

Supplementary Materials

Incremental Learning via Robust Parameter Posterior Fusion

Anonymous Authors

ABSTRACT

In the main manuscript, we propose a Robust Parameter Posterior Fusion (RP²F) framework, a novel approach for incremental learning that demonstrates promising performance. This supplementary appendix serves to enrich the description by providing supplementary analysis and details. Section 1 presents the pseudo code elucidating the operations of RP²F. Section 2 provides the mathematical proof of the MAP estimation in Eq. (10) described in the manuscript. Section 3 conducts empirical analyses to validate the Class-IL ability of RP²F. Section 4 explores the robustness of RP²F in the task order. Finally, Section 5 offers some details for carrying out experiments.

1 PSEUDO CODE

The training of the RP²F framework for task t involves a four-stage loop: 1) training the feature extractor θ_t ; 2) estimating the Hessian Λ_t surrounding θ_t ; 3) updating the fused posterior extractor $\theta_{1:t}$; 4) training the task-specific classifier τ_t . Within each task, the four-step iteration **repeats** multiple times until the model converges. We outline the training process of RP²F in Algorithm 1.

Algorithm 1: RP²F Training

Input: Datasets $\{D_1, \dots, D_T\}$

Output: Extractor $\theta_{1:T}$; Classifiers $\{\tau_1, \dots, \tau_T\}$

```

1 // Learning Task 1.
2 Initialize  $\theta_1, \tau_1$ 
3 for  $epoch\_id = \{1, \dots, n\_epochs\}$  do
4   // Refer to Eq. (19)
5    $\theta_1, \tau_1 \leftarrow$ 
6      $\theta_1, \tau_1 - \alpha \nabla_{\theta_1, \tau_1} L_{ce}(\tau_1 \circ \theta_1, D_1) + \lambda L_{robust\_pri}(\tau_1 \circ \theta_1)$ 
7  $\theta_{1:1} \leftarrow \theta_1$ 
8  $\Lambda_{1:1} \leftarrow \frac{\partial L_{ce}(\theta_1^* + \delta, D_1)}{\eta \partial \delta}$ 
9 // Learning Task 2, ..., T
10 for  $t = \{2, \dots, T\}$  do
11    $\theta_t \leftarrow \theta_{t-1}$ 
12   Initialize  $\tau_t$ 
13   for  $epoch\_id = \{1, \dots, n\_epochs\}$  do
14     // Refer to Eq. (20)
15      $\theta_t \leftarrow \theta_t - \alpha \nabla_{\theta_t} L_{ce}(\tau_t \circ \theta_t, D_t) + \lambda L_{robust\_pri}(\tau_t \circ \theta_t)$ 
16      $\Lambda_t \leftarrow \frac{\partial L_{ce}(\theta_t^* + \delta, D_t)}{\eta \partial \delta}$ 
17      $\theta_{1:t} \leftarrow \frac{\Lambda_{1:t-1} \theta_{1:t-1} + \Lambda_t \theta_t}{\Lambda_{1:t-1} + \Lambda_t}$ 
18      $\tau_t \leftarrow \tau_t - \alpha \nabla_{\tau_t} L_{ce}(\tau_t \circ \theta_{1:t}, D_t)$ 
19    $\Lambda_{1:t} \leftarrow \Lambda_{1:t-1} + \Lambda_t$ 
20 return  $\theta_{1:T}, \{\tau_1, \dots, \tau_T\}$ 

```

2 DETAILS ABOUT MAP ESTIMATION IN EQ. (10)

In the manuscript, Eq. (10) is dedicated to identify the optimal parameter $\theta_{1:t}^*$ that maximizes the posterior probability $P(\theta|D_1, \dots, D_t)$. This maximization process leverages the computational framework established in Eq. (8), where the posterior probability is expressed as a product of prior probabilities and likelihood functions corresponding to each task:

$$\begin{aligned} \theta_{1:t}^* &= \arg \max_{\theta} P(\theta|D_1, \dots, D_t) \\ &= \arg \max_{\theta} \prod_{t'=1}^t P(\theta)^{\frac{1}{t}} P(D_{t'}|\theta). \end{aligned} \quad (1)$$

