Anti-Retroactive Interference for Lifelong Learning

Runqi Wang[1], Yuxiang Bao[1], Baochang Zhang[1⋆], Jianzhuang Liu[2], Wentao Zhu[3], and Guodong Guo[4]

[1] Beihang University
[2] Huawei Noah's Ark Lab
[3] Kuaishou Technology
[4] Institute of Deep Learning, Baidu Research
{runqiwang,bczhang}@buaa.edu.cn

Abstract. Humans can continuously learn new knowledge. However, machine learning models suffer from drastic dropping in performance on previous tasks after learning new tasks. Cognitive science points out that the competition of similar knowledge is an important cause of forgetting. In this paper, we design a paradigm for lifelong learning based on meta-learning and associative mechanism of the brain. It tackles the problem from two aspects: extracting knowledge and memorizing knowledge. First, we disrupt the sample's background distribution through a background attack, which strengthens the model to extract the key features of each task. Second, according to the similarity between incremental knowledge and base knowledge, we design an adaptive fusion of incremental knowledge, which helps the model allocate capacity to the knowledge of different difficulties. It is theoretically analyzed that the proposed learning paradigm can make the models of different tasks converge to the same optimum. The proposed method is validated on the MNIST, CIFAR100, CUB200 and ImageNet100 datasets. The code is available at *https://github.com/bhrqw/ARI*.

Keywords: Lifelong Learning, Meta Learning, Background Attack, Associative Learning

1 Introduction

A standard benchmark for success in artificial intelligence is the ability to emulate human learning. However, at the current stage, the machine does not really understand what it has learned. It may just do rote memorization, which overlooks a critical characteristic of human learning: being robust to changing tasks and sequential experience. Future learning machines should be able to adapt to the ever-changing world. They should continuously learn new tasks without forgetting previously learned ones. Although many learning paradigms have been

⋆ Corresponding author.

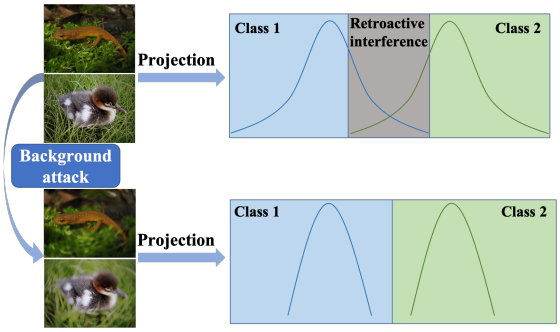

Fig. 1: In lifelong learning, the knowledge in the present stage competes with the previous memory and interferes with the previous learning, especially when the knowledge is similar. Therefore, we attack similar contents in both tasks to change the data distribution and avoid retroactive interference.

proposed, such as lifelong learning (LLL) [2,21], these problems have not been addressed well. Many researchers are brute-force and idealized in the construction of model training. In pedagogy and psychology, human learning and the cognitive process have been widely discussed, among which there are many theories worthy of reference. The learning process of new tasks results in catastrophic forgetting [10] of previous knowledge due to **retroactive interference** [25], which means that the content of later learning competes with the previous memory and interferes with the previous learning. This kind of competition causes confusion and forgetting of knowledge. This problem can be solved by capturing critical points of knowledge and removing redundant content to avoid the competition of knowledge, which is termed filter efficiency [27] in pedagogy. In the computer vision task of classification, different categories of images might have the same or similar backgrounds, such as a bicycle and a dog on a lawn. Machine learning models may mistake lawn features for bicycle features or dog features, which creates unnecessary memory competition in learning new knowledge.

In order to solve the problem of forgetting in the process of learning new tasks in a deep learning model, we propose a lifelong learning paradigm based on meta-learning and associative learning. We divide the model into two stages: extracting intra-class features and fusing inter-class features. In the first stage, we hope to avoid retroactive interference by reducing the competition between old and new knowledge. We need to accurately capture the critical knowledge of new tasks and focus on learning it, which can effectively avoid confusion of knowledge. In this way, incremental knowledge can complement rather than compete with existing knowledge. It is an anthropomorphic process that associates the original images with the foreground of the images, i.e., only learning the critical knowledge of the new task as a complement to the knowledge base. In this way, model information redundancy can be avoided, which is consistent with the machine learning theory in [19]. In order to realize this idea, we present a background attack method to attack the samples adversarially. Through the

spatial attention mechanism, the importance map of the image can be obtained. We believe that areas of low importance level in an image do not belong to the necessary information of its class, which may cause information redundancy and competition between classes as shown in Fig. 1. Therefore, we carry out an adversarial attack on non-critical areas, (*i.e.*, the background) and blur the data distribution in these areas, thus weakening the model's learning of unimportant information.

In the second stage, we combine the existing model with the model just learned. It is different from conventional incremental learning that updates the pre-trained model directly, which is easy to cause catastrophic damage to the model's weight distribution. We organize the knowledge to learn into different tasks, just like the chapters of a textbook. Each task is learned separately, and an independent model is outputted. Specifically, when learning a new task, a small number of samples are extracted from previous tasks for review, and then the models corresponding to these tasks are fused, which is consistent with Ausubel's theory [3] that points out that the most important thing in learning is whether the knowledge learned can form a system, *i.e.*, to complete the deduction of knowledge from the individual to the whole. Following this process, we chose a meta-training based method to generate models, which will be described in Sec. 3. To this end, we propose a novel task-specific fusion method, and show that our training process can ensure that these different models are converged to a common optimal one to reduce the information loss. Our contributions are summarized as follows:

- We combine the adversarial attack with meta-learning to extract features. The adversarial attack is performed on the image background to reinforce the model's attention to critical features.
- Based on human cognition, a new lifelong learning paradigm, *Anti-Retroactive Interference for lifelong learning* (ARI), is established to ensure that the machine learning model can integrate incremental knowledge more effectively. It is analyzed that the fusion method in ARI can ensure that the task-specific models are converged to the same optimal model to reduce the information loss caused by fusion.
- The proposed method is validated on the MNIST, CIFAR100, CUB200 and ImageNet100 datasets, and state-of-the-art results are obtained on all the benchmarks.

