Don't Drift Away: Advances and Applications of Streaming and Continual Learning

Andrea Cossu[1], Davide Bacciu[1], Alessio Bernardo[2], Emanuele Della Valle[2], Alexander Gepperth[3], Federico Giannini[2], Barbara Hammer[4], and Giacomo Ziffer[2]

1- University of Pisa 2- DEIB, Politecnico di Milano
3- University of Applied Sciences Fulda 4- Bielefeld University

Abstract. Non-stationary environments subject to concept drift require the design of adaptive models that can continuously learn and update. Two primary research communities have emerged to address this challenge: Continual Learning (CL) and Streaming Machine Learning (SML). CL manages virtual drifts by learning new concepts without forgetting past knowledge, while SML focuses on real drifts, rapidly adapting to evolving data distributions. However, a unified approach is needed to balance adaptation and knowledge retention. Streaming Continual Learning (SCL) bridges the gap between CL and SML, ensuring models retain useful past information while efficiently adapting to new data. We explore key challenges in SCL, including handling temporal dependencies in data streams and adapting latent representations for personalization and knowledge editing. Additionally, we identify promising SCL benchmarks which can foster and promote a unified research effort between CL and SML.

1 Introduction

The world we live in is inundated with data. The rise of Social Media and the Internet of Things in the last decade has driven rapid data growth, generating continuous, unbounded flows called data streams. These streams produce infinite sequences of elements that arrive over time and cannot be accessed simultaneously. A significant challenge is developing learning models that adapt to data's continuously evolving nature. Data may, in fact, change its distribution over time, causing the problem of **concept drift**. Concept drift is defined as an unforeseeable change in the data-generating process that induces a change in the statistical properties, which is more significant than random fluctuations or anomalies [1, 2]. A concept is the unobservable process that generates data.

Concept drifts can be categorized into two main types: virtual and real. Given a machine learning problem, a *virtual concept drift* happens when the input or the target distribution changes without affecting the rules that map the inputs to the desired targets. For example, in a classification problem that maps the inputs X to the labels y, a concept drift changes the probability $P(X|y)$ or $P(y)$ without affecting $P(y|X)$. On the other hand, a *real concept drift* changes the mapping rules between inputs and targets. In a classification problem, this change impacts the probability $P(y|X)$. For instance, in credit scoring, where the goal is to predict loan default risk based on factors like income and credit history, virtual concept drift occurs when the distribution of these factors shifts over time without altering their relationship with default risk, keeping the decision boundary unchanged. In contrast, real concept drift arises when economic changes, market trends, or new financial policies reshape this relationship, meaning that the same income and credit history may now indicate a different level of risk. This requires the model to adapt its decision boundary.

Due to concept drift, it is unfeasible to accumulate the data in a repository and train the model offline. First of all, the repository would contain inconsistent data due to concept drift during data acquisition, and, in addition, subsequent concept drift would render to model obsolete, thus requiring massive re-training using a new repository. The model should, instead, continuously learn and update using the data stream. Two communities emerged to reach this end: **Continual Learning (CL)** [3] and **Streaming Machine Learning (SML)** [4]. The main distinctions between the two lie in their objectives and the types of drifts they manage. The most prominent CL approaches, such as class-incremental learning, rely on data streams where each element is a large batch (experience) of unordered data points representing a concept. Each new experience introduces a virtual drift describing a new subproblem specified in a new feature distribution. The new concept does not contradict the previous since it is defined in a new feature subspace. The goal of a CL agent is to solve the new subproblem without forgetting the previous ones. Conversely, a data stream is a sequence of single data points in an SML setting, and drifts may be real. Since a real drift produces changes in the decision boundary, the new concept may contradict the previous. An SML model cannot, thus, solve both the machine learning problems together. The goal is, instead, to continuously detect changes and adapt as rapidly as possible to new problems, minimizing resource usage.

