Summarizing Stream Data for Memory-Constrained Online Continual Learning

Jianyang Gu[1,2], **Kai Wang**[2], **Wei Jiang**[1*], **Yang You**[2]

[1]Zhejiang University
[2]National University of Singapore
{gu_jianyang, jiangwei_zju}@zju.edu.cn, {kai.wang, youy}@comp.nus.edu.sg

Abstract

Replay-based methods have proved their effectiveness on online continual learning by rehearsing past samples from an auxiliary memory. With many efforts made on improving training schemes based on the memory, however, the information carried by each sample in the memory remains under-investigated. Under circumstances with restricted storage space, the informativeness of the memory becomes critical for effective replay. Although some works design specific strategies to select representative samples, by only employing a small number of original images, the storage space is still not well utilized. To this end, we propose to **S**ummarize the knowledge from the **S**tream **D**ata (SSD) into more informative samples by distilling the training characteristics of real images. Through maintaining the consistency of training gradients and relationship to the past tasks, the summarized samples are more representative for the stream data compared to the original images. Extensive experiments are conducted on multiple online continual learning benchmarks to support that the proposed SSD method significantly enhances the replay effects. We demonstrate that with limited extra computational overhead, SSD provides more than 3% accuracy boost for sequential CIFAR-100 under extremely restricted memory buffer. Code in https://github.com/vimar-gu/SSD.

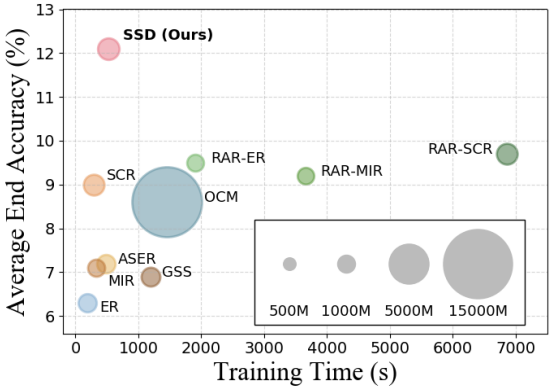

Figure 1: Under the restricted memory size of 100, the information contained in the memory is rather limited. By integrating the information from stream data into summarized samples, our proposed SSD largely enhances the replay effects. Experiments are conducted on the sequential CIFAR-100 benchmark (10 tasks). The scatter point size represents the GPU memory consumption at the training process.

Introduction

Continual learning (CL) aims to incrementally accumulate knowledge from a sequence of tasks (Gu et al. 2022; Li and Hoiem 2017; McCloskey and Cohen 1989; Ring 1998; Thrun 1998). Due to the explosive data collection by modern vision systems, storing all the images of past tasks for full fine-tuning is impractical for both computation and storage. Such restriction results in *catastrophic forgetting*, *i.e.*, adapting the model to new tasks causes knowledge forgetting and performance degradation of previous ones (McCloskey and Cohen 1989). In this paper, we focus on the more challenging online class-incremental CL setting, where each task consists of unique classes, with no task identity (Chaudhry et al. 2018b, 2019). Inside the task, the data is passed in a non-stationary stream, which can only be accessed once.

Recent literature addresses the forgetting problem from the perspectives of dynamic model architectures (Aljundi, Chakravarty, and Tuytelaars 2017; Fernando et al. 2017), parameter updating regularization (Kirkpatrick et al. 2017; Li and Hoiem 2017), pseudo sample generation (Shin et al. 2017) and memory-assisted replay (Lopez-Paz and Ranzato 2017; Rebuffi et al. 2017). Among them, replay-based methods maintain a small auxiliary memory containing images from past tasks. By mixing them into the current training process (Chaudhry et al. 2019), it provides direct accessibility to the past knowledge. Such a scheme brings stable performance improvements on relatively small computational overhead. Many efforts have been made on improving the training schemes to better leverage the memory (Mai et al. 2021; Zhang et al. 2022). However, the informativeness of samples in the memory remains under-investigated.

Under circumstances with restricted storage space, the information carried by each sample in the memory becomes critical to the replay effects. There are some works devoted to selecting proper data from the stream for the memory (Aljundi et al. 2019b; Shim et al. 2021). But by only employing a small amount of original images, the storage space is still underutilized. Efficiently condensing more informa-

*Corresponding Author

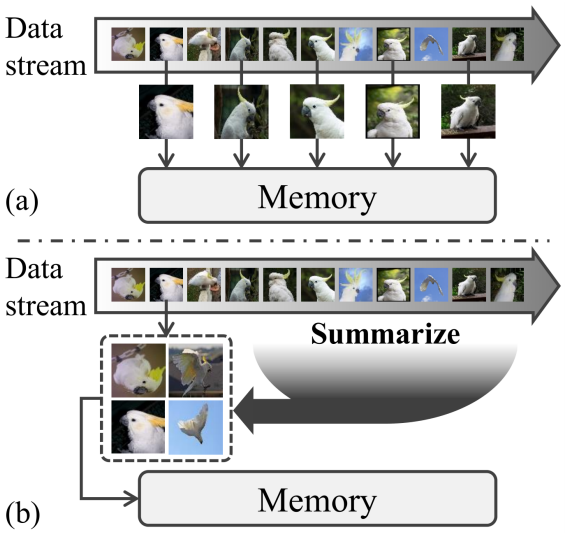

Figure 2: Concept Comparison: (a) Previous CL methods select original images from the stream data to form the auxiliary memory. (b) We propose to summarize stream data into informative samples to enhance the replay effects.

tion into the memory poses a challenging yet promising direction for further enhancing online continual learning.

In this work, we propose a novel Summarizing Stream Data (SSD) method to integrate the information from original images into a small amount of more informative samples. Specifically, initialized with first images in each task, the summarized samples are updated by sequentially distilling the knowledge from real images in the data stream. Firstly, the training gradients of real and summarized samples on the same network are matched.Thereby training the summarized samples provides similar parameter update results to the original images. Secondly, we employ the samples of previous tasks in the memory to help fit the overall distribution and provide more proper gradient supervision. Besides, the consistency on the relationship to the previous samples between real and summarized samples also serves as a constraint for establishing better distribution in the memory. By maintaining the consistency on multiple aspects, the summarized samples are more representative for the task data compared with original images of the same amount.