2 Related Work

2.1 Lifelong Learning

So far, lifelong learning methods can be divided into three groups. The first one is based on regularization. LwF [16] preserves the ancient knowledge by adding a distillation loss. In addition, the distillation loss is implemented by [22,4,36] to reduce forgetting. [30,36] propose bias correction strategies whereby the model can perform equally well on current and older classes by re-balancing the final

fully-connected layer. EWC [13] computes synaptic importance offline by calculating a Fisher information matrix. E-MAS-SDC proposed by [32] estimates the drift of previous tasks during the training of new tasks to make semantic drift compensation. RRR in [7] tries to save the correct attentions of previous images to avoid the attentions being affected by other tasks. The second group is about expanding the model with progressive learning and designing binary masks that directly map each task to the corresponding model architecture. MARK [12] keeps a set of shared weights among tasks. These shared weights are envisioned as a common knowledge base used to learn new tasks and enriched with new knowledge as the model learns new tasks. In [1], each convolutional layer is equipped with task-specific gating modules, selecting specific filters for a given task. The shortcomings of these methods are the extra model complexity and the need for a practical scheme to calculate the mask precisely. The third group is replay based and it gets popular recently. Replay based approaches are ideally suitable for lifelong learning in which tasks are added in turn. iTAML [21] introduces a meta-learning approach that seeks to maintain an equilibrium between all the encountered tasks, in the sense that it is unbiased towards class samples of majority and simultaneously minimizes forgetting.

2.2 Adversarial Training

Though the success of deep learning models has been demonstrated on various computer vision tasks, they are sensitive to adversarial attacks [9]. An imperceptible perturbation added to inputs may cause undesirable outputs. The Fast Gradient Sign Method (FGSM) is proposed in [8] to generate adversarial examples with a single gradient step. To defend the attacks, many methods have been proposed to defend against them. The most common method is adversarial training [18,15,26] with adversarial examples added to the training data. In this paper, we introduce adversarial training to the meta-learning process to obtain a robust model that can extract good features from very few available samples.

3 Proposed Method

We adopt a task-incremental learning setup where the model continuously learns new tasks, each containing a fixded number of novel classes. During the training process of task n, we have access to $\mathbb{M}_{n-1}$ and $\mathbb{D}_n$ where $\mathbb{M}_{n-1}$ is an exampler memory containing a small number of samples for old tasks, and $\mathbb{D}_n$ is the training data for task n, which contains pairs $(\mathbf{x}_i, y_i)$, with $\mathbf{x}_i$ being an image of class $y_i \in R_n$. Using $\mathbb{M}_{n-1}$ to train task n is a form of meta-learning. We define the set of classes on task n as $R_n = \{r_{n,1}, r_{n,2}, ..., r_{n,m}\}$, where $r_{n,1}$ is the first class in task n, and m is the number of classes in task n. Different tasks do not contain the same class: $R_t \cap R_s = \varnothing, t \neq s$. After learning all the tasks, we evaluate the learned model on all tasks $R = \cup_i R_i$.

Algorithm 1: Associative learning with background attack

Input: Training data $\mathbf{x}$;
Hyper-parameters: Epoch number S, $\varepsilon = \frac{8}{255}$;
Initialize model parameters $\boldsymbol{\theta}$;
Output: The network model;
Train an architecture for S epochs:
$t = 0$;
while $(t \leq S)$ **do**
  # First inference:
  Input $\mathbf{x}$;
  According to [29], calculate the spatial attention $\mathbf{A}$;
  Return $\mathbf{A}$;
  #Back propagation:
  According to Eq. 2, calculate $\mathbf{x}'$;
  # Second inference:
  Input $\mathbf{x}'$;
  #Back propagation:
  Update parameters $\boldsymbol{\theta}$;
  $t \leftarrow t + 1$.
end

3.1 Extracting Intra-Class Features

Lifelong learning requires the model to retain previous knowledge and learn new knowledge. However, if the previous and new knowledge have similar characteristics, it is easy to cause forgetting. Data are labeled for different classes according to different object features, but the background information is ignored, which may mislead the model's incremental learning. In order to eliminate similar characteristics between different classes and prevent retroactive interference, we design associative learning with background attack. This approach involves two processes. In the first process, the model learns from the original image to obtain the background region and conduct adversarial attack on it. This attack can disturb the distribution of the background and strengthen the feature extraction on the critical region of the image. In the second process, the model is trained with the attacked images. This approach associates the objects with different backgrounds, which avoids the negative effect of background on few-shot learning. Therefore, the model can effectively avoid over-fitting by associative learning.