Online CL (OCL) [5] represents a first step towards converging the two communities. OCL approaches the SML setting by splitting each experience into mini-batches provided over time and containing tens of data points. However, OCL still aligns with the CL objectives in virtual drifts.

Continual reinforcement learning (CRL) [3] assumes that the environment itself changes over time, thus partly invalidating previously learned policies, which is analogous to SML scenarios. Denker et al. [6] present "Reward-incremental reinforcement learning" as a part of this special session, where virtual and real concept drifts may occur due to non-stationary environments.

Conversely, avoiding forgetting assumes a new meaning in real streaming scenarios. Whereas adapting to the current concept is important, the knowledge associated with the previous concepts can be helpful when a drift introduces a new concept that resembles an earlier one, or when an old concept re-occurs over time. Additionally, parts of previous knowledge may remain valid after a drift. **Streaming Continual Learning (SCL)** denotes a significant innovation in this direction. The need for SCL was initially expressed by Gunasekara et al. [7] and then elaborated by Giannini et al. [8]. SCL is a unified setting combining SML and CL methodologies to quickly adapt to changes without forgetting meaningful information from the past. It ensures that models retain relevant knowledge while discarding obsolete information based on the evolving needs of the environment. SCL may offer a robust approach by balancing the retention of general knowledge and the ability to adapt to new data quickly.

Vaquet et al. [9] present "Compression-based kNN for Class Incremental Continual Learning" as part of this special session. The work investigates the challenge of forgetting in data streams from an SCL perspective. It explores the transfer of strategies from SML to CL. The authors propose a compression-based kNN scheme to address forgetting and re-occurring concepts, contributing to integrating SML and CL paradigms.

Finding an appropriate setting for applying SCL is crucial in many real-world streaming applications. This paper analyzes specific aspects of this challenge, such as handling temporal dependence in a data stream (Section 2) and adapting latent representations (Section 3). Section 4 emphasizes the need for a common benchmark between the SML and CL communities, which operate on different datasets and problems. Finally, Section 5 elaborates on our conclusions.

2 Learning from temporally dependent data

Temporal dependency refers to the situation where the value of a given data point depends on previous observations. Several works [10, 11, 12, 13, 14] highlighted the importance of accounting for temporal dependency when solving machine learning problems on data streams, as it reflects how the data evolves and how current outcomes can be predicted based on prior information. Formally, this can be expressed as: $\exists \tau \; P(a_t \mid b_{t-\tau}) \neq P(a_t)$, where $a_t \in X_t \cup \{y_t\}$ is the feature or target at time t, and $b_{t-\tau} \in X_{t-\tau} \cup \{y_{t-\tau}\}$ represents the feature or target at time $t - \tau$. Here, X_t and $X_{t-\tau}$ are the feature sets at times t and $t - \tau$, and y_t and $y_{t-\tau}$ are the corresponding targets. The temporal dependency indicates that past data points influence the current data point's probability distribution.

The presence of temporal dependencies influences many real-world scenarios. Let us consider, for example, the Internet of Things, where sensors continuously collect data over time. Particularly in the context of weather stations recording temperature, each new temperature reading is influenced by previous measurements. For instance, the current temperature is influenced by the temperature recorded in the past few hours or days. Temperature often follows specific patterns, such as daily fluctuations due to the day-night cycle or seasonal trends.

Similarly, in video surveillance systems, temporal dependency is fundamental for tasks like object tracking. The position of an object in one frame is strongly influenced by its position in the preceding frames, as objects in motion tend to follow a continuous path. This sequential relationship helps the system predict where the object will appear next.

Sometimes, the temporal dependencies observed in a data stream can result from spatial dependencies in the real world. Consider, for example, satellite imagery analysis. The temporal dependency may arise from the physical movement of the satellite along its orbit. As the satellite moves along a defined trajectory, each new observation is related to previous ones, not because of a time-based correlation but due to the spatial continuity of the observed region. At time t, the satellite captures an image of a specific Earth location. At time $t + \tau$, the satellite captures an image from a neighbouring location along its path.

