

Schematic Memory Persistence and Transience for Efficient and Robust Continual Learning

Yuyang Gao^a, Giorgio A. Ascoli^b, Liang Zhao^{a,*}

^a*Department of Computer Science, Emory University, Atlanta, GA, United States*

^b*Center for Neural Informatics, Bioengineering Department, and Krasnow Institute for Advanced Study, George Mason University, Fairfax, VA, United States*

Abstract

Continual learning is considered a promising step towards next-generation Artificial Intelligence (AI), where deep neural networks (DNNs) make decisions by continuously learning a sequence of different tasks akin to human learning processes. It is still quite primitive, with existing works focusing primarily on avoiding (catastrophic) forgetting. However, since forgetting is inevitable given bounded memory and unbounded task loads, ‘how to reasonably forget’ is a problem continual learning must address in order to reduce the performance gap between AIs and humans, in terms of 1) memory efficiency, 2) generalizability, and 3) robustness when dealing with noisy data. To address this, we propose a novel ScheMATIC memory peRsistence and Transience (SMART) ¹ framework for continual learning with external memory that builds on recent advances in neuroscience. The efficiency and generalizability are enhanced by a novel long-term forgetting mechanism and schematic memory, using sparsity and ‘backward positive transfer’ constraints with theoretical guarantees on the error bound. Robust enhancement is achieved using a novel short-term forgetting mechanism inspired by background information-gated learning. Finally, an extensive experimental analysis on both benchmark and real-world datasets demonstrates the effectiveness and efficiency of our model.

Keywords: deep learning, deep neural networks, continual learning, lifelong learning, schematic memory, memory efficiency, robustness

*Corresponding author

Email addresses: yuyang.gao@emory.edu (Yuyang Gao), ascoli@gmu.edu (Giorgio A. Ascoli), liang.zhao@emory.edu (Liang Zhao)

¹Code available at: <https://drive.google.com/file/d/10fSbvKq488wq1HyNsWX5tbHyvmiqPPv9>.

1. Introduction

An important step toward next-generation Artificial Intelligence (AI) (i.e., artificial general intelligence) is a promising new domain known as continual learning, where deep neural networks (DNNs) make decisions by continuously learning a sequence of different tasks, similar to the way humans learn [1]. Compared with the extensive research on traditional AI for learning isolated tasks individually, continual learning is still in its very primitive stage [1]. At present, the primary goal is essentially to avoid (catastrophic) forgetting of previously learned tasks when an agent is learning new tasks. This important issue is receiving a great deal of attention, with researchers proposing a number of different strategies which can be categorized into three main approaches. The first focuses on simply restricting changes in the magnitudes of all the model parameters [2, 3, 4, 5]. However, given the limited amount of neural resources available, this approach necessarily sacrifices the flexibility needed to fit new tasks by protecting old tasks [1]. An alternative approach used by dynamic architecture-based methods is to allow the model to incrementally expand the network whenever new tasks arrive. Constantly increasing the number of neurons is highly inefficient, however, and very different to the way biological neural networks function [6, 7, 8]. The third approach, replay-based methods, avoid this problem by storing the historical training data with the help of external memory [9, 10, 11, 12].

Among these three options, the replay-based methods are arguably more effective in terms of performance and bio-inspiration [13, 14] as a way to alleviate the catastrophic forgetting challenge and are thus becoming the preferred approach for continual learning models [15, 12]. Unfortunately, because the memory is bounded while the number of tasks is unbounded, new and old tasks must compete for memory, and hence forgetting is inevitable. ‘How to reasonably forget’ is still a major question with significant challenges remaining, including: **1) Memory inefficiency.** The performance of the replay-based models depend heavily on the size of the available memory in the replay buffer which is used to retain as many of the previous samples as possible. While existing works typically store the entire sample into 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 major 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) Vulnerability to noise and corruption.** Noise and corruption are ubiquitous in real-world data and are especially likely to be present in continual learning problems. However, continual learning under these conditions has not yet been thoroughly explored. Without sufficient consideration of noise and data corruption, existing models are very vulnerable to their effects during the learning process, especially for DNNs as these have a high capacity to fit noisy labels [16, 17].

To address these challenges, this paper proposes a novel ScheMATIC memory peRsistence and Transience (SMART) framework for continual learning with external memory based on recent advances in neuroscience. Compared to the research on memory persistence (i.e., remembering), the mechanisms responsible for memory transience (i.e., forgetting) have historically been under-explored in neuroscience until very recently [18]. Current research on forgetting has shown that forgetting is not a ‘failure’, but rather a dedicated mechanism to facilitate mnemonic processing and generalizability. ‘Forgetting’ can be executed at various time scales and may be either active or passive, which 1) reduces memory consumption, 2) improves generalizability by discarding useless, too-specific details, and 3) builds resistance to noise and errors that cannot be consolidated. The specific contributions presented in this paper are as follows:

- **Developing a new framework for efficient and robust continual learning.** This work aims to extend the current continual learning scenario to include more realistic and challenging settings that include noisy, irrelevant features, and data corruption. Beyond merely “avoid forgetting”, our new model leverages “passive and active forgetting” as well as “long-term and short-term forgetting”, building on the latest advances in neural-theoretical science.
- **Proposing a novel schematic-memory-driven long-term forgetting mechanism.** To actively enforce reasonable long-term forgetting, we go beyond existing work based on “episodic memory” to “schematic memory” [19, 20] by identifying and storing generalizable knowledge across learning tasks with sparsity and ‘backward positive transfer’ constraints.

The theoretical guarantee on the error bound of such a mechanism has also been analyzed.

- **Achieving an effective short-term forgetting mechanism via novel neural correlation consolidation.** We construct a novel short-term forgetting mechanism that efficiently gates undesired information such as noise and corrupted data in real-time. To do this, a new computationally-efficient regularization is proposed that is inspired by background information-gated learning.
- **Conducting comprehensive experiments to validate the effectiveness and efficiency of our proposed model.** Extensive experiments on one benchmark dataset and five real-world datasets demonstrate that the proposed model outperforms other comparison methods. A robust learning simulation, ablation studies, and case studies further demonstrate how the proposed components contribute to the robustness of the model against label corruption while at the same time ensuring the efficient use of memory.

2. Related work

2.1. *Continual learning with experience replay*

The central problem of continual learning is to overcome the catastrophic forgetting problem of neural networks. Nowadays experience replay has been shown to be the most effective method for mitigating catastrophic forgetting [9, 15, 21, 22, 23, 24, 25, 26, 27, 12]. Specifically, replay-based methods alleviate the forgetting of deep neural networks by replaying stored samples from the previous history when learning new ones. There are two main directions on how to leverage the exemplars in the memory to mitigate forgetting. The first direction is first proposed by iCaRL [15], which uses the knowledge distillation technique to prevent forgetting. Several other works have followed this direction including the End-to-End Incremental Learning (EEIL) [21], the Unified classifier [22], and the Bias Correction (BiC) [23]. More recently, REMIND [24] proposed to store and replay mid-level representations as a more effective strategy to mitigate forgetting. However, the replay at mid-level representation requires the freeze of low-level layers and a non-trivial amount of additional data for model pre-training. The second direction is first proposed by Gradient Episodic Memory (GEM)

[9], which treats the gradient of the samples in the replay buffer as extra constraints to the model when acquiring new knowledge during the course of learning. The Averaged Gradient Episodic Memory (A-GEM) [25] model further extended GEM and made the constraint computationally more efficient. Besides, several other directions have also been explored, such as Variational Continual Learning [26] that combines Bayesian inference with replay, while the Meta-Experience Replay model [27] combines replay with meta-learning.

However, the above works are typically built upon the assumption that the number of tasks and task boundaries are accessible before the learning start and will be used as additional information to divide the storage resource allocation for each task. This assumption is generally not available for many real-world applications. To overcome this issue, recently the boundary-free continual learning has been proposed and deals with the situation where task boundary and i.i.d. (Independent and identically distributed) assumption are not available. Several works have focused on proposing different strategies for sample selections, including reservoir sampling [10, 11], and gradient-based sample selection [12]. Here our work is closely related to the boundary-free continual learning problem, with additional challenges of modeling robustness where the training dataset may contain a certain amount of corrupted samples.