In adversarial training, we need to add perturbation to the images, which can increase the robustness of the model. However, now we use a background mask $\mathbf{B}$ to guide the model to attack the background regions of the images. The mask $\mathbf{B}$ has three forms:

$$\mathbf{B} = 1 - \mathbf{A}, \quad \mathbf{B} = 1 - \mathbf{A} \circ \mathbf{A}, \quad \mathbf{B} = \frac{1}{\mathbf{A}}, \tag{1}$$

where $\mathbf{A} \in \mathbb{R}^{s \times s}$ denotes the spatial attention obtained by [29], $\circ$ denotes the Hadamard product, $\mathbf{B} \in \mathbb{R}^{s \times s}$ is the mask for focusing on the background, and

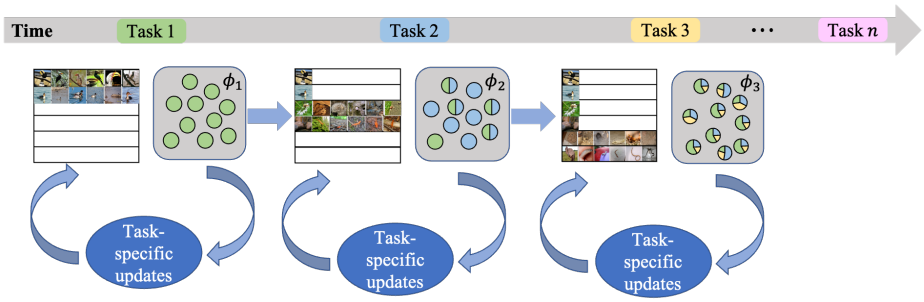

Fig. 2: We design a serial learning structure for lifelong learning. A small-scale rehearsal memory of the previous tasks is also used to fine-tune the new model to adapt to the new task.

$s \times s$ is the size of the image. In order to widen the distance between important and unimportant information in the attention and guide the background attack, we use the three forms of $\mathbf{B}$ in Eq. 1, making the unimportant regions (corresponding more to the background) prominent. Therefore, the attack guided by $\mathbf{B}$ tends to be more selective on the background.

We formulate the background attack model as:

$$\mathbf{x}' = \mathbf{x} + \mathbf{B} \circ \boldsymbol{\zeta} = \mathbf{x} + \mathbf{B} \circ (\varepsilon \mathbf{sgn}(\nabla_{\mathbf{x}} G(\boldsymbol{\theta}, \mathbf{x}, y))), \tag{2}$$

where $\mathbf{x} \in \mathbb{R}^{s \times s}$ is the clean input and $\mathbf{x}'$ is the adversarial counterpart. $\boldsymbol{\zeta}$ denotes the global perturbation of the clean input $\mathbf{x}$ which is designed based on [8]. y denotes the label of the input $\mathbf{x}$. ε is the perturbation bound, $\boldsymbol{\theta}$ denotes the parameters of the deep model, and G is the cross-entropy function.

The algorithm of the associative learning with adversarial background attack is listed in Algorithm 1, which associates clean input $\mathbf{x}$ with various adversarial inputs $\mathbf{x}'$. After the adversarial training with $\mathbf{x}'$, the model learns to be robust to the distribution shift [34] of background and thus can focus more on the foreground (object) features, reducing forgetting as shown in Fig. 1. Experimental verification is shown in Sec. 4.4.

3.2 Generating and Fusing Task-Specific Models

The adversarial images after the background attack are used as input to participate in training. Lifelong learning is a scenario in which tasks are entered serially. The base model should contain information about all learned tasks after learning a new task, as shown in Fig. 2, in which $\boldsymbol{\phi}_n$ denotes the base model after learning task n. The process of learning a new task is embedded in the task-specific updating. When updating in a new task, our meta-learning approach involves three phases: (1) generating task-specific models for all the seen tasks, (2) fusing the task-specific models into the base model, and (3) meta-training the base model, as shown in Fig. 3.

Algorithm 2: Training in task n

Input: $\mathbb{D}_n$, $\mathbb{M}_{n-1}$, ϕ_{n-1};
Hyper-parameters: task number n, epoch number S, image number J_i of task i, $i \in [1, n]$;
Output: The base model ϕ_n;
Train an architecture for S epochs;
$\phi_b^0 = \phi_{n-1}$, $t = 1$;
while $(t \leq S)$ **do**
  for $i = 1$ **to** n **do**
    $\{\hat{y}_j^i\}_{j=1}^{J_i} \leftarrow \phi_b^{t-1}\left(\{\mathbf{x}_j^i\}_{j=1}^{J_i}\right)$;
    $loss \leftarrow$ **Eq**. 7;
  end
  $\phi_i^t \leftarrow$ **Optimizer**$(\phi_b^{t-1}, loss)$;
  $\phi_f^t \leftarrow$ **Fusion**$[\phi_1^t, ..., \phi_n^t, \phi_b^{t-1}]$;
  $\phi_b^t \leftarrow \gamma\phi_f^t + (1-\gamma)\phi_b^{t-1}$;
  $t \leftarrow t + 1$;
end
$\phi_n \leftarrow$ **Meta train** (ϕ_b^S).