In robotics, a robot's actions, such as movement or decision-making, are often influenced by its previous states. For instance, the robot's current position and orientation are directly linked to where it was in earlier time steps. The robot moves through a sequence of actions and decides at each step based on the history of its movements.

Despite the crucial role of temporal dependency in streaming scenarios, SML and CL usually do not consider it and assume data points to be independent. **Time Series Analysis (TSA)** [15] is explicitly meant to address temporal dependency, even if rarely applied in streaming settings. In recent years, the forecasting potential of deep learning models has emerged [16], where variants of Recurrent Neural Networks (RNNs) are widely applied. Betti et al. [17] present "Stability of State and Costate Dynamics in Continuous Time Recurrent Neural Networks" as part of this session to study RNNs streaming applications. This work investigates stability in continuous-time RNNs, a crucial property for ensuring safe and reliable lifelong learning. It provides key conditions for achieving bounded state and costate dynamics, leveraging concepts from optimal control theory to enhance stability.

Moving towards an SCL perspective, integrating SML and CL methodologies could provide a viable solution for managing temporal dependencies when learning from data streams. Since TSA widely applies RNN models, one may use them online in an SML way to learn continuously and tame temporal dependencies between data points. Two works [18, 14] have proposed the application of RNNs in an SML scenario. Their solutions accumulate data points from a data stream in fixed-size mini-batches. Whenever a mini-batch is complete, they build sequences of items using a sliding window and train an RNN model continuously. This way, they tame temporal dependencies while learning continuously.

One may consider CL architectural strategies to boost the adaptation to new concepts and avoid forgetting crucial past information. CL is, in fact, explicitly meant to deal with deep learning models. Architectural strategies share parts of the network architecture between concepts, use transfer learning, and focus on one concept at a time (as an SCL perspective requires). These methods may boost the adaptation to new concepts since the model can reuse part of the previous knowledge and combine it with the one acquired from the current data. Moreover, the associated weights are usually frozen to remember the old concepts. Gated Incremental Memories [19] propose a new version of the CL architectural strategy of Progressive Neural Networks [20] to be applied on top of an RNN model in a CL scenario. Continuous Progressive Neural Networks [21] represent one of the first pioneering solutions that apply this approach continuously to manage real concept drifts while avoiding forgetting. Additionally, SML techniques can play a crucial role in detecting concept drifts and identifying when to expand the architecture of the network.

Finally, as the M5 challenge findings highlight [16], combining statistical machine learning and deep learning techniques may produce better results in handling TSA problems. Thus, an SCL setting could benefit from the collaboration of SML models (historically based on statistical machine learning) and CL strategies (that usually rely on deep learning techniques).

3 Efficient adaptation of latent representations

The advent of large foundation models [22] incentivized the usage of high-dimensional, hierarchical representations built during a *pre-trainining* phase and later *fine-tuned* for a specific downstream task. In a dynamic world where information is constantly changing, there are two key challenges to address: i) how to *add* new knowledge on top of the existing representations and ii) how to *update* existing representations.

Although related, i) requires building *new* representations while ii) requires *replacing* existing representations with updated versions. The former case is needed when the environment is subject to virtual drifts that introduce new information without invalidating the previous one. This is usually the case in CL scenarios, where new experiences introduce additional knowledge that the solution must incorporate while retaining previous knowledge. In contrast, the latter case is applicable when the environment is subject to real drifts that make previous information obsolete. Therefore, the solution must actively enforce the forgetting of obsolete information. A popular way of addressing both tasks in real-word applications is to re-create all representations from scratch, usually through a new pre-training phase with the new *and* old (but still relevant) data. Unfortunately, this approach is not sustainable as it requires carrying out a very expensive and time-consuming pre-training phase, even when only a small amount of novel information is added.