Employing the Laplace approximation, we estimate $P(\theta)^{\frac{1}{t}} P(D_{t'}|\theta)$ using a Gaussian distribution $\mathcal{N}(\theta|\theta_{t'}^*, H_{t'}^{*-1})$. Consequently, the formulation for deriving $\theta_{1:t}^*$ becomes:

$$\begin{aligned} \theta_{1:t}^* &\approx \arg \max_{\theta} \prod_{t'=1}^t \frac{\sqrt{\det(\Lambda_{t'}^*)}}{(2\pi)^{k/2}} \exp\left(-\frac{1}{2}(\theta - \theta_{t'}^*)^\top \Lambda_{t'}^* (\theta - \theta_{t'}^*)\right) \\ &= \arg \max_{\theta} \log \left(\prod_{t'=1}^t \frac{\sqrt{\det(\Lambda_{t'}^*)}}{(2\pi)^{k/2}} \exp\left(-\frac{1}{2}(\theta - \theta_{t'}^*)^\top \Lambda_{t'}^* (\theta - \theta_{t'}^*)\right) \right) \\ &= \arg \max_{\theta} \sum_{t'=1}^t \left[\log \left(\frac{\sqrt{\det(\Lambda_{t'}^*)}}{(2\pi)^{\frac{k}{2}}} \right) - \frac{1}{2}(\theta - \theta_{t'}^*)^\top \Lambda_{t'}^* (\theta - \theta_{t'}^*) \right] \\ &= \arg \max_{\theta} \sum_{t'=1}^t \left[-\frac{1}{2}\theta^\top \Lambda_{t'}^* \theta + \theta^\top \Lambda_{t'}^* \theta_{t'}^* - \frac{1}{2}(\theta_{t'}^*)^\top \Lambda_{t'}^* \theta_{t'}^* \right], \end{aligned} \quad (2)$$

where k denotes the number of dimensions of θ .

By taking the derivative of Equation 2 with respect to the parameter vector θ , we obtain the MAP estimation for θ :

$$\begin{aligned} \frac{d}{d\theta} \left(\sum_{t'=1}^t \left[-\frac{1}{2}\theta^\top \Lambda_{t'}^* \theta + \theta^\top \Lambda_{t'}^* \theta_{t'}^* \right] \right) &= 0 \\ \sum_{t'=1}^t [-\Lambda_{t'}^* \theta + \Lambda_{t'}^* \theta_{t'}^*] &= 0 \\ \left(\sum_{t'=1}^t \Lambda_{t'}^* \right) \theta &= \sum_{t'=1}^t \Lambda_{t'}^* \theta_{t'}^* \\ \theta &= \frac{\sum_{t'=1}^t \Lambda_{t'}^* \theta_{t'}^*}{\sum_{t'=1}^t \Lambda_{t'}^*}. \end{aligned} \quad (3)$$

Finally, the expression for θ as the weighted average of $\theta_{t'}^*$, weighted by their respective precisions $\Lambda_{t'}^*$, is taken as the optimal

Table 1: Class incremental learning results (ACC %) on 10-split CIFAR-100 and 10-split Tiny-ImageNet. We report the average accuracy (%) and the corresponding standard deviation over five runs with random seeds, and the higher the better. (*) indicates the upper-bound model that is jointly trained with all tasks. (†) imply the results are quoted from the origin paper or [20], in which the standard deviations may not be provided (HCR, NPCL, and VDFD).

Methods	Venue	Buffer	10-split CIFAR-100	10-split Tiny-ImageNet
Joint*	-	-	70.31	58.07
DER [2]	NeurIPS2020	500	34.24±1.4	17.75±1.1
DER++ [2]	NeurIPS2020	500	36.52±2.0	19.38±1.4
HCR† [22]	KBS2022	500	33.71	22.05
NPCL† [3]	NeurIPS2023	500	37.43	15.29±1.0
Refresh(DER++)† [24]	ICLR2024	500	38.49±0.8	20.81±1.3
FNO(DER++)† [4]	ICLR2024	500	40.81±0.7	22.45±0.4
OWM [26]	NMI2019	-	27.63±0.5	15.30±0.3
DI [25]	CVPR2020	-	11.19±0.3	8.19±0.2
ILCOC [19]	CVPRW2021	-	22.19±1.6	15.78±0.4
EFT [21]	CVPR2021	-	35.99±1.0	26.89±0.6
PASS [29]	CVPR2021	-	31.80±0.7	28.48±0.6
ABD [15]	ICCV2021	-	33.30±0.3	15.80±0.4
FAS [11]	ICLR2022	-	25.79±0.7	24.29±0.3
SSRE† [30]	CVPR2022	-	30.40±0.7	22.93±1.0
DCPOC [18]	PR2023	-	25.80±0.4	19.75±0.1
VDFD† [7]	PAMI2023	-	38.38	26.21
ANCL† [5]	CVPR2023	-	29.77±1.1	22.58±0.8
TA† [20]	WACV2024	-	34.17±0.3	24.78±0.9
MIND [1]	AAAI2024	-	33.52±0.4	26.46±0.1
RP ² F(Ours)	-	-	42.33±0.7	28.82±0.3

