AFEC: Active Forgetting of Negative Transfer in Continual Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

Continual learning aims to learn a sequence of tasks from dynamic data distributions. Without access to the old training samples, knowledge transfer from the old tasks to each new task is difficult to determine, which might be either positive or negative. If the old knowledge interferes with the learning of a new task, i.e. the forward knowledge transfer is negative, then precisely remembering the old tasks will further aggravate the interference, thus decreasing the performance of continual learning. By contrast, biological neural networks can actively forget the old knowledge that conflicts with the learning of a new experience, through regulating the learning-triggered synaptic expansion-convergence. Inspired by the biological active forgetting, we propose to actively forget the old knowledge that limits the learning of new tasks to benefit continual learning. Under the framework of Bayesian continual learning, we develop a novel approach named Active Forgetting with synaptic Expansion-Convergence (AFEC), which is formally consistent with the biological weight updating mechanism. We extensively evaluate AFEC on a variety of continual learning benchmarks, including CIFAR10 regression tasks, visual classification tasks and Atari reinforcement tasks, where AFEC achieves the state-of-the-art performance through effectively improving the learning of new tasks. Further, AFEC can be a plug-and-play approach that significantly boosts the performance of representative continual learning strategies.

1 Introduction

The ability to continually learn numerous tasks from dynamic data distributions is critical for deep neural networks, which needs to remember the old tasks by avoiding catastrophic forgetting [19] and effectively learn each new task by improving forward knowledge transfer [18]. Although numerous efforts in continual learning have been devoted to overcoming catastrophic forgetting, to more effectively learn each new task on the basis of remembering the old tasks is under addressed. In particular, since the old training samples are usually unavailable after learning a task, knowledge transfer from the old tasks to each new task is difficult to determine, which might be either positive or negative. If the forward knowledge transfer is *negative*, i.e., learning a new task from the old knowledge is worse than learning the new task on a randomly-initialized network [36, 18], then precisely remembering the old tasks will severely interfere with the learning of the new task, thus decreasing the performance of continual learning.

By contrast, *memory flexibility* is one of the key features of biological memory, which can effectively learn new experiences and remember the old knowledge at the same time, even if they conflict with each other [5]. This advantage is achieved by *active forgetting* of the old knowledge that interferes with the learning of new experiences [29, 5]. The latest data suggested that the signaling pathway that regulates active forgetting achieves its function through regulating a synaptic expansion-

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

convergence process (Fig. 1, see Appendix A for neuroscience background and our biological data). Specifically, learning a new experience triggers the expansion of biological synapses to create additional functional connections (synaptic expansion), which learn the new experience together with the previously-learned functional connections. Then, these functional connections are pruned to the amount before learning (synaptic convergence), where the previously-learned connections and the expanded connections compete with each other to survive. Therefore, the old knowledge encoded in the previously-learned connections is actively forgotten.

Inspired by such biological strategy, we propose to actively forget the old knowledge that interferes with the learning of new tasks without significantly increasing catastrophic forgetting, so as to benefit the ability of continual learning. Specifically, under the framework of Bayesian continual learning, we actively forget the posterior distribution that absorbs all the information of the old tasks with a forgetting factor to improve the learning of new tasks. Then we derive a novel approach named Active Forgetting with synaptic Expansion-Convergence (AFEC), which is formally consistent with the weight updating mechanism of biological active forgetting. Beyond regular weight regularization approaches [13, 1, 35, 2] that selectively penalize changes of the important parameters for the old tasks, AFEC applies a set of expanded parameters to learn each incremental task together with the main network parameters, where the expanded parameters can learn the new task without constraints of the old tasks to avoid potential negative transfer. Then, the forgetting factor regulates a penalty to selectively merge the main network parameters with the corresponding expanded parameters to learn a better overall representation of both the old tasks and the new tasks.

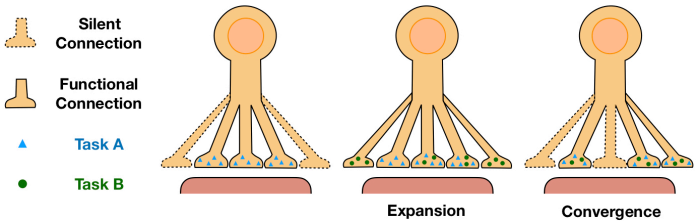

Figure 1: The biological active forgetting is achieved by regulating the learning-triggered synaptic expansion-convergence.

We evaluate AFEC first on continual learning of CIFAR10 [14] regression tasks, and then on continual learning of visual classification tasks, including three variants of CIFAR100 [14], CUB-200-2011 [31] and ImageNet-100 [9], and Atari reinforcement tasks [11], where AFEC achieves the state-of-the-art (SOTA) performance. We empirically validate that the performance improvement results from effectively improving forward knowledge transfer without increasing catastrophic forgetting. Further, AFEC can be a *plug-and-play* method that significantly boosts the performance of representative continual learning strategies, such as weight regularization [13, 1, 35, 2] and memory replay [22, 9, 6].

Our contributions include: (1) Inspired by the biological active forgetting, we propose a novel approach to actively forget the old knowledge that interferes with the learning of new tasks to improve continual learning; (2) Our method effectively improves the learning of new tasks, and achieves the SOTA performance of continual learning on CIFAR10 regression tasks, visual classification tasks and Atari reinforcement tasks in a plug-and-play way; and (3) To the best of our knowledge, we are the first to model the biological active forgetting at the level of synaptic structures. The efficacy of our method suggests a potential theoretical explanation for the mechanism of biological active forgetting and its function in memory flexibility.

