Supplementary Material of Knowledge Memorization and Rumination for Pre-trained Model-based Class-Incremental Learning

Zijian Gao, Wangwang Jia, Xingxing Zhang, Dulan Zhou,
Kele Xu, Feng Dawei, Yong Dou, Xinjun Mao, Huaimin Wang
{gaozijian19, wangwangjia, dulan_zhou, xukelele, yongdou, xjmao, hmwang}@nudt.edu.cn,
xxzhang1993@gmail.com, davyfeng.c@qq.com

1. Proof of the Theorem 1

Proof. In the main text, at task t, we have the learning problem

$$\underset{\boldsymbol{W}_t^{\mathrm{L}}}{\operatorname{argmin}} \left\|\boldsymbol{Y}_{1:t} - \boldsymbol{X}_{1:t}^{\mathrm{B}} \boldsymbol{W}_t^{\mathrm{L}}\right\|_F^2 + \gamma \left\|\boldsymbol{W}_t^{\mathrm{L}}\right\|_F^2,$$

which leads to optimal estimation

$$\hat{\boldsymbol{W}}_t^{\mathrm{L}} = \left(\boldsymbol{X}_{1:t}^{\mathrm{B}\ \mathrm{T}} \boldsymbol{X}_{1:t}^{\mathrm{B}} + \gamma \boldsymbol{I}\right)^{-1} \boldsymbol{X}_{1:t}^{\mathrm{B}\ \mathrm{T}} \boldsymbol{Y}_{1:t} = \boldsymbol{R}_t \boldsymbol{X}_{1:t}^{\mathrm{B}\ \mathrm{T}} \boldsymbol{Y}_{1:t}. \tag{1}$$

From the definition of the auto-correlation matrix, we can get the equation

$$\boldsymbol{R}_t = \left(\boldsymbol{R}_{t-1}^{-1} + \boldsymbol{X}_t^{\mathrm{B}\mathrm{T}} \boldsymbol{X}_t^{\mathrm{B}}\right)^{-1}.$$

According to the Woodbury matrix identity, we can complete the proof for the recursive formulation of $\boldsymbol{R}_t$ (i.B. equation 12 in the main text) as

$$\boldsymbol{R}_t = \boldsymbol{R}_{t-1} - \boldsymbol{R}_{t-1} \boldsymbol{X}_t^{\mathrm{B}\mathrm{T}} \left(\boldsymbol{I} + \boldsymbol{X}_t^{\mathrm{B}} \boldsymbol{R}_{t-1} \boldsymbol{X}_t^{\mathrm{B}\mathrm{T}}\right)^{-1} \boldsymbol{X}_t^{\mathrm{B}} \boldsymbol{R}_{t-1}. \tag{2}$$

We can break the estimation 1 into

$$\begin{aligned} \hat{\boldsymbol{W}}_t^{\mathrm{L}} &= \boldsymbol{R}_t \begin{bmatrix} \boldsymbol{X}_{1:t-1}^{\mathrm{B}\ \mathrm{T}} & \boldsymbol{X}_t^{\mathrm{B}\mathrm{T}} \end{bmatrix} \begin{bmatrix} \boldsymbol{Y}_{1:t-1} & 1 \\ 1 & \boldsymbol{Y}_t \end{bmatrix} \\ &= \left[\underbrace{\boldsymbol{R}_t \boldsymbol{X}_{1:t-1}^{\mathrm{B}\ \mathrm{T}} \boldsymbol{Y}_{1:t-1}}_{\text{the first term}} \quad \underbrace{\boldsymbol{R}_t \boldsymbol{X}_t^{\mathrm{B}\mathrm{T}} \boldsymbol{Y}_t}_{\text{the second term}} \right]. \end{aligned} \tag{3}$$