Generating task-specific models. We randomly sample a mini-batch $\mathbb{B}_n = \{(\mathbf{x}_k, y_k)\}_{k=1}^{K_n}$ from the current task n training data $\mathbb{D}_n$ and the memory bank $\mathbb{M}_{n-1}$, which contains a few samples for old tasks. $\mathbf{x}_k$ and y_k are the training images and their labels, respectively, and K_n is the image number of the batch. Therefore, the mini-batch of data for task-specific updates, as shown in Fig. 3, is represented as:

$$\mathbb{B}_n \sim \mathbb{D}_n \cup \mathbb{M}_{n-1}. \tag{3}$$

We sample the training data according to the tasks to construct $\mathbb{B}_\mu^i = \{(\mathbf{x}_j^i, y_j^i)\}_{j=1}^{J_i}$ for training the task-specific models ϕ_i, $i \in [1, n]$, where J_i is the image number of task i. The loss function in the task-specific updating is the binary cross-entropy loss with a regularizer from **dif**, which is defined next in Eq. 6. The binary cross-entropy is:

$$L(\phi_i(\{\mathbf{x}_j^i\}), \{y_j^i\}) = -\frac{1}{J_i}\sum_{j=1}^{J_i}(y_j^i \cdot \log\left(\phi_i\left(\mathbf{x}_j^i\right)\right) + \left(1 - y_j^i\right) \cdot \log(1 - \log(\phi_i(\mathbf{x}_j^i)))). \tag{4}$$

This helps to obtain task-specific models ϕ_i , thus providing a better estimate for gradient updates in the current task-specific training (described next) to obtain a base model. The training process of the specific tasks, *i.e.*, phase 1 in Fig. 3, is shown in the for-loop of Algorithm 2, which generates n independent models. In Algorithm 2, the Optimizer denotes some optimizer such as SGD. The function Fusion is described next (Eq. 9). ϕ_i^t is the task-specific model i at epoch t, and ϕ_b^t is the base model at epoch t. All these models $\phi_1, ..., \phi_n, \phi_b$ have the same structure.

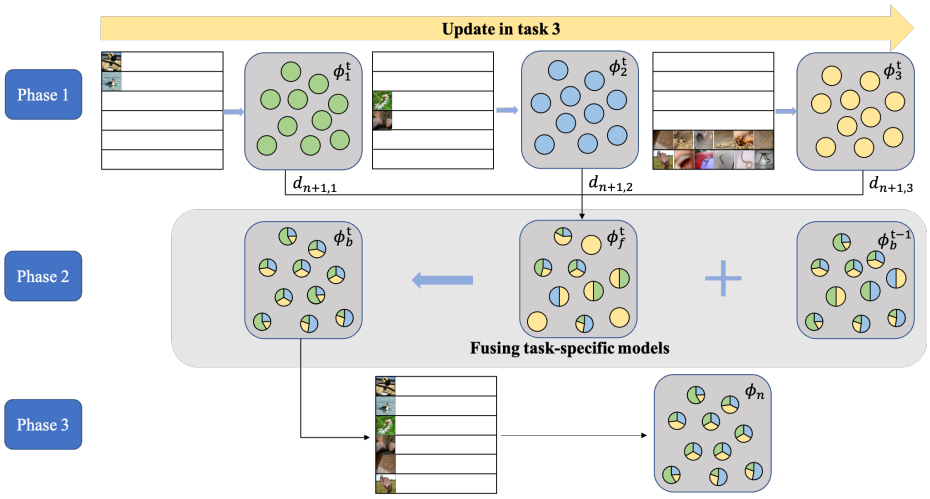

Fig. 3: Taking task 3 for example, the task-specific updating is divided into three phases. In phase 1, task-specific model training is carried out. It is noteworthy that only a small amount of previous task samples are used in the current training. In phase 2, task-specific models are fused. In phase 3, meta-training is performed on the fused model to obtain the incremental base model of task 3.

Fusing task-specific models. We combine the task-specific models ϕ_i^t generated during phase 1 to the base model ϕ_n in phase 2 of Fig. 3. We denote the set of the models at epoch t as:

$$\Phi^t = \{\phi_1^t, ..., \phi_n^t, \phi_b^{t-1}\}. \tag{5}$$

Due to the task-specific models being generated by different tasks, there may be large differences between their parameter values, which causes information loss in model fusion. We adopt a new strategy to use the Manhattan distance between a task-specific model and the base model as the fusion weight. When the gap between the two models is larger, the fusion weight is larger. The weight coefficients are calculated as follows. First, we define

$$\mathbf{dif} = \begin{bmatrix} 0 & d_{1,2} & \cdots & d_{1,n+1} \\ d_{2,1} & 0 & & d_{2,n+1} \\ \vdots & & \ddots & \vdots \\ d_{n+1,1} & d_{n+1,2} & \cdots & 0 \end{bmatrix}, \tag{6}$$

where $d_{1,2}$ denotes the Manhattan distance between ϕ_1^t and ϕ_2^t, and $d_{1,n+1}$ denotes the Manhattan distance between ϕ_1^t and ϕ_b^{t-1}. Considering the goal of model fusion is to minimize the differences among task-specific models and produce a fused model that performs well across tasks, we formulate the loss as two parts, the regularizer based on **dif** and the binary cross-entropy as shown

in Eq. 4.

$$loss = L\left(\hat{y}_j^i, y_j^i\right) + \sum_{a=1}^{i+1}\sum_{b=1}^{i+1}|d_{a,b}|^2. \tag{7}$$

To ensure that the sum of weights equals 1, each row of **dif** is transformed by the softmax function as:

$$\mathbf{dif}^* = \begin{bmatrix} d_{1,1}^* & d_{1,2}^* & \cdots & d_{1,n+1}^* \\ d_{2,1}^* & d_{2,2}^* & & d_{2,n+1}^* \\ \vdots & & \ddots & \vdots \\ d_{n+1,1}^* & d_{n+1,2}^* & \cdots & d_{n+1,n+1}^* \end{bmatrix}. \tag{8}$$

Finally, the fused model is formulated as:

$$\phi_f^t = \sum_{i=1}^{n+1}(d_{n+1,i}^* \cdot \phi_i^t), \tag{9}$$

where $\phi_{n+1}^t = \phi_b^{t-1}$. The reason we take the elements of the last row of $\mathbf{dif}^*$ as the weights is that the base model could adopt the knowledge from the task-specific models as much as possible. Thus more weights should be given to the task-specific model with a larger difference from the base model.

