sciendo

JAISCR, 2026, Vol. 16, No. 3, pp. 293 – 310
10.2478/jaiscr-2026-0015

SATSOM: SATURATION SELF-ORGANIZING MAPS FOR CONTINUAL LEARNING

Igor Urbanik*, Paweł Gajewski

Faculty of Computer Science, AGH University of Krakow
al. Adama Mickiewicza 30, Kraków, 30-059, Poland

**E-mail: iurbanik@student.agh.edu.pl*

Submitted: 24th September 2025; Accepted: 28th January 2026

Abstract

Continual learning poses a fundamental challenge for neural systems, which typically suffer from catastrophic forgetting when exposed to sequential tasks. Self-Organizing Maps (SOMs), despite their inherent interpretability and efficiency, also exhibit this vulnerability. In this paper, we introduce Saturation Self-Organizing Maps (SatSOM)—an extension designed to enhance knowledge retention in continual learning scenarios. SatSOM incorporates a novel saturation mechanism that progressively reduces the learning rate and neighborhood radius of neurons as they accumulate information. This dynamic effectively stabilizes well-trained neurons, redirecting new learning to underutilized regions of the map. To further accommodate tasks of unknown complexity, we introduce a dynamic variant capable of adaptive grid expansion. We evaluate SatSOM on sequential versions of the FashionMNIST and KMNIST datasets, showing that it significantly outperforms existing SOM-based methods and approaches the retention capabilities of a k-nearest neighbors (kNN) baseline. Ablation studies confirm the critical role of the saturation mechanism. SatSOM offers a lightweight and interpretable solution for sequential learning and provides a foundation for implementing adaptive plasticity in complex architectures.

Keywords: incremental learning, continual learning, catastrophic forgetting, self-organizing maps

1 Introduction

Intelligent agents functioning in real-world environments must continuously learn, adapting to new information while retaining prior knowledge [13]. This ability, known as continual or lifelong learning, poses a significant challenge in modern machine learning. Most artificial neural systems struggle with catastrophic forgetting [6], where training on new tasks or data distributions abruptly erases previously learned information. This phenomenon stems from the shared nature of representations in standard neural networks, where updating weights for new data can overwrite information critical for past tasks.

To quantify the extent of forgetting, we consider the k-Nearest Neighbors (kNN) algorithm as a useful reference point. Because kNN stores the entire history of training data and performs inference via direct comparison, it avoids forgetting by design. However, this retention comes at the cost of unbounded memory requirements and computational inefficiency during inference, making it im-

practical for scalable, real-world applications. In contrast, standard parametric models operate with fixed memory and computational budgets but generally fail to match the retention stability of kNN.

Self-Organizing Maps (SOMs) [11] present a compelling architecture that lies between these two extremes. As unsupervised models that project high-dimensional data onto a low-dimensional grid, SOMs rely on a competitive learning mechanism. Unlike fully connected networks where updates are global, SOMs update only the Best Matching Unit (BMU) and its immediate topological neighbors. This locality offers an intrinsic degree of resilience against catastrophic forgetting, as new information does not necessarily overwrite distal regions of the map.

However, this inherent resilience is insufficient for robust continual learning. As we demonstrate in this study, standard SOMs eventually succumb to forgetting when the data distribution shifts significantly; neurons previously specialized for old tasks are gradually pulled toward new data clusters, eroding prior representations. This limitation motivates the need for a mechanism that leverages the SOM's topological advantages while explicitly preventing the degradation of well-established knowledge.

In this work, we propose the Saturation Self-Organizing Map (SatSOM), a novel extension of Self-Organizing Maps designed to enhance knowledge retention while directing learning signals toward underutilized model components. Building upon prior efforts to adapt SOMs for continual learning, such as the PROPRE framework [8], which demonstrated the potential of SOMs for knowledge retention, SatSOM introduces a saturation mechanism that tracks the "maturity" of individual neurons. This mechanism is based on a novel *saturation* metric (conceptually distinct from traditional activation saturation in deep neural networks) that quantifies a neuron's cumulative exposure to relevant training data. Once a neuron becomes saturated, its weight updates are restricted, effectively freezing critical learned patterns while allowing less-utilized regions of the map to remain plastic. By selectively preserving learned representations in this manner, SatSOM addresses the stability-plasticity dilemma central to catastrophic forgetting and achieves robust knowledge retention in dynamic environments without the infinite memory costs associated with kNN methods. A high-level overview of the inference process is provided in Figure 1.

The primary contribution of this work is the demonstration that this saturation-based freezing mechanism significantly enhances knowledge retention. By explicitly preventing the overwriting of well-established representations, SatSOM achieves robustness superior to standard SOMs in dynamic environments. Furthermore, the principles underlying this mechanism provide a generalizable framework for stability. We posit that similar saturation-based constraints could be adapted for other neural architectures, potentially extending these benefits to more complex deep learning models.

We evaluate SatSOM through a rigorous experimental protocol involving two distinct sets of benchmarks. First, we provide a standard comparison against state-of-the-art approaches on image classification tasks. Second, we introduce a custom benchmark specifically designed for class-incremental learning across MNIST variants without data replay. This strict evaluation setting allows us to isolate model behavior and identify specific limitations in existing methods. Our findings suggest that the saturation mechanism offers a practical, interpretable, and effective solution for continual learning.

To facilitate reproducibility and further research, all model architectures and testing code used in this study are available in a public GitHub repository[1].

2 Related Work

A comprehensive survey by Delange et al. [3] details the landscape of continual learning, categorizing approaches into three primary families: regularization-based, rehearsal-based, and architectural or hybrid methods.

Regularization-based methods mitigate catastrophic forgetting by constraining weight updates to preserve knowledge from previous tasks. A representative example is Elastic Weight Consolidation (EWC) [10], which penalizes changes to parameters deemed important for prior tasks. Conversely,

[1] https://github.com/Radinyn/satsom

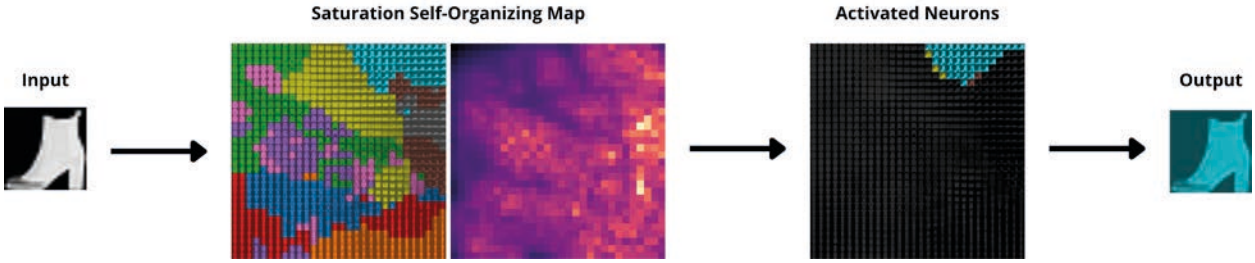

Figure 1. Symbolic visualization of SatSOM inference on the FashionMNIST dataset. An input image is processed to calculate activation levels for each neuron based on similarity. Label values associated with activated neurons are then aggregated to compute the final class probabilities.

rehearsal-based methods (or replay methods) retain a subset of past data to interleave with current training samples. For instance, iCaRL [17] combines a nearest-mean-of-exemplars classification rule with knowledge distillation to maintain feature representations in a class-incremental setting. Other efficient replay strategies include those proposed by Hayes et al. [9].

The third category, architectural methods, mitigates forgetting by dynamically modifying the network structure or isolating task-specific parameters. Within this domain, Self-Organizing Maps (SOMs) have emerged as a potential tool for continual learning. Gepperth and Karaoguz [8] demonstrated that SOMs can achieve competitive performance in continual learning scenarios, serving as a foundational inspiration for our approach. Further explorations into energy-based SOM training have also been conducted by Gepperth [7].

Several hybrid architectures have since integrated SOMs to enhance plasticity and stability. SOMPL, proposed by Bashivan et al. [1], runs a SOM layer in parallel with a standard supervised network. The SOM clusters inputs into task contexts to generate an output mask, selectively routing inputs to relevant network modules without requiring explicit task labels. Similarly, DendSOM [14] draws inspiration from biological dendrites, employing a layer of multiple SOMs where each acts as a dendrite modeling a specific subregion of the input space.