The first term obviously involves the historical data, while the second term does not. We aim to deal with it with the weight $\hat{\boldsymbol{W}}_{t-1}^{\mathrm{L}}$ of task $t-1$. Combing equation 2, we can rewrite the first term as

$$\begin{aligned} \boldsymbol{R}_t \boldsymbol{X}_{1:t-1}^{\mathrm{B}\ \mathrm{T}} \boldsymbol{Y}_{1:t-1} &= \boldsymbol{R}_{t-1} \boldsymbol{X}_{1:t-1}^{\mathrm{B}\ \mathrm{T}} \boldsymbol{Y}_{1:t-1} \\ &\quad - \boldsymbol{R}_{t-1} \boldsymbol{X}_t^{\mathrm{B}\mathrm{T}} \left(\boldsymbol{I} + \boldsymbol{X}_t^{\mathrm{B}} \boldsymbol{R}_{t-1} \boldsymbol{X}_t^{\mathrm{B}\mathrm{T}}\right)^{-1} \boldsymbol{X}_t^{\mathrm{B}} \boldsymbol{R}_{t-1} \boldsymbol{X}_{1:t-1}^{\mathrm{B}\ \mathrm{T}} \boldsymbol{Y}_{1:t-1} \\ &= \hat{\boldsymbol{W}}_{t-1}^{\mathrm{L}} - \boldsymbol{R}_{t-1} \boldsymbol{X}_t^{\mathrm{B}\mathrm{T}} \left(\boldsymbol{I} + \boldsymbol{X}_t^{\mathrm{B}} \boldsymbol{R}_{t-1} \boldsymbol{X}_t^{\mathrm{B}\mathrm{T}}\right)^{-1} \boldsymbol{X}_t^{\mathrm{B}} \hat{\boldsymbol{W}}_{t-1}^{\mathrm{L}} \end{aligned} \tag{4}$$

Let $\boldsymbol{A_t} = \left(\boldsymbol{I} + \boldsymbol{X}_t^{\mathrm{B}} \boldsymbol{R}_{t-1} \boldsymbol{X}_t^{\mathrm{B}\mathrm{T}}\right)^{-1}$, we have

$$\begin{aligned} \boldsymbol{I} = \boldsymbol{A_t} \boldsymbol{A_t}^{-1} &= \boldsymbol{A_t}(\boldsymbol{I} + \boldsymbol{X}_t^{\mathrm{B}} \boldsymbol{R}_{t-1} \boldsymbol{X}_t^{\mathrm{B}\mathrm{T}}) = \boldsymbol{A_t} + \boldsymbol{A_t}(\boldsymbol{X}_t^{\mathrm{B}} \boldsymbol{R}_{t-1} \boldsymbol{X}_t^{\mathrm{B}\mathrm{T}}) \\ &= \boldsymbol{A_t} + \left(\boldsymbol{I} + \boldsymbol{X}_t^{\mathrm{B}} \boldsymbol{R}_{t-1} \boldsymbol{X}_t^{\mathrm{B}\mathrm{T}}\right)^{-1} (\boldsymbol{X}_t^{\mathrm{B}} \boldsymbol{R}_{t-1} \boldsymbol{X}_t^{\mathrm{B}\mathrm{T}}) \end{aligned}$$

We can get $\boldsymbol{A_t} = \boldsymbol{I} - \left(\boldsymbol{I} + \boldsymbol{X}_t^{\mathrm{B}} \boldsymbol{R}_{t-1} \boldsymbol{X}_t^{\mathrm{B}\mathrm{T}}\right)^{-1} (\boldsymbol{X}_t^{\mathrm{B}} \boldsymbol{R}_{t-1} \boldsymbol{X}_t^{\mathrm{B}\mathrm{T}})$. Based on this, we take advantage of the recursive formulation $\boldsymbol{R}_t$ in equation 2 and rewrite equation 4 into