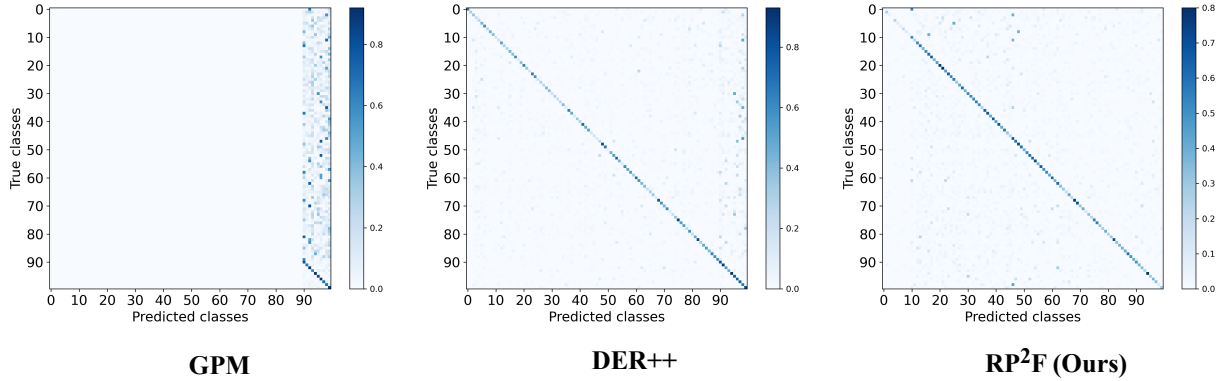


Figure 1: Confusion matrix of DER++, GPM, and RP²F on 10-Split CIFAR-100.

value of $\theta_{1:t}^*$:

$$\theta_{1:t}^* \approx \frac{\sum_{t'=1}^t \Lambda_{t'}^* \theta_{t'}^*}{\sum_{t'=1}^t \Lambda_{t'}^*}. \quad (4)$$

3 CLASS-IL ABILITY ANALYSIS

Different from Task-IL, Class Incremental Learning (Class-IL) [28] does not provide task identification during inference, and also garnered significant attention from the research community. Although Class-IL is not the primary focus of this study, we observe that our RP²F model is applicable to Class-IL, even under the more challenging conditions of the exemplar-free restriction [10].

Table 2: Hyperparameters for some baseline models and our RP²F. In this context, "KD" stands for "Knowledge Distillation"

