ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

General Continual Unsupervised Learning with Augmented Vector Quantization

Nasr Allah Aghelias[1], Bernard Girau[1] and Hervé Frezza-Buet[2] *

1- Université de Lorraine - CNRS - LORIA - France

2- Université de Lorraine - CentraleSupélec - CNRS - LORIA - France

Abstract. In this paper, we introduce a novel online heuristic for dynamic topological vector quantization models, called SAND (Stable Adaptive Network Drift-handling). This algorithm enhances existing models by enabling them to track non-stationary distributions, i.e., input streams composed of distinct contexts that drift over time. It employs an online out-of-distribution detector to identify context switches and subsequently updates the labels of vector quantization model nodes accordingly. The proposed algorithm is fully unsupervised, requiring no external labels to learn, recall, or adapt to varying contexts. SAND thus provides an efficient solution to continuously learn representations of evolving data streams in dynamic environments.

1 Introduction

Most existing continual learning methods [1, 2] address incremental supervised classification, where task identities are predefined. Only a few approaches are designed for true online operation [3, 4], and concept drift—the gradual or abrupt evolution of data distributions over time [5]—is rarely treated explicitly.

In this work, we formulate continual learning in a general setting where context boundaries and dynamics are unknown. We argue that a continual learning algorithm suitable for such non-stationary environments should: i) process data online in a single pass, ii) operate under limited computational resources, iii) incrementally discover and represent contexts based on similarities in the input distribution, and iv) recall and adapt previously learned contexts under drift.

Dynamic topological vector quantization (TVQ) methods have been explored for online and continual learning because they combine incremental representation learning with adaptive, self-organizing graph structures. Early methods such as Growing Neural Gas (GNG) [6] and its extensions—GWR [7] and Gamma-GWR [8]—demonstrated strong adaptability by dynamically adjusting their capacity and capturing temporal structure. However, they remain sensitive to noise and lack explicit mechanisms for detecting context switches or handling gradual drift. The SOINN family [9], including SOINN+ [10] and ASOINN [11], was designed to manage noisy data streams through adaptive similarity thresholds and structural pruning. Nevertheless, these models typically address staged distribution switches and adapt poorly when contexts evolve continuously.

Overall, GWR-type models are well-suited for drift adaptation but fragile under noise, while SOINN-type models excel in noise tolerance yet struggle with

*This work was supported by the projet ANR-23-IAS3-0001 SORLAHNA

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

gradual drift. To address these limitations, we propose the Stable Adaptive Network Drift-handling (SAND) heuristic—a lightweight mechanism that augments dynamic TVQ methods with explicit robustness to both noise and drift. SAND employs an Out-of-Distribution (OOD) detector to identify context switches and a controlled adaptation mechanism to refine or recall existing representations as contexts evolve. By coupling drift awareness with stability, SAND enables fully unsupervised online continual learning in non-stationary environments.

2 General Continual Unsupervised Learning

The usual machine learning paradigm relies on learning from i.i.d. samples $\{z_i\}_{1\leq i\leq N}$ obtained from an unknown distribution Z. Let us consider here this distribution as being parametrized and denoted by Z_θ. Throughout this paper, we take the example of samples in the plane, taken in a fixed rectangular area, where the parameter θ is the position of the center of a disk, included in that rectangle, having a fixed radius R. Z_θ is then made of a probability density that is constant inside the disk, with a value p_{in}, and constant elsewhere in the rectangle, with a smaller value p_{out} modelling noise. Extending this paradigm to the continual learning scheme consists in considering samples as provided by a non stationary distribution. They are given as a stream z_t, where elements are taken at each time t according to Z_{θ_t}. The continual learning paradigm introduces changing tasks, also referred to as contexts c_i. In our formulation, it means that $\theta_t \in \{c_0, \ldots, c_{k-1}\}$. Our example is made of $k = 3$ contexts, each one corresponds to a specific disk center. The z_t stream is such as θ_t is piecewise constant, meaning in our example that samples are provided by a given disk (with surrounding noise) for a while, before a new disk is used for the next samples, thus corresponding to a context switch, see figure 1(left).

