# Saliency-Augmented Memory Completion for Continual Learning

Guangji Bai* Chen Ling* Yuyang Gao* Liang Zhao*

**Abstract**

Continual Learning (CL) is considered a key step toward next-generation Artificial Intelligence. Among various methods, replay-based approaches that maintain and replay a small episodic memory of previous samples are one of the most successful strategies against catastrophic forgetting. However, since forgetting is inevitable given bounded memory and unbounded tasks, 'how to forget' is a problem continual learning must address. Therefore, beyond simply avoiding (catastrophic) forgetting, an under-explored issue is how to reasonably forget while ensuring the merits of human memory, including 1) storage efficiency, 2) generalizability, and 3) some interpretability. To achieve these simultaneously, our paper proposes a new saliency-augmented memory completion framework for continual learning, inspired by recent discoveries in memory completion/separation in cognitive neuroscience. Specifically, we innovatively propose to store the part of the image most important to the tasks in episodic memory by saliency map extraction and memory encoding. When learning new tasks, previous data from memory are inpainted by an adaptive data generation module, which is inspired by how humans "complete" episodic memory. The module's parameters are shared cross all tasks and it can be jointly trained with a continual learning classifier as bilevel optimization. Extensive experiments on several continual learning and image classification benchmarks demonstrate the proposed method's effectiveness and efficiency.

## 1 Introduction

An important step toward next-generation Artificial Intelligence (AI) is a promising new domain known as continual learning (CL), where neural networks learn continuously over a sequence of tasks, similar to the way humans learn [20]. Compared with traditional supervised learning, continual learning is still in its very primitive stage. Currently, the primary goal is essentially to avoid *Catastrophic Forgetting* [19] of previously learned tasks when an agent is learning new tasks. Continual learning aims to mitigate forgetting while updating the model over a stream of tasks.

To overcome this issue, researchers have proposed a number of different strategies. Among various approaches, the *replay-based methods* are arguably more effective in terms of performance and bio-inspiration [27] as a way to alleviate the catastrophic forgetting challenge and are thus becoming the preferred approach for continual learning models [24, 2]. However, the performance of these methods highly depends on the size of the episodic memory. Recent work [15] proved that to achieve optimal performance in continual learning, one has to store all previous examples in the memory, which is almost impossible in practice and counters the way how human brain works. Unlike avoiding (catastrophic) forgetting, the attempts on 'how to reasonably forget' is still a highly open question, leading to significant challenges in continual learning, including **1) Memory inefficiency.** The performance of replay-based models depends heavily on the size of the available memory in the replay buffer, which is used to retain as many previous samples as possible. While existing works typically store the entire sample in memory, we humans seldom memorize every detail of our experiences. Thus, compared to biological neural networks, some mechanisms must still be missing in current models; **2) Insufficient generalization power.** The primary focus of existing works is to avoid catastrophic forgetting by memorizing all the details without taking into account their usefulness for learning tasks. They typically rely on episodic memory for individual tasks without sufficient chaining to make the knowledge they learn truly generalizable to all potential (historical and future unseen) tasks. In contrast, human beings significantly improve generalizability during continual learning; **3) Obscurity of the memory and its importance to learning tasks.** Human being usually has a concise and clear clue on how the relevant memory is useful for the learning tasks. Such a clue could even be helpful for telling if the human has the necessary memory to be capable of a specific task at all. However, the majority of existing works in CL pay little attention to pursuing such transparency of continual learning models. For example, most works directly feed the images into their model for training without any explanation generated, which may prevent users from understanding which semantic features in the image are the most decisive ones and how the model is reasoning to make the final prediction. In addition, without an explanation generation mechanism, it is much

*Emory University, Atlanta, GA. {guangji.bai, chen.ling, yuyang.gao, liang.zhao}@emory.edu.

more challenging for model diagnosing or debugging, let alone understanding how knowledge is stored and refined through the continual learning process.

To jointly address these challenges, this paper proposes **S**aliency-**A**ugmented **M**emory **C**ompletion (**SAMC**) framework for continual learning, which is inspired by the *memory pattern completion* theory in cognitive neuroscience. The memory pattern completion process guides the abstraction of learning episodes (tasks) into semantic knowledge as well as the reverse process in recovering episodes from the memorized abstracts. Specifically, in this paper, instead of memorizing all historical training samples, we memorize their interpretable abstraction in terms of the saliency maps that most determine the prediction outputs for each learning task. **Our contribution includes,** 1). We develop a novel neural-inspired continual learning framework to handle the catastrophic forgetting. 2). We propose two techniques based on saliency map and image inpainting methods for efficient memory storage and recovery. 3). We design a bilevel optimization algorithm with theoretical guarantee to train our entire framework in an end-to-end manner. 4). We demonstrate our model's efficacy and superiority with extensive experiments.

## 2 Related Work

**Continual Learning (CL).** Catastrophic forgetting is a long-standing problem [27] in continual learning which has been recently tackled in a variety of visual tasks such as image classification [14, 24], object detection [34], etc.

Existing techniques in CL can be divided into three main categories [20]: 1) *regularization-based approaches*, 2) *dynamic architectures* and 3) *replay-based approaches*. Regularization-based approaches alleviates catastrophic forgetting by either adding a regularization term to the objective function [14] or knowledge distillation over previous tasks [17]. Dynamic architecture approaches adaptively accommodate the network architecture (e.g., adding more neurons or layers) in response to new information during training. Dynamic architectures can be explicit, if new network branches are grown, or implicit, if some network parameters are only available for certain tasks. Replay-based approaches alleviate the forgetting of deep neural networks by replaying stored samples from the previous history when learning new ones, and has been shown to be the most effective method for mitigating catastrophic forgetting.