Beyond the aforementioned categories, a complementary line of work focuses on detecting distributional shifts or concept drift in streaming data. A recent example is the approach proposed by Prowik et al. [15], which detects drift based on measurements of changes in feature ranks. Such drift detection signals can subsequently be employed to adapt models online, similarly to Hoeffding Trees [5].

To address the challenges of online, task-free learning, Pourcel et al. [16] introduced Dynamic Sparse Distributed Memory (DSDM). This method utilizes a semi-distributed associative memory composed of partially overlapping clusters. DSDM dynamically evolves to model non-stationary data streams and employs a local density-based pruning technique to manage memory constraints effectively.

Most relevant to our work is the Continual SOM (CSOM) introduced by Vaidya et al. [18], which generalizes the SOM for online, task-free settings. CSOM embeds each neuron with a running variance and utilizes neuron-specific learning parameters. While SatSOM shares conceptual similarities with CSOM, it introduces several distinct features. First, unlike CSOM which maintains a running variance, SatSOM employs a standard Euclidean distance while achieving comparable performance. Second, SatSOM introduces a quantile threshold parameter to select only the most relevant neurons, thereby improving stability. Third, the streamlined architecture facilitates comprehensive ablation studies to identify key factors for future research, such as adapting the saturation concept to other models. Finally, we present a direct comparison with a kNN baseline, demonstrating that SatSOM outperforms both the baseline and CSOM on the selected benchmark.

3 Methodology

The *Saturation Self-Organizing Map (SatSOM)* is founded on the principle that catastrophic forgetting can be mitigated by directing new learning to-

ward underutilized components of the model, while preserving the stability of representations that are already significant for prediction. This approach is naturally facilitated by the topological grid structure of the SOM.

Implementation is achieved by assigning each neuron a specific learning rate and neighborhood radius. These parameters decay over time based on the cumulative magnitude of updates applied to the neuron, effectively stabilizing ("freezing") it once a specific threshold is reached. We define this state as *saturation*—quantified here as the normalized difference between the initial and current learning rate, though it could explicitly be defined via the neighborhood radius as well.

SatSOM optimizes a set of prototypes (neuron weights) and associated label-prototypes. During inference, the model identifies the prototypes most similar to the input using a predefined distance metric (specifically, Euclidean distance) and aggregates their corresponding label-prototypes to generate a weighted output. This process is illustrated in Figure 2.

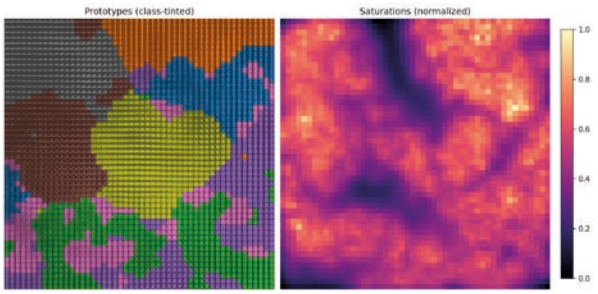

Figure 2. SatSOM prototypes trained on FashionMNIST. Neuron weights are visualized as images and color-coded based on the class with the highest probability in their respective label vector.

Our intuition is that the neighborhood radius decay guides new knowledge to the emptier parts of the map, while preserving the locality of changes. In contrast, learning rate decay works by preserving the more important neurons when no more space can be found. This view is further reinforced by the results of our ablation study (see Section 5).

3.1 Definitions

For simplicity, we assume that each SatSOM is a square map with a side length of n neurons. We denote their total by $N = n^2$.

Each neuron i has its own prototype vector w_i and prototype-label vector ℓ_i. We denote its coordinates within the grid as g_i. To facilitate the saturation mechanism, two additional parameters are stored, the learning rate λ_i and the neighborhood radius σ_i. We provide an intuitive visualization in Figure 3.

The model incorporates global hyperparameters λ_0 and σ_0 to initialize the learning rate and neighborhood radius of each neuron. These parameters decay exponentially during training at rates determined by α_λ and α_σ (Equation 8), mirroring the decay schedule of traditional SOMs [11].

For practical purposes, there are also two inference-time hyperparameters, p and q. Our empirical evaluation across a range of configurations indicates that the model is largely insensitive to reasonable choices of these parameters. They are necessary for the model to work on unbalanced datasets, as described further.

Although SatSOM has many hyperparameters, in this work we show that specific combinations of those parameters are effective for most tasks.

Finally, we also define saturation of a neuron as:

$$s_i = \frac{\lambda_0 - \lambda_i}{\lambda_0}, \quad i = 1, \ldots, N. \tag{1}$$

This measure provides a compact indicator of how much the adaptive capacity of a neuron has diminished during training. By expressing the decay of the learning rate in normalized form, the saturation quantifies the extent to which a neuron has stabilized.

3.2 Training

The training procedure begins by presenting a single example x together with its one-hot-encoded label y. First, the Best-Matching Unit (BMU) is found by computing Euclidean distances

$$d_i = \|x - w_i\|_2, \tag{2}$$

and selecting its index:

$$b = \arg\min_i d_i. \tag{3}$$

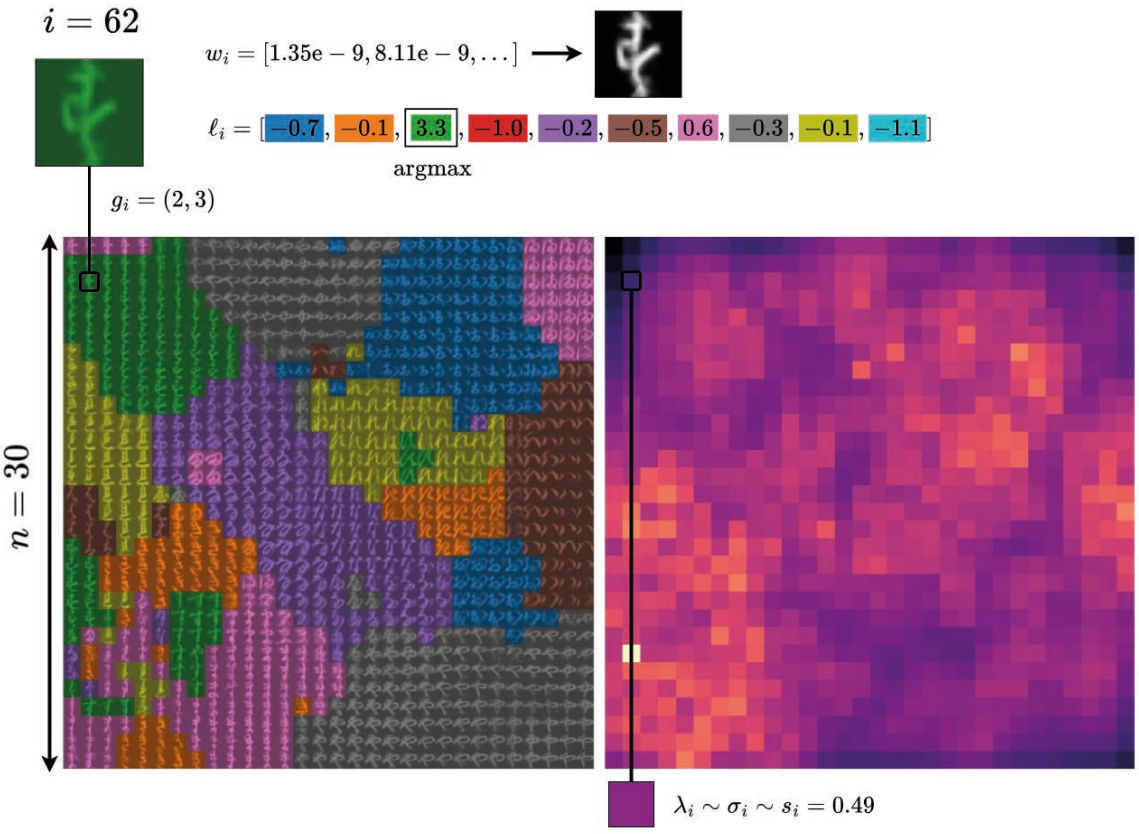

Figure 3. Visualization of a SatSOM trained on the KMNIST dataset, annotated with key mathematical elements.