2.2. Episodic and Schematic memory

Episodic memory is a neurocognitive system that enables human beings to remember past experiences [14]. Recently, episodic memory has been widely adopted in rehearsal based models to overcome catastrophic forgetting challenge in continual learning [9, 12]. However, the bottleneck of the rehearsal based models has been highly connected to the size of the memory (i.e. replay buffer) which is used to keep previous samples as much as possible. Storing the instance completely into the memory can be highly inefficient, especially when the size of the memory is limited [28]. In fact, we humans seldom memorize all the details about experiences, but rather we utilize the elements of similar experiences and efficiently integrate them together as schematic memory [19, 20]. Such a schema-like memory will help us better generalize and extract important information from a specific event, and efficiently remember only those that are critical and of importance to the downstream learning tasks. Therefore, this can be a great characteristic to have in conjunction with the episodic memory to fully utilize the limited memory.

2.3. Robust Learning with Noisy Labels

Learning from noisy labels has long been an essential study in machine learning and AI systems [29]. Noisy labels, which are referred to corrupted labels from the actual ground-truth labels, can inevitably degenerate the robustness of learned models, especially for deep neural networks as they have the high capacity to fit noisy labels [16, 17]. To handle this issue, existing works have been focused on estimating the noise transition matrix [30, 31], training the model on selected samples [32, 33, 34]. However, most of the existing works focus on the off-line learning problem and little has been studied on the robustness problem in the continual learning setting. This might be due to the fact that most existing robust learning algorithms rely heavily on a pre-acquired small clean dataset and use it to help the model deal with future corrupted samples based on the i.i.d assumption. On the other hand, the pre-acquisition of a clean dataset before learning starts is almost impossible in the continual learning setting. In addition, the data from the past may not be helpful for the model to detect the corruption in current data without i.i.d assumption. To the best of our knowledge, we are the first one to study the challenging problem of robust continual learning and to propose a model that can solve all the challenges efficiently and effectively.

3. Problem Formulation

In this paper, we focus on replay-based continual learning, which can be defined as an online supervised learning problem. The problem formulation and its associated challenges are as follows:

A stream of learning tasks arrives sequentially with the corresponding input-output pairs $\{X^{(0)}, Y^{(0)}\}, \dots, \{X^{(t)}, Y^{(t)}\}, \dots$, for time intervals $0, 1, \dots, t-1, t, \dots$. Each $X^{(t)} \in \mathbb{R}^{N_t, D}$, $Y^{(t)} \in \mathbb{R}^{N_t, 1}$, where N_t denotes the number of samples in task t and D is the number of input features. The data stream can be non-stationary, with no assumptions applied to the distributions such as the i.i.d. assumption.

Problem Formulation: Given such a stream of tasks, continual learning is commonly formulated as a process to train a learning agent $f : \mathbb{R}^{N_t, D} \rightarrow \mathbb{R}^{N_t, 1}$ parameterized by $\theta^{(t)}$ over each task $\{X^{(t)}, Y^{(t)}\}$ sequentially over time $t \in \mathbb{N}$, by minimizing the error in each task $\ell(f(X^{(t)}, \theta^{(t)}), Y^{(t)})$ for each task t . Moreover, such a learning agent typically has a memory with a fixed memory budget in order to avoid losing its competence in previously learned tasks, which is achieved by enforcing the *positive-backwards-transfer*

constraint: $\ell(f(X^{(i)}, \theta^{(t)}), Y^{(i)}) \leq \ell(f(X^{(i)}, \theta^{(t-1)}), Y^{(i)})$ over all the previous tasks $i \in [0, \dots, t-1]$.

This is a potentially very promising open problem that is currently attracting a great deal of attention from researchers, who are focusing on how to ensure the *positive-backwards-transfer* constraint with fixed memory size [9, 12]. However, other significant unsolved challenges remain, including : **Challenge 1: Memory inefficiency.** Existing works can only select whole data instances without the ability to identify and discard those input features that are useless for learning tasks. This significantly limits their memory efficiency, especially for high-dimensional, sparse data. Existing work neither ensures the original *positive-backwards-transfer* constraints nor knows the error bound; **Challenge 2: Weakness in improving generalizability.** Existing works focus on episodic memory without the human-like ability to abstract all the isolated memory into schematic memory [19, 20] and thus obtain continuing improvements in generalization power during continual learning; and **Challenge 3: Sensitivity to noise in continual learning** Existing works assume all data are clean without corruption, which is not realistic for real world data, making the existing models sensitive to noise.

4. SMART model

This section presents our proposed SMART model that addresses all the above-mentioned challenges and thus narrows the gap between human and machine continual learning. Section 4.1 introduces biological motivations, Sections 4.2 and 4.3 describe the mechanisms proposed to achieve long-term and short-term memory transience, and Section 4.4 describe the training algorithm for the new model.

4.1. Memory Persistence and Transience for Continual Learning

Existing works emphasize “how to remember”, but lack a model on how to actively and gracefully “forget” useless details (Challenges 1 and 2) and deal with noise (Challenge 3); human beings are much better at continual learning and suffer much less from these challenges [35]. The underlying mechanism through which humans achieve this so successfully has long been unclear, but recent research on “transience” (i.e., forgetting) and its interaction with “persistence” (i.e., remembering) had led to some interesting developments in neuroscientists’ understanding [18].

A great deal of evidence has been collected to indicate that transience happens at different time scales, with both long-term forgetting and short-term forgetting being indispensable for human life [36]. These perform different functions: long-term forgetting reduces memory consumption and encourages the consolidation by discarding unnecessary details, while short-term forgetting blocks out useless noise and spurious information. Hence, in our proposed model we aim to bridge the gap between human and machine intelligence by providing a new AI continual learning that incorporates the advantageous neuro-mechanisms in human brains, leveraging long-term forgetting to address Challenges 1 and 2 and short-term forgetting to handle Challenge 3.

To achieve long-term forgetting, we actively encourage *backward positive transfer* among the learning tasks, under schematic memory [19, 20]. Going beyond the episodic memory used in existing models, which stores historical tasks individually, schematic memory allows us to store only partial details $R_i(X^{(i)})$ of each task i by prioritizing the knowledge that is generalizable across tasks. Specifically, as shown in Equation (1), monotonic-increasing constraints over the sequential tasks’ performance are enforced on samples, with pruned input features specified by a regularization term $\Omega_e(\theta)$. This is described in more detail in Section 4.2.

To achieve short-term forgetting, a novel mechanism is developed to block spurious information such as noise and corrupted data in real-time. The mechanism developed to achieve this is based on background information-gated (BIG) learning theory [37, 38, 39]. A novel neural correlation consolidation regularization $\Omega_r^{(t)}(\theta)$ is proposed that facilitates efficient computation. Unlike $\Omega_e(\theta)$, this term is task-dependent and will be updated every time the model receives a new learning task (or for every batch if the learning problem is boundary-free). More details are provided in Section 4.3.

We can now integrate all the above terms and formulate the goal of the learning problem described in Section 3 as the following constrained optimization problem:

$$\begin{aligned} & \operatorname{argmin}_{\theta} \mathcal{L}(X^{(t)}; \theta) + \alpha \Omega_e(\theta) + \beta \Omega_r^{(t)}(\theta) \\ & \text{s.t. } \mathcal{L}(R_i(X^{(i)}); \theta) \leq \mathcal{L}(R_i(X^{(i)}); \theta^{(t-1)}) \end{aligned} \quad (1)$$

where $i = 0, \dots, t-1$. Here t is the index of the current example and i indexes the previous examples. n is the number of examples of the current observations. $\mathcal{L}(X^{(t)}; \theta)$ is the concise denotation of $\ell(f(X^{(t)}; \theta), Y^{(t)})$ where

$f(\cdot; \theta)$ is a model parameterized by θ and ℓ is the loss function. $R_i(X^{(t)})$ is a function that returns all the entries of $X^{(t)}$ except for the set of input features r_i , which have all-zero weights in the first hidden layer.

4.2. Long-term forgetting with schematic memory

This section focuses on two important issues: 1) how to design $\Omega_e(\theta)$ in order to actively “forget” unimportant input features in $X^{(i)}$ and only retain $R_i(X^{(i)})$, and an efficient way to enforce *positive backward transfer* constraints; and 2) theoretical analyses and guarantees for the error bound of the proposed efficient surrogate constraints.

We introduce a regularization term, denoted as $\Omega_e(\theta)$, that models and takes advantage of schematic memory [19, 20] to regularize the model by ensuring it only considers a subset of the input features, thus reducing the memory size requirement for storing the instance and also the computational power needed for future memory replay. Regularization techniques such as group lasso (e.g., $\ell_{2,1}$ -norm) [40, 41] can be readily used to enforce group sparsity of the weights connecting each input feature and all the first-layer hidden neurons.

