Supplementary of Balancing between Forgetting and Acquisition in Incremental Subpopulation Learning

Mingfu Liang[1], Jiahuan Zhou[2,(✉)], Wei Wei[1], and Ying Wu[1]

[1] Northwestern University
{mingfuliang2020,weiwei2022}@u.northwestern.edu, yingwu@northwestern.edu
[2] Peking University
jiahuanzhou@pku.edu.cn

Abstract. In this supplementary, we provide more details, analyses and discussions about our proposed method and also other comparison methods for the incremental subpopulation learning (ISL). As mentioned in our main paper's Fig. 2, we first provide more concrete examples to illustrate the difference between our proposed ISL and the Incremental Domain Learning (IDL) in Section A. Then in Section B, we discuss the potential limitations of our proposed method and also provide corresponding empirical exploration. In Section C, we provide more empirical analyses under different network structures and also the smallest dataset from the BREEDS datasets [34] to explore how our method behaves under different scenarios. We also provide the statistical analysis between our proposed proxy of forgetting estimation and the actual forgetting in Section C.3. In Section D, we provide more discussions about our two-stage method. In Section E, we include more discussions about the existing methods. In Section F, we provide the complete dataset description and statistics for constructing our ISL benchmark. In Section G, we report all the experimental details for benchmarking all compared methods in ISL. Finally, in Section H, we include more discussions of the related works about the related incremental learning settings like continual domain adaptation (CDA) [40] and Incremental Implicitly-Refined Classification (IIRC) [1], and also the generalized boosting theory [32,31]. The variance of Tab. 1 and 2 in our main paper are reported in Tab. 4 and 5, and the corresponding discussions are in Section E.3.

A More concrete examples about the difference between ISL and IDL.

In this section, we provide the detailed illustration of our proposed ISL and the IDL (includes New Instance and Continual Domain Adaptation settings) based on the datasets used in each setting.

Continual Domain Adaptation (CDA). As shown in Fig. 1 (A), in CDA, the new data distribution introduced in the new visual domains (e.g., paintings and cartoons styles) are only the manipulation of the existing subpopulations'

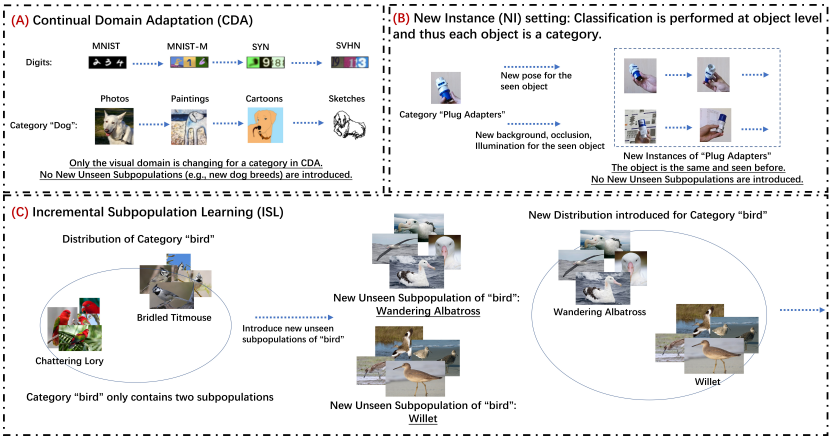

Fig. 1. Concrete Example of Incremental Subpopulation Learning (ISL) and Incremental Domain Learning (includes New Instance setting and Continual Domain Adaptation). (A) illustrates what the new distribution is introduced in Continual Domain Adaptation, and the illustration images come from [40]. (B) illustrates what exactly the new instances are introduced in the NI setting when the classification is performed at object level, and we provide the illustration of the NI setting at category level in Fig. 2 (A). (C) illustrates what unseen subpoopulations are introduced in ISL.

distribution within a category (e.g., changing the same subpopulation of "dog" like "Terriers" from photo-style to cartoon-style), but no new unseen subpopulations (e.g., new dog breeds) are introduced. The images come from the digit datasets [16,6,27] and PACS [17] dataset, and all these datasets are the benchmarks for CDA in [40].

New Instance (NI). The NI setting is first introduced in [21] for continual object recognition. The author also proposed a dataset called CORe50, which has 50 domestic objects with respect to 10 categories and each category has 5 distinct objects. As stated in Section 3 in [21], in the NI setting, we can perform the classification at object level, where each object is a category. And the classification can also be performed at category level, where each category contains 5 distinct objects in CORe50. Thus we explicitly show the NI setting at object level and category level in Fig 1 (B) and 2 (A) respectively, based on [21] and the official code[3] for the NI setting using the CORe50.

From the Fig. 1 (B), for the NI setting at object level, we can see that the new instances of the category (object) "Plug Adapters" still contain the same object seen before, while only the new pose or new environmental conditions of this seen object are introduced. In such a NI setting, we can not even create the subpopulation as defined in Fig. 1 in our main paper, and no unseen subpopulations are introduced. When the NI setting is at category level, then the category "Plug Adapters" is now comprised of 5 distinct objects (as shown in Fig. 2 (A)), and NI introduces new poses or new conditions for each seen object within this category. Again, all the objects in the new instances for a category

[3] https://github.com/vlomonaco/core50

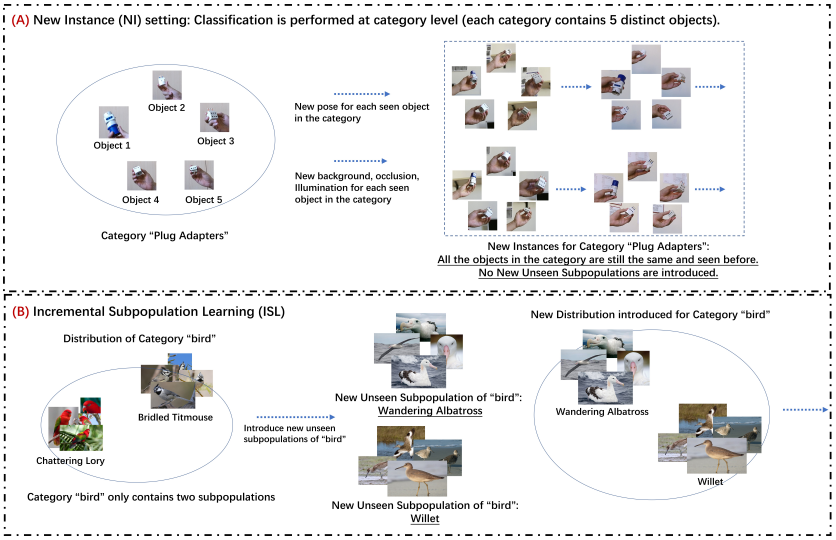

Fig. 2. Concrete Example of Incremental Subpopulation Learning (ISL) and New Instance setting (at category level). As stated in [21], for the New Instance (NI) setting, the classification can also be performed at category level, thus we introduce the concrete example of NI at the category level, as shown in (A). And we again compare it concretely with our proposed ISL, as shown in (B).

are still seen before and no unseen subpopulations are introduced. Thus, the NI setting at both levels do not introduce any new unseen subpopulations.

Our proposed ISL. As shown in Fig. 1 (C) and 2 (B), in ISL, we introduce *new subpopulations that are strictly unseen before*. This can not be satisfied in CDA or NI as described above. Each subpopulation in ISL is a distribution with sufficient variation, e.g., covers thousands of distinct objects in the nature world. Such a large variation in each subpopulation, and the large inter-subclass variance between each subpopulation, make the ISL much more challenging than CDA and NI, which also has been demonstrated in our main paper Section 4.1 based on the observations of our empirical results. Note that in ISL, all the subpopulations are under the same visual domain, i.e., the natural image.

B Discussions of Limitations

B.1 Feature Extractor Sharing

The first potential limitation comes from the feature extractor sharing. In our proposed method, we explore the possibility of sharing the feature extractor for ISL. As the feature extractor is learned in the *base step* and then is fixed and sharing during the *incremental steps*, one potential concern is that the model's performance may be limited and influenced by the *base step* training. For example, the size of the training dataset of the *base step* and also the size of network structure may potentially influence the capacity of the learned feature extractor.

Thus as mentioned in our main paper's Section 4.2, in Section C.1, with respect to the dataset size, we empirically explore our method under the smallest dataset of the BREEDS datasets [34], i.e., Living-17, using the same network structure as in our main paper, i.e., the ResNet-18 [9], to provide comparisons between existing methods. Then for the network structure, in Section C.2, we provide the results on the Entity-13 and Entity-30 under the ResNet-50 [9], which is much larger than the ResNet-18 used in our main paper. **We observe that our method can still consistently perform well under smaller dataset, smaller incremental steps, and also under larger network structure compared to existing methods. These empirical analyses relieve our potential concerns of feature extractor sharing, and also inspire our discussion of the stability and plasticity trade-off for ISL in Section E.2.** We believe that our proposed method could be a good baseline tailored to ISL based on the empirical results from both our main paper and the supplementary. We call our method a baseline since there will be much better approaches proposed for ISL where they can handle the stability and plasticity trade-off better than ours in the future, e.g., without feature extractor sharing. And our exploration will also be beneficial to the future study in ISL.

