Uncertainty-Aware Hierarchical Refinement for Incremental Implicitly-Refined Classification

Anonymous Author(s)
Affiliation
Address
email

Abstract

Incremental implicitly-refined classification aims at assigning hierarchical labels to the same instance encountered at different phases. Existing methods tend to fail in generating hierarchy-invariant descriptor when novel classes are inherited from the old ones. To address the issue, this paper explores the inheritance relations in the process of multi-level semantic increment, and propose an Uncertainty-Aware Hierarchical Refinement (UAHR) scheme. Our proposed scheme consists of a global representation extension strategy that enhances the discrimination of incremental representation by widening the corresponding margin distance, and a hierarchical distribution alignment strategy that refines the distillation process by explicitly determining the inheritance relationship of the incremental class. Particularly, the shifting subclasses are corrected under the guidance of hierarchical uncertainty, ensuring the consistency of the homogeneous features. Extensive experiments on benchmarks IIRC-CIFAR and IIRC-ImageNet-lite demonstrate the superiority of our proposed method over state-of-the-art.

1 Introduction

In recent years, deep learning has made huge breakthroughs in the field of computer vision, matching or even surpassing human performance on some recognition tasks. However, learning multiple tasks in sequence (*i.e.*, continual learning) remains a major challenge, which requires models to have the ability to aggregate different learning objectives into a coherent whole over time.

When a new task comes with the increase of identified classes (*i.e.*, class-incremental learning), joint training with all old and new data is too time-consuming and labor-intensive. Furthermore, most data from past tasks are unavailable. A simple alternative is to directly fine-tune the network with new data. In this case, the performance of the old class is severely degraded due to the bias of the representation and classifier towards the new class, which is also known as catastrophic forgetting.

To address the issue, existing methods maintain the performance of the old class by preserving the representative samples (*i.e.*, exemplar) and aligning the output distribution (*i.e.*, knowledge distillation). The joint optimization with incremental samples strikes a balance between stability and plasticity, resulting in better predictions that fall into the label space of the old or new class.

However, is incremental learning always a binary classification option that distinguishes between the old and new class? In existing setting (*e.g.*, CIFAR100 in Fig. 1), each instance is arranged in a certain incremental stage in an exclusive manner. However, in real life, people's semantic understanding of the same instance may be gradually enriched as the learning process proceeds. As shown in the lower row of Fig. 1, the semantics of the same image is refined from vehicles to bus in the IIRC-CIFAR setting. This paper focuses on the incremental ability to progressively learn and

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

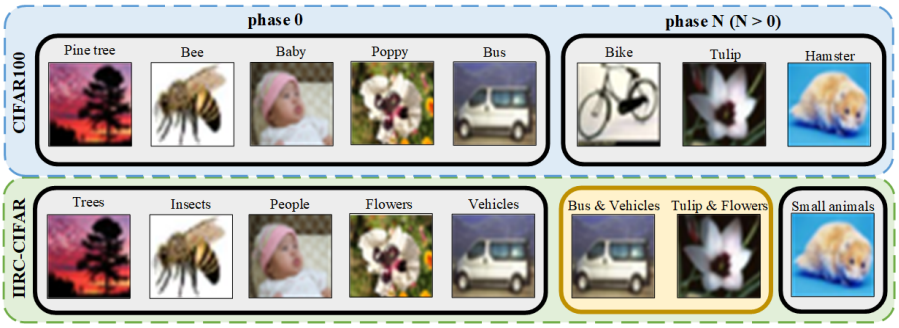


Figure 1: The setting of IIRC. Compared with the standard incremental process, superclasses and subclasses under IIRC setting are mixed in the incremental process, and the labels of subclasses images are refined progressively, even for the same image.

maintain multi-level semantic information, which is also known as Incremental Implicitly-Refined Classification (*i.e.* IIRC) [18].