The fusion model is combined with ϕ_b^{t-1} to form a new base model ϕ_b^t:

$$\phi_b^t = \gamma\phi_f^t + (1-\gamma)\phi_b^{t-1}, \tag{10}$$

where γ is a hyper-parameter that controls the speed of learning new information, *i.e.*, for higher γ the model prefers to learn new information and forget the old, and with smaller γ it learns little new knowledge.

Due to the regularization from **dif**, after a sufficient number of iterations, in the sense that t is large enough, the differences among the task-specific and base models $\{\phi_1^t, ..., \phi_n^t, \phi_b^{t-1}\}$ is decreasing gradually and all the models tend to have the same weights. In the supplementary material, we provide evidence to analyze that all the models converge to the same optimal weights. Moreover, an experiment is conducted in the ablation study to verify the convergence. When all the models, $\phi_1, ..., \phi_n, \phi_b$ are ideally optimized to the same model, they share the same knowledge, thus eliminating information loss in the task-specific model fusion.

Meta-training the base model. In phase 3 of Fig. 3, take a small number of samples from all learned tasks to form $\mathbb{M}_n$. $\mathbb{M}_n$ is used for meta-training of ϕ_b^S to further optimize the distribution of model parameters. After meta-training, ϕ_b^S is the model ϕ_n that learned task n.

4 Experiments and Results

We conduct experiments on several common datasets, including MNIST [33], CIFAR100 [14], CUB200 [28] and ImageNet100 which is a subset of ISLVRC 2012 [23]. We also perform ablation study to analyze different components of our approach.

4.1 Datasets

MNIST. MNIST contains 60k images of handwritten numbers in the training set and 10k samples in the test set. All the images are 28×28 pixels. In our experiment, MNIST is divided into 5 tasks with 2 classes per task.

CIFAR100. CIFAR100 consists of 60k pictures of 32×32 color images from 100 classes. Each class has 500 training and 100 testing samples. 100 classes are split into 10 tasks with 10 classes in each task.

CUB200. CUB200 contains 200 classes of birds with 11,788 images in total. The training set and the test set consist of 5994 and 5794 images, respectively. The 200 bird classes are split into 6 tasks in our experiment.

ImageNet100. ImageNet100, as a subset of ILSVRC2012, contains 100 classes and 130 thousand samples of 224×224 color images. Each class has about 1,300 training and 50 test samples. We split ImageNet100 into 10 tasks.

4.2 Implementation Details

Network architecture. For MNIST, a two-layer MLP is selected as the model. For CIFAR100 and CUB200, the network is $(ResNet-18(1/3))$ which is a reduced version of ResNet-18. For ImageNet100, the original ResNet-18 is used in the experiment. All the architectures used are added the spatial attention mechanism after the first layer.

Training details. For MNIST, each incremental training has 20 epochs. The initial learning rate is set to 0.1 and reduced to 1/2 of the previous learning rate after 5, 10, and 15 epochs. The weight decay is set to 0, the batch size is 256, and $\gamma = 0.1$. The optimizer is SGD.

For CIFAR100, each incremental training has 70 epochs. The initial learning rate starts from 0.01 and is reduced to 1/5 of the previous learning rate after 30 and 60 epochs. The weight decay is set to 0, the batch size is 512, and $\gamma = 0.1$. The optimizer is set to RAdam[21].

For CUB200 and ImageNet100, each incremental task is trained for 100 epochs. The learning rate starts from 0.1 initially and is reduced to 1/10 of the previous learning rate after 40, 70, and 90 epochs. The weight decay is set to 0, the batch size is 512, and $\gamma = 0.1$. The optimizer is RAdam[21].

For a fair comparison, we set the rehearsal memory size as 2,000 for MNIST and CIFAR100. For CUB200 and Imagenet100, the memory size is set as 3000. The perturbation bound $\epsilon = \frac{8}{255}$ and step size of $\frac{2}{255}$ is set for all the benchmarks.

4.3 Results and Comparison

In this section, we report the results on MNIST, CIFAR100, CUB200 and ImageNet100, and compare our ARI method with the state-of-the-art methods.

Small Scale. The compared typical lifelong learning approaches include Memory Aware Synapses (MAS) [2], LwF [16], Synaptic Intelligence (SI) [33], Elastic Weight Consolidation (EWC) [13], Gradient Episodic Memory (GEM) [17],

Deep Generative Replay (DGR) [24] and Incremental Task-Agnostic Meta learning (iTAML) [21]. As shown in Fig. 4, ARI outperforms all the others. Its average classification accuracy of 5 tasks is around 98.91%.

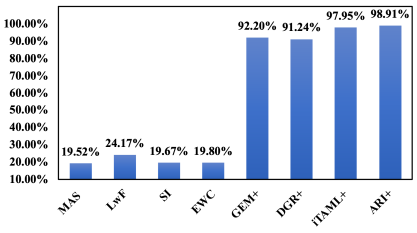

Fig. 4: Comparison results on the MNIST dataset. "+" indicates that the method is memory-based.