B.2 Prototype Storage

The second potential limitation may come from the prototype storage. Although the mean feature prototypes do not introduce privacy concern, the storage of them should still be discussed. Note that we have the same cost of prototype storage as the state-of-the-art (SOTA) non-exemplar-based (NEB) method, the PASS [48], as mentioned in our main paper Section 3.2. Formally, for both our method and the PASS, given the size of label space C, suppose in the t-th incremental step we introduce unseen subclasses to k_t classes and $k_t \leq C$, then we need to store k_t prototypes after training of the t-th step. This implies the worse case of the storage cost after t incremental steps is $t * C$.

Now we confirm the exact size of the mean feature prototype by saving them on the hard disk and also by observing the actual GPU memory usage of them. We find that: (1) *The storage of the prototypes in the hard disk is much smaller than saving the previous images.* For instance, we and the PASS use less than 250 MB to store all the prototypes after 13 incremental steps for the Entity-13, while for the common exemplar-based method [30,44,11], they needs more than 3 GB to store the exemplars from the previous training images of the learned subpopulations (i.e., 260 subclasses). (2) *The PASS will need increasingly larger GPU memory than ours when the incremental step is increasing.* As mentioned in our main paper Section 3.2, in our Stage-2 we do not need to do any back-propagation, thus we can search α_t directly in the CPU to avoid increasing GPU memory. However, for the PASS [48], it needs to load the prototypes to GPU to train on each incremental step since they need the prototypes to calculate the training objective function and then do error backpropagation.

Therefore, the cost of storage and GPU memory is mostly acceptable for our method in ISL. Compared to the SOTA NEB method, the PASS [48], our method

Table 1. Results on Living-17 dataset under ResNet-18. Smaller $\mathbb{F}_i$ and larger *Unseen*/*All* is better. Before incremental learning, "*Unseen*" is 60.00 for all the methods.

	2 Steps (Even Update)		
Method	*Unseen*	*All*	$\mathbb{F}_2$
Oracle	94.53	94.32	-
Finetune All	50.06	42.53	71.56
Finetune Last	46.94	44.29	63.23
EWC [15]	57.29	50.62	58.09
LwF [18]	71.82	70.94	30.26
LwF-MC [30]	75.59	72.44	28.18
MUC [20]	74.17	74.41	24.79
LwM [4]	73.06	73.32	24.26
PASS [48]	72.35	74.32	28.32
Ours	**76.29**	**80.47**	**4.5**

will be more beneficial to the edge-device where the GPU computation memory is limited, since we can avoid increasing GPU memory while the PASS can not. Moreover, in the future, we will explore the possibility of integrating different mean prototypes of the same class obtained from different incremental steps into only one class mean prototype, e.g., the moving average of those prototypes, such that the storage of our proposed method can be a fixed size C. The reason we did not explore this alternative in present paper is that as an initial attempt for ISL, we want to explore the complete power of our proposed method. The naive moving average of different mean feature prototypes of the same class may provide inaccurate estimation of the distance distortion in our Stage-2, and it will further lead to wrong estimation of the forgetting in the long run.

C More Analyses of Our Proposed Method

As mentioned in our main paper's Section 4.2, here we explore different factors that may influence our method and provide analyses of our forgetting estimation.

C.1 *Base Step* Training under Smaller Dataset

Now we investigate how the training in the *base step* may influence our proposed method. As the feature extractor is learned in the *base step*, and in our main paper and the BREEDS [34], the Entity-30 and Entity-13 have large amounts of training data in the *base step*, hence we may concern whether our method can perform consistently when the training data is small in the *base step*. So we use the smallest dataset, Living-17, in the BREEDS datasets for experiments. The Living-17 has 17 classes and each class has 4 subclasses. Again we follow the same dataset split in the BREEDS benchmark [34], where we randomly split the Living-17 into two splits. In the first split, it contains 2 subclasses with only 1300 images for each class, and we use this split for training the *base step* such that we can simulate the scenario where each class only comprises limited data in the *base step*. Then for the second split, since there are only 2 subclasses left for each class, thus we create a 2 Steps ISL protocol: in each incremental step, we introduce each class with 1 unseen subclass. This is an **even update**.

From Tab. 1, we observe that the conclusion is consistent with the ones in large datasets, i.e., the Entity-30 and Entity-13 in our main paper. Thus we

Table 2. Results on Entity-30 benchmark under ResNet-50. Smaller $\mathbb{F}_i$ and larger *Unseen*/*All* is better. Before incremental learning, "*Unseen*" is 51.40 for all the methods.

	4 Steps (Even Update)			8 Steps (Uneven Update)			15 Steps (Uneven Update)		
Method	*Unseen*	*All*	$\mathbb{F}_4$	*Unseen*	*All*	$\mathbb{F}_8$	*Unseen*	*All*	$\mathbb{F}_{15}$
Oracle	89.55	89.43	-	89.55	89.43	-	89.55	89.43	-
Finetune All	54.12	48.82	48.39	27.87	24.11	73.64	14.93	13.73	84.77
Finetune Last	55.52	60.58	29.28	29.80	32.72	59.51	19.68	21.96	71.32
EWC [15]	58.92	56.24	39.47	31.40	29.21	68.26	20.52	19.98	78.36
LwF [18]	62.88	56.95	35.39	34.88	29.65	64.98	33.58	31.25	61.83
LwF-MC [30]	**68.12**	66.15	27.69	48.45	45.10	49.92	35.87	34.93	61.77
MUC [20]	63.75	59.66	30.44	38.23	32.90	60.67	37.35	35.26	57.23
LwM [4]	62.33	57.75	35.58	34.92	29.97	60.19	33.50	30.71	61.89
PASS [48]	66.83	70.95	20.31	47.38	51.46	44.46	41.73	36.27	60.89
Ours	64.25	**73.95**	**3.39**	**58.07**	**72.46**	**3.68**	**56.23**	**71.62**	**4.99**

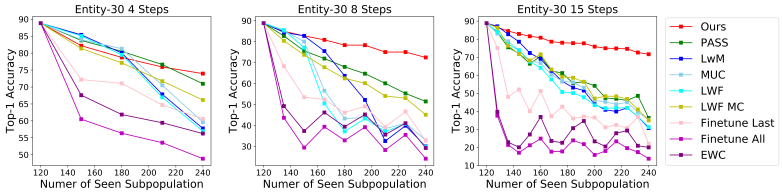

Fig. 3. Average top-1 test accuracy of each step under 3 protocols of Entity-30 using ResNet-50.

believe that our method may consistently perform well under different size of training data in the *base step*. We conjecture that the reason why the feature extractor is so powerful is that: the heavy data augmentation we used as in the BREEDS [34] may prevent the overfitting on the small size of data and encourage the feature extractor to learn to extract discriminative features for each class.

C.2 Network Structure

We choose a larger network structure, i.e., the ResNet-50, and conduct experiments under the same protocols on the Entity-30 and Entity-13, as shown in Tab. 2 and 3 and also Fig. 3 and 4. We observe the same conclusions under the ResNet-50 as under the ResNet-18 in our main paper: our method can consistently and significantly outperform the existing NEB methods for ISL under challenging and sufficiently long protocols. Thus we further believe that our proposed method can be a good baseline for ISL to alleviate the subpopulation shifting problem under different network structure.

C.3 Statistical Analysis between the Proposed Proxy of Forgetting Estimation and the Actual Forgetting

We are also curious about whether the proposed proxy of forgetting estimation $l_{\text{dist}}(\alpha_t)$ (Eqn. 9 in our main paper) is statistically related to the actual forgetting on the seen population. $l_{\text{dist}}(\alpha_t)$ is defined as the relative distance distortion of the class representative prototype between the last step's classifier $G_{\phi_{t-1}}$ and new classifier $G_{\phi'_t}$ under different α_t. The actual forgetting $AF(\alpha_t)$ on the seen

Table 3. Results on Entity-13 benchmark under ResNet-50. Smaller $\mathbb{F}_i$ and larger *Unseen*/*All* is better. Before incremental learning, "*Unseen*" is 63.65 for all the methods.