Replay-based methods mainly include three directions: namely rehearsal methods, constrained optimization and pseudo rehearsal. *Rehearsal methods* directly retrieve previous samples from a limited size memory together with new samples for training [9]. While simple in nature, this approach is prone to overfitting the old samples from the memory. As an alternative, *constrained optimization* methods formulate backward/forward transfer as constraints in the objective function. GEM [18] constrains new task updates to not interfere with previous tasks by projecting the estimated gradient on the feasible region outlined by previous task gradients through first order Taylor series approximation. A-GEM [8] further extended GEM and made the constraint computationally more efficient. EPR [28] uses zero-padding for memory efficiency while lacks theoretical guarantee on its performance. Finally, *pseudo-rehearsal* methods typically utilize generative model such as GAN [13] or VAE [22] to generate previous samples from random inputs and have shown the ability to generate high-quality images recently [27]. Readers may refer to [20] for a more comprehensive survey on continual learning.

**Saliency Detection.** Saliency detection is to identify the most informative part of input features. It has been applied to various domains including CV [4, 12], NLP [25], etc. The salience map approach is exemplified by [32] to test a network with portions of the input occluded to create a map showing which parts of the data have influence on the output. For example, Class Activation Mapping (CAM, [33]) modifies image classification CNN architectures by replacing fully-connected layers with convolutional layers and global average pooling, thus achieving class-specific feature maps. Grad-CAM [29] generalizes CAM by visualizing the linear combination of the last convolutional layer's feature map activations and label-specific weights, which are calculated by the gradient of prediction w.r.t the feature map activations.

## 3 Problem Formulation

Continual learning is defined as an online supervised learning problem. Following the learning protocol in [18], we consider a training set $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_T\}$ consisting of $T$ tasks, where $\mathcal{D}_t = \{(\mathbf{x}_i^{(t)}, \mathbf{y}_i^{(t)})\}_{i=1}^{n_t}$ contains $n_t$ input-target pairs $(\mathbf{x}_i^{(t)}, \mathbf{y}_i^{(t)}) \in \mathcal{X} \times \mathcal{Y}$. While each learning task arrives sequentially, we make the assumption of *locally i.i.d*, i.e., $\forall\ t, (\mathbf{x}_i^{(t)}, \mathbf{y}_i^{(t)}) \overset{iid}{\sim} P_t$, where $P_t$ denotes the data distribution for task $t$ and *i.i.d* for *independent and identically distributed*.

Given such a stream of tasks, our **goal** is to train a learning agent $f_\theta : \mathcal{X} \to \mathcal{Y}$, parameterized by $\theta$, which can be queried *at any time* to predict the target $\mathbf{y}$ given associated unseen input $\mathbf{x}$ and task id $t$. Moreover, we require that such a learning agent can only store a small amount of seen samples in an episodic memory $\mathcal{M}$ with fixed budget. Under the goal, we are interested in how to achieve *continual learning without forgetting* problem setting defined as following: Given predictor $f_\theta$, the loss

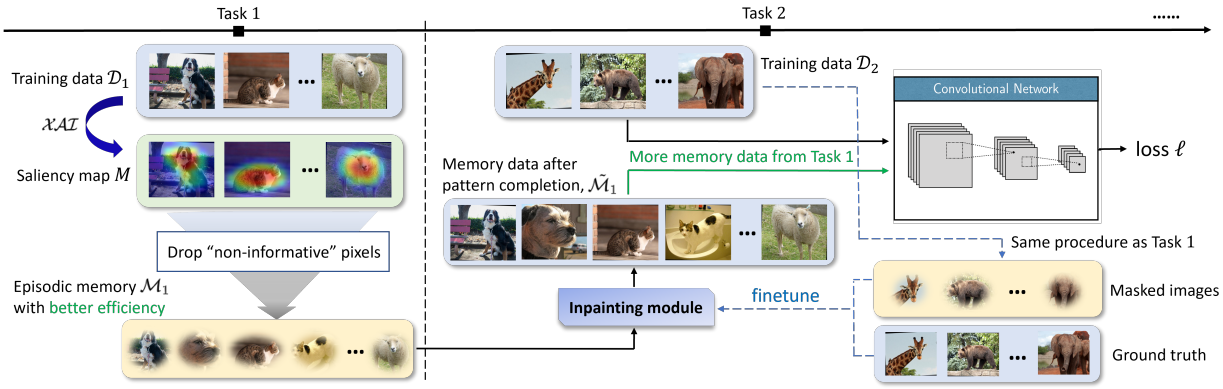


Figure 1: **An illustrative overview of the proposed Saliency-Augmented Memory Completion (SAMC) architecture with two tasks.** At task 1, we generate saliency map to extract the informative regions in each image and only store them into episodic memory. At task 2, we apply inpainting model to recover masked images and feed them together with current data into the model for training. We follow same procedure as at task 1 for image extraction and utilize both masked images and ground truth to finetune our inpainting model.

on the episodic memory of task $k$ is defined as

$$\ell(f_\theta, \mathcal{M}_k) \coloneqq |\mathcal{M}_k|^{-1} \sum_{(\mathbf{x}_i, k, \mathbf{y}_i)} \phi(f_\theta(\mathbf{x_i}, k), \mathbf{y}_i), \tag{3.1}$$

$\forall\, k < t$, where $\phi$ can be *e.g.* cross-entropy or MSE. We consider *constrained optimization* to avoid the losses from increasing, which in turn allows the so called *positive backward transfer* [18]. More specifically, when observing the triplet $(\mathbf{x}, \mathbf{y}, t)$ from the current task $t$, we solve the following *inequality-constrained* problem:

$$\begin{aligned} &\min_\theta \ell\big(f_\theta(\mathbf{x}, t), \mathbf{y}\big), \quad \text{s.t.,} \\ &\ell\big(f_\theta, \mathcal{M}_k\big) \le \ell\big(f_\theta^{t-1}, \mathcal{M}_k\big), \end{aligned} \tag{3.2}$$

where $f_\theta^{t-1}$ denotes the predictor state at the end of learning task $t-1$ and $k = 1, 2, \cdots, t-1$. After the training of task $t$, a subset of training samples will be stored into the episodic memory, i.e., $\mathcal{M} = \mathcal{M} \cup \{(\mathbf{x}_i^{(t)}, \mathbf{y}_i^{(t)})\}_{i=1}^{m_t}$, where $m_t$ is the memory buffer size for the current task. During the training of task $t+1$, previously stored samples will serve as $\mathcal{M}_t$ in Eq. 3.2.