As an alternative, CL approaches allow to incrementally update existing representations through a *continual pre-training* phase [23], which only uses new data. Continual pre-training requires much less time and resources than a pre-training phase from scratch. However, by only considering new data, the update phase must take care of preserving previous knowledge. Usually, continual pre-training is performed by self-supervised methods, which have been proven to be more resistant to forgetting than their supervised counterparts.
SCL offers the opportunity to add new knowledge from a stream of new data. In fact, self-supervised approaches require a large amount of data and tend to underperform when applied online. Ad-hoc variations of self-supervised approaches designed for the online setting can mitigate this effect, as proposed by Cignoni et al. in one of this special session's papers [24].

A relevant application requiring the addition of new information is *personalization* (of pre-trained models). Available models often lack the ability to quickly adapt to a single user's preferences and habits, departing from the knowledge contained in their original version and specializing it to the user's needs and context. For example, it is very easy for a model to recognize a cup from a video stream. It is more difficult to personalize the model's behavior to distinguish between the user's cup and all other cups. A personalized model is asked to learn this distinction for several objects and contexts, with only a few user-specific examples to use during the training phase. SCL approaches can empower the personalization of existing models by combining efficient techniques for updating latent representations in environments where data is scarce and part of a temporally correlated stream like in videos.

In many cases, updating existing information is more common than adding new one. In the presence of real drifts, previous information becomes obsolete and must be replaced to avoid making incorrect predictions. One can update new information by first removing outdated information and then adding the corresponding updated representation. Forgetting of existing representations is considered an issue in CL. This hides the assumption that most CL works focus on virtual drifts, where previous information remains valid. Instead, *unlearning* approaches [25] allow to selectively erase existing knowledge without affecting other representations. This special session presents a work on continual learning, which extends unlearning to erase information over time [26]. Other approaches include model editing [27], with which the model predictions on specific facts are targeted and switched. Scaling model editing to an SCL environment with a stream of high-frequency data remains challenging [28].

Keeping any model, let alone a foundation model, up to date with the most recent news and most relevant information requires taking care of both points we discussed. Research is still in its infancy, proposing and exploring different ways to advance in this task. However, we argue that any model deployed in the real-world will have to cope with its non-stationarity and, ultimately, its time-dependency in an efficient way. The re-training approach cannot scale when billions of personalized models are deployed anywhere in the world. Information that is crucial for one user may not be relevant at all for others. In fact, it may be detrimental. Updates must be continual, based on a small set of data, as well as energy- and time-efficient. Adjusting and personalizing the behavior of predictive models and assistants will require SCL techniques able to satisfy these requirements.

4 Benchmarks for SCL

SCL targets real-world environments where data comes in high-volume, high-frequency, non-stationary streams. Fundamental to the development and assessment of SCL approaches is the identification of datasets and benchmarks that can be used by the research community. Here, we list existing attempts and promising directions that encompass multiple domains: from satellite images to natural language and videos, we show how SCL approaches can have a direct impact on relevant real-world applications.

CLEAR [29] is one of the first benchmarks targeting gradual drifts connected to the temporal evolution of visual concepts. The dataset includes both labelled and unlabeled data. CLEAR can be directly used by SCL approaches along the lines mentioned in Section 3: the unlabeled data constitutes the backbone of the continual pre-training phase, while the labeled data can be used to either fine-tune the resulting model or to include specific information (e.g., through continual model editing which requires labeled data). The dataset can also be considered through the lenses of real drifts, as it continually updates the concepts related to an object (e.g., a personal computer) over a large time span.

