# Prototype Reminiscence and Augmented Asymmetric Knowledge Aggregation for Non-Exemplar Class-Incremental Learning (Supplementary Materials)

Wuxuan Shi[1], Mang Ye[1,2*],
[1]School of Computer Science, Wuhan University, Wuhan, China
[2] Hubei Luojia Laboratory, Wuhan, China
https://shiwuxuan.github.io/PRAKA-project

## 1. Explanation of Details

### 1.1. Formulation of Evaluation Metrics

We report average incremental accuracy and average forgetting in the main text. The average incremental accuracy $A_K$ calculates the average accuracy of all $K$ incremental phases (including the initial phase). For any phase $k$, the average accuracy over all seen classes is $a_k$. The average incremental accuracy can be expressed as:

$$A_K = \frac{1}{K+1}\sum_{k=0}^{K} a_k. \tag{1}$$

At phase $k$, the forgetting of task $j$ is quantified as:

$$f_j^k = \max_{l\in 0,\dots,k-1}\left(a_{l,j} - a_{k,j}\right), \forall j < k, \tag{2}$$

where $a_{n,m}$ is the accuracy of task $n$ after training the training of phase $m$. The average forgetting of the entire incremental learning process can be formulated as follows:

$$F_K = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{k-1}\sum_{j=0}^{k-1} f_j^k. \tag{3}$$

### 1.2. Standard Deviation of the Average Incremental Accuracy.

The results of the average incremental accuracy (*i.e.*, Table 1 in the main text) are obtained by averaging three replicate experiments, and we set a different random seed for each run. To illustrate the stability of our method, we report the standard deviation of these three results. As shown in Tab. 1, the random seed has little impact on the results of our approach.

### 1.3. Detailed Values of the Accuracy Curves

For comparison in subsequent work, we show the detailed values of the accuracy curves (*i.e.*, Figure 4 in the main text) in Tab. 2, Tab. 3 and Tab. 4.

### 1.4. Illustration of Datasets

CIFAR-100 [3] contains 60,000 images of 32×32 size, with a total of 100 classes, each class consisting of 500 training images and 100 test images. Tiny-ImageNet [4] consists of 200 classes, with 500 training photos, 50 validation images, and 50 test images per class, where the im-

Table 1. Quantitative comparisons of the average incremental accuracy (%) with other methods at different task number settings on CIFAR-100, TinyImageNet, and ImageNet-Subset. The red footnotes in the last row represent the standard deviation on three different runs.

| Methods | CIFAR-100 | | | TinyImageNet | | | ImageNet-Subset |
|---|---|---|---|---|---|---|---|
| | 5 phases | 10 phases | 20 phases | 5 phases | 10 phases | 20 phases | 10 phases |
| MUC [6] | 49.42 | 30.19 | 21.27 | 32.58 | 26.61 | 21.95 | 35.07 |
| SDC [8] | 56.77 | 57.00 | 58.90 | — | — | — | 61.12 |
| PASS [9] | 63.47 | 61.84 | 58.09 | 49.55 | 47.29 | 42.07 | 61.80 |
| SSRE [10] | 65.88 | 65.04 | 61.70 | 50.39 | 48.93 | 48.17 | 67.69 |
| Ours | **70.02**±0.22 | **68.86**±0.43 | **65.86**±0.35 | **53.32**±0.20 | **52.61**±0.02 | **49.83**±0.13 | **68.98**±0.15 |

Table 2. Detailed values of classification accuracy under the setting of 5 phases.

| Dataset | Phase | | | | | |
|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 |
| CIFAR-100 [3] | 81.90 | 74.85 | 70.34 | 66.81 | 64.71 | 61.55 |
| TinyImageNet [4] | 63.06 | 55.91 | 54.01 | 51.79 | 48.83 | 46.36 |

Table 3. Detailed values of classification accuracy under the setting of 10 phases.