$$\begin{aligned} \boldsymbol{R}_t \boldsymbol{X}_{1:t-1}^{\mathrm{B}\ \mathrm{T}} \boldsymbol{Y}_{1:t-1} &= \hat{\boldsymbol{W}}_{t-1}^{\mathrm{L}} - \boldsymbol{R}_{t-1} \boldsymbol{X}_t^{\mathrm{B}\mathrm{T}} \left(\boldsymbol{I} - \left(\boldsymbol{I} + \boldsymbol{X}_t^{\mathrm{B}} \boldsymbol{R}_{t-1} \boldsymbol{X}_t^{\mathrm{B}\mathrm{T}}\right)^{-1} \boldsymbol{X}_t^{\mathrm{B}} \boldsymbol{R}_{t-1} \boldsymbol{X}_t^{\mathrm{B}\mathrm{T}}\right) \boldsymbol{X}_t^{\mathrm{B}} \hat{\boldsymbol{W}}_{t-1}^{\mathrm{L}} \\ &= \hat{\boldsymbol{W}}_{t-1}^{\mathrm{L}} - \left(\boldsymbol{R}_{t-1} - \boldsymbol{R}_{t-1} \boldsymbol{X}_t^{\mathrm{B}\mathrm{T}} \left(\boldsymbol{I} + \boldsymbol{X}_t^{\mathrm{B}} \boldsymbol{R}_{t-1} \boldsymbol{X}_t^{\mathrm{B}\mathrm{T}}\right)^{-1} \boldsymbol{X}_t^{\mathrm{B}} \boldsymbol{R}_{t-1}\right) \boldsymbol{X}_t^{\mathrm{B}\mathrm{T}} \boldsymbol{X}_t^{\mathrm{B}} \hat{\boldsymbol{W}}_{t-1}^{\mathrm{L}} \\ &= \hat{\boldsymbol{W}}_{t-1}^{\mathrm{L}} - \boldsymbol{R}_t \boldsymbol{X}_t^{\mathrm{B}\mathrm{T}} \boldsymbol{X}_t^{\mathrm{B}} \hat{\boldsymbol{W}}_{t-1}^{\mathrm{L}}. \end{aligned}$$

Hence, we can complete the proof and rewrite the estimation 3:

$$\hat{\boldsymbol{W}}_t^{\mathrm{L}} = \left[\underbrace{\hat{\boldsymbol{W}}_{t-1}^{\mathrm{L}} - \boldsymbol{R}_t \boldsymbol{X}_t^{\mathrm{B}\mathrm{T}} \boldsymbol{X}_t^{\mathrm{B}} \hat{\boldsymbol{W}}_{t-1}^{\mathrm{L}}}_{\text{all historical data}} \quad \underbrace{\boldsymbol{R}_t \boldsymbol{X}_t^{\mathrm{B}\mathrm{T}} \boldsymbol{Y}_t^{\text{train}}}_{\text{task data}} \right].$$

At task t, we can recursively obtain weight matrix $\hat{\boldsymbol{W}}_t^{\mathrm{L}}$ using only current task data, auto-correlation matrix $\boldsymbol{R}_{t-1}$ and the previous task weight $\boldsymbol{W}_{t-1}^{\mathrm{L}}$, the latter two representing all historical data. □

2. Pseudo Code

We outline the pipeline of our method in Algorithm 1. For the first task, an adapter is trained with cross-entropy loss, and knowledge is memorized using the analytical classification head via a least squares solution. For subsequent tasks, one adapter is trained while another is updated through weight interpolation. Following this, we facilitate the recursive memorization of new task knowledge and introduce the knowledge rumination mechanism, which refines and reinforces old knowledge using fine-grained features.

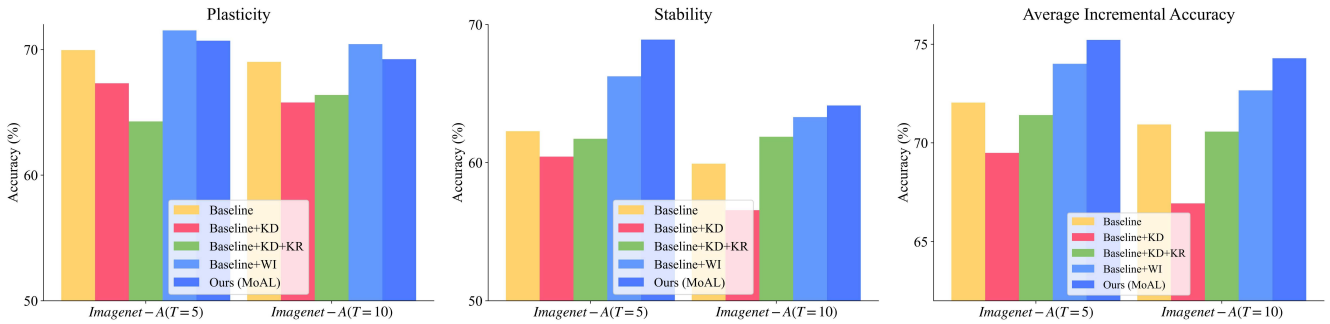

Figure 1. Detailed comparisons of different components in MoAL. The notations KD, KR, and WI represent Knowledge Distillation, Knowledge Rumination, and Weight Interpolation, respectively.