VQ consists here in using a graph of prototypes, as provided by TVQ algorithms, for representing the current Z_{θ_t}. Detecting as soon as possible which of the contexts currently drives the sample stream and having a representation of its distribution is the challenge addressed by unsupervized continual learning. The number of contexts and the shape of each $Z_{c_{i_t}}$ are to be discovered online from the z_t stream only. The robustness to drifting contexts is also addressed here. Drifting means that the set of contexts depends on time, i.e. that $\theta_t \in \{c_{0t}, \ldots, c_{k-1t}\}$, where each c_{it} is slowly changing. In our example, we will use a slow continual rotational motion of the disk centers, see figure 1(right).

3 Stable Adaptive Network Drift-handling (SAND)

The Stable Adaptive Network Drift-handling (SAND) heuristic is designed to complement dynamic Topological Vector Quantization (TVQ) models such as GNG, GWR, or SOINN. It operates after each training step of the learning model and enables the system to detect and adapt to context switches and concept drift in an entirely unsupervised and online fashion. A pseudocode description is provided in Algorithm 1.

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

Each node $\mathbf{n} \in \mathcal{V}$ in the TVQ graph maintains a context-frequency vector $\mathbf{n}$.ctx, where each entry $\mathbf{n}$.ctx$[k]$ quantifies how strongly the node is associated with context k. These frequency profiles act as soft memory traces that allow the model to (i) recognize when a previously encountered context reappears, and (ii) update or forget obsolete associations when the data distribution evolves.

Algorithm 1 Stable Adaptive Network Drift-handling (SAND)

Initialize parameters : `current_ctx` = tmp, `nb_ctx` = 0, `stability_idx` = 0, `is_rec_ctx` = `false`, `is_stable` = `false`

while true **do**
  $z \leftarrow$ new training pattern *// data are provided online as a stream*
  train(z) *// apply the underlying TVQ algorithm (GNG, GWR, SOINN, ...) to the current training pattern*
  $\mathbf{n}^* = \text{argmin}_{\mathbf{n} \in \mathcal{V}} \|z - \mathbf{n}.w\|$ *// find the best matching unit* $\mathbf{n}^*$
  `dist` $\leftarrow$ `OOD`(z) *// compute the out-of-distribution score of the input* z
  `S.push(dist)`
  `stability_idx` += 1
  if `S.mean()` $> switch_{ctx}$ **then** *// if* `S`*'s mean is greater the context switch threshold*
    `stability_idx` $\leftarrow$ 0 *// reset the* `stability_idx`
    if `is_rec_ctx` and `is_stable` **then** *// if the previously seen context is a recurring one*
      for all $\mathbf{n}$ in $\mathcal{V}$ **do**
        if $\sum_{k=0}^{\texttt{nb\_ctx}-1} \mathbf{n}.\text{ctx}[k] < \texttt{nb\_ctx} \cdot \delta$ and $|\mathcal{V}| > 3$ **then**
          $\mathcal{V} \leftarrow \mathcal{V} \setminus \{\mathbf{n}\}$ *// delete obsolete vertices that result from context drift or noise*
      `is_stable` $\leftarrow$ `false`
  else
    if `stability_idx` $< N$ **then** *// as long as the input is unstable*
      $(\mathbf{n}^*)$.tmp += 1 *// increment the frequency of the context 0 for the BMU*
    else if `stability_idx` $= N$ **then** *// once stability is ascertained*
      `is_stable` $\leftarrow$ `true`
      if `nb_ctx` $= 0$ **then** *// if no context was discovered yet*
        `current_ctx` $\leftarrow$ `nb_ctx` *// and set the current context to 0*
        `nb_ctx` += 1
      else
        $k^* = \underset{k \in [\![0, \texttt{nb\_ctx}-1]\!]}{\text{argmax}} |\mathcal{D}_k^\delta \cap \mathcal{D}_{\text{tmp}}|$ *// compute the context* k^* *which has the highest overlap with the temporary context*
        $\tau \leftarrow |\mathcal{D}_{k^*}^\delta \cap \mathcal{D}_{\text{tmp}}| / |\mathcal{D}_{\text{tmp}}|$ *// compute the overlap between the temporary context and* k^*
        if $\tau < rec_{ctx}$ **then** *// if the overlap is less than a predefined threshold*
          `current_ctx` $\leftarrow$ `nb_ctx` *// set the current context to a new context* `nb_ctx`
          `nb_ctx` += 1
          `is_rec_ctx` $\leftarrow$ `false` *// no recurring context has been detected*
        else *// otherwise*
          `current_ctx` $\leftarrow k^*$ *// set the current context to* k^*
          `is_rec_ctx` $\leftarrow$ `true` *// which is the recurring context in this case*
        for all $\mathbf{n}$ in $\mathcal{V}$ **do** *// for all nodes in the graph*
          $\mathbf{n}$.ctx[`current_ctx`] $\leftarrow \mathbf{n}$.tmp *// assign the frequency of the temporary context to the current context*
          $\mathbf{n}$.tmp $\leftarrow 0$ *// then reset the frequency of the temporary context*
      else *// as long as the input is stable increment the frequency*
        $(\mathbf{n}^*)$.ctx[`current_ctx`] += 1 *// of the current context for the BMU*