2 Related Work

Continual learning needs to minimize catastrophic forgetting and maximize forward knowledge transfer. Existing work in continual learning mainly focuses on mitigating catastrophic forgetting. Representative approaches include: weight regularization [13, 1, 35, 2], which selectively penalizes changes of the previously-learned parameters; parameter isolation [25, 11], which allocates a dedicated parameter subspace for each task; and memory replay [22, 28, 9], which approximates and replays the old data distributions through storing old training data, their embedding or learning a generative model (we provide a conceptual analysis in Appendix B). In particular, Adaptive Group Sparsity based Continual Learning (AGS-CL) [11] proposed to regularize the group sparsity with separation of the important nodes for the old tasks to prevent catastrophic forgetting, which toke advantages of weight regularization and parameter isolation, and achieved the SOTA performance on various continual learning benchmarks.

Several studies suggested that forward knowledge transfer is critical for continual learning [18, 4], which may be either positive or negative as the training data distributions are changing. Although it is highly nontrivial to mitigate potential negative transfer while overcoming catastrophic forgetting, the efforts that specifically consider this challenging issue are limited. [3] developed a method to mitigate negative transfer when fine-tuning tasks on a pretrained network. [23] tried to maximize transfer and minimize interference from a memory buffer that stored a small amount of old training data. Similarly, [6, 17, 33] attempted to more effectively balance stability and plasticity with the memory buffer in class incremental learning, while [32] stored and updated the old feature embedding. By contrast, since pretraining or old training data might not be available in continual learning, we mainly focus on a more restrict yet realistic setting that a neural network incrementally learns a sequence of tasks *from scratch*, *without* storing old training data. Further, we extend our method to the scenarios where pretraining or memory buffer can be used, as well as the scenarios other than classification tasks, such as regression tasks and reinforcement tasks.

3 Method

In this section, we first describe the framework of Bayesian continual learning [13, 21]. Under such framework, we propose an active forgetting strategy, which is formally consistent with the biological weight updating mechanism of synaptic expansion and synaptic convergence.

3.1 Basics of Bayesian Continual Learning

Continual learning needs to remember the old tasks and learn each new task effectively. Let's consider a simple case that a neural network with parameter θ continually learns two independent tasks, task A and task B, from their training datasets D_A^{train} and D_B^{train} [13]. The training dataset of each task is only available when learning the task.

Bayesian Learning: After learning D_A^{train}, the posterior distribution

$$p(\theta|D_A^{train}) = \frac{p(D_A^{train}|\theta)p(\theta)}{p(D_A^{train})}$$

incorporates the knowledge of task A. Then, we can get the predictive distribution for the test data of task A:

$$p(D_A^{test}|D_A^{train}) = \int p(D_A^{test}|\theta)p(\theta|D_A^{train})d\theta.$$

As the posterior $p(\theta|D_A^{train})$ is generally intractable (except very special cases), we must resort to approximation methods, such as the Laplace approximation [13] or other approaches of approximate inference [21]. Let's take Laplace approximation as an example. If $p(\theta|D_A^{train})$ is smooth and majorly peaked around the mode $\theta_A^* = \arg\max_\theta \log p(\theta|D_A^{train})$, we can approximate it with a Gaussian distribution whose mean is θ_A^* and covariance is the inverse Hessian of the negative log posterior (detailed in Appendix C.1).

Bayesian Continual Learning: Next, we want to incorporate the new task into the posterior, which uses the posterior $p(\theta|D_A^{train})$ as the prior of the next task [13]:

$$p(\theta|D_A^{train}, D_B^{train}) = \frac{p(D_B^{train}|\theta)\, p(\theta|D_A^{train})}{p(D_B^{train})}. \tag{1}$$

Then we can test the performance of continual learning by evaluating

$$p(D_A^{test}, D_B^{test}|D_A^{train}, D_B^{train}) = \int p(D_A^{test}, D_B^{test}|\theta)p(\theta|D_A^{train}, D_B^{train})d\theta. \tag{2}$$

Similarly, $p(\theta|D_A^{train}, D_B^{train})$ can be approximated by a Gaussian using Laplace's approximation whose mean is the mode of the posterior:

$$\theta_{A,B}^* = \arg\max_\theta \log\, p(\theta|D_A^{train}, D_B^{train}) \tag{3}$$

$$= \arg\max_\theta \log\, p(D_B^{train}|\theta) + \log\, p(\theta|D_A^{train}) - \underbrace{\log\, p(D_B^{train})}_{const.}. \tag{4}$$

This MAP estimation is also known as the Elastic Weight Consolidation (EWC) [13]:

$$L_{\mathrm{EWC}}(\theta) = L_{\mathrm{B}}(\theta) + \frac{\lambda}{2}\sum_i F_{A,i}(\theta_i - \theta^*_{A,i})^2, \tag{5}$$

where $L_{\mathrm{B}}(\theta)$ is the loss for task B and i is the label of each parameter. F_A is the Fisher Information matrix (FIM) of θ^*_A on D_A^{train} (detailed in Appendix C.1), which indicates the "importance" of parameter i for task A. The hyperparameter λ explicitly controls the penalty that selectively merges each θ_i to $\theta^*_{A,i}$ to alleviate catastrophic forgetting.

