelleri, Diaulula sandiegensis, Doris montereyensis, Hermissenda crassicornis, Hermissenda opalescens, Leukoma staminea, Phidiana hiltoni, Triopha catalinae.
- **Actinopterygii**: Cyprinus carpio, Lepomis cyanellus, Lepomis gibbosus, Lepomis macrochirus, Micropterus salmoides, Oncorhynchus mykiss, Pterois volitans, Salvelinus fontinalis, Zanclus cornutus.
- **Amphibia**: Gastrophryne carolinensis, Hyla chrysoscelis, Hyla versicolor, Lithobates catesbeianus, Lithobates pipiens, Lithobates sphenocephalus, Lithobates sylvaticus, Plethodon cinereus, Rana dalmatina.
- **Aves**: Acridotheres tristis, Aegithalos caudatus, Cassiculus melanicterus, Junco hyemalis oreganus, Larus californicus, Motacilla cinerea, Oreothlypis celata, Quiscalus quiscula, Sturnus vulgaris.
- **Mammalia**: Lepus californicus, Lepus europaeus, Lontra canadensis, Odocoileus hemionus californicus, Procyon lotor, Sylvilagus floridanus, Tamias striatus, Tamiasciurus douglasii, Ursus americanus.
- **Reptilia**: Ctenosaura similis, Micrurus tener, Pantherophis alleghaniensis, Pantherophis emoryi, Sceloporus consobrinus, Sceloporus undulatus, Thamnophis marcianus, Trachemys scripta elegans, Uta stansburiana.
- **Arachnida**: Argiope aurantia, Centruroides vittatus, Gasteracantha cancriformis, Latrodectus geometricus, Leucauge venusta, Nephila clavipes, Peucetia viridans, Phidippus johnsoni, Uroctonus mordax.
- **Insecta**: Abaeis nicippe, Apis mellifera, Argia moesta, Danaus gilippus, Dythemis fugax, Haematopis grataria, Olla v-nigrum, Plathemis lydia, Polistes dominula.

The class list above will be released together with the codes. Group iNaturalist has complementary characteristics with Group ImageNet: the classes on Group iNaturalist are more fine-grained; the number of samples for each class is highly imbalanced.

2.2. Transferability Measures and COSA

Semantic Distance. On Group ImageNet, all classes can map to the synsets in the WordNet hierarchy [4], and it is easy to leverage the WordNet module in NLTK [5] to calculate the Wu-Palmer distance [6]. On Group iNaturalist, since the classes are too fine-grained and can not map to the synsets in the WordNet hierarchy, we construct a simple 3-layer hierarchy manually. The first layer has only one node which is the root of the tree. The second layer has 9 nodes representing 9 superclasses. For each superclass, its 9 subclasses are positioned in the third layer.

Visual Distance. For numerical stability in calculating the statistical distance mentioned in Sec. 3.2, an identity matrix multiplied by 10^{-6} is added to the covariance matrix to ensure that the covariance matrix is positive definite when taking inversion or matrix logarithm.

Class Order Search Algorithm (COSA). k-medoids [7] is implemented by the Python module *PyClustering* [8]. The whole process of performing the reverse greedy search is as follows. Taking an incremental protocol 10×10 on Group ImageNet for example, we first use k-medoids to choose 90 classes out of the 100 classes, and the 10 classes that are not chosen make up the classes in the final increment. Then, we choose 80 classes out of the previously chosen 90 classes, and the 10 classes that are not chosen make up the classes in the penultimate increment, and so forth. Finally, a unique class order can be obtained.

2.3. CIL Method

The codes are implemented in Tensorflow 2 [9][2]. For Group ImageNet and Group iNaturalist, we adopt a modified ResNet-18 [10] that accepts an input size of 64×64. An Adam optimizer [11] is used and the base learning rate is 0.005 by cross validation. The learning rate is divided by 10 at 0.7 and 0.9 of the total number of epochs inspired by the implementation of iCaRL [1]. The weight decay is heuristically set to 10^{-4}. As for the exemplar selection strategy, we simply use random selection since it does not have much difference in performance with other selection strategies as noted by [12, 13, 14].

iCaRL [1].The difference with the original version of iCaRL is that a softmax cross-entropy loss is employed instead of a binary cross-entropy loss, because empirically softmax cross-entropy yields better performance.

End-to-End Incremental Learning (EEIL) [15]. The number of epochs in the balanced fine-tuning stage is set to 30 which is the same as the setting in the original paper of EEIL.

Large Scale Incremental Learning (LSIL) [16]. The number of epochs in the 2nd stage (i.e. bias correction stage) is twice the number of epochs in the 1st stage. The weighting factor of the L2 regularization on β in the 2nd stage is 0.1, and the ratio of train/validation split on the exemplars is 9:1 as the authors recommend in the paper.