The proposed SSD method significantly enhances the replay effects by constructing the auxiliary memory with more informative samples. As shown in Fig. 1, under circumstances with restricted memory sizes, the information lack largely influences the replay effects when only original data is employed. With limited extra computational overhead, the proposed SSD method boosts the average end accuracy by more than 2%. We also show that SSD universally improves the performance for multiple popular CL baselines.

Related Works

Continual Learning requires a model to consecutively learn new tasks without forgetting the learned knowledge for previous tasks (McCloskey and Cohen 1989; Ring 1998; Thrun 1998). Due to the restricted access to the past data, the *catastrophic forgetting* issue becomes the major obstruction from practical appliance. Recent literature mainly addresses the problem by designing dynamic model architectures (Aljundi, Chakravarty, and Tuytelaars 2017; Fernando et al. 2017; Kang et al. 2022), regularizing the parameter updating (Kirkpatrick et al. 2017; Lee et al. 2017; Li and Hoiem 2017), generating pseudo samples (Shin et al. 2017; Smith et al. 2021; Xiang et al. 2019), or employing auxiliary memories for storing previous information (Lopez-Paz and Ranzato 2017; Rebuffi et al. 2017; Tiwari et al. 2022). Based on the differences of data increment form, CL is divided into task-incremental and class-incremental settings. The *task-incremental* setting provides the task identity, according to which different networks can be applied. For the *class-incremental* setting, the task identity is not provided, and different tasks share no overlapping classes. In this paper, we focus on the challenging class-incremental setting.

Online Continual Learning focuses on a more practical setting where the non-stationary data stream is passed only once for training (Chaudhry et al. 2018b; Pham, Liu, and Steven 2022; Sun et al. 2022; Prabhu et al. 2023; Harun et al. 2023; Ghunaim et al. 2023). Replay-based methods achieve noticeable results in online CL tasks (Aljundi et al. 2019a; Chaudhry et al. 2019). Chaudhry *et al.* employ an auxiliary memory to store previous images, and re-introduce the information into the current model training (Chaudhry et al. 2019). Aljundi *et al.* select the samples that are most interfered by new incoming samples for replay (Aljundi et al. 2019a). DER and SCR incorporate prediction logits matching and contrastive learning for retaining the previous knowledge (Buzzega et al. 2020; Mai et al. 2021). Most recently, some materials further improve the performance from the perspectives of better training schemes (Gu et al. 2022; Guo, Liu, and Zhao 2022; Zhang et al. 2022).

With many efforts made on improving the replay training techniques, the information carried by the images inside the memory is often overlooked. GSS (Aljundi et al. 2019b) selects samples with variant gradient direction to form the memory. Shim *et al.* utilize the Shapley Value as the sample selection criterion (Shim et al. 2021). In this work, we mainly focus on improving the informativeness of the memory, and propose to summarize the knowledge of the stream data into informative samples for more effective replay.

Dataset Distillation aims to integrate the information from large-scale datasets into much smaller ones for reducing the storage and calculation burdens. Dataset distillation (DD) methods are roughly divided into two categories: meta learning methods (Nguyen, Chen, and Lee 2020; Nguyen et al. 2021; Loo et al. 2022; Wang et al. 2018), matching-based methods (Kim et al. 2022; Wang et al. 2022; Zhao, Mopuri, and Bilen 2021; Zhao and Bilen 2023; Liu et al. 2023b; Lu et al. 2023; Du et al. 2023; Liu et al. 2023a) and generative methods (Cazenavette et al. 2022; Wang et al. 2023; Gu et al. 2023). By distilling the information from the whole dataset into several synthetic images, the informativeness of single sample is largely improved. Sangermano *et al.* employs DD on CL to linearly combine the original im-

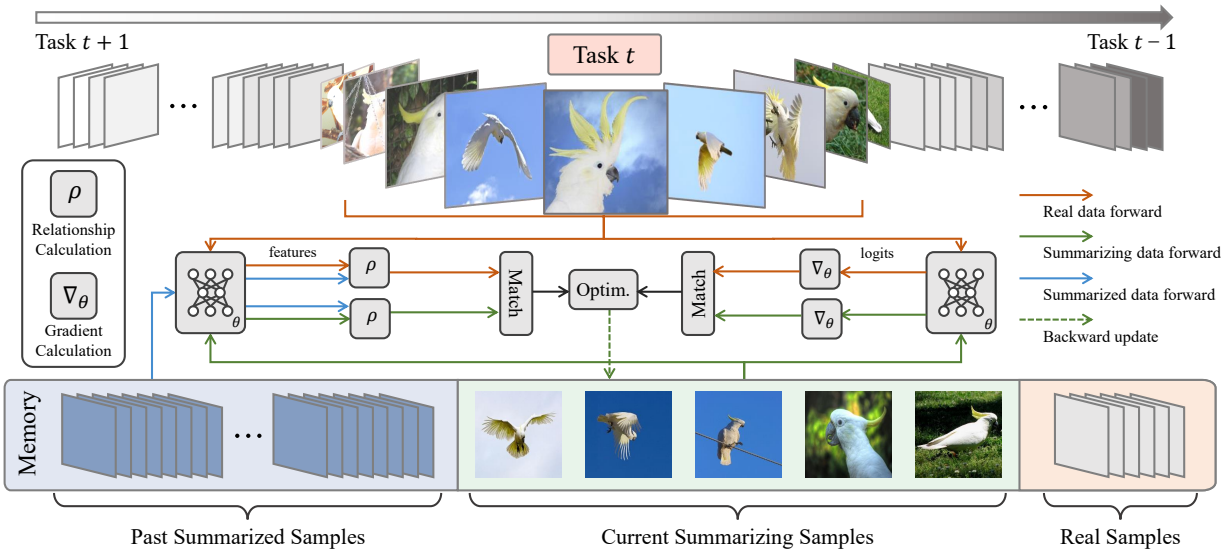

Figure 3: The pipeline of our proposed Summarizing Stream Data (SSD) method. The memory is composed of the summarized samples of previous tasks (Past Summarized Samples), the samples of the current task that are being summarized (Current Summarizing Samples) and real samples. For the current summarizing samples, both the training gradients and the relationships to the past knowledge are constrained to be consistent with real images.

ages into a small set (Sangermano et al. 2022). However, the method is only eligible for datasets with simple image structures and similar backgrounds. Inspired by dataset distillation methods and the properties of online CL, we propose to summarize stream data into more informative samples.

