Supplementary Material for "Learning to Imagine: Diversify Memory for Incremental Learning using Unlabeled Data"

Abstract

This supplementary material is for our main manuscript "Learning to Imagine: Diversify Memory for Incremental Learning using Unlabeled Data". We evaluate our method with a tighter memory budget and show that our method is robust to the size of memory. Besides, we put our generator at different positions of backbone to further analyze our method. Evaluation with different trade-off parameters and the ablation studies on ImageNet-Subset are also reported. To analyze the actual memory cost of the proposed method, we also count and analyze the storage cost of the proposed method.

1. Incremental Learning with Tighter Memory Budget.

We compare our method with other state-of-the-art methods with a memory size of 1000 in the main paper. To further verify the robustness and effectiveness of our method, we train our method under the condition that only 500 exemplars are available. Experimental results are shown in Table S1. Although only keeps 500 exemplars in memory, our method is still comparable to other methods with 1000 or even 2000 exemplars. Specifically, on CIFAR-100, compared with the recent method DDE [3] which uses 1000 exemplars, our method outperforms by 1.66% under the 5-step setting and performs 1.25% better under the setting of 10-step. Compared with advanced methods which keep 2000 exemplars, our performance is superior to DER, PODNet, UCIR and Icarl by 0.65%, 1.24%, 6.41% , 9.87% respectively under the setting of 5-step incremental learning. On ImageNet-Subset, our method also achieves comparable accuracy with other methods that have double or even quadruple memory budgets.

Method	CIFAR-100		ImageNet-Subset	
	5-step Avg Acc.	10-step Avg Acc.	5-step Avg Acc.	10-step Avg Acc.
Memory size = 2000				
Icarl [4]	56.29	52.42	65.04	68.72
UCIR [2]	59.66	55.77	70.84	68.09
PODNet [1]	64.83	64.03	75.54	74.58
DDE [3]	65.42	64.12	76.71	75.41
Ours	**68.01**	**66.47**	**77.20**	**76.76**
Memory size = 1000				
UCIR [2]	61.68	58.30	68.13	64.04
PODNet [1]	61.40	58.92	74.50	70.40
DDE [3]	64.41	62.20	71.20	69.05
Ours	**67.08**	**64.41**	**75.73**	**74.94**
Memory size = 500				
Ours	**66.07**	**63.45**	**69.20**	**66.95**

Table S1. Comparison of the proposed method with state-of-the-art methods at different budgets. We mark the best results at each memory size in **bold**.

2. Further Analysis on the Position of G

We argue that shallow layers of DNNs learn general knowledge while deeper layers learn more task-specific knowledge. The feature generator works well when it locates at the deeper layer of the backbone to tackle forgetting occurs at the deeper layers, but when it locates at too deep layers, the semantic decoupling learning would fail to extract semantic-irreverent information from unlabeled data. In the main paper, we introduce our feature generator G which is put after stage 3 of ResNet. We also conduct experiments of different positions of G, including putting G after stages 1, 2, 3 and 4 of ResNet, respectively. Results in Table S2 show that our method is effective when the generator is inserted after stages 2,3 and 4 of the ResNet, but it fails to tackle forgetting when it is inserted after stage 1 of ResNet. It is mainly because locating the generator at the

The position of G	Avg Acc.
After Stage1	44.92
After Stage2	60.68
After Stage3	66.47
After Stage4	66.31

Table S2. The impact of different positions of G. Ten steps incremental setting on CIFAR-100 is adopted.

too shallow layers will hinder it from alleviating the forgetting on the deep layers. Moreover, our method achieves the best performance when plugged after stage 3.

3. The Impact of Trade-off Parameters

Evaluation with different λ. In the main paper, λ is set to 2 for all experiments to defaults. We conduct experiments on both CIFAR-100 and ImageNet-Subset datasets to indicate the insensitivity to λ of our method. Similar to the main manuscript, we conduct the following experiments on 10-step setting. Other hyperparameters are kept to defaults. Results in Table S3 show that the proposed method consistently performs well when $\lambda \in [1, 5]$ on two datasets. When λ goes too large, the performance on CIFAR-100 is going down because the model mainly focuses on semantic-decoupling contrastive learning and leaves limited capacity for semantic contrastive learning.

	λ	1	2	5	10
Avg. acc.	CIFAR-100	66.24	66.47	66.20	65.31
	ImageNet-Subset	76.41	76.76	76.26	76.36

Table S3. Experiments under different λ. CIFAR-100 and ImageNet. During all experiments, we keep other parameters as defaults.

Evaluation with different λ_{cyc}. In the main paper, we set hyperparameter $\lambda_{cyc} = 0.3$ for all experiments. We conduct experiments of different λ_{cyc} on both CIFAR-100 and ImageNet-Subset datasets. All experiments are conducted on 10-step setting and we keep other parameters as defaults. Results are shown in Table S5. We can see that our method is not sensitive to hyperparameter λ_{cyc} on both CIFAR-100 and ImageNet-Subset datasets and works well when $\lambda_{cyc} \in [0.01, 0.5]$. When $\lambda_{cyc} > 0.5$, the performance starts to decline because the feature generator focuses too much on cycle constraint and hinders the semantic contrastive learning as well as semantic-decoupling contrastive learning.