3. More Experiments

3.1. Class Orders do Matter in CIL

The final classification accuracies of recent CIL methods under two class orders *even* and *group* on Group iNaturalist are shown in Table 1. From the results, it can be seen that *even* almost constantly outperforms *group*. An exception is IL2M [17] in the 27×3 protocol of Group iNaturalist. The reason is that absence of the distillation loss in IL2M mentioned in Sec. 4.6 makes it less sensitive to class orders.

Then, we display the experiments of *even* and *group* by varying the epochs during training or the number of total exemplars

[2]The code will be released to public.

M on Group ImageNet. The results are displayed in Table 2. From these tables, it can be observed that *even* almost consistently outperforms *group*, which is consistent with previous conclusions.

3.2. Relationship with Transferability

The estimated transferability of *even* and *group* on Group iNaturalist are summarized in Table 3. It can be seen that *even* generally has higher transferability (i.e. lower distance) than *group* for almost all settings, which is consistent with the findings in Sec. 4.3.

3.3. Effectiveness of COSA

In Table 3, we list the estimated transferability of the class order obtained by COSA (denoted as *greedy* since it takes a greedy search) on Group iNaturalist. It can be seen that *greedy* generally has comparable or higher transferability than *even*, which verifies the effectiveness of COSA. Also, we find that *greedy* based on MD-LED (Sup./SS) does not have higher transferability than *even* on Group iNaturalist. The possible reason is that we simply add MD and LED with no weighting factors (as mentioned in Footnote 4), which is less suitable for the more difficult Group iNaturalist. The performance of methods under *greedy* is comparable to or better than that under *even* (Table 4, Table 5, Table 6, Table 7), which again verifies the effectiveness of COSA.

The visualization of the class orders searched by COSA (denoted as *greedy*), *even*, and *group* on Group ImageNet in the 10×10 incremental protocol is shown in Fig. 1. From the figure, it can be seen that the classes inside each class group for *greedy* is highly interleaved as expected. Besides, we find that:

- The searched class orders differ not much, because the visual distances defined in these feature spaces can reflect the relatedness among the classes equally well;
- The classes in earlier increments tend to be more different, because COSA takes a reverse search and the search space is larger for earlier increments which incurs more possibilities (in a 10×10 protocol, there are C_{100}^{10} candidates for the final increment, $C_{100}^{10}C_{90}^{10}$ for the penultimate one, and so forth);
- Using different visual distances has a lesser impact on the searched class order than using different feature extractors.

From the visualization of the class orders on Group iNaturalist using a 9×9 incremental protocol (Fig. 2), we can draw similar conclusions as above.

To better show the superiority of COSA in some difficult cases, we further visualize the confusion matrices of a CIL method LSIL under *group* and the searched order by COSA on Group ImageNet (Fig. 3 and Fig. 4), which correspond to an accuracy of 21.22% and 42.78% respectively. In the visualization of these two confusion matrices, the classes are ordered by superclasses (i.e. class 1 to 10 belong to the same superclass, class 11 to 20 belong to the same superclass, and so forth). It can be seen that under the class order of *group*, the CIL method has more confusions between different superclasses compared with the searched order by COSA. The reason is that under the class order of *group* the new classes all come from the same superclass, which are dissimilar to previous classes and learning these new classes cannot offer a good review of the old classes (based on the analysis in Sec. 4.5 of the main paper). Thus, the model cannot distinguish old classes from new ones pretty well, resulting in more inter-superclass confusions.

4. Potential Applications

The proposed transferability measures can be used for model recommendation similar to other transferability measures [19]. COSA can be leveraged in the scenarios below:

Class Incremental Learning (CIL). In standard CIL, images and labels arrive simultaneously, and they are learned instantly. In open-world scenarios, images or labels may be absent.

- **w/o labels**: there exist unknown classes, where the images can be obtained by novelty detection methods. Then, the images can be clustered into underlying classes to facilitate the labeling process. Finally, COSA can be used to determine the labeling orders of these underlying classes. In situations where the labeling process cannot be done once and for all (e.g. time-consuming, expensive, etc.) but the model is required to be updated, COSA offers a more principled way of "labeling and learning";
- **w/o images**: in situations where the classes are known/predefined but the images need to be collected, COSA can offer a class order of collecting images. For example, in service robots where the semantic SLAM [20, 21] is deployed, the robot can identify different instances and recognize their basic categories when building the occupancy map. After that, it can automatically decide the orders based on the coarse information provided by semantic SLAM and go around the objects to capture more viewpoints to learn these instances.
- **w/ images and w/ labels** (standard CIL): learning in a multi-step class incremental way may outperform learning instantly, as shown in a recent work [22]. It is where COSA can be potentially used.

(Generalized) Zero/Few-Shot Learning ((G)ZSL/FSL). COSA can be used to determine the split of classes[3] with the highest transferability, which is possible to be extended to determine the split of classes with the lowest transferability. By using these two special splits, we can test the robustness of methods in these fields.

COSA can also be adapted and used in exemplar management in CIL [1, 15] by replacing the statistical distance with the sample distance. Ideally, we want to keep the exemplars that can easily transfer to other samples. An adapted COSA

[3] Seen/Unseen classes for (G)ZSL and novel/base classes for (G)FSL.