Around this BMU, each neuron i acquires a neighborhood strength, based on the distance between its and BMU coordinates:

$$\theta_i = \exp\left(-\frac{\|g_i - g_b\|^2}{2\sigma_b \sigma_i}\right). \tag{4}$$

Each prototype w_i is then nudged toward the sample by

$$w_i \leftarrow w_i + \lambda_i \theta_i (x - w_i). \tag{5}$$

Concurrently, the label matrix is updated by first calculating the class probabilities (softmax):

$$p_{i,c} = \frac{\exp(\ell_{i,c})}{\sum_{c'} \exp(\ell_{i,c'})}, \tag{6}$$

where c denotes class index, computing the gradient $\nabla \ell_i = p_i - \ell_i$, and applying

$$\ell_i \leftarrow \ell_i - \lambda_i \theta_i \nabla \ell_i. \tag{7}$$

This is equivalent to using cross-entropy loss. Afterwards, both the per-neuron learning rate and radius decay multiplicatively according to the neighborhood strength:

$$\lambda_i \leftarrow \lambda_i \exp(-\alpha_\lambda \theta_i), \qquad \sigma_i \leftarrow \sigma_i \exp(-\alpha_\sigma \theta_i). \tag{8}$$

3.3 Inference

The model makes a prediction by creating a weighted average of select label-prototypes. A symbolic visualization is shown in Figure 1.

We start with normalizing the distance ($\varepsilon > 0$, a small constant added for numerical stability):

$$\tilde{d}_i = \frac{d_i - \min_j d_j}{\max_j d_j - \min_j d_j + \varepsilon}. \tag{9}$$

Next, we disable under-saturated neurons, i.e. those that do not yet store any information:

$$\tilde{d}_i \leftarrow 1 \quad \text{if} \quad s_i < \varepsilon. \tag{10}$$

We compute a cut-off quantile threshold among the enabled neurons such that only the fraction q of neurons closest to the input is used in the prediction:

$$\tau = \text{quantile}\left(\{\tilde{d}_j : s_j \geq \varepsilon\}, q\right), \tag{11}$$

and clamp any $\tilde{d}_i$ exceeding τ to unity:

$$\tilde{d}_i \leftarrow 1 \quad \text{if } \tilde{d}_i > \tau. \tag{12}$$

In effect, this procedure is analogous to selecting the nearest neighbors in a kNN classifier, as it prioritizes neurons in the local vicinity of the best-matching unit while mitigating the influence of more prevalent classes that constitute larger portions of the network, making the method robust on unbalanced datasets.

Finally, distances are converted into neuron *proximities* via the exponent (p), sharpening the fall-off:

$$h_i = \left(1 - \tilde{d}_i\right)^p. \tag{13}$$

We use the label matrix to compute the final prediction, weighted by proximity:

$$\hat{y} = \frac{1}{N} \sum_{i=1}^{N} h_i \ell_i. \tag{14}$$

4 Evaluation

The primary objective of our experimental analysis is to rigorously assess the knowledge retention capabilities of SatSOM in strictly online, class-incremental settings. We adopt a two-tiered evaluation strategy: first, we benchmark the model on raw pixel data using standard datasets (FashionMNIST, KMNIST) to validate its fundamental learning dynamics. Second, we extend the evaluation to complex, high-dimensional feature spaces (CIFAR-10, CIFAR-100, CORe50) using pre-trained embeddings, isolating the model's ability to manage stability-plasticity trade-offs when feature extraction is handled externally.

4.1 Metrics

We employ standard continual learning metrics to quantify performance. Given a prediction function $f : \mathcal{X} \to \mathcal{Y}$ and a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, the average accuracy is:

$$\text{ACC}(f, \mathcal{D}) = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}(f(x_i) = y_i), \tag{15}$$

where $\mathbf{1}(\cdot)$ is the indicator function.

We report the Average Accuracy (ACC) over all classes at the end of the entire training sequence.

Let the scenario consist of tasks $\{T_1, \ldots, T_K\}$. We denote by f_k the model after training on task T_k.

To measure the stability of the most recently acquired knowledge, we report Last Accuracy (LA),

defined as the performance on the final task immediately after learning it

$$\mathrm{LA} = \mathrm{ACC}(f_K, T_K). \tag{16}$$

We quantify forgetting using the Forgetting Measure which quantifies the average degradation in performance on previously learned tasks after completing all tasks. Let $A_k^{\max} = \max_{j\in\{k,\ldots,K\}} \mathrm{ACC}(f_j, T_k)$ be the best accuracy achieved on task T_k at any point during training. Then the forgetting measure is

$$\mathrm{FM} = \frac{1}{K-1}\sum_{k=1}^{K-1}\left(A_k^{\max} - \mathrm{ACC}(f_K, T_k)\right). \tag{17}$$

Positive values indicate forgetting; zero indicates no forgetting.

Additionally, Backward Transfer measures the influence of new learning on old tasks. Using the same notation as above, BWT is defined as

$$\mathrm{BWT} = \frac{1}{K-1}\sum_{k=1}^{K-1}\left(\mathrm{ACC}(f_K, T_k) - \mathrm{ACC}(f_k, T_k)\right). \tag{18}$$

Positive values indicate improvement of old tasks and negative values indicate interference.

4.2 Direct Evaluation

We first evaluate SatSOM in a challenging class-incremental setting where the model must learn directly from raw pixel intensities. We utilize two 10-class benchmarks: FashionMNIST [19] and KuzushijiMNIST (KMNIST) [2], using a 70-30 train-validation split. Training is organized into distinct phases, where each phase presents samples from a single class exclusively. Once a phase concludes, that class is never revisited. This protocol—single-pass, online, class-incremental learning—represents a severe test of catastrophic forgetting.

We benchmark against two baselines: a deterministic kNN ($k = 5$), which serves as an upper bound for retention (due to its full memory of the dataset), and the DSDM architecture [16]. Notably, all models are evaluated in a strict single-pass setting. The hyperparameters are detailed in Table 1.

The performance trajectory across all 10 phases is illustrated in Figure 4, with class-specific breakdowns in Figure 5. Quantitative comparisons are summarized in Table 2.

Table 1. Hyperparameters used in the experimental evaluation.

kNN	SatSOM	DSDM	Value
k	–	–	5
–	n	–	100
–	λ_0	–	0.5
–	σ_0	–	10
–	α_λ	–	0.01
–	α_σ	–	0.2
–	p	–	10
–	q	–	0.001
–	–	β	2
–	–	λ	0.01

SatSOM demonstrates robust long-term retention, exhibiting a stability profile closer to the kNN baseline than to the plastic DSDM architecture. It outperforms DSDM significantly on both datasets. Crucially, SatSOM achieves this without requiring task identifiers or boundaries, highlighting its versatility for autonomous scenarios.

To visualize the internal dynamics, Figures 6 and 7 display the evolution of prototypes and neuron saturation during a KMNIST run. The relatively low saturation levels suggest that the network retains sufficient plasticity to accommodate future classes if needed, a balance controlled by the choice of hyperparameters.

Table 2 compares SatSOM against other SOM-based methods. SatSOM achieves the highest accuracy among alike approaches. While DendSOM records favorable FM and BWT scores, this is an artifact of its low baseline accuracy (FM and BWT are relative metrics; a model that learns little has little to forget). SatSOM, conversely, maintains high absolute accuracy with minimal forgetting.

