Uncertainty-Aware Hierarchical Refinement for Incremental Implicitly-Refined Classification (Supplementary Materials)

Jian Yang[1*] **Kai Zhu**[1,*,‡] **Kecheng Zheng**[2] **Yang Cao**[1,3,†]
[1] University of Science and Technology of China [2] Ant Group
[3] Institute of Artificial Intelligence, Hefei Comprehensive National Science Center
{yangjian12138@mail, zkzy@mail}.ustc.edu.cn
zkccloud@gmail.com forrest@ustc.edu.cn

A. Additional Experiments

A.1. More feature visualization for IIRC-ImageNet-lite

To better evaluate the effectiveness of the proposed hierarchical distribution alignment strategy (HDA), we visualize the feature distributions of old and new classes using t-SNE on the 0-th configuration of IIRC-ImageNet-lite. Specifically, we visualize the features of some classes (*i.e.*, kitchen appliances, curtain-screen, fungus, box, garment, pouched mammal, bus, big cat, dog, and watercraft categories) in phase 0. Meanwhile, in phase N (we set N = 8 in the Fig. 1), we visualize the features of new category (*i.e.*, truck) and some subclasses (*i.e.*, lion, papillon, standard poodle, container ship categories) of some categories of phase 0.

As shown in Fig. 1, we observe that the feature distributions of some categories using the baseline intersect each other in the 0-th phase. Thanks to the global representation extension strategy (GRE), the features extracted from our proposed method are largely separately distributed and have an obvious classification interval. In phase 8, the distribution of a subclass in baseline completely deviates from the corresponding superclass, and the distribution of the new superclass is severely mixed with the old superclass. After introducing the proposed HDA, the distribution of the increasing subclasses falls into the distribution of the corresponding superclasses, and the distribution of the new superclass is separated from that of the old superclass.

A.2. Confusion Matrix For IIRC-ImageNet-Lite

We combine the first 30 classes from phase 0, the first ten classes from phase 2, the first ten classes from phase 4, the first ten classes from phase 6, and the first 20 classes from phase 8 on IIRC-Imagenet-lite to complete the confusion matrix. As shown in Fig. 2, iCaRL-Norm presents a more severe confusion during the optimization of new classes, which is corrected by our method.

A.3. More Uncertainty Experiments

We present and compare more classical uncertainty methods to demonstrate our approach's superiority numerically. Specifically, the entropy-based uncertainty method 'mean' and energy-based uncertainty method 'std' is widely used in the OOD [1] [12] and regression task [6], respectively. The uncertainty method 'cov' based on the coefficient of variation is widely used in the segmentation domain [10] [13]. As shown in Fig. 3, We applied these uncertainty methods in the IIRC setting and found suboptimal performance due to the confusion on the hierarchical relationship, illustrating the improvement of our proposed method.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

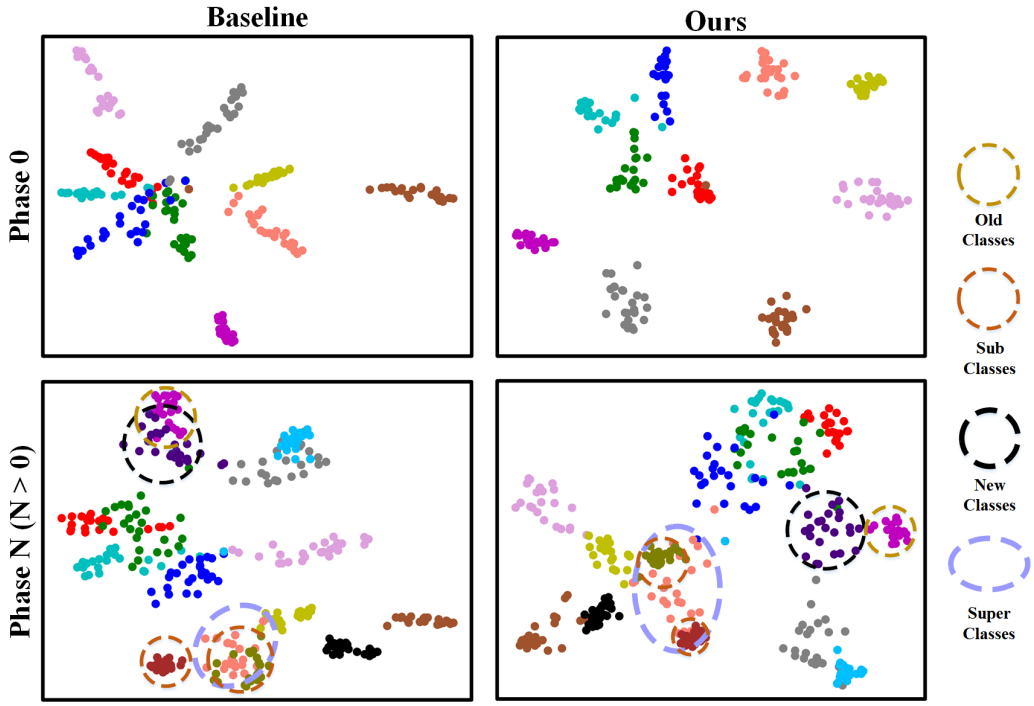
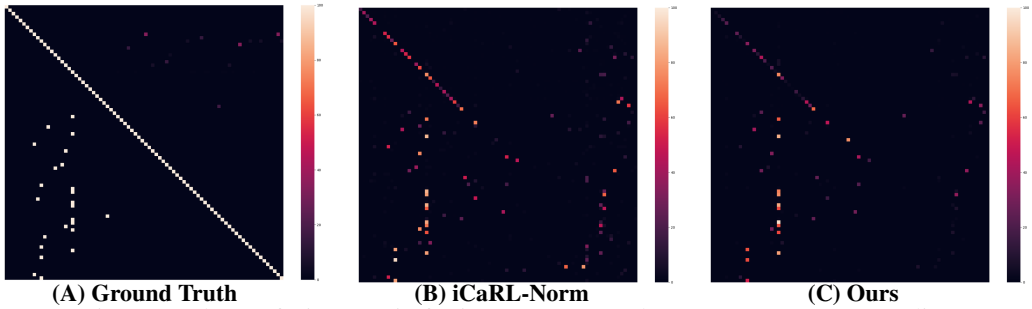