Since the algorithm operates without labels, SAND relies on an out-of-distribution (OOD) detector to signal potential distributional changes. In our implementation, the detector is an online Gaussian Mixture Model (GMM) that outputs a novelty score for each incoming sample. These scores are buffered within a sliding window $\mathbf{S}$ of fixed capacity N, whose moving average represents the system's short-term perception of novelty. To prevent spurious context switches, this average is smoothed over time, yielding a stable estimate robust to noise and transient fluctuations. When a new input z arrives, the best-matching unit (BMU(z)) denoted as $\mathbf{n}^*$ is first recomputed to ensure it still exists, since

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

nodes may be dynamically deleted by the TVQ model. The OOD detector then processes z and updates $\mathbf{S}$; simultaneously, a stability index is incremented to record how long the system has remained in its current state.

If the mean of $\mathbf{S}$ exceeds a context-switch threshold, SAND interprets this as evidence of a potential new context. The stability index is reset, marking the start of a transition phase. If the system is leaving a previously stable regime, SAND performs selective forgetting: nodes whose cumulative context frequency falls below a drift threshold δ are removed. This pruning mechanism prevents obsolete representations from persisting after substantial distributional drift.

If the OOD average remains below the threshold, the system is considered stable, and SAND updates the frequency vectors accordingly. During early adaptation (when the stability index is smaller than N), updates accumulate in a temporary context buffer tmp, representing short-term evidence of a possible emerging pattern. Once the stability index surpasses N, the system transitions to context consolidation, where the temporary context tmp is either merged with an existing one or promoted to a new context. This decision is made by comparing the set of nodes associated with the temporary context, $\mathcal{D}_{\text{tmp}} \coloneqq \{\mathbf{n} \in \mathcal{V} \mid \mathbf{n}.\text{tmp} > 0\}$, with those associated with each known context k, $\mathcal{D}_k^\delta \coloneqq \{\mathbf{n} \in \mathcal{V} \mid \mathbf{n}.\text{ctx}[k] > \delta\}$. The overlap between $\mathcal{D}_{\text{tmp}}$ and each $\mathcal{D}_k^\delta$ measures how similar the temporary distribution is to previously seen ones. The context k^* with the largest overlap is selected, and if this overlap falls below a recurrence threshold, the temporary context is declared novel and assigned a new index; otherwise, it is merged with k^*. Afterward, temporary frequencies are transferred to the active context and reset, completing the adaptation cycle.

