Continual Reinforcement Learning in 3D Non-stationary Environments

Vincenzo Lomonaco  
University of Bologna  
Italy  
vincenzo.lomonaco@unibo.it

Karan Desai  
University of Michigan  
United States  
kdexd@umich.edu

Eugenio Culurciello  
Purdue University  
United States  
euge@purdue.edu

Davide Maltoni  
University of Bologna  
Italy  
davide.maltoni@unibo.it

Abstract

High-dimensional always-changing environments constitute a hard challenge for current reinforcement learning techniques. Artificial agents, nowadays, are often trained off-line in very static and controlled conditions in simulation such that training observations can be thought as sampled i.i.d. from the entire observations space. However, in real world settings, the environment is often non-stationary and subject to unpredictable, frequent changes. In this paper we propose and openly release CRLMaze, a new benchmark for learning continually through reinforcement in a complex 3D non-stationary task based on ViZDoom and subject to several environmental changes. Then, we introduce an end-to-end model-free continual reinforcement learning strategy showing competitive results with respect to four different baselines and not requiring any access to additional supervised signals, previously encountered environmental conditions or observations.

1. Introduction

In the last decade we have witnessed a renewed interest and major progresses in reinforcement learning (RL) especially due to recent deep learning developments [3]. State-of-the-art RL agents are now able to tackle fairly complex problems involving high-dimensional perceptual data, which were even unthinkable to solve without explicit supervision before [34, 45].

However, much of these progresses have been made in very narrow and isolated tasks, often in simulation with thousands of trials and with the common assumption of a stationary, fully-explorable environment from which to sample observations i.i.d. or approximately so. Even in the case of more complex tasks and large environments, a common technique known as memory replay [34, 16, 23] is adopted, consisting in storing old observations in an external memory buffer to simulate an i.i.d. sampling. Roughly the same result can be also achieved through multiple replicas of the agent randomly spawned in the environment and collecting several different observations at the same time, hence approximating the coverage of the entire observations space [33].

Nevertheless, dealing with single agents in the real-world and subject to computational and memory constraints these solutions suddenly appear less practical. This is especially true with always-changing environments and multi-task settings where re-sampling is impossible and storing old observations is no longer an option since it would require a constant grow in terms of memory consumption and computational power needed to re-process these observations. On the other hand, if the memory replay buffer is limited in size, the agent suddenly incurs in the phenomenon known in literature as catastrophic forgetting, being unable to re-
tain past knowledge and skills in previously encountered environmental conditions or tasks [31, 41, 13, 24].

Learning continually from data is a topic of steadily growing interest for the machine learning community and concerns itself with the idea of improving adaptation and generalization capabilities of current machine learning models by providing efficient updating strategies when new observations become available without storing, re-sampling or re-processing the previous ones (or as little as possible). While much of the focus and research efforts in continual learning have been devoted to multi-task settings (where a single model is exposed to a sequence of distinct and well-defined tasks over time) [36, 6], several practical scenarios would also benefit artificial agents that learn continually in complex non-stationary reinforcement environments.

In this paper, we focus on the more complex problem of a single task, constantly changing over time. As it has been shown in some supervised contexts, the clear separation in tasks (i.i.d. by parts), along with the presence of a supervised "task label" t [11], greatly helps taming the problem of forgetting [30, 2]. We argue that learning without any notion of task or distributional shift (both during training and inference), at least from an external oracle, is a more natural approach worth pursuing for improving the autonomy of every artificial learning agent.

The original contributions of this paper can be summarized as follows:

- We design and openly release a new benchmark, CRLMaze based on VizDoom [20], for assessing continual reinforcement learning (CRL) techniques in an always-changing object-picking task. CRLMaze is composed of 4 scenarios (Light, Texture, Object, All) of incremental difficulty and a total of 12 maps. To the best of our knowledge, this is one of the first attempts to scale continual reinforcement learning to complex 3D non-stationary environments.

- We provide 4 continual reinforcement learning baselines for each scenario.