4.3 Evaluation with Pre-trained Backbone

To rigorously test the scalability of SatSOM, we extend our evaluation to Latent Rehearsal settings. Following protocols similar to DSDM [16], we freeze a ResNet50 backbone pre-trained on ImageNet and perform continual learning solely on the generated embeddings. While this simplifies the feature extraction burden, it isolates the plasticity-stability trade-off of the classification mechanism

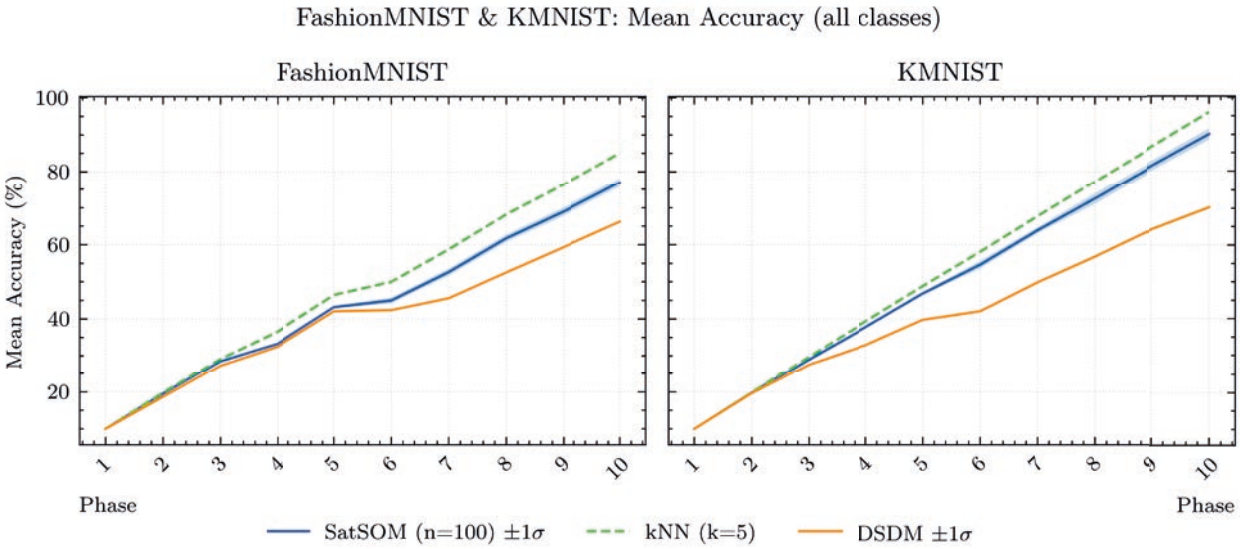

Figure 4. Evolution of mean accuracy across all observed classes throughout the 10 training phases.

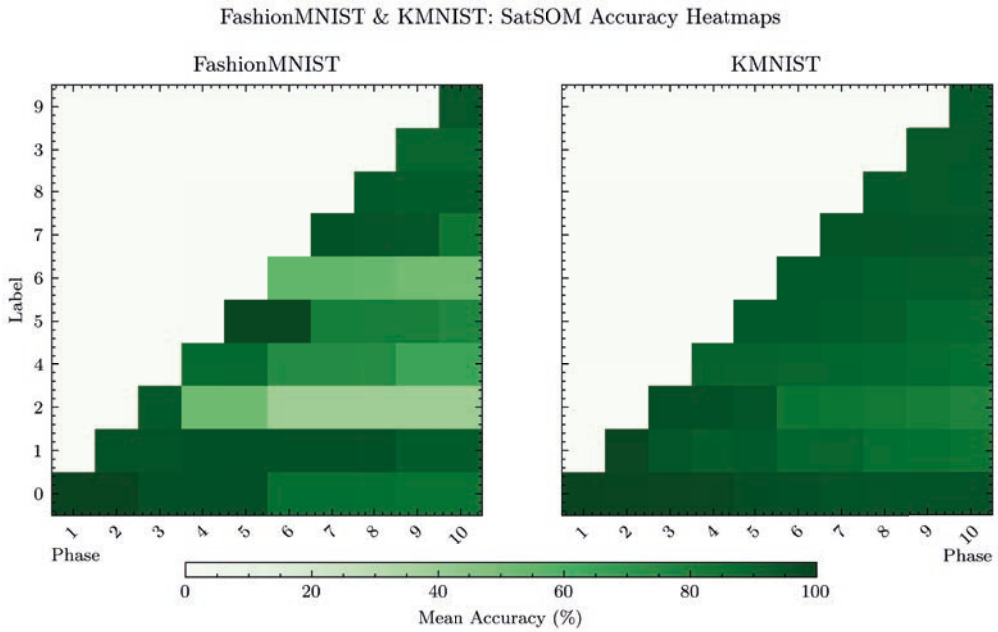

Figure 5. Heatmap of mean SatSOM accuracy per class. Performance fluctuations are likely attributable to inter-class visual similarities typical in these datasets.

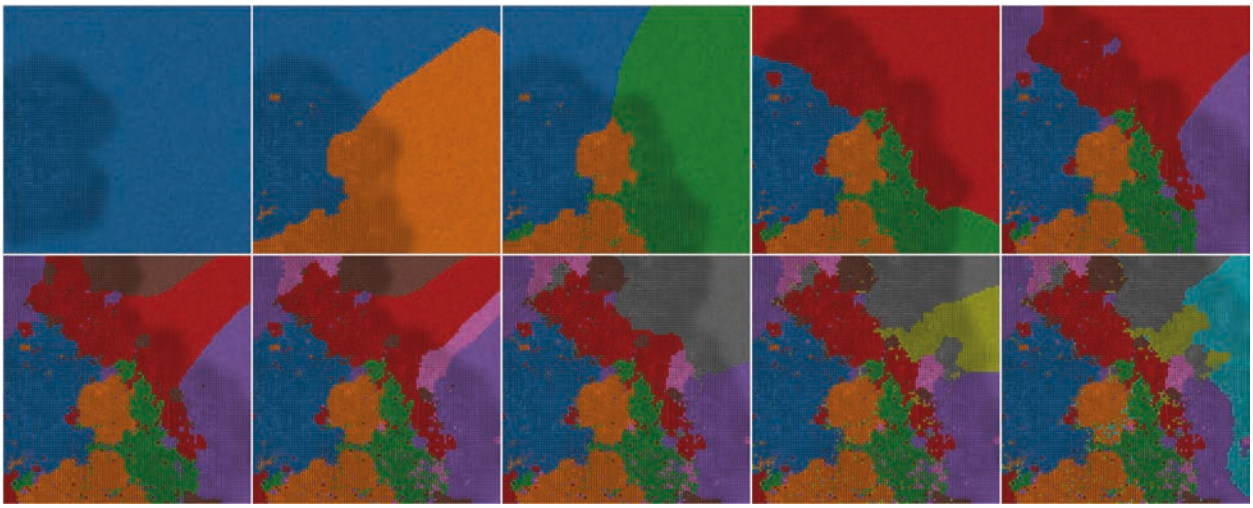

Figure 6. Evolution of prototypes during a single training pass on the KMNIST dataset (ordered left to right, top to bottom).

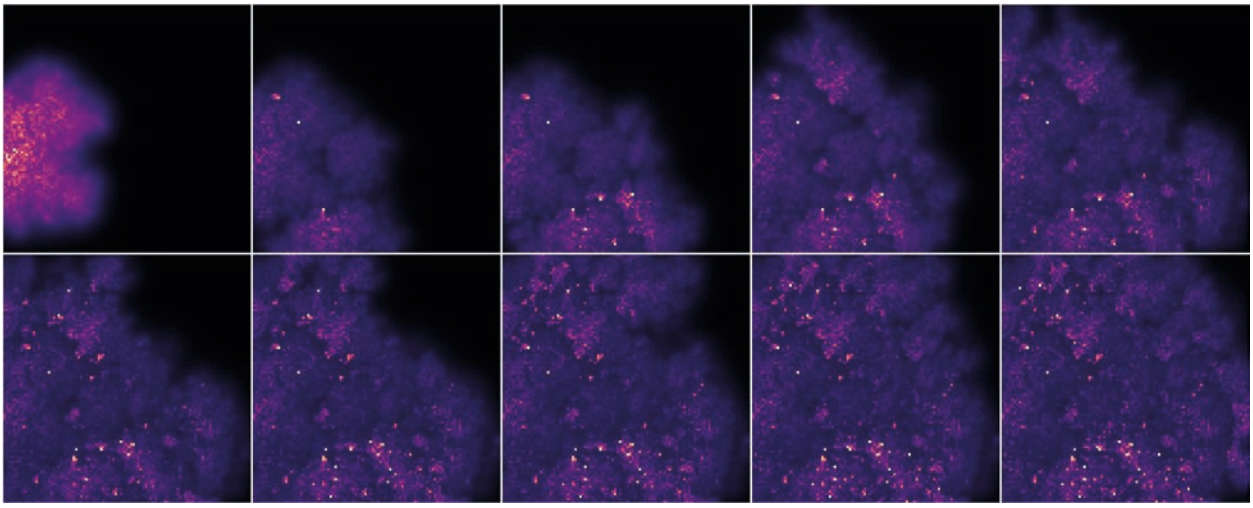

Figure 7. Evolution of neuron saturation corresponding to the run in Figure 6..