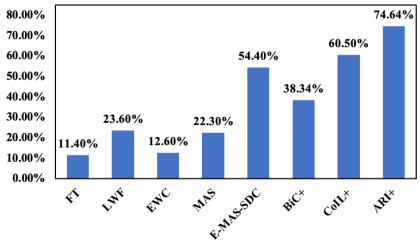

Fig. 5: The average classification accuracy on CUB200, with 6 tasks learned incrementally. "+" indicates that the method is memory-based.

Medium Scale. ARI attains significant advantages on CIFAR100 compared with other state-of-the-art approaches. For the 10-task lifelong learning, as shown in Table 1 ARI achieves the classification accuracy of 80.88% which surpasses all the previous methods.

Table 1: Comparison among different lifelong learning methods on CIFAR100. The accuracy of task t is the average accuracy of all $1, 2, .., t$ tasks.

Dataset	Methods	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7	Task 8	Task 9	Task 10
CIFAR100	DMC[35]	88.11%	76.30%	67.53%	62.19%	57.85%	52.87%	48.59%	43.88%	40.32%	36.28%
	LwF [16]	89.30%	70.13%	54.25%	45.78%	39.83%	36.08%	31.67%	28.86%	24.37%	23.86%
	SI [33]	88.85%	51.76%	40.35%	33.66%	32.01%	29.87%	27.71%	25.97%	24.31	23.51%
	EWC [13]	88.98%	52.37%	48.37%	38.26%	31.64%	26.14%	21.88%	19.94%	18.76%	16.03%
	MAS [2]	88.16%	42.31%	36.16%	35.89%	33.29%	25.97%	21.77%	18.84%	18.11%	15.86%
	RWalk [5]	89.57%	55.12%	40.19%	32.54%	29.13%	25.89%	23.61%	21.84%	19.32%	17.91%
	iCARL [22]	88.74%	78.13%	72.39%	67.23%	63.69%	60.18%	56.35%	54.38%	51.87%	49.46%
	Bic [30]	-	84.70%	-	71.60%	-	63.68%	-	58.12%	-	53.74%
	iTAML [21]	**89.15%**	**89.03%**	**87.32%**	**86.18%**	**84.31%**	**82.12%**	80.65%	79.06%	78.42%	77.79%
	ARI	88.60%	86.90%	85.77%	84.55%	83.10%	81.75%	**81.57%**	**80.98%**	**80.20%**	**80.88%**

We calculate the metrics BWT [17] and FWT [17] to measure forgetting and learning. As shown in Table 2, although the BWT value of GEM is the highest, its accuracy (65.4%) is much lower than ours (80.88%). As mentioned in [17], the BWT and FWT of two methods can indicate their performances only when they have similar accuracies.

We also evaluate ARI's efficiency on CIFAR100. The memory complexity is similar to other memory-based methods. Its extra memory addition is the dictionary to hold the parameters of the specific models. This extra memory is

Table 2: Comparison of forgetting metrics on CIFAR100.

	UCIR	GEM	PODNet[6]	iTAML	iTAML+RRR	ARI
BWT	-8.5%	**1.2%**	-16.3%	-11.5%	-8.5%	-7.5%
FWT	-5.56%	0.47%	-5.58%	0.14%	0.77%	**1.18%**

only about 100MB, which is negligible compared with the memory requirement during training (7200MB). Its time complexity increases by 20% due to the background adversary. The CIFAR100 experiment takes 3.3 hours by one TITAN XP when total epochs=70 and batch size=512.

In Fig. 5, We compare different methods on CUB200 with 6 incremental tasks where BiC [30] and CoIL [37] are memory-based methods. ARI surpasses CoIL by 14.14%, which illustrates that ARI is less prone to catastrophic forgetting.

Large Scale. We compare ARI with the state-of-the-art algorithms on the large scale dataset ImageNet100. The comparison results are listed in Table 3, where Mem% denotes the proportion of the memory size $\mathbb{M}$ in the Imagenet100 training set. ARI outperforms Fixed representations (FixedRep) and other methods of lifelong learning. ARI increases the accuracy on ImageNet100 by 5.22% (from 74.10% to 79.32%).

Table 3: Comparison of different approaches on ImageNet100.

Datasets	Methods	Accuracy	Mem%
ImageNet100	iCaRL [22]	63.50%	2%
	UCIR [11]	69.09%	2%
	MARK [12]	69.43%	10%
	DER [31]	66.70%	2%
	RPSnet [20]	74.10%	2%
	ARI	**79.32%**	3%

The above results illustrate ARI's consistent effectiveness and superiority on small, medium, and large scale datasets over other methods.

4.4 Ablation Study

In this section, we conduct extensive experiments to verify the effects of the proposed background attack and task-specific model fusion.

Similar characteristics lead to forgetting. First we provide a toy experiment to demonstrate that retroactive interference leads to forgetting, as shown in Fig. 6. We construct a dataset with 10 categories, each containing 100 training and 50 test images. We replace the training image backgrounds with similar backgrounds but do not change the test images. We form 5 tasks, each with 2

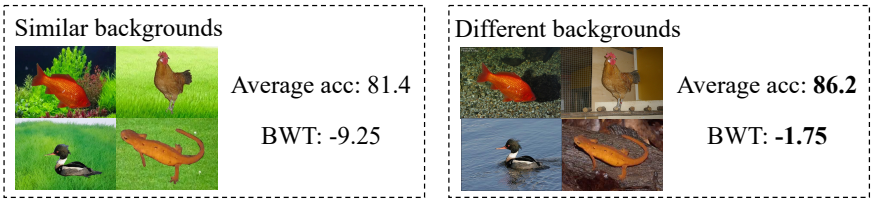

Fig. 6: Similar characteristics lead to the forgetting of lifelong learning.

categories. Average accuracy and BWT are evaluated after all tasks are trained, and the result is compared with its counterpart from the original training images with different backgrounds. The results show that similar characteristics cause forgetting.