Our goal above poses significant challenges to existing work: 1). The performance of replay-based methods highly depends on the memory size. Existing work typically considered storing entire samples into the memory, which was inefficient in practice. 2). Memorizing all the details could be problematic, and how to capture the most important and generalizable knowledge through episodic memory is under-explored. 3). Interpretability and transparency over $f_\theta$ are typically ignored in existing work, which results in an obscure model with insufficient explanations on how the model is reasoning towards its predictions and how knowledge is stored and refined through the entire continual learning process.

## 4 Proposed Method

In this section, we introduce our proposed Saliency-augmented Memory Completion (SAMC) that addresses all the challenges mentioned earlier and thus narrows the gap between AI and human learning. We innovatively utilize Explainable AI ($\mathcal{XAI}$) methods to select the most salient regions for each image and only store the selected pixels in our episodic memory, thus achieving controllable and better memory efficiency. During the training phase, we leverage learning-based image inpainting techniques for memory completion, where each partially stored image will be restored by the inpainting model. A bilevel optimization algorithm is designed and allows our entire framework to be trained in an end-to-end manner. Our theoretical analyses show that, as long as we properly select the salient region for each image and the inpainting model is well trained, we can simultaneously achieve better memory efficiency and positive backward transfer with a theoretical guarantee.

### 4.1 Saliency-augmented Episodic Memory.

In this section, we demonstrate how to leverage the saliency map generated by saliency methods to select the "informative" region of each image and how we design the episodic memory structure to store the images efficiently.

Saliency-based methods generate visual explanation via saliency map, which can be regarded as the impact of specific feature map activations on a given prediction. Given an input image $I \in \mathbb{R}^{H \times W \times C}$, a classification ConvNet $f_\theta$ predicts $I$ belongs to class $c$ and produces the class score $f_\theta^c(I)$ (short as $f_\theta^c$). The saliency map $M_I \in \mathbb{R}^{H \times W}$ is generated by assigning high intensity values to the relevant image regions that contribute more

to the model's prediction, i.e.,

$$M_I = \Phi(I, f_\theta, c), \tag{4.3}$$

where $\Phi$ can be any visual explainer that can generate saliency map as Eq. 4.3. In this paper, we consider Grad-CAM as our choice of $\Phi$, due to that it can pass important "sanity checks" regarding their sensitivity to data and model parameters [1], which differentiate Grad-CAM with many other explanation methods [10].

Suppose we are training on task $t$ with predictor state $f_\theta^{t-1}$. Given a mini-batch $\mathcal{B}_t = \{(\mathbf{x}_i^{(t)}, \mathbf{y}_i^{(t)})\}_{i=1}^{bsz}$ from the current task's data $\mathcal{D}_t$ ($bsz$ denotes mini-batch size), we generate the saliency map $M$ following Eq. 4.3. Intuitively speaking, pixels with higher magnitude in saliency map are more likely to be involved in the region of the target class object while those with lower magnitude are more likely to be the non-target objects or background regions. Specifically, for any input pair $(x, y) \in \mathcal{B}_t$, the masked image $x'$ is

$$x' = \mathbb{1}\{M_x > \mu\} = \mathbb{1}\{\Phi(x, f_\theta, y) > \mu\}, \tag{4.4}$$

where $\mathbb{1}\{\cdot\}$ is the indicator function and $\mu$ is the threshold for controlling each image's drop ratio.

We denote the mini-batch after extraction as $\mathcal{B}_t'$. Due to the sparsity in $x'$, it is inefficient to store the entire images in episodic memory. We propose to store extracted images in the format of sparse matrix. Specifically, for each image in $\mathcal{B}_t'$, we first convert it into Coordinate Format (COO) [5], i.e., $[(\text{row}_{x'}, \text{col}_{x'}, \text{value}_{x'})] = \text{COO}(x')$, where the left-hand-side is a list of (rwo,col,value) tuples. After transforming the images into COO format, we store them into the episodic memory as $\mathcal{M}_t = \mathcal{M}_t \cup \text{COO}(\mathcal{B}_t')$. We consider a simple way for adding samples into memory similar to [18], where the samples are added into memory following *first-in-first-out* (FIFO) and the last group of samples are stored in memory in the end.

### 4.2 Memory Completion with Inpainting.

As shown in Eq. 3.2, when we are training on task $t$ we need to calculate the loss on previous samples stored in episodic memory. However, due to the missing pixels, classification models (e.g., CNN) cannot be operated on such ragged images. There are various existing works on how to retrieve images with missing pixels, and we consider the following two approaches:

**Rule-based Inpainting.** First approach is to handle the missing pixels by prescribed rules [7, 30]. One advantage of such methods is that they introduce neither extra model parameters nor training costs. Also, one does not need to worry about any potential forgetting of the inpainting method itself during continual learning since it is purely based on hand-crafted rules. One naive candidate of rule-based inpainting is simply padding all missing pixels with zeros. We have included it as a baseline in this paper for comparison.