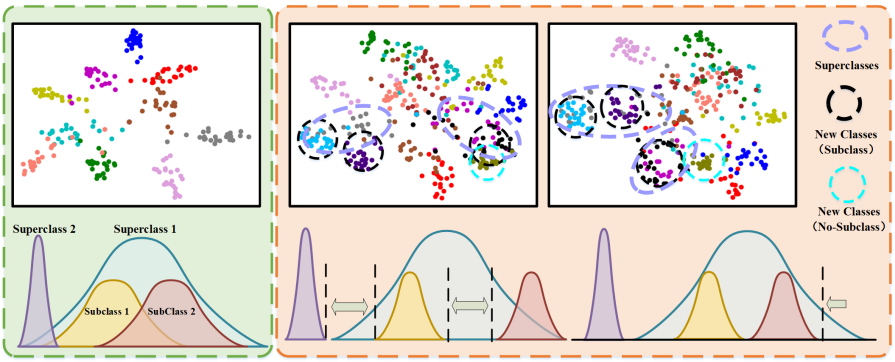
How to discern the semantic inheritance relationship in a hierarchical incremental scenery? We find that the incremental performance for multi-level recognition decreases dramatically when applying classical class-incremental learning methods directly to IIRC setting. Our analysis is mainly attributed to the following reasons. As shown in Fig. 2, on the one hand, although the probability distributions of classes with inheritance relationships show a obvious consistency in the initial phase, the incremental subclass inherited from a certain old class gradually outgroups under the supervision of new labels. It destroys the integrity of the representation of the whole old class, thus losing the semantic relevance of the hierarchical labels. On the other hand, some incremental classes inherit from none of the existing classes, leading to the feature confusion due to the lack of old supervision.

To this end, we propose an Uncertainty-Aware Hierarchical Refinement (UAHR) scheme, which exploits the correlation of hierarchical distributions to guide the optimization of incremental representation. A global representation extension strategy is proposed to widen the distribution distance among all new classes in the embedding space, enhancing their discriminative properties. A hierarchical distribution alignment strategy is further proposed to correct the optimization of the shifting subclasses by aligning with the distribution of the whole superclass, ensuring the consistency of the hierarchical uncertainty. Specifically, we use RBF mapping to explicitly measure the distance between the training samples and the class centers, quantifying the feature correlation. In the incremental phase, we calculate the entropy distribution of new classes in the old embedding space for the estimate of hierarchical uncertainty. The resulting differences and similarities are utilized to identify the multi-level semantic relationships, refining the subsequent distillation objectives. Comprehensive experimental results on IIRC-CIFAR and IIRC-ImageNet-lite datasets demonstrate the superiority of our method.

2 Related Work

2.1 Uncertainty Estimation

The uncertainty of DNN models [1] 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 adjustment strategy for the number of models and model structure. **Single deterministic methods** give the prediction based on one single forward pass within a deterministic network. J. Van Amersfoort [2] adopts 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 the 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 [3], and experiments show that the method significantly improves the calibration



(A) Intial phase **(B) Incremental phase**

Figure 2: Motivation of our method. The class with inheritance relationships have consistency in the initial distribution. As incremental learning proceeds, the subclasses shift away from the distribution of the corresponding superclasses, and mix with new classes without subclasses.

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 [16], in which 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" [17] 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.

2.2 Class-Incremental Learning

Existing class incremental learning methods can be mainly divided into four types. **Regularization-based** methods preserve high-weight parameters by estimating the importance of individual model parameters [4] [5] [6], while allowing unimportant parameters to be updated flexibly to ensure the learning ability for new classes. However, in the study of [7], it is shown that such methods have poor generalization performance in class incremental learning. **Distillation-based** methods encourage the model to learn new tasks, while the model representation obtained by the current training data is forced to mimic the representation of the old model. The new model in [8] completes knowledge distillation by matching the output of the sigmoid function with the old model. LUCIR [9] mitigates forgetting by distilling the output of softmax function with the temperature scale and L2 regularization. LwM [10] reduces forgetting by matching gradient attention maps from the highest-scoring labels. PODNet [11] performs knowledge distillation between old and new models by retaining intermediate features pooled along width and height, and controls the balance between previous knowledge and new information, resulting in a more adaptive incremental representation. **Rehearsal-based** methods work by storing a limited number of representative examples or adopting generative model for old class samples when training new tasks. Incremental Classifier Representation Learning (iCaRL) [12] retains a small number of samples for each class to approximate class centroids and make predictions based on the nearest class average classifier. [13] generates previously observed class samples by using generative adversarial networks (GANS) [14]. **Structure-based** methods keep the learned parameters related to the previous classes unchanged, and assign new parameters in different forms, such as unused parameters, extra networks, etc., to learn new knowledge. DER [15] concatenates the extracted features by adding a new feature extractor in the incremental phase. A sparsity error is adopted to encourage the model to compress parameters as much as possible. Different from existing incremental method focusing on the invariant semantic concepts, our method is designed for the progressive understanding of the incremental semantic information, facilitating better feature update and retention.