In essence, SAND equips neural models of topological vector quantization with a lightweight yet powerful drift-handling mechanism. It continuously monitors input novelty, dynamically allocates and reuses contexts, and prunes outdated nodes—allowing fully unsupervised continual learning in non-stationary streaming environments.

4 Experiments

The input stream consists of three evolving contexts, denoted c_{0t}, c_{1t}, and c_{2t}, where t represents an arbitrary time step in the stream. Each context c_{kt}, for all $k \in \{0, 1, 2\}$, corresponds to a uniformly sampled disk in the plane $\mathbb{C}$ of fixed radius 1, whose center evolves over time according to $e^{i(\omega t + 2\pi k/3)}$ where ω is a constant angular velocity controlling the drift speed. In this setup, the three disks rotate synchronously around the origin in the two-dimensional plane $\mathbb{C}$, thus modeling continuous, cyclic concept drift. To increase realism and test robustness to noise, uniformly distributed background samples are added from the box $[-3, 3] \times [-3, 3]$. In our experimental setup, a context switch occurs every 10^4 iterations. The input data distributions are shown in figure 1, in which the colors allow us to differentiate the contexts, whereas the data is provided unlabeled to the learning algorithm.

In this paper, we introduce the following rationale to quantify the perfor-

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

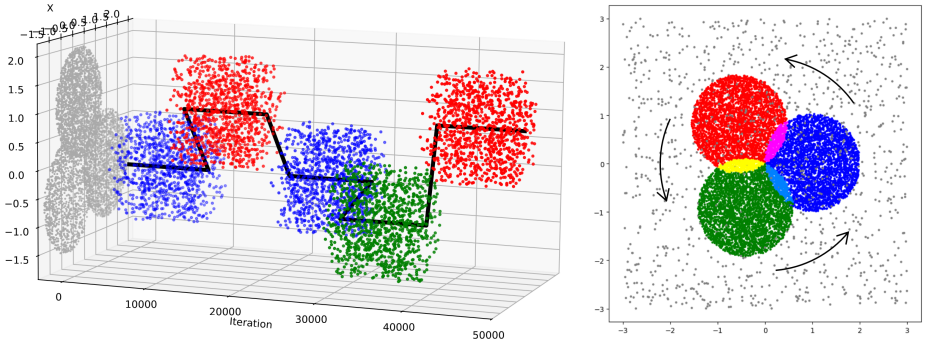

Fig. 1: Left: Time evolution of the input stream (context switches, no drift), noisy points not represented. The thick line shows the position of the current disk center. Right: Evolution of distribution (contexts' drift).

mance of SAND for unsupervised continual learning. We consider K-Means as providing a reference performance for the quantization of a fixed dataset with a given number of prototypes. At each time t, we generate a set D_t of 10^3 samples from Z_{θ_t} *without noise*, we apply K-Means to these samples and compute its quantization error $E_t^{K-\text{Means}} = \frac{1}{|D_t|}\sum_{z\in D_t} \|\text{BMU}^{K-\text{Means}}(z).w - z\|$. The idea is to compare it to the SAND-GWR graph considering only the nodes corresponding to the current context k^*, that is the set $\mathcal{D}^{\delta}_{k^*}$ as defined in section 3. Hence, we use $K = |\mathcal{D}^{\delta}_{k^*}|$ prototypes for K-means. Then, we compute E_t^{SAND} using D_t and the vertices in $\mathcal{D}^{\delta}_{k^*}$. We define the quantization score as the ratio $S_t = E_t^{\text{K-Means}}/E_t^{\text{SAND}}$. A low score S_t signifies a divergence of SAND-GWR's representation from the reference K-Means solution, whereas a score close to 1 indicates that SAND-GWR performs comparably to the reference K-Means despite the presence of noise, context switches and the distribution drift.