	5 Steps (Even Update)			10 Steps (Even Update)			13 Steps (Uneven Update)		
Method	*Unseen*	*All*	$\mathbb{F}_5$	*Unseen*	*All*	$\mathbb{F}_{10}$	*Unseen*	*All*	$\mathbb{F}_{13}$
Oracle	91.63	91.54	-	91.63	91.54	-	91.63	91.54	-
Finetune All	60.41	60.25	38.02	52.45	53.06	46.33	43.49	44.79	55.83
Finetune Last	68.45	75.20	14.56	65.00	72.03	20.42	53.01	59.01	36.22
EWC [15]	63.20	63.03	34.02	57.03	58.40	40.64	46.97	48.58	51.21
LwF [18]	66.94	65.53	31.43	59.54	58.91	37.58	51.18	51.11	46.61
LwF-MC [30]	69.94	68.61	28.35	56.75	57.47	41.19	59.98	60.93	37.05
MUC [20]	65.92	65.42	32.06	61.37	61.07	33.67	53.18	52.68	44.56
LwM [4]	68.55	66.92	30.15	62.69	61.88	32.14	54.54	53.34	41.81
PASS [48]	**74.66**	76.76	15.62	67.48	69.98	24.66	51.46	54.11	43.63
Ours	73.08	**79.93**	**2.94**	**71.08**	**78.54**	**2.00**	**70.34**	**79.22**	**3.28**

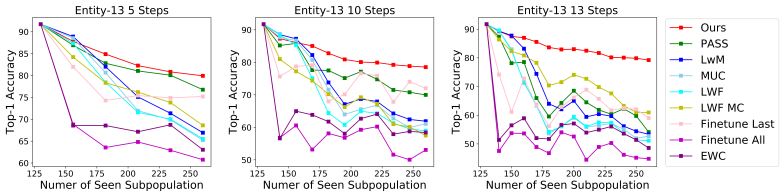

Fig. 4. Average top-1 test accuracy of each step under 3 protocols of Entity-13 using ResNet-50.

population is measured by the held-out test set of all the previous incremental step using the classifier $G_{\phi_t} = G_{\phi_{t-1}} + \alpha_t \cdot G_{\phi'_t}$ under different α_t.

We leverage the Spearman's rank correlation coefficient[4] [22] (Spearman Correlation) to assess the relationship between the proxy estimation and the actual forgetting. Spearman Correlation is a nonparametric measure of rank correlation between two variables. Here we take 13 Steps Entity-13 as an example and we calculate the Spearman score of the $l_{\text{dist}}(\alpha_t)$ and $AF(\alpha_t)$ after each incremental step. For 13 Steps Entity-13, we obtain the average Spearman correlation score $\rho = 0.9525$ over 13 incremental steps, which implies that the proposed proxy of the forgetting has a strong statistical correlation with the actual forgetting. The same observations can also be obtained from different datasets and protocols.

D More Discussions of Our Two-Stage Method

D.1 Illustration of the Relationship between $l_{dist}(\alpha_t)$ and $l_{val}(\alpha_t)$

To better illustrate the relationship between the $l_{dist}(\alpha_t)$ and $l_{val}(\alpha_t)$ in Eqn. 10 in our main paper, we give some representative samples shown in Fig. 5 with detailed descriptions. We can observe that when the α_t becomes larger as shown by the blue points from the left to the right in each image, the relative improvement of the current step's validation accuracy $l_{val}(\alpha_t)$ for the unseen subpopulation is increasing, while the relative distance distortion $l_{dist}(\alpha_t)$ is also increasing to reflect the increasingly large forgetting on the seen population approximately.

[4] https://en.wikipedia.org/wiki/Spearman%27s_rank_correlation_coefficient

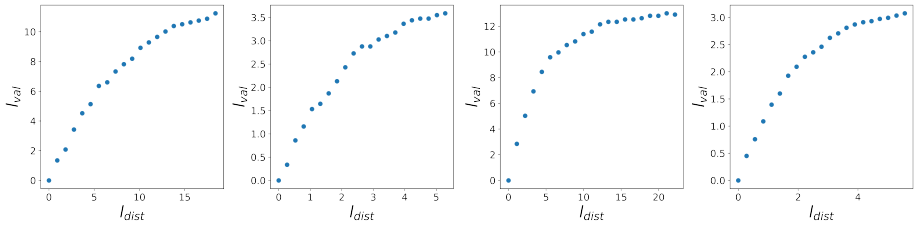

Fig. 5. Representative examples to demonstrate the relationship between $l_{dist}(\alpha_t)$ and $l_{val}(\alpha_t)$. From the left to the right, each image represents the relationship between $l_{dist}(\alpha_t)$ and $l_{val}(\alpha_t)$ on four steps randomly sampled from the 13 Steps Entity-13. In each image, the blue point from the left to the right represents that the α_t increases from 0 to 1 with the interval of 0.05. When the α_t is increasing, the relative improvement of the current step's validation accuracy $l_{val}(\alpha_t)$ may also increase since the classifier $G_{\phi_t} = G_{\phi_{t-1}} + \alpha_t \cdot G_{\phi'_t}$ is now dominated by the new classifier $G_{\phi'_t}$; Meanwhile, the relative distance distortion $l_{dist}(\alpha_t)$ may also become larger since the classifier G_{ϕ_t} is biasing to the unseen subpopulation, which approximately implies the larger forgetting on the seen population. The same phenomenon can be observed in all the experimental protocols under the Entity-13, Entity-30 and Living-17 datasets.

Both of them empirically show the challenge of balancing acquisition and forgetting in ISL. These empirical observations further verify our design of $l_{dist}(\alpha_t)$ and $l_{val}(\alpha_t)$ in that they can approximately model the balance between the acquisition and forgetting in ISL. Note that we also observe the same phenomenon in different protocols under different datasets.

D.2 Empirical Comparison between the Cross Entropy Loss and Our Proposed Stage-1

Now we further empirically analyze the difference between the cross entropy loss with softmax, and our proposed Stage-1 where the softmax layer is also incorporated. To better illustrate the difference, we randomly choose 1 class in the Living-17 dataset over 17 classes, and we only learn 1 unseen subclass for this specific class in an incremental step. After training for one incremental step, we compare the L1-norm of the weight difference of the last linear layer, i.e., the classifier, before and after training based on the cross entropy loss with softmax and our proposed Stage-1 incorporated with softmax layer, respectively.

The result is shown in Fig. 6 with detailed descriptions. We observe that due to the two reweighting mechanisms in our proposed Stage-1, we can much better avoid largely changing other class classifiers in this extreme **uneven update** for ISL, shown in the second-left figure in Fig. 6. This also leads to much smaller forgetting of the seen population than the cross entropy loss, shown in the rightmost figure in Fig. 6. Therefore, the empirical results further verify the formal discussion in Section 3.2 in our main paper that the reweighting mechanism may alleviate unnecessary updating for the classifier, such that we can efficiently acquire the unseen subpopulation and also alleviate the unnecessary

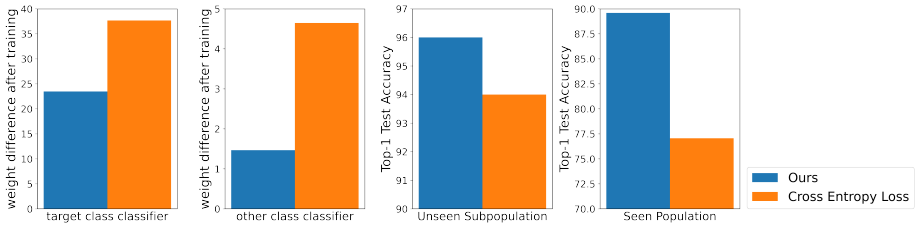

Fig. 6. Representative examples to demonstrate how the classifier's behavior is related to the balance between the acquisition of unseen subpopulationes and the forgetting of the seen population. **In general, we expect the model can use only the small but necessary update to learn the unseen subpopulation to avoid bringing much more forgetting on the seen population.** When we only introduce one unseen subpopulation to a specific class, i.e., the target class, both the cross entropy loss and our proposed Stage-1 can have a sufficient update for the target class's classifier to learn its unseen subpopulation, shown in the leftmost and second-right figures. However, the cross entropy loss updates all other class classifiers with a much larger scale than ours, i.e., 4x of ours, shown in the second-left figure. **This leads to more than 12% of test accuracy drop on the seen population than ours**, shown in the rightmost figure. When the incremental step increases, the forgetting of the seen population may further aggravate under the cross entropy loss.

forgetting on the seen population, with the same spirit as the Occam's Razor[5]. Note that in this experiment, our Stage-1 is also incorporated with the softmax layer after the last classifier layer to obtain the model prediction and then we optimize the Eqn. 7 in our main paper. Therefore, our proposed Stage-1 can also further alleviate the adverse effect brought by the softmax layer compared to the cross entropy. Thus we also incorporate the softmax layer after the last classifier layer in our Stage-1 for all of our experiments in our main paper and supplementary so that we can provide consistent comparison to the cross entropy loss. The softmax layer does not change the prediction from the linear classifier since the softmax function only normalizes the prediction to probability, thus our method's prediction is still the same and consistent after the softmax.