**Learning-based Inpainting.** An alternative is to utilize learning-based inpainting models, and several deep learning models have demonstrated their state-of-the-art performance on various image inpainting tasks [31, 21]. However, the trade-off of their excellent performance is the potential higher memory cost and computational inefficiency. Moreover, it is also challenging to ensure the model generalization for each task during training without severe forgetting.

In this work, we consider autoencoder [6] as our learning-based inpainting model since 1) autoencoder has been utilized as the backbone in many state-of-the-art deep learning-based inpainting models [31, 21], which has proven its capability of learning any useful salient features from the masked image, 2) autoencoder is rather a light-weight model which does not bring heavy burden to our framework. Specifically, the autoencoder $\mathcal{A}_\xi(x')$ can be divided into two parts: an encoder $p_{\xi_e}(z|x')$ that is parameterized by $\xi_e$, and a decoder $p_{\xi_d}(\tilde{x}|z)$ that is parameterized by $\xi_d$, where $\xi = \{\xi_e, \xi_d\}$, $x'$ is the masked image, $\tilde{x}$ is the recovered image, and $z$ is a compressed bottleneck (latent variable). Inpainting autoencoder aims at characterizing the conditional probability

$$\mathcal{A}_\xi(x') = p_{\xi_d}(\tilde{x}|z) \cdot p_{\xi_e}(z|x') \tag{4.5}$$

to reconstruct the image $\tilde{x}$ from its masked version $x'$. Moreover, $\mathcal{A}_\xi(\cdot)$ can be trained with the objective of minimizing the reconstruction loss $\text{argmin}_\xi \, ||x^* - \mathcal{A}_\xi(x')||_2^2$, where $x^*$ is the ground truth image.

Furthermore, to maximize the number of samples into the memory $\mathcal{M}_k$, it is imperative to adopt a relatively large drop ratio $\mu$. However, the generalization power of the model would be affected since the details of the original image can hardly be recovered in $\tilde{x}$ under a high drop rate. To ensure the model retains its generalization power under such circumstance, we leverage a novel image inpatinting framework that incorporates both rule-based inpainting $\mathcal{R}(\cdot)$ and the autoencoder $\mathcal{A}_\xi(x')$ to recover the image from coarse to fine. Specifically, given masked image $x'$ as defined in Eq. 4.4, we propagate it through rule-based inpainting module $\mathcal{R}(\cdot)$ to get a coarse prediction and then feed the output after rule-based inpainting into autoencoder $\mathcal{A}_\xi(\cdot)$ for further refinement, i.e., $\tilde{x} = \mathcal{A}_\xi(\mathcal{R}(x'))$. If we are training on task $t$, given masked images stored in episodic memory $\mathcal{M}_k, \forall\, k < t$, we convert them from COO format back to standard sparse image tensor and generate the memory with pattern completion $\tilde{\mathcal{M}}_t$ as

$$\tilde{\mathcal{M}}_k = \mathcal{A}_\xi\Big(\mathcal{R}\big(\text{COO}^{-1}(\mathcal{M}_k)\big)\Big), \quad \forall\, k < t, \tag{4.6}$$

where $\mathrm{COO}^{-1}$ denotes the decoding process from COO format back to sparse matrix.

**4.3 Bilevel Optimization.** In each incremental phase, we optimize two groups of learnable parameters: 1) the model parameter of continual learning classifier, $\theta$, and 2) the inpainting model's parameter $\xi$. Directly optimizing the entire framework could be hard since $\theta$ and $\xi$ are coupled. In this paper, we consider formulating the problem as bilevel optimization, i.e.,

$$\begin{aligned} &\text{Upper:} \quad \xi^* = \operatorname{argmin}_\xi \|\mathbf{x} - \mathcal{A}_\xi(\mathcal{R}(\mathbf{x}'))\|_2^2 \\ &\text{Lower:} \quad \text{s.t.} \quad \theta^* = \operatorname{argmin}_\theta \ell\big(f_\theta(\mathbf{x}, t), \mathbf{y}\big) \\ &\qquad\qquad \text{s.t. } \ell(f_\theta, \tilde{\mathcal{M}}_k) \le \ell(f_\theta^{t-1}, \tilde{\mathcal{M}}_k), \end{aligned} \tag{4.7}$$

where $k < t$ and $\mathbf{x}' = \mathbb{1}\{\Phi(\mathbf{x}, f_{\theta^*}, \mathbf{y}) > \mu\}$. The upper-level corresponds to minimizing the image reconstruction loss on the current task, where each raw image is masked by saliency map and then inpainted by the inpainting model. The lower-level corresponds to the standard objective as Eq. 3.2 for training continual learning classifier $f_\theta$ while the constraints are based on inpainted images $\tilde{\mathcal{M}}_k$ instead of actual images $\mathcal{M}_k$.

**Optimizing $\theta$:** Given $\xi$, the objective function over $\theta$ becomes a constrained optimization problem. To handle the inequality constraints, we follow [18] but with a clearer theoretical explanation.

Definition 4.1. *Given loss function $\ell$ and an input triplet $(x, y, t)$ or episodic memory after completion $\tilde{\mathcal{M}}_k$, define the loss gradient vector as*

$$g \coloneqq \partial \ell(f_\theta(\mathbf{x}, t), \mathbf{y})/\partial\theta, \quad \tilde{g}_k \coloneqq \partial \ell(f_\theta, \tilde{\mathcal{M}}_k)/\partial\theta. \tag{4.8}$$

The first lemma below proves the *sufficiency* of enforcing the constraint over the loss gradient in order to guarantee *positive backward transfer* as in Eq. 3.2.