Methods	Hyperparameters
LwF [8]	learning rate: 0.01 batch size: 32 temperature: 2 KD weight: 5e-4
SI [27]	learning rate: 0.1 (CIFAR-100), 0.05 (Tiny-ImageNet) batch size: 16 regularizer weight: 0.5
GEM [9]	learning rate: 0.03 (CIFAR-100), 0.05 (Tiny-ImageNet) batch size: 32 (CIFAR-100), 64 (Tiny-ImageNet)
DER [2]	learning rate: 0.03 batch size: 32 KD weight: 0.1
DER++ [2]	learning rate: 0.03 batch size: 32 KD weight α : 0.1 β : 0.5
EFT [21]	learning rate: 0.005 (CIFAR-100), 0.002 (Tiny-ImageNet) batch size: 64 (CIFAR-100), 16 (Tiny-ImageNet)
PASS [29]	learning rate: 1e-4 (Tiny-ImageNet), 5e-4 (CIFAR-100) batch size: 64 (CIFAR-100), 32 (Tiny-ImageNet) KD weight: 10 (Tiny-ImageNet), 0.2 (CIFAR-100) prototype weight: 0.05 (CIFAR-100), 0.5 (Tiny-ImageNet)
GPM [12]	learning rate: 0.02 (CIFAR-10), 0.01 (CIFAR-100), 0.02 (Tiny-ImageNet) batch size: 64 (CIFAR-10, CIFAR-100), 32 (Tiny-ImageNet)
Adam-NSCL [23]	learning rate: 1e-5 (CIFAR-10), 1e-4 (CIFAR-100), 5e-5 (Tiny-ImageNet) batch size: 32 (CIFAR-10, CIFAR-100), 16 (Tiny-ImageNet)
FAS [11]	learning rate: 0.001 (CIFAR-100), 5e-4 (Tiny-ImageNet) batch size: 32 (CIFAR-100), 16 (Tiny-ImageNet)
DCPOC [18]	learning rate: 5e-5 batch size: 8 (CIFAR-100), 32 (Tiny-ImageNet) λ_1 : 20 (CIFAR-100), 10 (Tiny-ImageNet) λ_2 : 10 (CIFAR-100), 0.01 (Tiny-ImageNet)
PRAKA [13]	learning rate: 0.001 batch size: 64 (CIFAR-100), 128 (Tiny-ImageNet) temperature: 0.1 (CIFAR-100), 0.2 (Tiny-ImageNet) KD weight: 20 (CIFAR-100), 15 (Tiny-ImageNet)
MIND [1]	learning rate: 0.005 (CIFAR-100), 0.002 (Tiny-ImageNet) batch size: 256 temperature: 6.5 (CIFAR-100), 12 (Tiny-ImageNet)
RP ² F (ours)	learning rate: 0.05 (CIFAR-100), 0.3 (Tiny-ImageNet) batch size: 32 (CIFAR-100), 1024 (Tiny-ImageNet) parameter-robustness regularizer weight λ : 1e-5

During the inference phase for Class-IL, for a given sample x without task identification, we concatenate all task-specific classifiers into a unified one $\tau^* = [\tau_1^* \dots \tau_T^*]$ to perform the prediction as follows:

$$\hat{y} = f(x; \tau^* \circ \theta_{1:T}^*). \quad (5)$$

To evaluate the Class-IL performance of the RP²F model, we compare it against several established Class-IL baselines, including DER [2], DER++ [2], HCR [22], NPCL [3], Refresh [24], FNO [4], OWM [26], DI [25], ILCOC [19], EFT [21], PASS [29], ABD [15], FAC [11], SSRE [30], DCPOC [18], VDFD [7], ANCL [5], TA [20], and MIND [1]. For exemplar-based methods, we provide them with an additional buffer that can store up to 500 samples. The results on 10-split CIFAR-100 and 10-split Tiny-ImageNet are presented in Table 1. As can be seen, our work also demonstrates competitive performance in the Class-IL setting. For the 10-split CIFAR-100, we achieve a classification accuracy of 42.33%, with a margin of 1.52%

over the second-best method. Similarly, as for the more challenging 10-split Tiny-ImageNet dataset, our work is 0.34% more accurate than the second-best method. All these results confirm the superior performance of RP²F in the Class-IL setting.

We further evaluate the Class-IL performance of RP²F under various lengths of task sequences. Following [15], we conduct experiments on CIFAR-100 with 5, 10, and 20 tasks. The comparison involved several methods including DGR [14], LWF [8], DI [25], ABD [15], PASS [29], DCPOC [18], and DLCPA [17], with their respective performances detailed in Table 3. RP²F outperformed all other methods in each setting, surpassing the second-best method by margins of 3.7%, 2.2%, and 3.8%, respectively. These results validate the robustness of RP²F for the length of the task sequence.

Additionally, we compute the confusion matrices of DER++, GPM, and RP²F, to analysis the balance of classifications in Class-IL. As illustrated in Figure 1, the classification of GPM tends to

Table 3: Class incremental learning results (ACC %) on CIFAR-100 for various task numbers (5, 10, 20).