- We propose an end-to-end, model-free continual reinforcement learning strategy, CRL-Unsup, which is agnostic to the changes in the environment and does not exploit a memory replay buffer or any distribution-specific over-parametrization, showing competitive results with respect to the supervised baselines (see section 4). The core insight of our strategy is to consolidate past memories through regularization as in [21], but proportionally to the difference between the expected reward and the actual reward (hence encoding a novel environmental condition in which the agent is unable to operate).

All the environments and the code to reproduce and expand the experiments discussed in this paper are available at: https://github.com/vlomonaco/crlmaze.

The rest of the paper is organized as follows: in Section 2, the CRLMaze benchmark is described; In Section 3, the CRL strategies used for the experiments reported in Section 4 are outlined. Finally, in Section 5, key questions and future work in this area are discussed.

2. CRLMaze: a 3D Non-stationary Environment

Continual Learning (CL) in reinforcement learning environments is still in its infancy. Despite the obvious interest in applying CL to less supervised settings and the early, promising results in this context [40, 48], reinforcement learning tasks constitute a much more complex challenge where it is generally more difficult to disentangle the complexity introduced by distributional shifts from those introduced by the lack of a strong supervision.

It is also worth noting that state-of-the-art reinforcement learning algorithms and current hardware computational capabilities do not make experimentation and prototyping easily accomplished on complex environments where physical simulation constitute an heavy computational task per se. In a continual learning context, the problem becomes even harder since an exposition of the same model to sequential streams of observations is needed (and cannot be parallelized by definition). This is why recent reinforcement learning algorithms for continual learning have been tested only on arguably simple tasks of low/medium input space dimension and complexity [21, 1, 35].

Nevertheless, at the same time, state-of-the-art reinforcement learning algorithms have started tackling more complex problems in 3D static environments. VizDoom [20], followed soon after by other research platforms like DeepMind Labs [4] and Malmo [17], allowed researchers to start exploring new interesting research directions with the aim of scaling up current reinforcement learning algorithms.

VizDoom is a reinforcement learning API build around the famous ZDoom game engine and providing all the necessary utilities to train a RL agent in arbitrary complex environments. This framework is particularly interesting since it has been open-sourced to both Windows and Unix systems and it was already built on the idea of flexibility and customizability, allowing users to create custom maps and modify behavioral responses of the environment through the simple Action Code Script (ACS) language.

In this paper, we propose an original 3D ViZDoom environment for continual reinforcement learning and an object-picking task named CRLMaze\(^1\) (see Fig. 1). The task consists of learning how to navigate in a complex maze and

\(^1\)In particular, we used Slade3 as the environment editor.
pick up “column bricks” while avoiding “flaming lanterns” (see Fig. 2). However, the environment in this case is non-stationary meaning that is subject to several environmental changes leading to major difficulties for standard reinforcement learning algorithms.

For properly assessing novel continual reinforcement learning strategies in the aforementioned 3D complex environment we split the benchmark in four different scenarios of incremental difficulty with respect to different environmental changes (see Fig. 2):

- **Light**: In this scenario the illumination of the environment is altered over time. While intuitively this scenario may appear as one of the easiest, as we will see in the experimental section 4, it constitutes one of the most difficult since visual features from the environment do not change only in terms of RGB pixel magnitudes by a scalar factor, but also in terms of agent visibility (i.e. the radius in the 3D space up to which the RGB colors saturates to complete black), as shown in Fig. 2 (top row).

- **Texture**: In this scenario walls textures are changed over time. The ability to pass over invariant features of the background is often taken for granted in many supervised tasks with state-of-the-art deep architectures [38]. However, as shown in the past, reinforcement learning agents are quite fragile also with respect to minor environmental changes [18].

- **Object**: In this scenario the shapes and colors of the objects are changed over time. Invariance with respect to object shapes and colors is another important property every learning system should possess when facing real-world conditions where surrounding objects appearances are subject to constant changes due to deterioration and substitutions.