Lemma 4.1. *$\forall\, k < t$, with small enough step size $\alpha$,*

$$\langle g, \tilde{g}_k \rangle \ge 0 \;\Rightarrow\; \ell(f_\theta, \tilde{\mathcal{M}}_k) \le \ell(f_\theta^{t-1}), \tilde{\mathcal{M}}_k). \tag{4.9}$$

Guaranteed by the above lemma, we approximate the inequality constraints in Eq. 4.7 by computing the angle between $g$ and $\tilde{g}_k$, i.e., $\langle g, \tilde{g}_k \rangle \ge 0$, $\forall\, k < t$. Whenever the angle is greater than 90°, the proposed gradient $g$ will be projected to the closest gradient $\tilde{g}$ in $\ell_2$-norm that keeps the angle within 90°. For more details please refer to [18] due to limited space.

**Optimizing $\xi$:** Following the training procedure of bilevel optimization [11], we take a couple of gradient descent steps over the inpainting model parameter $\xi$ when samples from new task arrive, i.e.,

$$\xi \leftarrow \xi - \beta \cdot \nabla_\xi \|\mathbf{x} - \mathcal{A}_\xi(\mathcal{R}(\mathbf{x}'))\|_2^2, \tag{4.10}$$

where $\beta$ is the step size for $\xi$. It has been shown that *early stopping* in SGD can be regarded as implicit regularization [23], and we adopt this technique over $\xi$ to help mitigate the forgetting of the inpainting model. We only take a few steps of SGD over $\xi$ that helps control any detrimental semantic drift of the autoencoder and thus alleviate potential forgetting.

The next lemma shows that the difference between continual learning model's outputs over the original image and inpainted one can be upper bounded by factors related to Grad-CAM's saliency map and the inpainting model's reconstruction error.

Lemma 4.2. *Given input $x \in \mathbb{R}^d$, suppose $\tilde{x}$ is the inpainted sample generated by our proposed method. Denote $y_x^c$ as $f_\theta$'s prediction over $x$ that belongs to class $c$, and $\Delta x \coloneqq \max_{1\le i\le d} |x_i - \tilde{x}_i|$, then:*

$$err(x, \tilde{x}) \coloneqq |y_x^c - y_{\tilde{x}}^c| \le \mu \cdot \Delta x \cdot d^{-2}. \tag{4.11}$$

Our main theorem below guarantees that, with proper choice of the threshold $\mu$ for pixel dropping and inpainting model with bounded reconstruction error, enforcing the constraint over the memory after pattern completion $\tilde{\mathcal{M}}$ is equivalent to that over the memory with ground-truth images.

Theorem 4.1. *Suppose $\ell$ is smooth with Lipschitz constant $L$, and the first order derivative of $\ell$, i.e., $\partial\ell/\partial\theta$ is Lipschitz continuous with constant $L_\theta$. We have:*

$$\langle g, \tilde{g}_k \rangle > 0 \;\Rightarrow\; \langle g, g_k \rangle > 0, \quad \text{as } \tilde{x} \to x, \tag{4.12}$$

*where $g_k$ is the gradient vector on ground-truth images.*

All formal proofs can be found in the appendix.

**4.4 Memory Complexity.** The complexity to store samples into the memory is $\mathcal{O}(T \cdot (1 - \bar{\mu}) \cdot |\mathcal{M}_t|)$, where $T$ is number of tasks, $\bar{\mu}$ is the average pixel drop ratio determined by $\mu$ in Eq. 4.4, and $|\mathcal{M}_t|$ is the episodic memory size for each task. The greater the $\bar{\mu}$, the lower the memory complexity to store the same amount of samples. If we consider learning-based inpainting method and denote the number of parameters of the inpainting model as $S$, the overall memory cost becomes $\mathcal{O}(T \cdot (1-\bar{\mu}) \cdot |\mathcal{M}_t| + S)$. The cost of storing the inpainting model is often dwarfed by that of storing images, since each image is a high-dimensional tensor and the cost for storing the inpainting model is constant w.r.t $T$.

## 5 Experiment

In this section, we compare our proposed method SAMC with several state-of-the-art CL methods on commonly used benchmarks. More details and additional results can be found in the appendix. Code available at `https://github.com/BaiTheBest/SAMC`

Table 1: Performance comparison (%) on image classification datasets. The means and standard deviations are computed over five random runs. When used, episodic memories contain 10 samples per task.

| Model | Split CIFAR-10 | | Split CIFAR-100 | | Split mini-ImageNet | |
|---|---|---|---|---|---|---|
| | ACC (↑) | BWT (↑) | ACC (↑) | BWT (↑) | ACC (↑) | BWT (↑) |
| finetune | 63.24 ± 2.03 | -17.91 ± 1.03 | 41.06 ± 3.00 | -17.36 ± 2.54 | 33.31 ± 1.67 | -16.40 ± 0.32 |
| EWC | 65.64 ± 1.79 | -18.37 ± 0.98 | 42.22 ± 1.69 | -16.54 ± 2.94 | 32.66 ± 1.91 | -14.83 ± 0.67 |
| LwF | 67.22 ± 1.68 | -16.78 ± 0.73 | 43.79 ± 1.53 | -16.23 ± 2.16 | 34.17 ± 1.85 | -13.72 ± 0.55 |
| iCaRL | 72.53 ± 1.42 | -12.43 ± 0.79 | 48.78 ± 1.01 | -15.94 ± 1.18 | 40.03 ± 1.62 | -12.16 ± 0.42 |
| GEM | 70.82 ± 1.55 | -15.82 ± 0.69 | 50.28 ± 1.28 | -13.56 ± 1.11 | 40.78 ± 1.98 | -9.87 ± 0.36 |
| ER | 72.13 ± 1.78 | -12.78 ± 0.83 | 49.72 ± 1.82 | -13.84 ± 2.59 | 40.04 ± 1.89 | -13.43 ± 0.53 |
| MER | 71.09 ± 1.32 | -13.69 ± 0.56 | 47.37 ± 2.31 | -15.23 ± 1.27 | 39.63 ± 1.77 | -11.93 ± 0.48 |
| EBM | 72.20 ± 1.77 | -12.26 ± 0.61 | 51.29 ± 1.28 | -11.03 ± 1.29 | 39.52 ± 1.48 | -11.75 ± 0.44 |
| EEC | 73.48 ± 1.91 | -12.13 ± 0.57 | 53.03 ± 1.65 | -10.69 ± 1.32 | 37.32 ± 1.99 | -12.46 ± 0.56 |
| **SAMC (ours)** | **75.37 ± 1.21** | **-11.12 ± 0.63** | **56.16 ± 1.01** | **-8.24 ± 1.47** | **46.98 ± 1.42** | **-4.12 ± 0.21** |

