Neurocomputing 500 (2022) 1063–1074

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Go ahead and do not forget: Modular lifelong learning from event-based data

Vadym Gryshchuk [a,*], Cornelius Weber [a], Chu Kiong Loo [b], Stefan Wermter [a]

[a] Knowledge Technology, Dept.of Informatics, University of Hamburg, Vogt-Koelln-Strasse 30, Hamburg 22527, Germany
[b] University of Malaya, Department of Artificial Intelligence, Lembah Pantai, Kuala Lumpur 50603, Malaysia

ARTICLE INFO

Article history:
Received 5 February 2022
Revised 3 May 2022
Accepted 28 May 2022
Available online 01 June 2022

Keywords:
Lifelong learning
Habituation
Event-based data
Bio-inspired artificial intelligence

ABSTRACT

Lifelong learning is a long-standing aim for artificial agents that act in dynamic environments in which an agent needs to accumulate knowledge incrementally without forgetting previously learned representations. Contemporary methods for incremental learning from images are predominantly based on frame-based data recorded by conventional shutter cameras. We investigate methods for learning from data produced by event cameras and compare techniques to mitigate forgetting while learning incrementally. We propose a model that is composed of both, feature extraction and incremental learning. The feature extractor is utilized as a self-supervised sparse convolutional neural network that processes event-based data. The incremental learner uses a habituation-based method that works in tandem with other existing techniques. Our experimental results show that the combination of different existing techniques with our proposed habituation-based method can help avoid catastrophic forgetting even more, while learning incrementally from the features provided by the extraction module.

1. Introduction

The human brain possesses an extraordinary ability to learn and accumulate relevant knowledge from a vast amount of information. Due to synaptic plasticity, the brain can learn new representations or even eradicate previously learned information [1]. Synaptic plasticity is a basis for shaping memory and learning. This leads to the notion known as lifelong learning [2], which is a long-standing challenge for artificial agents [3].

Bio-psychological views on the process of learning inspired us to investigate the applicability and transfer of human learning to the area of artificial intelligence. Experiments on the primary visual cortex (V1) in macaques showed limited ability for reorganization after lesions. Moreover, visual cortex plasticity is very high during an early period of development and decreases with brain development [4,5]. Though a direct transfer of these experimental results is not always reasonable, they provide a solid foundation for the development of new methods [6].

For our experiments, we base our research on the data produced by event cameras. Event cameras are biologically inspired dynamic sensors, asynchronously capturing brightness changes per pixel location for a minuscule time step. Compared to conventional shutter cameras, event sensors have a higher dynamic range and lower power consumption. Event cameras encode not only space but also time at a very high resolution. This encoding leads to the application of event cameras to challenging scenarios, in which fast responses and a high temporal resolution play a prevalent role.

However, the processing of event-based data requires new methods or adaptability to current approaches in deep learning to reveal the true potential of the event cameras. The use of event cameras for challenging environments requires rethinking the appropriate methodology. Thus, learning systems that can learn in harsh environments with power limitations in a continuous way are needed. Self-supervised learning [7,8] and continuous learning through experience replay [9–11] provide the foundation for the construction of such systems. Many of the state-of-the-art methods for lifelong learning rely on the feature extractor that is learned in a supervised way [11,9]. However, unsupervised learning is more appropriate, since it does not require labelled data and it circumvents the bias-variance tradeoff. Some methods propose to use powerful extractors like GPT-2 that are trained without any supervision, but they are applied to the context of language processing [12]. However, none of these methods were used in scenarios in which an agent accumulates information from event-based data incrementally. Therefore, we propose to use a feature

* Corresponding author.
E-mail addresses: vadym.gryshchuk@protonmail.com (V. Gryshchuk), cornelius.weber@uni-hamburg.de (C. Weber), chu.kiong.loo@uni-hamburg.de (C.K. Loo), stefan.wermter@uni-hamburg.de (S. Wermter).

https://doi.org/10.1016/j.neucom.2022.05.101
0925-2312/

extractor for event-based data that is trained without any supervision.

Hereby we follow a two-staged approach: firstly, we train a feature extractor, inspired by learning in the visual cortex, from the event-based data without any labels; secondly, the lifelong learning component is trained on the data provided by the feature extractor. The first stage attempts learning the same way infants do, not using any labels to train a feature extractor. The learning process merely relies on self-supervision, which has been getting a lot of attention recently in the research community [13,8]. The second stage operates on the features provided by the model in the first stage. These features are extracted representations from the input data.

We use a sparse convolutional neural network (SCNN) [14,15] trained in a self-supervised way as a feature extractor for the continuous learning module. The learned and extracted representations from SCNN are used then by the continuous learning module to enable learning in dynamic and novel environments. For the design of the continuous learning module, we follow the methodology presented by Ven et al. [9] and introduce the habituation method [16,11] as an additional technique to mitigate catastrophic forgetting. Habituation regulates the adjustability of the model's parameters to novel information, thus restricting changes to previously learned knowledge.

Although we use SCNN trained in a self-supervised way as a feature extractor, we evaluate additionally Phased LSTM [17] that can alternatively be used to extract features from the event-based data. Respectively, we train the Phased LSTM in a supervised way. However, based on the results presented in Section 4.2, impermissibility of Phased LSTM for long event-based sequences (Section 5.1), and biological plausibility of self-supervised learning for feature representation, we focus on the SCNN.

The main goal of this paper is to develop an approach for lifelong learning from event-based data. We consider object classification tasks as the evaluation routine. Thus, the main research question is defined as:

What methodology is suitable for lifelong learning from event-based data?.

The main contribution is:

- A habituation-based technique to mitigate catastrophic forgetting in neural networks which are trained with the backpropagation method. The proposed habituation-based method controls the rate of change to the neuronal weights, which encode information for the previous experiences, with each newly learned task.
- A modular system for lifelong learning from event-based data that uses sparse convolutional neural networks and that combines the habituation mechanism with the already existing techniques to mitigate catastrophic forgetting.

This paper is organized as follows. Section 2 provides the background information that is needed to follow our approach. Our method is introduced and explained in Section 3. Section 4 outlines the experiments we use and their results. Finally, in Section 5 and Section 6, we discuss the implications of achieved results and summarize our work respectively.

2. Background

In this section, we first describe two alternative methods for processing data from event cameras: Phased LSTM and histogram representations. Then we describe methods to mitigate catastrophic forgetting, which are to be used in combination.

2.1. Processing of Events

An event camera is a vision sensor that captures changes in brightness intensity per pixel location. This change is called an event. An event from a vision sensor is a tuple *(x, y, t, p)*, where *(x, y)* are pixel coordinates, *t* is a time step, and *p* is a polarity value, which indicates an increase or decrease of brightness. The brightness sensitivity of an event camera is determined by a threshold, and if the brightness change is larger than the threshold, a $p \in \{-1, +1\}$ will be recorded.

Phased LSTM: Since each pixel in an event camera responds independently to brightness change, the generated asynchronous output carries challenges for the processing of such data. The event-based sequences can be processed by event-by-event methods or methods that group events [18]. The event-by-event methods process events sequentially. As an example of such a method, Phased LSTM [17] extends LSTM [19] by introducing a new time gate, which allows the updates to cell and hidden states only during its open periods. A time gate is a rhythmic oscillator and it is controlled by three parameters: a period ψ, the phase shift *s* of the period, and the ratio *r* of the duration of the phase within ψ. The parameter *r* controls the open states of a time gate.

Histogram representations: Another approach is to group events to image-like data. A histogram is one of the possibilities to convert events to frame representations of events [20]. To convert events to a histogram, the occurrences of brightness changes at any pixel location over a particular period of time are counted. Specifically, the events at pixel locations (x, y) that refer to a brightness increase are stored in one histogram (h^{+}), and the events that capture brightness decrease are saved to another histogram (h^{-}) [20]:

$$h^{+}(x,y) = \sum_{t_i \in T, p_i = +1} \delta(x - x_i, y - y_i), \tag{1a}$$

$$h^{-}(x,y) = \sum_{t_i \in T, p_i = -1} \delta(x - x_i, y - y_i), \tag{1b}$$

where

$$\delta_{kj} = \begin{cases} 0 & \text{if } k \neq j, \\ 1 & \text{if } k = j. \end{cases} \tag{2}$$

Two histograms act then as two input channels to a convolutional neural network (CNN). Since an event camera reacts only to brightness change, a lot of pixel locations in a histogram can contain no values. Thus, a histogram represents the salient events in a scene captured by an event camera. However, a conventional convolutional operation causes dilation when the input is sparse. Therefore, a Sparse CNN, which preserves sparseness, is a more reasonable choice [15].