Similarly, Wild-Time [30] provides a benchmark focused on temporal distribution shifts. Unlike other distribution shifts, temporal shifts have an inherent structure given by timestamped metadata, which allows models to leverage past trends to predict future variations. The benchmark includes five real-world datasets covering domains such as drug discovery, patient prognosis, and news classification, where temporal shifts significantly impact model performance. Wild-Time evaluates a range of methods, including domain generalization, continual learning, self-supervised learning and ensemble learning, examining their ability to generalize across time. It introduces two evaluation strategies: Eval-Fix, which uses a fixed time split for straightforward benchmarking, and Eval-Stream, a streaming evaluation for continual learning approaches. Experimental results highlight that existing methods struggle with temporal distribution shifts, with an average performance drop of 20% from in-distribution to out-of-distribution data, emphasizing the challenge of evolving real-world environments.

Managing temporal dependencies between sequential images is a critical challenge when dealing with object detection tasks. CLAD [31] could serve as a useful benchmark in this context, particularly for autonomous driving applications. It focuses on object classification and object detection tasks, where the data consists of sequential snapshots captured over time. Another relevant dataset in this context is KITTI [32], which captures 6 hours of real-world traffic scenarios using various sensors, including stereo cameras and a 3D laser scanner. The dataset provides synchronized, timestamped data with object labels in the form of 3D tracklets, making it suitable for tasks like stereo vision and object detection.

CLOC [33] is a dataset of geolocalized, time-stamped images from various locations around the world. The objective is to infer the geographical coordinates of the input image. Interestingly, the authors cast the problem into an online CL problem, which is strongly aligned with our SCL framework. CLOC can be used out-of-the-box to assess the performance of SCL approaches. For example, by employing drift detectors methods that can track the change in geographical location to update the predictor.

EGO4D [34] is a large-scale dataset of several daily-life activity videos from the egocentric views of different users. EGO4D provides an excellent foundation for the personalization task already discussed in Section 3. SCL approaches, taking into consideration the temporal correlation (Section 2) of video streams, are perfectly apt to this task. Due to its resource-intensive character, video processing in SCL will likely be one of the most challenging tasks to tackle, but also one of the most impactful ones.

Natural language processing is another crucial test-bed for SCL. Firehose [35] is a benchmark for personalized language models that includes tweets from multiple users over time. The objective is to develop user-specific language models that adapt to the users' needs. TemporalWiki [36] is a dataset of time-stamped Wikipedia/Wikidata pages that allows us to carefully monitor the amount of knowledge added, removed or changed over time. As such, TemporalWiki is an ideal candidate to assess the performance of SCL approaches in unlearning outdated knowledge or in editing specific pieces of changed information.

Finally, this special session also introduces an SCL benchmark for land use classification [37], which includes satellite images at a fine-grained resolution and subject to drifts over time. Correctly classifying land use requires an understanding of both the spatial context and the temporal evolution of the observed location under different environmental conditions. Predictive models deployed directly on satellites may not access large computational resources, making this use-case fit for several SCL applications.

5 Conclusion

SCL does not aim at blending CL and SML, such that one is absorbed into the other. CL and SML still preserve their own objectives, which are worth studying even in isolation. CL revealed the brittleness of learned representations in non-stationary environments, and is helping in building more robust ones. SML designed several learning approaches that work with drifting data, in environments with limited resources and tight time constraints. SCL works at the intersection of such environments, in applications where models need to learn latent representations and keep them up to date over time while being able to respond quickly and adapt to new temporally correlated data.

Learning latent representations (mapping high-level, semantic concepts into a shared high-dimensional space) is often a slower process than the predictive one (mapping an input into the correct output), requiring gathering knowledge from multiple sources and preserving its consistency. For this reason, SCL systems may be required to act at different temporal resolutions, with one part of the system (played by the SML algorithm) being very responsive at external stimuli and providing predictions and decisions on-the-fly. The other part (played by the CL algorithm) works at a slower pace and receives feedback from the fast component to update its representations, which collectively form the shared knowledge base of the entire SCL model. Even though this view only sketches the nature of a possible SCL approach, we hope it can also provide guidance for those researchers interested in pursuing this ambitious challenge.

