Towards Continual No-Regret Learning

Anonymous authors
Paper under double-blind review

Abstract

Continual learning is a task in which a learning algorithm needs to constantly adapt. Modern reinforcement learning algorithms demonstrated strong performance across a large selection of problems. However, certain assumptions they make about the environment are violated in the continual learning setting. We thus turn to regret minimization algorithms, which have strong hindsight performance guarantees while making minimal assumptions about the environment. We present a novel framework which extends the guarantees of the regret minimizer to recent history. In particular, this allows it to model the impact of its own actions on the environment, and adapt accordingly. We combine our framework with regret minimizers which are able to work with continuous observations and maximize the expected reward. We can thus get the best of both worlds—an algorithm with strong hindsight guarantees which simultaneously maximizes expected reward akin to reinforcement learning. We study the advantages of our algorithm in small, illustrative environments.

1 Introduction

In the domain of *continual learning*, one seeks to find an algorithm that can continuously adapt to changes in the environment (Abel et al., 2023). This is challenging for the learner as the environment may be too complex for the agent to model accurately. Any sufficiently realistic scenario has this feature as, on top of many other sources of complexity, the environment also includes other learning agents. Moreover, the learner doesn't get to change its decision counterfactually—once an action is selected, there is no going back.

Reinforcement learning (RL) has seen wide-spread success in many domains, ranging from autonomous driving to training large language models (Wurman et al., 2022; Ouyang et al., 2022). In ergodic Markovian environments, RL algorithms provably converge to an optimal policy—one which maximizes the expected aggregated reward (Sutton et al., 1998). However, to do so RL needs to visit every state infinitely often, repeatedly starting in some initial state and exploring all the possible trajectories.

Moreover, realistic environments provide different feedback if different actions are used, breaking the Markovian property. This comes up naturally in the context of game theory, where the environment includes other learning agents. In this setting, RL is no longer guaranteed to converge, and fails to converge even in simple games (Nisan et al., 2007). Regret minimization has become a key building block of algorithms for solving games, including those with imperfect information and adversarial feedback.

Regret minimization algorithms have strong hindsight performance guarantees in any environment with bounded rewards (Blackwell et al., 1956). Informally, given a set of so-called *experts*, the regret minimization algorithm is guaranteed to asymptotically match the best expert in terms of average reward. A natural example of this *comparison class* are all experts who only use one fixed action. *Counterfactual regret minimization* (CFR) was introduced in order to efficiently extend regret minimization to sequential games (Zinkevich et al., 2007). The comparison class of CFR consists of all the experts who play a single action in a given state of the game. In other words, each expert follows a single edge in the game tree. By repeatedly traversing the whole game tree, CFR asymptotically finds a strategy in each node of the game tree. Following this strategy is guaranteed to dominate choosing any of the fixed experts while traversing the tree.

CFR leverages the regret minimization guarantees by providing additional context-based experts. This is straightforward in a sequential game where one can enumerate all the possible states. However, similar to RL, CFR effectively allows the agent to travel back in time and counterfactually change its strategy. In fact, one can show that minimizing the counterfactual regret is impossible without traversing the whole tree (Arora et al., 2012).[1]

In this paper, we introduce a novel framework in which the experts in the comparison class condition on recent history. In this way, the algorithm is able to model the effect of its own actions on the environment, and adapt appropriately. Moreover, it operates in a fully online 'single-lifetime' setting, never gaining any counterfactual feedback. We combine our algorithm with the *learning not to regret* framework and meta-learn the regret minimizer to maximize expected reward, while keeping the regret minimization guarantees (Sychrovský et al., 2024). We thus get the best of both worlds—an algorithm with strong hindsight guarantees which simultaneously maximizes expected reward akin to RL.

1.1 Related Work

Many recent papers argue that the Markovian property is fundamentally mismatched to continual learning, where the environment is vast and continually evolving. The "Big World" approach highlights the "small agent, big world" dilemma: an agent of bounded capacity cannot hope to discover, let alone optimize over, the entire state space (Kumar et al., 2024). For an agent, a problem is continual if the agent needs to constantly adapt and never settles on a given strategy (Ring, 1994; Abel et al., 2023). Empirically grounded benchmarks such as Jelly Bean World (Platanios et al., 2020) were introduced to stress-test agents in settings where distribution shift is gradual, tasks are unannounced, and the agent never "finishes" learning.

Hindsight rationality has been proposed in the context of continual learning before (Bowling and Elelimy, 2025). The authors argue for minimizing a notion of regret that compares the agent's performance to other policies applied along the trajectory the agent followed. They show that widely used RL algorithms do not minimize this notion of regret in continual learning tasks. Interestingly, simply using the past agent's policies already exceeds the performance of the RL agent. Our proposed framework CRM follows this line by allowing experts to condition on recent history. This strikes a middle ground: richer than static-action experts, yet tractable without full environment resets or perfect simulators. By proving regret bounds against this history-aware class we address the conceptual critiques above while respecting the impossibility of full-policy regret in a one-shot, non-Markovian world.