Table 1. Classification accuracies (%) of methods with the *even* and *group* class order on *Group iNaturalist* similar to Table 1.

Method	**9×9**		**27×3**	
	even	group	even	group
iCaRL [1]	**36.07 ± 0.81**	33.39 ± 2.12	**42.95 ± 0.55**	40.15 ± 0.81
EEIL [15]	**35.35 ± 1.21**	33.43 ± 3.15	**46.07 ± 1.78**	45.30 ± 0.79
LSIL [16]	**29.12 ± 2.92**	27.12 ± 3.39	**39.05 ± 1.77**	37.74 ± 2.29
IL2M [17]	**30.37 ± 1.09**	29.56 ± 2.57	40.74 ± 0.98	**41.18 ± 1.57**
WA [18]	**35.88 ± 1.50**	34.51 ± 2.19	**44.98 ± 2.09**	44.17 ± 0.84
Post-scaling [2]	**36.57 ± 0.43**	35.03 ± 2.76	**47.04 ± 1.00**	45.84 ± 1.03

Table 2. Classification accuracies (%) of methods with the even and group class order on Group ImageNet using a different number of epochs and a different number of total exemplars under a 10×10 incremental protocol. The organization of the table is similar to Table 1.

Method	**10×10** (20 epochs)		**10×10** (500 exemplars)	
	even	group	even	group
iCaRL [1]	**42.17 ± 0.60**	41.45 ± 1.02	**24.68 ± 1.03**	24.21 ± 1.84
EEIL [15]	**46.46 ± 0.45**	45.24 ± 0.69	**21.11 ± 1.96**	18.46 ± 2.31
LSIL [16]	**37.30 ± 0.80**	18.08 ± 4.40	**26.48 ± 0.99**	10.13 ± 2.34
IL2M [17]	**35.60 ± 0.73**	31.43 ± 1.38	**19.62 ± 0.68**	16.41 ± 1.28
WA [18]	**42.78 ± 0.95**	41.08 ± 1.41	**37.10 ± 0.44**	27.40 ± 3.20
Post-scaling [2]	**47.18 ± 1.14**	45.04 ± 1.33	**36.72 ± 0.54**	30.80 ± 2.08

Table 3. Estimated transferability of our searched class order *greedy*, *even* and *group* on *Group iNaturalist* using different CIL protocols similar to Table 2. The lower, the better.

Transferability Measure based on ___	**9×9**			**27×3**		
	greedy	even	group	greedy	even	group
WD (Sup.)	$6.14_{\times 10^3}$	$6.27_{\times 10^3}$	$6.79_{\times 10^3}$	$4.51_{\times 10^3}$	$4.60_{\times 10^3}$	$5.03_{\times 10^3}$
WD (SS)	$4.72_{\times 10^1}$	$4.66_{\times 10^1}$	$4.89_{\times 10^1}$	$2.70_{\times 10^1}$	$3.14_{\times 10^1}$	$3.27_{\times 10^1}$
MD-LED (Sup.)	$1.93_{\times 10^8}$	$2.06_{\times 10^8}$	$2.17_{\times 10^8}$	$1.66_{\times 10^8}$	$1.50_{\times 10^8}$	$1.56_{\times 10^8}$
MD-LED (SS)	$1.65_{\times 10^3}$	$1.45_{\times 10^3}$	$1.5_{\times 10^3}$	$1.15_{\times 10^3}$	$1.12_{\times 10^3}$	$1.04_{\times 10^3}$
Wu-Palmer	$2.53_{\times 10^1}$	$2.57_{\times 10^1}$	$2.99_{\times 10^1}$	$1.80_{\times 10^1}$	$1.85_{\times 10^1}$	$2.21_{\times 10^1}$

Table 4. Classification accuracies (%) of methods under the searched class orders by COSA (i.e. *greedy*) based on different transferability measures on Group ImageNet. The protocol is 20×5. The organization of the table is similar to Table 3.

Method	**Searched Order based on ___**						**Reference**	
	WD (Sup.)	WD (SS)	MD-LED (Sup.)	MD-LED (SS)	Wu-Palmer	average	even	group
iCaRL [1]	47.10	48.38	46.94	47.82	46.90	47.43	47.33	44.38
EEIL [15]	54.68	54.26	54.88	55.06	56.06	54.99	54.35	52.68
LSIL [16]	53.40	53.36	52.58	51.36	54.16	52.97	54.04	36.94
IL2M [17]	43.94	42.88	43.56	42.10	44.28	43.35	42.75	38.98
WA [18]	53.68	53.50	53.96	54.26	53.88	53.86	53.88	49.87
Post-scaling [2]	56.10	55.82	56.84	57.00	56.52	56.46	55.86	52.86

Table 5. Classification accuracies (%) of methods under the searched class orders by COSA (i.e. *greedy*) based on different transferability measures on Group ImageNet. The protocol is 50×2. The organization of the table is similar to Table 3.