2.2. Lifelong Learning

Methods that are successfully used to mitigate catastrophic forgetting rely on regularization-based techniques [21], growing architectures [11] or use replay mechanisms [11,9]. Regularization-based methods restrict the updates to the model's parameters that are important for encoding previous knowledge. One of these methods that estimates this importance is synaptic intelligence [21] which introduces a regularization term that is added to the total loss to penalize changes to important parameters while learning a new task. We will present and evaluate a simpler method using a neuron habituation mechanism. The methods that utilize replay mechanisms store either some previous samples or learn the representations of previously learned data. A generative model, in particular a variational autoencoder, can be used to learn latent representations of data [9].

The approach for continual learning proposed by Ven et al. [9] incorporates different techniques. In the following, we describe the main methods that they apply:

Replay through feedback: Generative replay is one of the methods to mitigate catastrophic forgetting [22]. Furthermore, a replay of memories is evident in the brain, where the representations learned by the hippocampus are replayed to the cortex [23,24]. Generative replay is modelled as a variational autoencoder (VAE) that has an explicit density estimation [25]. Thus, a latent representation in the VAE acts as a generator for previous experiences. However, another model would be necessary to classify samples. Instead of using two separate models, the encoder part of the VAE is shared with a classifier, and a softmax layer is added as a separate head after the last layer in the encoder. Feedback of a generative component alters not only the VAE's parameters but also that of a classifier. During replay generated samples are labelled by the classifier not only with the most likely predicted class but with the predicted probabilities for all possible classes, which is known as distillation. The final loss for N learned episodes is calculated as follows:

$$\mathscr{L}_{total} = \frac{1}{N}(\mathscr{L}^{C}_{current} + \mathscr{L}^{R}_{current}) + (1 - \frac{1}{N})(\mathscr{L}^{D}_{replayed} + \mathscr{L}^{R}_{replayed}), \tag{3}$$

where $\mathscr{L}_{current}$ is the loss for the data of the current episode, $\mathscr{L}_{replayed}$ is the loss for the replayed data of previous episodes, $\mathscr{L}^{C}$ is the cross-entropy classification loss, $\mathscr{L}^{R}$ is the reconstruction loss of the VAE, and $\mathscr{L}^{D}$ is the distillation loss for the generated data with the predicted probabilities for all possible classes and defined as:

$$\mathscr{L}^{D} = -T^{2}\sum_{c=1}^{C}\hat{y}_{c} log p^{T}_{\theta}(Y = c|\mathbf{x}), \tag{4}$$

where c is the class index, T is the temperature parameter that is used to scale logits in a softmax layer, p^{T}_{θ} is the probability distribution of an output layer defined by θ and $T, \hat{y}_{c}$ is the predicted distilled probability, and $\mathbf{x}$ is input data. Thus, all learning episodes are equally weighted, and the model considers the errors of a classifier and a generator. Sampled data from a generator are used as a training set together with data of the current task to mitigate forgetting of a generator. During the first task, a classifier and a generator are trained on the actual input data. During the second task and afterwards, a model is trained additionally with the generated samples from a generator. A classifier is used to predict classes for the actual and generated samples. These labelled samples are used then as training data for a generator.

Conditional replay: The prior over the latent representation of a VAE, often marked as z, is primarily modelled as a normal Gaussian distribution. Complex representations profit though from the richer density estimates. The Gaussian mixture is utilized for z, where each mode represents a class. Though an explicit number of learnable classes in real-life scenarios is not available, it provides nevertheless the ability to replay specific data conditioned on classes, as it happens in the brain too.

Internal gating: A subset of neurons are gated in the decoder part of a network [26]. The gating is conditioned on the classes that are being replayed. The gating operation resets all activation values of the selected neurons to zero. Gating resembles inhibition in the brain, where some neurons can reduce the responses of the other neurons, thus forcing selective attention [27,28]. Yet, the inhibitory process in the brain is much more complicated resulting in complex brain states [29]. Nevertheless, just simple gating conditioned on internal context can deliver better results.

Internal replay: Two replay routines are possible: replaying directly input data or just hidden learned representations. The latter approach is more biologically inspired [24]. Moreover, experimental studies in neuroscience suggest that lower levels of the visual cortex reveal less plasticity [30].

In our approach, we incorporate the techniques proposed by Ven et al. [9] and show that our architecture for lifelong learning from event-based data can utilize the same methods that are applied to frame-based images.

3. Modular Lifelong Learning

We propose an architecture that consists of a feature extractor and a component for continuous learning, visualized in Fig. 1. To the best of our knowledge, there are no approaches for direct comparison. Lungu et al. use a memory-based method for incremental learning of hand gestures [10]. However, they evaluated their approach on much simpler event-based data than we do. Section 5.1 compares more in detail the application of our approach and theirs to real-life scenarios.

3.1. Feature Extraction

The feature extraction module serves as a feature provider for the supervised continuous learning module. It is an independent module that can be represented with any model that processes event-based data and provides representative features. Since event-based data are sequence data encoding visual information, recurrent neural networks and convolutional neural networks are the most appropriate current methods for learning from such data. Thus, to design a feature extraction module, we compare Phased LSTM and Sparse CNN as possible models to extract features from events. We train Phased LSTM in a supervised way and Sparse CNN in a self-supervised way following the batch learning strategy. On the one hand, we use Phased LSTM as the event-by-event method, and on the other hand, we utilize Sparse CNN that learns from histograms as grouped events. The comparison of both models should give a future perspective on their practical use, rather than a direct comparison and validation, which is not the aim of our research. Therefore, based on the results provided in Section 4 and our goal to learn the feature extractor without any provided labels, we select only the self-supervised approach using Sparse CNN for the feature extraction module. Self-supervised learning is a subset of unsupervised learning, where no labels are provided during training. We follow the same strategy for self-supervised learning proposed by Chen et al. [8], however, instead of frame-based input, we provide events as histograms. The model applies random augmentations directly to histograms, thus learning in a contrastive way by maximizing the agreement between two augmented representations of the same object (Fig. 1, bottom).

3.2. Continuous Learning

The module for continuous learning operates on the features provided by the feature extraction module. The learning process follows the incremental strategy, where a model has access only to some object categories during a learning episode. Thus, a learning episode contains only a subset of non-repeating objects that belong to the same class or classes. We base our model (Fig. 1, top) on the architecture proposed by Ven et al. called brain-inspired replay [9].

Methods for incremental learning usually contain different techniques to mitigate catastrophic forgetting [11,9]. We extend the network presented in [9] by introducing habituation [16]. Thus, we add and combine the habituation method with the techniques presented and evaluated by Ven et al. [9] to boost the performance of the model while learning incrementally. Habituation is a decrease of response to repeated stimuli, found in neurons [16].

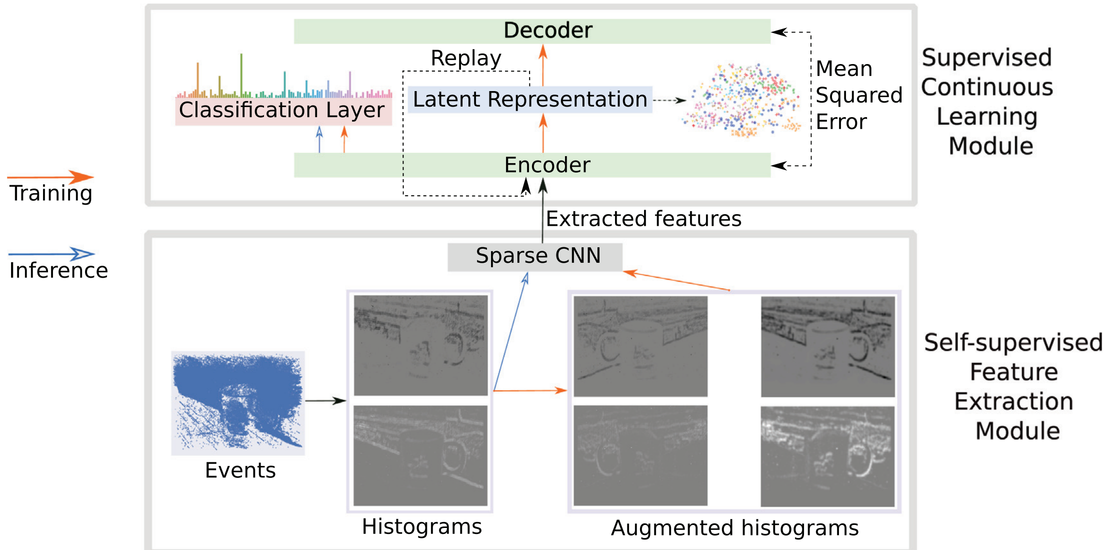


Fig. 1. Illustration of the proposed architecture. Sparse CNN trained in a self-supervised way extracts features from the events represented as histograms (for better visualization purposes the histograms are converted here to grayscale images). The continuous learning component learns incrementally from the extracted features by utilizing internal replay through variational autoencoder, synaptic intelligence, and habituation.