D.3 Implementation Details about the Stage-2

Here we provide the implementation details of the Stage-2 in Section 3.2 of our main paper. As we mentioned in our main paper's line 431-432, the α_t can be searched by the simple line search on the objective function l_α as the Eqn. 10 in our main paper. In practice, we empirically observe that the proper α_t mostly lays in the range of $(0, 2]$, and thus to speed up the searching procedure, we discretize the above range with the interval of 0.05 to readily find the proper α_t for all the experiments. In most cases, $l_{val}(\alpha_t)$ and $l_{dist}(\alpha_t)$ are under the same scale, as shown in Fig. 5. And when their scales are mismatched, we will rescale the smaller term such that both of them are under the same scale to have the

[5] https://en.wikipedia.org/wiki/Occam%27s_razor

same importance in Eqn. 10. The re-scaling is: before optimizing the l_α, we first sample several α_t from the range of $(0, 2]$ with an interval of 0.05, then we obtain a list of ratios by the division of $l_{val}(\alpha_t)$ and $l_{dist}(\alpha_t)$ under different α_t. We rescale the smaller term with the largest ratio to avoid scale mismatch and then optimize the l_α. For more details, please refer to our official code[6]. This works smoothly in all of our experiments and provides robust scaling of $l_{val}(\alpha_t)$ and $l_{dist}(\alpha_t)$ to ensure that the solution of l_α can properly balance the forgetting and acquisition. For the held-out validation set D_t^{val}, we randomly sample 10% of the current step's training data D_t^{train} as the D_t^{val}.

E More Discussions of the Existing Methods

E.1 About the Objective Function for Acquisition in ISL

LwF-MC [30] use the binary cross entropy loss with sigmoid instead of the cross entropy loss with softmax for other existing methods. The sigmoid operator treats each class's prediction to be separated in the last linear layer but without decoupling the feature representation learning as described in [30], and the update of each class classifier may also be separated. This may be beneficial when we have **uneven update** in ISL. For example, we can observe from Tab. 1 and 2 in our main paper that LwF-MC mostly outperforms other compared method under **uneven update** (e.g., 13 Steps Entity-13).

However, completely separate each class may also bring other adverse effects since we can also observe that in some of the protocols with **even update** in Tab. 1 and 2 in our main paper and the Tab. 2 and 3 in the supplementary, the LwF-MC can not achieve the best overall performance over other existing NEB methods. Our proposed method is better than LwF-MC since without separating each class, the reweighting mechanism in our Stage-1 can explicitly emphasize the hard sample and class to effectively learn the new unseen subpopulation while also intrinsically defy the forgetting.

E.2 About the Stability and Plasticity Trade-off in ISL

In our proposed incremental subpopulation learning (ISL) setting, given our specific target of the subpopulation shifting problem, if we can have a proper design to achieve a better stability and plasticity trade-off, e.g., our proposed two-stage model, then freezing and sharing the feature extractor of the CNN may be reasonable and beneficial to ISL. Firstly, Santurkar et al. [34] found that only finetuning the last layer, i.e., the classifier, can largely reduce the performance drop on the unseen subpopulation compared to other methods (see lines 703-709 in Section H.1). This inspires us to conjecture the reason of the subpopulation shifting problem: the feature extractor learned on seen population can extract discriminative features for each class, but the classifier may have

[6] https://github.com/wuyujack/ISL

Table 4. Results on Entity-30 benchmark under ResNet-18 with standard deviation under shuffling of the incremental steps' order. Smaller $\mathbb{F}_i$ and larger *Unseen*/*All* is better. Before incremental learning, "*Unseen*" is 50.18±1.06 for all the methods.

	4-Steps (Even Update)			8 Steps (Uneven Update)			15 Steps (Uneven Update)		
Method	*Unseen*	*All*	$\mathbb{F}_4$	*Unseen*	*All*	$\mathbb{F}_8$	*Unseen*	*All*	$\mathbb{F}_{15}$
Finetune All	53.72±0.50	48.08±1.19	47.75±2.50	26.45±0.63	23.08±0.47	73.86±5.87	14.68±5.44	13.77±5.71	84.49±6.72
Finetune Last	55.25±0.20	58.30±0.27	32.43±1.19	30.85±4.32	32.50±0.26	60.82±4.19	19.98±5.59	21.56±4.35	72.40±6.55
EWC [15]	56.17±0.40	54.10±0.18	40.69±0.16	30.50±2.52	29.00±2.50	66.94±3.87	22.20±4.71	23.68±5.03	74.03±5.37
LwF [18]	62.67±0.05	58.85±1.11	32.32±0.72	34.52±0.10	29.69±1.06	64.38±2.79	32.62±6.43	31.17±4.97	62.51±6.16
LwF-MC [30]	**68.28±1.08**	64.43±1.34	28.20±0.68	46.93±2.25	43.69±2.50	50.88±0.92	34.53±5.97	33.79±5.52	62.36±5.18
MUC [20]	62.98±0.24	59.59±0.22	29.45±0.75	36.17±3.21	31.83±3.05	61.49±1.09	34.15±4.47	32.54±4.48	60.65±6.19
LwM [4]	63.32±0.18	59.20±0.92	33.13±0.88	42.47±5.15	38.90±4.87	55.59±0.45	33.43±5.20	30.78±5.47	61.23±5.26
PASS [48]	64.50±0.52	69.37±0.40	21.79±0.71	48.85±1.70	54.99±2.56	40.50±5.57	32.13±5.92	39.75±4.12	58.27±3.72
Ours	64.73±2.21	**72.88±0.11**	**4.16±0.22**	**58.63±0.92**	**72.14±0.39**	**2.30±0.13**	**56.87±0.27**	**71.69±0.41**	**3.48±0.28**

biased to the seen subpopulations and emphasize the features that are less discriminative for the unseen subpopulations. Thus the model may misclassify the unseen subpopulations. This is reasonable since the CNN can learn the generic and discriminative feature extractor [45,47,36,5] to readily perform the transfer learning [45,5], while the CNN can also be easily biased by the learning data. **However, only finetuning the last layer can not achieve the balance between the acquisition of unseen subpopulation and the forgetting of seen population in the long run in ISL**, as demonstrated in our main paper. *Thus we need a better design to exploit the benefits from the feature extractor, while we also need to balance the forgetting and acquisition in ISL.* This is exactly the motivation to design our method: in Stage-1 we enforce the model to reduce the prediction error progressively such that we can effectively learn the unseen subpopulation to alleviate the stability concern; in Stage-2 we explicitly disentangle the forgetting and acquisition to achieve a better balance of them. Note that our method is tailored to ISL since in other IL settings like CIL, the unseen new classes can be totally different from the old ones and the old feature extractor may not be able to extract the discriminative features for the new classes without training on them.

E.3 More discussions of our empirical results

As shown in Tab. 4 and 5, we observe that given each protocol, our proposed method can consistently perform well under different orders of the incremental steps for learning the unseen subpopulation with small variance. Note that in present paper we shuffle the incremental step's index to provide the analysis of order shuffling since we only have limited GPU resources to conduct the experiments. In the future we will further explore shuffling all the subclasses order in each incremental step and provide more comprehensive analysis.

E.4 Discussion of LwF-like methods

The superiority of the LwF-like methods can not be consistently maintained in the long run for ISL. From Fig. 4 and 5 in our main paper, we observe that both the LwF [18] and its variants (LwM [4] and MUC [20]) may have higher average accuracy than other methods in the early steps. However, their performance degrades significantly in longer steps and **uneven update** (e.g., 8 and 15 Steps

Table 5. Results on Entity-13 benchmark under ResNet-18 with standard deviation under shuffling of the incremental steps' order. Smaller $\mathbb{F}_i$ and larger *Unseen*/*All* is better. Before incremental learning, "*Unseen*" is 62.03±1.32 for all the methods.

	5 Steps (Even Update)			10 Steps (Even Update)			13 Steps (Uneven Update)		
Method	*Unseen*	*All*	$\mathbb{F}_5$	*Unseen*	*All*	$\mathbb{F}_{10}$	*Unseen*	*All*	$\mathbb{F}_{13}$
Finetune All	61.54±2.71	59.16±1.89	37.79±3.91	51.55±3.31	50.88±1.38	46.76±3.32	41.98±0.11	41.72±0.57	56.97±0.10
Finetune Last	65.52±1.94	71.15±0.36	18.89±2.60	61.52±0.11	67.37±3.17	25.47±1.29	49.89±2.42	55.23±2.79	40.31±0.60
EWC [15]	63.85±3.44	63.48±2.23	32.99±3.67	55.63±0.17	57.31±3.46	36.53±2.86	47.51±0.75	48.54±0.82	50.49±2.14
LwF [18]	66.91±2.34	64.82±2.06	31.47±3.69	59.97±1.05	59.17±1.76	36.26±1.06	51.14±0.57	51.05±0.81	46.31±3.90
LwF-MC [30]	67.57±2.24	65.96±2.52	30.64±5.41	59.58±3.55	59.22±0.24	38.42±1.93	59.45±1.40	59.70±0.84	37.02±2.54
MUC [20]	67.51±2.01	65.88±2.91	30.00±5.70	62.17±3.21	61.98±0.52	31.45±1.25	53.58±2.24	52.89±2.57	43.74±4.70
LwM [4]	69.69±1.05	67.61±1.20	28.22±3.50	63.49±1.03	62.25±0.31	31.72±0.61	51.05±0.15	50.80±0.70	46.31±4.51
PASS [48]	**73.12±1.05**	75.44±1.03	16.73±2.75	65.63±0.14	68.51±1.53	26.55±1.35	50.48±3.26	52.49±3.63	43.76±4.83
Ours	72.02±1.08	**78.92±0.20**	**3.29±0.50**	**68.31±0.72**	**77.53±0.16**	**3.35±0.38**	**69.69±1.14**	**78.75±0.26**	**3.35±0.41**

Entity-30). This shows that the LwF-like method can not consistently strike a great balance between acquisition and forgetting in the long run in ISL, as also demonstrated in our main paper.