| Datasets | Phase | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| CIFAR-100 [3] | 81.90 | 76.89 | 73.65 | 71.25 | 70.13 | 67.20 | 65.86 | 64.11 | 63.57 | 62.49 | 60.41 |
| TinyImageNet [4] | 63.06 | 57.65 | 54.68 | 52.88 | 53.19 | 51.72 | 50.75 | 49.69 | 48.23 | 46.69 | 45.16 |
| ImageNet-Subset [2] | 81.12 | 78.47 | 74.47 | 71.66 | 69.34 | 68.21 | 66.05 | 64.45 | 62.80 | 60.97 | 61.30 |

Table 4. Detailed values of classification accuracy under the setting of 20 phases.

| Datasets | Phase | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| CIFAR-100 [3] | 83.25 | 79.44 | 76.39 | 73.76 | 72.67 | 71.60 | 69.29 | 68.75 | 67.37 | 65.10 |
| TinyImageNet [4] | 63.06 | 58.17 | 57.87 | 56.28 | 55.03 | 53.31 | 52.57 | 52.24 | 51.27 | 50.03 |

| Datasets | Phase | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| CIFAR-100 [3] | 64.93 | 64.09 | 61.68 | 61.39 | 58.55 | 58.31 | 58.76 | 57.89 | 57.26 | 56.53 | 56.20 |
| TinyImageNet [4] | 49.60 | 47.99 | 48.10 | 46.33 | 46.52 | 45.09 | 44.53 | 43.52 | 42.69 | 41.79 | 40.58 |

age size is 64×64. It offers more phases and incremental classes for comparing the sensitivities of various approaches. ImageNet-Subset is a subset of 100 classes randomly extracted from ImageNet-1k [2] (random seed 1993). It has about 1300 training images and 50 test images per class. The image size of ImageNet-Subset is 256×256, which is much larger than those of the other two datasets. For the incremental configuration of classes for all datasets, please refer entirely to [9].

## 2. Analysis

### 2.1. Comparison of Weight in the FC Layer.

For further analysis of the role of our asymmetric knowledge aggregation (AKA), we present the norms of the weight vectors after all incremental phases are completed on CIFAR-100 (5 phases) in Fig. 1. As shown in Fig. 1 (a), is clear that the norms of the weight vectors vary greatly between classes in the baseline model. In Fig. 1 (b) and (c), the number of classes was increased to 400 by label augmentation. To demonstrate the role of self-supervised label augmentation, we take out the norms of the weights corresponding to the non-augmented classes (the part used for testing without AKA) to plot Fig. 1 (c). As we analyzed in the introduction section of the main text, vanilla SLA can make the classifier more balanced. However, there are many invalid parts in the weights of past classes. For comparison, we show the norms of the weight vectors in the refined classifier $\mathcal{G}_O$ obtained after AKA in Fig. 1 (d). The classifier learned with AKA has a minor variance of the norms, and its norms of the new classes are improved. Experiments demonstrate that AKA discards invalid weights to learn a more refined classifier and increases the attention of the new task.

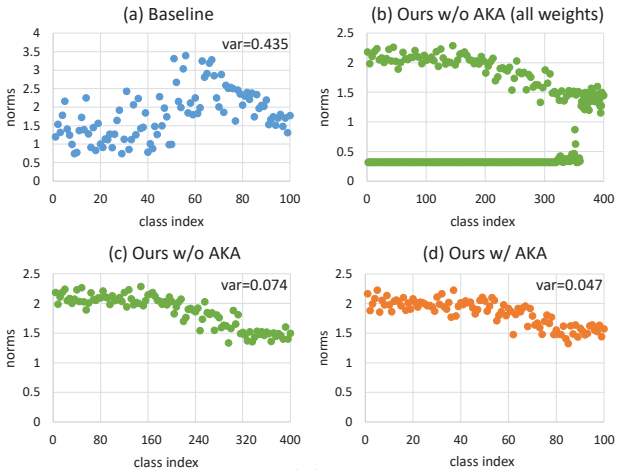


Figure 1. Norms of the weight vectors in the fully connected (FC) layer after learning all classes incrementally. (a) Weight norms of the classifier in the baseline model. (b) Weight norms of the classifier (the part used for testing w/o AKA) trained with SLA. (c) Weight norms of the non-augmented classes in the classifier trained with SLA. (d) Weight norms of the refined classifier $\mathcal{G}_O$ obtained by AKA. The "var" denotes the variance of all norms.