Figure 1: The impact of our method on IIRC-ImageNet-lite. (1) The upper row shows the effectiveness of global representation extension strategy that is able to extend the distribution distance between the superclasses; (2) The lower row shows the effectiveness of the hierarchical distribution alignment that separate the distribution of new superclass from that of old superclass.



(A) Ground Truth (B) iCaRL-Norm (C) Ours

Figure 2: The confusion matrix for iCaRL-Norm and ours on IIRC-ImageNet-lite.

A.4. Decision Process

We take the tulip class in phase 3 on IIRC-CIFAR as an example to visualize the decision process, as shown in Fig. 4. We assume that there are 20 samples of the tulip class in the current batch. The corresponding output on the old model is obtained for constructing cross-hierarchical label relationships, including the features and entropy values. After getting the single-label output of each sample for each phase, the occurrence frequency of each label is counted. If it is less than 10, the class is irrelevant to the tulip. If it is greater than 10, it is judged to be a relevant label.

In order to judge whether it is a superclass-subclass relationship or a brother relationship, we obtain the standard deviation of the features corresponding to the relevant labels and the standard deviation of the features of the current tulip class. The class with the smallest standard deviation distance from the tulip class is the superclass of tulip, and the rest are brother classes.

For the samples judged to be the superclass and the output entropy value less than 0.5, the output entropy will be added to a margin distance value and maintain the same entropy value for those

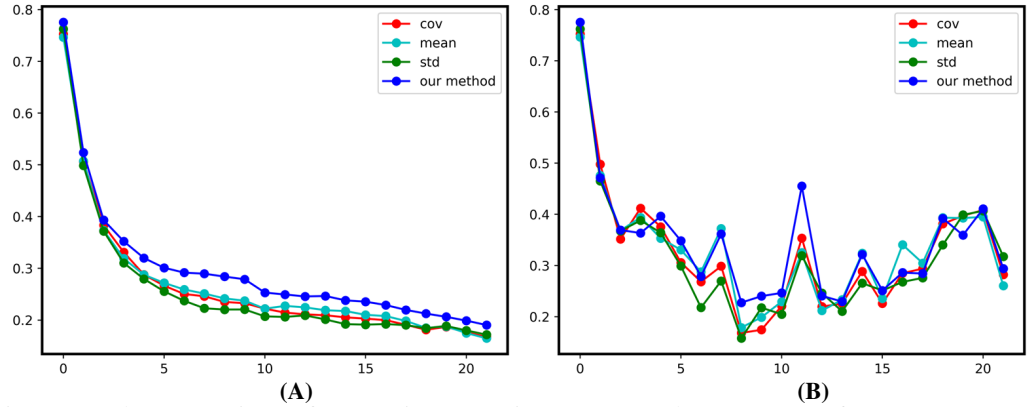


Figure 3: The comparison of uncertainty experiments. (A) The average performance on IIRC-CIFAR. (B) The PW-JS values indicating the performance of newly incremental phase.