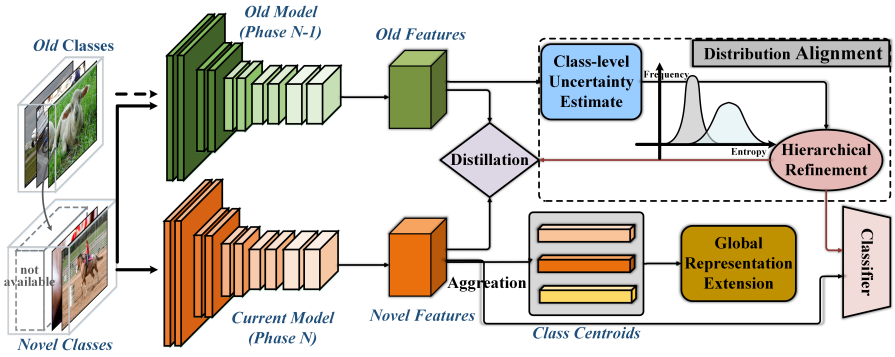


Figure 3: Our proposed Uncertainty-Aware Hierarchical Refinement scheme for IIRC.

3 My Method

3.1 Problem Description and Analysis

In real life, people's semantic understanding of the same thing is gradually enriched as the learning process proceeds. To simulate such a situation, N phases are arranged in order, denoted as $\{T_1, \cdots, T_N\}$. Each phase contains three sets, which are used for training, validating, and testing respectively. Each sample can have one or two labels. In the case of two labels, one label is a subclass and the other label is a superclass. The number of samples in the superclass is always larger than the subclass, and it increases with the number of subclasses. During training, we always follow an incomplete information setup, *i.e.*, samples with multiple labels are only provided with the label in the current phase. In the first phase, only a subset of all superclasses is used to train the model. Subsequent phases contain a mixture of superclasses and subclasses. In the training phase, the model is validated in the in-phase validation set which only contains current phase labels. In the evaluation phase, the model is evaluated on the post-phase validation set and the test set (both of them contain the complete labels).

In this setting, we face two challenging problems: (1) How to judge the relationship between labels across phases, especially the superclass-subclass relationship? And how to guide the optimization of class distribution to fall into the distribution of superclass exactly? (2) How to reduce the confusion among the feature distributions of new classes in the incremental phase? Therefore, We propose two strategies. (1) Global Representation Extension. During the new learning phase, the uncertainty between each class center is calculated by the RBF kernel. Then the uncertainty is minimized for training. (2) Hierarchical Distribution Alignment. The uncertainty of the classes across stages are calculated by calculating the output entropy of the new class samples with old model. Then it is used to discriminate the superclass-subclass relationship and the brother relationship. Finally, the optimization direction of the new model is guided by changing the output of the old model. Thus, we can keep the invariance of the superclass distribution.

3.2 Global Representation Extension

In the incremental phase, optimizing only the data of new classes causes great bias in the representations and classifiers. To maintain the stability of the feature space, we use the distance between representations as a measure of uncertainty. The optimization of new classes is achieved by comparing the output entropy with the target labels. While the stability of the old class representations is ensured by maintaining the same output between the old model and the new model with the new samples. Specifically, we use the RBF distance [2] for the calculation of representation extension loss. We use the RBF kernel to map representation to infinite dimensional space for comparison, which is shown as follows,

$$l_{div} = \sum_{c=0}^{n_b} K(h_\theta(x)_c, h_\theta(x)_{j_{near}}) = exp\left[\frac{\frac{1}{n_d}\|h_\theta(x)_c - h_\theta(x)_{j_{near}}\|_2^2}{2\sigma^2}\right] \tag{1}$$

where n_b denotes the number of classes contained in the current batch, $h_\theta(*)$ denotes the output features of the picture after global pooling, n_d denotes the output dimension after global pooling, and $h_\theta(*)_i$ denotes the average feature value of a particular class i. The symbol j in the formula is not equal to c, and j is the class number with the smallest distance to class c in the feature space. The optimization of this loss forces the extension of the distance between the class centers, obtaining better discrimination in the class distribution. Subsequently, we use a linear layer to output the entropy of the samples to judge the class of these samples. BCEWithLogitsLoss is adopted for optimization, which is shown as follows,

$$l_{cls} = -\frac{1}{N}\sum_{i=1}^{N}[Y_i \ln(sigmoid(f_\theta(F_i))) + (1 - Y_i)\ln(1 - sigmoid(f_\theta(F_i)))] \tag{2}$$