Method	**Searched Order based on ___**						**Reference**	
	WD (Sup.)	WD (SS)	MD-LED (Sup.)	MD-LED (SS)	Wu-Palmer	average	even	group
iCaRL [1]	53.02	54.30	54.32	52.84	53.14	53.52	53.89	52.80
EEIL [15]	65.34	64.72	64.68	63.90	65.16	64.76	65.20	64.05
LSIL [16]	63.02	63.20	63.38	63.20	63.74	63.31	63.95	61.05
IL2M [17]	53.44	55.26	54.26	52.20	56.20	54.27	54.85	53.81
WA [18]	63.94	62.86	63.32	61.18	63.46	62.95	63.44	62.82
Post-scaling [2]	66.34	65.90	65.40	63.50	66.74	65.58	66.40	65.49

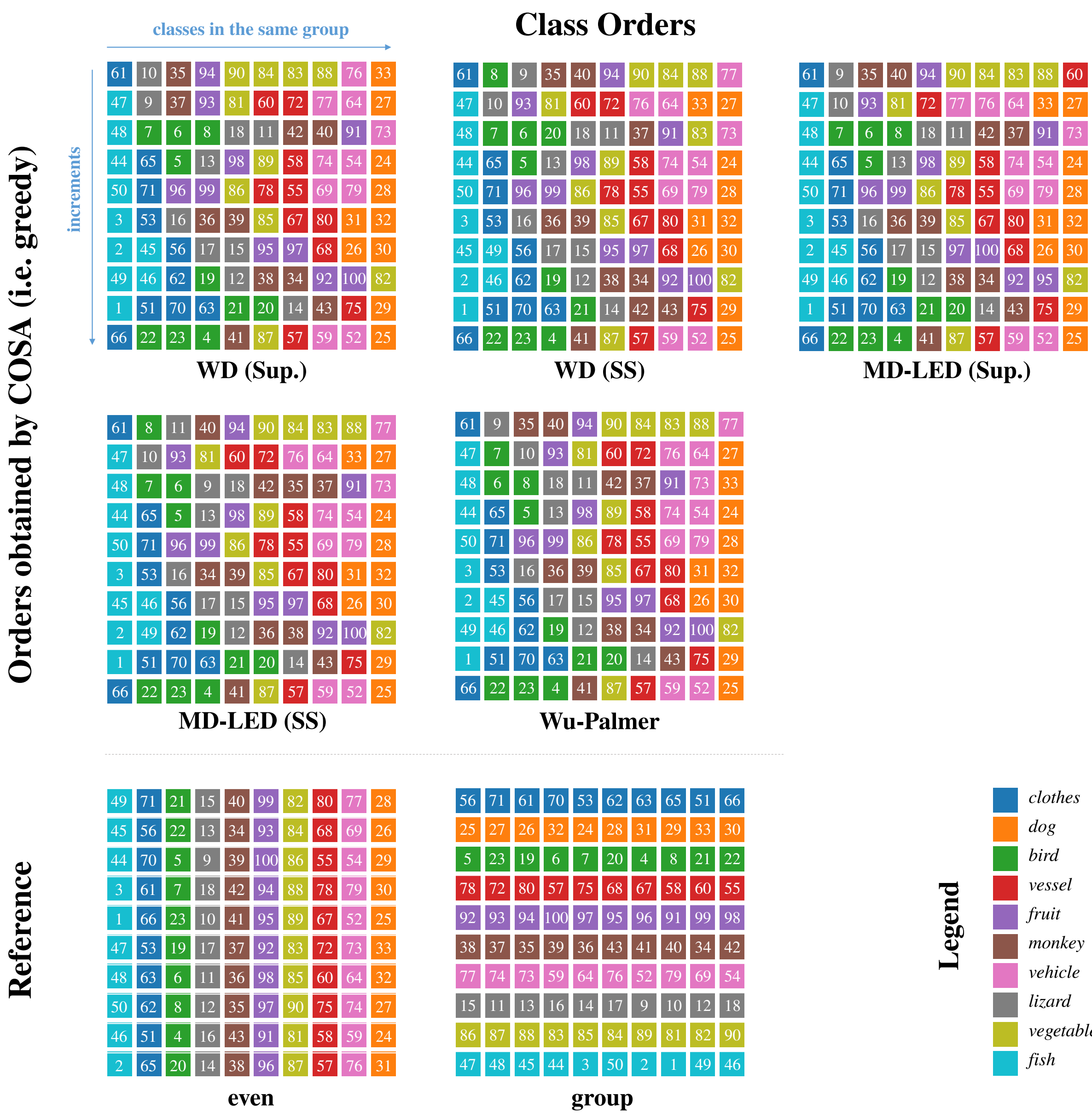

Fig. 1. Visualization of the class orders obtained by COSA (i.e. *greedy*), *even* and *group* using a 10×10 incremental protocol on Group ImageNet. The class order is represented as a large square, where 10 rows indicate 10 class increments, and 10 columns indicate that each class increment has 10 classes. Each small square represents a class, and its color implies the superclass denoted at the right of the figure. The number is the class ID in the dataset.