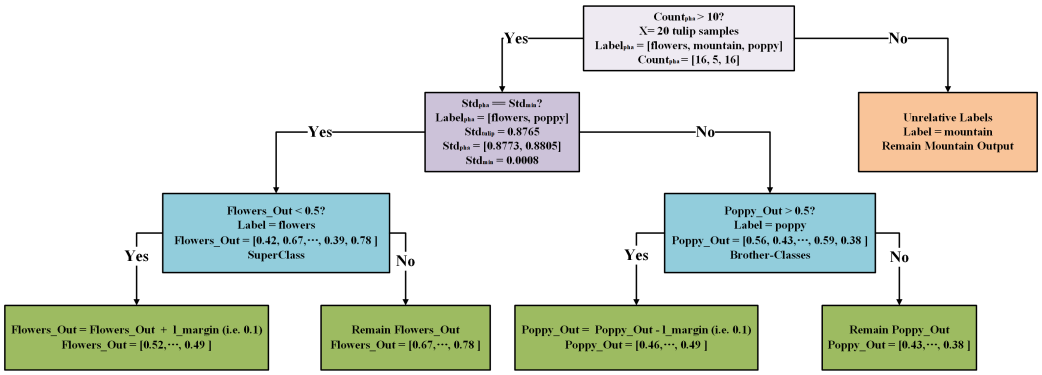


Figure 4: The decision process of tulip class on IIRC-CIFAR.

greater than 0.5. For brother classes, the entropy value greater than 0.5 is subtracted by a margin distance value, while the entropy value less than 0.5 remains unchanged.

A.5. More Comparisons With SOTA

Considering that the experimental results of the HCV paper and IIRC paper are different in task configurations, we follow the setting of HCV [14] and conduct a fair comparison experiment and find that SOTA results are obtained on his proposed IIRC-ImageNet-Subset. In the IIRC-Cifar experiment of the HCV paper, better results were achieved because that HCV constructed a stronger labeling relationship under a small number of categories. In contrast, our method focuses on the discovery of class relations during continuous increment in the IIRC setting and the construction of feature distributions corresponding to the hierarchical relations, which are directly embeddable in the HCV method. Therefore, we also conducted experiments on this and obtained a new SOTA on IIRC-Cifar. The results of our experiments are shown in Table. 3.

To verify the effectiveness of RBF Kernel, we conduct a comparison experiment between the methods with and without (*i.e.*, simple-dis) RBF mapping. The experimental results are shown in Table. 4.

Class hierarchical, global representations, and entropy-based margin controllers are widely used in various domains, including few-shot and incremental learning. However, we argue that our paper's corresponding definitions, specific implementations, and functions are entirely different. We conduct some statistical comparisons with the provided work [7, 8, 2] to demonstrate the superiority of our method further. To demonstrate the superiority of our method, we replace our global representation with the one in [8] (*i.e.*, Global_Repre), and find that the performance exhibits a significant degradation as Table. 1 shown due to the inability to adapt to unknown hierarchical relations. We conduct an experimental comparison with the entropy-based method utilized in [2]. As Table. 2 shows, our HDA component can better maintain hierarchical relationships.

phase	model	
	Ours	Global_Repre
0	**0.7753**	0.7662
1	**0.5237**	0.4561
2	**0.3931**	0.2824
3	**0.3521**	0.2876
4	**0.3199**	0.2742
5	**0.3010**	0.2542
6	**0.2917**	0.2459
7	**0.2896**	0.2328
8	**0.2840**	0.2167
9	**0.2788**	0.2131
10	**0.2531**	0.2038
11	**0.2494**	0.1858
12	**0.2457**	0.1655
13	**0.2465**	0.1565
14	**0.2381**	0.1470
15	**0.2355**	0.1366
16	**0.2290**	0.1325
17	**0.2196**	0.1231
18	**0.2125**	0.1125
19	**0.2063**	0.1102
20	**0.1989**	0.1169
21	**0.1905**	0.1080

Table 1: The average performance on IIRC-CIFAR between our methods and Global_Repre after each phase using the precision-weighted Jaccard Similarity.

B. Detailed Explanation

B.1. The Details of Experiment Instructions

Setting of Experimental Metrics. In Fig. 5 (A) and (B) of the main text, we conduct experiments on ten different task configurations of the IIRC-CIFAR and report the mean and standard deviation. In Fig. 5 (D), we run experiments on five task configurations of the IIRC-ImageNet-lite and report the mean and standard deviation. In Fig. 5 (E), we run experiments on the IIRC-ImageNet-Subset ten times and report the mean. In Fig. 5 (F), we run experiments on five different task configurations of the IIRC-ImageNet-full and report the mean.

Training Parameters. For IIRC-CIFAR training, the number of training epochs per phase is 140, but the number of training epochs in the first phase is 280. We set the batch size to 128 and the random seed to 100, using ResNet32 as the backbone. We use SGD as the optimizer with a momentum value of 0.9. The learning rate is initialized to 1.0, and the learning rate decrease strategy is an adaptively adjusted learning rate (Reduce LR On Plateau). The learning rate is adjusted to 0.1 of the current value, and 20 buffer samples are saved for each class. For the IIRC-ImageNet-lite, the number of training epochs per phase is 100, and the number of the first phase is 200. We use ResNet50 as the backbone with a learning rate initialized to 0.1, and the rest of the parameters are configured in the same way as IIRC-CIFAR.

Dataset Configuration. Download the CIFAR100 and ImageNet into "./data" folder. For IIRC-CIFAR, set dataset_path to the directory where CIFAR100 is located. For IIRC-ImageNet-Lite, set the dataset_path to the next level directory of ImageNet.