Method	Avg Acc.
Baseline	70.59
+ $\mathcal{L}_{SC}$	76.35
+ $\mathcal{L}_{SC}$ + $\mathcal{L}_{SDC}$	76.48
+ $\mathcal{L}_{SC}$ + $\mathcal{L}_{SDC}$ + $\mathcal{L}_{SC}^{cyc}$ + $\mathcal{L}_{SDC}^{cyc}$	76.76

Table S4. Effectiveness of each objective function during training G. 'Baseline' denotes training model without G while other methods use G for feature generator. Experiments are conducted on ImageNet-Subset under the 10-step incremental setting.

4. Ablation Study on ImageNet-Subset

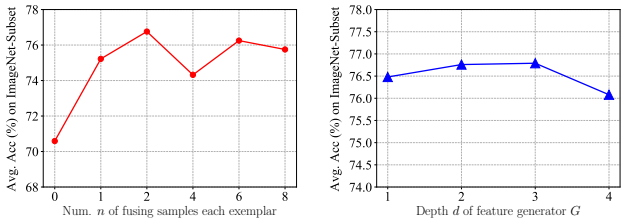
In the main manuscript, we conduct an ablation study on CIFAR-100. In this section, we conduct more experiments on ImageNet-Subset to further illustrate the effectiveness of our proposed objective function.

4.1. Evaluation with different objective functions

We conduct experiments to analyze the effect of $\mathcal{L}_{SC}$, $\mathcal{L}_{DSC}$, $\mathcal{L}_{SDC}^{cyc}$ and $\mathcal{L}_{SC}^{cyc}$ in the training process of G. Our baseline is to train the deep model with the task-specific training dataset D_i and the limited exemplar memory $\mathcal{M}$ in each task without feature generator G.

Results are shown in Table S4. Experiment results show that the proposed feature generator G boosts the performance about 6% only using loss $\mathcal{L}_{SC}$. Using $\mathcal{L}_{SDC}$, $\mathcal{L}_{SDC}^{cyc}$ as well as $\mathcal{L}_{SC}^{cyc}$ could further improve the performance.

4.2. Impact of hyperparameters n and d



(a) Experiments under different numbers of generated samples.

(b) Experiments on different depths of feature generator G.

Figure S1. We investigate the impact of the number of generated samples (a) and the impact of depth of feature generator G (b). All experiments are conducted on ImageNet-Subset and use ImageNet-900 as the unlabeled dataset.

In the main paper, n denotes the number of generated samples in each training batch and d denotes the number of residual blocks in generator G. Experimental results of different n on ImageNet-Subset are shown in Fig. S1a. From the figure, we can observe that when $n \in [1, 8]$, the accuracy of our method varies from 74% to 76%, which is higher than the baseline (n = 0) by a large margin. This indicates the effectiveness of our method. As for the depth d of the feature generator G, results shown in Fig. S1b indicate that our method is not sensitive to the depth of the generator G. As the depth d increases from 1 to 4, the average accuracy fluctuates within a narrow range, reaching approximately around 76%.

Discussion. When generating more samples per exemplar, the performance will drop slightly both on CIFAR-100 (see Fig.3 in the main paper) and ImageNet-Subset (see Fig. S1). It is mainly because generating more samples will

	λ_{cyc}	0.01	0.1	0.3	0.5	0.8	1	2
Avg. acc.	CIFAR-100	66.32	66.56	66.47	66.36	66.15	65.84	65.41
	ImageNet-Subset	76.60	76.40	76.76	76.63	76.36	76.45	76.41

Table S5. Experiments of the impact of λ_{cyc} on CIFAR-100 and ImageNet. During all experiments, we set other parameters to defaults.

Method	10 steps on ImageNet-Subset			
	#Exemplars	Mem. (exemplar)	Mem. cost (G)	Avg. Acc
PODNet [1]	20	2.38MB	0	74.58%
Ours-full	20	2.38MB	2.70MB	76.76%
Ours-half	20	1.19MB	2.70MB	74.94%
Ours-light	20	2.38MB	0.51MB	76.52%

Table S6. Memory cost (in MB) for exemplars and generators (FP32) of each class on ImageNet-Subset.

hinder model learning new tasks because all generated samples are considered as old-task samples, leading to an imbalance between old tasks and new tasks. To verify this, we generate more samples and re-weight the learning losses on old tasks and new tasks. Then, Avg. Acc increases to 66.28% on CIFAR-100 10-step setting with $n = 4$ and 66.37% with $n = 8$, which achieve similar performance with $n = 2$. Since each batch has different unlabeled data, the generated samples of the same exemplar are changing, thus $n = 2$ is enough.

5. Discussion about Memory Cost

To analyze the actual memory cost of the proposed method, we count the storage cost of the exemplars and our feature generator. Under the setting of ImageNet-SubSet, we counted the average storage space the exemplar of (saved images) in JPEG format and the storage cost for our proposed feature generator in FP32. As shown in Tab. S6, we compare our method with PODNet and our method outperforms PODNet at the cost of 2.70MB extra space for storing our generator. When we cut down half memory budge, our performance still outperforms PODNet ('Ours-half' in the table). Furthermore, the memory cost for feature generator G can be reduced by simplifying the residual block to a lighter structure: $Relu \rightarrow BN \rightarrow 1{\times}1Conv. \rightarrow BN$, in which $1{\times}1Conv.$ is used to fuse different channels come from two different feature maps (see Fig.2 in the main paper). When adopting such a lightweight generator, the performance barely drops, showing that our method is flexible ('Ours-light' in the table).

References

[1] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 1, 3

[2] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1

[3] Xinting Hu, Kaihua Tang, Chunyan Miao, Xian-Sheng Hua, and Hanwang Zhang. Distilling causal effect of data in class-incremental learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1

[4] 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)*, 2017. 1