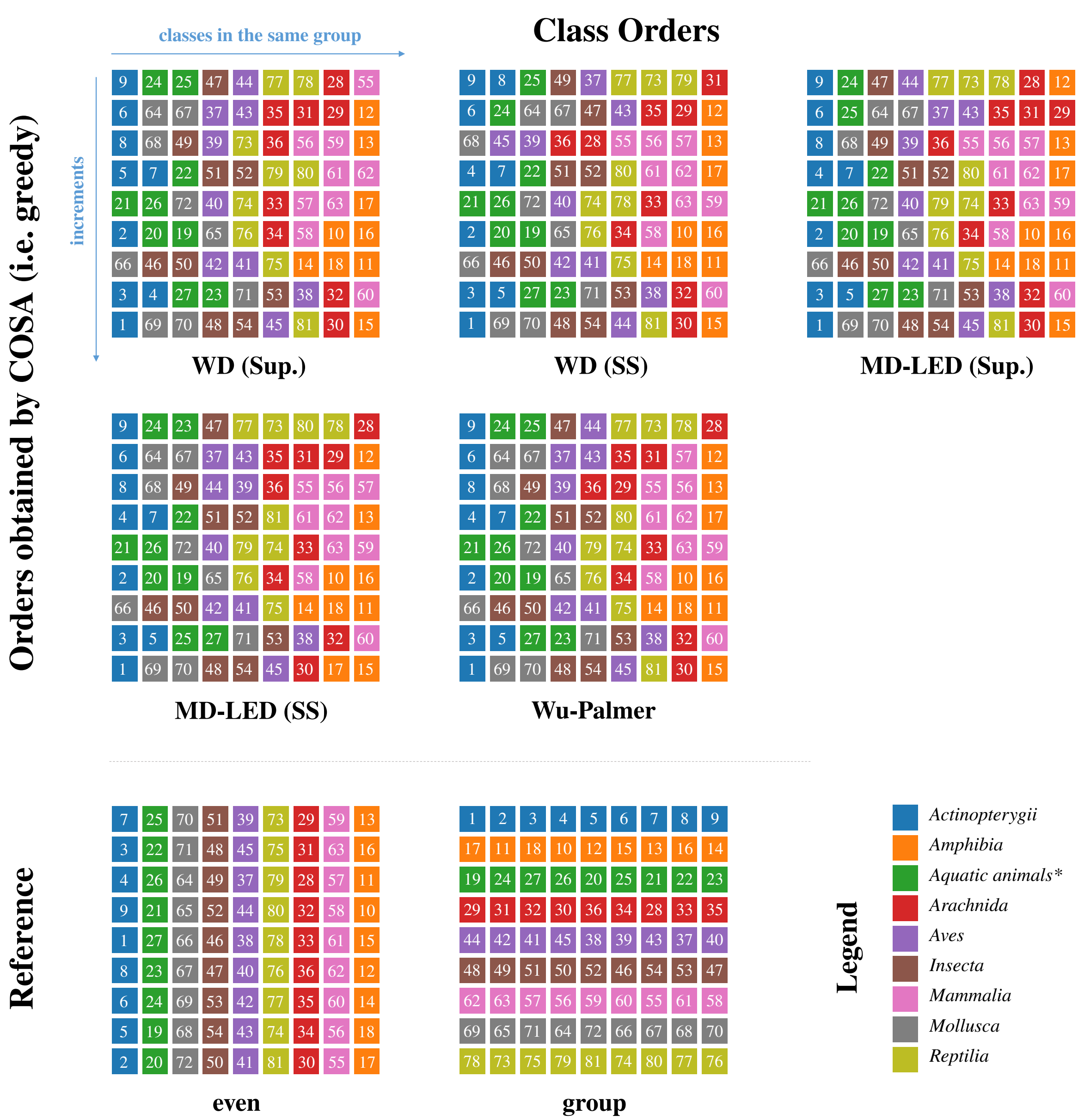

Fig. 2. Visualization of the class orders obtained by COSA (i.e. *greedy*), *even* and *group* using a 9×9 incremental protocol on Group iNaturalist. The organization of the figure is similar to Fig. 1.