- **All**: In this environment lights, textures as well as objects are subject to change over time. This scenario is also proposed with the idea of providing a comprehensive scenario for 3D environments in complex non-stationary settings, combining all the environmental condition variations proposed in the previous ones.

For all the scenarios, changes are not gradual but happening at three specific points equidistant in time (the total number of training episodes is fixed and considered a property of the environment) and practically implemented as different ZDoom maps faced sequentially ($M_1 \rightarrow M_2 \rightarrow M_3$, see Fig. 2). The agent is randomly spawned at fixed positions depicted as white points in Fig. 1 with a random visual angle. The environment starts with 75 randomly spawned column objects and 50 lantern objects. Catching a column increases the reward of 100 once collected while touching a lantern decreases the reward of 200. Even if in our exploratory experiments we noted that a shaping reward is not necessary to train the agents up to convergence, a weak shaping reward of 0.7 has been added when the go-forward action is chosen to improve environment exploration and ultimately speed-up learning convergence. A new object for each category is also randomly spawned every 3 ticks for roughly maintaining the amount of objects in the environment stable as when the objects are collected by the agent they disappear.
Table 1. Some common environments used for continual or meta reinforcement learning. The proposed benchmark, CRLMaze (bottom), shows significant advancements in terms of task complexity and non-stationary elements.

<table>
<thead>
<tr>
<th>Environment Name</th>
<th>Input Dim.</th>
<th>3D</th>
<th>Non-Stationary Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Locomotion Environment [1]</td>
<td>14</td>
<td>yes</td>
<td>2 over 16 joints torques are scaled down by a constant factor.</td>
</tr>
<tr>
<td>MiniGrid [7]</td>
<td>6×6×3</td>
<td>no</td>
<td>Vertical velocity of pellet increased of 0.03 from default 0.608.</td>
</tr>
<tr>
<td>Catcher [39]</td>
<td>256×256×3</td>
<td>no</td>
<td>Pipe gap decreased 5 from default 100.</td>
</tr>
<tr>
<td>Flappy Bird [39]</td>
<td>288×512×3</td>
<td>no</td>
<td>Randomly generated worlds from the same distribution.</td>
</tr>
<tr>
<td>Krazy World [46]</td>
<td>10×10×3</td>
<td>no</td>
<td>Randomly generated mazes from the same distribution.</td>
</tr>
<tr>
<td>Mazes [46]</td>
<td>20×20</td>
<td>no</td>
<td>60 different games available in the platform.</td>
</tr>
<tr>
<td>Arcade Learning Environment [29]</td>
<td>210×160×3</td>
<td>no</td>
<td>Randomly generated maps with 3 levels of difficulty.</td>
</tr>
<tr>
<td>CoinRun [10]</td>
<td>64×64×3</td>
<td>no</td>
<td>Randomly generated maps from the same distribution.</td>
</tr>
<tr>
<td>CoinRun Platform [10]</td>
<td>64×64×3</td>
<td>no</td>
<td>Randomly generated mazes of different sizes.</td>
</tr>
<tr>
<td>RandomMazes [10]</td>
<td>[3×3 - 25×25]</td>
<td>no</td>
<td>Light/visibility, walls textures, object shape and colors are changed within the same object picking task.</td>
</tr>
<tr>
<td>CRLMaze</td>
<td>320×240×3</td>
<td>yes</td>
<td></td>
</tr>
</tbody>
</table>

In Tab. 1, we report some of the common environments and platforms used in the context of continual and meta-reinforcement learning and compare them with the proposed CRLMaze.

While still acknowledging the limited number of environmental variations introduced in the benchmark, we believe CRLMaze shows a significant advancement in terms of task complexity and non-stationary elements introduced with respect to recent environments made available by the community, being them usually of low input dimensionality, not often running on a complex 3D engine and with very limited non-stationary dynamics. For example, in [39], the Catcher environment is based on simple 2D physics and the only non-stationary element introduced is a change in the vertical velocity of the pallet which is only slightly increased of 5% from its default value.