2 Preliminaries

2.1 Regret Minimization

An **online algorithm** m for the regret minimization task repeatedly interacts with an **environment** through available actions $\mathcal{A}$. The goal of a regret minimization algorithm is to maximize its hindsight performance, i.e., to minimize regret. Importantly, in the regret minimization framework, one is not restricted in the nature of the environment.

Formally, at each step $t \in \mathbb{N}$, the algorithm submits a **strategy** $\boldsymbol{\sigma}^t \in \Delta^{|\mathcal{A}|}$. Subsequently, it observes the expected **reward** $\boldsymbol{x}^t \in \mathbb{R}^{|\mathcal{A}|}$ for each of the actions from the environment, which depends on the strategy in the rest of the game. The difference in reward obtained under $\boldsymbol{\sigma}^t$ and any fixed action

[1] For example, consider a binary choice with one good and one bad outcome. Without having access to the other outcome, the agent is unable to minimize the counterfactual regret, as that would require the agent to choose the good outcome.

strategy is called the **instantaneous regret** $\boldsymbol{r}(\boldsymbol{\sigma}^t, \boldsymbol{x}^t) = \boldsymbol{x}^t(\boldsymbol{\sigma}^t) - \langle \boldsymbol{\sigma}^t, \boldsymbol{x}^t(\boldsymbol{\sigma}^t)\rangle \mathbf{1}$. The **cumulative regret** throughout time T is $\boldsymbol{R}^T = \sum_{t=1}^{T} \boldsymbol{r}(\boldsymbol{\sigma}^t, \boldsymbol{x}^t)$.

The goal of a regret minimization algorithm is to ensure that the regret grows sublinearly for any sequence of rewards. One way to do that is for m to select $\boldsymbol{\sigma}^{t+1}$ proportionally to the positive parts of $\boldsymbol{R}^t$, known as *regret matching* (Blackwell et al., 1956). This algorithm guarantees that the **external regret** $R^{\text{ext},T} = |\boldsymbol{R}^T|_\infty$ satisfies $R^{\text{ext},T} \in O(\sqrt{T})$ where $|\boldsymbol{R}^T|_\infty = \max_{a \in \mathcal{A}} \boldsymbol{R}^T(a)$.

Instead of thinking of the action the regret minimization algorithm chooses from as direct environment interactions, one can treat them as **experts**. The task of a regret minimizer is then to continuously adapt a distribution over the experts in such a way that no single expert is asymptotically strictly better. The set of all such experts forms a so-called **comparison class** of the regret minimization algorithm.

3 Continual Learning Framework

We begin by formalizing the interaction of the agent with the environment ε. Let $\mathcal{A}$ and $\mathcal{O}$ be sets of fixed size. We refer to $a \in \mathcal{A}$ as **actions**, and $o \in \mathcal{O}$ as **observations**. A **history** h^s of length $s \in \mathbb{N}_0$ is a sequence of s action-observation pairs. We define the set of all histories as

$$\mathcal{H} = \bigcup_{s=0}^{\infty} (\mathcal{A} \times \mathcal{O})^s.$$

For $L \in \mathbb{N}_0$, we use $\mathcal{H}^{\leq L} = \bigcup_{s=0}^{L} (\mathcal{A} \times \mathcal{O})^s$ to denote the set of histories of length at most L. Finally, for $l \in \mathbb{N}$, we define ***l*-suffix** $\mathcal{S}_l(h^s)$ of a history h^s as the last l action-observation pairs in h^s.

Definition 1. *An environment $\varepsilon : \mathcal{A} \times \mathcal{H} \to \Delta(\mathcal{O})$ maps actions and histories to a distribution over observations. The environment has an associated reward function $x_\varepsilon : \mathcal{A} \times \mathcal{O} \times \mathcal{H} \to [-1, 1]$.*

This definition subsumes the Markov decision process (MDP), which is typically used by reinforcement learning algorithms (Sutton et al., 1998). These algorithms are often trained to maximize some aggregate reward, such as the time-average, or discounted future reward known as return. When the environment is oblivious, maximizing reward is indeed identical to minimizing regret (Lattimore and Szepesvári, 2018). However, in general, policy gradient algorithms are not able to minimize regret, see Section 5 for an example.

4 Continual No-Regret Learning

A straightforward way to introduce no-regret learning is to simply use a regret minimization algorithm along the sequence of interactions with the environment. This would guarantee that we find a strategy that would do at least as well as any of the fixed actions in hindsight. Depending on the application, this guarantee may not be particularly strong as one action typically does not dominate the other. Additionally, this agent does not take the history into consideration when choosing the next strategy.