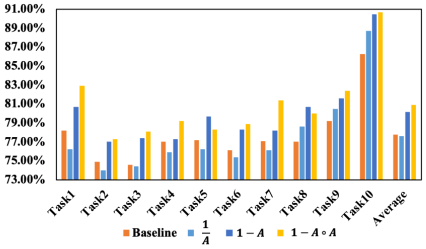

Fig. 7: The impacts of various types of background mask on the lifelong learning process on CIFAR100. The baseline is the model without the attack.

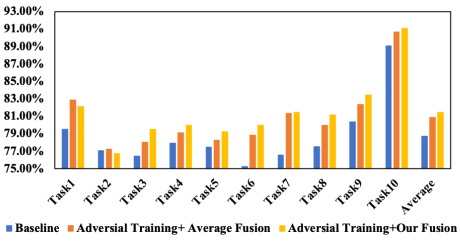

Fig. 8: The classification accuracy in lifelong learning process on CIFAR100. The baseline is the same model but without the attack and with average model fusion.

The effects of different B. During the adversarial training process, we apply different ways to make background attacks. We set the mask $\mathbf{B}$ as $(\frac{1}{\mathbf{A}})$, $(1-\mathbf{A})$ and $(1-\mathbf{A}\circ\mathbf{A})$, and perform experiments respectively. As shown in Fig. 7, the performance varies according to the background mask $\mathbf{B}$. The results performed by $(1-\mathbf{A}\circ\mathbf{A})$ are better than other methods. In order to analyze the cause of the various impact, we randomly sample 5 attention masks and list their distributions in Table 4. Since the masks have values close to 0, the attack with $\mathbf{B}=(\frac{1}{\mathbf{A}})$ would be so huge that it decreases the robustness of the model. Moreover, because the values are closer to 1 than to 0, $\mathbf{B}=(1-\mathbf{A}\circ\mathbf{A})$ can widen the distance between foreground and background more effectively than $\mathbf{B}=1-\mathbf{A}$. In our experiments, the attack using $\mathbf{B}=(1-\mathbf{A}\circ\mathbf{A})$ performs the best, meaning that it can guide background attack more effectively. The visualization results of $\mathbf{A}$ and $\mathbf{B}$ are presented in the supplementary material.

The effects of model fusion. To verify the effect of our proposed model fusion method, we compare the lifelong learning results with and without our model fusion on CIFAR100 while keeping the other settings unchanged. The results are shown in Fig. 8. It could be observed that our fusion operation makes the learning better incrementally. To verify whether the task-specific models tend to be similar, we test the values of **dif**. In Fig. 9, we intercept the 90 – 100 epochs on the CUB200 benchmark. The task number n equals 6 as shown in Eq. 6. The vertical axis represents the distances between task-specific models and the base model. As the training progresses, the distance gradually converges to 0. Through our model fusion method, different task-specific models can converge to the optimal one, thus eliminating information loss and retroactive interference in the task-specific model fusion, which illustrates the effectiveness of our method.

Table 4: We conduct 5 tests and analyze the data distribution of **A**. *std* denotes the standard deviation. The majority of the values are closer to 1 than to 0.

	test1	test2	test3	test4	test5
$[0, 0.3)$	2.03%	1.19%	4.22%	2.62%	1.72%
$[0.3, 0.5)$	7.62%	5.86%	17.43%	12.62%	4.65%
$[0.5, 0.7)$	75.03%	29.42%	59.59%	55.43%	81.93%
$[0.7, 1]$	15.32%	63.54%	18.76%	29.33%	11.70%
std	14.35%	13.72%	17.48%	15.14%	14.52%

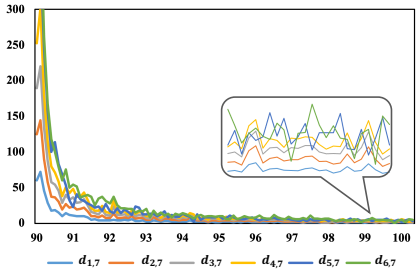

Fig. 9: The distance between task-specific models and base model on CUB200.

5 Conclusion

Lifelong learning aims to learn a single model that can continuously adapt to the new knowledge without overriding existing knowledge. We develop a meta-learning approach to train a base model which can be efficiently optimized for lifelong learning. First, a background attack method is introduced to extract critical features and avoid retroactive interference. Then, an adaptive weight fusion mechanism is presented according to the distances between the base and the task-specific models. Our experiments demonstrate consistent improvements across a range of classification datasets, including ImageNet100, CUB200, CIFAR100, and MNIST.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant 62076016.