F Dataset Description and Statistics

Here we describe our dataset choice and statistics in details. Our dataset choice is based on the BREEDS datasets proposed in [34] recently. The BREEDS datasets are designed to precisely simulate the real-world subpopulation shifting and they are constructed based on the ImageNet [3]. The creation of BREEDS is first roughly splitting the classes and their subclasses in the ImageNet [3] based on the WordNet semantic hierarchy. Then the author recruited a large amount of human annotators to precisely examine whether these subclasses images are visually coherent to their corresponding classes and sharing similar visual characteristics. They also largely edit the ImageNet dataset to fulfill the requirement. This is essential to synthesize subpopulation shifting since we can not expect a model to generalize well on arbitrary subpopulations.

The BREEDS datasets comprise 4 datasets, i.e., Entity-13, Entity-30, Living-17, Non-Living-26, with a total of 0.86 millions (M) of images. We leverage the Entity-13 and Entity-30 in our main paper and Living-17 in our supplementary. The dataset statistics are shown in Tab. 6. We follow the same dataset splitting as BREEDS that we randomly separate the Entity-13, Entity-30 and Living-17 into two splits respectively, i.e., the source and target splits, where each split has equal number of subclasses for each class. The source split is used for the *base step* training. Then we further separate the target split into different *incremental steps* based on the number of subclasses to create our ISL protocols. The protocols' details are stated in our main paper's Section 4. Note that all the splits are generated by the same random seed used in the BREEDS. For the Entity-13, the association between the classes and their subclasses is presented in Tab. 9; For the Entity-30, the association is presented in Tab. 10 and 11. Note that in different ISL protocols of the Entity-13 and Entity-30, we split the "Unseen Subclasses in *Incremental Steps*" (shown in Tab. 9, 10 and 11) as mentioned in the Section 4 of our main paper for each incremental step. For the Living-17 used in our supplementary, we also provide the details in Tab. 12.

Table 6. Dataset statistics for each dataset used in our paper. For the **Incremental Steps**, we report the total number of images over all the *incremental steps*.

	Base Step		**Incremental Steps**	
Dataset	Train	Test	Train	Test
Entity-30	154263	6000	153565	6000
Entity-13	167120	6500	167592	6500
Living-17	44200	1700	44200	1700

G Experimental Details

In this section we provide the complete experimental details that we use to create the ISL benchmark. As mentioned in our main paper, we leverage the recent proposed Continual Hyperparameter Framework (CHF) in [2] as a standard to choose the *general training hyperparameters* for all the methods, and also the *specific model hyperparameters* for the comparison methods. The *general training hyperparameters* include training epoch of each incremental steps, initial learning rate, weight decay and momentum for all the methods. The *specific model hyperparameters* for the comparison methods are used to balance the forgetting and acquisition, and we will describe these hyperparameters for each comparison method later. For our method, since the proper α_t is searched by optimizing the Eqn. 10 in our main paper, thus we do not need to introduce extra *specific model hyperparameters* to control the forgetting and acquisition.

G.1 Continual Hyperparameter Framework (CHF)

The CHF uses only the training data of each incremental step to determine the hyperparameters for general incremental learning since in real-world application we can not access even the hold-out test set in each previous incremental step. CHF can avoid being over-optimistic of a method's performance and also provide a fair comparison to all compared methods. CHF has also been used in a recent large-scale empirical survey for class incremental learning (CIL) in [24].

G.2 Workflow of the CHF

The workflow of the CHF [2] comprises two phases in each incremental step: (1) we first finetune a copy of the last step's model on the unseen subpopulation data. The learning rate is obtained by a coarse grid search that aims for high accuracy on the held-out validation set of the current step's training data. (2) Then in the second stage, the model begins to train with the searched learning rate on the current step's training data. We first set the *specific model hyperparameters* to be maximum such that the forgetting of the seen population should be minimum. We also define a threshold p to indicate the maximum drop of the current step's validation accuracy compared to the accuracy we obtained by finetuning in the first phase. If the model can not achieve the validation accuracy higher than $1-p$ of the finetuning accuracy, then we decrease the *specific model hyperparameters* with a decay ratio β until the model can meet our goal.

Table 7. *Base step*'s training details for each dataset under ResNet-18 and ResNet-50 in our paper for all the comparison methods and our method.

Base Step's Training Details	**Entity-13**	**Entity-30**	**Living-17**
Learning Rate (LR)	0.1	0.1	0.1
Training Epoch	300	300	450
Batch Size	128	128	128
Weight Decay	1e-4	1e-4	1e-4
10-fold LR Drop	Every 100 epoch	Every 100 epoch	Every 150 epoch
Data Augmentation	The same as the BREEDS [34] benchmark		

G.3 Training details of the *base step*.

As mentioned in Section F, the BREEDS benchmark simulates the subpopulation shifting by splitting each dataset into the source and target splits, and we choose the source split for the *base step* training. We follow the same training details in BREEDS [34] based on their official GitHub repository[7] such that the empirical results of the incremental subpopulation learning (ISL) in our paper can be directly compared with the results in [34] without ISL. And we can also explore whether the ISL may alleviate the subpopulation shifting problem. The training details are reported in Tab. 7. The data augmentation comprises random resize crop (224x224), random horizontal flip, lighting, and color jitter, etc. All of them are the same as the BREEDS [34]. We reproduce the BREEDS benchmark [34] based on the ImageNet training code from [19,14], where we obtain a very close or even the same results under ResNet-18 and ResNet-50 [9], and observe the same subpopulation shifting problem as in [34].

G.4 Training details of the *incremental steps*.

All the methods (i.e., all the comparison methods and our method) are initialized with the same *base step's model* and then start incremental learning for a fair comparison. We follow the CHF [2] and a recent systematic empirical benchmark [24] to search the learning rate in range of {0.1, 0.05, 0.01, 0.005} given that the initial learning rate of the *base step* is 0.1. The threshold p is set to 0.2 and the decay ratio β is 0.5 as the common usage in CHF [2,24]. The training epoch for each *incremental step* is 20 for all the methods, as it is enough for fine-tuning a previous model on the unseen subpopulation to achieve around 95% top-1 accuracy on the current step's validation set (held out from the training set). The batch size and data augmentation in the *incremental step* are the same as the ones in the *base step* for all the methods. We use SGD with the momentum as 0.9, the weight decay as 1e-4 and the constant learning rate scheduler for all the method given that the incremental training epoch is relatively small. All the code is implemented in PyTorch [28]. The existing methods are implemented based on their official implementation and also based on the large and public GitHub repository[8] proposed by [24] and the code for our proposed method is