3.2 Active Forgetting with Synaptic Expansion-Convergence

However, if precisely remembering task A interferes with the learning of task B, e.g., task A and task B are too different, it might be useful to *actively forget* the original data, similar to the biological strategy of active forgetting. Based on this inspiration, we introduce a forgetting factor β and replace $p(\theta|D_A^{train})$ that absorbs all the information of D_A^{train} with a weighted product distribution [8, 20]:

$$p_m(\theta|D_A^{train}, \beta) = \frac{p(\theta|D_A^{train})^{(1-\beta)}p(\theta)^{\beta}}{Z}, \tag{6}$$

where we use m to denote that we are 'mixing' $p(\theta|D_A^{train})$ and $p(\theta)$ to produce the new distribution p_m, and Z is the normalizer that depends on β. The normalizer keeps $p_m(\theta|D_A^{train}, \beta)$ following a Gaussian distribution if $p(\theta|D_A^{train})$ and $p(\theta)$ are both Gaussian (detailed in Appendix C.2). When $\beta \to 0$, p_m will be dominated by $p(\theta|D_A^{train})$ and remember all the information about task A. When $\beta \to 1$, p_m will actively forget all the information about task A. Modified from Eqn. (2), our target becomes:

$$p(D_A^{test}, D_B^{test}|D_A^{train}, D_B^{train}, \beta) = \int p(D_A^{test}, D_B^{test}|\theta)p(\theta|D_A^{train}, D_B^{train}, \beta)d\theta. \tag{7}$$

We first need to determine β, which decides how much information from task A is forgotten to maximize the probability of learning task B well. A good β should be as follows:

$$\beta^* = \arg\max_{\beta} p(D_B^{train}|D_A^{train}, \beta) = \arg\max_{\beta} \int p(D_B^{train}|\theta)p_m(\theta|D_A^{train}, \beta)d\theta. \tag{8}$$

Since the integral is difficult to solve, we can make a grid search to determine β, which should be between 0 and 1. Next, we use a Laplace's approximation of $p(\theta|D_A^{train}, D_B^{train}, \beta)$ (detailed in Appendix C.3), and the MAP estimation is

$$\begin{aligned}\theta^*_{A,B} &= \arg\max_{\theta} \log\, p_m(\theta|D_A^{train}, D_B^{train}, \beta) \\ &= \arg\max_{\theta} (1-\beta)\left(\log\, p(D_B^{train}|\theta) + \log\, p(\theta|D_A^{train})\right) + \beta\, \log\, p(\theta|D_B^{train}) + const..\end{aligned} \tag{9}$$

Then we obtain the loss function of Active Forgetting with synaptic Expansion-Convergence (AFEC):

$$L_{\mathrm{AFEC}}(\theta) = L_{\mathrm{B}}(\theta) + \frac{\lambda}{2}\sum_i F_{A,i}(\theta_i - \theta^*_{A,i})^2 + \frac{\lambda_e}{2}\sum_i F_{e,i}(\theta_i - \theta^*_{e,i})^2, \tag{10}$$

where θ_e are the expanded parameters only for the new task, optimized by $L_{\mathrm{B}}(\theta_e)$. Two weight-merging regularizers selectively merge each θ_i with $\theta^*_{A,i}$ for the old tasks and $\theta^*_{e,i}$ for the new task. Similar to [13], we use a hyperparameter λ to explicitly control the penalty that prevents catastrophic forgetting, while β is integrated into a hyperparameter $\lambda_e \propto \beta/(1-\beta)$ to control the penalty that promotes active forgetting. Therefore, to actively forget the original posterior in Eqn. (6) results in expanding a set of parameter θ_e only for the new task, and the subsequent convergence of parameters through the weight-merging regularization, formally consistent with the weight updating mechanism of biological active forgetting. Since the proposed active forgetting is integrated into the third term, our method can be used in a *plug-and-play* way to improve continual learning. Also, here we use Laplace's approximation to approximate the intractable posteriors, which can be other strategies of approximate inference [21] in further work. Note that θ^*_e and F_e are *not* stored in continual learning, and the architecture of the main network is *fixed*. Thus, AFEC does not cause additional storage cost compared with regular weight regularization approaches such as [13, 35, 1, 2].

Next, we conceptually analyze how AFEC mitigates potential negative transfer in continual learning in Fig. 2. After learning task A, regular weight regularization approaches [13, 35, 1, 2] need to copy and save the old parameters, and approximate the importance of each parameter to task A. Then, the network continually learns task B, penalizing the deviance of each parameter from the old parameter, weighted by its importance. However, if task A conflicts with task B, such strategy will further interfere with the learning of task B. Different from regular weight regularization approaches, AFEC expands a set of parameters only for task B without constraints of the old tasks. Then, both the main network parameters and the expanded parameters learn the new task. Meanwhile, the main network parameters selectively merge with both the old parameters and the expanded parameters, dependent on their contributions to the overall representations of task A and task B. The old knowledge is actively forgotten through the competition between the old parameters and the extended parameters in the selective weight-merging.