Where Y_i denotes the ground truth label of the sample i, N denotes the batch size of training,F_i denotes the feature of the i_{th} sample, f_θ denotes the classifier of the current model, note that the labels here as well as the final output of the model contain only the classes of current phase. The optimization of this loss allows the model to eventually judge the label of a sample by whether its output is greater than 0. And the closer to the class center, the larger the output value. Finally, we use the output value of the new sample on the old model as a supervised signal. Thus, the current model learns new classes while maintaining the knowledge of the old model. As shown below:

$$l_{dis} = BCEWithLogitsLoss(f_\theta(h_\theta(x))[:, : n_{old}], f^*_\theta(h^*_\theta(x)/\alpha)) \tag{3}$$

Where $h_\theta()$ denotes the backbone of the current model, $f_\theta()$ denotes the classifier of the current model, $h^*_\theta()$ denotes the backbone of the old model, $f^*_\theta()$ denotes the classifier head of the old model, α is a scale superparameter of distillation,n_{old} denotes the number of classes that have been learned. x is a training sample of the current phase. By optimizing this loss, the distribution of old class features keeps stable in the feature space.

3.3 Hierarchical Distribution Alignment

In incremental phase, focusing on the label relationships across stages is the key to solving the IIRC problem. The old model with fixed parameters can map samples into the old feature space, while the new subclass samples can be mapped into the space, which is close to the corresponding superclass distribution. We calculate the uncertainty of the labels across phases by counting the entropy of new samples on old model. Then the label relationships can be inferred from the hierarchical uncertainty. At the same time, we flexibly adjusting the distribution margin distance, to correct the shift of subclass samples and mitigate the confusion caused by the brother class.

Specifically, as shown in Fig. 3, we take the new class C as an example. We obtain all the mapping features on the old model with the new samples of class C and the output entropy as well. Then we count the single-label output for each phase and sample. According to the single label matrix, we calculate the uncertainty of a single label, by counting the number of samples, which are judged to belong to that single class. And then making the number divide by the number of total samples. Then we process the binarization of uncertainty results. The label that final result is 1 has a similar distribution to the current specific class C (possibly a superclass-subclass or brother relationship). After completing the initial matching of uncertainty distributions, more precise distribution matching is accomplished by the standard deviation(*i.e.*, std) values of the global pooling features. We calculate the standard deviation of each similar old class with the same operation. Since direct superclass-subclass relationships have closer standard deviation values on the feature distributions. So the old class label with the closest std value is judged as the superclass of the current class C, and the rest of the classes are brother classes. The detailed procedure is shown in the pseudo-code Algorithm 1.

After obtaining the label relationships, we use selective distribution alignment distillation mechanism to guide the representation learning of models. When the new class distribution falls largely within the old class, but there is still a partial shift. We add a margin distance to the output entropy of the corresponding old class. When the new class distribution has a crossover interval with the old class, but the gap of standard deviation between the distributions is large, we need to perform the reverse optimization by subtracting a margin distance value from the output entropy of the old class. So we can achieved the alignment of the distribution and mitigate the confusion caused by the brother class by this strategy.

Algorithm 1 Acquisition of Hierarchical Semantic Relationship

Input: The outputs of the samples of class c on old model $output$ and the number of samples $height$, the number of classes for each phase $Stage$, the global pooled features of the sample of class c on old model $feat$, and the number of the current phase idx.