Table 2. Comparison of SatSOM and other selected continual learning methods. Reference values for vanilla SOM, DendSOM, and CSOM are sourced from [18].

Dataset	Method	ACC (↑)	LA (↑)	FM (↓)	BWT (↑)
FashionMNIST (step 1)	kNN	85.03	91.77	7.49	-7.49
	DSDM	66.25 ± 0.28	80.37 ± 0.36	15.70 ± 0.32	15.69 ± 0.33
	vanilla SOM	32.07 ± 0.03	57.52 ± 0.03	37.01 ± 0.03	-28.27 ± 0.02
	DendSOM	16.62 ± 0.03	27.36 ± 0.03	$\mathbf{11.34 \pm 0.03}$	$\mathbf{-11.92 \pm 0.03}$
	CSOM	75.13 ± 2.68	88.23 ± 0.56	13.11 ± 2.93	-14.56 ± 3.25
	SatSOM	$\mathbf{76.98 \pm 1.15}$	$\mathbf{90.30 \pm 0.54}$	14.93 ± 1.67	-14.79 ± 1.66
KMNIST (step 1)	kNN	96.05	97.65	1.79	-1.79
	DSDM	70.26 ± 0.56	73.49 ± 0.61	3.58 ± 0.37	-3.58 ± 0.37
	vanilla SOM	23.65 ± 0.03	54.00 ± 0.05	35.28 ± 0.01	-33.79 ± 0.02
	DendSOM	10.95 ± 0.01	12.92 ± 0.01	$\mathbf{2.02 \pm 0.00}$	$\mathbf{-2.18 \pm 0.01}$
	CSOM	81.60 ± 2.82	88.61 ± 1.60	7.04 ± 1.54	-7.78 ± 1.73
	SatSOM	$\mathbf{90.06 \pm 1.55}$	$\mathbf{94.74 \pm 0.62}$	5.42 ± 1.64	-5.20 ± 1.67

itself. It should be noted that this setting inherently favors methods relying on static similarity (like kNN) because the feature space is fixed.

For these experiments, we utilize the hyperparameters from Table 1, with the map size n increased to 250 to accommodate the higher complexity of the datasets.

We report results on CIFAR-10 and CIFAR-100 in class-incremental settings with varying step sizes (Table 3). SatSOM demonstrates competitive performance, marginally outperforming DSDM on CIFAR-10, though it trails on the more diverse CIFAR-100. Additionally, on the CORe50 benchmark [12] (Table 4), tested in the Multi-Task New Classes (MT-NC) setting, SatSOM achieves respectable accuracy, although it does not yet match state-of-the-art specialized architectures.

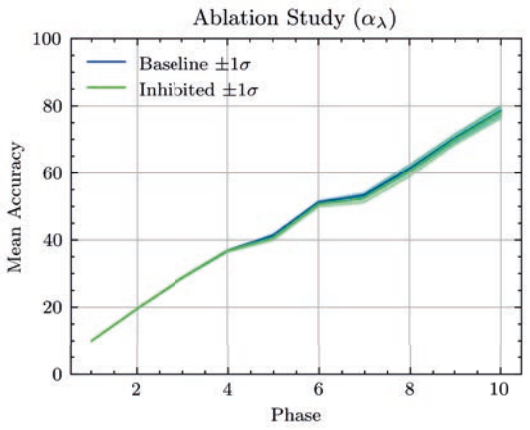

Figure 8. Mean SatSOM accuracy (FashionMNIST) comparison between the base model and the inhibited model with $\alpha_\lambda = 0$. The plot shows no impact due to the model's effectively unconstrained memory ($n = 100$).

Table 4. Results on the CORe50 benchmark.

Dataset	Method	ACC (↑)
CORe50	kNN	**72.48**
	DSDM	71.48 ± 0.26
	SatSOM	62.79 ± 2.22

Table 3. Results on CIFAR-10 and CIFAR-100 in a class-incremental setting.

Dataset	Method	ACC (↑)	LA (↑)	FM (↓)	BWT (↑)
CIFAR-10 (step 1)	kNN	**86.38**	**92.81**	7.15	−7.15
	DSDM	78.30 ± 0.42	84.13 ± 0.36	**6.48 ± 0.25**	**-6.48 ± 0.25**
	SatSOM	79.83 ± 0.48	92.57 ± 0.52	14.80 ± 0.65	−14.15 ± 0.67
CIFAR-100 (step 2)	kNN	**61.29**	**72.58**	11.52	-11.52
	DSDM	51.91 ± 0.36	61.55 ± 0.33	**9.86 ± 0.34**	**-9.84 ± 0.35**
	SatSOM	44.34 ± 0.46	61.92 ± 1.67	20.44 ± 1.50	−17.94 ± 1.55
CIFAR-100 (step 5)	kNN	**61.29**	**71.95**	11.22	-11.22
	DSDM	56.19 ± 0.58	63.75 ± 0.77	**7.96 ± 0.27**	**-7.96 ± 0.27**
	SatSOM	44.94 ± 1.13	63.03 ± 1.40	19.63 ± 1.97	−19.05 ± 2.02

5 Ablation Study

To isolate the impact of key algorithmic components, we conducted an ablation study using the FashionMNIST dataset. We replicated the experimental protocol from Section 4.2, while selectively disabling specific hyperparameters. All other settings were maintained as specified in Table 1, and each configuration was evaluated across 10 independent runs.

5.1 Learning Rate Decay

Inhibition of α_λ by setting it to zero required additional change to avoid the quantile threshold disabling all the neurons (see Equation 11). For this test we temporarily re-defined saturation (see Equation 1) on the basis of σ instead of λ:

$$s_i = \frac{\sigma_0 - \sigma_i}{\sigma_0}, \quad i = 1, \ldots, N. \tag{19}$$

We also ran this test for two map sizes, the common $n = 100$ (Figure 8) and memory constrained $n = 20$ (Figure 9). The purpose of this is to show the situation in which learning rate decay comes into play.

The effect of α_λ is noticeable only in the $n = 20$ example. This behavior showcases α_λ as an important hyperparameter preventing knowledge loss in memory constrained environments and reinforces α_σ as the main factor when SatSOM is not yet overlearned.

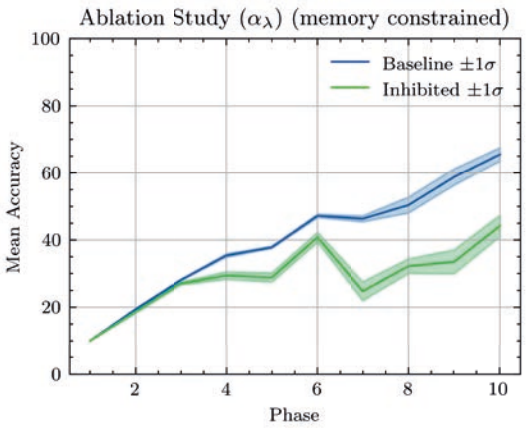

Figure 9. Mean SatSOM accuracy (FashionMNIST) comparison between the base model and the inhibited model with $\alpha_\lambda = 0$ in a memory constrained environment ($n = 20$). The effect of α_λ is clearly visible.

5.2 Neighborhood Radius Decay

In the next test, we set α_σ to 0. Although the model demonstrates some knowledge retention, it quickly loses its ability to learn new classes (Figure 10). The ones on which the model was already trained already occupy too much of the map (see Figure 11). This also shows that SatSOM's memory is strictly limited by the choice of n.

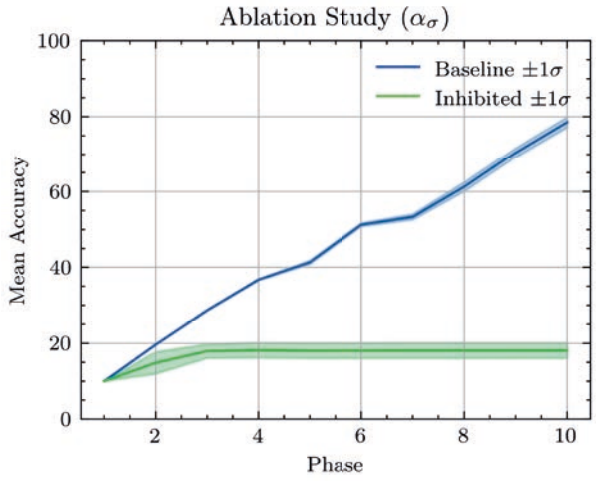

Figure 10. Mean SatSOM accuracy (FashionMNIST) comparison between the base model and the inhibited model with $\alpha_\sigma = 0$. There is a visible cut-off of accuracy in the inhibited model.