Habituation was successfully applied to self-organizing networks [31,11]. All neurons are initialized at the beginning with a habituation value of 1. When a neuron responds to input, its habituation value (also called habituation counter [11]) is reduced. The more a neuron responds, the lower its counter value becomes. During training, the calculated updates to the weights are then scaled by the habituation counter. Thus, a habituation counter suspends a neuron's plasticity by not allowing large updates to its weights. We modify slightly the habituation update rule presented in [11] and define it as:

$$\Delta h_i = \tau \cdot (1 - h_i) - \tau, \tag{5}$$

where h_i is the habituation counter of a neuron i, and τ is the decay rate that controls the steepness of decay. The larger the decay rate is, the faster the habituation counters drop.

In contrary to self-organizing networks, where a weight update rule is defined as the difference between input and neuron weights, scaled by a learning rate and a habituation rate, we apply habituation counters to gradients. Updates to the parameters in a neural network are performed through a backpropagation algorithm [32]. The gradients of the error function with respect to parameters are collected and the parameters are adjusted by these gradients. It means that all intermediate gradients for all operations are stored. We introduce a habituation counter only for the neurons in the last dense layer of the encoder, since the encoder represents a sensory information processing unit. Thus, the gradient of a neuron r_i in the last encoder layer l is scaled by the habituation counter and computed as:

$$\frac{\partial \mathcal{L}}{\partial r_i^l} \leftarrow \frac{\partial \mathcal{L}}{\partial r_i^l} \cdot h_i, \tag{6}$$

where ∂ is a partial derivative, $\mathcal{L}$ is the loss function defined in Eq. 3, and $\frac{\partial \mathcal{L}}{\partial r_i^l}$ is defined as:

$$\frac{\partial \mathcal{L}}{\partial r_i^l} = \sum_j \phi\prime(s_j^{l+1}) w_{ji} \frac{\partial \mathcal{L}}{\partial s_j}, \tag{7}$$

where s_j is a neuron in the layer $l+1, w_{ji}$ are the weights between neurons s_j and r_i, ϕ is the activation function of a neuron that defines an output given an input, and $\phi\prime$ is the derivative of ϕ. The calculated and downscaled gradient in layer l is propagated to all lower layers.

We compare different combinations of brain-inspired replay, synaptic intelligence and habituation to investigate the effect of the habituation-based approach on the mitigation of catastrophic forgetting while learning incrementally.

4. Experimental Results

In this section, we perform experimental studies to evaluate our proposed system for lifelong learning. Firstly, we describe the datasets we use. Secondly, we evaluate the feature extraction module by providing classification accuracy and visualizing feature maps. Finally, we investigate the increase of the model's performance in terms of classification accuracy during incremental learning when using the proposed habituation method in tandem with the investigated techniques proposed by Ven et al.[9].

4.1. Datasets

We train and evaluate the proposed model on the N-Caltech101 and N-MNIST datasets [33].

The N-Caltech101 dataset is based on the Caltech101 dataset [34] with 101 object categories and contains event-based representations recorded by an event camera from static images. While an image was shown on a screen, an event camera made three saccadic movements to record events. The range of values for x and y coordinates are [0, 239] and [0, 179] respectively. We use the same training and test sets as used in [35][1]. It should be noted that the whole dataset has an unbalanced number of samples per class, thus classification accuracy as a metric has a biased interpretation. However, our aim is to compare different strategies for lifelong learning and show the possible benefit of the habituation-based method.

The N-MNIST[2] dataset is converted from the MNIST dataset [36] which contains 70,000 grayscale images of digits. N-MNIST is created in the same way as the N-Caltech101 dataset. The resulting event-

[1] N-Caltech101 is available at http://rpg.ifi.uzh.ch/datasets/gehrig_et_al_iccv19/N-Caltech101.zip

[2] N-MNIST is available athttps://www.garrickorchard.com/datasets/n-mnist

based data contain the values in the range [0, 33] as indices for both x and y coordinates. Thus, a recorded sample in the N-MNIST dataset contains much fewer events than a sample from N-Caltech101.

4.2. Feature Extraction Module

To first evaluate the feature extraction module under optimal conditions, the whole training set is utilized to learn a feature extraction module following the batch learning strategy. A sample can contain thousands of events, thus the training time of Phased LSTM becomes intractable since Phased LSTM processes events sequentially. We address this issue in Section 5.1. Therefore, we randomly select only 5% of the events, but at least 5000 of them, when processing a sample from the N-Caltech101 dataset and at most 2000 events from the N-MNIST sample. A histogram for a sample from N-Caltech101 is created from at most 50,000 consecutive events; this interval of events is randomly placed over the whole sequence of events. For a sample from N-MNIST, all events are used to create a histogram.

The Phased LSTM receives the pixel locations (x, y) with the corresponding polarity values as input. Table 1 lists the hyper-parameters that were used. We select randomly a subset of events from each sample in all datasets and train a model for 20 epochs. Period ψ of each neuron is initialized uniformly from exponential space and is learnable. The number of features in the hidden state of the Phased LSTM is 128.

We use the same Sparse CNN model proposed in [35], which is composed of 12 convolutional layers, and of which the first 11 perform submanifold convolutional operation. Each convolutional layer is followed by a batch normalization layer. Additionally, a max-pooling layer is introduced before the third, fifth, seventh, and ninth submanifold convolutional layers. The last layer of the Sparse CNN model is a transformation function that converts the sparse output to a dense convolutional layer containing 256 kernel filters of the size 2×3. The model is trained for 100 epochs using a batch size of 128 samples.

Table 2 shows the classification accuracy of the Phased LSTM and the classification accuracy of a linear classifier trained on top of the Sparse CNN. Phased LSTM achieves worse results on N-Caltech101, however, it operates on a portion of events, which can lead to a drop in performance. Yet, using even 5% of events for the samples from the N-Calthech101 dataset requires at least 6 times as much training time as the Sparse CNN without considering the GPU memory consumption. Furthermore, a feature extractor that is trained in a supervised way on the same training set that is used for the continuous learning module can provide a biased judgement. Thus, either a feature extractor that is trained without labels or a feature extractor that is used to extract features from different data is a more reasonable approach. In the following experiments, we use Sparse CNN as a feature extractor for the continuous learning module. The same training set that is used to train the feature extractor is used to train the model of a continuous learning module.

For a better understanding of what features are learned, we pass histograms of a sample through each convolutional layer of the Sparse CNN and save the output produced by the feature maps of the last convolutional layer. Fig. 2 visualizes feature maps after the last convolutional layer containing 256 kernel filters. The histograms in Fig. 2 contain noisy data (black squares) since some pixel locations have a very high number of recorded brightness changes contrary to other pixel loactions. However, feature maps provide distinctive features, such as the contours of objects, which some kernel filters have learned. Thus, kernel filters learned to focus on the contours of the objects. Feature maps of each object show a pattern, where background colours are repeated (from left to right). Such behaviour can be explained by the saccadic movements of an event camera and the random placement of a sliding window over events. Thus, in addition to the learned holistic features, the model learned to represent brightness changes in general. Due to the planar movements of the recorded data, non-monotonic complex backgrounds have a great impact on the extracted features, as can be seen in Fig. 2 (b).

4.3. Continuous Learning Module

To evaluate the proposed habituation-based method, we combine habituation (H) with the brain-inspired replay (BIR) [9] and synaptic intelligence (SI) [21] methods. All common hyper-parameters are the same for all methods to provide a fair comparison. The extracted features provided by the Sparse CNN have a dimension of 1536 features per sample. The encoder and decoder are composed of two fully connected layers each, having 2000 units per layer. The latent representation layer is of size 4096. We train the continuous learning module for 200 and 3000 iterations per learning episode when using N-Caltech101 and N-MNIST datasets respectively. Based on a different mask for each class, the activations of 60% of the randomly selected neurons in the decoder part of the model are set to zero during training.