Declare an initial label matrix M_{ori} of size $(height, idx)$.
for each $j \in [0, height]$ **do**
    for each $k \in [0, idx]$ **do**
        S_{max}, L_{max} = max($output[j, Stage[k]{:}Stage[k+1]]$)
        $M_{ori}[j, k] = L_{max}$
    end for
end for
Declare a phase label statistics matrix M_{pha} of size $(2, idx)$.
for each $j \in [0, idx]$ **do**
    L_{pha}, C_{pha} = countmax($M_{ori}[:, j]$)
    if $L_{pha} < height / 2$ **then**
        L_{pha} = -1
    end if
    $M_{pha}[0, j] = L_{pha}$
    $M_{pha}[1, j] = C_{pha}$
end for
For all labels in the current M_{pha} that are not -1, an equal number of corresponding sample features are randomly extracted from $feat$, and the standard deviation is calculated for each, forming Std_{pha}.
Calculate the standard deviation std for the current $feat$, declared as Std_c.
$Super$ = argmin($|Std_{pha} - Std_c|$)

Output: the super class of current class C : $Super$.

Specifically, according to the label relationships, when the current new class C is not related to the old class, the highest output entropy by the sample of class C on the old model, is subtracted by a margin value. And when the current new class C is a subclass of the old class, the output entropy value of the class C sample on the old model, which is corresponding to the superclass, is added one margin value. At the same time, the highest output entropy value of the non-superclass is subtracted by a margin value. Our selective distribution alignment distillation loss can be obtained by:

$$l_{dis} = BCEWithLogitsLoss(f_\theta(h_\theta(x))[:, : n_{old}], y^{new}) \tag{4}$$

where, y^{new} is the new output of the old model, after performing our hierarchical distribution alignment strategy. In the end, our complete loss is:

$$l_{all} = l_{cls} + l_{dis} + l_{div} * \gamma \tag{5}$$

where γ is the learning weight for the optimization of the global representation extension.

4 Experiments

4.1 Dataset and Settings

Based on the semantic relevance of labels, CIFAR100 [19] and ImageNet [20] datasets are integrated to form datasets, where class labels are two-level hierarchy[18]. Each label starts as a leaf node (i.e., subclass) and similar labels are assigned a common parent node (i.e., superclass). The integrated datasets are called IIRC-CIFAR and IIRC-ImageNet-lite. **IIRC-CIFAR.** Ten superclasses are set up, each with about 4 to 8 subclasses. In incremental phases, each new phase introduces 5 classes for IIRC-CIFAR. In total, IIRC-CIFAR involves 22 phases with 10 preset class orders for multiple tests, which is called as phase configuration. **IIRC-ImageNet-lite.** In IIRC-ImageNet-lite, sixty-three superclasses are set up, and the number of subclasses that belong to one superclasses varies greatly, from 3 to 118. There are a total of 10 phases with 30 classes per phase. 5 preset class orders are fixed for multiple tests.

The superclasses of each dataset are combined by extracting 40% samples from each subclass, while the corresponding subclasses are saved with 80% samples, so the superclasses and subclasses share

GRE	HDA	IIRC-CIFAR				
		phase 0	phase 5	phase 10	phase 15	phase 21
		78.35	26.48	21.27	18.81	17.78
√		75.67	24.12	18.96	18.29	17.60
	√	77.06	29.31	24.73	22.42	18.38
√	√	77.53	30.11	25.31	23.56	19.05

Table 1: Ablation study of our method on IIRC-CIFAR.

Method	IIRC-CIFAR			
	phase 5	phase 10	phase 15	phase 21
$Common\ Uncertainty$	25.70	20.89	19.55	15.32
$Ours$	30.11	25.31	23.56	19.05

Table 2: Comparison to common uncertainty method on IIRC-CIFAR.

20% samples. When the superclass contains more than 8 subclasses, the samples of the subclasses are extracted in the proportion of $\frac{8}{number\ of\ subclasses} * 40\%$. The training process follows the incomplete information setting, *i.e.*, if a sample has more than one label, only the label of the current phase is provided. The validation set also follows the incomplete information setting. After a certain number of training phases, the overall performance is evaluated on a validation set with complete information setting and a test set as well. Both of them contain samples of all seen classes and their corresponding complete labels.