### 2.2. Ablation Study on Average Forgetting

As a supplement to section 4.3 of the main text, we present the ablation experiments with average forgetting as an evaluation metric. As shown in Tab. 5, prototype reminiscence (PR) significantly alleviates catastrophic forgetting in all cases. Self-supervised label augmentation (SLA) increases this metric to some extent by improving generalizability. Asymmetric knowledge aggregation (AKA) substantially improves the performance of the model on the new task, *i.e.*, increases $a_{l,j}$ in Equation Eq. (2) (with little impact on $a_{k,j}$), which causes a rise in the forgetting metric, but is harmless for the overall performance.

Table 5. Ablation study (in average forgetting) of our method on CIFAR-100 and TinyImageNet datasets.

| Components | | | CIFAR-100 | | | TinyImageNet | | |
|---|---|---|---|---|---|---|---|---|
| PR | SLA | AKA | 5 phases | 10 phases | 20 phases | 5 phases | 10 phases | 20 phases |
| | | | 37.35 | 39.00 | 44.29 | 27.82 | 38.75 | 47.93 |
| ✓ | | | 10.39 | 11.71 | 15.85 | 6.98 | 11.41 | 19.96 |
| | ✓ | | 31.39 | 31.48 | 33.21 | 23.37 | 29.84 | 37.78 |
| ✓ | ✓ | | **7.18** | **6.42** | **10.30** | **4.57** | **5.10** | **9.05** |
| ✓ | ✓ | ✓ | 12.59 | 14.65 | 17.39 | 11.84 | 13.95 | 18.51 |

Table 6. Quantitative comparisons of the average incremental accuracy (%) with other prototype augmentation approaches on CIFAR-100 dataset.

| Methods | CIFAR-100 | | |
|---|---|---|---|
| | 5 phases | 10 phases | 20 phases |
| Baseline | 56.27 | 51.02 | 43.98 |
| Over-sample [1] | 56.45 | 51.26 | 44.72 |
| Gaussian Noise [5] | 60.06 | 55.44 | 47.09 |
| PR | **66.21** | **63.80** | **57.31** |

Table 7. Results of SSRE [10] (in average incremental accuracy) with self-supervised label augmentation on CIFAR-100 dataset.

| Methods | CIFAR-100 | | |
|---|---|---|---|
| | 5 phases | 10 phases | 20 phases |
| SSRE | 65.88 | 65.04 | 61.70 |
| SSRE+SLA | 66.15 | 65.31 | 61.72 |

## 2.3. Comparison with Other Prototype Augmentation Approaches

To compare the different approaches fairly, we apply different augmentations on the baseline model (only use KD) for evaluation. The results on CIFAR-100 are summarized in Tab. 6. Over-sample is randomly sampling the same number of prototypes as the batch size as old class features to train the classifier. For the Gaussian Noise, following PASS [9], it is denoted as $e * r$, where $e \sim \mathcal{N}(0, 1)$. $r$ is computed in the first task as: $r^2 = \frac{1}{K_1 * D} \sum_{k=1}^{K_1} \mathrm{Tr}\left(\Sigma_{1,k}\right)$, where $K_1$ is the number of classes of the first task, $D$ is the dimension of the deep feature space, $\Sigma_{1,k}$ is the covariance matrix for the features from class $k$, and the Tr operation computes the trace of a matrix. As can be seen, our method has a clear advantage, and the gap grows further as the difficulty increases (the number of incremental phases increases).