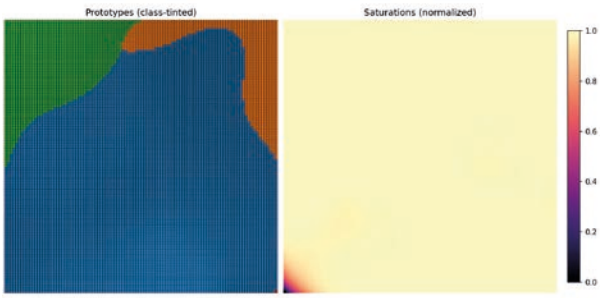

Figure 11. Example SatSOM after 10 phases with inhibited neighborhood radius decay. Only 3 classes are present.

5.3 Quantile Threshold

Inhibiting q by setting it to 1 diminishes the model performance by including many irrelevant neurons in the final label calculation (Figure 12). SatSOM output becomes more sensitive to class sizes, as their relative size on the map weighs on the output, increasing the standard deviation.

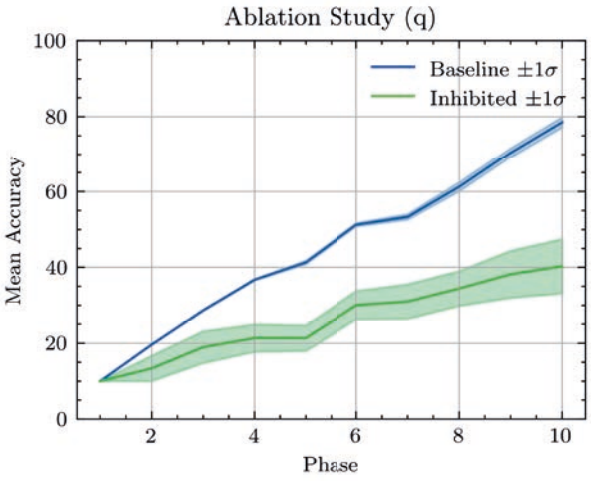

Figure 12. Mean SatSOM accuracy (FashionMNIST) comparison between the base model and the inhibited model with $q = 1$. The base model achieved superior results in all the runs.

6 Hyperparameter Analysis

We conducted a basic hyperparameter grid search during the initial prototyping phase to gain additional insights and intuitions. The exact grid of tested parameter values is presented in Table 5.

Table 5. Hyperparameter search value grid.

Parameter	Values
n	50, 100
λ_0	0.5
σ_0	$\frac{n}{2}$
α_λ	0.01
α_σ	10^{-3}, 10^{-2}, 10^{-1}, 1
p	10
q	10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}

To reduce computational cost during the hyperparameter search, we limited training to two phases. The first phase included classes 0, 1, 2, 4, 5, 6, 7, and 8 simultaneously, while the second phase comprised classes 3 and 9. SatSOM was trained 10 times for each hyperparameter configuration, and the results were subsequently aggregated. The presence of some evidently suboptimal parameter combinations led to poor performance in certain models, which affected the aggregated average accuracy and increased standard deviation. This effect is visible in the results presented in Figures 13, 15, and 14.

Table 6. Classes used in the test

Phase 1	Phase 2
0, 1, 2, 4, 5, 6, 7, 8	3, 9

As shown in Figure 13, and according to all our other tests, the grid size (n) has no observable effect on the model accuracy as long as it is large enough. It is, as expected, a primary determinant of the model's maximum capacity.

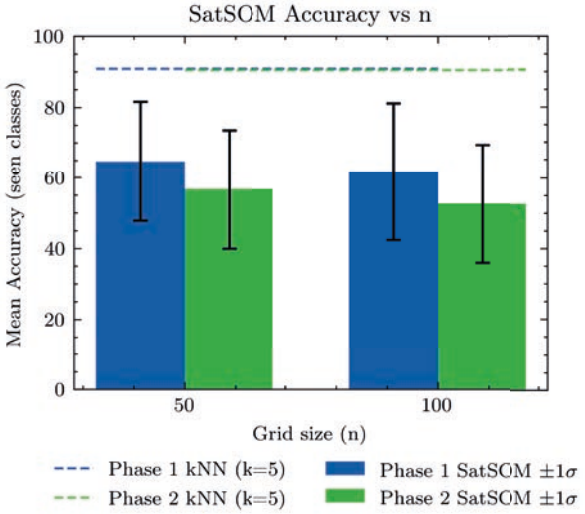

Figure 13. Mean SatSOM accuracy with respect to the n hyperparameter. There is no clear difference between the two values.

We found that the decay of the neighborhood radius (α_σ) between 0.01 and 0.1 is suitable for most applications (Figure 14). Values outside of that range tend to produce unstable results or be clearly ineffective.

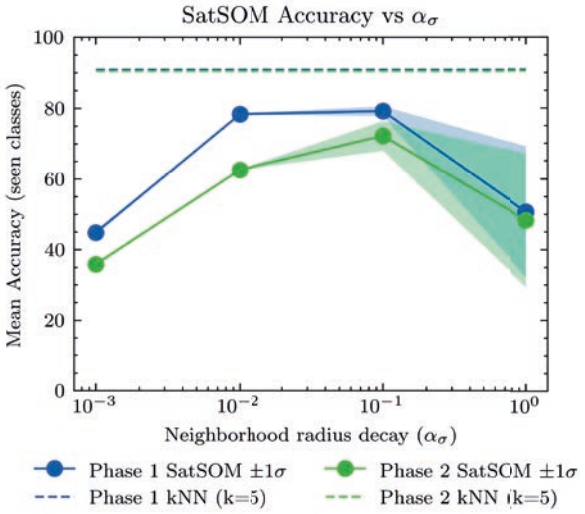

Figure 14. Mean SatSOM accuracy as a function of the α_σ hyperparameter. Values near the beginning of the range produce unsatisfactory results, while higher values have a high standard deviation.

As shown in Figure 15, most of the quantile threshold values (q) worked relatively well, with a slight advantage on values closer to 0.01.

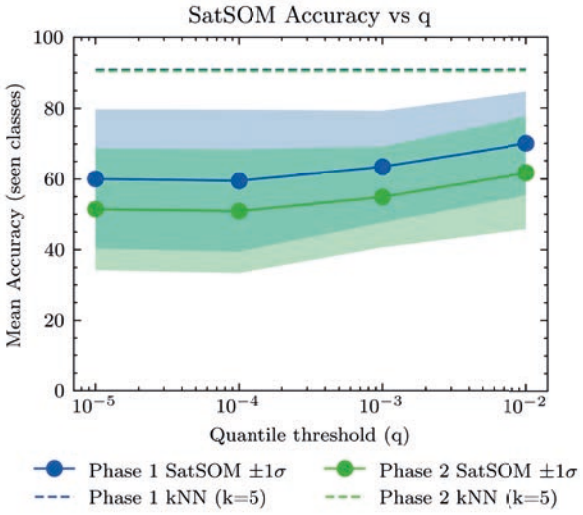

Figure 15. Mean SatSOM accuracy as a function of the q hyperparameter. All values within the tested range yield satisfactory results.

Although we do not have formal evidence, our empirical observations during testing suggest that reducing the initial neighborhood radius (σ_0) improves knowledge retention, albeit at the cost of requiring more input data. In contrast, increasing σ_0 accelerates training, but tends to reduce knowledge retention. Based on these findings, we recommend using a lower value of σ_0 when retention is a priority. Typically $\sigma_0 = \frac{n}{2}$ is used as a starting point.