Since the *positive backward transfer* constraints in Equation (1) are prohibitively hard to maintain efficiently, here we propose to transform them into left-hand-side of Equation (B.1), which incorporates schematic memory and theoretical guarantee on the transformation error ε stated in Theorem 1.

$$g_*(t, i) \cdot g(i, i, t-1)^T \geq 0 \quad \implies \quad \mathcal{L}(R_i(X^{(i)}); \theta) \leq \mathcal{L}(R_i(X^{(i)}); \theta^{(t-1)}) + \varepsilon \quad (2)$$

where ε is the approximation error with bound analyzed in Theorem 1. Denote $\tilde{\theta}^{(i)} \subseteq \theta^{(i)}$ as the parameter set excluding the weights of all-zero features r_i , so its size is $\tilde{J} = J - |r_i| \cdot c$ where c is the number of neurons in the first layer and J is the total number of parameters. Denote the gradient $g(k, i, s) = \partial \mathcal{L}(R_i(X^{(k)}); \tilde{\theta}^{(s)}) / \partial \tilde{\theta}$ and the j -th partial derivative is $g_j(k, i, s) = \partial \mathcal{L}(R_i(X^{(k)}); \tilde{\theta}^{(s)}) / \partial \theta_j$. $g_*(t, i) = \partial \mathcal{L}(R_i(X^{(t)}); \tilde{\theta}) / \partial \tilde{\theta}$. Task index $i = 0, \dots, t-1$.

Theorem 1. *The theoretical error bound of ε approaches*

$\sum_s \min_{k \leq J - |r_i|c} \max_{j \neq k} \left| \lambda - \frac{g_j(t, i, s)}{g_k(t, i, s)} \cdot \lambda \right| \cdot \|g(i, i, s)\|_2$ *when the gradient step λ is sufficiently small.*

Proof. Theorem 1 can be directly proved based on Lemma 1 and Theorem 2. \square

Based on the the above theorem, we can easily obtain several remarks:

Remark 1: If $\forall (j \leq J - |r_i| \cdot c) : |\lambda/g_j(t, i, s) - \lambda/g_k(t, i, s)| \rightarrow 0$, then the error $\varepsilon \rightarrow 0$, which means the original constraints can be precisely enforced.

Remark 2: The larger the number of useless input features, the larger $|r_i|$ will be, and hence the smaller ε will be.

Lemma 1. We have $\mathcal{L}(R_i(X^{(i)}); \theta) \approx \mathcal{L}(R_i(X^{(i)}); \theta^{(t-1)}) + \sum_{s=0}^S \sum_j \frac{\partial \ell(f(R_i(X^{(i)}); \tilde{\theta}^{(s)}), Y^{(i)})}{\partial \tilde{\theta}_j^{(s)}} \cdot \lambda$, which means the left-hand-side is infinitely approaching to the right-hand-side when $\lambda \rightarrow 0$.

Proof. Please refer to Appendix A for the detailed proof. \square

Theorem 2. We have the following Equation:

$$\sum_j \frac{\partial \ell(f(R_i(X^{(i)}); \tilde{\theta}^{(s)}), Y^{(i)})}{\partial \tilde{\theta}_j^{(s)}} \cdot \lambda \leq \min_k \max_{j \neq k} \left| \lambda - \frac{g_j(t, i, s)}{g_k(t, i, s)} \cdot \lambda \right| \cdot \|g(i, i, s)\|_2 \quad (3)$$

Proof. Please refer to Appendix B for the detailed proof. \square

4.3. Short-term Forgetting with Neuronal Correlation Consolidation

Human beings are good at learning in noisy environments by distinguishing and blocking spurious associations from real ones via short-term forgetting [18]. One of the neuroscience theories utilized to explain this phenomenon is called background information-gated (BIG) learning theory [37, 38, 39], where the previously learned knowledge and the real associations between the concepts allow only useful and accurate information through the gate for use in future learning. This ensures spurious co-occurrences of events and associations between concepts that are formed by random noise or mislabels will not be learned. To achieve such a learning selectivity in DNNs, it is first necessary to identify the nodes that encode important concepts learned in the past, as well as the strong associations that are encoded in the synaptic weights between the nodes.

It is nontrivial to embed BIG theory into DNNs. To do this, we first need to express this theory at the cell-level utilizing relevant BIG-based computational models such as [39]. Only then can we approximate the biological neuronal operations with artificial neurons in BNNs, including neuron importance and the correlation weights between neurons. This can be considered loosely analogous to the degree of similarity of the axonal connection pattern of biological neurons in BNNs. Suppose the matrix $\theta_{(l+1)} \in \mathbb{R}^{N_l \times N_{l+1}}$ represents

all the weights between neurons in layers l and $l + 1$ in the DNNs, where N_l and N_{l+1} represent the numbers of neurons, respectively:

$$A^{(l)} = \frac{1}{N_{(l+1)}^2} (h(\theta_{(l+1)}) \cdot h(\theta_{(l+1)})^\top) \odot (h(\theta_{(l+1)}) \cdot h(\theta_{(l+1)})^\top) \quad (4)$$

where $h(\cdot)$ is an indicator function that outputs 1 if the input is non-zero and 0 otherwise, and thus discretizes the weights in the DNNs to simulate the binary nature of the connections in BNNs. The layer-wise neuron correlation matrix $A^{(l)} \in \mathbb{R}^{N_l \times N_l}$ is a symmetric square matrix that models all the pairwise neuron correlations in layer l . Each entry $A_{i,j}^{(l)}$ models the correlation between neuron i and neuron j in terms of the similarity of their connectivity patterns. The higher the value, the stronger the correlation between the two.

Although the discretization in $h(\cdot)$ would make the computation in Equation (4) more biologically plausible, it makes the model non-differentiable and difficult to train and optimize. In addition, in practice, it is more common to have very small weights than true zero weights and it is therefore hard to define an appropriate threshold to establish the hard partition. Therefore, to mitigate these, we propose a differentiable approximation of the discrete version of $h(\cdot)$ as follows ²:

$$h(\theta_{(l+1)}) = |\tanh(\theta_{(l+1)})| \quad (5)$$

where $|\cdot|$ represents the element-wise absolute operator and $\tanh(\cdot)$ represents the element-wise hyperbolic tangent function. The values of $h(\theta_{(l+1)}) \in \mathbb{R}^{N_l \times N_{l+1}}$ will be positive and in the range of $[0, 1)$ with the value representing the relative connectivity strength of the synapse between neurons.

Notably, the neuron correlation matrix $A^{(l)}$ can also be interpreted as an adjacency matrix which reflects the hidden relationship between neurons within layer l , even though there is no actual connection between them. Thus, similar to the way degree centrality is used to estimate the importance of the node in network science, here we can treat $A^{(l)}$ as a weighted adjacency matrix and compute the weighted degree centrality of each neuron as its importance: $p_i^{(l)} = \sum A_{i,\cdot}^{(l)}$.

Now the neuron importance $p_i^{(l)}$ has been computed for each neuron, we

²Similar to the ReLU activation function, our formulation introduces a non-differentiable point at zero; we follow the conventional setting by using the sub-gradient for model optimization.

can naturally assume that the connection between two important neurons will also be important, thus the importance of $[\theta_{(l)}]_{i,j}$ can be estimated by: $P_{i,j}^{(l)} = p_i^{(l)} * p_j^{(l+1)}$.

Adopting the methods used in existing works on lifelong learning regularization design in [3, 4], we apply the estimated synaptic importance $P_{i,j}^{(l)}$ to regularize the changes of each parameter $[\theta_{(l)}]_{i,j}$ during the course of continual learning via a standard square loss terms as follows:

$$\Omega_r^{(t)}(\theta) = \sum_l \sum_i^{N^{(l)}} \sum_j^{N^{(l+1)}} \left(P_{i,j}^{(l)} ([\theta_{(l)}]_{i,j} - [\hat{\theta}_{(l)}]_{i,j})^2 \right) \quad (6)$$

where $[\hat{\theta}_{(l)}]_{i,j}$ is the stored weight between neurons i and j that learned from the previous task.

4.4. Algorithm

Algorithm 1 summarizes our proposed SMART learning algorithm. The efficient regularization $\Omega_e(\theta)$ will enable the schematic memory to only store partial details $R_t(X^{(t)})$ of task t , as shown in Line 7. The robust regularization $\Omega_r^{(t)}(\theta)$ regularizes the model parameters and protects the important weights that encode the knowledge learned from previous tasks from label corruption. When the memory is full, we adopt the GSS-Greedy Sample-Selection strategy presented in [12].