Algorithm 1 Momentum-based **A**nalytical **L**earning

Initialize: Training dataset $\mathcal{D}_t$ for task $t = 1, 2, \ldots, T$, pre-trained model weights θ_0, an analytical classification head (comprised of a random buffer layer f_{B} and a linear layer f_{L}) and prototype correction network f_{C}.
if task $t = 1$ **then**
  Initialize an adapter f_{θ_0}.
  Update weights θ_0 via cross-entropy loss L_{CE}.
  Collect expanded feature set $\boldsymbol{X}_1^{\mathrm{B}}$ via Eq. 2.
  Compute optimal weights $\boldsymbol{W}_1^{\mathrm{L}}$ via Eq. 4.
  Save auto-correlation matrix R_1.
  Save the prototypes of new classes.
end if
for each task $t \in \{2, \ldots, T\}$ **do**
  ## Momentum-based Adapter Weight Interpolation
  Initialize two adapters f_{θ_t} and $f_{\hat{\theta}}$ with weights θ_{t-1}.
  Update weights θ_t via L_{CE} and weights $\hat{\theta}$ via Eq. 5.
  ## Knowledge Memorization
  Collect expanded feature set $\boldsymbol{X}_t^{\mathrm{B}}$ via Eq. 2.
  Update auto-correlation matrix R_t via Eq. 10.
  Update weights $\boldsymbol{W}_t^{\mathrm{L}}$ via Eq. 9.
  Save the prototypes of new classes.
  ## Knowledge Rumination
  Train f_C via Eq. 11. and correct the old prototypes.
  Obtain fine-grained features $\hat{\boldsymbol{X}}_o^{\mathrm{fe}}$ via Eq. 12 and expanded features set $\hat{\boldsymbol{X}}_o^{\mathrm{B}}$ of old classes via Eq. 2.
  Update the auto-correlation matrix $\boldsymbol{R}_t$ via Eq. 15.
  Update the weight matrix $\boldsymbol{W}_t^{\mathrm{L}}$ via Eq. 14.
end for

3. Comparison to Replay-based Methods

In Table 1, we compare MoAL with competitive replay-based methods (storing 20 instances per class) using the same PTM. The results are excerpted from [8]. Even without saving instances, MoAL shows substantial improvements on CIFAR-100 and ImageNet-R, outperforming the best baseline methods by 4.35% and 5.58% on $\bar{A}$ and A_T for CIFAR-100, and by 3.11% and 4.85% on $\bar{A}$ and A_T for ImageNet-R, achieving superior results with a clear margin and further proving its effectiveness.

Table 1. Comparison to traditional replay-based CIL methods.

Method	Instances	CIFAR-100 ($T = 10$)		Imagenet-R ($T = 10$)	
		$\bar{A}$	A_T	$\bar{A}$	A_T
iCaRL [3]	20 / classes	82.46	73.87	72.96	60.67
DER [6]	20 / classes	86.04	77.93	80.48	74.32
FOSTER [5]	20 / classes	89.87	84.91	81.34	74.48
MEMO [7]	20 / classes	84.08	75.79	74.80	66.62
MoAL	0	**94.22**	**90.49**	**84.45**	**79.33**

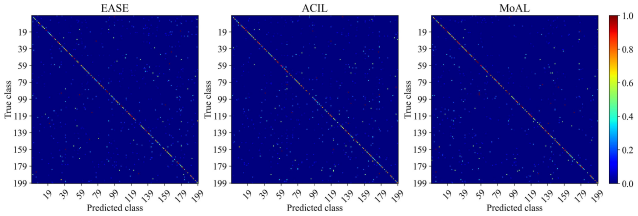

Figure 2. Confusion matrices of EASE, ACIL, and our MoAL for five incremental tasks on ImageNet-A.