[7] https://github.com/MadryLab/BREEDS-Benchmarks
[8] https://github.com/mmasana/FACIL

release in our official GitHub repository[9]. Based on the CHF, the searched initial learning rate is 0.005 for each *incremental step* of the Entity-30 and the Entity-13 for the compared method, and 0.01 for the Living-17. For our method, the CHF searched result is 0.05, 0.01, 0.005 for 4, 8 and 15 Steps Entity-30 respectively and 0.1, 0.05, 0.05 for 5, 10, 13 Steps Entity-13. We can see that under some protocols the CHF searched initial learning rate appears relatively large for our proposed method. This is due to the methodology difference between the compared methods and our proposed two-stage method that disentangles the acquisition and forgetting into two stages separately: For the compared methods, they couple the acquisition and forgetting in a single objective function and optimize them simultaneously, and thus the choice of the learning rate will both influence the acquisition of unseen subpopulation and forgetting on seen population. Therefore, the CHF will search for relatively small learning rate for the compared methods to avoid aggravating the forgetting such that they have a better balance between acquisition and forgetting. While for our method, the learning rate is **only** used in the "gain-acquisition" stage, i.e., Stage-1, to progressively acquire the new subpopulation, and in our Stage-2 we do not perform any learning for the unseen subpopulation, as stated in our main paper Section 3.2. Since in Stage-1 our ultimate goal is to progressively acquire the unseen subpopulations, thus the CHF will search for relatively large learning rate to ensure that the unseen subpopulation can be acquire as good as possible in Stage-1 without concerning any forgetting on the seen population. However, we also need to note that although our method seems to be able to be trained with relatively large learning rate in our Stage-1 training to acquire the unseen subpopulation, this **does not** mean that our final performance on the unseen subpopulation will always be better than the compared methods. **This is because our final performance on the unseen subpopulation is further controlled on our Stage-2 and if in Stage-2 the searched α_t prefers to maintain more on the seen population's performance, then the full acquisition of the unseen subpopulation from Stage-1 will not be preserved after the linear combination in Stage-2**. We could also exactly observe that under some protocols, i.e., 4 Steps Entity-30 and 5 Steps Entity-13, our method **does not** have the largest acquisition ("*Unseen*") on the unseen subpopulation compared to other methods, although our "*All*" performance is better only because we could achieve a better balance between forgetting and acquisition than other compared methods. This implies that although our Stage-2 can achieve a better balance between the acquisition and forgetting, this balance may still be sacrificing some of the acquisition on the unseen subpopulation achieved in our Stage-1 training. **This further shows that there is still a large room for us to improve our Stage-2 to achieve much better balance between forgetting and acquisition, and thus we view our method as only a baseline method instead of a good-enough method for the ISL.**

Now we describe the maximum of the *specific model hyperparameters* for each comparison method to be decreased in the CHF, which is mostly based on [2,24]:

[9] https://github.com/wuyujack/ISL

EWC [15]: We follow the [24] to fuse the old and new importance weights by 0.5 to avoid the storage of the importance weights for each incremental steps. The loss function is combined with the cross entropy loss with softmax and the EWC loss for regularizing the forgetting based on the Fisher Information Matrix, where the balance between the cross entropy loss and the EWC loss is by a hyperparameter on the EWC loss, starting from 1000.

LwF [18]: The loss function of LwF comprises the cross entropy loss with softmax and the knowledge distillation loss with a temperature scaling parameter. The knowledge distillation loss is for the forgetting regularization. We follow the same implementation from [24,2] to implement the LwF and we fixed the temperature scaling parameters to 2 as proposed in [18] and used in most of the literature. The balancing of the acquisition and forgetting is also controlled by a hyperparameter on the distillation loss, starting from 40.

LwF-MC [30]: The LwF-MC is proposed in [30] as an alternative of the iCaRL [30] but without storing the previous training images. The loss function of LwF-MC comprises the binary cross entropy loss with sigmoid and a distillation loss [30] for the forgetting regularization. The hyperparameter of the distillation loss starts from 10.

MUC [20]: We follow the official implementation of the MUC [20] [10]. Based on the existing literature [20,48], we use the strong version of MUC, i.e., with the LwF. MUC is a variant of the LwF by adding multiple classifiers during each *incremental step*'s training and encouraging those classifiers to have a large discrepancy such that the LwF can perform better. The discrepancy is enforced based on another unlabeled out-of-distribution (OOD) dataset [20] which is different from the training data of each incremental step. Thus the loss function comprises the cross entropy loss, the distillation loss and a discrepancy loss. We follow the default usage of the SVHN [20] dataset as the unlabeled OOD dataset and the same training details as the official implementation of the discrepancy loss. Then the *model specific hyperparameter* is on the knowledge distillation loss, which starts from 40.

LwM [4]: We follow the same implementation from [24] as there does not exist official implementation for LwM. LwM is also a variant of LwF and its loss function consists of the cross entropy loss, distillation loss from LwF and the LwM loss, where the LwM loss is a forgetting regularization based on the intermediate feature visualization of the old and new models. Thus we have two hyperparameters on the distillation loss and LwM loss respectively, where the former starts from 40 and the latter starts from 2 as the default usage in [24].

PASS [48]: PASS is the recent SOTA non-exampler-based (NEB) method for the CIL proposed in [48]. We follow the official implementation from the author[11]. The loss function of PASS consists of the cross entropy loss, the knowledge distillation loss and the prototype augmentation loss, where the last two term is for the forgetting regularization and they both start from 40 such that we can also cover the default choice in the official implementation.

[10] https://github.com/liuyudut/MUC
[11] https://github.com/Impression2805/CVPR21_PASS

Table 8. Difference between our ISL and other IL settings.

	Training Supervision Needs		New Data in Incremental Learning with	
IL Settings	category label	subpopulation label	New Category	New Subpopulation
ISL	Yes	No	No	Yes (Strictly)
IDL [39,21,40]	Yes	No	No	No
CIL [2]	Yes	No	Yes	No
IIRC [1]	Yes	Yes	Yes	Yes (Not Strictly, with also seen subpopulation)

Note that all these comparison methods are general for incremental learning and they can be readily used for the ISL without bells and whistles. For the naive baselines, i.e., the "Finetune Last" and "Finetune All", they do not introduce any *specific model hyperparameters* to balance the forgetting and acquisition. Both the "Finetune Last" and "Finetune All" are optimized with the cross entropy loss with softmax. For our proposed method, as the proper α_t is searched by optimizing the Eqn. 10 in our main paper, thus we do not need to introduce extra *specific model hyperparameters* to control the forgetting and acquisition.

H More Discussions of Related Works

H.1 Related Incremental Learning Settings

Here we provide more discussions about the related IL settings like the continual domain adaptation (CDA) [40] and Incremental Implicitly-Refined Classification (IIRC) [1]. Tab. 8 provides the concrete difference between our ISL and other IL settings discussed in our main paper's Section 2.

More Comparisons to the CDA [40]. The existing state-of-the-art method [40] for the CDA proposes using diverse and heavy data augmentation for randomizing the domain distribution to make the model inherently robust against forgetting and generalize to new domain. However, this strategy can not alleviate the subpopulation shifting problem and the forgetting issue in the ISL, which coincides with the empirical finding in [34]. In the BREEDS benchmarks, Santurkar et. al. [34] found that the subpopulation shifting can not be alleviated by using strong data augmentation [46] (like Gaussian and Erase noise), adversarial training [23] or even training with a stylized version of ImageNet to encourage the model to rely more on shape rather than texture [7]. However, these robust interventions techniques are commonly used to effectively perform the domain adaptation [38,10] (DA) and domain generalization [41] (DR).

The reason why those effective techniques in DA and DR can not alleviate the subpopulation shifting is due to the difference between the CDA and ISL: The unseen subpopulation is both semantically and visually coherent with the seen population under the same visual domain (i.e., natural image), thus they may have a similar form of representations in the feature space. However, for the CDA, there are different visual domains (e.g., photo and cartoon styles) introduced in each incremental step, and the representations in each domain are intrinsically different and they need to be appropriately aligned to perform

the domain adaptation [37,43,33]. Moreover, the distribution of the unseen subpopulation can not be explicitly simulated by neither strong and diverse data augmentation nor the adversarial training used in both CDA and domain adaptation and generalization. Thus, those effective methods in CDA and domain adaptation may not alleviate the subpopulation shifting problem.

Therefore, the different sources of shifting, i.e., the domain shifting and the subpopulation shifting, differentiate the CDA and ISL.

More Comparisons to the IIRC [1] In IIRC's [1] original paper, the author did not introduce a specific method for the IIRC setting to explicitly learn the association of different label hierarchies, i.e., the coarse and fine labels, during the IIRC. During the IIRC, the model needs to do multi-label prediction for each image, and even with specific adaptation, the existing exemplar-based method, e.g., the iCaRL [30] and LUCIR [11], can still not perform well in the IIRC.

Differently, in our work we further propose a new method as the first and a good baseline tailored to the ISL. We also believe that studying this well-defined and isolated shifting, i.e., the subpopulation shifting, in incremental learning can facilitate clean analysis and provide much more insights for our specific concern.

H.2 General Boosting Theory [32,31]

Here we provide more discussions about the general boosting theory [32,31]. The general aspect of the boosting with respect to the gradient descent was first proposed in [25]. The gradient descent view of the boosting is general [35] and can theoretically unify the existing boosting algorithms by the functional gradient techniques [35]. Saberian et al. [32,31] further established the multi-class boosting theory under the general gradient descent aspect of the boosting. Besides the theoretical development of the boosting theory, the general idea of boosting or ensembling are largely leveraged in existing works [42,12,13] in computer vision, but few of them explicitly built their methods based on the general boosting theory. Moghimi et al. [26] first proposed to leverage the multi-class boosting theory to learn many different CNNs, e.g., VGG-16, for the image classification task and achieved better result than a single CNN. Han et. al. [8] proposed to incrementally boost the CNN for facial action unit recognition, but they are not doing the incremental learning since in [8] they train the model with all the data and thus there is no forgetting. Pentina et. al. [29] theoretically studied the lifelong learning with weighted majority votes of different learned predictors, which has similar spirit with the boosting mechanism of model selection. However, Pentina et. al. [29] did not provide any practical algorithm to implement their theoretical analysis. To the best of our knowledge, we are the first work to connect the generalized boosting theory with the incremental learning and also show that it is desirable tailored to the incremental subpopulaiton learning (ISL) given that in the ISL, we do not need to extend the classifier and thus we can incrementally learning a new classifier to replace the old one by the functional gradient techniques of boosting, without adding many new CNNs. We also propose a new mechanism, i.e., the Stage-2, to achieve the balance between the

acquisition of the unseen population and the forgetting of the seen population tailored to the ISL.