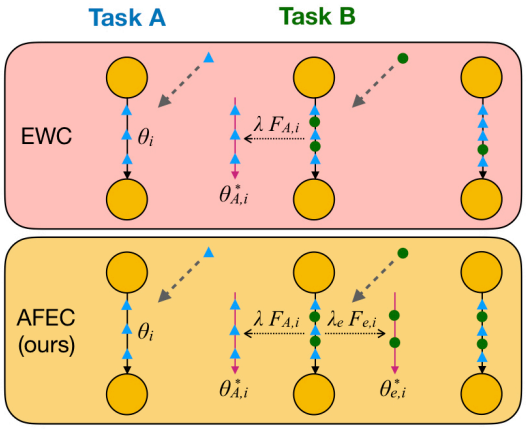

Figure 2: Conceptual analysis of AFEC in continual learning. The yellow circle represents a neuron, and the arrow between two neurons represents a synapse.

4 Experiment

In this section, we evaluate AFEC on a variety of continual learning benchmarks, including: CIFAR10 regression tasks, which is a toy experiment with mutual negative transfer between tasks; visual classification tasks, where the forward knowledge transfer can be either positive or negative, with small- or large-scale images; and Atari reinforcement tasks, where the forward knowledge transfer is severely negative.[1] All the experiments are averaged by 5 runs with different random seeds.

4.1 CIFAR10 Regression Tasks

First, we propose CIFAR10 regression tasks to verify the efficacy of AFEC to alleviate negative transfer in continual learning. CIFAR10 dataset [14] contains 50,000 training samples and 10,000 testing samples of 10-class colored images of size 32×32. The regression task is to evenly map the ten classes around the origin of the two-dimensional coordinates and train the neural network to predict the angle of the origin to each class (see Fig. 3). We change the relative position of the ten classes to construct different regression tasks with mutual negative transfer, in which remembering the old knowledge severely interferes with the learning of a new task.

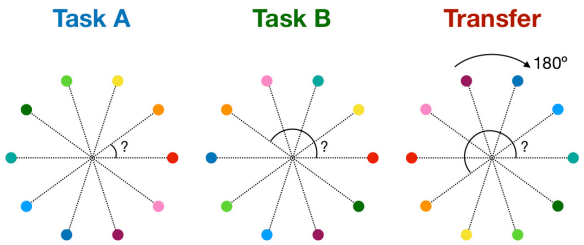

Figure 3: CIFAR10 regression tasks. Each circle represents the position of a class. Task A and Task B use different relative positions. Transfer applies the same relative position as Task A, but rotates by several phases.

As shown in Fig. 4, a two-layer LeNet network continually learns two different regression tasks. After learning the first task, simply fine-tuning the network to learn the second task results in severe catastrophic forgetting, where the performance on the first task is almost random prediction ($\sim 10\%$). Representative weight regularization approaches, such as MAS [1] and EWC [13], can effectively remember the old tasks, but limits the learning of new tasks. In particular, larger strength of the weight regularization results in better performance of the first task but worse performance of the second task. By contrast, AFEC improves the learning of new tasks on the basis of effectively remembering the old tasks, so as to achieve better average accuracy of continual learning. Note that EWC is equal to AFEC

[1] Our code is included in Supplementary Materials.

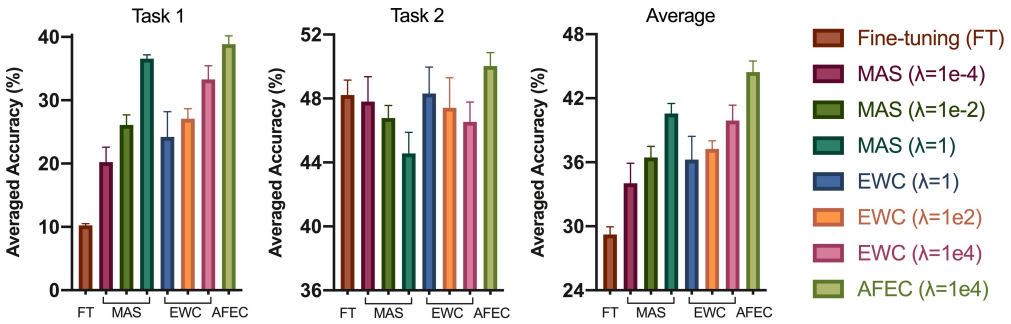

Figure 4: Continual learning of two CIFAR10 regression tasks. Larger strength of weight regularization can better remember the old task while limits the learning of the new task. By contrast, AFEC (ours) can more effectively learn a new task while remembering the old task.

($\beta = 0$), which is the ablation of the proposed active forgetting, so the performance improvement of AFEC on EWC proves the effectiveness of our approach. Similarly, we use EWC as the ablation of active forgetting in the following experiments. Then we evaluate AFEC on a variety of architectures and larger amounts of tasks in Table. 1. AFEC substantially improves EWC on four commonly-used classifier architectures and continual learning of ten regression tasks.

Table 1: Continual learning of CIFAR10 regression tasks with various architectures. We present the averaged accuracy (%) of five runs for two-task and ten-task, and of five rotations for transfer experiment.