Code. The code will be available on https://github.com/ArrowYJ/UAHR.

Environment Requirements. On IIRC-CIFAR, we use GeForce GTX 1080. On IIRC-ImageNet-lite, we use NVIDIA GeForce RTX 3090. For the runtime environment we use PyTorch 1.5.0, Torchvision 0.6.0, NumPy 1.18.5, Pillow 7.0.0, lmdb 1.0.0, NumPy 1.18.5, seaborn 0.10.1, pytest 5.4.3, pytest-cov 2.9.0, and mllogger by running "pip install "mllogger[all]"".

phase	model	
	Ours	Entropy-based
0	**0.7753**	0.7626
1	**0.5355**	0.5072
2	**0.5158**	0.3733
3	**0.4638**	0.3190
4	**0.4256**	0.2875
5	**0.3011**	0.2719
6	**0.2910**	0.2664
7	**0.2844**	0.2512
8	**0.2773**	0.2418
9	**0.2626**	0.2374
10	**0.2531**	0.2220
11	**0.2388**	0.2275
12	**0.2295**	0.2246
13	**0.227**	0.2190
14	**0.2389**	0.2174
15	**0.2356**	0.2028
16	**0.2227**	0.2077
17	**0.2020**	0.1983
18	**0.2039**	0.1850
19	**0.2007**	0.1876
20	**0.2094**	0.1749
21	**0.1905**	0.1675

Table 2: The average performance on IIRC-CIFAR between our method and Entropy-based after each phase using the precision-weighted Jaccard Similarity.

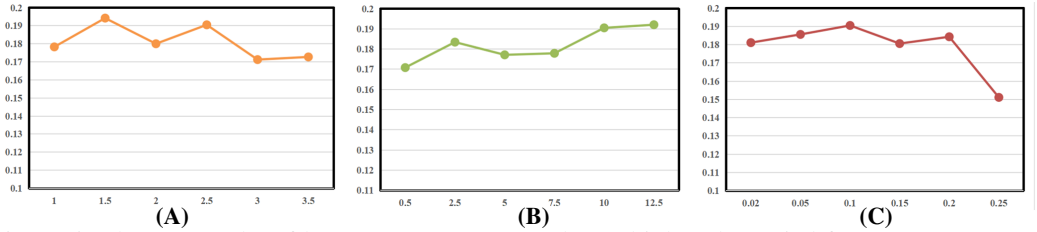


Figure 5: The test results of hyperparameters. (A) The multiple values tried for temperature. (B) The multiple values tried for . (C) The multiple values tried for l_margin.

B.2. The Details of Hyperparameter

Method Hyperparameter Our proposed method UAHR has three hyperparameters: temperature, , l_margin. The temperature is responsible for controlling the scaling parameters when calculating $\mathrm{BCELossWithLogits}$, is the learning weight of the GRE component, and l_margin is the boundary distance for adjusting the output of the old class in the HDA component.

By experimenting with three sets of control variables, we determined the rough choice of the hyperparameter. As shown in Fig. 5, the horizontal coordinates are the values of the corresponding hyperparameter, and the vertical coordinates are the values of the final PW-JS: Fig. (A) shows multiple values tried for temperature, and we can see that it is sufficient to control the value between 1 and 2.5. We used the value of 2.5 in the main experiment. Fig. (B) shows the trial of multiple values of , and we can see that it is enough to choose a value greater than 2.5, while we chose 10.0 in the main experiment. Fig. (C) shows the trial of multiple values of l_margin, and we can see that it is enough to choose a value less than 0.2. Moreover, we chose 0.10 in the main experiment.

B.3. Detailed Values of the Curves

Detailed values of the experimental plots of the paper are presented. (1) The exact values of the (A) graph of Fig. 6 for the main experiment are shown in Table. 5. (2) The exact values of the (B) plot of Fig. 6 for the main experiment are shown in Table. 6. (3) Detailed values of the complete phase of Table 1 for the main experiment are shown in Table. 7. (4) Detailed values of the (D) plot of Fig.

phase	model		
	UAHR+infer HCV	HCV-iCaRL-CNN	UAHR
0	0.7834	0.784	**0.8029**
1	**0.6655**	0.5861	0.571
2	**0.5158**	0.4324	0.4171
3	**0.4638**	0.3976	0.3588
4	**0.4256**	0.3832	0.3273
5	**0.4055**	0.3653	0.2987
6	**0.3910**	0.3735	0.2839
7	**0.3844**	0.3568	0.2697
8	**0.3773**	0.3437	0.2696
9	**0.3626**	0.3477	0.267
10	**0.3593**	0.3323	0.2464
11	**0.3388**	0.3268	0.2517
12	**0.3295**	0.3273	0.2494
13	0.317	**0.3174**	0.2440
14	**0.3189**	0.2963	0.2413
15	**0.3048**	0.2954	0.2398
16	**0.3027**	0.2885	0.2392
17	**0.2920**	0.2795	0.2246
18	**0.2939**	0.2725	0.2136
19	**0.2907**	0.2702	0.2163
20	**0.2894**	0.2669	0.2140
21	**0.2869**	0.2554	0.2076

Table 3: The average performance on IIRC-2-CIFAR after each phase using the precision-weighted Jaccard Similarity.