Metrics. The evaluation metric uses the precision-weighted Jaccard similarity (PW-JS) to measure the performance of the model on phase k after training on phase j, as follows:

$$R_{jk} = \frac{1}{n_k} \sum_{i=1}^{n_k} \frac{|Y_{ki} \cap \widehat{Y}_{ki}|}{|Y_{ki} \cup \widehat{Y}_{ki}|} \times \frac{|Y_{ki} \cap \widehat{Y}_{ki}|}{|\widehat{Y}_{ki}|} \tag{6}$$

In the formula $j \geq k$, $\widehat{Y}_{ki}$ is the prediction value of the model for i_{th} samples of k_{th} phase, Y_{ki} is the ground truth, and n_k is the number of samples in this phase. And for the evaluation of the overall performance, we use the average precision-weighted Jaccard similarity of the model after training on phase j for all seen classes, as follows.

$$R_j = \frac{1}{n} \sum_{i=1}^{n} \frac{|Y_i \cap \widehat{Y}_i|}{|Y_i \cup \widehat{Y}_i|} \times \frac{|Y_i \cap \widehat{Y}_i|}{|\widehat{Y}_i|} \tag{7}$$

4.2 Ablation Study

To prove the effectiveness of our method, we conduct a set of ablation experiments on the IIRC-CIFAR dataset. One of the important components in our scheme is the global representation extension strategy, referred to as the GRE component. The other is hierarchical distribution alignment strategy, referred to as the HDA component. To verify the functionality of the GRE and HDA components, we conducted experiments in the case of phase configuration 0 of IIRC-CIFAR. As shown in Table 1, the HDA component alone achieves a improvement of 0.6% in the last phase, while the GRE component keeps the same results as baseline with an initial lag of 2.68%. And with both components added, 1.27% increase over baseline is achieved. It proves that both components play a positive role in the evaluation of the overall incremental performance.

In Fig. 5 (C), we show the performance of each phase, which is evaluated over the test samples at a specific phase j, after training on that phase(R_{jj} using Equation 6). Such cure graphs evaluate the learning ability for new classes. We can see that the HDA component alone achieves a great improvement, while adding the GRE component further enhances the learning ability with a small optimization on top of the HDA component.

4.3 Analysis

The impact of global representation extension. To explore the impact of global representation extension strategy, we visualize the t-SNE features of the baseline and our method. We select data

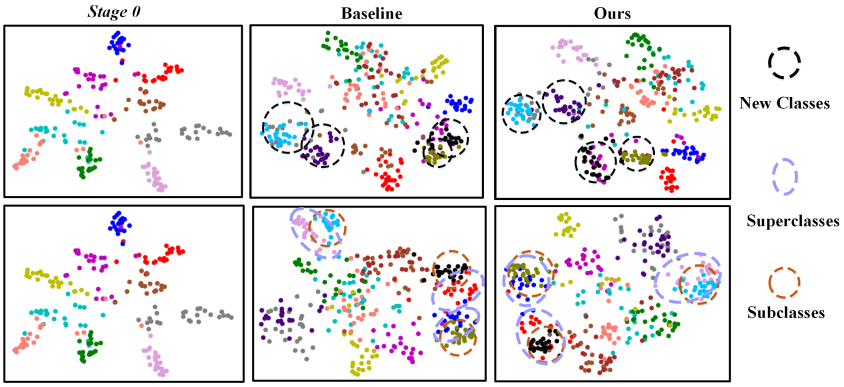


Figure 4: The impact of our method on the representations. (1) The upper row shows the effect of extend the distribution distance between the incremental classes. (2) The lower row shows the effect of the hierarchical distribution alignment.