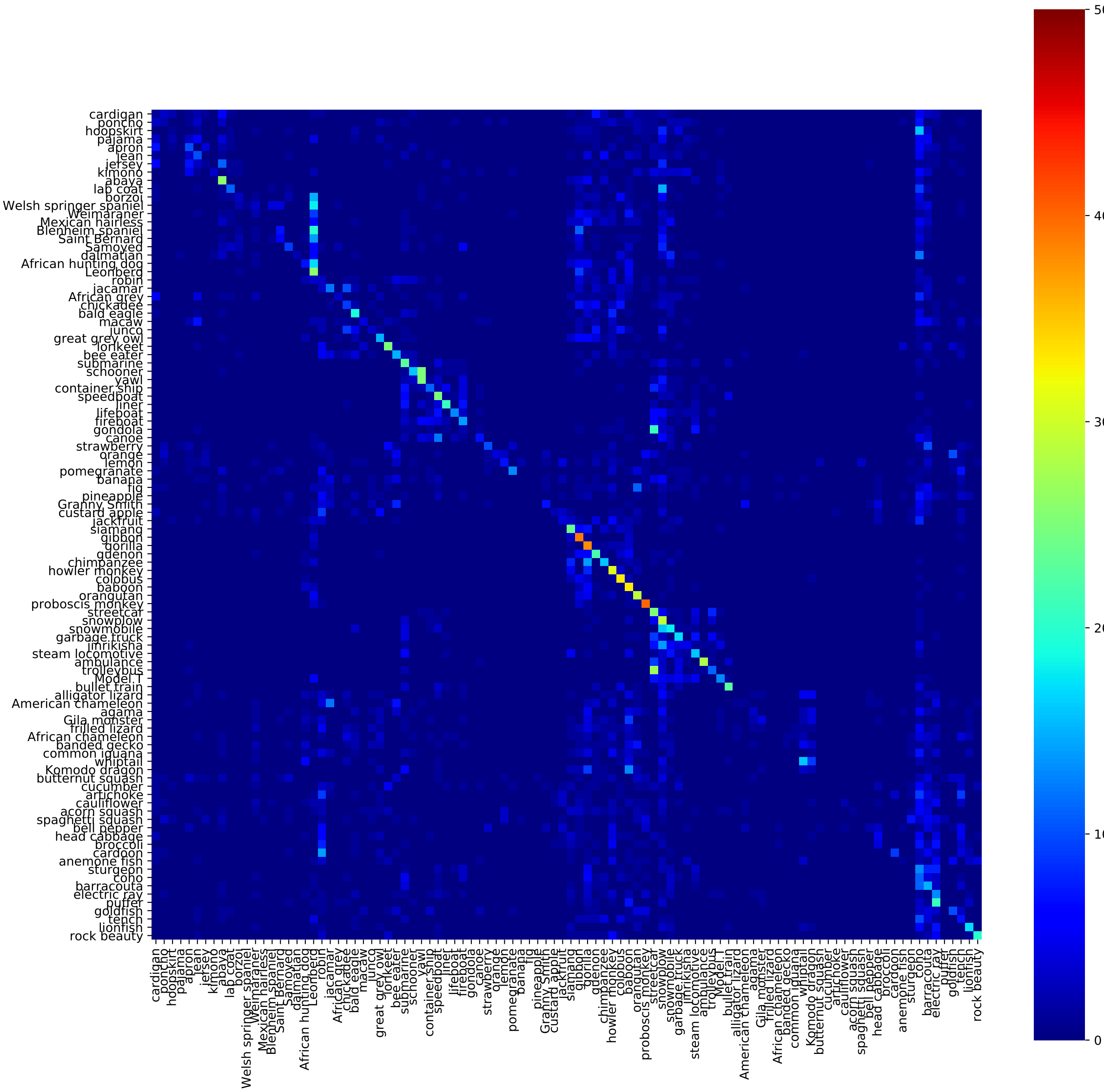

Fig. 3. Visualization of the confusion matrix of LSIL under *group* on Group ImageNet.