## 2.4. Additional Results of Adding Self-supervised Label Augmentation to SSRE

Self-supervised label augmentation has been demonstrated to be beneficial for NECIL in our approach and PASS [9]. To make a fair comparison with the SOTA method, we add SLA to SSRE [10] and conduct experiments. As reported in Tab. 7, SLA has little effect on SSRE. It's because in our method and PASS, the feature extractor has a fixed structure. Past and current knowledge is stored in the same structure, hence learning a generalizable and transferable feature extractor reduces the hindrance of learning new tasks and the forgetting of old tasks. However, SSRE has a dynamic structure that includes a main branch and a side branch. When learning a new task, it only optimizes the side branch and merges it into the main branch after each phase of learning is completed. On the one hand, the main branch does not require generalization to new tasks because it does not actively learn new knowledge but relies on the integration of the side branch. On the other hand, the side branch is reset after learning a task, and thus the generalization ability to unknown tasks is inconsequential. Therefore, the improvement of generalization capability brought by SLA is of little help to SSRE.

Table 8. Quantitative comparisons of the final accuracy (%) under different settings (*i.e.*, 5, 10 and 20 phases). B0 indicates the number of base classes is zero, where all 100 classes are evenly divided into 5, 10, and 20 phases.

| Methods | CIFAR-100 (B0) | | |
|---|---|---|---|
| | 5 phases | 10 phases | 20 phases |
| ABD [7] | 43.90 | 33.70 | 20.00 |
| SSRE [10] | 44.60 | 34.39 | 23.12 |
| Ours | **51.73** | 39.45 | **33.51** |

Table 9. Quantitative comparisons of the average incremental accuracy (%) under different settings (*i.e.*, 5, 10 and 20 phases). B50 indicates the number of base classes is 50 (40 for the 20 phases setting), where the initial model is trained on the base classes, and the remaining classes are divided into 5, 10, and 20 phases for subsequent incremental processes. The B50 setting corresponds to the experimental configuration described in the main text.

| Methods | CIFAR-100 (B50) | | |
|---|---|---|---|
| | 5 phases | 10 phases | 20 phases |
| ABD [7] | 63.85 | 62.46 | 57.40 |
| SSRE [10] | 65.88 | 65.04 | 61.70 |
| Ours | **70.02** | **68.86** | **65.86** |

## 2.5. Results with Different NECIL Settings

To demonstrate the superiority of our suggested approach, we perform more comparison tests on CIFAR-100 with different NECIL settings. Smith *et al.* [7] proposes a data-free CIL setting, which is analogous to that of NECIL [9]. The setting adopted in [7] divides all 100 classes into 5 phases, 10 phases and 20 phases and is called B0. In contrast, the setting in [9] select the first 50 classes (40

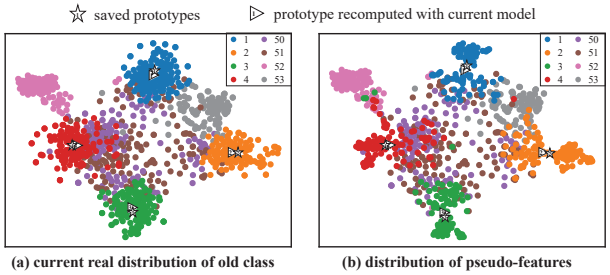


Figure 2. Visualization of the saved and real prototypes.

Table 10. Quantitative comparisons of the average incremental accuracy (%) on ImageNet-1K [2].

| Methods | PASS | SSRE | Ours |
|---|---|---|---|
| ImageNet-1K (5 phases) | 49.94 | 49.88 | **53.87** |
| ImageNet-1K (10 phases) | 46.94 | 47.89 | **50.22** |

classes for the 20 phases setting, which is for the subsequent classes to be evenly divided) as the 1-st phase and evenly split the remaining classes for $K-1$ phases, which is called B50. Following [10], we report the final accuracy under the B0 setting, *i.e.*, the average classification accuracy of the model for all seen classes after completing all phases of training. We report the average incremental accuracy under the B50 setting. Thanks to the previous work of Zhu *et al*. [10], we can compare our method with the SOTA data-free CIL methods [7] in both settings, respectively. As reported in Tab. 8 and Tab. 9, our method achieves superior performance under both the B0 and B50 settings, which indicates that the proposed prototype reminiscence and augmented asymmetric knowledge aggregation are effective for solving the NECIL problem.