6 for the main text experiments are shown in Table. 8. (5) The exact values of the (E) plot of Fig. 6 for the main experiment are shown in Table. 9. (6) Detailed values of the (F) plot of Fig. 6 for the main text experiments are shown in Table. 10.

B.4. Related work of Uncertainty Estimation

The uncertainty of DNN models [3] mainly comes from the following factors: (1) The diversity in real-world scenarios. (2) The inherent error of the measurement system. (3) The structural error of the DNN model. (4) Errors during DNN training. (5) Errors caused by unknown data. For the model uncertainty, existing solutions can be divided into four types according to the number of models and model structure adjustment strategy. **Single deterministic methods** give the prediction based on one single forward pass within a deterministic network. J. Van Amersfoort [11] adopts the RBF network to learn a linear transformation on the logarithm and classify the input according to the distance between the transformed logarithm and the class centroids. In this case, the uncertainty of the data can be directly estimated from a distance between the centroids. **Bayesian methods** cover all kinds of stochastic DNNs. A simple solution for computing the Gaussian approximation of the last layer is proposed [4], and experiments show that the method significantly improves the calibration and alleviates the overconfidence prediction of the ReLU network. **Ensemble methods** combine the predictions of several different deterministic networks at inference. An ensemble training process is introduced to quantify the prediction uncertainty within the DNN [5], in which a member network is designed with two heads representing the value and uncertainty of the prediction data. **Test-time augmentation methods** give the prediction based on one single deterministic network but augment the input data at test-time to generate several predictions, which are used to evaluate the certainty of the prediction. A method called "greedy policy search" [9] captures more uncertainty by building a test-time augmentation strategy, which chooses the amount of augmentation to be included in a fixed-length policy. In this paper, the hierarchical uncertainty is estimated by the class-specific entropy distribution, promoting the optimization of incremental models.

C. Limitation and Society Impact

As we can see in the confusion Fig. 2, our approach weakens the confusion, but there is still much room for improvement in learning new classes. We consider that catastrophic forgetting leads to

phase	model	
	Ours	simple-dis
0	0.7753	**0.7883**
1	**0.5355**	0.5237
2	**0.5158**	0.3931
3	**0.4638**	0.3464
4	**0.4256**	0.3107
5	**0.3011**	0.2897
6	**0.2910**	0.2664
7	**0.2844**	0.2562
8	**0.2773**	0.2547
9	**0.2626**	0.2487
10	**0.2531**	0.2462
11	**0.2388**	0.2375
12	**0.2295**	0.2246
13	**0.227**	0.2259
14	**0.2389**	0.2249
15	**0.2356**	0.2246
16	**0.2227**	0.2177
17	**0.2020**	0.2083
18	**0.2039**	0.1998
19	**0.2007**	0.1955
20	**0.2094**	0.1924
21	**0.1905**	0.1802

Table 4: The average performance on IIRC-CIFAR between our method and simple-dis after each phase using the precision-weighted Jaccard Similarity.

significant challenges in constructing our label relationships at a later phase, and this is one area where we would like to improve performance.

This work has the following potential positive impact on society. Since our method requires representative samples of previously stored classes during incremental learning, this may violate data privacy from previous phases. So we will also try to weaken catastrophic forgetting by not storing data from past phases.