Table 2: Comparison of different pattern completion methods. Episodic memories contain 10 samples per task.

| Method | Split CIFAR-10 | | Split CIFAR-100 | | Split mini-ImageNet | |
|---|---|---|---|---|---|---|
| | ACC (↑) | BWT (↑) | ACC (↑) | BWT (↑) | ACC (↑) | BWT (↑) |
| Zero-padding | 68.43 ± 1.69 | -16.73 ± 0.83 | 47.91 ± 1.41 | -14.32 ± 0.93 | 38.73 ± 1.79 | -11.62 ± 0.53 |
| Rule-based inpainting | 73.63 ± 1.43 | -12.93 ± 0.45 | 54.27 ± 1.23 | -10.26 ± 0.85 | 44.10 ± 1.51 | -6.72 ± 0.32 |
| Autoencoder (w./o rule-based) | 71.02 ± 1.53 | -13.90 ± 0.75 | 51.92 ± 1.68 | -12.73 ± 1.02 | 42.63 ± 1.81 | -7.83 ± 0.59 |
| Autoencoder (w./ rule-based) | **75.37 ± 1.21** | **-11.12 ± 0.63** | **56.16 ± 1.01** | **-8.24 ± 1.47** | **46.98 ± 1.42** | **-4.12 ± 0.21** |

**5.1 Experiment Setups. Datasets.** We perform experiments on three widely-used image classification datasets in continual learning: Split CIFAR-10, Split CIFAR-100, and Split mini-ImageNet [9]. CIFAR-10 consists of 50,000 RGB training images and 10,000 test images belonging to 10 object classes. Similar to CIFAR-10, CIFAR-100, except it has 100 classes containing 600 images each. ImageNet-50 is a smaller subset of the iLSVRC-2012 dataset containing 50 classes with 1300 training images and 50 validation images per class. For Split CIFAR-10, we consider 5 tasks where each task contains two classes. For Split CIFAR-100 and Split mini-ImageNet, we consider 20 tasks where each task includes five classes. The image size for CIFAR is $32 \times 32$ and for mini-ImageNet is $84 \times 84$.

**Architectures.** Following [18], we use a reduced ResNet18 with three times fewer feature maps across all layers on all three datasets. Similar to [18, 9], we train and evaluate our algorithm in 'multi-head' setting where a task id is used to select a task-specific classifier head. For learning-based inpainting method, we consider the inpainting autoencoder with three layer encoder and three layer decoder with convolutional structure, respectively. Please refer to appendix for more detail.

**Comparison Methods.** We compare our method with several groups of continual learning methods, including:

- **Practical Baselines:** *1) Finetune*, a popular baseline, naively trained on the data stream.
- **Regularization methods:** *2) EWC* [14], a well-known regularization-based method; *3) LwF* [17], another regularization-based method for CNNs.
- **Replay-based Methods:** *4) iCaRL* [24], a class-incremental learner that classifies using a nearest exemplar algorithm; *5) GEM* [18], a replay approach based on an episodic memory of parameter gradients; *6) ER* [9], a simple yet competitive experience method based on reservoir sampling; *7) MER* [26], another replay approach inspired by meta-learning.
- **Pseudo-rehearsal Methods:** *8) EBM* [16], an energy-based method for continual learning; *9) EEC* [3], an autoencoder-based generative method.

**Evaluation Metrics.** We evaluate the classification performance using the ACC metric, which is the average test classification accuracy of all tasks. We report backward transfer, i.e., BWT [18] to measure the influence of new learning on the past knowledge. Negative BWT indicates forgetting so the bigger the better. We put detailed experimental settings in the appendix.

**5.2 Experiment Results. Classification Performance.** Table 1 shows the overall experimental results, where the memory size per task is set to 10 for all datasets, so the total memory buffer size is 50, 200, 200 on CIFAR-10, CIFAR-100 and mini-ImageNet, respectively. On each dataset, our proposed SAMC outperforms the baselines by significant margins, and the gains in performance are especially substantial on more challenging datasets where number of tasks is large and the sample size for each task is small, e.g., CIFAR-100 and mini-ImageNet. Specifically, our model achieves ∼12% and ∼15% relative improvement ratio in accuracy, while achieves ∼5% and ∼6% lower backward transfer

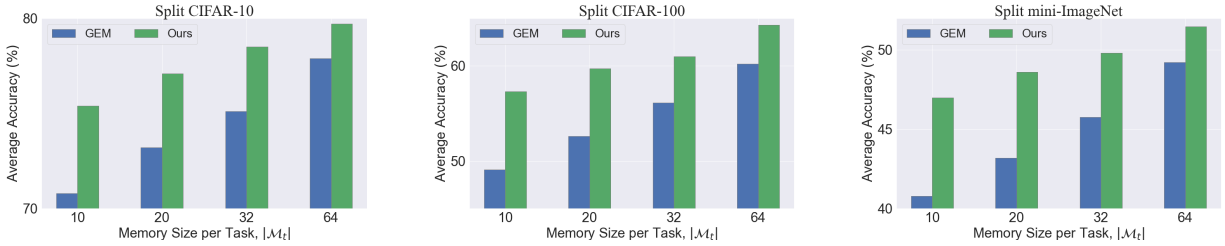


Figure 2: **Effects of memory size.** We compare ACC for varying memory size per task on Split CIFAR-10, Split CIFAR-100 and Split mini-ImageNet, respectively. Number of tasks for three datasets are 5, 20, and 20.