3. Continual Reinforcement Learning Strategies

Learning over complex and large non-stationary environments is a hard challenge for current reinforcement learning systems. Recent works in this research area include meta-learning [12, 35, 1], hierarchical learning [50, 49] and continual learning approaches [21, 43, 37]. While both meta-learning and hierarchical learning work around the idea of imposing some structural dependencies among the learned concepts, continual learning is generally agnostic with this regard, being more focused on addressing the non-stationary nature of the underlying distributions [36].

Consolidating and preserving past memories while being able to generalize and learn new concepts and skills is a well known challenge for both artificial and biological learning systems, generally acknowledged as the plasticity-stability dilemma [32]. Since gradient-based architectures are generally skewed towards plasticity and prone to catastrophic forgetting, much of the research in continual learning with deep architectures has been devoted to the integration of consolidation processes in order to improve stability [6, 14].

However, the general focus of continual reinforcement learning research has been devoted to multi-task scenarios [21, 43] where consolidation can be achieved more easily and only when there is a change of task. CRLMaze constitutes a step forward in the evaluation of new continual reinforcement learning strategies that have to deal with substantial, unpredictable changes in the environment within the same task and without any additional supervised signal indicating (virtual or real) shifts in the underlying input-output distribution. We regard at this situation as the most realistic (and difficult) setting every agent should be able to deal within real-world conditions, where learning is mostly unsupervised and autonomous.

In recent literature, this problem has been tackled by using external generative models of the environment in order to detect big changes in input space [21]. However, recent evidences in behavioral experiments on rats suggests, more generally, behavioral correlates of synaptic consolidation especially when the subject is exposed to novel or strong external stimuli (e.g. a foot shock) [8, 9]. Following this inspiration, in this paper we propose a new strategy CRLUnsup, where the central idea is to consolidate memory only when a substantial difference between the expected reward and the actual one is detected, i.e. when the agent encounters an unexpected situation.

Hence, distributional shifts can be detected just by looking at the ability of the agent to actually perform the task: this can be approximated and practically implemented as the difference between a short-term \( r_{\text{mavg}} \) and a long-term \( r_{l\text{mavg}} \) reward moving average that, when goes under a particular threshold (\( \eta \)), triggers the memory consolidation procedure\(^2\). The long-term moving average encodes the ex-

\(^2\) It is interesting to note that a similar technique is also the basis of the MACD indicator [43] widely used in automated trading systems to detect
Figure 3. Short and long-term moving average (computed over 6 and 50 episodes, respectively) of the average cumulative reward during training in a single run of the light scenario. Dotted lines indicate when the environment is changed. In this example, the difference between the short-term and long-term moving average goes under $\eta = -70$ when the environment changes.

For consolidation in CRL-Unsup we employ the end-to-end regularization approach firstly introduced in [21] and known as Elastic Weight Consolidation (EWC): the basic idea is to preserve the parameters proportionally to their importance in the approximation of a specific distribution (i.e. the Fisher information). More efficient consolidation techniques through regularization derived from EWC have been recently proposed [43, 51, 30, 27]. However, for simplicity, we used its basic implementation. In eq. 1 and eq. 2 the loss function $L$ of the CRL-Unsup strategy for a single consolidation step is reported$^3$ where $L_{A2C}$ is the standard A2C loss function composed of the value and policy loss as defined in [33]; $\lambda$ is an hyper-parameter encoding the strength of the consolidation (i.e. reducing plasticity); $F_k$ is the Fisher information for the weight $\theta_k$ while $\theta^*_k$ indicates the optimal weight to consolidate. However, since the Fisher information can not be computed on the new data distribution, $F$ is computed at fixed steps in times and only the latest one is used in the regularization term when the threshold is exceeded.

$$L = L_{A2C} + \frac{\lambda}{2} \cdot \sum_k F_k (\theta_k - \theta^*_k)^2$$  \hspace{1cm} (1)

$$\lambda = \begin{cases} \alpha & \text{if } r^s_{mavg} - r^l_{mavg} \leq \eta \\ 0 & \text{otherwise} \end{cases}$$  \hspace{1cm} (2)

In Figure 3, an example in the light scenario of the short and long-term moving average (computed over 6 and 50 episodes, respectively) of the training average cumulative reward is reported. In order to better compare and understand the performance of the aforementioned strategy, on each of the considered environmental changes (i.e. light, texture, changing market conditions and issue buy/sell signals.