Method

Preliminaries

In this paper, we focus on the online class-incremental continual learning (CL) setting. Consider a sequence of tasks $\mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_N\}$, each of which consists of unique data and classes $\mathcal{T}_t = \{(x_{ti}, y_{ti})\}_{i=1}^{I_t}$, where N is the total task number and I_t is the data number of task $\mathcal{T}_t$. The class-incremental setting indicates that $\{x_{ti}\} \cap \{x_{tj}\} = \emptyset$ and $\{y_{ti}\} \cap \{y_{tj}\} = \emptyset$ for $\forall j \neq i$. For the offline CL setting, each task $\mathcal{T}_t$ can be accessed for multiple epochs to reach convergence. Yet for the online setting, each data inside the task (x_{ti}, y_{ti}) is passed only once in a non-stationary stream, with no task identity provided. The target of continual learning is to maximize the overall accuracy on all the tasks.

Replay-based methods maintain a small auxiliary memory $\mathcal{M} = \{(x_m, y_m)\}_{m=1}^{K}$ to store the images from past tasks, where K is the memory size. As shown in Fig. 2 (a), most of previous methods select original samples in the stream data to form the memory. During each iteration of current task, a batch of samples $\mathcal{B}_m = \{(x_{mi}, y_{mi})\}_{i=1}^{B_m}$ is sampled from the memory randomly or under certain strategies for joint training with current data $\mathcal{B}_t = \{(x_{ti}, y_{ti})\}_{i=1}^{B_t}$, where $B_m \leq K$ and B_t are the mini-batch size of the replay data and the current task, respectively. The joint training process is formulated as follows,

$$\phi^* = \arg\min_{\phi} \mathcal{L}_t(\phi; \mathcal{B}_t) + \lambda \mathcal{L}_m(\phi; \mathcal{B}_m), \qquad (1)$$

where ϕ is the parameters of the training model, $\mathcal{L}_t(\cdot;\cdot)$ is the standard training objective for the task, $\mathcal{L}_m(\cdot;\cdot)$ is the objective for the replay, and λ is a coefficient hyper-parameter for the replay objective. By simultaneously viewing the data from the current and past tasks, the model can obtain new knowledge while reduce the catastrophic forgetting for the past. However, when the storage space is restricted, the memory informativeness becomes critical for effective replay. And we argue that by only employing the original images, the storage space is still underutilized. Therefore, we propose to Summarize Stream Data (SSD) so that the limited memory contains richer information regarding to the whole data stream.

Summarizing Stream Data

Instead of selecting original samples from the stream data, as shown in Fig. 2 (b), we propose to integrate the information from the whole stream data into several informative images. By maintaining summarized samples that better represent the corresponding data distribution, the more informative auxiliary memory helps improve the replay effects. Even under restricted memory size, the proposed SSD method provides stable performance improvements on the average accuracy over the past tasks. The detailed pipeline of the proposed SSD method is presented in Fig. 3.

Dynamic memory. We first design a dynamic memory to maintain both summarized and original samples. When the data stream brings in new classes, a small specified number of memory will be initialized with the first images of the corresponding classes. In previous replay-based methods, images stored in any memory position can be replaced with new ones. As we intend to summarize as much information from the data stream as possible, these initial-

Slices	CIFAR-100			
	100×1	50×2	20×5	10×10
Acc	**51.6**$_{\pm 0.4}$	51.2$_{\pm 0.4}$	50.7$_{\pm 0.6}$	48.9$_{\pm 0.6}$

Table 1: Evaluation results of images separately distilled from different slices on CIFAR-100. Experiments conducted under the setting of 50 images per class.

ized positions are fixed to only be updated with summarizing information instead of direct replacement. For the other memory positions, standard reservior update is conducted as in (Chaudhry et al. 2019). Thereby the memory space is not wasted when the experienced task number is small. Under circumstances with restricted space, it is important to keep the example number per class balanced, as in (Rebuffi et al. 2017; Yoon et al. 2022). Thus the initialization position number for each class k is pre-defined to be the memory size K divided by the total class number. For convenience, we define the summarized images and original images in the memory as $\mathcal{M}_S$ and $\mathcal{M}_O$ in the follows.

Data summarizing. The main purpose of the proposed SSD method is to increase the richness of the information carried by the stored samples and to minimize the distance between the memory and the original data distribution to obtain the optimized memory $\mathcal{M}^*$:

$$\mathcal{M}^* = \arg\min_{\mathcal{M}} \mathbf{D}(\mathcal{M}, \mathcal{T}), \tag{2}$$

where $\mathbf{D}(\cdot, \cdot)$ represents the distance metric. We expect the informative samples to have similar training responses to the whole original image set. Inspired by dataset distillation methods, given the stream data, we implement the summarizing process by sequentially distilling the knowledge from mini-batches of real data into the summarized samples of the corresponding classes (Zhao, Mopuri, and Bilen 2021).

We define the summarized samples of class c as $\mathcal{M}_c$, whose size is pre-defined as k, and the data of the same class in the current mini-batch as $\mathcal{B}_c$. The optimization target of the summarizing process is to minimize the distance between $\mathcal{M}_c$ and $\mathcal{B}_c$. DC (Zhao, Mopuri, and Bilen 2021) proposes a simple yet effective metric that is the distance of the training gradients on the same network between the summarized and original images. By matching the model updating metrics throughout the stream data, the summarized samples are updated to approach the training effects of the whole real dataset. The gradient matching objective is formulated as follows,

$$\mathcal{L}_g = \mathbf{D}\left(\nabla_\theta \mathcal{L}_t^{'}(\theta; \mathcal{M}_c), \nabla_\theta \mathcal{L}_t^{'}(\theta; \mathcal{B}_c)\right), \tag{3}$$

where $\mathcal{L}_t^{'}(\cdot; \cdot)$ here is a training objective that differs from the standard training objective in Eq. 1, and θ is the parameters of the model employed for summarizing.