	Methods	LeNet [16]	VGG11 [30]	VGG11BN [30]	ResNet10 [7]
Two-Task	Fine-tuning	29.23 $\pm$0.72	46.37 $\pm$0.11	46.54 $\pm$0.29	60.67 $\pm$1.52
	EWC [13]	39.91 $\pm$1.44	73.55 $\pm$1.26	82.00 $\pm$0.32	71.94 $\pm$1.61
	AFEC (ours)	**44.45** $\pm$1.03	**77.76** $\pm$0.09	**86.07** $\pm$0.24	**75.67** $\pm$1.19
Ten-Task	Fine-tuning	46.57 $\pm$0.68	18.03 $\pm$0.03	18.08 $\pm$0.04	54.97 $\pm$1.33
	EWC [13]	49.95 $\pm$1.81	79.39 $\pm$1.12	85.98 $\pm$0.07	82.91 $\pm$0.22
	AFEC (ours)	**53.50** $\pm$1.70	**82.50** $\pm$0.47	**88.31** $\pm$0.11	**85.33** $\pm$0.31
Transfer	Fine-tuning	38.93 $\pm$0.80	80.37 $\pm$0.40	84.30 $\pm$0.10	85.69 $\pm$0.93
	EWC [13]	35.87 $\pm$0.87	76.66 $\pm$0.44	82.25 $\pm$0.11	84.96 $\pm$0.91
	AFEC (ours)	**40.90** $\pm$1.35	**83.81** $\pm$0.42	**86.30** $\pm$0.17	**87.80** $\pm$0.66

Further, we evaluate the ability of transfer learning on CIFAR10 regression tasks. After continual learning of two different regression tasks, the neural network fixes the embedding layer and only fine-tunes a linear output layer to learn a new task that is similar to the first task. Specifically, the similar task applies the same relative position as the first task, but rotates for $60°$, $120°$, $180°$, $240°$ and $300°$ (we average the performance of the five rotations). Therefore, the neural network can learn the similar task well if it effectively remembers and transfers the relative position learned in the first task. As shown in Table 1, AFEC can more effectively learn the similar task, while EWC is even worse than sequentially fine-tuning without weight regularization.

4.2 Visual Classification Tasks

Dataset: We evaluate continual learning on a variety of benchmark datasets for visual classification, including CIFAR100, CUB-200-2011 and ImageNet-100. CIFAR100 [14] contains 100-class colored images of the size 32×32, where each class includes 500 training samples and 100 testing samples. CUB-200-2011 [31] is a large-scale dataset including 200 classes and 11,788 colored images of birds. ImageNet-100 [9] is a subset of iILSVRC-2012 [24], consisting of randomly selected 100 classes of images and 1300 samples per class. We follow the regular preprocessing pipeline of CUB-200-2011 and ImageNet-100 as [11], which randomly resizes and crops the images to the size of 224×224 before experiment.

Benchmark: We consider five representative benchmarks of visual classification tasks to evaluate continual learning in different aspects. The first three are on CIFAR100, with forward knowledge transfer from more negative to more positive (detailed in Fig. 5), while the second two are on large-scale images. (1) CIFAR100-SC [34]: CIFAR100 can be split as 20 superclasses (SC) with 5 classes per superclass dependent on semantic similarity, where each superclass can be a classification task. Since the superclasses are semantically different, forward knowledge transfer in such a task sequence is relatively more negative. (2) CIFAR100 [22]: The 100 classes in CIFAR100 are randomly split as 20 classification tasks with 5 classes per task. (3) CIFAR10/100 [11]: The 10-class CIFAR10 are randomly split as 2 classification tasks with 5 classes per task, followed by 20 tasks with 5 classes per task randomly split from CIFAR100. This benchmark is adapted from [11] to keep the number

Table 2: Averaged accuracy (%) of all the tasks learned so far in continual learning of visual classification tasks, averaged by 5 different random seeds (see Appendix D for error bar). * AFEC is our method described in Sec. 3.2, while *w/* AFEC is the adaptation of our method to representative weight regularization methods.

	CIFAR100-SC		CIFAR100		CIFAR10/100		CUB-200 w/ PT		CUB-200 w/o PT		ImageNet-100	
Methods	A_{10}	A_{20}	A_{10}	A_{20}	A_2	A_{2+20}	A_5	A_{10}	A_5	A_{10}	A_5	A_{10}
Fine-tuning	32.58	28.40	40.92	33.53	78.96	37.81	78.75	78.13	31.91	39.82	50.56	44.80
AGS-CL [11]	55.19	53.19	71.24	69.99	86.27	80.42	82.30	81.84	32.69	40.73	51.48	47.20
EWC [13]	52.25	51.74	68.72	69.18	85.07	77.75	81.37	80.92	32.90	42.29	76.12	73.82
* AFEC (ours)	**56.28**	**55.24**	**72.36**	**72.29**	**86.87**	**81.25**	**83.65**	**82.04**	**34.36**	43.05	**77.64**	75.46
MAS [1]	52.76	52.18	67.60	69.41	84.97	77.39	79.98	79.67	31.68	42.56	75.48	74.72
w/ AFEC (ours)	55.26	54.89	69.57	71.20	86.21	80.01	82.77	81.31	34.08	42.93	75.64	**75.66**
SI [35]	52.20	51.97	68.72	69.21	85.00	76.69	80.14	80.21	33.08	42.03	73.52	72.97
w/ AFEC (ours)	55.25	53.90	69.34	70.13	85.71	78.49	83.06	81.88	34.04	**43.20**	75.72	74.14
RWALK [2]	50.51	49.62	66.02	66.90	85.59	73.64	80.81	80.58	32.56	41.94	73.24	73.22
w/ AFEC (ours)	52.62	51.76	68.50	69.12	86.12	77.16	83.24	81.95	33.35	42.95	74.64	73.86

of class per task the same as benchmark (1, 2), where the large amounts of training data in the first two CIFAR10 tasks bring a relatively more positive transfer. (4) CUB-200 [11]: The 200 classes in CUB-200-2011 are randomly split as 10 classification tasks with 20 classes per task, where the training samples are relatively limited (around 60 images per class). (5) ImageNet-100 [22]: The 100 classes in ImageNet-100 are randomly split as 10 classification tasks with 10 classes per task.