$^3$Please note that in the basic EWC implementation a regularization term was not used.

Please note that in the basic EWC implementation a regularization term was not used.
object, all) four different baselines are here introduced and assessed:

1. **Multienv**: This approach can be considered as a reference baseline and not properly a CRL strategy since it consists in training the agent over all the possible environmental conditions (i.e., maps $M_1$, $M_2$ and $M_3$) of each scenario at the same time. Having access to all the maps at the same time makes the distribution stationary and eliminates the catastrophic forgetting problem. It will be considered as an upper bound for the other strategies as generally acknowledged in continual learning [36, 26, 28].

2. **CRL-Naive**: This approach, like the homonym strategy in the supervised context [26, 30], consists in just continuing the learning process without variations and independently w.r.t. the changes in the environment. Learning through reinforcement in complex non-stationary environments without any memory replay is known to suffer from catastrophic forgetting, instability and convergence difficulties while learning. This strategy is usually considered as a lower bound.

3. **CRL-Sup**: This approach can be considered as a second baseline in which the distributional shift supervised signal (i.e., when the map changes) is actually provided to the model for memory consolidation purposes. In this case the standard application of EWC with the loss described in eq. 1 is performed but is perfectly synchronized with the end of the training on each map.

4. **CRL-Static**: In this strategy, the memory is consolidated (i.e., the regularization term added) at fixed steps in time, independently of the changes in the environment. As we will see in the experiments results, this may be very difficult to tune and rather inefficient, depending on the memory consolidation technique used. In fact, when learning from scratch an early and “blind” consolidation of memory may also hurt performance and actually hamper the ability of learning in the future.

### 4. Experiments and Results

For all the experiments we use a simple batched-A2C with synchronous updates [47], but only within the same map (the actual environment with fixed static settings), so that when the map changes the model cannot access in any way previous environmental conditions. The architecture of the agent used for these experiments is a plain 3-layers ConvNet ($3 \times 3$ kernels with 32 feature maps each) with ReLU activations, followed by a fully connected layer encodings the three possible actions $A = \{\text{turn-left, turn-right and move-forward}\}$

Each training and test episode has a fixed runtime of 1000 ticks. However, the agent is allowed to make an action every 4 frames, maintaining the action chosen based on the first frame fixed for the other three. This allows a smoother interaction with the environment and allows to the agent to not stall in ambiguous situations even if completely stateless (e.g., in front of a wall).

For the batched-A2C implementation, the synchronous gradient update takes place every 20 frames (covering 80 ticks of the total 1000 ticks of the full episode length) and 20 different agents are spawned in parallel in 20 VizDoom instances of the same environment. The discount factor is fixed to $\gamma = 0.99$ for all the environments. More details about the experimental procedure, implementation details and all the hyper-parameters used are available in the section 4.1.

In order to evaluate and compare the performance of each strategy we use the $A$ metric, defined in [11] as an extension of [28]. Performance are evaluated at the end of the training on each map $M_i$ on 300 testing episodes, 100 for each different map $M_j$, even the ones not already encountered. Given the test cumulative reward matrix $R \in \mathbb{R}^{3 \times 3}$, which contains in each entry $R_{i,j}$ the test episodes average cumulative reward of the model on map $M_j$ after observing the last training episode from map $M_i$; $A$ can be defined as follows:

$$A = \frac{\sum_{i,j} N R_{i,j}}{N(N+1)}$$

where $N = 3$ and $A$ is essentially the average of the lower triangular matrix of $R$, which roughly encodes how the model is performing on the current environmental change and the already encountered ones, on average. In Table 2 the average cumulative reward and the $A$ metric results for the Light, Texture, Object, All scenarios and the 4 different CRL strategies is reported. It is worth noting that, the scenarios difficulty can vary substantially from a cumulative reward average of ~200 for the agents trained in the light scenario to ~600 for the Object one, which turns out to be the easiest one in our experiments.