Task Number	5 tasks	10 tasks	20 tasks
DGR[14]	14.4	8.1	4.1
LwF[8]	17.0	9.2	4.7
DI[25]	18.8	10.9	5.7
ABD[15]	43.9	33.7	20.0
PASS[29]	45.2	30.8	17.4
DCPOC[18]	33.1	27.5	20.5
DLCFA[17]	46.3	40.1	26.4
RP ² F (Ours)	50.0	42.3	30.2

the classes of the last task. This phenomenon is expected, as GPM is designed for Task-IL and lacks a specific design to counteract classification bias. For DER++, the phenomenon of classification bias is significantly reduced, thanks to the stored old-task exemplars. Significantly, the confusion matrix for RP²F displays pronounced diagonal concentration, indicating a robust balance in classification across tasks. This result underscores the effectiveness of RP²F in achieving equitable performance across all classes in the Class-IL setting.

Table 4: Incremental learning result (ACC %) of RP²F on a 10-split CIFAR-100 with five random task orders.

Methods	CIFAR-100	
	Class-IL	Task-IL
Task order 1	42.73	83.07
Task order 2	42.01	82.83
Task order 3	43.29	83.34
Task order 4	42.04	83.07
Task order 5	42.19	82.62
Average	42.45±0.49	82.99±0.24

4 ROBUSTNESS ANALYSIS ON TASK ORDER

This subsection investigates the robustness of RP²F to the task order. We randomly shuffle the task order on CIFAR-100 and retrain RP²F incrementally five times. Table 4 presents the results for each order and the average performance. As observed, RP²F exhibits relatively stable performance across different scenarios, indicating its robustness to task orders.

Table 5: Dataset statistics.

Dataset	CIFAR-100	Tiny-ImageNet
Input size	3 × 32 × 32	3 × 64 × 64
# Classes	100	200
# Training samples per class	450	450
# Validation samples per class	50	50
# Testing samples per class	100	100

5 EXPERIMENTAL DETAILS

This section provides details of the experimental setup. All experiments detailed in our manuscript and appendix were conducted on

a workstation running Ubuntu 16.04, equipped with 18 Intel Xeon 2.60GHz CPUs, 256 GB of memory, and 6 NVIDIA RTX3090 GPUs. Python 3.8 was used to implement all the methods.

Hyperparameters tuned for both the baseline methods, which were implemented by ourselves, and the proposed RP²F are summarized in Table 2. Additionally, statistical details of the datasets (CIFAR-100 [6] and Tiny-ImageNet [16]) are provided in Table 5.