4.3.1. Comparison of Methods

We use the class-incremental learning strategy, during which a model has access only to the data of the current learning episode. During each episode, a model learns to classify objects belonging to specific object categories which are randomly selected and never shown in the following episodes again. The accuracy is measured always on learned object classes so far from the current and previous episodes.

Table 1
Hyper-parameters used to train Phased LSTM.

	Batch size	Max./Min. events	Learning rate	Min./Max. period	Open gate ratio
N-MNIST	32	2000/-	0.003	ln(1)/ln(1e6)	0.05
N-Caltech101	30	5%/5000	0.003	ln(1)/ln(1e6)	0.05

Table 2
Evaluation of the feature extraction module on the N-Caltech101 and N-MNIST datasets. Phased LSTM is trained in a supervised way. Sparse CNN is evaluated by adding and training a linear classifier on top of frozen features. †Unbalanced training and test sets. ‡ Balanced training and test sets.

	Phased LSTM (supervised)		Sparse CNN (self-supervised)		
	Training	Test	Training	Test Top-1	Test Top-5
N-Caltech101†	35.35	30.90	51.49	42.38	62.60
N-MNIST‡	94.78	94.07	92.75	92.51	99.65

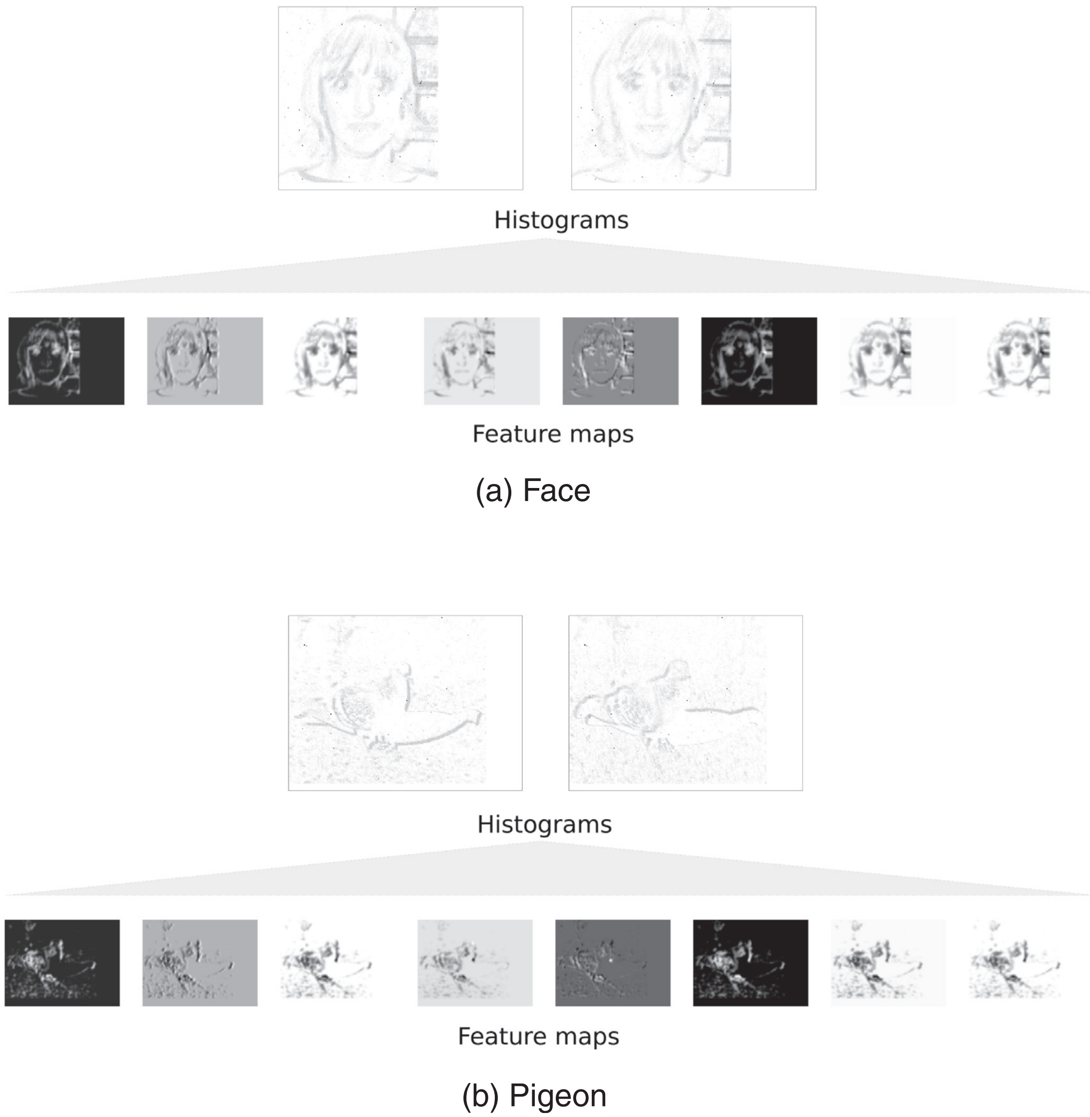


Fig. 2. Visualization of feature maps produced by the last convolutional layer containing 256 kernel filters. Only each consecutive 32nd feature map is visualized (from left to right).

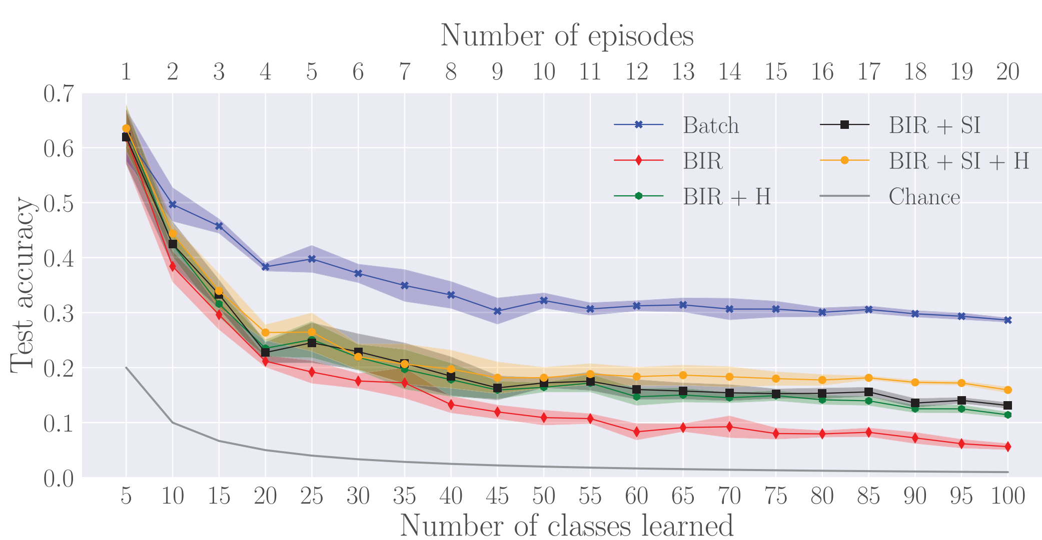


Fig. 3. Class-incremental learning on N-Caltech101 over 20 learning episodes. The proposed habituation method combined with synaptic intelligence and brain-inspired replay achieves the best performance. Batch: all data are used up to the current episode during training, BIR: brain-inspired replay, H: habituation, SI: synaptic intelligence.

Fig. 3 illustrates class-incremental learning on N-Caltech101. The number of learning episodes is set to 20. Each learning episode contains samples from 5 different non-repeating object categories. The shaded areas show the standard error of the mean (SEM). The experiment was executed for five trials, and each trial, a new seed and the random order of classes were used. The *Batch* method, in contrary to other approaches, considers all data up to the current learning episode and could be considered as an upper bound. During the first learning episodes, the test accuracy drops drastically for the incremental learning and batch strategies. This rapid decrease is caused by the low representational power of features provided by the feature extractor. Afterwards, the accuracy gradually drops down together with the batch learning accuracy. The $BIR + H$ and $BIR + SI$ methods achieve, after learning data of all episodes, on average a classification accuracy of 11.42 ± 0.50 SEM and 13.12 ± 0.53 SEM, respectively. The combination of synaptic intelligence and habituation in $BIR + SI + H$ provides a slight but significant increase in the test accuracy: 15.94 ± 0.54 SEM. Though the test accuracy is low at the end, this is not a primary interest of our investigation. The methods that can reduce the gap between the *Batch* and incremental learning strategies constitute one of the main objectives of this work.