In order to save extra computational overhead, the summarizing is conducted every τ iterations. We provide the ablation study on the summarizing interval in Fig. 4d.

Past-assisted Summarizing

In addition to the training gradients, we propose to take better advantage of the past accumulated knowledge to help the information summarizing of the current task.

Summarizing model training. In the standard dataset distillation process, the gradient matching is accompanied with model updating in an alternative manner. As complete a training process as possible is simulated in order that the gradients of different training stages can be matched. The parameters θ is first randomly initialized. At each training iteration where a new batch of stream data $\mathcal{B}_t$ is presented, θ is updated as follows,

$$\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_t^{'}(\theta; \mathcal{B}_t), \tag{4}$$

where η is the learning rate for the summarizing model. Only the real images participate in the update of the summarizing model, which prevents the knowledge leakage of the summarized samples.

For the continual learning, however, the class number is not fixed throughout the training process. Thus, the model has to be re-initialized when new classes come. As we conduct the experiments under the class-incremental setting, past classes will be no more presented in the stream data. If the summarizing model only constructs the decision boundaries for the current classes, the provided training gradients will lose insights on the past knowledge. We provide an empirical evidence here of conducting dataset distillation on different slices of CIFAR-100 in Tab. 1. It suggests that the model trained with more classes at the same time provides more proper gradient supervision.

Therefore, we propose to employ the real images in the memory $\mathcal{M}_O$ to assist the summarizing model update:

$$\theta \leftarrow \theta - \eta \nabla_\theta \left(\mathcal{L}_t^{'}(\theta; \mathcal{B}_t) + \mathcal{L}_t^{'}(\theta; \mathcal{M}_O)\right). \tag{5}$$

Relationship matching. Since the model has obtained rough sketch on the past data, we further introduce a relationship matching to enhance the summarizing effects. Setting the extracted features of previous summarized samples as anchors, we explicitly constrain the mean features of summarized samples and real images to have consistent relationship to the anchors. The objective for relationship matching is formulated as:

$$\mathcal{L}_r = \mathbf{D}\left(\rho(\mathcal{M}_c, \mathcal{M}_S \backslash \mathcal{M}_c, \theta), \rho(\mathcal{B}_c, \mathcal{M}_S \backslash \mathcal{M}_c, \theta)\right), \tag{6}$$

$$\rho(\mathcal{X}, \mathcal{Y}, \theta) = \mathbf{D}\left(\overline{\mathcal{F}(\theta; \mathcal{X})}, \mathcal{F}(\theta; \mathcal{Y})\right), \tag{7}$$

where ρ represents the relationship calculation, $\mathcal{M}_S \backslash \mathcal{M}_c$ stands for the other summarized samples except for $\mathcal{M}_c$, $\mathcal{F}(\cdot; \cdot)$ represents the feature extraction process, and $\overline{\mathcal{F}}$ is the mean feature. The relationship consistency helps establish better overall distribution of summarized samples inside the memory. Combining the gradient matching, the complete summarizing objective $\mathcal{L}_s$ is formulated as:

$$\mathcal{L}_s = \mathcal{L}_g + \gamma \mathcal{L}_r, \tag{8}$$

where γ is the coefficient for relationship matching.

Method	Mini-ImageNet			CIFAR-100			Tiny-ImageNet		
	K=100	K=500	K=1000	K=100	K=500	K=1000	K=200	K=1000	K=2000
fine-tune		$4.2_{\pm 0.3}$			$5.8_{\pm 0.2}$			$2.3_{\pm 0.2}$	
iid offline		$47.7_{\pm 0.2}$			$50.2_{\pm 0.3}$			$25.9_{\pm 0.4}$	
EWC++ (Chaudhry et al. 2018a)		$4.7_{\pm 0.5}$			$5.2_{\pm 0.4}$			$2.1_{\pm 0.3}$	
LwF (Li and Hoiem 2017)		$9.6_{\pm 0.6}$			$12.2_{\pm 0.6}$			$6.0_{\pm 0.3}$	
AGEM (Chaudhry et al. 2018b)	$5.0_{\pm 0.5}$	$4.5_{\pm 0.4}$	$4.5_{\pm 0.6}$	$5.6_{\pm 0.4}$	$5.7_{\pm 0.3}$	$5.4_{\pm 0.4}$	$2.3_{\pm 0.4}$	$2.2_{\pm 0.3}$	$2.4_{\pm 0.3}$
ER (Chaudhry et al. 2019)	$5.7_{\pm 0.6}$	$7.1_{\pm 0.7}$	$7.6_{\pm 0.7}$	$6.3_{\pm 0.4}$	$9.3_{\pm 0.6}$	$10.1_{\pm 1.1}$	$2.3_{\pm 0.3}$	$2.5_{\pm 0.2}$	$2.7_{\pm 0.2}$
MIR (Aljundi et al. 2019a)	$6.1_{\pm 0.5}$	$8.8_{\pm 0.6}$	$10.6_{\pm 1.2}$	$7.1_{\pm 0.3}$	$9.7_{\pm 0.4}$	$11.3_{\pm 0.5}$	$2.5_{\pm 0.5}$	$2.9_{\pm 0.3}$	$3.0_{\pm 0.3}$
GSS (Aljundi et al. 2019b)	$5.9_{\pm 0.6}$	$7.3_{\pm 0.8}$	$7.3_{\pm 0.7}$	$6.9_{\pm 0.5}$	$9.1_{\pm 0.5}$	$11.5_{\pm 0.4}$	$2.4_{\pm 0.4}$	$2.6_{\pm 0.3}$	$2.8_{\pm 0.4}$
ASER$_\mu$ (Shim et al. 2021)	$6.4_{\pm 0.4}$	$12.6_{\pm 0.6}$	$15.2_{\pm 0.6}$	$7.2_{\pm 0.3}$	$10.3_{\pm 0.6}$	$13.5_{\pm 0.7}$	$2.8_{\pm 0.4}$	$4.2_{\pm 0.4}$	$5.4_{\pm 0.5}$
DVC (Gu et al. 2022)	$8.3_{\pm 0.5}$	$12.6_{\pm 1.0}$	$15.2_{\pm 0.8}$	$10.4_{\pm 0.4}$	$16.6_{\pm 0.6}$	$20.2_{\pm 0.4}$	$3.4_{\pm 0.2}$	$5.1_{\pm 0.3}$	$6.8_{\pm 0.4}$
SCR (Mai et al. 2021)	$8.3_{\pm 0.4}$	$18.0_{\pm 0.6}$	$24.4_{\pm 0.3}$	$9.0_{\pm 0.5}$	$20.6_{\pm 0.4}$	$26.6_{\pm 0.6}$	$3.0_{\pm 0.2}$	$5.8_{\pm 0.2}$	$8.3_{\pm 0.3}$
OCM (Guo, Liu, and Zhao 2022)	-	-	-	$8.6_{\pm 0.5}$	$18.6_{\pm 0.4}$	$28.2_{\pm 0.6}$	$2.5_{\pm 0.3}$	$\mathbf{7.7}_{\pm 0.6}$	$\mathbf{9.9}_{\pm 0.5}$
RAR (Zhang et al. 2022)	$9.1_{\pm 0.6}$	$19.6_{\pm 0.7}$	$\mathbf{26.2}_{\pm 0.6}$	$9.7_{\pm 0.3}$	$21.2_{\pm 0.5}$	$28.5_{\pm 0.4}$	$3.5_{\pm 0.4}$	$6.7_{\pm 0.7}$	$9.2_{\pm 0.6}$
SSD (Ours)	$\mathbf{10.3}_{\pm 0.6}$	$\mathbf{19.8}_{\pm 0.4}$	$25.8_{\pm 0.3}$	$\mathbf{12.1}_{\pm 0.4}$	$\mathbf{23.0}_{\pm 0.6}$	$\mathbf{28.8}_{\pm 0.5}$	$\mathbf{3.6}_{\pm 0.2}$	$6.4_{\pm 0.5}$	$8.9_{\pm 0.3}$