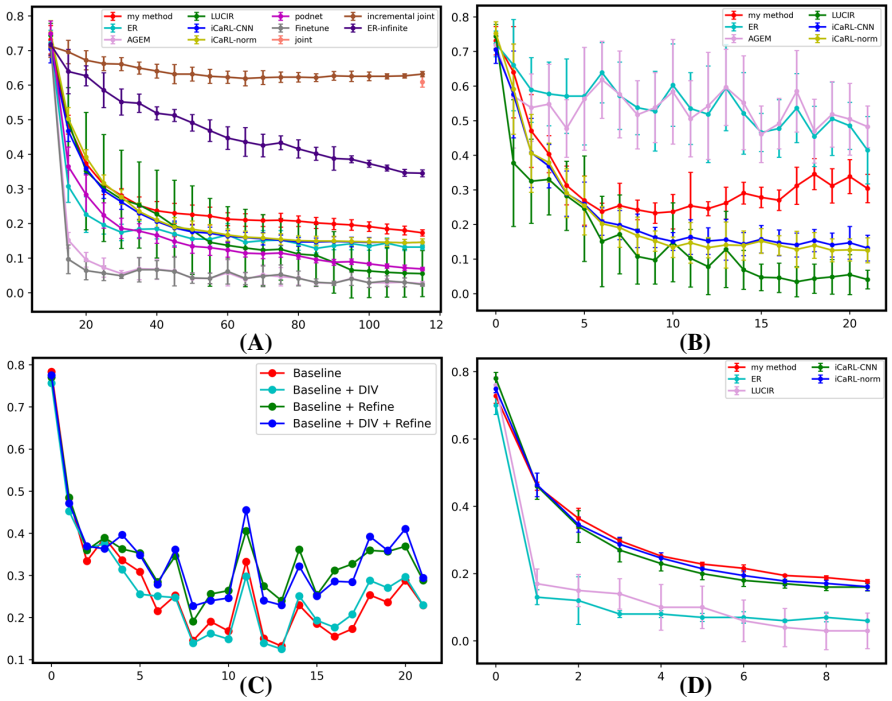
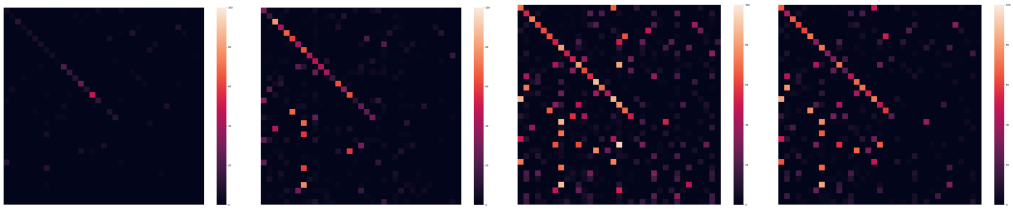


Figure 5: Curve graphs of the comparison experiments. (A) The overall performance comparison on the IIRC-CIFAR dataset. (B) The performance of the incremental phase on IIRC-CIFAR dataset. (C) It shows the performance of two components of our method on the evaluation of new class learning ability. (D) The performance comparison on the IIRC-ImageNet-lite dataset.

from phase 0 and phase 4 of IIRC-CIFAR with the phase configuration 0. In the upper part of Fig. 4, we can see that there is an overlap among the representation distributions of the new classes in the baseline method. While the overlap between the increasing classes is weakened and the distance between the increasing class feature distributions is expanded in our method. This proves that using the distance between representations as a measure of uncertainty is effective and the reduction of this uncertainty leads to the extension of the distance between the individual class centers.

The impact of the hierarchical uncertainty. To explore the effect of hierarchical uncertainty, we first compare different uncertainty methods. As shown in Table. 2, we use other common values to construct the label relationships. It can be seen from the Table. 2 that our method achieves better experimental results, which proves that the entropy output of the network and the standard deviation



(A)Fine Tune **(B)LUCIR** **(C)iCaRL-CNN** **(D)Ours**

Figure 6: Confusion matrix for different methods.

of the features have the greatest impact on the cross-hierarchical uncertainty estimation among the statistical properties of the superclass and the corresponding subclass.

To explore the effect of the hierarchical distribution alignment strategy. We visualize the t-SNE features for the baseline method and our method on the IIRC-CIFAR dataset with phase configuration 0 by selecting data from phase 0 and phase 11. In the bottom half of Fig. 4, we can see that there is a large shift between the distributions of subclasses and corresponding superclasses in baseline. While this shift is greatly corrected in our method, so that the distribution of the increasing subclasses falls in the distribution of the corresponding superclasses. It demonstrates that adjusting the optimization direction about the feature distribution according to the labeling relationship guides the new model to better learn the representational relationship between the old and new classes and deepen it gradually.

4.4 Comparison with SOTA

To comprehensively measure the performance of our method under IIRC setting, we reproduced the classical methods ER, AGEM, LUCIR, iCaRL-CNN, iCaRL-norm, podnet, etc and compare with them.

Average PW-JS. It can be seen in Fig. 5 (A) that our method achieves the best until the end among all methods with the incomplete information in IIRC-CIFAR. As shown in Fig. 5 (D), our methods also maintains the highest scores in larger IIRC-imagenet-lite, which demonstrates the effectiveness of our methods.