phase	model					
	iCaRL-norm	LUCIR	iCaRL-CNN	ER	podnet	Ours
0	**0.75(0.035)**	0.74(0.035)	0.70(0.040)	0.72(0.036)	**0.75(0.037)**	0.73(0.041)
1	**0.50(0.033)**	0.49(0.10)	0.47(0.030)	0.31(0.046)	0.36(0.056)	0.49(0.027)
2	**0.39(0.021)**	0.35(0.170)	0.36(0.015)	0.23(0.052)	0.28(0.057)	0.37(0.016)
3	**0.32(0.0239)**	0.30(0.155)	0.30(0.023)	0.20(0.026)	0.22(0.037)	0.31(0.028)
4	**0.28(0.021)**	0.27(0.142)	0.26(0.021)	0.17(0.024)	0.19(0.029)	**0.28(0.020)**
5	0.23(0.025)	0.25(0.124)	0.23(0.018)	0.18(0.024)	0.18(0.021)	**0.25(0.025)**
6	0.21(0.026)	0.23(0.127)	0.21(0.021)	0.18(0.029)	0.17(0.023)	**0.24(0.027)**
7	0.19(0.028)	0.19(0.133)	0.19(0.020)	0.17(0.034)	0.15(0.016)	**0.23(0.029)**
8	0.18(0.024)	0.18(0.128)	0.18(0.020)	0.16(0.018)	0.13(0.020)	**0.23(0.028)**
9	0.18(0.021)	0.15(0.127)	0.17(0.018)	0.15(0.020)	0.13(0.022)	**0.22(0.026)**
10	0.17(0.018)	0.14(0.112)	0.16(0.016)	0.17(0.033)	0.12(0.021)	**0.21(0.018)**
11	0.16(0.018)	0.13(0.112)	0.16(0.015)	0.15(0.015)	0.11(0.011)	**0.21(0.018)**
12	0.16(0.014)	0.12(0.103)	0.16(0.018)	0.15(0.029)	0.11(0.014)	**0.21(0.017)**
13	0.15(0.016)	0.13(0.089)	0.15(0.015)	0.15(0.028)	0.11(0.012)	**0.21(0.020)**
14	0.15(0.014)	0.11(0.092)	0.15(0.013)	0.14(0.019)	0.11(0.009)	**0.21(0.015)**
15	0.15(0.014)	0.11(0.089)	0.15(0.011)	0.13(0.013)	0.10(0.009)	**0.20(0.016)**
16	0.15(0.014)	0.09(0.089)	0.15(0.014)	0.14(0.011)	0.09(0.009)	**0.20(0.016)**
17	0.15(0.011)	0.07(0.080)	0.15(0.014)	0.14(0.018)	0.09(0.008)	**0.20(0.015)**
18	0.15(0.011)	0.06(0.077)	0.15(0.014)	0.13(0.011)	0.08(0.004)	**0.19(0.014)**
19	0.15(0.011)	0.06(0.072)	0.15(0.010)	0.14(0.015)	0.08(0.006)	**0.18(0.014)**
20	0.14(0.010)	0.06(0.069)	0.14(0.010)	0.13(0.010)	0.07(0.008)	**0.18(0.013)**
21	0.15(0.010)	0.06(0.067)	0.15(0.010)	0.13(0.006)	0.07(0.005)	**0.17(0.009)**

Table 5: The average performance on IIRC-CIFAR after each phase using the precision-weighted Jaccard Similarity with the standard deviation between brackets.

phase	model					
	ER	AGEM	LUCIR	iCaRL-CNN	iCaRL-norm	Ours
0	0.72(0.037)	0.72(0.031)	0.74(0.035)	0.71(0.040)	**0.75(0.033)**	0.73(0.041)
1	0.66(0.132)	0.57(0.150)	0.38(0.183)	0.58(0.125)	0.59(0.131)	**0.64(0.132)**
2	0.59(0.093)	0.54(0.097)	0.33(0.122)	0.41(0.100)	0.41(0.113)	**0.47(0.105)**
3	0.58(0.092)	0.55(0.115)	0.33(0.102)	0.37(0.063)	0.38(0.056)	**0.40(0.053)**
4	0.57(0.108)	0.48(0.083)	0.28(0.100)	0.29(0.065)	0.29(0.070)	**0.31(0.056)**
5	0.57(0.141)	0.56(0.149)	0.25(0.152)	0.26(0.064)	0.25(0.085)	**0.27(0.071)**
6	0.64(0.087)	0.62(0.111)	0.15(0.131)	0.21(0.040)	0.20(0.050)	**0.24(0.024)**
7	0.57(0.097)	0.58(0.125)	0.17(0.114)	0.20(0.042)	0.19(0.056)	**0.25(0.065)**
8	0.54(0.078)	0.52(0.099)	0.11(0.079)	0.18(0.049)	0.17(0.054)	**0.24(0.029)**
9	0.53(0.115)	0.54(0.101)	0.10(0.069)	0.16(0.029)	0.15(0.029)	**0.23(0.030)**
10	0.60(0.119)	0.58(0.151)	0.15(0.114)	0.15(0.026)	0.13(0.029)	**0.24(0.050)**
11	0.54(0.104)	0.51(0.110)	0.10(0.083)	0.16(0.050)	0.15(0.059)	**0.25(0.097)**
12	0.52(0.140)	0.54(0.155)	0.08(0.081)	0.15(0.038)	0.13(0.029)	**0.25(0.020)**
13	0.60(0.124)	0.60(0.110)	0.13(0.110)	0.16(0.059)	0.14(0.067)	**0.26(0.046)**
14	0.52(0.118)	0.55(0.136)	0.07(0.056)	0.14(0.033)	0.14(0.042)	**0.29(0.031)**
15	0.47(0.055)	0.46(0.082)	0.05(0.038)	0.16(0.040)	0.15(0.036)	**0.28(0.029)**
16	0.48(0.083)	0.49(0.073)	0.05(0.042)	0.15(0.037)	0.14(0.028)	**0.27(0.030)**
17	0.54(0.100)	0.59(0.118)	0.03(0.044)	0.14(0.035)	0.13(0.052)	**0.31(0.054)**
18	0.45(0.087)	0.47(0.092)	0.04(0.042)	0.15(0.032)	0.14(0.037)	**0.35(0.045)**
19	0.51(0.106)	0.52(0.095)	0.05(0.050)	0.14(0.035)	0.13(0.033)	**0.31(0.050)**
20	0.49(0.068)	0.50(0.104)	0.06(0.058)	0.15(0.047)	0.13(0.032)	**0.34(0.049)**
21	0.41(0.097)	0.48(0.060)	0.04(0.027)	0.13(0.037)	0.13(0.036)	**0.30(0.041)**