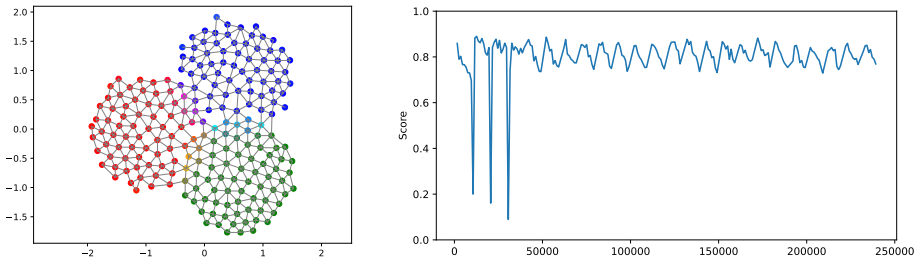

Fig. 2: SAND-GWR topology, and its quantization score w.r.t. iterations.

Figure 2(left) illustrates the final topology of SAND-GWR after training on 2.4×10^5 samples. Color of vertex $\mathbf{n}$ depends on the values of k for which we have $\mathbf{n}.\text{ctx}[k] > \delta$. The model successfully captures the underlying data distribution while effectively filtering out noise. Figure 2(right) demonstrates that, despite

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

substantial noise and spatial overlap between contexts, SAND-GWR sustains stable performance after sequentially learning all input contexts. Despite brief drops in the quantization score during context transitions—reflecting adaptation to new distributions—the model rapidly recovers, with S_t fluctuating around 0.8. This level is noteworthy given the concept drift and noisy inputs, especially considering that the upper quantization bound for a GWR, determined by its intrinsic quantization quality, was measured around 0.9 under ideal i.i.d. conditions (not detailed here). Hence, SAND-GWR's performance remains close to this ceiling, highlighting its robustness and efficiency in dynamic environments.

5 Conclusion and future work

In this paper, we introduced a heuristic that enables vector quantization methods to incrementally learn contexts from evolving data streams. Applied to GWR, the method demonstrated promising results: despite the presence of noise, the network successfully learned the identities of multiple contexts and reliably recognized them when re-encountered. Moreover, the enhanced GWR adapted previously learned contexts when they underwent drift. These results suggest that the proposed heuristic provides a general mechanism for context discovery that can be seamlessly integrated into a wide range of vector quantization algorithms. This opens up new perspectives for future work.

References

[1] Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *IEEE transactions on pattern analysis and machine intelligence*, 46(8):5362–5383, 2024.

[2] G. I Parisi, R. Kemker, J. L Part, C. Kanan, and S. Wermter. Continual lifelong learning with neural networks: A review. *Neural networks*, 113:54–71, 2019.

[3] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

[4] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. *arXiv preprint arXiv:1812.00420*, 2018.

[5] G. Ditzler, M. Roveri, C. Alippi, and R. Polikar. Learning in nonstationary environments: A survey. *IEEE Computational Intelligence Magazine*, 10(4):12–25, 2015.

[6] Bernd Fritzke. A growing neural gas network learns topologies. *Advances in neural information processing systems*, 7, 1994.

[7] Stephen Marsland, Jonathan Shapiro, and Ulrich Nehmzow. A self-organising network that grows when required. *Neural networks*, 15(8-9):1041–1058, 2002.

[8] G. I Parisi, J. Tani, C. Weber, and S. Wermter. Lifelong learning of human actions with deep neural network self-organization. *Neural Networks*, 96:137–149, 2017.

[9] Shen Furao and Osamu Hasegawa. An incremental network for on-line unsupervised classification and topology learning. *Neural networks*, 19(1):90–106, 2006.

[10] Chayut Wiwatcharakoses and Daniel Berrar. Soinn+, a self-organizing incremental neural network for unsupervised learning from noisy data streams. *Expert Systems with Applications*, 143:113069, 2020.

[11] Furao Shen and Osamu Hasegawa. A fast nearest neighbor classifier based on self-organizing incremental neural network. *Neural networks*, 21(10):1537–1547, 2008.