Performance on new classes. The PW-JS values indicating the performance of newly incremental phase are shown in Fig. 5. It can be seen that the ER algorithm and the AGM algorithm are superior in the ability to learn new classes. Combined with their performance on overall phases, it is clear that they obtain an advantage on the new class at the complete expense of the old class performance. Among all algorithms that are beneficial for suppressing the catastrophic forgetting, our method achieves the best, and outperforms existing methods in the ability of learning new classes.

We conduct an additional set of confusion graph experiments where we extracted the data from the IIRC-CIFAR with the phases 0, 1, 2, 7, 12, 17. They are combined and presented in Fig. 6. We can find that the LUCIR algorithm has a strong retention effect for old classes, while it can learn part of the labeling relationship when the number of phases is relatively small, but presents a weakness in learning new classes of phases when the number of phases increases. The iCaRL-CNN algorithm, on the other hand, is able to maintain old classes and learn new classes, but is very confusing for the construction of labeling relations. Our approach is able to maintain old classes and learn new classes more accurately, making a great improvement in maintaining label relations.

5 Conclusion

In this paper, a novel Uncertainty-Aware Hierarchical Refinement scheme is proposed for IIRC task. A global representation extension strategy is presented to enhance the discrimination of incremental classes, and the misleading distillation process is refined with a hierarchical distribution alignment strategy. Consequently, our method involves a multi-level semantic scenery into the incremental learning. Experimental results show the superiority our method in both stability and plasticity.

References

[1] Gawlikowski J, Tassi C, Ali M, et al.A Survey of Uncertainty in Deep Neural Networks[J]. 2021.

[2] J. Van Amersfoort, L. Smith, Y. W. Teh, and Y. Gal. Uncertainty estimation using a single deep deterministic neural network. in Proceedings of the 37th International Conference on Machine Learning. PMLR, 2020, pp.9690-9700.

[3] A. Kristiadi, M. Hein, and P. Hennig, "Being bayesian, even just a bit, fixes overconfidence in relu networks," in International Conference on Machine Learning. PMLR, 2020, pp. 5436–5446.

[4] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In ECCV, 2018.

[5] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka GrabskaBarwinska, et al. Overcoming catastrophic forgetting in neural networks. Proceedings of the National Academy of Sciences, 114(13):3521–3526, 2017.

[6] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In ICML, 2017.

[7] Gido M. van de Ven and Andreas S. Tolias. Three scenarios for continual learning. arXiv preprint arXiv:1904.07734, 2019.

[8] Zhizhong Li and Derek Hoiem. Learning without forgetting. IEEE T-PAMI, 40(12):2935–2947, 2017.

[9] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In CVPR, 2019.

[10] Prithviraj Dhar, Rajat Vikram Singh, Kuan-Chuan Peng, Ziyan Wu, and Rama Chellappa. Learning without memorizing. In CVPR, 2019.

[11] Arthur Douillard, Matthieu Cord, Charles Ollion, and Thomas Robert. PODNet: Pooled outputs distillation for small-tasks incremental learning. In ECCV, 2020.

[12] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. iCaRL: Incremental classifier and representation learning. In CVPR, 2017.

[13] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In NIPS, 2017.

[14] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative Adversarial Nets. In NIPS, 2014.

[15] Yan, Shipeng and Xie, Jiangwei and He, Xuming. DER: Dynamically Expandable Representation for Class Incremental Learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2021

[16] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in Advances in neural information processing systems, 2017, pp. 6402–6413.

[17] D. Molchanov, A. Lyzhov, Y. Molchanova, A. Ashukha, and D. Vetrov, "Greedy policy search: A simple baseline for learnable test-time augmentation," arXiv preprint arXiv:2002.09103, vol. 2, no. 7, 2020.

[18] Mohamed Abdelsalam, Mojtaba Faramarzi, Shagun Sodhani, and Sarath Chandar. IIRC: Incremental Implicitly-Refined Classification. In CVPR, 2021.

[19] Alex Krizhevsky. Learning multiple layers of features from tiny images. Unvieristy of Toronto: Technical Report, 2009.

[20] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition, pages 248–255. Ieee, 2009.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section 2.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Supplementary Materials.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Supplementary Materials.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See Supplementary Materials.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Supplementary Materials.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]