The habituation-based method (H) has two hyper-parameters: a decay rate τ and the fraction γ of neurons with the highest activation values that are allowed to be habituated during each learning episode. Synaptic intelligence (SI) has a strength parameter c that regulates a tradeoff between past and new experiences. Larger values for c restrict more the updates to the weights that are

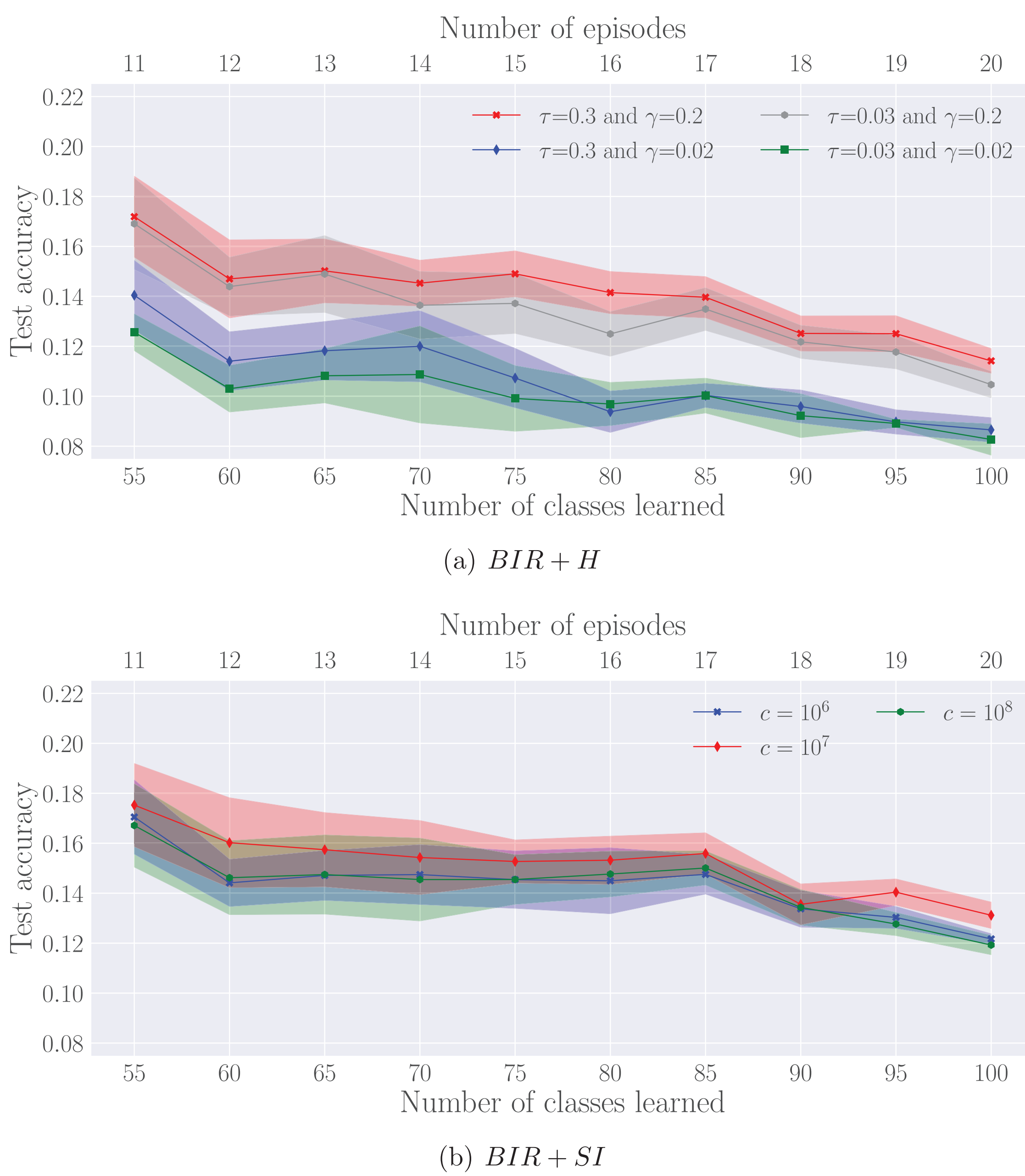


Fig. 4. Comparision of hyper-parameters used for habituation and synaptic intelligence during class-incremental learning on N-Caltech101 over 20 learning episodes. For better visualization purposes only results starting from the 11th learning episode are shown. (a) Classification accuracy for the strategy $BIR + H$ using various values for the decay rate τ and the fraction γ of neurons that are allowed to be habituated. (b) Classification accuracy for the strategy $BIR + SI$ using three different values for the strength parameter c. BIR: brain-inspired replay, H: habituation, SI: synaptic intelligence.

important for previously learned experiences. Fig. 4 compares different values for the hyper-parameters used in the strategies *BIR + H* ((a)) and *BIR + SI* ((b)). Consequently, the strength parameter c was set to 10^7, and the γ and τ values were set to 0.3 and 0.2 respectively for all our experiments, unless indicated otherwise.

4.3.2. Short Learning Horizon

It is obvious that the classification accuracy of the continuous learning component depends also on the representative power of the extracted features. While other recent approaches can provide more representative features from event-based data [37,38], in this paper, we concentrate on the techniques that can be used to mitigate catastrophic forgetting. To show the benefit of well-learned features by the extraction module on the classification accuracy, we evaluate our proposed method on the N-MNIST dataset. Fig. 5 illustrates class-incremental learning on N-MNIST, which contains 10 classes. The number of learning episodes is set to 5, thus each episode contains samples from 2 different non-repeating digit categories. The *BIR + SI + H* learning strategy yields the best performance: 83.57 ± 0.55 SEM.

4.3.3. Long Learning Horizon

Since real-life scenarios are not fully comparable with experimental settings containing a small number of learning episodes, we try to mimic such conditions by setting the number of learning episodes to 50. Fig. 6 shows that the combination of synaptic intelligence and habituation in *BIR + SI + H* preserves the best performance by yielding a test accuracy of 12.20 ± 0.29 SEM, while *BIR + SI* achieves 10.49 ± 0.36 SEM. The standard error of the mean over 5 learning trials is lower in comparison to the results achieved over 20 learning episodes (Fig. 3). This could be explained by the

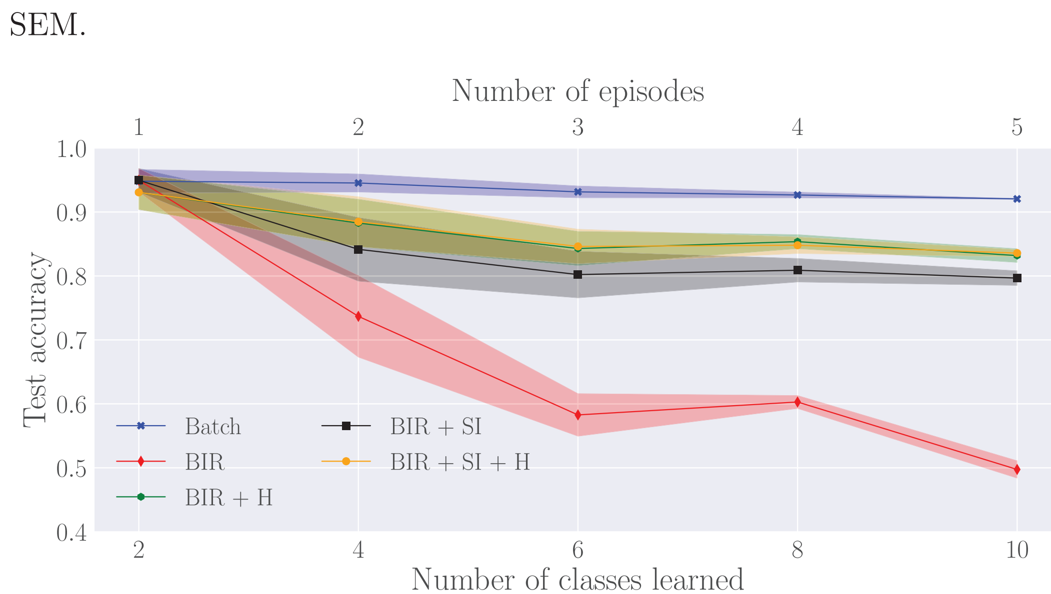


Fig. 5. Class-incremental learning on N-MNIST over 5 learning episodes. The proposed habituation method combined with synaptic intelligence and brain-inspired replay achieves the best performance. Batch: all data are used up to the current episode during training, BIR: brain-inspired replay, H: habituation, SI: synaptic intelligence.