Algorithm 1: SMART algorithm

```

1: Input:  $n, M, \alpha, \beta, \epsilon \leftarrow 1e^{-4}$ 
2: Initialize:  $\theta, \hat{\theta}, R, \mathcal{M}, t \leftarrow 0$ 
3: repeat
4:   Receive:  $\{X^{(t)}, Y^{(t)}\}$ 
5:    $\{\hat{X}, \hat{Y}\} \leftarrow \{(X^{(t)}, Y^{(t)}), \mathcal{M}\}$ 
6:    $\theta^{(t)} \leftarrow \operatorname{argmin}_{\theta} \ell(f(\hat{X}; \theta), \hat{Y}) + \alpha \Omega_e(\theta) + \beta \Omega_r^{(t)}(\theta)$ 
7:    $\mathcal{M} \leftarrow \{R_t(X^{(t)}), Y^{(t)}\} \cup \mathcal{M}$ 
8:   if  $\operatorname{len}(\mathcal{M}) > M$  then
9:      $\mathcal{M} \leftarrow \text{Sample-Selection}(\mathcal{M}, M)$ 
10:  end if
11:   $\hat{\theta} \leftarrow \theta^{(t)}$ 
12:   $t \leftarrow t + 1$ 
13: until run out of observations

```

The feature sparsity enforced by $R_i(\cdot)$ and schematic memory regularization $\Omega_e(\theta)$ make the SMART algorithm more efficient in terms of both computation complexity and memory complexity. Specifically, at task t the degree of complexity is bounded by the number of useless features $|r_t|$, as shown in the following remarks:

Memory complexity: The memory complexity required to store the samples $\{R_t(X^{(t)}), Y^{(t)}\}$ into the memory is $O(N_t(D - |r_t|))$. The greater the $|r_t|$, the lower the memory complexity to store the same amount of samples.

Computational complexity: The computational complexity of Equation (B.1) is $O(J - |r_t| \cdot c)$. The greater the number of useless features $|r_t|$, the lower the computational complexity of the constraints.

5. Experiments

In this section, we studied the Disjoint MNIST as the benchmark dataset along with several Civil Unrest datasets for the real-world application of continual event forecasting. All the experiments were conducted on a 64-bit machine with an Intel(R) Xeon(R) W-2155 CPU 3.30GHz processor, 32GB memory, and an NVIDIA TITAN Xp GPU.

Disjoint MNIST Dataset: The MNIST Benchmark [42] image dataset divided into 5 tasks similar with the settings in [4, 12, 43]. Here each task is a binary classification problem between two different digits. 1k examples were used per task for training and the results on all test examples were reported.

Civil Unrest Datasets: Each dataset was obtained from one Latin American country. Five datasets were used respectively for the countries: Brazil, Colombia, Mexico, Paraguay, and Venezuela. The tweet texts from Twitter were adopted as the model inputs. The goal is to utilize one date input to predict whether there will be an event in the next day in different cities (i.e. tasks). The event forecasting results were validated against a well-established labeled event set, the Gold Standard Report [44]. More details about how the datasets are converted into the continual learning setting can be found in Appendix C.

Model settings: To ensure a fair comparison, all the experiments used the same feed-forward neural network of 2 hidden layers and 400 neurons. The Adam optimizer [45] was used with a learning rate of 0.0001 for all the models. The sparsity factor α introduced in the proposed method was set to 0.0005 by default and the robust regularization factor β was set via a grid search across the range 0.0001 to 0.01, depending on the corruption ratio. The training batch size was set to 50 and all the models were trained over 100 iterations per batch for all the datasets.

Comparison methods:

Online Clustering in the gradient space (GSS-Clust): A reply-based method with an online clustering method in the gradient space for the sample

Table 1: Average test accuracy on disjoint MNIST with different buffer sizes. Results are averaged over 10 different random seeds.

Buffer Size Method	100	200	300	400	500
GSS-Clust	75.53 \pm 1.60	80.95 \pm 0.98	85.86 \pm 2.18	88.38 \pm 1.31	89.46 \pm 1.33
FSS-Clust	70.72 \pm 2.67	75.13 \pm 1.05	80.48 \pm 2.24	86.19 \pm 2.59	86.12 \pm 0.62
GSS-IQP	78.03 \pm 1.70	86.93 \pm 1.56	88.08 \pm 0.89	90.27 \pm 0.33	91.88 \pm 0.57
GSS-Greedy	68.21 \pm 3.67	77.61 \pm 2.55	87.14 \pm 1.56	89.96 \pm 0.97	91.40 \pm 0.53
SMART	85.28 \pm 1.42	90.46 \pm 1.25	91.63 \pm 0.91	92.71 \pm 1.17	93.23 \pm 0.49

selection. The distance metric was set to the Euclidean distance and the doubling algorithm for incremental clustering described in [46] was adopted.

Online Clustering in the feature space (FSS-Clust): Similar to GSS-Clust, a reply-based method with an online clustering method in the feature space for the sample selection.

IQP Gradients (GSS-IQP): A reply-based method with the surrogate proposed in [12] to select samples that minimized the feasible region.

Gradient greedy selection (GSS-Greedy): A reply-based method with an efficient greedy selection variant proposed in [12].

5.1. Experiments on Disjoint MNIST Benchmark

Efficient Continual Learning with the Clean Dataset: As shown in Table 1, we compared the model performance on the Disjoint MNIST dataset across different buffer sizes. The results are averaged over 10 different random seeds and the best results are highlighted in boldface. Overall, our proposed SMART model outperformed all the baseline methods by a significant margin. Moreover, the gap between SMART and all the other baseline methods was bigger when the buffer size was extremely limited. For instance, SMART outperformed the baseline methods by 9% - 25% when the buffer size was 100 and by 5% - 20% when the buffer size was 200. This is because the proposed feature selection supported the efficient usage of memory for applications where the memory is extremely limited. Notably, the SMART model was able to achieve an equivalent learning capacity to those achieved by the baseline models with 200 additional samples in the buffer space.

To illustrate how the proposed feature selection enhanced the efficient usage of memory in the SMART model, we compared the samples stored in the buffer during the course of continual learning for all models, as shown in Figure 1. Although the buffer size was only fixed to store a maximum of 300 full samples, the SMART model easily exceeded the cap, storing 500 effective

Table 2: Average test accuracy on disjoint MNIST under different corruption ratios. The buffer size of all the models was fixed to store a maximum of 300 full samples. Results are averaged over 10 different random seeds.

Ratio Model	Clean	10%	20%	30%	40%	50%
GSS-Clust	85.86 \pm 2.18	68.28 \pm 1.98	65.21 \pm 2.92	60.42 \pm 1.44	59.23 \pm 0.87	56.11 \pm 0.89
FSS-Clust	80.48 \pm 2.24	69.68 \pm 1.64	66.41 \pm 2.30	62.31 \pm 1.83	61.58 \pm 1.12	58.53 \pm 1.09
GSS-IQP	88.08 \pm 0.89	69.45 \pm 0.53	63.95 \pm 0.87	59.28 \pm 0.97	59.55 \pm 0.84	56.76 \pm 1.19
GSS-Greedy	87.14 \pm 1.56	75.75 \pm 0.93	71.88 \pm 1.87	64.92 \pm 1.95	62.99 \pm 1.02	59.09 \pm 1.34
SMART	93.23 \pm 0.49	85.29 \pm 1.58	81.35 \pm 1.44	73.91 \pm 1.65	70.08 \pm 2.16	66.34 \pm 2.63

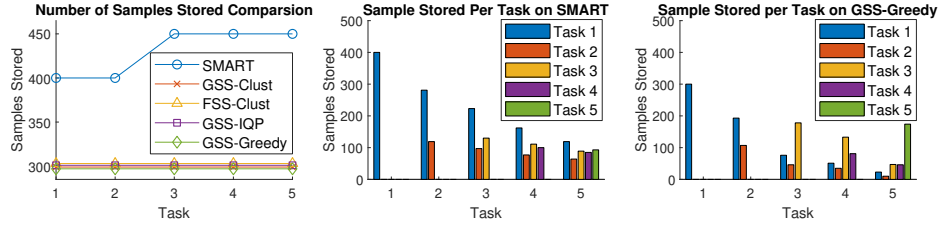


Figure 1: A comparison of the total number of samples stored and the samples stored for each task in the buffer during the course of continual learning between SMART and the GSS-Greedy model. The buffer size was fixed to store a maximum of 300 full samples.

samples during the learning process. This means that with the same amount of storage space, the SMART model can preserve more samples and enjoy a better performance even if the storage space is extremely limited.