4. Visual Comparison

To provide a visual comparison, we show the confusion matrix of EASE [8], ACIL [10], and ours on ImageNet-A. As illustrated in Figure 2, our method (MoAL) demonstrates a clearer diagonal pattern with reduced off-diagonal noise compared to EASE and ACIL, indicating higher classification accuracy and fewer misclassifications. The confusion matrix of MoAL reflects better stability and adaptability across tasks, effectively retaining old class knowledge while learning new classes. This result highlights MoAL's ability to maintain a balance between plasticity and stability, significantly outperforming the state-of-the-art methods.

Table 2. Comparison of last-task accuracy A_T using self-supervised PTMs (on ImageNet-1K) and supervised PTM (on ImageNet-21K).

PTMs	DINO- ImageNet-1K [1]								iBOT- ImageNet-1K [9]								ViT-B/16-IN21K			
Methods	ImageNet-A		ImageNet-R		CUB-200		Stanford Cars		ImageNet-A		ImageNet-R		CUB-200		Stanford Cars		CUB-200		Stanford Cars	
	$T=10$	5	10	5	10	5	10	5	10	5	10	5	10	5	10	5	10	5	10	5
Finetune	24.95	29.82	60.73	64.68	53.52	65.65	48.08	54.78	28.11	32.39	64.67	68.52	53.10	64.85	44.61	53.22	70.91	78.58	43.52	56.64
ACIL	32.32	40.55	69.30	70.28	73.58	79.01	60.47	61.75	39.83	47.07	72.32	74.07	79.18	81.38	64.73	72.97	88.17	88.25	66.74	76.25
CODA-Prompt	21.66	26.14	53.85	57.87	52.37	62.13	35.49	47.70	16.26	19.29	47.15	56.50	45.93	55.47	31.93	45.69	76.93	84.10	32.67	44.37
LAE	23.37	26.99	56.52	61.37	52.25	63.23	37.78	52.85	23.50	26.73	57.75	64.13	46.69	59.25	26.72	50.97	72.43	77.18	19.69	45.84
DS-AL	30.41	33.31	63.63	65.30	70.40	71.93	51.26	53.14	33.44	35.35	67.27	69.78	71.12	72.65	52.52	54.57	87.66	87.83	55.77	58.67
SimpleCIL	24.16	24.16	45.03	45.03	61.41	61.41	27.37	27.37	24.09	24.09	46.17	46.17	58.86	58.86	22.64	29.87	85.24	85.24	38.26	38.26
APER	28.04	32.78	62.05	64.00	65.9	70.06	30.44	37.56	32.78	40.88	65.12	67.77	66.62	70.82	29.78	41.61	86.01	86.22	45.86	53.51
EASE	34.69	38.64	67.80	69.95	65.18	72.94	39.21	50.98	43.91	45.69	72.15	73.10	67.77	73.92	45.75	54.78	83.12	85.11	37.68	44.76
MoAL (Ours)	**36.80**	**42.53**	**71.83**	**73.83**	**79.69**	**81.26**	**64.99**	**67.59**	**47.14**	**51.15**	**74.77**	**76.43**	**80.66**	**82.70**	**69.31**	**76.81**	**89.27**	**89.82**	**69.78**	**79.35**
	+2.11	+1.98	+2.53	+3.55	+6.11	+2.25	+4.52	+5.84	+3.23	+4.08	+2.45	+2.36	+1.48	+1.32	+4.58	+3.84	+1.10	+1.57	+3.04	+3.10

Table 3. Detailed plasticity and stability comparison.

Metrics	Stability				Plasticity			
Methods	ImageNet-A		CUB-200		ImageNet-A		CUB-200	
	$T=10$	5	10	5	10	5	10	5
ACIL	59.91	62.26	88.48	88.41	69.02	69.96	90.14	90.20
LAE	47.39	50.59	72.48	77.12	56.90	59.64	81.52	84.18
DS-AL	52.33	54.76	87.21	86.87	61.22	60.49	91.85	92.00
SimpleCIL	48.96	50.17	85.48	85.31	57.86	56.72	89.86	89.46
APER	55.17	61.28	86.35	86.41	63.30	66.09	90.53	90.25
EASE	50.33	58.31	82.40	84.10	59.61	68.15	91.71	91.60
MoAL (Ours)	**64.13**	**68.91**	**89.81**	**89.40**	**69.24**	**70.71**	**92.62**	**93.88**

Table 4. Ablation study on selective reinforcement.