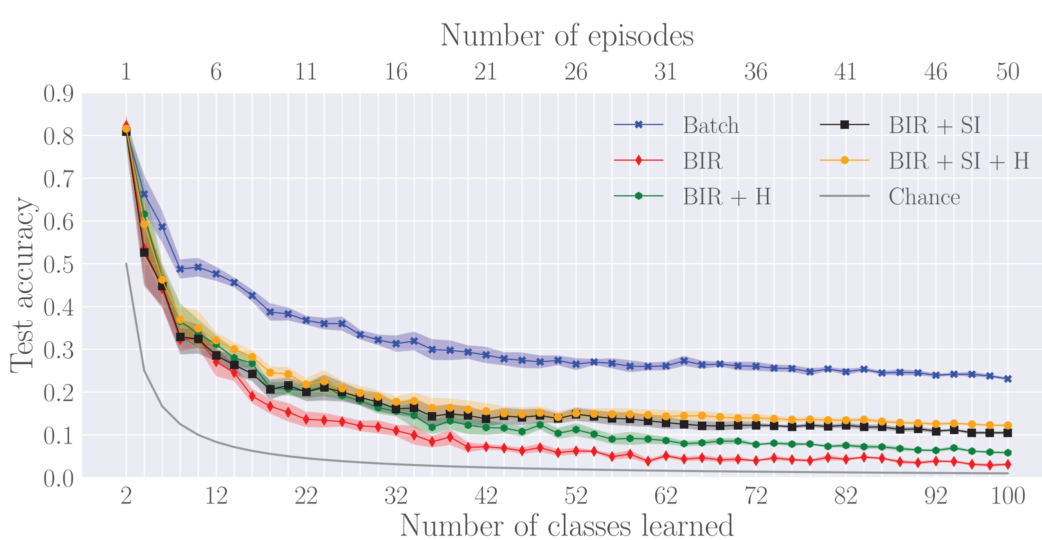


Fig. 6. Class-incremental learning on N-Caltech101 over 50 learning episodes. The proposed habituation method combined with synaptic intelligence and brain-inspired replay achieves the best performance. Batch: all data are used up to the current episode during training, BIR: brain-inspired replay, H: habituation, SI: synaptic intelligence.

lesser complexity of tasks since each added task contains samples from only 2 different classes.

5. Discussion

5.1. Learning from Event-based Sequences

Event-based sequences impose a substantial restriction on the selection of methods that can be used for learning from events. Due to the asynchronous nature of events and their sparsity in the time domain, more effective methods are required. Two approaches are possible: model-driven and data-driven methods. Phased LSTM is an example of a model-driven approach, in which a new time gate is introduced to learn representations from asynchronous event-based data. Thus, events are processed directly with no or little modifications. Though Phased LSTM is a conceptually simple method for processing time-based data, it cannot deal with event-based sequences that can have millions of time steps, as computation time becomes intractable.

Yet, Phased LSTM is a rational choice for sequential processing of asynchronous data, the training time of Phased LSTM becomes intractable since event cameras have a microsecond temporal resolution.

Thus, event data impels researchers in machine learning to create better approaches for the representation of events. In this work, we converted events into a histogram. A histogram neglects the time domain and uses only a part of events by applying a sliding window. If a window is too large, then detailed temporal information gets lost; if a too small window size is used, then many bins remain empty, particularly for sparse event data, and learning of features becomes more difficult. We have set the window size to 50,000 for the N-Caltech101 dataset, which is the same value that was used in [35].

As our experiments show, Phased LSTM is not a practical method for event-based data that can have thousands of events. To better understand the scaling properties of Phased LSTM and Sparse CNN with respect to input, we need to look at data processed by these two methods. We discard the internal complexity of Phased LSTM and Sparse CNN and consider only the size of input data. Table 3 shows the length of sequences in both datasets averaged over all samples. Since Phased LSTM processes data sequentially, it scales with $\mathcal{O}(N)$, where N is the number of events per sample. Scaling of Sparse CNN depends on the distribution of N events across the input resolution P for each discretization interval. Thus, Sparse CNN scales with $\mathcal{O}(2P)$, where the factor of 2 is due to having one histogram for brightness increase and another one for brightness decrease. The input size of samples from NMNIST is 34×34 and of N-Calctech101 is 180×240, which results in $\mathcal{O}(34 \cdot 34 \cdot 2)$ and $\mathcal{O}(180 \cdot 240 \cdot 2)$, respectively. Since the time complexity of Phased LSTM depends extensively on the number of events, and event-based data contains a large number of events, as provided in Table 3, Phased LSTM becomes impermissible for long sequences. Yet, we hypothesize that Phased LSTM can achieve better results than Sparse CNN if more events per sample can be considered, as shown in Table 2 for the NMNIST dataset.

In this paper, we consider complex input data with many learning episodes for incremental learning from events. The most closely related work, which studies incremental learning from event-based data, was done by Lungu et al. [10]. They use the iCaRL incremental learning algorithm, which stores the selected samples of previously learned classes [39]. However, the methods and the datasets used in their work differ substantially from ours. In our experiments, we use samples with different backgrounds, while their dataset of hand symbols has a simple and not moving background. Since the datasets we use for our experiments are converted from frame-based images, the background has the same rate of motion as an object of interest. This setting is not always true for real-life scenarios, but sometimes even more evident. For example, if the object of interest is a person standing on a shore and a ship is passing in the background, an event camera will capture mostly the ship but not the person. This can be a limitation of event cameras for object recognition. Nonetheless, the methods applied to object recognition build a basis for online object recognition and tracking.

Table 3
Length of event sequences averaged over all training samples.

NMNIST	N-Caltech101
4,171	116,515

5.2. Habituation

We compare the proposed habituation method in tandem with the already existing techniques investigated by Ven et al.[9]. The conducted experiments show that the addition of a simple habituation mechanism can increase the model's performance in terms of classification accuracy while learning incrementally, as shown in Fig. 3. The achieved results depend on the decay rate τ and the fraction γ of neurons that are allowed to be habituated, which regulate the rigidity and plasticity of the model (Fig. 4 (a)). Yet, the γ and τ values set to 0.3 and 0.2 respectively show an appropriate setting for short (Fig. 5) and and long (Fig. 6) learning horizons. The fixed γ and τ values can provide a constraint for real-life scenarios with an unknown learning horizon.

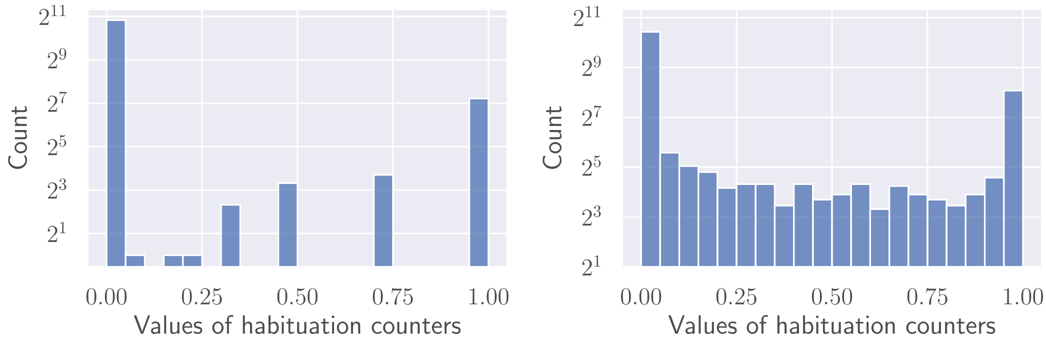
Habituation counters can be used to investigate the capacity of a model to learn new information. The habituation counters that are close to zero for most of the neurons indicate the saturation of a model. Thus, these neurons cannot learn fully novel information due to the restriction imposed by habituation counters. On the other hand, when most of the habituation counters are close to one, a model can learn new experiences without highly imposed restrictions. Fig. 7 shows the distribution of habituation counters in the last layer of the encoder after incremental learning of all 20 episodes on N-Caltech101. Lower values for τ or γ, or both, preserve a model's capacity to acquire new representations over many learning episodes (Fig. 7 (b)), while higher values are more suitable for the scenarios with a small number of learning episodes (Fig. 7 (a)). Neurons with habituated counters close to zero will not be able to learn new information anymore since the gradients of the neuron's weights will be scaled to a minuscule number. In Fig. 4 (a), we show the influence of the τ and γ parameters on the test accuracy. Higher values for τ and γ represent a low-plasticity setting (red curve), while lower values let the model forget previous experiences quicker (green curve), thus representing a high-plasticity setting.