Robust Continual Learning with Label Corruption: To further examine the robustness of our proposed SMART framework, we conducted additional experiments on the Disjoint MNIST dataset using different ratios of corrupted labels in the training samples. As shown in Table 2, the SMART model once again appeared to be the most robust model for different levels of corruption. For instance, when 10% of the training samples were corrupted, the performance of the SMART model only dropped by 8.5% compared to the clean training data. This contrasts markedly with the performance of all the other baseline methods under the same conditions, all of which dropped significantly, ranging from 13% - 21% worse than their clean training counterparts. As a consequence, SMART outperformed the baseline methods even more when corruption was present, outperforming the baseline methods by as much as 15% - 25% under 30% corruption. Other corruption ratios followed the same trend. This observation further demonstrates the superior robustness properties of our proposed robust learning regularization.

Ablation study of Neuronal Correlation Consolidation (NCC)

Table 3: Average test accuracy for the robust continual learning experiments on Split MNIST datasets under different corruption ratios. The results are averaged over 10 random seed runs.

Corruption Ratio Regularization	Clean	10%	30%	50%	70%
Vanilla	59.21±2.04	58.19±1.26	58.99±0.54	58.69±0.30	57.27±0.56
EWC	58.85±2.59	58.62±1.68	58.83±0.65	58.59±0.34	57.53±0.74
SI	65.76±3.09	64.07±4.53	63.26±6.37	62.12±6.81	57.73±8.91
L2	66.00±3.73	62.20±1.98	59.13±0.56	58.77±0.22	57.57±0.77
MAS	68.57±6.85	63.81±5.95	61.24±6.00	59.31±5.91	56.99±3.76
NCC	82.24±0.91	76.64±1.68	70.45±2.29	65.83±2.16	64.21±2.94

Regularization: We also conducted an ablation study of the proposed NCC regularization and compared it with existing continual learning regularization methods, including L2, EWC [3, 47], SI [4], MAS [5]. The hyperparameter is tuned by a grid search, and the results with the best setting are reported. For a fair comparison, all comparison methods use the same neural network architecture, which is a multi-layered perceptron with two hidden layers of 400 nodes each, followed by a softmax output layer. Both hidden layers use ReLU for the activation function. The loss function is a standard cross-entropy for classification. All models are trained for 4 epochs per task with mini-batch size 128 using the Adam optimizer.

As shown in Table 3, the overall performance of the proposed NCC regularization outperforms all baseline regularization methods by a significant margin on both clean data and data with various corruption ratios. Specifically, the model with NCC regularization outperforms the model without any regularization (i.e. Vanilla) by 12% - 32% and outperforms the models with other existing regularization by up to 30%. Interestingly, the baseline model with SI regularization also achieves some level of robustness, probability because the neuroscience inspiration behind the SI algorithm also enhanced the robustness property of the model. However, overall it is not as effective as the proposed NCC model. Besides, the performance of the SI model is much unstable with the relatively highest standard deviations across almost all corruption ratios. A similar issue can be also observed in MAS regularization.

Beside just comparing the performance under the same corruption ratio, we are also interested in how the performance evolved as the corruption ratio gradually increases. As we can see in Table 3, all the baseline models tend to converge to around 57% accuracy as the corruption ratio continues to increase. However, the model with NCC regularization tends to converge to around 64%

accuracy in the same scenario, leaving an around 12% performance gap. This observation demonstrates the superior robustness property of the proposed NCC regularization.

5.2. Experiments on Continual Event Forecasting

Table 4: Average test accuracy for the robust continual learning experiments on civil unrest datasets under different label corruption ratios. The buffer size of all the models were fixed to store a maximum of 300 full samples. Results are averaged over 10 different random seeds.

Dataset	Ratio Model	Clean	10%	20%	30%	40%
Brazil	GSS-Clust	78.57 \pm 1.43	71.66 \pm 0.72	65.68 \pm 2.37	61.37 \pm 2.01	53.51 \pm 2.81
	FSS-Clust	79.59 \pm 1.84	72.46 \pm 0.80	65.31 \pm 2.98	60.55 \pm 3.16	53.85 \pm 1.47
	GSS-IQP	78.02 \pm 0.98	71.75 \pm 0.86	65.87 \pm 0.84	62.83 \pm 1.24	54.15 \pm 0.97
	GSS-Greedy	77.59 \pm 2.37	73.70 \pm 0.42	67.84 \pm 0.85	63.27 \pm 1.98	55.94 \pm 3.72
	SMART	83.68 \pm 2.10	76.78 \pm 1.14	71.38 \pm 0.58	66.59 \pm 0.37	57.36 \pm 2.18
Colombia	GSS-Clust	71.65 \pm 0.88	66.39 \pm 1.15	64.52 \pm 4.63	58.53 \pm 2.95	51.39 \pm 1.31
	FSS-Clust	76.27 \pm 2.10	70.40 \pm 0.95	64.97 \pm 1.65	60.53 \pm 1.70	54.15 \pm 3.25
	GSS-IQP	77.23 \pm 1.02	72.05 \pm 1.00	64.78 \pm 2.42	61.37 \pm 2.07	54.29 \pm 1.79
	GSS-Greedy	75.71 \pm 1.68	71.17 \pm 1.21	66.79 \pm 1.08	61.38 \pm 2.96	52.85 \pm 0.81
	SMART	81.43 \pm 1.12	74.76 \pm 1.26	70.92 \pm 2.71	65.12 \pm 3.21	57.93 \pm 1.16
Mexico	GSS-Clust	64.51 \pm 0.85	60.94 \pm 0.90	57.42 \pm 0.98	56.19 \pm 0.87	53.78 \pm 0.74
	FSS-Clust	64.94 \pm 0.98	61.52 \pm 0.96	58.42 \pm 0.92	56.46 \pm 0.62	54.27 \pm 0.71
	GSS-IQP	66.87 \pm 0.56	63.32 \pm 0.64	59.03 \pm 0.78	56.55 \pm 0.41	53.73 \pm 0.91
	GSS-Greedy	65.45 \pm 0.16	62.12 \pm 0.79	59.26 \pm 0.52	56.73 \pm 0.70	53.57 \pm 0.53
	SMART	70.98 \pm 1.54	67.61 \pm 0.45	63.48 \pm 0.46	59.34 \pm 0.43	56.58 \pm 0.81
Paraguay	GSS-Clust	66.81 \pm 2.18	64.36 \pm 1.56	60.16 \pm 1.37	57.84 \pm 1.95	54.27 \pm 0.73
	FSS-Clust	67.01 \pm 0.95	63.84 \pm 1.36	59.53 \pm 1.24	57.64 \pm 1.58	53.75 \pm 2.07
	GSS-IQP	64.17 \pm 1.94	62.93 \pm 0.91	58.47 \pm 1.47	56.35 \pm 1.73	52.25 \pm 1.45
	GSS-Greedy	65.77 \pm 1.47	63.68 \pm 0.28	59.61 \pm 1.64	57.70 \pm 1.57	54.40 \pm 1.60
	SMART	69.14 \pm 1.83	66.03 \pm 0.66	62.11 \pm 2.63	58.91 \pm 0.74	55.12 \pm 1.06
Venezuela	GSS-Clust	66.00 \pm 1.45	63.61 \pm 1.54	59.87 \pm 0.93	56.14 \pm 0.64	52.56 \pm 0.55
	FSS-Clust	69.28 \pm 1.19	64.73 \pm 0.76	60.57 \pm 0.43	57.47 \pm 0.74	51.73 \pm 0.57
	GSS-IQP	69.67 \pm 0.79	65.46 \pm 0.39	60.78 \pm 0.88	57.18 \pm 0.48	53.54 \pm 0.37
	GSS-Greedy	69.70 \pm 0.75	66.52 \pm 0.31	61.91 \pm 0.71	57.38 \pm 1.27	52.73 \pm 0.63
	SMART	73.82 \pm 0.18	69.23 \pm 0.68	64.78 \pm 0.63	60.73 \pm 1.07	55.95 \pm 0.75

Efficient Continual Learning with the Clean Datasets: The column ‘Clean’ in Table 4 summaries the model performance on the five civil unrest datasets. The buffer size of all the models tested here was fixed to store a maximum of 300 full samples. The results were averaged over 10 different random seeds and the best results are highlighted in boldface. Overall, our proposed SMART model achieved the best results across the datasets for all five countries, and outperforming all the baseline methods by a significant margin. On average, SMART outperformed baseline methods by 5% - 8% across the five datasets. This is because the proposed schematic memory enhanced both the efficient usage of the memory buffer and facilitated a more semantically meaningful feature selection. To further illustrate this effect, we visualized the total number of samples stored in the buffer of the SMART model during the course of continual learning over the five Civil Unrest Datasets, as shown in Figure 2. Again the buffer size is fixed to store