Table 2: Average end accuracy on Mini-ImageNet (10 tasks), CIFAR-100 (10 tasks) and Tiny-ImageNet (20 tasks) with different memory sizes K. The averaged results of 10 runs are reported. The upper part shows results of memory-agnostic methods, and the lower part contains results of replay-based methods.

Modules			CIFAR-100		
D	S	P	K=100	K=500	K=1000
-	-	-	$9.0_{\pm 0.5}$	$20.6_{\pm 0.4}$	$26.6_{\pm 0.6}$
✓	-	-	$10.7_{\pm 0.5}$	$21.3_{\pm 0.4}$	$27.2_{\pm 0.7}$
✓	✓	-	$11.5_{\pm 0.7}$	$22.3_{\pm 0.6}$	$28.2_{\pm 0.6}$
✓	✓	✓	$\mathbf{12.1}_{\pm 0.4}$	$\mathbf{23.0}_{\pm 0.6}$	$\mathbf{28.8}_{\pm 0.5}$

Table 3: Ablation study on the effectiveness of each component in SSD. D: dynamic memory; S: summarizing data stream; P: past assistance. K refers to the memory size.

Experiments

Datasets and Evaluation Metric

We evaluate our methods on three popular continual learning benchmarks. **Sequential CIFAR-100** splits the CIFAR-100 dataset into 10 tasks, each with 10 non-overlapping classes (Krizhevsky, Hinton et al. 2009). **Sequential Mini-ImageNet** splits the Mini-ImageNet dataset into 10 tasks, and each task contains 10 classes (Vinyals et al. 2016). **Sequential Tiny-ImageNet** splits the Tiny-ImageNet dataset into 20 tasks, each of which consists of 10 independent classes (Deng et al. 2009). For performance comparison, we mainly report the average end accuracy with variance of all tasks when the whole training process is over and each experiment is conducted 10 times.

Implementation Details

For training the sequential tasks, following (Mai et al. 2021), we adopt a reduced ResNet-18 (He et al. 2016) architecture for all the experiments. In order to simulate the circumstances with restricted memory space, we set the memory size to contain 1, 5 and 10 images per class. We also extend the memory space to more ideal settings in the ablation study. An SGD optimizer is adopted for parameter updating, with the learning rate set as 0.1. By default, SCR (Mai et al. 2021) is adopted as the baseline, together with its training settings including objectives $\mathcal{L}_t$ and $\mathcal{L}_m$.

For the stream data summarizing, a 3-layer ConvNet is employed to extract the features and calculate the training gradients (Zhao, Mopuri, and Bilen 2021; Kim et al. 2022). An SGD optimizer with a learning rate η of 0.01 and a momentum of 0.9 is employed to update the summarizing model as in Eq. 5. The training objective $\mathcal{L}_t^{'}(\cdot;\cdot)$ is set as the standard cross entropy loss. The distance metric $\mathbf{D}$ is set as the euclidean distance. The summarizing interval τ is set as 6 and the similarity matching coefficient γ is set as 1.

Comparison with State-of-the-art Methods

We conduct the standard online class-incremental continual learning experiments on restricted memory sizes in Tab. 2. All the results are reproduced by us. For RAR (Zhang et al. 2022), we reproduce its results on SCR (Mai et al. 2021) for fair comparison. The result of OCM (Guo, Liu, and Zhao 2022) on Mini-ImageNet is not provided in the original paper, and constrained by computational overhead, we are not able to reproduce it.

We first present the results of some memory-agnostic methods in the upper part of the table. "fine-tune" stands for naive CL without addressing the catastrophic forgetting problem, and "iid offline" represents training all the data together for 50 epochs, acting as the upper bound. The results of replay-based methods are presented in the lower part. Under circumstances with extremely small memory size (K=100), the performance difference between methods is relatively small compared to larger sizes. It indicates that the limited information contained in the memory largely influences the replay effects. Among the replay-based methods, SSD is the only method achieving comparable performance with LwF (Li and Hoiem 2017) under a restricted memory size of 100 on CIFAR-100 and Mini-ImageNet benchmarks.