### 2.6. Feature Drift

We visualize the saved prototypes and the real prototypes (computed with the current model and the old task data after finishing the learning of new task) in the Fig. 2. Prototypes drift to a certain extent but are acceptable. In addition, we show the real distribution of old class features. The comparison shows that our method can generate approximate distributions at the class boundaries.

### 2.7. Comparison on large-scale datasets

To further illustrate the superiority of our method, we conducted experiments on the large-scale ImageNet-1K dataset. As shown in Tab. 10, compared to the two SOTA methods PASS [9] and SSRE [10] (as both original papers were not experimented on ImageNet-1K, the results were reproduced from open source code), our approach makes a notable improvement.

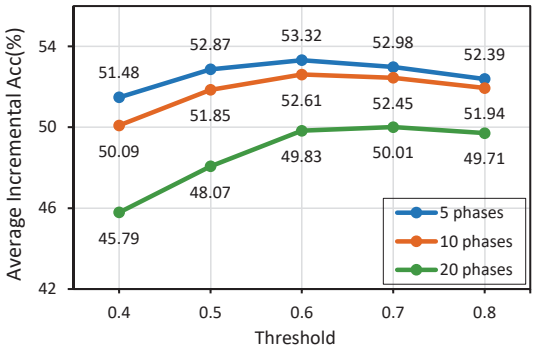


Figure 3. Influence of the threshold $\eta$ in prototype reminiscence.

### 2.8. Impact of the Threshold on TinyImageNet

In experiments on all datasets, we use a threshold of 0.6 and get promising results. As a supplement to Figure 5 in the main text, the impact of threshold $\eta$ on TinyImageNet is shown in Fig. 3. It exhibits a similar trend as on CIFAR-100. Fine-tuning the threshold may achieve better results.

## References

[1] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority oversampling technique. *JAIR*, pages 321–357, 2002. 3

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 2, 4

[3] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 1, 2

[4] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015. 1, 2

[5] Jialun Liu, Yifan Sun, Chuchu Han, Zhaopeng Dou, and Wenhui Li. Deep representation learning on long-tailed data: A learnable embedding augmentation perspective. In *CVPR*, pages 2970–2979, 2020. 3

[6] Yu Liu, Sarah Parisot, Gregory Slabaugh, Xu Jia, Ales Leonardis, and Tinne Tuytelaars. More classifiers, less forgetting: A generic multi-classifier paradigm for incremental learning. In *ECCV*, pages 699–716, 2020. 1

[7] James Smith, Yen-Chang Hsu, Jonathan Balloch, Yilin Shen, Hongxia Jin, and Zsolt Kira. Always be dreaming: A new approach for data-free class-incremental learning. In *ICCV*, pages 9374–9384, 2021. 3, 4

[8] Lu Yu, Bartlomiej Twardowski, Xialei Liu, Luis Herranz, Kai Wang, Yongmei Cheng, Shangling Jui, and Joost van de Weijer. Semantic drift compensation for class-incremental learning. In *CVPR*, pages 6982–6991, 2020. 1

[9] Fei Zhu, Xu-Yao Zhang, Chuang Wang, Fei Yin, and Cheng-Lin Liu. Prototype augmentation and self-supervision for incremental learning. In *CVPR*, pages 5871–5880, 2021. 1, 2, 3, 4

[10] Kai Zhu, Wei Zhai, Yang Cao, Jiebo Luo, and Zheng-Jun Zha. Self-sustaining representation expansion for non-exemplar class-incremental learning. In *CVPR*, pages 9296–9305, 2022. 1, 3, 4