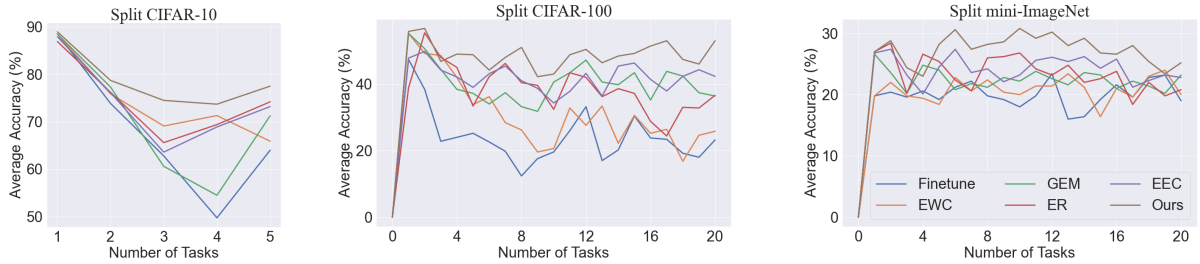


Figure 3: **Evolution of test accuracy at the first task, as more tasks are learned.** We compare ACC of the first task over the entire training process on Split CIFAR-10, Split CIFAR-100 and Split mini-ImageNet. When used, episodic memories contain 10 samples per task. The legend is same and shown in the right sub-figure only.

compared with the second best method on CIFAR-100 and mini-ImageNet, respectively. This substantial performance improvement can be attributed to better memory efficiency and generalizability provided by SAMC. EWC [14] performs relatively poor without multiple passes over the dataset and only achieves similar accuracy as finetune baseline. GEM is favored most on CIFAR-100 where it outperformed other methods by some margin. Experience replay methods like ER and MER achieved better performance on CIFAR-100. which is consistent with recent studies on tiny size memory [9]. Pseudo-rehearsal methods [16, 3], though resemble to our proposed method in a way, lack the interpretability and suffer from forgetting in the generative model, so achieve lower accuracy than SAMC.

**Comparison of Pattern Completion Methods.** We investigate the effectiveness of different inpainting techniques on all three datasets. As shown in Table 2, naive zero-padding achieves relatively poor performance due to the fact that an image with a large number of pixels with zero value could change the model's prediction dramatically. In fact, our theoretical analysis in Lemma 4.2 tells that the more the inpainted pixel value differs from the original one, the larger $\Delta x$ will be, which results in a more significant prediction error. The rule-based method can achieve consistent results on all the datasets, which may be attributed to its zero forgetting. Autoencoder alone performs a bit worse than rule-based method possibly due to its forgetting, but with the help of rule-based prepcossing and early stopping autoencoder can achieve the best performance.

**Effects of Memory Size.** As shown in Figure 2, we compare our proposed method SAMC with GEM [18] over varying memory buffer size. Both GEM [18] and SAMC benefit from larger memory size since the accuracy is an increasing function of memory size for both methods. SAMC's improvement in performance over GEM is more substantial in lower data regime, which is possibly due to that sample size is proportional to the memory size. When sample size is small model may easily overfit thus our extra samples can provide maximal benefits, while when sample size is already large enough to learn the model there is no need for extra data. Our model achieves best performance gain in handling few-shot CL which we believe is the most challenging while meaningful case in practice.

**Effectiveness in Handling Forgetting.** As shown in Figure 3, we demonstrate the evolution of first task's test accuracy throughout the entire training process. As can be seen, the proposed method SAMC (red curve) is almost on top of all the curves for the three datasets. Thanks to the memory efficiency of our algorithm and the generalization ability brought by the inpainting model, the proposed method can mitigate catastrophic forgetting to a higher level than other approaches, even with a very small memory buffer.

**Memory Usage Analysis.** As shown in Table 3, we analyze the disc space and training time required by our model on mini-ImageNet dataset and demonstrate that *our computational bottleneck is negligible compared*

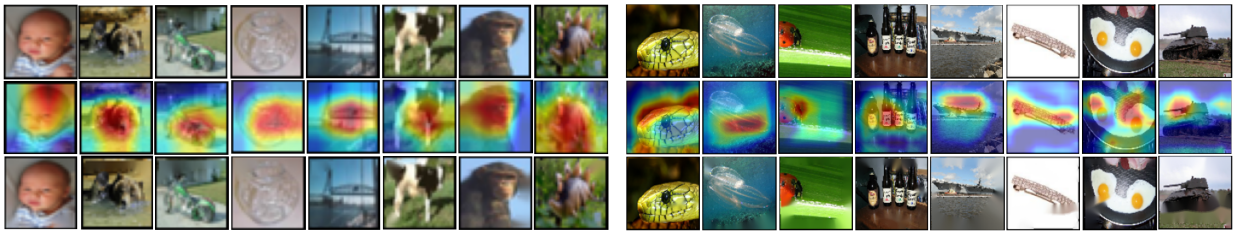

Figure 4: **Visualization of saliency map and inpainted images generated by SAMC. Left:** CIFAR-100. **Right:** mini-ImageNet. **First row:** Ground truth; **Second row:** Saliency map; **Third row:** Inpainted image.