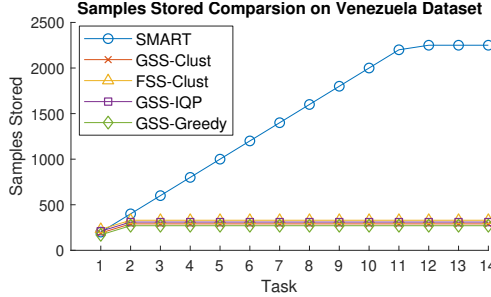


Figure 2: A comparison of the total number of samples stored in the buffer during learning on the Venezuela Dataset. The buffer size was fixed to store at a maximum of 300 full samples. The SMART model was able to greatly exceed the cap as only a very few features were stored for each sample. Other counties followed the same trend.

at a maximum of 300 full samples. This time SMART is able to massively boost the number of effective samples stored, ranging from 600% to 1000%. This is due to the fact that only a very few features are actually important for the prediction task and the schematic memory introduced that helps the model to gradually forget most of the non-useful features over time gradually builds up the most important features for the learning task. We conducted a case study in Section 5.5 and showed how the top selected features that are highly relevant to civil unrest events evolved on Brazil dataset.

Robust Continual Learning with Label Corruption: Table 4 shows the robust lifelong learning experiments on five civil unrest datasets under different label corruption ratios. Once again, the SMART model was the most robust model against different levels of corruption for all five datasets. This observation confirms the superior robustness property of the proposed model across various datasets and application domains.

5.3. Comparison with Task-Aware Methods

In this section, we compare the proposed method with State-of-the-art task-aware methods, which leverage the task boundary to decide and manage the memory allocation for samples.

Comparison methods:

GEM [9]: stores a fixed amount of random examples per task and uses them to provide constraints when learning new examples.

iCaRL [15]: follows an incremental classification setting. It also stores a fixed number of examples per class but uses them to rehearse the network

when learning new information.

Table 5: Performance comparison with state-of-the-art task-aware replay methods on MNIST dataset and civil unrest datasets. The average test accuracy are averaged over 10 different random seeds. The buffer size of all the models were fixed to store a maximum of 300 full samples.

Model \ Dataset	iCaRL	GEM	SMART
Disjoint MNIST	83.27 \pm 2.93	88.93 \pm 1.07	91.63 \pm 0.91
Civil-Brazil	73.76 \pm 3.45	78.58 \pm 0.44	83.68 \pm 2.10
Civil-Colombia	73.60 \pm 1.72	75.29 \pm 1.04	81.43 \pm 1.12
Civil-Mexico	63.11 \pm 0.89	64.94 \pm 1.11	70.98 \pm 1.54
Civil-Paraguay	63.21 \pm 0.58	64.96 \pm 1.57	69.14 \pm 1.83
Civil-Venezuela	64.61 \pm 2.19	67.61 \pm 0.49	73.82 \pm 0.18

Naturally, like other sample selection based methods, our method ignores those tasks information which can place us at a disadvantage. Despite this disadvantage, our proposed model was able to get a competitive or even better performance across different application domains. As shown in Table 5, the proposed model outperformed the baselines by 3% - 10% on MINST dataset and by 6% - 13% on Civil Unrest datasets, which further demonstrated the superior advantages of the efficient sample storage as well as the sample selection algorithm in the SMART model.

5.4. Parameter Sensitivity Analysis

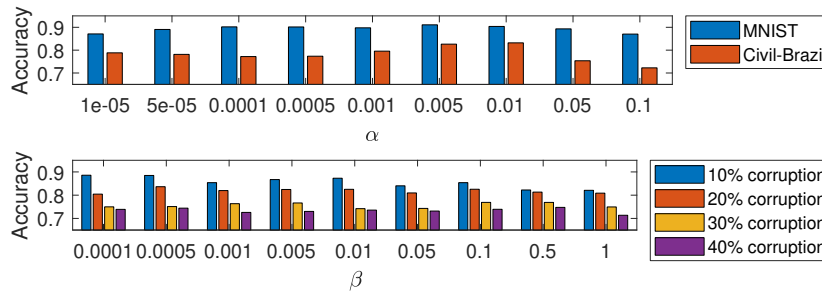


Figure 3: The sensitivity analysis of two regularization factors α and β for the long-term and short-term forgetting regularization. The buffer size was fixed to store a maximum of 300 full samples.

There are two hyper-parameters in the proposed SMART model, where α controls the schematic memory regularization and β controls the proposed neuronal correlation consolidation regularization. Figure 3 shows the effect on the overall accuracy of the model when varying α and β respectively. For α the results for Brazil within civil unrest datasets and MNIST dataset are shown here. As shown in the top bar chart in Figure 3, by varying α across the range from 0.00001 to 0.1, the performance of the MNIST dataset is stable, with the fluctuation ranges less than 4%. For the civil unrest dataset, the fluctuation range is 10%. The best performance is obtained when $\alpha = 0.005$. We can also see a clear trend where the accuracy decrease when α is too large or too small. The bottom bar chart illustrates the performance of the model versus β on the MNIST dataset with various corruption ratios. The fluctuation ranges around 4-6% on accuracy. It is also worth noting that in general when the corruption ratio increases, the β to get the best performance also increases accordingly. Specifically, the β that achieve the best performance are 0.0001, 0.0005, 0.1, 0.5 for corruption ratio 10%, 20%, 30%, and 40%, respectively. This demonstrated the effectiveness of the proposed robust regularization, as the model needs to rely more on the NCC regularization to be more robust against high corruption in the dataset.

5.5. Case Study of Features (Keywords) Selected by SMART for the Brazil Civil Unrest Dataset

Table 6: Top 15 features (keywords) selected by SMART for the Brazil Civil Unrest datasets over time (all keywords have been translated to English using Google Translate). The keywords in boldface are commonly selected during the course of learning and have high correspondences with civil unrest events.

T1	T2	T3	T4	T5	T6	T7	T8
development	failures	failures	failures	workplace accident	essential	essential	essential
prison	new law	essential	essential	essential	accident at work	accident at work	accident at work
damage	prison	prepare march	workplace accident	mother earth	private property	private property	private property
prepare march	private property	purse	private property	mobilization	failures	failures	nation country
essential	damage	human chain	prison	impose	new Law	new Law	negotiations
deforestation	assistants	private property	reforestation	private property	mother earth	nation country	coup
private property	demonstration	prison	nation country	new Law	reforestation	safety	new Law
sanctions	essential	investors	purse	dictatorship	nation country	reforestation	criminal act
police operation	prepare march	workplace accident	boo	demand	alliance	war crime	property rights
solidarity	war crime	dictatorship	prepare march	reforestation	energy production	property rights	failures
prohibition	human chain	genocide	human chain	finance	prepare march	organized	military action
opportunities	corrupt	alliance	demand	security	war crime	property rights	energy production
atrocities	property directors	new Law	trials	human chain	incinerate	criminalize	alternatives
pollution	left	nation country	alliance	alliance	victims	unemployment	conflict
environmentalists	coup	war crime	rebel	nation country	necessity	necessity	war crime

Table 6 shows a case study of the top 15 features (keywords) selected by

SMART for the Brazil Civil Unrest datasets over time. all keywords have been translated from Portuguese to English using Google Translate. The keywords in boldface are commonly selected during the course of learning and have high correspondences with civil unrest events. We can see a clear trend from left to right that during the course of continual learning, the SMART model was able to capture more relevant keywords that can be best used to predict civil unrest events. Thus with the help of schematic memory, the SMART model was able to gradually forget most of the non-useful features over time and gradually build up the most important features for the learning task.

6. Conclusion

This paper proposes a novel ScheMATIC memory peRsistence and Transience (SMART) framework for continual learning with external memory based on recent advances in neuroscience. The new framework’s efficiency and generalizability are enhanced by a novel long-term forgetting mechanism and schematic memory, using sparsity and ‘backward positive transfer’ constraints with theoretical guarantee on the error bound. Furthermore, a robust enhancement is introduced that incorporates a novel short-term forgetting mechanism inspired by background information-gated (BIG) learning theory. Finally, an extensive experimental analysis on both benchmark and real-world datasets demonstrates the effectiveness and efficiency of the proposed models.

7. Acknowledgement

This work was supported by the NIH Grant No. R01NS39600, the 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] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, S. Wermter, Continual lifelong learning with neural networks: A review, *Neural Networks* (2019).
- [2] M. K. Benna, S. Fusi, Computational principles of synaptic memory consolidation, *Nature neuroscience* 19 (12) (2016) 1697.