Architecture: We follow [11] to use a CNN architecture with 6 convolution layers and 2 fully connected layers for benchmark (1, 2, 3), and AlexNet [15] for benchmark (4, 5). Since one purpose of continual learning is to learn a usable model from incrementally collected data, we learn the network from scratch. Following [11], we also try AlexNet with ImageNet pretraining for CUB-200.

Baseline: First, we consider a restrict yet realistic setting of continual learning *without* access to the old training data, and perform multi-head evaluation [2]. Since AFEC is a weight regularization approach, we mainly compare with the representative approaches that follow a similar idea, such as EWC [13], MAS [1], SI [35] and RWALK [2]. We also compare with AGS-CL [11], the SOTA method that takes advantage of weight regularization and parameter isolation. We reproduce the results of all the baselines from the officially released code of [11], where we do an extensive search of hyperparameters and report the best performance for fair comparison (detailed in Appendix D). Then, we relax the restriction of using old training data and plug AFEC in representative memory replay approaches, where we perform single-head evaluation [2] (detailed in Appendix G).

Averaged Accuracy: In Table 2, we summarize the averaged accuracy of all the tasks learned so far during continual learning of visual classification tasks. AFEC achieves the best performance in all the continual learning benchmarks and significantly outperforms EWC [13], i.e. the ablation of active forgetting in AFEC. In particular, AGS-CL [11] is the SOTA method on relatively small-scale images and CUB-200 with ImageNet pretraining (CUB-200 w/ PT). While, AFEC achieves a better performance than AGS-CL on small-scale images from scratch and CUB-200 w/ PT, and substantially outperforms AGS-CL on the two benchmarks of large-scale images from scratch. Further, since regular weight regularization approaches are generally in a re-weighted weight decay form, AFEC can be easily adapted to such approaches (the adaptation is detailed in Appendix F) and effectively boost their performance on the benchmarks above.

Knowledge Transfer: Next, we evaluate knowledge transfer in the three continual learning benchmarks developed on CIFAR100 in Fig. 5. We first present the accuracy of learning each new task during continual learning, where AFEC learns each new task significantly better than other baselines. Since continual learning of more tasks leads to less network resources for a new task, the overall trend of all the baselines is *declining*, indicating the necessity to improve forward knowledge transfer on the basis of overcoming catastrophic forgetting. Then we calculate forward transfer (FWT) [18], i.e. the averaged influence that learning the previous tasks has on a future task, and backward transfer (BWT) [18], i.e. the averaged influence that learning a new task has on the previous tasks (see Appendix E for computation details). FWT is from more negative to more positive in CIFAR100-SC, CIFAR100

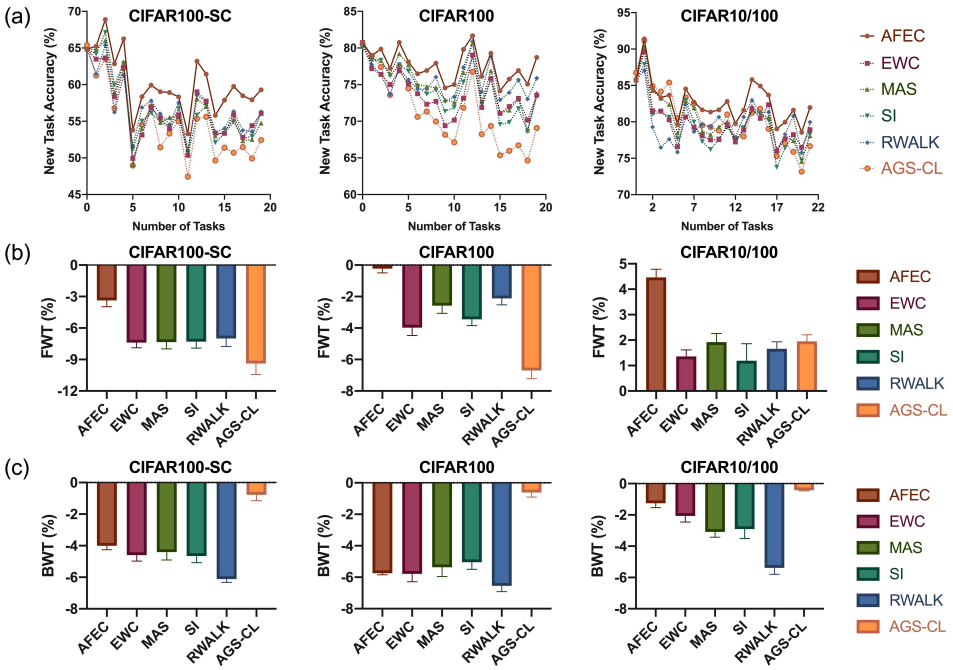

Figure 5: Knowledge transfer in continual learning. (a) The accuracy of each new task in continual learning. (b) Forward Transfer (FWT), which is from more negative to more positive on CIFAR100-SC, CIFAR100 and CIFAR10/100. (c) Backward Transfer (BWT).