References

1. Abdelsalam, M., Faramarzi, M., Sodhani, S., Chandar, S.: Iirc: Incremental implicitly-refined classification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11038–11047 (2021) 1, 17, 18
2. Delange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G., Tuytelaars, T.: A continual learning survey: Defying forgetting in classification tasks. IEEE Transactions on Pattern Analysis and Machine Intelligence (2021) 13, 14, 15, 16, 17
3. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255. IEEE (2009) 12
4. Dhar, P., Singh, R.V., Peng, K.C., Wu, Z., Chellappa, R.: Learning without memorizing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5138–5146 (2019) 5, 6, 7, 11, 12, 16
5. Dhillon, G.S., Chaudhari, P., Ravichandran, A., Soatto, S.: A baseline for few-shot image classification. In: International Conference on Learning Representations (2020), https://openreview.net/forum?id=rylXBkrYDS 11
6. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: International Conference on Machine Learning. pp. 1180–1189. PMLR (2015) 2
7. Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F.A., Brendel, W.: Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In: International Conference on Learning Representations (2019), https://openreview.net/forum?id=Bygh9j09KX 17
8. Han, S., Meng, Z., Khan, A.S., Tong, Y.: Incremental boosting convolutional neural network for facial action unit recognition. Advances in Neural Information Processing Systems **29**, 109–117 (2016) 18
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016) 4, 14
10. Hoffman, J., Tzeng, E., Park, T., Zhu, J.Y., Isola, P., Saenko, K., Efros, A., Darrell, T.: CyCADA: Cycle-consistent adversarial domain adaptation. In: Dy, J., Krause, A. (eds.) Proceedings of the 35th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 80, pp. 1989–1998. PMLR (10–15 Jul 2018), https://proceedings.mlr.press/v80/hoffman18a.html 17
11. Hou, S., Pan, X., Loy, C.C., Wang, Z., Lin, D.: Learning a unified classifier incrementally via rebalancing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 831–839 (2019) 4, 18
12. Huang, G., Li, Y., Pleiss, G., Liu, Z., Hopcroft, J.E., Weinberger, K.Q.: Snapshot ensembles: Train 1, get m for free. International Conference on Learning Representations (2017) 18
13. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4700–4708 (2017) 18

14. Huang, Z., Liang, S., Liang, M., Yang, H.: Dianet: Dense-and-implicit attention network. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 4206–4214 (2020) 14
15. Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al.: Overcoming catastrophic forgetting in neural networks. Proceedings of the National Academy of Sciences **114**(13), 3521–3526 (2017) 5, 6, 7, 11, 12, 16
16. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998) 2
17. Li, D., Yang, Y., Song, Y.Z., Hospedales, T.M.: Deeper, broader and artier domain generalization. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 5542–5550 (2017) 2
18. Li, Z., Hoiem, D.: Learning without forgetting. IEEE Transactions on Pattern Analysis and Machine Intelligence **40**(12), 2935–2947 (2017) 5, 6, 7, 11, 12, 16
19. Liang, S., Huang, Z., Liang, M., Yang, H.: Instance enhancement batch normalization: An adaptive regulator of batch noise. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 4819–4827 (2020) 14
20. Liu, Y., Parisot, S., Slabaugh, G., Jia, X., Leonardis, A., Tuytelaars, T.: More classifiers, less forgetting: A generic multi-classifier paradigm for incremental learning. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVI 16. pp. 699–716. Springer (2020) 5, 6, 7, 11, 12, 16
21. Lomonaco, V., Maltoni, D.: Core50: a new dataset and benchmark for continuous object recognition. In: Conference on Robot Learning. pp. 17–26. PMLR (2017) 2, 3, 17
22. Lyerly, S.B.: The average spearman rank correlation coefficient. Psychometrika **17**(4), 421–428 (1952) 7
23. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: International Conference on Learning Representations (2018) 17
24. Masana, M., Liu, X., Twardowski, B., Menta, M., Bagdanov, A.D., van de Weijer, J.: Class-incremental learning: survey and performance evaluation on image classification. arXiv preprint arXiv:2010.15277 (2020) 13, 14, 15, 16
25. Mason, L., Baxter, J., Bartlett, P., Frean, M.: Boosting algorithms as gradient descent in function space. In: Advances in Neural Information Processing Systems. vol. 12, pp. 512–518 (1999) 18
26. Moghimi, M., Belongie, S.J., Saberian, M.J., Yang, J., Vasconcelos, N., Li, L.J.: Boosted convolutional neural networks. In: BMVC. vol. 5, p. 6 (2016) 18
27. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning. NIPS Workshop on Deep Learning and Unsupervised Feature Learning (2011) 2
28. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Advances in Neural Information Processing Systems **32**, 8026–8037 (2019) 14
29. Pentina, A., Urner, R.: Lifelong learning with weighted majority votes. Advances in Neural Information Processing Systems **29**, 3612–3620 (2016) 18
30. Rebuffi, S.A., Kolesnikov, A., Sperl, G., Lampert, C.H.: icarl: Incremental classifier and representation learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017) 4, 5, 6, 7, 10, 11, 12, 16, 18

31. Saberian, M., Vasconcelos, N.: Multiclass boosting: Margins, codewords, losses, and algorithms. Journal of Machine Learning Research **20**(137), 1–68 (2019), http://jmlr.org/papers/v20/17-137.html 1, 18
32. Saberian, M.J., Vasconcelos, N.: Multiclass boosting: Theory and algorithms. In: NIPS. pp. 2124–2132 (2011) 1, 18
33. Saito, K., Watanabe, K., Ushiku, Y., Harada, T.: Maximum classifier discrepancy for unsupervised domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018) 18
34. Santurkar, S., Tsipras, D., Madry, A.: {BREEDS}: Benchmarks for subpopulation shift. In: International Conference on Learning Representations (2021), https://openreview.net/forum?id=mQPBmvyAuk 1, 4, 5, 6, 10, 12, 14, 17
35. Schapire, R.E., Freund, Y.: Boosting: Foundations and algorithms. Kybernetes (2013) 18
36. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 618–626 (2017) 11
37. Sun, B., Saenko, K.: Deep coral: Correlation alignment for deep domain adaptation. In: European Conference on Computer Vision. pp. 443–450. Springer (2016) 18
38. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7167–7176 (2017) 17
39. Van de Ven, G.M., Tolias, A.S.: Three scenarios for continual learning. NeurIPS - Continual Learning workshop (2018) 17
40. Volpi, R., Larlus, D., Rogez, G.: Continual adaptation of visual representations via domain randomization and meta-learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4443–4453 (2021) 1, 2, 17
41. Volpi, R., Namkoong, H., Sener, O., Duchi, J.C., Murino, V., Savarese, S.: Generalizing to unseen domains via adversarial data augmentation. In: Advances in Neural Information Processing Systems (2018) 17
42. Walach, E., Wolf, L.: Learning to count with cnn boosting. In: European Conference on Computer Vision. pp. 660–676. Springer (2016) 18
43. Wang, M., Deng, W.: Deep visual domain adaptation: A survey. Neurocomputing **312**, 135–153 (2018) 18
44. Wu, Y., Chen, Y., Wang, L., Ye, Y., Liu, Z., Guo, Y., Fu, Y.: Large scale incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 374–382 (2019) 4
45. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? Advances in Neural Information Processing Systems **27**, 3320–3328 (2014) 11
46. Zhong, Z., Zheng, L., Kang, G., Li, S., Yang, Y.: Random erasing data augmentation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 13001–13008 (2020) 17
47. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2921–2929 (2016) 11
48. Zhu, F., Zhang, X.Y., Wang, C., Yin, F., Liu, C.L.: Prototype augmentation and self-supervision for incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5871–5880 (2021) 4, 5, 6, 7, 11, 12, 16

Table 9. The class and their corresponding subclasses split for incremental subpopulation learning on Entity-13. In the *base step*, each class training data comprises the data from the subclasses in the middle column accordingly. For the *incremental step*, we split the unseen subclasses in the rightmost column based on different protocols.