Method	CIFAR-100 K=100	K=500	K=1000
ER (Chaudhry et al. 2019)	6.3	9.3	10.1
+SSD	**7.5** (+1.2)	**10.8** (+1.5)	**12.9** (+2.8)
DVC (Gu et al. 2022)	10.4	16.6	20.2
+SSD	**12.0** (+1.6)	**18.5** (+1.9)	**21.8** (+1.6)
SCR (Mai et al. 2021)	9.0	20.6	26.6
+SSD	**12.1** (+3.1)	**23.0** (+2.4)	**28.8** (+2.2)

Table 4: The experiment results of applying SSD on multiple baseline methods. K refers to the memory size.

Method	CIFAR-100 K=100	K=500	K=1000
SCR (Mai et al. 2021)	9.0	20.6	26.6
SCR+GSS	9.9 (+0.9)	19.3 (-1.3)	24.4 (-2.2)
SCR+ASER	11.7 (+2.7)	21.8 (+1.2)	26.9 (+0.3)
Ours	**12.1** (+3.1)	**23.0** (+2.4)	**28.8** (+2.2)

Table 5: Comparisons between the proposed SSD method and other memory construction methods.

Besides, for all the memory sizes, SSD surpasses the baseline method (SCR), proving that SSD manages to improve the informativeness of the memory. Compared with state-of-the-art methods OCM and RAR, SSD performs better on small memory size and comparable on larger sizes with much less computational cost, as shown in Fig. 1.

Ablation Study and Analysis

Component analysis. We present the experiment results of adding each proposed component to the baseline method in Tab. 3. With SCR (Mai et al. 2021) set as the baseline, we first construct the dynamic memory. When the memory space is restricted, a class-balanced sampler becomes essential to provide necessary knowledge rehearsal for each class. With class-balanced knowledge, the dynamic memory achieves certain improvements on all memory sizes. But the margin is decayed as the memory size grows.

Then we introduce the proposed summarizing operation with training gradient matching to integrate information into memory samples. The summarized samples provide steady performance gains over the dynamic memory, which proves the effectiveness of the summarizing process. In addition to the training gradients, we also employ past summarized samples to provide better gradient supervision and global distribution guidance to the current summarizing process. The past-assisted summarizing further promotes the total accuracy margin to more than 2% over the SCR baseline with limited extra computational cost.

Appliance on other baselines. The proposed summarizing method is orthogonal to most of the previous continual learning methods. In order to validate the generalization capability of SSD, we apply it on multiple baselines in Tab. 4. On all the baselines SSD achieves stable performance gains by improving the informativeness of the memory. We also observe that with more appropriate memory replay training method, SSD generally obtains higher performance gain. With more rational training schemes, the more informative memory is even better utilized.

Method	CIFAR-100 K=100	K=500	K=1000
RAR (Zhang et al. 2022)	9.7	21.2	28.5
SSD	12.1 (+2.4)	23.0 (+1.8)	28.8 (+0.3)
SSD-5	13.5 (+3.8)	24.4 (+3.2)	30.4 (+1.9)
SSD-10	**13.7** (+4.0)	**24.7** (+3.5)	**31.1** (+2.6)

Table 6: The experiment results of extending training iterations. SSD-5 and SSD-10 indicate 5 and 10 replays inside each iteration, respectively.

Comparing memory-construction methods. There are also some previous works devoted to constructing more effective auxiliary memories. We compare the performance of applying them to the same SCR baseline in Tab. 5. GSS (Aljundi et al. 2019b) adopts a greedy strategy to maximize the gradient variance inside the memory. ASER (Shim et al. 2021) selects samples with high Shapley Values to form the memory. The results suggest that all the methods improve the memory informativeness under extremely small memory size. When the memory size is gradually enlarged, GSS fails to provide representative samples, and the performance margin of ASER also degrades fast. By contrast, SSD provides information gain across all memory sizes.

Scalability to larger memory. In addition to the circumstances with restricted memory space, CL can also be applied on more ideal environments with sufficient storage. We validate the scalability of the proposed SSD method in Fig. 4a. As the memory space increases, SSD stably provides performance improvements on both ER and SCR baselines. It validates from the performance aspect that SSD is able to be applied under larger memory space.

Computational cost analysis. The proposed SSD method significantly improves the informativeness of samples in the auxiliary memory. Compared to previous CL methods where original samples are selected from the data stream, SSD involves a distilling operation to integrate the information into memory samples. We analyze the required extra computational overhead in Fig. 4b. As the memory size increases, although there are more samples that require to be optimized, the extra computational cost still remains stable. Besides, SSD requires much less training time than RAR (Zhang et al. 2022) yet achieves better or comparable results. It validates from the computational overhead that the proposed SSD method is capable for real-world appliance.

Extending training iterations. RAR (Zhang et al. 2022) implements the CL training process with more memory replays inside each iteration, which is helpful for retaining the past knowledge. We also apply the repeated replay into our SSD method in Tab. 6. Through increasing the replay itera-

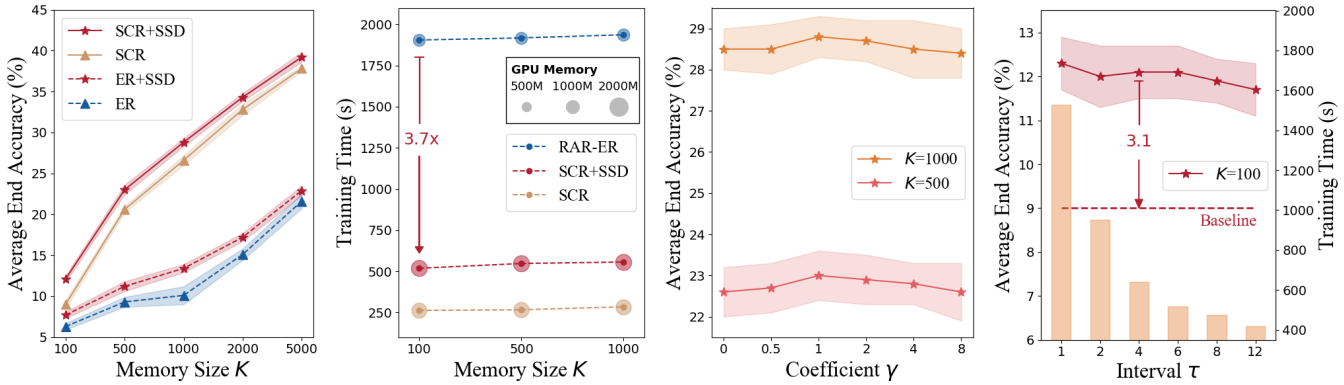

Figure 4: (a) The experiment results of increasing the memory size. (b) The computational cost comparison. (c) The parameter analysis on the relationship matching coefficient γ. (d) The parameter analysis on the summarizing interval τ.