Metrics	Settings	ImageNet-A		ImageNet-R	
		$T=10$	5	10	5
Stability	Full Reinforcement	63.49	68.93	79.22	81.91
	Selective Reinforcement	64.13	68.91	78.70	81.27
Plasticity	Full Reinforcement	56.93	60.01	76.58	79.15
	Selective Reinforcement	69.24	70.71	85.09	85.16
A_T	Full Reinforcement	62.48	64.65	78.12	79.88
	Selective Reinforcement	64.06	67.22	79.33	81.38
$\bar{A}$	Full Reinforcement	72.90	73.32	83.63	84.40
	Selective Reinforcement	74.29	75.22	84.45	85.39

5. More Results on Different PTM Backbones and Datasets

Table 2 presents experiments on well-known self-supervised backbones (DINO [1] and iBOT [9]) pre-trained on ImageNet-1K, and fine-grained datasets (CUB [4] and Stanford Cars [2]). MoAL always outperforms others, with even clearer advantages when the pre-trained and downstream datasets are more distinct. These further highlight the robustness and superiority of our method, emphasizing the necessity of incrementally improving PTM adaptivity.

6. Detailed Component Analysis

Here, to analyze the effect of different components of our method, we present a more detailed analysis of plasticity (average accuracy on new classes) and stability (average accuracy on old classes). As shown in Figure 1, the results reveal that knowledge distillation alone leads to a noticeable decline in both plasticity and stability, highlighting its limitations in improving model adaptability and its tendency to cause knowledge forgetting due to the unfrozen feature space. In contrast, our weight interpolation mechanism significantly enhances both plasticity and stability, demonstrating superior model adaptability and generalization compared to conventional knowledge distillation techniques widely used in CIL. Additionally, the knowledge rumination mechanism further boosts stability, effectively reinforcing old knowledge. By combining these components, MoAL achieves clear advantages over other methods, showcasing their mutual compatibility and synergistic effects.

7. Plasticity and Stability Comparison

As shown in Table 3, MoAL consistently outperforms existing methods in both plasticity and stability across all benchmarks, emphasizing the significance of our contributions.

8. Ablation study on selective reinforcement

In Table 4, we conduct a detailed ablation study using the ViT-B/16-IN21K model, comparing the performance of selective and full old knowledge reinforcement. In terms of stability, they yield similar performances, as selective reinforcement successfully ruminates the old task knowledge the model had not learned. However, full reinforcement harms plasticity due to the over-reinforcement of old knowledge. As a result, selective reinforcement clearly outperforms in terms of overall performance on A_T and $\bar{A}$. We also found that the plasticity loss in the full reinforcement setting is compensated in future tasks through the rumination process, allowing it to perform better than existing works. This further demonstrates the superiority of our proposed knowledge rumination module.

9. Multiple Seeds

In the main paper, we conduct experiments on various datasets using the random seed 1993. In this section, we repeat the experiments with different random seeds (1993, 1994, 1995, 1996) and present the accuracy curves of various methods in Figure 3. The results clearly show that our method consistently outperforms existing approaches by significant margins, demonstrating its effectiveness and robustness across different seed configurations.