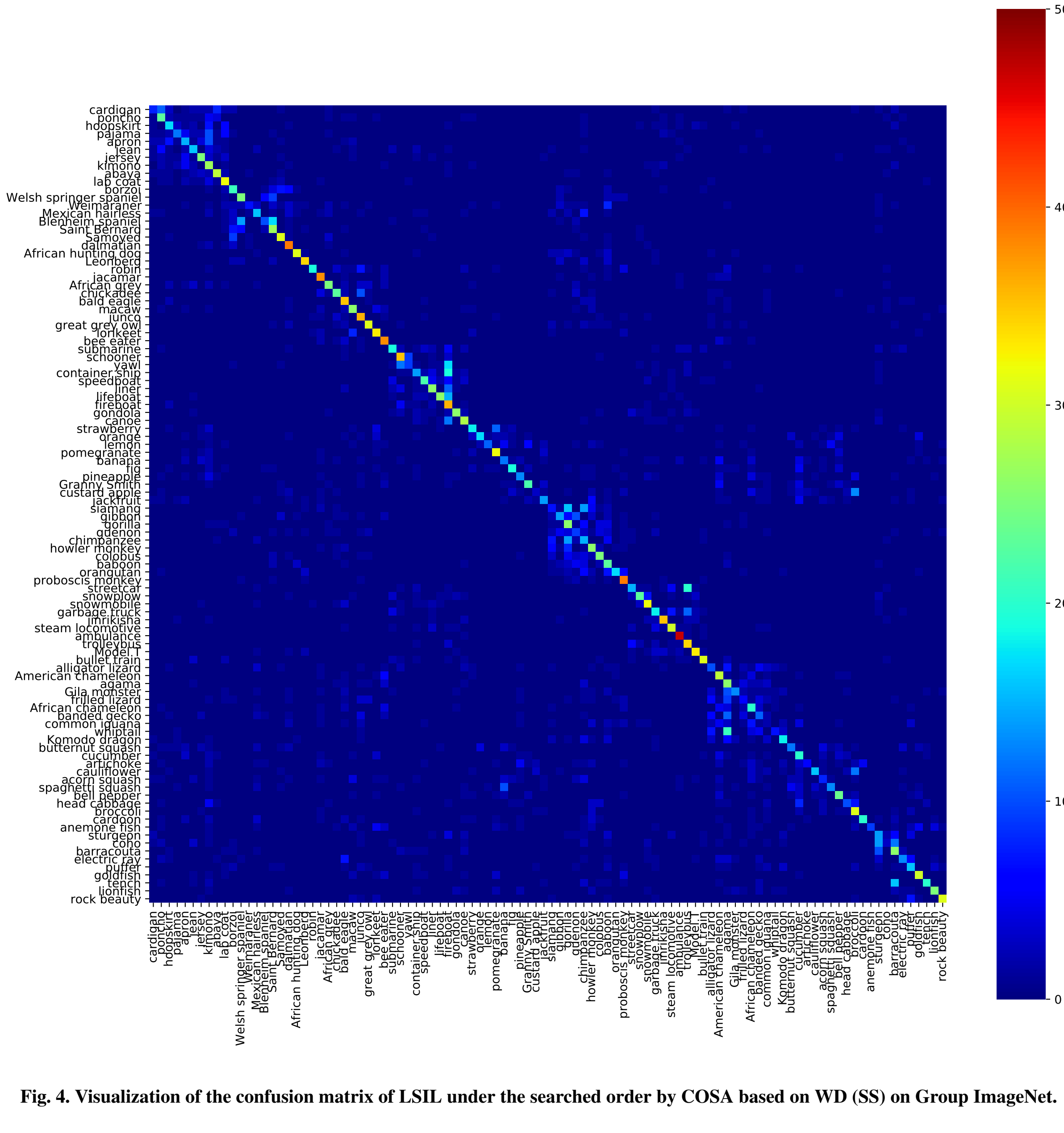

Fig. 4. Visualization of the confusion matrix of LSIL under the searched order by COSA based on WD (SS) on Group ImageNet.

Table 6. Classification accuracies (%) of methods under the searched class orders by COSA (i.e. *greedy*) based on different transferability measures on Group iNaturalist. The protocol is 9×9. The organization of the table is similar to Table 3.

Method	Searched Order based on ___						Reference	
	WD (Sup.)	WD (SS)	MD-LED (Sup.)	MD-LED (SS)	Wu-Palmer	average	even	group
iCaRL [1]	36.26	35.06	35.73	36.79	36.39	36.05	36.07	33.39
EEIL [15]	33.09	35.47	33.90	34.59	34.99	34.41	35.35	33.43
LSIL [16]	29.40	27.70	24.44	29.04	26.63	27.44	29.12	27.12
IL2M [17]	30.00	28.61	28.23	30.85	29.39	29.42	30.37	29.56
WA [18]	36.13	35.90	35.42	38.33	37.02	36.56	35.88	34.51
Post-scaling [2]	37.00	35.86	36.55	35.82	38.24	36.69	36.57	35.03

Table 7. Classification accuracies (%) of methods under the searched class orders by COSA (i.e. *greedy*) based on different transferability measures on Group ImageNet. The protocol is 27×3. The organization of the table is similar to Table 3.

Method	Searched Order based on ___						Reference	
	WD (Sup.)	WD (SS)	MD-LED (Sup.)	MD-LED (SS)	Wu-Palmer	average	even	group
iCaRL [1]	43.78	41.83	41.01	43.83	41.42	42.37	42.95	40.15
EEIL [15]	46.57	45.90	45.75	45.07	44.47	45.55	46.07	45.30
LSIL [16]	40.44	39.36	38.90	38.13	38.52	39.07	39.05	37.74
IL2M [17]	41.75	41.47	41.36	42.22	40.29	41.42	40.74	41.18
MDFCIL [18]	46.23	44.32	43.76	44.62	44.17	44.62	44.98	44.17
Post-scaling [2]	47.10	46.51	46.50	47.14	45.95	46.64	47.04	45.84

can decide which exemplars should be stored, and which exemplars can be deleted to make room for future classes. The preliminary experiment that shows the potential of COSA in exemplar management is displayed in the supplementary material. Specifically, we perform the *reverse greedy search* on the samples of each class in the current feature space, which is implemented by iteratively leveraging the k-medoids [7] algorithm to select the most representative images. The results of using our selection strategy (denoted as *greedy*) and the prevalent *random* selection which is also used throughout the main paper are shown in Table 8. From the table, it can be seen that *greedy* generally outperforms *random* selection, which shows its potential in exemplar management.

References

[1] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, C. H. Lampert, iCaRL: Incremental classifier and representation learning, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2001–2010.