By considering the average $A$ metric across all the scenarios for each strategy (at the bottom of Table 2) or in the last column of Fig. 4, it is possible to compare the different strategies independently of the peculiarities of each specific scenario. In this case we can observe how the CRL-Sup strategy constitutes, as we would expect, the best approach in terms of absolute $A$ performance. However, the proposed CRL-Unsup strategy, while not exploiting any additional supervised signal, reasonably approximates its performance with a gap of ~50 cumulative reward points. The CRL-Static and CRL-Naive approaches perform similarly on the $A$ metric, but while the CRL-Naive approach is almost consistently better on the last map $M_3$ at the end of the training, it seems
Table 2. Average cumulative reward matrix $R$ and $A$ metric result for each scenario and CRL strategy. Results highlighted in black and blue represent the best and the second best performing strategies on each scenario. A gray background is used in the cells involved in the computation of the $A$ metric. Results are computed over 10 runs for each strategy and benchmark for a total of 160 runs.

<table>
<thead>
<tr>
<th></th>
<th>CRL-Naive</th>
<th>CRL-Sup</th>
<th>CRL-Static</th>
<th>CRL-Unsup</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$M_1$</td>
<td>$M_2$</td>
<td>$M_3$</td>
<td>$M_1$</td>
</tr>
<tr>
<td>Light</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$M_1$</td>
<td>93</td>
<td>-460</td>
<td>144</td>
<td>283</td>
</tr>
<tr>
<td>$M_2$</td>
<td>-987</td>
<td>528</td>
<td>-842</td>
<td>-236</td>
</tr>
<tr>
<td>$M_3$</td>
<td>-892</td>
<td>1063</td>
<td>938</td>
<td>-232</td>
</tr>
<tr>
<td>A</td>
<td>123.96</td>
<td></td>
<td></td>
<td>181.97</td>
</tr>
<tr>
<td>Texture</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$M_1$</td>
<td>877</td>
<td>544</td>
<td>57</td>
<td>1196</td>
</tr>
<tr>
<td>$M_2$</td>
<td>-115</td>
<td>1360</td>
<td>504</td>
<td>-80</td>
</tr>
<tr>
<td>A</td>
<td>499.81</td>
<td></td>
<td>574.39</td>
<td></td>
</tr>
<tr>
<td>Object</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$M_1$</td>
<td>930</td>
<td>-974</td>
<td>-1005</td>
<td>1365</td>
</tr>
<tr>
<td>$M_2$</td>
<td>962</td>
<td>1045</td>
<td>-988</td>
<td>1160</td>
</tr>
<tr>
<td>$M_3$</td>
<td>-738</td>
<td>-214</td>
<td>1013</td>
<td>254</td>
</tr>
<tr>
<td>A</td>
<td>496.67</td>
<td></td>
<td>792.59</td>
<td></td>
</tr>
<tr>
<td>All</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$M_1$</td>
<td>1268</td>
<td>-1000</td>
<td>-1000</td>
<td>1579</td>
</tr>
<tr>
<td>$M_2$</td>
<td>-490</td>
<td>1346</td>
<td>-991</td>
<td>301</td>
</tr>
<tr>
<td>$M_3$</td>
<td>-764</td>
<td>-219</td>
<td>815</td>
<td>-389</td>
</tr>
<tr>
<td>A</td>
<td>325.88</td>
<td></td>
<td>464.04</td>
<td></td>
</tr>
</tbody>
</table>

Avg. A | 361.57 ± 154.18 | 503.24 ± 219.96 | 361.03 ± 116.30 | 449.47 ± 210.78

Figure 4. $A$ metric results for each CRL strategy and scenario. Dotted lines indicate the average cumulative reward on 300 test episodes for the multienv upper-bound. Better viewed in colors.

more sensitive to forgetting than the CRL-Static approach on previously encountered maps.