Figure 5: Visualizations of original images (left), data stream (middle) and summarized images (right) on CIFAR-100. The information of color, structure and texture are integrated into the summarized images, helping improve the replay effects. Best viewed in color.

tions, the rich information carried by summarized samples is better utilized, which further enlarges the performance gain of SSD. When the memory has larger space and contains more diverse information, the repeated play provides more obvious improvements.

Visualization

We provide the comparison between the original images and the summarized ones in Fig. 5 to explicitly explain the effectiveness of the proposed summarizing method. Generally, more diverse colors that appear in the real distribution are introduced into the summarized images. For orchid (the first row) specifically, the initialized flowers are of yellow color, while most of real samples are pink. By modifying part of the yellow flowers into pink, the bias is corrected, and the original yellow features are not completely eliminated. For bicycles and cans, the structure and texture information are largely enhanced during the summarizing process, respectively. The summarized images contain much richer information compared with the original ones, which helps improve the replay effects. Note that SSD is different from the color jittering augmentation, where the color is modified in a random manner. More visualizations are presented in the supplementary material.

Parameter Analysis

Relationship Matching Coefficient. Training gradients and relationship to the past knowledge simultaneously serve as the matching metrics for distilling the information from the data stream. From the results in Fig. 4c, we can observe that adding relationship matching generally improves the average accuracy. And the best results are achieved when setting the coefficient $\gamma = 1$.

Summarizing Interval. The interval τ influences the frequency of distilling information, and the calculation time of CL. As shown in Fig. 4d, SSD achieves the best performance when conducting summarizing at each iteration. However, it requires a large amount of calculation time. As the interval increases from 2 to 6, there is no obvious performance drop. Based on the comprehensive consideration of performance and computational overhead, we set the interval $\tau = 6$. Under even larger summarizing intervals, SSD still achieves higher accuracy compared to the baseline.

Conclusion

In this paper, we propose a Summarizing Stream Data (SSD) method to improve the informativeness of the auxiliary memory for replay-based online continual learning methods. Through sequentially integrating the information from the stream data, the summarized samples are more representative for the whole data distribution compared with original images. With limited extra computational overhead, SSD helps significantly improve the replay effects, especially under circumstances with restricted memory space. The efficient and effective summarizing method inspires future works to design better means to constructing memories.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 62173302 and in part by the Zhejiang Provincial Natural Science Foundation of China under Grant Z24F030007. Yang You's research group is being sponsored by NUS startup grant (Presidential Young Professorship), Singapore MOE Tier-1 grant, ByteDance grant, ARCTIC grant, SMI grant, Alibaba grant, and Google grant for TPU usage.

References

Aljundi, R.; Belilovsky, E.; Tuytelaars, T.; Charlin, L.; Caccia, M.; Lin, M.; and Page-Caccia, L. 2019a. Online Continual Learning with Maximal Interfered Retrieval. In *NeurIPS*, 11849–11860.

Aljundi, R.; Chakravarty, P.; and Tuytelaars, T. 2017. Expert gate: Lifelong learning with a network of experts. In *CVPR*, 3366–3375.

Aljundi, R.; Lin, M.; Goujaud, B.; and Bengio, Y. 2019b. Gradient based sample selection for online continual learning. *NeurIPS*, 32.

Buzzega, P.; Boschini, M.; Porrello, A.; Abati, D.; and Calderara, S. 2020. Dark experience for general continual learning: a strong, simple baseline. *NeurIPS*, 33: 15920–15930.

Cazenavette, G.; Wang, T.; Torralba, A.; Efros, A. A.; and Zhu, J.-Y. 2022. Dataset distillation by matching training trajectories. In *CVPR*, 4750–4759.

Chaudhry, A.; Dokania, P. K.; Ajanthan, T.; and Torr, P. H. 2018a. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *ECCV*, 532–547.

Chaudhry, A.; Ranzato, M.; Rohrbach, M.; and Elhoseiny, M. 2018b. Efficient Lifelong Learning with A-GEM. In *ICLR*.

Chaudhry, A.; Rohrbach, M.; Elhoseiny, M.; Ajanthan, T.; Dokania, P. K.; Torr, P. H.; and Ranzato, M. 2019. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Du, J.; Jiang, Y.; Tan, V. Y.; Zhou, J. T.; and Li, H. 2023. Minimizing the accumulated trajectory error to improve dataset distillation. In *CVPR*, 3749–3758.

Fernando, C.; Banarse, D.; Blundell, C.; Zwols, Y.; Ha, D.; Rusu, A. A.; Pritzel, A.; and Wierstra, D. 2017. Pathnet: Evolution channels gradient descent in super neural networks. *arXiv preprint arXiv:1701.08734*.

Ghunaim, Y.; Bibi, A.; Alhamoud, K.; Alfarra, M.; Al Kader Hammoud, H. A.; Prabhu, A.; Torr, P. H.; and Ghanem, B. 2023. Real-time evaluation in online continual learning: A new hope. In *CVPR*, 11888–11897.

Gu, J.; Vahidian, S.; Kungurtsev, V.; Wang, H.; Jiang, W.; You, Y.; and Chen, Y. 2023. Efficient Dataset Distillation via Minimax Diffusion. *arXiv preprint arXiv:2311.15529*.

Gu, Y.; Yang, X.; Wei, K.; and Deng, C. 2022. Not Just Selection, but Exploration: Online Class-Incremental Continual Learning via Dual View Consistency. In *CVPR*, 7442–7451.

Guo, Y.; Liu, B.; and Zhao, D. 2022. Online continual learning through mutual information maximization. In *ICML*, 8109–8126. PMLR.

Harun, M. Y.; Gallardo, J.; Hayes, T. L.; Kemker, R.; and Kanan, C. 2023. SIESTA: Efficient Online Continual Learning with Sleep. *arXiv preprint arXiv:2303.10725*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.