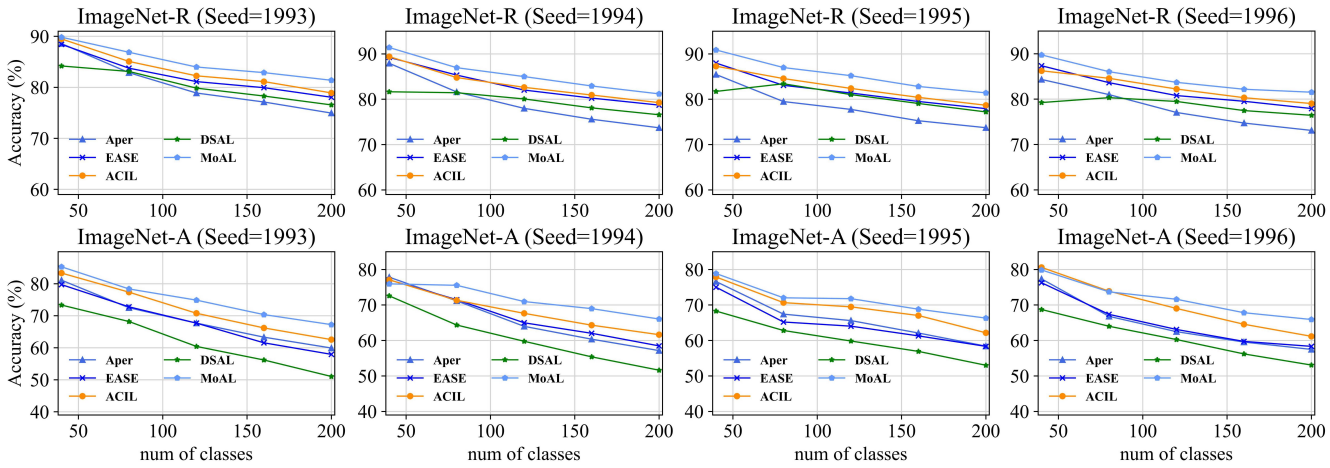

Figure 3. The average incremental accuracy $\bar{A}(\%)$ curves of various methods and different seeds.

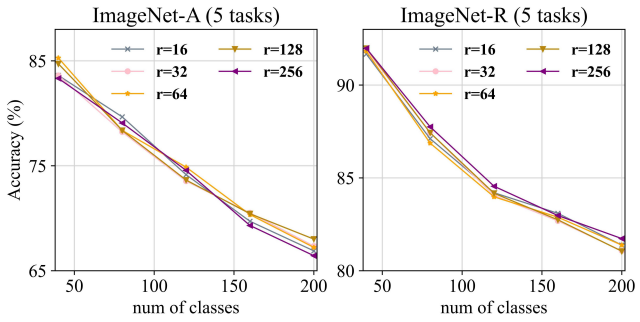

Figure 4. Ablation studies of the adapter projection dimension r.

10. Adapter Projection Dimension

We conduct ablation experiments on the projection dimension r in the adapter, as shown in Figure 4. The results on ImageNet-A and ImageNet-R reveal that increasing the dimension does not always lead to improved performance. On ImageNet-A, dimensions of 16 and 256 result in decreased performance, whereas dimensions of 32, 64, and 128 achieve better performance, albeit with a higher parameter count compared to 16. To strike a balance between performance and the number of training parameters, we select a projection dimension of 64 in our experiments. Moreover, these results highlight the robustness of MoAL, as it consistently outperforms existing methods even with a projection dimension as low as 16.

References

[1] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 3

[2] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013. 3

[3] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2001–2010, 2017. 2

[4] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 3

[5] Fu-Yun Wang, Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. Foster: Feature boosting and compression for class-incremental learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 398–414, 2022. 2

[6] Shipeng Yan, Jiangwei Xie, and Xuming He. Der: Dynamically expandable representation for class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3013–3022, 2021. 2

[7] Da-Wei Zhou, Qi-Wei Wang, Han-Jia Ye, and De-Chuan Zhan. A model or 603 exemplars: Towards memory-efficient class-incremental learning. In *International Conference on Learning Representations (ICLR)*, 2023. 2

[8] Da-Wei Zhou, Hai-Long Sun, Han-Jia Ye, and De-Chuan Zhan. Expandable subspace ensemble for pre-trained model-based class-incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 23554–23564, 2024. 2

[9] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. Image bert pre-training with online tokenizer. In *International Conference on Learning Representations*, 2021. 3

[10] Huiping Zhuang, Zhenyu Weng, Hongxin Wei, Renchunzi Xie, Kar-Ann Toh, and Zhiping Lin. Acil: Analytic class-incremental learning with absolute memorization and privacy protection. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 11602–11614, 2022. 2