Thus, different values for τ and γ will affect the model's performance in terms of its ability to preserve previous knowledge and learn new information. Though the equilibrium between plasticity and rigidity are exclusive in this sense, a preference has to be defined beforehand.

5.3. Limitations

The following limitations are worth mentioning that are common for devising systems for real-life scenarios:

Data: The used datasets are created in artificial conditions, where an event camera performs saccadic movements to record motion. Thus, the recorded events represent planar movements rather than smooth event flow that is found in natural conditions.



(a) $\tau = 0.3$, $\gamma = 0.2$ (b) $\tau = 0.03$, $\gamma = 0.02$

Fig. 7. Distribution of habituation counters for the neurons of the last dense layer in the encoder after incremental learning of all 20 episodes on N-Caltech101 following the strategy $BIR + SI + H$ that combines brain-inspired replay (*BIR*), synaptic intelligence (*SI*), and habituation (*H*). Vertical axes are scaled to log_2. (a) Larger values for the decay rate τ and the fraction γ of habituating neurons let habituation counters decrease faster towards 0, thus restricting further plasticity of a model. (b) Smaller values of τ and γ preserve the plasticity.

Consequently, systems that can operate on events in real-time are of high interest.

Feature extraction module: The features provided by the feature extraction module for the continuous learning module need to be universal and generalizable. In our experimental settings, we train Sparse CNN on the data that is also used for incremental learning. This setting requires a dataset for feature detector pretraining, and may not be suitable for novel data from an unknown distribution.

5.4. Future Research

The performance of the proposed method depends on the quality of extracted features. Thus, a model that can learn meaningful representations plays an important role. Moreover, a feature extractor should be a universal representational module for all kinds of input data. We view the extraction module as an independent learning component, which can be swapped with any technique that achieves best results.

The introduction of habituation to the continuous learning module showed a positive effect on test accuracy. However, a scenario with many learning episodes can bring the model to its limits. Thus, further investigation and research on methods that can provide better results for incremental learning are needed. We contemplate that an additional plasticity modulation will create better representations inside a network. Specifically, separate inhibitory and excitatory neurons have the potential to create a trade-off between a controlled forgetting and acquisition of novel information [40]. The use of a larger and more sophisticated encoder-decoder architecture could provide better results and be more applicable for complex tasks.

6. Conclusion

We presented an architecture for lifelong learning, consisting of a feature extractor and a module for continuous learning. We showed that the Phased LSTM is not a favourable method for learning long event-based sequences for large data. The Sparse CNN trained in a self-supervised way achieves better results although histograms discard short time-scale information. To mitigate catastrophic forgetting, a combination of brain-inspired replay and synaptic intelligence with a simple habituation method, which was previously applied to self-organizing neural networks, yields the best performance over a class-incremental learning of 100 classes. We highlighted the main challenges of the presented system for real-life scenarios, in which powerful feature extractors operating on event flow data are required. The use of event-based data recorded in natural conditions and optimal methods for learning from events are yet to be investigated. With this presented approach, we provide not only an additional technique to mitigate catastrophic forgetting while learning incrementally, but also insights into the application of event cameras for scenarios in which incremental accumulation of knowledge is crucial.

Code availability

Our code is available from: http://software.knowledge-technology.info.

Funding

The authors gratefully acknowledge the support in part by the Georg Forster Research Fellowship for Experienced Researchers (Neurosynthelligence) from the Alexander von Humboldt Foundation, and in part by the German Research Foundation (DFG) on Crossmodal Learning under Grant TRR169.

CRediT authorship contribution statement

Vadym Gryshchuk: Conceptualization, Methodology, Software, Validation, Investigation, Resources, Writing - original draft, Writing - review & editing, Visualization. **Cornelius Weber:** Conceptualization, Validation, Resources, Writing - review & editing, Supervision. **Chu Kiong Loo:** Validation, Writing - review & editing, Supervision. **Stefan Wermter:** Resources, Writing - review & editing, Supervision, Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work is a substantial extension, selected and invited after a presentation at ESANN 2021/29th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning [41].

References

[1] L.F. Abbott, S.B. Nelson, Synaptic plasticity: taming the beast, Nature Neuroscience 3 (2000) 1178–1183, https://doi.org/10.1038/81453.

[2] G. Parisi, R. Kemker, J. Part, C. Kanan, S. Wermter, Continual lifelong learning with neural networks: A review, Neural Networks 113 (2019) 54–71, https://doi.org/10.1016/j.neunet.2019.01.012.

[3] Z. Chen, B. Liu, Lifelong machine learning, second edition, Synthesis Lectures on Artificial Intelligence and Machine Learning 12 (2018) 1–207. doi: 10.2200/S00832ED1V01Y201802AIM037..

[4] L. Zaadnoordijk, T.R. Besold, R. Cusack, The next big thing(s) in unsupervised machine learning: Five lessons from infant learning, arXiv e-prints (2020). arXiv:2009.08497..

[5] J. Fiser, R.N. Aslin, Statistical learning of new visual feature combinations by infants, Proceedings of the National Academy of Sciences 99 (2002) 15822–15826. doi: 10.1073/pnas.232472899..

[6] B.A. Richards, T.P. Lillicrap, P. Beaudoin, Y. Bengio, R. Bogacz, A. Christensen, C. Clopath, R.P. Costa, A. de Berker, S. Ganguli, C.J. Gillon, D. Hafner, A. Kepecs, N. Kriegeskorte, P. Latham, G.W. Lindsay, K.D. Miller, R. Naud, C.C. Pack, P. Poirazi, P. Roelfsema, J. Sacramento, A. Saxe, B. Scellier, A.C. Schapiro, W. Senn, G. Wayne, D. Yamins, F. Zenke, J. Zylberberg, D. Therien, K.P. Kording, A deep learning framework for neuroscience, Nature Neuroscience 22 (2019) 1761–1770, https://doi.org/10.1038/s41593-019-0520-2.

[7] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, A. Joulin, Unsupervised learning of visual features by contrasting cluster assignments, in: H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, volume 33, Curran Associates Inc, 2020, pp. 9912–9924. URL:https://proceedings.neurips.cc/paper/2020/file/70feb62b69f16e0238f741fab228fec2-Paper.pdf..

[8] T. Chen, S. Kornblith, M. Norouzi, G. Hinton, A simple framework for contrastive learning of visual representations, in: H.D. III, A. Singh (Eds.), Proceedings of the 37th International Conference on Machine Learning, volume 119 of Proceedings of Machine Learning Research, PMLR, 2020, pp. 1597–1607. URL:https://proceedings.mlr.press/v119/chen20j.html..

[9] G.M. van de Ven, H.T. Siegelmann, A.S. Tolias, Brain-inspired replay for continual learning with artificial neural networks, Nature Communications 11 (2020) 4069, https://doi.org/10.1038/s41467-020-17866-2.

[10] I.A. Lungu, S.-C. Liu, T. Delbruck, Incremental learning of hand symbols using event-based cameras, IEEE Journal on Emerging and Selected Topics in Circuits and Systems 9 (2019) 690–696, https://doi.org/10.1109/JETCAS.2019.2951062.

[11] G.I. Parisi, J. Tani, C. Weber, S. Wermter, Lifelong learning of spatiotemporal representations with dual-memory recurrent self-organization, Frontiers in Neurorobotics 12 (2018) 78, https://doi.org/10.3389/fnbot.2018.00078.

[12] F. Sun, C. Ho, H. Lee, LAMOL: LAnguage MOdeling for Lifelong Language Learning, in: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26–30, 2020, OpenReview.net, 2020. URL: https://openreview.net/forum?id=Skgxcn4YDS..

[13] Y.M. Asano, C. Rupprecht, A. Vedaldi, A critical analysis of self-supervision, or what we can learn from a single image, in: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26–30, 2020, OpenReview.net, 2020. URL:https://openreview.net/forum?id=B1esx6EYvr..

[14] B. Graham, L. van der Maaten, Submanifold sparse convolutional networks, CoRR abs/1706.01307 (2017). arXiv:1706.01307..

[15] B. Graham, M. Engelcke, L. van der Maaten, 3D semantic segmentation with submanifold sparse convolutional networks, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 9224–9232. doi: 10.1109/CVPR.2018.00961..