7 Extension: Adaptive Capacity

As demonstrated in previous sections, the information capacity of a SatSOM is fundamentally constrained by its lattice size, N. However, the optimal model capacity for a continuous stream of tasks is rarely known *a priori*. To address this, we propose an adaptive extension of the architecture: the *Growing SatSOM*. We hypothesize that a dynamic expansion mechanism allows the model to approximate the optimal value of N relative to the task complexity. For computational simplicity, we restrict grid

expansion to discrete increments equal to the initial grid dimension, n, maintaining the two-dimensional topology. Newly added neurons are initialized randomly.

7.1 Boundary-Driven Expansion

To mitigate boundary effects, the grid expands whenever a BMU is located within the effective neighborhood radius of an edge. We define a trigger condition controlled by a scaling factor $\beta \in [0, 1]$:

$$d_{\text{edge}} \leq \max(\beta\sigma_{\text{BMU}}, 1), \tag{20}$$

where d_{edge} denotes the distance from the BMU to the grid boundary. $\beta = 1$ ensures the full neighborhood is preserved, while $\beta = 0$ limits sensitivity to the outermost neurons.

We evaluate two expansion strategies: *Symmetric*, which extends all grid dimensions uniformly, and *Directional*, which extends only the boundary closest to the BMU, allowing for non-square topologies. Figure 16 illustrates these strategies.

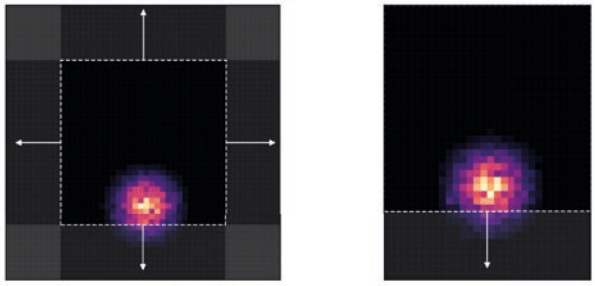

Figure 16. Comparison of Symmetric (left) and Directional (right) grid growth schemes visualized on an example saturation map.

7.2 Grid Re-centering Strategy

To optimize grid utilization and mitigate unnecessary expansion caused by spatial drift (cf. Figure 7), we introduce a re-centering mechanism. This process realigns the active region, defined as the set of neurons where $s_i > 0$, with the geometric center of the grid.

The algorithm computes the axis-aligned bounding box of the active neurons and translates the entire grid using a toroidal shift (wrap-around) to center this bounding box. Since neurons with zero saturation do not contribute to the model output (Equation 10), this transformation leaves the model functionally equivalent, with no change in classification accuracy. The operation is visualized in Figure 17.

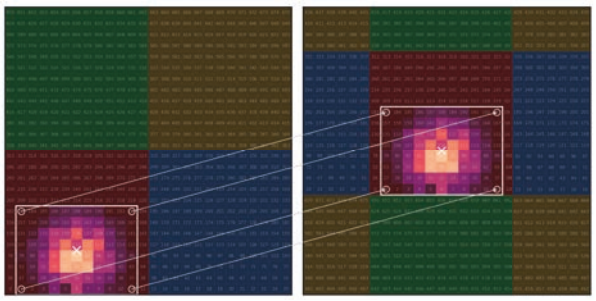

Figure 17. Visualization of the toroidal shift operation on a saturation map. Neurons are numerically indexed, and quadrants are color-coded to trace the spatial transformation.

7.3 Validation of the Dynamic Mechanism

We evaluate the dynamic extension on CIFAR-10 (ResNet18 backbone) using the hyperparameters in Table 1, modified with $n = 10$ and $\sigma_0 = 3$ to facilitate growth.

Table 7. Hyperparameters of the growing SatSOM experiment.

Parameter	Values
Scheme	Symmetric, Directional
Centering	TRUE, FALSE
β	0, 0.1, 0.2, ..., 0.9, 1
n	10
σ_0	3

Results indicate that the Directional scheme consistently produces more compact grids than the Symmetric baseline (Figure 18). Centering further reduces grid size, particularly for the Symmetric scheme. As classification accuracy remains stable across all configurations (Figure 19), we identify Directional growth with Centering as the optimal trade-off between storage efficiency and performance.

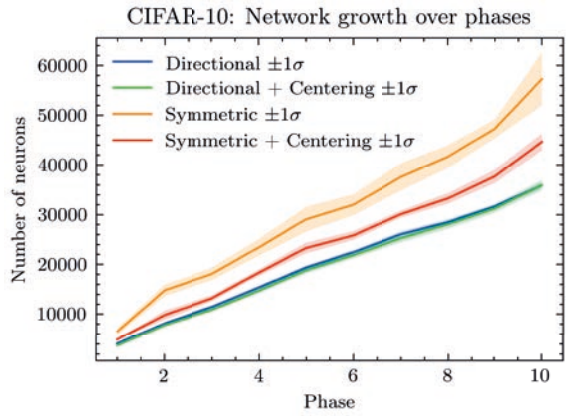

Figure 18. Number of neurons (N) of a growing SatSOM through different phases of the class-incremental CIFAR-10 experiment. Aggregated by method hyperparameters.

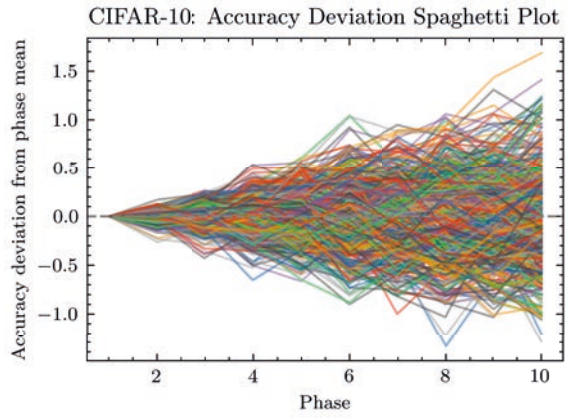

Figure 19. Deviation from the mean accuracy of all of t he CIFAR-10 experiment runs (400 trials) through the phases. The deviation is not significant.

Regarding topology, the Directional scheme maintains a near-square aspect ratio throughout the training (Figure 20), with no significant distortion caused by Centering. Additionally, the sensitivity parameter β showed negligible impact on these results (Figure 21).

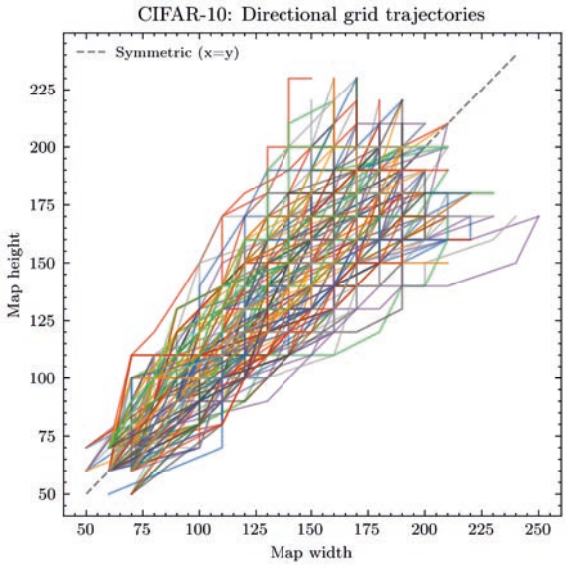

Figure 20. Grid shape with the Directional scheme of all of the CIFAR-10 experiment runs (400 trials) through the phases. Symmetric square grid shape is marked as the dashed ($x = y$) line.

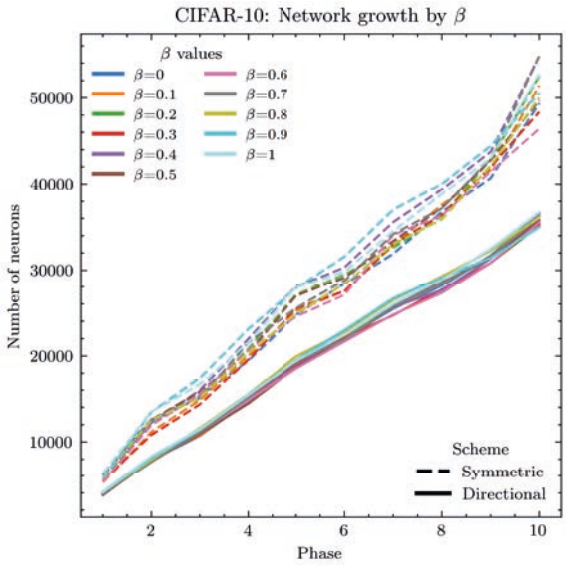

Figure 21. Growing SatSOM size (N) aggregated according to the β hyperparameter. No significant pattern is visible.