Table 6: Per phase performance over the test samples of a specific task j, after training on that phase with the standard deviation between brackets on IIRC-CIFAR.

phase	model			
	Baseline	Baseline + GRE	Baseline + HDA	Baseline + GRE + HDA
0	**0.783**	0.770	0.771	0.775
1	**0.544**	0.514	0.528	0.524
2	**0.394**	0.374	0.389	0.393
3	0.330	0.310	0.337	**0.352**
4	0.291	0.275	0.314	**0.320**
5	0.265	0.268	0.293	**0.301**
6	0.254	0.238	0.277	**0.292**
7	0.241	0.230	0.273	**0.290**
8	0.231	0.224	0.265	**0.284**
9	0.231	0.223	0.266	**0.279**
10	0.213	0.215	0.247	**0.253**
11	0.201	0.202	0.246	**0.249**
12	0.194	0.215	0.244	**0.246**
13	0.183	0.192	0.234	**0.247**
14	0.185	0.191	0.229	**0.238**
15	0.188	0.196	0.224	**0.236**
16	0.191	0.194	0.222	**0.229**
17	0.183	0.200	0.212	**0.220**
18	0.184	0.192	0.199	**0.213**
19	0.185	0.188	0.197	**0.206**
20	0.183	0.186	0.192	**0.199**
21	0.178	0.183	0.184	**0.190**

Table 7: The performance of ablation study on IIRC-CIFAR with our method.

phase	model				
	ER	LUCIR	iCaRL-CNN	iCaRL-Norm	Ours
0	0.70(0.027)	0.76(0.025)	**0.78(0.018)**	0.75(0.011)	0.73(0.022)
1	0.13(0.022)	0.17(0.044)	**0.46(0.039)**	**0.46(0.035)**	**0.46(0.012)**
2	0.12(0.071)	0.15(0.048)	0.34(0.047)	0.35(0.023)	**0.36(0.030)**
3	0.08(0.010)	0.14(0.045)	0.27(0.035)	0.29(0.021)	**0.30(0.009)**
4	0.08(0.010)	0.10(0.068)	0.23(0.022)	**0.25(0.016)**	**0.25(0.001)**
5	0.07(0.012)	0.10(0.063)	0.20(0.018)	0.21(0.013)	**0.23(0.006)**
6	0.07(0.017)	0.06(0.062)	0.18(0.018)	0.19(0.016)	**0.22(0.010)**
7	0.06(0.004)	0.04(0.057)	0.17(0.013)	0.18(0.013)	**0.19(0.003)**
8	0.07(0.013)	0.03(0.056)	0.16(0.010)	0.17(0.009)	**0.19(0.005)**
9	0.06(0.002)	0.03(0.053)	0.16(0.011)	0.16(0.011)	**0.18(0.010)**
$performancedroppingrate$	0.914	0.960	0.795	0.787	**0.753**

Table 8: The average performance on IIRC-ImageNet-lite after each phase using the precision-weighted Jaccard Similarity with the standard deviation between brackets.

phase	model						
	iCaRL-cnn	ER	Finetune	HCV(iCaRL-cnn)	iCaRL-norm	HCV(LUCIR)	Ours
1	0.846	0.8521	0.8467	0.8523	0.8433	0.8456	**0.8525**
2	0.4622	0.3587	0.3228	**0.5362**	0.4712	0.5014	0.4965
3	0.3247	0.1991	0.1681	**0.4058**	0.3302	0.3822	0.356
4	0.2665	0.1399	0.1032	**0.3097**	0.2577	0.3023	0.2895
5	0.2327	0.1219	0.1828	**0.2949**	0.224	0.2299	0.2694
6	0.1946	0.1848	0.1635	**0.2606**	0.19.1	0.1808	0.2306
7	0.1755	0.1455	0.113	**0.2261**	0.1699	0.1523	0.2127
8	0.1676	0.1376	0.212	0.2011	0.1566	0.1282	**0.216**
9	0.1669	0.0748	0.0316	0.1396	0.1562	0.1282	**0.2158**
10	0.1316	0.0693	0.0324	0.1371	0.1513	0.1271	**0.2132**
11	0.1603	0.0677	0.0322	0.1263	0.1531	0.1256	**0.2109**

Table 9: The average performance on IIRC-ImageNet-Subset after each phase using the precision-weighted Jaccard Similarity.