Table 3: Memory & computational cost analyses. Our method can achieve 60% memory saving with only 7.5% extra computational cost on mini-ImageNet.

| | Disc Consupmtion (MB) | Running Time (sec) |
|---|---|---|
| GEM | 44 (model) + 413 (images) | 133 |
| SAMC | 44 + 413*$\frac{1}{3}$ (↓) + 0.5 (AE,↑) | 133 + 7 (↑) + 3 (↑) |
| ratio | ∼60% (↓) | ∼7.5% (↑) |

*to our memory saving*. The autoencoder (AE) we use is very light-weight and we apply early stopping for its finetuning. By adding quantitative analyses we found our SAMC can achieve **60%** memory saving with only **7.5%** extra computational cost. Specifically, on mini-ImgNet, SAMC requires 44 MB for ResNet18, 137 MB for images ($\mu$=0.6) and less than 1 MB for AE, while GEM baseline requires 44 MB for ResNet and 413 MB for real images. Meanwhile, by measuring the training time for the first two tasks in single epoch, SAMC takes 7s and 3s for computation induced by Grad-CAM and autoencoder, respectively, in addition to GEM's 130s training time, which leads to a marginal 7.5% (=(7+3)/130) computational cost. Notice that Grad-CAM itself does NOT introduce any extra parameter.

**Qualitative Analysis.** We visualize the saliency map and inpainted images by SAMC in Figure 4. Both saliency map and inpainted images are visually reasonable. Saliency map localization is more accurate on larger images such as mini-ImageNet. The inpainting model generates high-quality images close to the ground truth though there exist occasional and small obscure regions in some cases.

## 6 Conclusion

This paper proposes a new continual learning framework with external memory based on memory completion/separation theory in neuroscience. By utilizing saliency map, we extract the most informative regions in each image. We store the extracted images in sparse matrix format, achieving better memory efficiency. Inpainting method is used for recovering images from memory with better generalizability. We propose a bilevel optimization algorithm to train the entire framework and we provide theoretical guarantee of the algorithm. Finally, an extensive experimental analysis on commonly-used real-world datasets demonstrates the effectiveness and efficiency of the proposed model.


## Acknowledgement

This work was supported by the National Science Foundation (NSF) Grant No. 1755850, No. 1841520, No. 2007716, No. 2007976, No. 1942594, No. 1907805, a Jeffress Memorial Trust Award, Amazon Research Award, NVIDIA GPU Grant, and Design Knowledge Company (subcontract number: 10827.002.120.04).


## References


[1] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *arXiv preprint arXiv:1810.03292*, 2018.

[2] Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. *arXiv preprint arXiv:1903.08671*, 2019.

[3] Ali Ayub and Alan R Wagner. Eec: Learning to encode and regenerate images for continual learning. *arXiv preprint arXiv:2101.04904*, 2021.

[4] Guangji Bai and Liang Zhao. Saliency-regularized deep multi-task learning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 15–25, 2022.

[5] Nathan Bell and Michael Garland. Efficient sparse matrix-vector multiplication on cuda. Technical report, Citeseer, 2008.

[6] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[7] Marcelo Bertalmio, Andrea L Bertozzi, and Guillermo Sapiro. Navier-stokes, fluid dynamics, and image and video inpainting. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–I. IEEE, 2001.

[8] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong

learning with a-gem. *arXiv preprint arXiv:1812.00420*, 2018.

[9] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019.

[10] Sayna Ebrahimi, Suzanne Petryk, Akash Gokul, William Gan, Joseph E Gonzalez, Marcus Rohrbach, and Trevor Darrell. Remembering for the right reasons: Explanations reduce catastrophic forgetting. *arXiv preprint arXiv:2010.01528*, 2020.

[11] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017.

[12] Yuyang Gao, Tong Steven Sun, Guangji Bai, Siyi Gu, Sungsoo Ray Hong, and Zhao Liang. Res: A robust framework for guiding visual explanation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 432–442, 2022.

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[14] 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.

[15] Jeremias Knoblauch, Hisham Husain, and Tom Diethe. Optimal continual learning has perfect memory and is np-hard. In *International Conference on Machine Learning*, pages 5327–5337. PMLR, 2020.

[16] Shuang Li, Yilun Du, Gido M van de Ven, and Igor Mordatch. Energy-based models for continual learning. *arXiv preprint arXiv:2011.12216*, 2020.

[17] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

[18] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30:6467–6476, 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] German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019.

[21] Jialun Peng, Dong Liu, Songcen Xu, and Houqiang Li. Generating diverse structure for image inpainting with hierarchical vq-vae. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10775–10784, 2021.

[22] Yunchen Pu, Zhe Gan, Ricardo Henao, Xin Yuan, Chunyuan Li, Andrew Stevens, and Lawrence Carin. Variational autoencoder for deep learning of images, labels and captions. *Advances in neural information processing systems*, 29:2352–2360, 2016.

[23] Aravind Rajeswaran, Chelsea Finn, Sham Kakade, and Sergey Levine. Meta-learning with implicit gradients. 2019.

[24] 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.

[25] Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 1085–1097, 2019.

[26] 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.

[27] Anthony Robins. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, 7(2):123–146, 1995.

[28] Gobinda Saha and Kaushik Roy. Saliency guided experience packing for replay in continual learning. *arXiv preprint arXiv:2109.04954*, 2021.

[29] 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 international conference on computer vision*, pages 618–626, 2017.

[30] Alexandru Telea. An image inpainting technique based on the fast marching method. *Journal of graphics tools*, 9(1):23–34, 2004.

[31] Raymond A Yeh, Chen Chen, Teck Yian Lim, Alexander G Schwing, Mark Hasegawa-Johnson, and Minh N Do. Semantic image inpainting with deep generative models. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5485–5493, 2017.

[32] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[33] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.

[34] Wang Zhou, Shiyu Chang, Norma Sosa, Hendrik Hamann, and David Cox. Lifelong object detection. *arXiv preprint arXiv:2009.01129*, 2020.