[2] C. He, R. Wang, X. Chen, A tale of two CILs: The connections between class incremental learning and class imbalanced learning, and beyond, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshop on Continual Learning, 2021, pp. 233–248.

[3] G. Van Horn, O. Mac Aodha, Y. Song, Y. Cui, C. Sun, A. Shepard, H. Adam, P. Perona, S. Belongie, The iNaturalist species classification and detection dataset, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 8769–8778.

[4] G. A. Miller, WordNet: a lexical database for english, Communications of the ACM 38 (11) (1995) 39–41.

[5] E. Loper, S. Bird, NLTK: the natural language toolkit, arXiv preprint cs/0205028 (2002).

[6] Z. Wu, M. Palmer, Verbs semantics and lexical selection, in: Annual Meeting of the Association for Computational Linguistics (ACL), 1994, pp. 133–138.

[7] A. K. Jain, R. C. Dubes, Algorithms for Clustering Data, Prentice-Hall, Inc., 1988.

[8] A. V. Novikov, Pyclustering: Data mining library, Journal of Open Source Software 4 (36) (2019) 1230.

[9] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. A. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, X. Zheng, Tensorflow: A system for large-scale machine learning, in: 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI), 2016, pp. 265–283.

[10] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.

[11] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: International Conference on Learning Representations (ICLR), 2015.

[12] K. Javed, F. Shafait, Revisiting distillation and incremental classifier learning, in: Asian Conference on Computer Vision (ACCV), Springer, 2018, pp. 3–17.

[13] Y. Wu, Y. Chen, L. Wang, Y. Ye, Z. Liu, Y. Guo, Z. Zhang, Y. Fu, Incremental classifier learning with generative adversarial networks, arXiv preprint arXiv:1802.00853 (2018).

[14] A. Chaudhry, P. K. Dokania, T. Ajanthan, P. H. Torr, Riemannian walk for incremental learning: Understanding forgetting and intransigence, in: European Conference on Computer Vision (ECCV), 2018, pp. 532–547.

[15] F. M. Castro, M. J. Marín-Jiménez, N. Guil, C. Schmid, K. Alahari, End-to-end incremental learning, in: European Conference on Computer Vision (ECCV), 2018, pp. 233–248.

[16] Y. Wu, Y. Chen, L. Wang, Y. Ye, Z. Liu, Y. Guo, Y. Fu, Large scale incremental learning, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 374–382.

[17] E. Belouadah, A. Popescu, IL2M: Class incremental learning with dual memory, in: IEEE International Conference on Computer Vision (ICCV), 2019, pp. 583–592.

[18] B. Zhao, X. Xiao, G. Gan, B. Zhang, S.-T. Xia, Maintaining discrimination and fairness in class incremental learning, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 13208–13217.

[19] A. Achille, M. Lam, R. Tewari, A. Ravichandran, S. Maji, C. C. Fowlkes, S. Soatto, P. Perona, Task2Vec: Task embedding for meta-learning, in: IEEE International Conference on Computer Vision (ICCV), 2019, pp. 6430–6439.

[20] J. Civera, D. Gálvez-López, L. Riazuelo, J. D. Tardós, J. M. M. Montiel, Towards semantic SLAM using a monocular camera, in: IEEE/RSJ

Table 8. Classification accuracies (%) of methods on CIFAR-100 [23] and Group ImageNet using different exemplar selection strategies and a 10×10 incremental protocol. Each result is averaged by 3 different random class orders. Greedy is our selection strategy introduced in this paper.

Method	CIFAR-100		Group ImageNet	
	random	greedy	random	greedy
iCaRL [1]	35.10 ± 0.76	**36.80 ± 0.59**	41.78 ± 0.88	**42.20 ± 0.76**
EEIL [15]	37.61 ± 0.19	**39.45 ± 0.97**	46.15 ± 1.58	**47.07 ± 1.02**
LSIL [16]	39.04 ± 0.98	**41.48 ± 0.95**	37.30 ± 3.56	**39.52 ± 3.08**
IL2M [17]	26.71 ± 1.99	**27.94 ± 1.67**	36.06 ± 0.73	**36.69 ± 0.74**
WA [18]	39.94 ± 0.59	**40.62 ± 0.61**	46.92 ± 1.04	**47.38 ± 0.77**
Post-scaling [2]	38.68 ± 0.39	**40.59 ± 0.42**	47.92 ± 1.18	**48.52 ± 0.81**

International Conference on Intelligent Robots and Systems, 2011, pp. 1277–1284.

[21] S. L. Bowman, N. Atanasov, K. Daniilidis, G. J. Pappas, Probabilistic data association for semantic SLAM, in: IEEE International Conference on Robotics and Automation (ICRA), 2017, pp. 1722–1729.

[22] O. Stretcu, E. A. Platanios, T. M. Mitchell, B. Póczos, Coarse-to-fine curriculum learning, in: International Conference on Learning Representations (ICLR) Workshop, 2020.

[23] A. Krizhevsky, G. Hinton, Learning multiple layers of features from tiny images (2009).