Class	Subclasses in *Base Step*	Unseen Subclasses in *Incremental Steps*
garment	trench coat, abaya, gown, poncho, military uniform, jersey, cloak, bikini, miniskirt, swimming trunks	lab coat, brassiere, hoopskirt, cardigan, pajama, academic gown, apron, diaper, sweatshirt, sarong
bird	African grey, bee eater, coucal, American coot, indigo bunting, king penguin, spoonbill, limpkin, quail, kite	prairie chicken, red-breasted merganser, albatross, water ouzel, goose, oystercatcher, American egret, hen, lorikeet, ruffed grouse
reptile	Gila monster, agama, triceratops, African chameleon, thunder snake, Indian cobra, green snake, mud turtle, water snake, loggerhead	sidewinder, leatherback turtle, boa constrictor, garter snake, terrapin, box turtle, ringneck snake, rock python, American chameleon, green lizard
arthropod	rock crab, black and gold garden spider, tiger beetle, black widow, barn spider, leafhopper, ground beetle, fiddler crab, bee, walking stick	cabbage butterfly, admiral, lacewing, trilobite, sulphur butterfly, cicada, garden spider, leaf beetle, long-horned beetle, fly
mammal	Siamese cat, ibex, tiger, hippopotamus, Norwegian elkhound, dugong, colobus, Samoyed, Persian cat, Irish wolfhound	English setter, llama, lesser panda, armadillo, indri, giant schnauzer, pug, Doberman, American Staffordshire terrier, beagle
accessory	bib, feather boa, stole, plastic bag, bathing cap, cowboy boot, necklace, crash helmet, gasmask, maillot	hair slide, umbrella, pickelhaube, mitten, sombrero, shower cap, sock, running shoe, mortarboard, handkerchief
craft	catamaran, speedboat, fireboat, yawl, airliner, container ship, liner, trimaran, space shuttle, aircraft carrier	schooner, gondola, canoe, wreck, warplane, balloon, submarine, pirate, lifeboat, airship
equipment	volleyball, notebook, basketball, handheld computer, tripod, projector, barbell, monitor, croquet ball, balance beam	cassette player, snorkel, horizontal bar, soccer ball, racket, baseball, joystick, microphone, tape player, reflex camera
furniture	wardrobe, toilet seat, file, mosquito net, four-poster, bassinet, chiffonier, folding chair, fire screen, shoji	studio couch, throne, crib, rocking chair, dining table, park bench, chest, window screen, medicine chest, barber chair
instrument	upright, padlock, lighter, steel drum, parking meter, cleaver, syringe, abacus, scale, corkscrew	maraca, saltshaker, magnetic compass, accordion, digital clock, screw, can opener, odometer, organ, screwdriver
man-made structure	castle, bell cote, fountain, planetarium, traffic light, breakwater, cliff dwelling, monastery, prison, water tower	suspension bridge, worm fence, turnstile, tile roof, beacon, street sign, maze, chainlink fence, bakery, drilling platform
wheeled vehicle	snowplow, trailer truck, racer, shopping cart, unicycle, motor scooter, passenger car, minibus, jeep, recreational vehicle	jinrikisha, golfcart, tow truck, ambulance, bullet train, fire engine, horse cart, streetcar, tank, Model T
produce	broccoli, corn, orange, cucumber, spaghetti squash, butternut squash, acorn squash, cauliflower, bell pepper, fig	pomegranate, mushroom, strawberry, lemon, head cabbage, Granny Smith, hip, ear, banana, artichoke

Table 10. The class and their corresponding subclasses split for incremental subpopulation learning (ISL) on Entity-30. In the *base step*, each class training data comprises the data from the subclasses in the middle column accordingly. For the *incremental step*, we split the unseen subclasses in the rightmost column based on different protocols.

Class	Subclasses in *Base Step*	Unseen Subclasses in *Incremental Steps*
serpentes	green mamba, king snake, garter snake, thunder snake	boa constrictor, green snake, ringneck snake, rock python
passerine	goldfinch, brambling, water ouzel, chickadee	magpie, house finch, indigo bunting, bulbul
saurian	alligator lizard, Gila monster, American chameleon, green lizard	Komodo dragon, African chameleon, agama, banded gecko
arachnid	harvestman, barn spider, scorpion, black widow	wolf spider, black and gold garden spider, tick, tarantula
aquatic bird	albatross, red-backed sandpiper, crane, white stork	goose, dowitcher, limpkin, drake
crustacean	crayfish, spiny lobster, hermit crab, Dungeness crab	king crab, rock crab, American lobster, fiddler crab
carnivore	Italian greyhound, black-footed ferret, Bedlington terrier, basenji	flat-coated retriever, otterhound, ShihTzu, Boston bull
insect	lacewing, fly, grasshopper, sulphur butterfly	long-horned beetle, leafhopper, dung beetle, admiral
ungulate	llama, gazelle, zebra, ox	hog, hippopotamus, hartebeest, warthog
primate	baboon, howler monkey, Madagascar cat, chimpanzee	siamang, indri, capuchin, patas
bony fish	coho, tench, lionfish, rock beauty	sturgeon, puffer, eel, gar
barrier	breakwater, picket fence, turnstile, bannister	chainlink fence, stone wall, dam, worm fence
building	bookshop, castle, mosque, butcher shop	grocery store, toyshop, palace, beacon
electronic equipment	printer, pay-phone, microphone, computer keyboard	modem, cassette player, monitor, dial telephone
footwear	clog, Loafer, maillot, running shoe	sandal, knee pad, cowboy boot, Christmas stocking

Table 11. The class and their corresponding subclasses split for incremental subpopulation learning on Entity-30. In the *base step*, each class training data comprises the data from the subclasses in the middle column accordingly. For the *incremental step*, we split the unseen subclasses in the rightmost column based on different protocols.

Class	Subclasses in *Base Step*	Unseen Subclasses in *Incremental Steps*
garment	academic gown, apron, miniskirt, fur coat	jean, vestment, sarong, swimming trunks
headdress	pickelhaube, hair slide, shower cap, bonnet	bathing cap, cowboy hat, bearskin, crash helmet
home appliance	washer, microwave, Crock Pot, vacuum	toaster, espresso maker, space heater, dishwasher
kitchen utensil	measuring cup, cleaver, coffeepot, spatula	frying pan, cocktail shaker, tray, caldron
measuring instrument	digital watch, analog clock, parking meter, magnetic compass	barometer, wall clock, hourglass, digital clock
motor vehicle	limousine, school bus, moped, convertible	trailer truck, beach wagon, police van, garbage truck
musical instrument	French horn, maraca, grand piano, upright	acoustic guitar, organ, electric guitar, violin
neckwear	feather boa, neck brace, bib, Windsor tie	necklace, stole, bow tie, bolo tie
sports equipment	ski, dumbbell, croquet ball, racket	rugby ball, balance beam, horizontal bar, tennis ball
tableware	mixing bowl, water jug, beer glass, water bottle	goblet, wine bottle, coffee mug, plate
tool	quill, combination lock, padlock, screw	fountain pen, screwdriver, shovel, torch
vessel	container ship, lifeboat, aircraft carrier, trimaran	liner, wreck, catamaran, yawl
dish	potpie, mashed potato, pizza, cheeseburger	burrito, hot pot, meat loaf, hotdog
vegetable	zucchini, cucumber, butternut squash, artichoke	cauliflower, spaghetti squash, acorn squash, cardoon
fruit	strawberry, pineapple, jackfruit, Granny Smith	buckeye, corn, ear, acorn

Table 12. The class and their corresponding subclasses split for incremental subpopulation learning on Living-17. In the *base step*, each class training data comprises the data from the subclasses in the middle column accordingly. For the *incremental step*, we split the unseen subclasses in the rightmost column based on different protocols.

Class	Subclasses in *Base Step*	Unseen Subclasses in *Incremental Steps*
salamander	eft, axolotl	common newt, spotted salamander
turtle	box turtle, leatherback turtle	loggerhead, mud turtle
lizard	whiptail, alligator lizard	African chameleon, banded gecko
snake	night snake, garter snake	sea snake, boa constrictor
spider	tarantula, black and gold garden spider	garden spider, wolf spider
grouse	ptarmigan, prairie chicken	ruffed grouse, black grouse
parrot	macaw, lorikeet	African grey, sulphur-crested cockatoo
crab	Dungeness crab, fiddler crab	rock crab, king crab
dog	bloodhound, Pekinese	Great Pyrenees, papillon
wolf	coyote, red wolf	white wolf, timber wolf
fox	grey fox, Arctic fox	red fox, kit fox
domestic cat	tiger cat, Egyptian cat	Persian cat, Siamese cat
bear	sloth bear, American black bear	ice bear, brown bear
beetle	dung beetle, rhinoceros beetle	ground beetle, long-horned beetle
butterfly	sulphur butterfly, admiral	cabbage butterfly, ringlet
ape	gibbon, orangutan	gorilla, chimpanzee
monkey	marmoset, titi	spider monkey, howler monkey