References

1. Abati, D., Tomczak, J., Blankevoort, T., Calderara, S., Cucchiara, R., Bejnordi, B.E.: Conditional channel gated networks for task-aware continual learning. In: CVPR (2020)
2. Aljundi, R., Babiloni, F., Elhoseiny, M., Rohrbach, M., Tuytelaars, T.: Memory aware synapses: Learning what (not) to forget. In: ECCV (2018)
3. Ausubel, D.P., Fitzgerald, D.: The role of discriminability in meaningful learning and retention. Journal of Educational Psychology **52**(5), 266 (1961)
4. Castro, F.M., Marín-Jiménez, M.J., Guil, N., Schmid, C., Alahari, K.: End-to-end incremental learning. In: ECCV (2018)
5. Chaudhry, A., Dokania, P.K., Ajanthan, T., Torr, P.H.: Riemannian walk for incremental learning: Understanding forgetting and intransigence. In: ECCV (2018)
6. Douillard, A., Cord, M., Ollion, C., Robert, T., Valle, E.: Podnet: Pooled outputs distillation for small-tasks incremental learning. In: ECCV (2020)
7. Ebrahimi, S., Petryk, S., Gokul, A., Gan, W., Gonzalez, J., Rohrbach, M., Darrell, T.: Remembering for the right reasons: Explanations reduce catastrophic forgetting. In: ICLR (2021)
8. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: ICLR (2015)
9. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT press (2016)
10. Hadsell, R., Rao, D., Rusu, A.A., Pascanu, R.: Embracing change: Continual learning in deep neural networks. Trends in Cognitive Sciences (2020)
11. Hou, S., Pan, X., Loy, C.C., Wang, Z., Lin, D.: Learning a unified classifier incrementally via rebalancing. In: CVPR (2019)
12. Hurtado, J., Raymond-Saez, A., Soto, A.: Optimizing reusable knowledge for continual learning via metalearning. In: NeurIPS (2021)
13. Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al.: Overcoming catastrophic forgetting in neural networks. Proceedings of the National Academy of Sciences **114**(13), 3521–3526 (2017)
14. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images. Tech. rep. (2009)
15. Kurakin, A., Goodfellow, I., Bengio, S.: Adversarial examples in the physical world. In: ICLR (2016)
16. Li, Z., Hoiem, D.: Learning without forgetting. IEEE Transactions on Pattern Analysis and Machine Intelligence **40**(12), 2935–2947 (2017)
17. Lopez-Paz, D., Ranzato, M.: Gradient episodic memory for continual learning. In: NeurIPS (2017)
18. Na, T., Ko, J.H., Mukhopadhyay, S.: Cascade adversarial machine learning regularized with a unified embedding. In: ICLR (2017)
19. Peng H, Long F, D.C.: Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. IEEE Transactions on Pattern Analysis and Machine Intelligence pp. 1226–1238 (2005)
20. Rajasegaran, J., Hayat, M., Khan, S.H., Khan, F.S., Shao, L.: Random path selection for continual learning. In: NeurIPS (2019)
21. Rajasegaran, J., Khan, S., Hayat, M., Khan, F.S., Shah, M.: itaml: An incremental task-agnostic meta-learning approach. In: CVPR (2020)
22. Rebuffi, S.A., Kolesnikov, A., Sperl, G., Lampert, C.H.: icarl: Incremental classifier and representation learning. In: CVPR (2017)

23. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International Journal of Computer Vision **115**(3), 211–252 (2015)
24. Shin, H., Lee, J.K., Kim, J., Kim, J.: Continual learning with deep generative replay. In: NeurIPS (2017)
25. Sternberg, R.J., Sternberg, K., Mio, J.: Cognitive psychology. Cengage Learning Press (2012)
26. Tramèr, F., Boneh, D., Kurakin, A., Goodfellow, I., Papernot, N., McDaniel, P.: Ensemble adversarial training: Attacks and defenses. In: ICLR (2018)
27. Vogel, E.K., McCollough, A.W., Machizawa, M.G.: Neural measures reveal individual differences in controlling access to working memory. Nature **438**(7067), 500–503 (2005)
28. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The caltech-ucsd birds-200. Technical Report CNS-TR-2010-001 (2011)
29. Woo, S., Park, J., Lee, J.Y., Kweon, I.S.: Cbam: Convolutional block attention module. In: ECCV (2018)
30. Wu, Y., Chen, Y., Wang, L., Ye, Y., Liu, Z., Guo, Y., Fu, Y.: Large scale incremental learning. In: CVPR (2019)
31. Yan, S., Xie, J., He, X.: Der: Dynamically expandable representation for class incremental learning. In: CVPR (2021)
32. Yu, L., Twardowski, B., Liu, X., Herranz, L., Wang, K., Cheng, Y., Jui, S., Weijer, J.v.d.: Semantic drift compensation for class-incremental learning. In: CVPR (2020)
33. Zenke, F., Poole, B., Ganguli, S.: Continual learning through synaptic intelligence. In: ICML (2017)
34. Zhang, C., Zhang, M., Zhang, S., Jin, D., Zhou, Q., Cai, Z., Zhao, H., Yi, S., Liu, X., Liu, Z.: Delving deep into the generalization of vision transformers under distribution shifts. arXiv (2021)
35. Zhang, J., Zhang, J., Ghosh, S., Li, D., Tasci, S., Heck, L., Zhang, H., Kuo, C.C.J.: Class-incremental learning via deep model consolidation. In: WACV (2020)
36. Zhao, B., Xiao, X., Gan, G., Zhang, B., Xia, S.T.: Maintaining discrimination and fairness in class incremental learning. In: CVPR (2020)
37. Zhou, D., Ye, H., Zhan, D.: Co-transport for class-incremental learning. In: ACM MM (2021)