Results for each specific scenario roughly confirm this trend with exception of the light scenario (the most difficult) where a CRL-Static approach seems to prevail even the CRL-Sup one. We postulate that, in this case, a more frequent consolidation in the direction of the natural gradient as shown in [42] may help to stabilize learning in complex environments even within the same environmental conditions.

Finally, in Fig. 4, the average $A$ metric for each strategy is reported along with the Multienv upper bound. In this case the upper bound is not an $A$ metric but simply the average test cumulative reward (on all the test maps) of a agent trained simultaneously on the three environmental conditions of each scenario. It is worth noting the conspicuous gap w.r.t. the best performing continual reinforcement learning strategy of each scenario, suggesting the need of further research on CL approaches for RL.

4.1. Implementation Details

In this section additional details about the experiments are reported. All the code, environments and setup scripts to reproduce the experiments are openly released at: https://github.com/vlomonaco/crlmaze. In order to properly compare the performance of the proposed CRL strategies a total of 200 runs (10 for each CRL strategy and scenario) has been conducted for more than 40 hours of computation on a single machine with 32 CPU cores and 1 NVIDIA GTX Titan X.

Hyper-parameters used in the experiments are reported instead in Tab. 3. Hyper-parameters have been chosen for each strategy in order to maximize the $A$ metric at the end of each run. Parallel instances indicates the number of ViZDoom instances and agents running in parallel for the roll-outs always fixed to 20. Episode size is the number of frames (not considering the skip-rate of 4 as explained in section 4) after which a weights update is performed. The $r_{max}^s$ and $r_{max}^l$ size parameters represent instead the
number of training episodes to consider for the short and long-term moving average, respectively.

Focusing only on the strategies employing consolidation \( \eta, \alpha \) and \( \lambda \) are the parameters already described in section 4 while Fisher freq., Fisher clip and Fisher sample size represent respectively i) the computing frequency of the fisher matrix in terms of training episodes, ii) the clipping value of the importance magnitude as described in [30], and iii) the number of episodes used to estimate the fisher information of each weight.

For more details about the experiments please refer to the extended preprint [25].

5. Discussion and Conclusions

In this work we introduced and openly released a new environment and benchmark for easily assessing continual reinforcement learning algorithms on a complex 3D non-stationary environment. The preliminary experiments introduced in section 4 on four different scenarios and 5 different strategies show that the proposed unsupervised approach without any distributional shift supervised signal, external model or distribution-specific over-parametrization is not only possible but may be competitive with respect to a standard supervised counterpart.

However, as we observed in some experiments (where there is a noticeable gap between the CRL-Sup and CRL-Unsup strategies), the detection of a timely consolidation signal can be sometimes critical. For example, in case of positive forward transfer followed by a possible negative backward transfer [28] (i.e. being able to perform well on new conditions but impacting negatively on learned knowledge about the previous ones) memory consolidation cannot take place just by looking at the training cumulative reward curve since steadily growing. This problem may be tackled by looking at an additional regularization loss for reconstructing the input frame (or predicting the next one) since changes in the input space may be more evident. In this way, while still using a single end-to-end model and constructing more robust features [15, 22], it would be possible to integrate the benefit of both approaches when learning continuously.

In the future we plan to expand this work in several other directions. Firstly by moving towards a more flexible and more principled solution where the consolidation is proportional to the expected reward difference encoded directly in the loss function. Secondly by integrating more accurate synaptic plasticity models as shown in [19, 5] and going beyond mere consolidation processes which tend to quickly saturate the model learning capacity.

Finally, we plan to extend our evaluation where existing environmental changes are discretized by providing additional training maps for each category and by adding a new environmental change category where the size of the maze is substantially varied.

While still in their infancy we can foresee a new generation of reinforcement learning algorithms which can learn continually in complex non-stationary environments, opening the door to artificial learning agents which can autonomously acquire new knowledge and skills in unpredictable, real-word settings.
References