REFERENCES

- [1] Jacopo Bonato, Francesco Pelosin, Luigi Sabetta, and Alessandro Nicolosi. 2024. MIND: Multi-task incremental network distillation. (2024).
- [2] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and SIMONE CALDERARA. 2020. Dark experience for general continual learning: A strong, simple baseline. In *Advances in Neural Information Processing Systems*, Vol. 33. 15920–15930.
- [3] Saurav Jha, Dong Gong, He Zhao, and Lina Yao. 2023. NPCL: Neural processes for uncertainty-aware continual learning. In *Advances in Neural Information Processing Systems*.
- [4] Hoyong Kim and Kangil Kim. 2024. Fixed non-negative orthogonal classifier: Inducing zero-mean neural collapse with feature dimension separation. In *International Conference on Learning Representations*.
- [5] Sanghwan Kim, Lorenzo Noci, Antonio Orvieto, and Thomas Hofmann. 2023. Achieving a better stability-plasticity trade-Off via auxiliary networks in continual learning. In *IEEE Conference on Computer Vision and Pattern Recognition*. 11930–11939.
- [6] Alex Krizhevsky. 2012. Learning multiple layers of features from tiny images. *University of Toronto* (05 2012).
- [7] Xiaorong Li, Shipeng Wang, Jian Sun, and Zongben Xu. 2023. Variational data-free knowledge distillation for continual learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 10 (2023), 1–17.
- [8] Zhizhong Li and Derek Hoiem. 2017. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40, 12 (2017), 2935–2947.
- [9] David Lopez-Paz and Marc Aurelio Ranzato. 2017. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, Vol. 30. 6467–6476.
- [10] Marc Masana, Xialei Liu, Bartłomiej Twardowski, Mikel Menta, Andrew D. Bagdanov, and Joost van de Weijer. 2022. Class-incremental learning: survey and performance evaluation on image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022), 1–20.
- [11] Zichen Miao, Ze Wang, Wei Chen, and Qiang Qiu. 2022. Continual learning with filter atom swapping. In *International Conference on Learning Representations*.
- [12] Gobinda Saha, Isha Garg, and Kaushik Roy. 2021. Gradient projection memory for continual learning. In *International Conference on Learning Representations*.
- [13] Wuxuan Shi and Mang Ye. 2023. Prototype reminiscence and augmented asymmetric knowledge aggregation for non-exemplar class-incremental learning. In *IEEE International Conference on Computer Vision*. 1772–1781.
- [14] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. 2017. Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems*, Vol. 30.
- [15] James Smith, Yen-Chang Hsu, Jonathan Balloch, Yilin Shen, Hongxia Jin, and Zsolt Kira. 2021. Always be dreaming: A new approach for data-free class-incremental learning. In *IEEE International Conference on Computer Vision*.
- [16] Stanford. 2015. Tiny imageNet challenge (CS231n). <http://tiny-imagenet.herokuapp.com/>. (2015). <http://tiny-imagenet.herokuapp.com/>.
- [17] Wenju Sun, Qingyong Li, Wen Wang, and Yangli-ao Geng. 2023. Towards plastic and stable exemplar-free incremental learning: A dual-learner framework with cumulative parameter averaging. *arXiv preprint arXiv:2310.18639* (2023).
- [18] Wenju Sun, Qingyong Li, Jing Zhang, Danyu Wang, Wen Wang, and Yangli ao Geng. 2023. Exemplar-free class incremental learning via discriminative and comparable parallel one-class classifiers. *Pattern Recognition* 140 (2023), 109561.
- [19] Wenju Sun, Jing Zhang, Danyu Wang, Yangli-ao Geng, and Qingyong Li. 2021. ILCO: An incremental learning framework based on contrastive one-class classifiers. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 3580–3588.
- [20] Filip Szatkowski, Mateusz Pyla, Marcin Przewięzlikowski, Sebastian Cygert, Bartłomiej Twardowski, and Tomasz Trzcinski. 2024. Adapt your teacher: Improving knowledge distillation for exemplar-free continual learning. In *IEEE Winter Conference on Applications of Computer Vision*. 1977–1987.
- [21] Vinay Kumar Verma, Kevin J Liang, Nikhil Mehta, Piyush Rai, and Lawrence Carin. 2021. Efficient feature transformations for discriminative and generative continual learning. In *IEEE Conference on Computer Vision and Pattern Recognition*. 13865–13875.
- [22] Qiang Wang, Jiayi Liu, Zhong Ji, Yanwei Pang, and Zhongfei Zhang. 2022. Hierarchical correlations replay for continual learning. *Knowledge-Based Systems* 250

- (2022), 109052.
- [23] Shipeng Wang, Xiaorong Li, Jian Sun, and Zongben Xu. 2021. Training networks in null space of feature covariance for continual learning. In *IEEE Conference on Computer Vision and Pattern Recognition*. 184–193.
- [24] Zhenyi Wang, Yan Li, Li Shen, and Heng Huang. 2024. A unified and general framework for continual learning. In *International Conference on Learning Representations*.
- [25] Hongxu Yin, Pavlo Molchanov, Jose M. Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K. Jha, and Jan Kautz. 2020. Dreaming to distill: data-free knowledge transfer via deepinversion. In *IEEE Conference on Computer Vision and Pattern Recognition*.
- [26] Guanxiong Zeng, Yang Chen, Bo Cui, and Shan Yu. 2019. Continual learning of context-dependent processing in neural networks. *Nature Machine Intelligence* 1, 8 (2019), 364–372.
- [27] Friedemann Zenke, Ben Poole, and Surya Ganguli. 2017. Continual learning through synaptic intelligence. In *International Conference on Machine Learning*. 3987–3995.
- [28] Da-Wei Zhou, Qi-Wei Wang, Zhi-Hong Qi, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. 2023. Deep class-incremental learning: a survey. *arXiv preprint arXiv:2302.03648* (2023).
- [29] Fei Zhu, Xu-Yao Zhang, Chuang Wang, Fei Yin, and Cheng-Lin Liu. 2021. Prototype augmentation and self-supervision for incremental learning. In *IEEE Conference on Computer Vision and Pattern Recognition*. 5871–5880.
- [30] Kai Zhu, Wei Zhai, Yang Cao, Jiebo Luo, and Zheng-Jun Zha. 2022. Self-sustaining representation expansion for non-exemplar class-incremental learning. In *IEEE Conference on Computer Vision and Pattern Recognition*. 9296–9305.