8 Conclusion

In this paper, we introduced SatSOM, a novel extension of the SOM algorithm designed to mitigate catastrophic forgetting in continual learning scenarios. The core innovation of SatSOM is a saturation mechanism that modulates the learning

rate and neighborhood radius based on individual neuronal training history. As a neuron accumulates knowledge, its plasticity diminishes, effectively stabilizing its weights. This dynamic directs new information toward unsaturated neurons, thereby minimizing interference with previously learned representations. Furthermore, the local nature of the model facilitates the removal of obsolete data through the selective re-initialization of prototypes.

Our empirical results demonstrate that SatSOM significantly outperforms other SOM-based methods in continual learning tasks. It yields performance competitive with DSDM, particularly on datasets with lower dimensionality, and exhibits retention capabilities approximating those of a kNN baseline. Crucially, ablation studies confirm that the saturation mechanism is the primary driver of these improvements, underscoring its efficacy in managing the stability-plasticity trade-off.

This research contributes to the development of continual learning systems that are efficient, stable, and compatible with strict online constraints. Unlike state-of-the-art approaches that often require external memory buffers, explicit task boundaries, or generative rehearsal, SatSOM offers a lightweight, interpretable alternative. Its fixed memory footprint makes it particularly promising for resource-constrained applications.

To address scenarios where task complexity is unknown *a priori*, we also introduced a dynamic variant: the Growing SatSOM. This extension demonstrates that simple, boundary-driven expansion rules allow the model to adapt its capacity on the fly. By combining directional growth with a grid re-centering strategy, the model approximates the optimal network size for a given task without compromising the storage efficiency or stability of the core algorithm.

The primary limitation of the current approach lies in the inherent trade-off between local specificity and global abstraction. As the effective neighborhood radius decays, the network becomes increasingly *localist*—a phenomenon also observed in CSOM [18]. While this behavior enhances retention, it constrains the model's ability to capture high-level abstractions that span large regions of the data manifold. Future work will address this by exploring hierarchical architectures, similar to those proposed in [4], to enable global abstraction while maintaining continual learning capabilities at the local level.

Several other promising research directions emerge from this study. First, the saturation principle could be adapted to deep neural network architectures, potentially through layer-wise or neuron-specific plasticity control to mitigate forgetting in deep representations. Similarly, the mechanism aligns well with neuromorphic paradigms; self-organizing variants of Operational Neural Networks, such as those proposed by Zhang et al [20], could integrate saturation-based plasticity with high biological fidelity.

Beyond architectural modifications, SatSOM shows potential for integration into hybrid systems, acting as a guide for replay mechanisms [9]. Finally, exploring the deployment of SatSOM on resource-constrained hardware would validate its utility for on-device learning in robotics and embedded environments.

Ultimately, SatSOM offers a biologically grounded and visually interpretable framework for continual learning. Our findings confirm that the adaptive modulation of plasticity is a potent mechanism for mitigating catastrophic forgetting. This work establishes a robust foundation for scaling these principles to complex architectures and deployment in open-ended, dynamic environments.

9 Acknowledgements

We thank Dr. Leszek Grzanka for his support in running the experiments.

This research was supported by the funds assigned to AGH University by the Polish Ministry of Education and Science.

We gratefully acknowledge Polish high-performance computing infrastructure PLGrid (HPC Center: ACK Cyfronet AGH) for providing computer facilities and support within computational grant no. PLG/2025/018713.

10 Compliance with Ethical Standards

This research adheres to all applicable ethical standards and guidelines. All datasets used in this work are publicly available and widely used in the research community for benchmarking machine learning models. The authors declare that they have no conflict of interest.

References

[1] Pouya Bashivan, Martin Schrimpf, Robert Ajemian, Irina Rish, Matthew Riemer, and Yuhai Tu. Continual learning with self-organizing maps, 2019.

[2] Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. Deep learning for classical japanese literature, 2018. cite arxiv:1812.01718Comment: To appear at Neural Information Processing Systems 2018 Workshop on Machine Learning for Creativity and Design.

[3] Matthias Delange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Greg Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. IEEE Transactions on Pattern Analysis and Machine Intelligence, PP:1–1, 02 2021.

[4] Michael Dittenbach, Dieter Merkl, and Andreas Rauber. Growing hierarchical self-organizing map. volume 6, pages 15 – 19 vol.6, 02 2000.

[5] Pedro M. Domingos and Geoff Hulten. Mining high-speed data streams. In Knowledge Discovery and Data Mining, 2000.

[6] Robert M French. Catastrophic forgetting in connectionist networks. Trends in cognitive sciences, 3(4):128–135, 1999.

[7] Alexander Gepperth. An energy-based som model not requiring periodic boundary conditions. In 2017 12th International Workshop on Self-Organizing Maps and Learning Vector Quantization, Clustering and Data Visualization (WSOM), pages 1–6, 2017.

[8] Alexander Gepperth and Cem Karaoguz. Incremental learning with self-organizing maps. In 2017 12th International Workshop on Self-Organizing Maps and Learning Vector Quantization, Clustering and Data Visualization (WSOM), pages 1–8. IEEE, 2017.

[9] Tyler L. Hayes, Nathan D. Cahill, and Christopher Kanan. Memory efficient experience replay for streaming learning. In 2019 International Conference on Robotics and Automation (ICRA), page 9769–9776. IEEE Press, 2019.

[10] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. Proceedings of the National Academy of Sciences, 114(13):3521–3526, 2017.

[11] T. Kohonen. The self-organizing map. Proceedings of the IEEE, 78(9):1464–1480, 1990.

[12] Vincenzo Lomonaco and Davide Maltoni. Core50: a new dataset and benchmark for continuous object recognition. In Sergey Levine, Vincent Vanhoucke, and Ken Goldberg, editors, Proceedings of the 1st Annual Conference on Robot Learning, volume 78 of Proceedings of Machine Learning Research, pages 17–26. PMLR, 13–15 Nov 2017.

[13] German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. Neural networks, 113:54–71, 2019.

[14] Kosmas Pinitas, Spyridon Chavlis, and Panayiota Poirazi. Dendritic self-organizing maps for continual learning, 2021.

[15] Piotr Porwik, Tomasz Orczyk, Krzysztof Wrobel, and Benjamin Mensah Dadzie. A novel method for drift detection in streaming data based on measurement of changes in feature ranks. Journal of Artificial Intelligence and Soft Computing Research, 15(2):147–166, 2025.

[16] Julien Pourcel, Ngoc-Son Vu, and Robert M French. Online task-free continual learning with dynamic sparse distributed memory. In European Conference on Computer Vision, pages 739–756. Springer, 2022.

[17] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 5533–5542, 2017.

[18] Hitesh Vaidya, Travis Desell, Ankur Mali, and Alexander Ororbia. Neuro-mimetic task-free unsupervised online learning with continual self-organizing maps, 2024.

[19] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. arXiv e-prints, page arXiv:1708.07747, August 2017.

[20] Junming Zhang, Hao Dong, Jinfeng Gao, Ruxian Yao, Gangqiang Li, and Haitao Wu. Self-organized operational neural networks for the detection of atrial fibrillation. Journal of Artificial Intelligence and Soft Computing Research, 14(1):63–75, 2023.

Igor Urbanik received the B. Eng. degree in computer science from AGH University of Kraków in 2025 and is currently an M. Eng. student in data science at the same institution. His research focuses on robotics, reinforcement learning, and embodied artificial intelligence.
https://orcid.org/0009-0008-2202-7649

Paweł Gajewski received the B.S. and M.Sc. degrees in computer science from the Jagiellonian University, Kraków, Poland, in 2010 and 2012, respectively, and the Ph.D. degree in computer science from the AGH University of Krakow, Poland, in 2024. He was a Visiting Researcher with Universität Bremen, Germany, and the University of Aberdeen, U.K., during his doctoral studies. He is currently an Assistant Professor with the AGH University of Krakow. His research interests include reinforcement learning for robotics, offline reinforcement learning, time series forecasting, biologically inspired neural network models, and adaptive robot skill learning.
https://orcid.org/0000-0003-0931-2476