References

[1] Jie Lu, Anjin Liu, Fan Dong, Feng Gu, João Gama, and Guangquan Zhang. Learning under Concept Drift: A Review. *IEEE Trans. Knowl. Data Eng.*, 31(12):2346–2363, 2019.

[2] João Gama, Pedro Medas, Gladys Castillo, and Pedro Pereira Rodrigues. Learning with Drift Detection. In *SBIA*, volume 3171 of *LNCS*, pages 286–295. Springer, 2004.

[3] Timothée Lesort, Vincenzo Lomonaco, Andrei Stoian, Davide Maltoni, David Filliat, and Natalia Díaz-Rodríguez. Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges. *Inf. Fusion*, 58:52–68, 2020-06-01.

[4] Albert Bifet, Ricard Gavaldà, Geoff Holmes, and Bernhard Pfahringer. *Machine learning for data streams: with practical examples in MOA*. MIT press, 2018.

[5] Zheda Mai, Ruiwen Li, Jihwan Jeong, David Quispe, Hyunwoo Kim, and Scott Sanner. Online continual learning in image classification: An empirical survey. *Neurocomputing*, 469:28–51, 2022-01-16.

[6] Yannick Denker and Alexander Gepperth. Reward-incremental reinforcement learning. In *ESANN*, 2025.

[7] Nuwan Gunasekara, Bernhard Pfahringer, Heitor Murilo Gomes, and Albert Bifet. Survey on Online Streaming Continual Learning. In *Thirty-Second International Joint Conference on Artificial Intelligence*, volume 6, pages 6628–6637, 2023.

[8] Federico Giannini, Giacomo Ziffer, Andrea Cossu, and Vincenzo Lomonaco. Streaming Continual Learning for Unified Adaptive Intelligence in Dynamic Environments. *IEEE Intelligent Systems*, 39(6):81–85, November 2024.

[9] Valerie Vaquet, Jonas Vaquet, Fabian Hinder, and Barbara Hammer. Compression-based kNN for Class Incremental Continual Learning. In *ESANN*, 2025.

[10] Giacomo Ziffer, Alessio Bernardo, Emanuele Della Valle, Vitor Cerqueira, and Albert Bifet. Towards time-evolving analytics: Online learning for time-dependent evolving data streams. *Data Science*, 6(1-2):1–16, 2023.

[11] Indre Zliobaite, Albert Bifet, Jesse Read, Bernhard Pfahringer, and Geoff Holmes. Evaluation methods and decision theory for classification of streaming data with temporal dependence. *Mach. Learn.*, 98(3):455–482, 2015.

[12] Jesse Read, Ricardo Araújo Rios, Tatiane Nogueira, and Rodrigo Fernandes de Mello. Data Streams Are Time Series: Challenging Assumptions. In *BRACIS (2)*, volume 12320 of *Lecture Notes in Computer Science*, pages 529–543. Springer, 2020.

[13] Vinícius M. A. de Souza, Denis Moreira dos Reis, André Gustavo Maletzke, and Gustavo E. A. P. A. Batista. Challenges in benchmarking stream learning algorithms with real-world data. *Data Min. Knowl. Discov.*, 34(6):1805–1858, 2020.

[14] Giacomo Ziffer, Federico Giannini, and Emanuele Della Valle. Tenet: Benchmarking Data Stream Classifiers in Presence of Temporal Dependence. In *2024 IEEE International Conference on Big Data (BigData)*, pages 1187–1196, 2024.

[15] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[16] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. M5 accuracy competition: Results, findings, and conclusions. *IJOF*, 38(4):1346–1364, 2022.

[17] Alessandro Betti, Marco Gori, and Stefano Melacci. Stability of State and Costate Dynamics in Continuous Time Recurrent Neural Networks. In *ESANN*, 2025.

[18] Álvaro C Lemos Neto, Rodrigo A Coelho, and Cristiano L de Castro. An Incremental Learning Approach Using Long Short-Term Memory Neural Networks. *Journal of Control, Automation and Electrical Systems*, pages 1–9, 2022.