- [3] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, et al., Overcoming catastrophic forgetting in neural networks, *Proceedings of the national academy of sciences* 114 (13) (2017) 3521–3526.
- [4] F. Zenke, B. Poole, S. Ganguli, Continual learning through synaptic intelligence, in: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, JMLR. org, 2017, pp. 3987–3995.
- [5] R. Aljundi, F. Babiloni, M. Elhoseiny, M. Rohrbach, T. Tuytelaars, Memory aware synapses: Learning what (not) to forget, in: *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 139–154.
- [6] R. M. French, Dynamically constraining connectionist networks to produce distributed, orthogonal representations to reduce catastrophic interference, *network* 1111 (1994) 00001.
- [7] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, R. Hadsell, Progressive neural networks, *arXiv preprint arXiv:1606.04671* (2016).
- [8] A. Mallya, S. Lazebnik, Packnet: Adding multiple tasks to a single network by iterative pruning, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7765–7773.
- [9] D. Lopez-Paz, M. Ranzato, Gradient episodic memory for continual learning, in: *Advances in Neural Information Processing Systems*, 2017, pp. 6467–6476.
- [10] A. Chaudhry, M. Rohrbach, M. Elhoseiny, T. Ajanthan, P. K. Dokania, P. H. Torr, M. Ranzato, Continual learning with tiny episodic memories, *arXiv preprint arXiv:1902.10486* (2019).
- [11] D. Isele, A. Cosgun, Selective experience replay for lifelong learning, in: *Thirty-second AAAI conference on artificial intelligence*, 2018.
- [12] R. Aljundi, M. Lin, B. Goujaud, Y. Bengio, Gradient based sample selection for online continual learning, in: *Advances in Neural Information Processing Systems*, 2019, pp. 11816–11825.

- [13] A. Robins, Catastrophic forgetting, rehearsal and pseudorehearsal, *Connection Science* 7 (2) (1995) 123–146.
- [14] E. Tulving, Episodic memory: From mind to brain, *Annual review of psychology* 53 (1) (2002) 1–25.
- [15] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, C. H. Lampert, icarl: Incremental classifier and representation learning, in: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 2001–2010.
- [16] D. Arpit, S. Jastrzębski, N. Ballas, D. Krueger, E. Bengio, M. S. Kanwal, T. Maharaj, A. Fischer, A. Courville, Y. Bengio, et al., A closer look at memorization in deep networks, in: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, JMLR. org, 2017, pp. 233–242.
- [17] C. Zhang, S. Bengio, M. Hardt, B. Recht, O. Vinyals, Understanding deep learning requires rethinking generalization, *arXiv preprint arXiv:1611.03530* (2016).
- [18] B. A. Richards, P. W. Frankland, The persistence and transience of memory, *Neuron* 94 (6) (2017) 1071–1084.
- [19] K. Nelson, Remembering: A functional developmental perspective, in: *Memory: interdisciplinary approaches*, Springer, 1989, pp. 127–150.
- [20] M. Sprenger, *Learning and memory: The brain in action*, ASCD, 1999.
- [21] F. M. Castro, M. J. Marín-Jiménez, N. Guil, C. Schmid, K. Alahari, End-to-end incremental learning, in: *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 233–248.
- [22] S. Hou, X. Pan, C. C. Loy, Z. Wang, D. Lin, Learning a unified classifier incrementally via rebalancing, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 831–839.
- [23] Y. Wu, Y. Chen, L. Wang, Y. Ye, Z. Liu, Y. Guo, Y. Fu, Large scale incremental learning, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 374–382.

- [24] T. L. Hayes, K. Kafle, R. Shrestha, M. Acharya, C. Kanan, Remind your neural network to prevent catastrophic forgetting, in: European Conference on Computer Vision, Springer, 2020, pp. 466–483.
- [25] A. Chaudhry, M. Ranzato, M. Rohrbach, M. Elhoseiny, Efficient lifelong learning with a-gem, arXiv preprint arXiv:1812.00420 (2018).
- [26] C. V. Nguyen, Y. Li, T. D. Bui, R. E. Turner, Variational continual learning, arXiv preprint arXiv:1710.10628 (2017).
- [27] M. Riemer, I. Cases, R. Ajemian, M. Liu, I. Rish, Y. Tu, G. Tesauero, Learning to learn without forgetting by maximizing transfer and minimizing interference, arXiv preprint arXiv:1810.11910 (2018).
- [28] R. S. Gardner, M. R. Uttaro, S. E. Fleming, D. F. Suarez, G. A. Ascoli, T. C. Dumas, A secondary working memory challenge preserves primary place strategies despite overtraining, *Learning & Memory* 20 (11) (2013) 648–656.
- [29] D. Angluin, P. Laird, Learning from noisy examples, *Machine Learning* 2 (4) (1988) 343–370.
- [30] J. Goldberger, E. Ben-Reuven, Training deep neural-networks using a noise adaptation layer (2016).
- [31] G. Patrini, A. Rozza, A. Krishna Menon, R. Nock, L. Qu, Making deep neural networks robust to label noise: A loss correction approach, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1944–1952.
- [32] L. Jiang, Z. Zhou, T. Leung, L.-J. Li, L. Fei-Fei, Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels, arXiv preprint arXiv:1712.05055 (2017).
- [33] E. Malach, S. Shalev-Shwartz, Decoupling" when to update" from" how to update", in: Advances in Neural Information Processing Systems, 2017, pp. 960–970.
- [34] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang, M. Sugiyama, Co-teaching: Robust training of deep neural networks with extremely noisy labels, in: Advances in neural information processing systems, 2018, pp. 8527–8537.

- [35] R. S. Gardner, M. Mainetti, G. A. Ascoli, Older adults report moderately more detailed autobiographical memories, *Frontiers in Psychology* 6 (2015) 631.
- [36] R. S. Gardner, G. A. Ascoli, The natural frequency of human prospective memory increases with age., *Psychology and aging* 30 (2) (2015) 209.
- [37] J. Larkin, J. McDermott, D. P. Simon, H. A. Simon, Expert and novice performance in solving physics problems, *Science* 208 (4450) (1980) 1335–1342.
- [38] W. Kintsch, The role of knowledge in discourse comprehension: A construction-integration model., *Psychological review* 95 (2) (1988) 163.
- [39] M. Mainetti, G. A. Ascoli, A neural mechanism for background information-gated learning based on axonal-dendritic overlaps, *PLoS computational biology* 11 (3) (2015).
- [40] M. Yuan, Y. Lin, Model selection and estimation in regression with grouped variables, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 68 (1) (2006) 49–67.
- [41] J. M. Alvarez, M. Salzman, Learning the number of neurons in deep networks, in: *Advances in Neural Information Processing Systems*, 2016, pp. 2270–2278.
- [42] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, *Proceedings of the IEEE* 86 (11) (1998) 2278–2324.
- [43] Y.-C. Hsu, Y.-C. Liu, A. Ramasamy, Z. Kira, Re-evaluating continual learning scenarios: A categorization and case for strong baselines, in: *NeurIPS Continual learning Workshop*, 2018.
URL <https://arxiv.org/abs/1810.12488>
- [44] GSR Dataset, <https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/EN8FUW>. accessed Sep 2018.
URL <https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/EN8FUW>

- [45] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).
- [46] M. Charikar, C. Chekuri, T. Feder, R. Motwani, Incremental clustering and dynamic information retrieval, SIAM Journal on Computing 33 (6) (2004) 1417–1440.
- [47] J. Schwarz, J. Luketina, W. M. Czarnecki, A. Grabska-Barwinska, Y. W. Teh, R. Pascanu, R. Hadsell, Progress & compress: A scalable framework for continual learning, arXiv preprint arXiv:1805.06370 (2018).
- [48] B. O’Connor, R. Balasubramanyan, B. R. Routledge, N. A. Smith, From tweets to polls: Linking text sentiment to public opinion time series, ICWSM 11 (122-129) (2010) 1–2.

Appendix A. Lemma 1’s Proof

Lemma 1. $\mathcal{L}(R_i(X^{(i)}); \theta) \approx \mathcal{L}(R_i(X^{(i)}); \theta^{(t-1)}) + \sum_{s=0}^S \sum_j \frac{\partial \ell(f(R_i(X^{(i)}); \tilde{\theta}^{(s)}), Y^{(i)})}{\partial \tilde{\theta}_j^{(s)}} \cdot \lambda$,
the LHS is infinitely approaching to the RHS when $\lambda \rightarrow 0$.