[16] J.C. Stanley, Computer simulation of a model of habituation, Nature 261 (1976) 146–148, https://doi.org/10.1038/261146a0.

[17] D. Neil, M. Pfeiffer, S.-C. Liu, L.S.T.M. Phased, Accelerating recurrent network training for long or event-based sequences, in: Proceedings of the 30th International Conference on Neural Information Processing Systems, Curran Associates Inc., Red Hook, NY, USA, 2016, pp. 3889–3897, URL:https://dl.acm.org/doi/pdf/10.5555/3157382.3157532.

[18] G. Gallego, T. Delbrück, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A.J. Davison, J. Conradt, K. Daniilidis, D. Scaramuzza, Event-based vision: A survey, IEEE Transactions on Pattern Analysis and Machine Intelligence 44 (2022) 154–180, https://doi.org/10.1109/TPAMI.2020.3008413.

[19] S. Hochreiter, J. Schmidhuber, Long Short-Term Memory, Neural Computation 9 (1997) 1735–1780, https://doi.org/10.1162/neco.1997.9.8.1735.

[20] A.I. Maqueda, A. Loquercio, G. Gallego, N. García, D. Scaramuzza, Event-based vision meets deep learning on steering prediction for self-driving cars, in: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18–22, 2018, IEEE Computer Society, 2018, pp. 5419–5427. doi: 10.1109/CVPR.2018.00568..

[21] F. Zenke, B. Poole, S. Ganguli, Continual learning through synaptic intelligence, in: D. Precup, Y.W. Teh (Eds.), Proceedings of the 34th International Conference on Machine Learning, volume 70 of Proceedings of Machine Learning Research, PMLR, 2017, pp. 3987–3995, URL:https://proceedings.mlr.press/v70/zenke17a.html.

[22] H. Shin, J.K. Lee, J. Kim, J. Kim, Continual learning with deep generative replay, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, Curran Associates Inc., Red Hook, NY, USA, 2017, pp. 2994–3003, URL:https://dl.acm.org/doi/10.5555/3294996.3295059.

[23] G. Rothschild, E. Eban, L.M. Frank, A cortical-hippocampal-cortical loop of information processing during memory consolidation, Nature Neuroscience 20 (2017) 251–259, https://doi.org/10.1038/nn.4457.

[24] H.F. Ólafsdóttir, D. Bush, C. Barry, The role of hippocampal replay in memory and planning, Current Biology 28 (2018) R37–R50, https://doi.org/10.1016/j.cub.2017.10.073.

[25] D.P. Kingma, M. Welling, Auto-encoding variational Bayes, in: 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14–16, 2014, Conference Track Proceedings, 2014. URL:http://arxiv.org/abs/1312.6114..

[26] N.Y. Masse, G.D. Grant, D.J. Freedman, Alleviating catastrophic forgetting using context-dependent gating and synaptic stabilization, Proc. Natl. Acad. Sci. USA 115 (2018) E10467–E10475, https://doi.org/10.1073/pnas.1803839115.

[27] D. van Moorselaar, H. Slagter, Inhibition in selective attention, Annals of the New York Academy of Sciences 1464 (2020) 204–221, https://doi.org/10.1111/nyas.14304.

[28] K.V. Kuchibhotla, J.V. Gill, G.W. Lindsay, E.S. Papadoyannis, R.E. Field, T.A.H. Sten, K.D. Miller, R.C. Froemke, Parallel processing by cortical inhibition enables context-dependent behavior, Nature neuroscience 20 (2017) 62–71, https://doi.org/10.1038/nn.4436.

[29] D. Gandolfi, A. Bigiani, C.A. Porro, J. Mapelli, Inhibitory plasticity: From molecules to computation and beyond, International Journal of Molecular Sciences 21 (2020), https://doi.org/10.3390/ijms21051805.

[30] C.N. Levelt, M. Hübener, Critical-period plasticity in the visual cortex, Annual Review of Neuroscience 35 (2012) 309–330, https://doi.org/10.1146/annurev-neuro-061010-113813, pMID: 22462544.

[31] S. Marsland, J. Shapiro, U. Nehmzow, A self-organising network that grows when required, Neural Netw. 15 (2002) 1041–1058, https://doi.org/10.1016/S0893-6080(02)00078-3.

[32] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors, Nature 323 (1986) 533–536, https://doi.org/10.1038/323533a0.

[33] G. Orchard, A. Jayawant, G.K. Cohen, N. Thakor, Converting static image datasets to spiking neuromorphic datasets using saccades, Frontiers in Neuroscience 9 (2015) 437, https://doi.org/10.3389/fnins.2015.00437.

[34] R. Li Fei-Fei, P. Perona Fergus, Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories, in: 2004 Conference on Computer Vision and Pattern Recognition Workshop, 2004, p. 178, https://doi.org/10.1109/CVPR.2004.383.

[35] N. Messikommer, D. Gehrig, A. Loquercio, D. Scaramuzza, Event-based asynchronous sparse convolutional networks, in: Computer Vision - ECCV 2020–16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII, volume 12353 of Lecture Notes in Computer Science, Springer, 2020, pp. 415–431. doi: 10.1007/978-3-030-58598-3_25..

[36] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (1998) 2278–2324, https://doi.org/10.1109/5.726791.

[37] D. Kostadinov, D. Scaramuzza, Unsupervised feature learning for event data: Direct vs inverse problem formulation, in: 25th International Conference on Pattern Recognition, ICPR 2020, Virtual Event/ Milan, Italy, January 10–15, 2021, IEEE, 2020, pp. 5981–5987. doi: 10.1109/ICPR48806.2021.9412631..

[38] F. Paredes-Vallés, K.Y.W. Scheper, G.C.H.E. de Croon, Unsupervised learning of a hierarchical spiking neural network for optical flow estimation: From events to global motion perception, IEEE Trans. Pattern Anal. Mach. Intell. 42 (2020) 2051–2064, https://doi.org/10.1109/TPAMI.2019.2903179.

[39] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, C.H. Lampert, iCaRL: Incremental classifier and representation learning, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 5533–5542. doi: 10.1109/CVPR.2017.587..

[40] J.M. Allred, K. Roy, Controlled forgetting: Targeted stimulation and dopaminergic plasticity modulation for unsupervised lifelong learning in spiking neural networks, Frontiers in Neuroscience 14 (2020) 7, https://doi.org/10.3389/fnins.2020.00007.

[41] V. Gryshchuk, C. Weber, C.K. Loo, S. Wermter, Lifelong learning from event-based data, in: ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6–8 October 2021, i6doc.com publ., 2021. doi: 10.14428/esann/2021.ES2021-146..

Vadym Gryshchuk received his BSc and MSc in Computer Science at the University of Hamburg. His studies and research focused on neural-based knowledge processing and artificial intelligence. Currently, he is pursuing his PhD. His main research interests include machine learning, cognitive systems, neural networks, lifelong machine learning, brain-inspired computing, and intelligent assistive technologies.

Cornelius Weber graduated in physics, Bielefeld, Germany, and received his PhD in computer science at the Technische Universität Berlin. Following positions were a postdoctoral fellow in Brain and Cognitive Sciences, University of Rochester, USA; research scientist in Hybrid Intelligent Systems, University of Sunderland, UK; Junior Fellow at the Frankfurt Institute for Advanced Studies, Germany. Currently he is Lab Manager at Knowledge Technology, University of Hamburg. His interests are in computational neuroscience, development of visual feature detectors, neural models of representations and transformations, reinforcement learning and robot control, grounded language learning, human-robot interaction, and related applications in social assistive robotics.

Chu-Kiong Loo is Full Professor at the Department of Artificial Intelligence, Faculty of Computer Science & Information Technology, Universiti Malaya, Malaysia. His major interests are in continual lifelong learning, explainable AI, brain-inspired machine intelligence for robotics, and hybrid intelligence for healthcare. He has been a research fellow under the Georg Forster Fellowship award, Germany and JSPS fellowship award, Japan.

Stefan Wermter is Full Professor at the University of Hamburg, Germany, and Director of the Knowledge Technology Institute in the Dept. of Informatics. He has previously held positions at the University of Dortmund, University of Massachusetts, the International Computer Science Institute in Berkeley and the University of Sunderland. His main research interests are in neural networks, hybrid knowledge technology, neuroscience-inspired computing, cognitive robotics, natural language processing and human-robot interaction. He is co-coordinator of the international collaborative research centre on Crossmodal Learning (TRR-169) and he currently serves as the President of the European Neural Network Society.