[19] Andrea Cossu, Antonio Carta, and Davide Bacciu. Continual Learning with Gated Incremental Memories for sequential data processing. In *IJCNN*, pages 1–8. IEEE, 2020.

[20] Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive Neural Networks. *arXiv: 1606.04671 [cs]*, June 2016.

[21] Federico Giannini, Giacomo Ziffer, and Emanuele Della Valle. cPNN: Continuous Progressive Neural Networks for Evolving Streaming Time Series. In *PAKDD (4)*, volume 13938 of *Lecture Notes in Computer Science*, pages 328–340. Springer, 2023.

[22] Rishi et al. Bommasani. On the Opportunities and Risks of Foundation Models, 2022.

[23] Andrea Cossu, Antonio Carta, Lucia Passaro, Vincenzo Lomonaco, Tinne Tuytelaars, and Davide Bacciu. Continual pre-training mitigates forgetting in language and vision. *Neural Networks*, 179:106492, 2024.

[24] Giacomo Cignoni, Andrea Cossu, Alexandra Gómez Villa, Joost van de Weijer, and Antonio Carta. Replay-free online continual learning with self-supervised multipatches. In *ESANN Proceedings*, 2025.

[25] Ayush K. Tarun, Vikram S. Chundawat, Murari Mandal, and Mohan Kankanhalli. Fast yet effective machine unlearning. *IEEE Transactions on Neural Networks and Learning Systems*, 35(9):13046–13055, 2024.

[26] Alexander Krawczyk and Alex Gepperth. Continual unlearning through memory suppression. In *ESANN Proceedings*, 2025.

[27] Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. Fast Model Editing at Scale. In *International Conference on Learning Representations*, October 2021.

[28] Akshat Gupta, Anurag Rao, and Gopala Anumanchipalli. Model editing at scale leads to gradual and catastrophic forgetting. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 15202–15232. Association for Computational Linguistics, 2024.

[29] Zhiqiu Lin, Jia Shi, Deepak Pathak, and Deva Ramanan. The CLEAR Benchmark: Continual LEArning on Real-World Imagery. In *Thirty-Fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, January 2022.

[30] Huaxiu Yao, Caroline Choi, Bochuan Cao, Yoonho Lee, Pang Wei W Koh, and Chelsea Finn. Wild-time: A benchmark of in-the-wild distribution shift over time. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 10309–10324. Curran Associates, Inc., 2022.

[31] Eli Verwimp, Kuo Yang, Sarah Parisot, Lanqing Hong, Steven McDonagh, Eduardo Pérez-Pellitero, Matthias De Lange, and Tinne Tuytelaars. Clad: A realistic continual learning benchmark for autonomous driving. *Neural Networks*, 161:659–669, 2023.

[32] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[33] Zhipeng Cai, Ozan Sener, and Vladlen Koltun. Online Continual Learning with Natural Distribution Shifts: An Empirical Study with Visual Data. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8261–8270, Montreal, QC, Canada, October 2021. IEEE.

[34] Kristen Grauman et al. Ego-exo4d: Understanding skilled human activity from first- and third-person perspectives. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 19383–19400. IEEE, 2024.

[35] Hexiang Hu, Ozan Sener, Fei Sha, and Vladlen Koltun. Drinking From a Firehose: Continual Learning With Web-Scale Natural Language. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):5684–5696, May 2023.

[36] Joel Jang, Seonghyeon Ye, Changho Lee, Sohee Yang, Joongbo Shin, Janghoon Han, Gyeonghun Kim, and Minjoon Seo. Temporalwiki: A lifelong benchmark for training and evaluating ever-evolving language models. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 6237–6250. Association for Computational Linguistics, 2022.

[37] Lorenzo Iovine, Giacomo Ziffer, Andrea Proia, and Emanuele Della Valle. Towards streaming land use classification of images with temporal distribution shifts. In *ESANN Proceedings*, 2025.