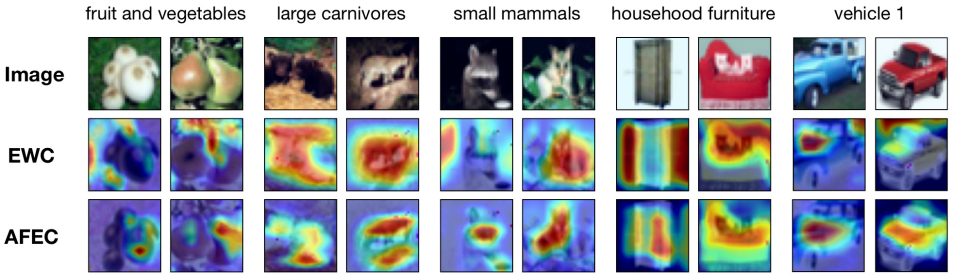

Figure 6: Visualization of predictions of the latest task after continual learning on CIFAR100-SC. We present the results on five different random seeds, which determine five different superclasses.

and CIFAR10/100, while AFEC achieves the highest FWT among all the baselines. The BWT of AFEC is comparable as EWC, indicating that the proposed active forgetting does not cause additional catastrophic forgetting. Therefore, the performance improvement of AFEC is achieved by effectively improving the learning of new tasks in continual learning. Due to the progressive parameter isolation, although AGS-CL achieves the best BWT, its ability to learn each new task drops more rapidly than other baselines. Thus, it underperforms AFEC in Table 2.

Visual Explanation: To explicitly show how AFEC improves continual learning, in Fig. 6 we use Grad-CAM [27] to visualize predictions of the latest task after continual learning on CIFAR100-SC, where FWT is more negative as discussed above. The predictions of EWC overfit the background information since it attempts to best remember the old tasks with severe negative transfer, which limits the learning of new tasks. By contrast, the visual explanation of AFEC is significantly better than EWC, indicating the efficacy of the proposed active forgetting to address potential negative transfer and benefit the learning of new tasks.

Plugging-in Memory Replay: We further consider plugging AFEC in representative memory replay approaches with single-head evaluation [2] in Appendix G. On CIFAR100 and ImageNet-100, we follow [9, 6] to first learn 50 classes and then continually learn the other 50 classes by 5 phases (10 classes per phase) or 10 phases (5 classes per phase) with a small memory buffer of 2000 images. AFEC substantially boosts the performance of representative memory replay approaches such as iCaRL [22], LUCIR [9] and PODNet [6].

4.3 Atari Reinforcement Tasks

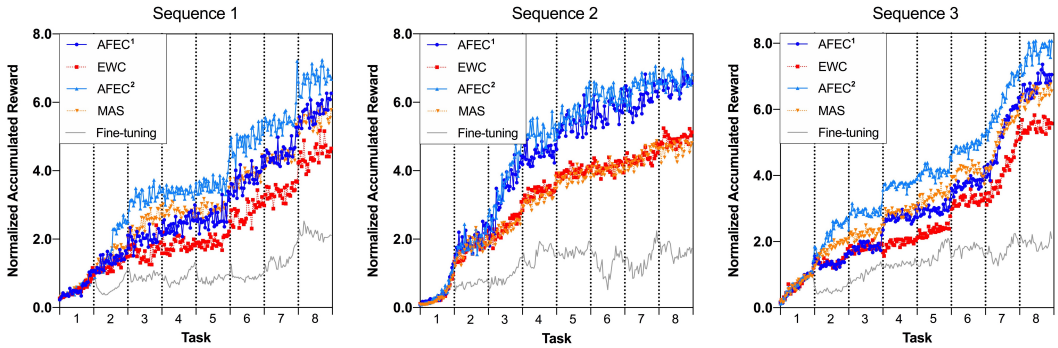

Figure 7: Continual learning of Atari reinforcement tasks. AFEC^1 is our method described in Sec. 3.2, while AFEC^2 is the adaptation of our method to MAS. AFEC^1 and AFEC^2 substantially boost the performance of EWC and MAS, respectively.

Next, we evaluate the performance of AFEC in continual learning of Atari reinforcement tasks (Atari games). We follow the same implementations as [11] to sequentially learn eight randomly selected Atari games, and reproduce all the baselines with the officially released code. Specifically, we applies a CNN architecture consisting of 3 convolution layers with 2 fully connected layers and identical PPO [26] for all the methods (detailed in Appendix H). The evaluation metric is the normalized accumulated reward: the evaluated rewards are normalized with the maximum reward of fine-tuning in each task and accumulated. We observe that the performance is largely influenced by random initialization and the order of Atari games, so we average the performance by five runs with different random seeds, and present the results on three different orders of the task sequence.

For continual learning of Atari reinforcement tasks, negative transfer is usually much more severe than visual classification tasks, possibly because the optimal policies of each Atari games are highly different. We first measure and average the normalized rewards of learning each task with a randomly initialized network, which are 2.16, 1.44 and 1.68 on the three task sequences, respectively. That is to say, the initialization sequentially learned from the old tasks results in an averaged performance decline by 53.67%, 30.66% and 40.56%, compared with random initialization. Then, we evaluate the maximum reward of learning each new task in Table 3, and the normalized accumulated reward of continual learning in Fig. 7. AFEC effectively improves the learning of new tasks and thus boosts the performance of EWC and MAS in continual learning, particularly when learning more incremental tasks. AFEC also achieves a significantly better performance than the reproduced results of AGS-CL on its officially released code [11] (see Appendix H for an extensive analysis).

Table 3: Averaged performance increase of learning each new task on Atari reinforcement tasks.