Kang, H.; Mina, R. J. L.; Madjid, S. R. H.; Yoon, J.; Hasegawa-Johnson, M.; Hwang, S. J.; and Yoo, C. D. 2022. Forget-free continual learning with winning subnetworks. In *ICLR*, 10734–10750. PMLR.

Kim, J.-H.; Kim, J.; Oh, S. J.; Yun, S.; Song, H.; Jeong, J.; Ha, J.-W.; and Song, H. O. 2022. Dataset Condensation via Efficient Synthetic-Data Parameterization. *arXiv preprint arXiv:2205.14959*.

Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A. A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13): 3521–3526.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

Lee, S.-W.; Kim, J.-H.; Jun, J.; Ha, J.-W.; and Zhang, B.-T. 2017. Overcoming catastrophic forgetting by incremental moment matching. *NeurIPS*, 30.

Li, Z.; and Hoiem, D. 2017. Learning without forgetting. *T-PAMI*, 40(12): 2935–2947.

Liu, S.; Ye, J.; Yu, R.; and Wang, X. 2023a. Slimmable dataset condensation. In *CVPR*, 3759–3768.

Liu, Y.; Gu, J.; Wang, K.; Zhu, Z.; Jiang, W.; and You, Y. 2023b. DREAM: Efficient Dataset Distillation by Representative Matching. In *ICCV*.

Loo, N.; Hasani, R.; Amini, A.; and Rus, D. 2022. Efficient Dataset Distillation using Random Feature Approximation. In *NeurIPS*.

Lopez-Paz, D.; and Ranzato, M. 2017. Gradient episodic memory for continual learning. *NeurIPS*, 30.

Lu, Y.; Chen, X.; Zhang, Y.; Gu, J.; Zhang, T.; Zhang, Y.; Yang, X.; Xuan, Q.; Wang, K.; and You, Y. 2023. Can pre-trained models assist in dataset distillation? *arXiv preprint arXiv:2310.03295*.

Mai, Z.; Li, R.; Kim, H.; and Sanner, S. 2021. Supervised contrastive replay: Revisiting the nearest class mean classifier in online class-incremental continual learning. In *CVPR*, 3589–3599.

McCloskey, M.; and Cohen, N. J. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, 109–165. Elsevier.

Nguyen, T.; Chen, Z.; and Lee, J. 2020. Dataset Meta-Learning from Kernel Ridge-Regression. In *ICLR*.

Nguyen, T.; Novak, R.; Xiao, L.; and Lee, J. 2021. Dataset distillation with infinitely wide convolutional networks. *NeurIPS*, 34: 5186–5198.

Pham, Q.; Liu, C.; and Steven, H. 2022. Continual Normalization: Rethinking Batch Normalization for Online Continual Learning. In *ICLR*.

Prabhu, A.; Cai, Z.; Dokania, P.; Torr, P.; Koltun, V.; and Sener, O. 2023. Online continual learning without the storage constraint. *arXiv preprint arXiv:2305.09253*.

Rebuffi, S.-A.; Kolesnikov, A.; Sperl, G.; and Lampert, C. H. 2017. icarl: Incremental classifier and representation learning. In *CVPR*, 2001–2010.

Ring, M. B. 1998. CHILD: A first step towards continual learning. In *Learning to learn*, 261–292. Springer.

Sangermano, M.; Carta, A.; Cossu, A.; and Bacciu, D. 2022. Sample condensation in online continual learning. In *IJCNN*, 01–08. IEEE.

Shim, D.; Mai, Z.; Jeong, J.; Sanner, S.; Kim, H.; and Jang, J. 2021. Online class-incremental continual learning with adversarial shapley value. In *AAAI*, 9630–9638.

Shin, H.; Lee, J. K.; Kim, J.; and Kim, J. 2017. Continual learning with deep generative replay. *NeurIPS*, 30.

Smith, J.; Hsu, Y.-C.; Balloch, J.; Shen, Y.; Jin, H.; and Kira, Z. 2021. Always be dreaming: A new approach for data-free class-incremental learning. In *ICCV*, 9374–9384.

Sun, S.; Calandriello, D.; Hu, H.; Li, A.; and Titsias, M. 2022. Information-theoretic Online Memory Selection for Continual Learning. In *ICLR*.

Thrun, S. 1998. Lifelong learning algorithms. In *Learning to learn*, 181–209. Springer.

Tiwari, R.; Killamsetty, K.; Iyer, R.; and Shenoy, P. 2022. Gcr: Gradient coreset based replay buffer selection for continual learning. In *CVPR*, 99–108.

Vinyals, O.; Blundell, C.; Lillicrap, T.; Wierstra, D.; et al. 2016. Matching networks for one shot learning. *Advances in neural information processing systems*, 29.

Wang, K.; Gu, J.; Zhou, D.; Zhu, Z.; Jiang, W.; and You, Y. 2023. DiM: Distilling Dataset into Generative Model. *arXiv preprint arXiv:2303.04707*.

Wang, K.; Zhao, B.; Peng, X.; Zhu, Z.; Yang, S.; Wang, S.; Huang, G.; Bilen, H.; Wang, X.; and You, Y. 2022. Cafe: Learning to condense dataset by aligning features. In *CVPR*, 12196–12205.

Wang, T.; Zhu, J.-Y.; Torralba, A.; and Efros, A. A. 2018. Dataset distillation. *arXiv preprint arXiv:1811.10959*.

Xiang, Y.; Fu, Y.; Ji, P.; and Huang, H. 2019. Incremental learning using conditional adversarial networks. In *ICCV*, 6619–6628.

Yoon, J.; Madaan, D.; Yang, E.; and Hwang, S. J. 2022. Online Coreset Selection for Rehearsal-based Continual Learning. In *ICLR*.

Zhang, Y.; Pfahringer, B.; Frank, E.; Bifet, A.; Lim, N. J. S.; and Jia, A. 2022. A simple but strong baseline for online continual learning: Repeated Augmented Rehearsal. In *NeurIPS*.

Zhao, B.; and Bilen, H. 2023. Dataset condensation with distribution matching. In *WACV*, 6514–6523.

Zhao, B.; Mopuri, K. R.; and Bilen, H. 2021. Dataset Condensation with Gradient Matching. *ICLR*, 1(2): 3.