phase	model				
	ER	LUCIR	iCaRL-cnn	iCaRL-norm	Ours
0	0.7	0.75	**0.78**	0.76	0.7417
1	0.13	0.1	0.47	**0.503**	0.4859
2	0.1	0.14	0.34	0.341	**0.35**
3	0.08	0.13	0.27	0.277	**0.2875**
4	0.08	0.12	0.23	**0.246**	0.2438
5	0.1	0.11	0.21	**0.222**	0.2212
6	0.07	0.1	0.19	0.208	**0.209**
7	0.06	0.11	0.17	**0.189**	0.187
8	0.06	0.1	0.16	0.181	**0.186**
9	0.06	0.09	0.15	0.17	**0.179**
10	0.06	0.08	0.15	0.166	**0.174**
11	0.06	0.08	0.14	0.158	**0.177**
12	0.06	0.08	0.14	0.153	**0.166**
13	0.06	0.07	0.13	0.157	**0.165**
14	0.06	0.07	0.13	0.155	**0.163**
15	0.05	0.07	0.13	0.146	**0.156**
16	0.05	0.06	0.12	0.14	**0.153**
17	0.05	0.06	0.12	0.14	**0.151**
18	0.05	0.05	0.12	0.137	**0.147**
19	0.05	0.05	0.11	0.131	**0.144**
20	0.05	0.05	0.11	0.127	**0.14**
21	0.04	0.05	0.11	0.123	**0.137**
22	0.04	0.04	0.1	0.127	**0.137**
23	0.04	0.04	0.1	0.123	**0.134**
24	0.04	0.03	0.1	0.124	**0.135**
25	0.03	0.02	0.1	0.121	**0.133**
26	0.04	0.02	0.09	0.115	**0.128**
27	0.03	0.02	0.09	0.114	**0.124**
28	0.03	0.02	0.08	0.11	**0.118**
29	0.03	0.02	0.08	0.105	**0.115**
30	0.03	0.02	0.08	0.09	**0.108**
31	0.02	0.02	0.08	0.09	**0.106**
32	0.02	0.02	0.08	0.08	**0.102**
33	0.02	0.02	0.08	0.08	**0.09**
34	0.01	0.01	0.07	0.08	**0.09**

Table 10: The average performance on IIRC-ImageNet-full after each phase using the precision-weighted Jaccard Similarity.

References

[1] Wenhu Chen, Yilin Shen, Hongxia Jin, and William Wang. A variational dirichlet framework for out-of-distribution detection. *arXiv preprint arXiv:1811.07308*, 2018.

[2] Wen-Hsuan Chu, Yu-Jhe Li, Jing-Cheng Chang, and Yu-Chiang Frank Wang. Spot and learn: A maximum-entropy patch sampler for few-shot image classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6251–6260, 2019.

[3] Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, et al. A survey of uncertainty in deep neural networks. *arXiv preprint arXiv:2107.03342*, 2021.

[4] Agustinus Kristiadi, Matthias Hein, and Philipp Hennig. Being bayesian, even just a bit, fixes overconfidence in relu networks. In *International conference on machine learning*, pages 5436–5446. PMLR, 2020.

[5] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

[6] Dan Levi, Liran Gispan, Niv Giladi, and Ethan Fetaya. Evaluating and calibrating uncertainty prediction in regression tasks. *Sensors*, 22(15):5540, 2022.

[7] Aoxue Li, Tiange Luo, Zhiwu Lu, Tao Xiang, and Liwei Wang. Large-scale few-shot learning: Knowledge transfer with class hierarchy. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 7212–7220, 2019.

[8] Aoxue Li, Tiange Luo, Tao Xiang, Weiran Huang, and Liwei Wang. Few-shot learning with global class representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9715–9724, 2019.

[9] Alexander Lyzhov, Yuliya Molchanova, Arsenii Ashukha, Dmitry Molchanov, and Dmitry Vetrov. Greedy policy search: A simple baseline for learnable test-time augmentation. In *Conference on Uncertainty in Artificial Intelligence*, pages 1308–1317. PMLR, 2020.

[10] Abhijit Guha Roy, Sailesh Conjeti, Nassir Navab, Christian Wachinger, Alzheimer's Disease Neuroimaging Initiative, et al. Bayesian quicknat: Model uncertainty in deep whole-brain segmentation for structure-wise quality control. *NeuroImage*, 195:11–22, 2019.

[11] Joost Van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal. Uncertainty estimation using a single deep deterministic neural network. In *International conference on machine learning*, pages 9690–9700. PMLR, 2020.

[12] Apoorv Vyas, Nataraj Jammalamadaka, Xia Zhu, Dipankar Das, Bharat Kaul, and Theodore L Willke. Out-of-distribution detection using an ensemble of self supervised leave-out classifiers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 550–564, 2018.

[13] Guotai Wang, Wenqi Li, Michael Aertsen, Jan Deprest, Sébastien Ourselin, and Tom Vercauteren. Aleatoric uncertainty estimation with test-time augmentation for medical image segmentation with convolutional neural networks. *Neurocomputing*, 338:34–45, 2019.

[14] Kai Wang, Xialei Liu, Luis Herranz, and Joost van de Weijer. Hcv: Hierarchy-consistency verification for incremental implicitly-refined classification. *arXiv preprint arXiv:2110.11148*, 2021.