	Sequence 1	Sequence 2	Sequence 3
AFEC^1 on EWC	+35.28%	+50.55%	+28.00%
AFEC^2 on MAS	+30.09%	+61.12%	+26.63%

5 Conclusion

In this work, we draw inspirations from the biological active forgetting and propose a novel approach to improve forward knowledge transfer in continual learning.[2] Our method achieves the SOTA performance on a variety of continual learning benchmarks through effectively improving the learning of new tasks. Moreover, our method can be a plug-and-play approach to boost the performance of representative continual learning strategies. Interestingly, active forgetting of the posterior distribution that absorbs all the information of the old tasks can derive a weight updating strategy formally consistent with the synaptic expansion and synaptic convergence (detailed Appendix A), and is functionally consistent with the advantage of biological active forgetting in memory flexibility [5]. This connection provides a potential theoretical explanation for the mechanism of biological active forgetting and its cognitive function. We will further explore it on artificial neural networks and biological neural networks in the future.

[2] We discuss the limitations and potential negative social impacts in Appendix K.

References

[1] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 139–154, 2018.

[2] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 532–547, 2018.

[3] Xinyang Chen, Sinan Wang, Bo Fu, Mingsheng Long, and Jianmin Wang. Catastrophic forgetting meets negative transfer: Batch spectral shrinkage for safe transfer learning. 2019.

[4] Natalia Díaz-Rodríguez, Vincenzo Lomonaco, David Filliat, and Davide Maltoni. Don't forget, there is more than forgetting: new metrics for continual learning. *arXiv preprint arXiv:1810.13166*, 2018.

[5] Tao Dong, Jing He, Shiqing Wang, Lianzhang Wang, Yuqi Cheng, and Yi Zhong. Inability to activate rac1-dependent forgetting contributes to behavioral inflexibility in mutants of multiple autism-risk genes. *Proceedings of the National Academy of Sciences*, 113(27):7644–7649, 2016.

[6] 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)*, volume 12365, pages 86–102, 2020.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[8] Geoffrey E Hinton. Products of experts. 1999.

[9] 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*, pages 831–839, 2019.

[10] Xinting Hu, Kaihua Tang, Chunyan Miao, Xian-Sheng Hua, and Hanwang Zhang. Distilling causal effect of data in class-incremental learning. *arXiv preprint arXiv:2103.01737*, 2021.

[11] Sangwon Jung, Hongjoon Ahn, Sungmin Cha, and Taesup Moon. Continual learning with node-importance based adaptive group sparse regularization. *arXiv e-prints*, pages arXiv–2003, 2020.

[12] Steven M Kay. *Fundamentals of statistical signal processing*. Prentice Hall PTR, 1993.

[13] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[14] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. 25:1097–1105, 2012.

[16] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[17] Yaoyao Liu, Bernt Schiele, and Qianru Sun. Meta-aggregating networks for class-incremental learning. *arXiv preprint arXiv:2010.05063*, 2020.

[18] David Lopez-Paz et al. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, pages 6467–6476, 2017.

[19] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989.

[20] Thomas Minka. Power ep. Technical report, Technical report, Microsoft Research, Cambridge, 2004.

[21] Cuong V Nguyen, Yingzhen Li, Thang D Bui, and Richard E Turner. Variational continual learning. *arXiv preprint arXiv:1710.10628*, 2017.

[22] 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*, pages 2001–2010, 2017.

[23] Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. *arXiv preprint arXiv:1810.11910*, 2018.

[24] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[25] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

[26] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[27] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 618–626, 2017.

[28] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems*, pages 2990–2999, 2017.

[29] Yichun Shuai, Binyan Lu, Ying Hu, Lianzhang Wang, Kan Sun, and Yi Zhong. Forgetting is regulated through rac activity in drosophila. *Cell*, 140(4):579–589, 2010.

[30] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

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

[32] Shipeng Wang, Xiaorong Li, Jian Sun, and Zongben Xu. Training networks in null space of covariance for continual learning. *arXiv preprint arXiv:2103.07113*, 2021.

[33] Shipeng Yan, Jiangwei Xie, and Xuming He. Der: Dynamically expandable representation for class incremental learning. *arXiv preprint arXiv:2103.16788*, 2021.

[34] Jaehong Yoon, Saehoon Kim, Eunho Yang, and Sung Ju Hwang. Scalable and order-robust continual learning with additive parameter decomposition. *arXiv preprint arXiv:1902.09432*, 2019.

[35] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3987–3995. JMLR. org, 2017.

[36] Wen Zhang, Lingfei Deng, and Dongrui Wu. Overcoming negative transfer: A survey. *arXiv preprint arXiv:2009.00909*, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See the abstract and introduction with the manuscript.
   (b) Did you describe the limitations of your work? [Yes] See Appendix K.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Appendix K.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes] We carefully read the ethics review guidelines and ensured that our paper conforms to them.
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Sec. 3.
   (b) Did you include complete proofs of all theoretical results? [Yes] See Sec. 3, Appendix C and F.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Our code is included in supplementary materials.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Sec. 4 and Appendix D, G and H.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Sec. 4.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix J.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Sec. 4.
   (b) Did you mention the license of the assets? [Yes] See Appendix I.
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A] We do not include new assets.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] We do not include new assets.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] We do not use crowdsourcing or conducted research with human subjects.
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] We do not use crowdsourcing or conducted research with human subjects.
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] We do not use crowdsourcing or conducted research with human subjects.