Proof. Define $\theta^{(t,0)} \equiv \theta^{(t-1)}$ and $\theta \equiv \theta^{(t,S)}$, we have:

$$\ell(f(R_i(X^{(i)}); \theta), Y^{(i)}) \approx \ell(f(R_i(X^{(i)}); \theta^{(t,S-1)}), Y^{(i)}) + \sum_j \frac{\partial \ell(f(R_i(X^{(i)}); \theta^{(S-1)}), Y^{(i)})}{\partial \theta_j^{(S-1)}} \cdot \lambda \quad (\text{A.1})$$

$$\approx \ell(f(R_i(X^{(i)}); \theta^{(t,S-2)}), Y^{(i)}) + \sum_{s=S-2}^{S-1} \sum_j \frac{\partial \ell(f(R_i(X^{(i)}); \theta^{(s)}), Y^{(i)})}{\partial \theta_j^{(s)}} \cdot \lambda \quad (\text{A.2})$$

$$\dots \quad (\text{A.3})$$

$$\approx \ell(f(R_i(X^{(i)}); \theta^{(t,1)}), Y^{(i)}) + \sum_{s=1}^S \sum_j \frac{\partial \ell(f(R_i(X^{(i)}); \theta^{(s)}), Y^{(i)})}{\partial \theta_j^{(s)}} \cdot \lambda \quad (\text{A.4})$$

$$\approx \ell(f(R_i(X^{(i)}); \theta^{(t,0)}), Y^{(i)}) + \sum_{s=0}^S \sum_j \frac{\partial \ell(f(R_i(X^{(i)}); \theta^{(s)}), Y^{(i)})}{\partial \theta_j^{(s)}} \cdot \lambda \quad (\text{A.5})$$

$$= \ell(f(R_i(X^{(i)}); \theta^{(t-1)}), Y^{(i)}) + \sum_{s=0}^S \sum_j \frac{\partial \ell(f(R_i(X^{(i)}); \theta^{(s)}), Y^{(i)})}{\partial \theta_j^{(s)}} \cdot \lambda \quad (\text{A.6})$$

$$= \ell(f(R_i(X^{(i)}); \theta^{(t-1)}), Y^{(i)}) + \sum_{s=0}^S \sum_j \frac{\partial \ell(f(R_i(X^{(i)}); \tilde{\theta}^{(s)}), Y^{(i)})}{\partial \tilde{\theta}_j^{(s)}} \cdot \lambda \quad (\text{A.7})$$

and we have

$$\ell(f(R_i(X^{(i)}); \theta), Y^{(i)}) = \lim_{\lambda \rightarrow 0} \left(\ell(f(R_i(X^{(i)}); \theta^{(t-1)}), Y^{(i)}) + \sum_{s=0}^S \sum_j \frac{\partial \ell(f(R_i(X^{(i)}); \tilde{\theta}^{(s)}), Y^{(i)})}{\partial \tilde{\theta}_j^{(s)}} \cdot \lambda \right) \quad (\text{A.8})$$

where $\theta = \theta^{(t-1)} + S \cdot \lambda$. \square

Appendix B. Theorem 2's Proof

Since the *positive backward transfer* constraints are prohibitively hard to maintain efficiently, here we propose to transform them into left-hand-side of Equation (B.1), which incorporates schematic memory and theoretical guarantee on the transformation error stated in Theorem 1.

$$g_*(t, i) \cdot g(i, i, t-1)^\top \geq 0 \quad \implies \quad \mathcal{L}(R_i(X^{(i)}); \theta) \leq \mathcal{L}(R_i(X^{(i)}); \theta^{(t-1)}) + \varepsilon \quad (\text{B.1})$$

Theorem 2. *We have the following Equation:*

$$\sum_j \frac{\partial \ell(f(R_i(X^{(i)}); \tilde{\theta}^{(s)}), Y^{(i)})}{\partial \tilde{\theta}_j^{(s)}} \cdot \lambda \leq \min_k \max_{j \neq k} \left| \lambda - \frac{g_j(t, i, s)}{g_k(t, i, s)} \cdot \lambda \right| \cdot \|g(i, i, s)\|_2 \quad (\text{B.2})$$

Proof. Then Equation (B.1) can be rewritten as: $\sum_j g_j(t, i, s) g_j(i, i, s) \geq 0$. Also, suppose the step of the optimization in t is small enough, then we have $g_j(t, i, s) \leq 0$. Defining a positive scalar $\xi \geq 0$ such that $\sum_j g_j(t, i, s) g_j(i, i, s) -$

$\xi = 0$, we obtain:

$$\sum_j \frac{\partial \ell(f(R_i(X^{(i)}); \tilde{\theta}), Y^{(i)})}{\partial \tilde{\theta}_j} \cdot \lambda \quad (\text{B.3})$$

$$= \sum_j g_j(i, i, s) \cdot \lambda \quad (\text{B.4})$$

$$= \sum_{j \neq k} \left(\lambda - \frac{g_j(t, i, s)}{g_k(t, i, s)} \cdot \lambda \right) g_j(i, i, s) + \frac{\xi}{g_k(t, i, s)} \quad \left(\sum_j g_j(t, i, s) g_j(i, i, s) - \xi = 0 \right) \quad (\text{B.5})$$

$$\leq \sum_{j \neq k} \left(\lambda - \frac{g_j(t, i, s)}{g_k(t, i, s)} \cdot \lambda \right) g_j(i, i, s) \quad (\xi \geq 0, g_k(t, i, s) \leq 0) \quad (\text{B.6})$$

$$\leq \sum_{j \neq k} \left| \lambda - \frac{g_j(t, i, s)}{g_k(t, i, s)} \cdot \lambda \right| \cdot |g_j(i, i, s)| \quad (\text{B.7})$$

$$= \sum_{j \neq k} \left| \lambda - \frac{g_j(t, i, s)}{g_k(t, i, s)} \cdot \lambda \right| \cdot \frac{|g_j(i, i, s)|}{\|g(i, i, s)\|_2} \|g(i, i, s)\|_2 \quad (\text{B.8})$$

$$\leq \sum_{j \neq k} \left| \lambda - \frac{g_j(t, i, s)}{g_k(t, i, s)} \cdot \lambda \right| \cdot \frac{|g_j(i, i, s)|}{\|g(i, i, s)\|_1} \|g(i, i, s)\|_2 \quad (\text{B.9})$$

$$\leq \max_{j \neq k} \left| \lambda - \frac{g_j(t, i, s)}{g_k(t, i, s)} \cdot \lambda \right| \cdot \|g(i, i, s)\|_2 \quad (\text{Holder's inequality}) \quad (\text{B.10})$$

Since $k = 1, \dots, J$, we finally have:

$$\sum_j \frac{\partial \ell(f(R_i(X^{(i)}); \theta), Y^{(i)})}{\partial \theta_j} \cdot \lambda \leq \min_k \max_{j \neq k} \left| \lambda - \frac{g_j(t, i, s)}{g_k(t, i, s)} \cdot \lambda \right| \cdot \|g(i, i, s)\|_2 \quad (\text{B.11})$$

□

Appendix C. Continual learning experimental setup on civil unrest datasets

Each dataset was obtained from one Latin American country. Five datasets were used respectively for the countries: Brazil, Colombia, Mexico, Paraguay, and Venezuela. The tweet texts from Twitter were adopted as the model inputs. The goal is to utilize one date input to predict whether there will be an event in the next day in different cities (i.e. tasks). In each case the data for the period from July 1, 2013 to February 9, 2014 was used for

training and the data from February 10, 2014 to December 31, 2014 for the performance evaluation. The event forecasting results were validated against a well-established labeled event set, the Gold Standard Report [44]³, GSR is a collection of civil unrest news reports from the most influential newspaper outlets in Latin America [48]. An example of a labeled GSR event is given by the tuple: (City="Maracaibo", State="Zulia", Country="Venezuela", Date="2013-01-19", Event="True").

Although the original datasets were not ready to use in the context of continual learning, it can be easily converted. We first selected around 10 to 15 big cities in each country (i.e. dataset) that have high populations as the tasks. The model will need to learn to perform event forecasting in each city one by one. To deal with the class imbalance problem, we up-sampled the positive samples (i.e. there will be an event happen the next day) and down-sampled the negative ones, yielding around 300 training samples per task. Following the online continual learning setup, the model will need to perform learning sequentially on each task's training samples in an online fashion (i.e. samples from the previous batch will not be accessible anymore unless they are stored into the replay buffer).

³Available: <https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/EN8FUW>