Instead, we want to minimize regret with respect to all histories of length up to $L \in \mathbb{N}_0$. Let h^{t-1} be the history of length $t-1$ that we see at step t in the environment. To this end, we define cumulative regret after $T \in \mathbb{N}$ steps conditioned on $h \in \mathcal{H}^{\leq L}$ as

$$\boldsymbol{R}^T_{|h} = \sum_{t=1}^{T} \boldsymbol{r}(\boldsymbol{\sigma}^t, \boldsymbol{x}^t)\delta(h = \mathcal{S}_{|h|}(h^{t-1})),$$

where $\delta(h = \mathcal{S}_{|h|}(h^{t-1}))$ is the indicator function of h being the $|h|$-suffix of h^{t-1}. For $T = 0$, we define $\boldsymbol{R}^0_{|h} = \mathbf{0}$. Informally, our proposed algorithm internally uses a regret minimizer m. It uses m to ensure vanishing history-conditioned regret in terms of the number of times the history has been encountered. We refer to this family of algorithms as *conditional regret minimization* (CRM).

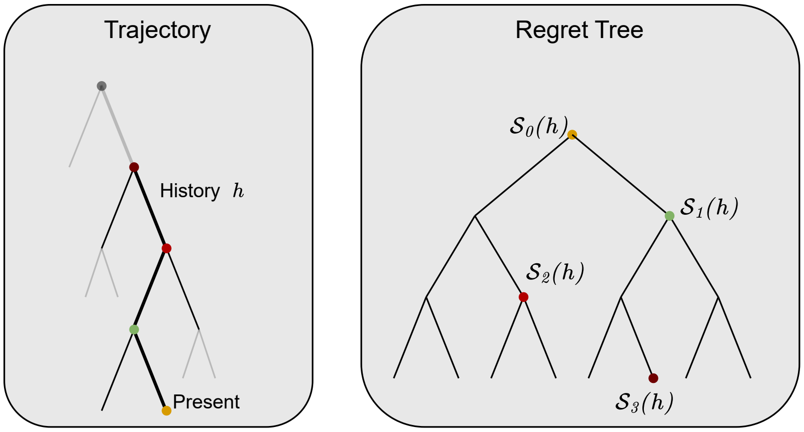

Figure 1: Example of a history h (left) and the associated nodes in the regret tree of the CRM algorithm with depth $L = 3$ (right) in an environment with $|\mathcal{A}| = 2$ actions. For each l-suffix of h, we find the associated node in the regret tree by following the l-suffix from the root. Note the l-suffixes of h are not hierarchical subsets, and the associated nodes are thus not in consecutive sub-trees.

Algorithm 1: Conditional Regret Minimization

```
Input: Regret minimizer $m$, $L \in \mathbb{N}_0$
Initialize: Regret tree $\{\boldsymbol{R}^0_{|h} \leftarrow \mathbf{0} \in \mathbb{R}^{|\mathcal{A}|} : h \in \mathcal{H}^{\leq L}\}$
function NextStrategy($h^{t-1}$)
    $\boldsymbol{R}^t_{\mathcal{S}}(a) \leftarrow \max_{l \in \{0,\ldots,L\}} \boldsymbol{R}^t_{|\mathcal{S}_l(h^{t-1})}(a)$ for every $a \in \mathcal{A}$
    $\boldsymbol{\sigma}^t \leftarrow m(\boldsymbol{R}^{t-1}_{\mathcal{S}})$
function ObserveReward($\boldsymbol{x}^t$, $h^{t-1}$)
    for $l \in \{0, \ldots, L\}$
        $\boldsymbol{R}^t_{|\mathcal{S}_l(h^{t-1})} \leftarrow \boldsymbol{R}^{t-1}_{|\mathcal{S}_l(h^{t-1})} + \boldsymbol{r}(\boldsymbol{\sigma}^t, \boldsymbol{x}^t)$
```

Formally, at each step t, given the history h^{t-1}, CRM chooses a strategy with respect to a regret $\boldsymbol{R}^t_{\mathcal{S}}$ where, for every $a \in \mathcal{A}$,

$$\boldsymbol{R}^t_{\mathcal{S}}(a) = \max_{l \in \{0,\ldots,L\}} \boldsymbol{R}^t_{|\mathcal{S}_l(h^{t-1})}(a). \tag{1}$$

After observing the reward, it accumulates the instantaneous regret to all nodes of the regret-tree given by $\mathcal{S}_l(h^{t-1}), l \in \{0, \ldots, L\}$; see Algorithm 1 and Figure 1. Note that the reward is $\mathbf{0}$ if h is not a $|h|$-suffix of h^{t-1}. Therefore, we only need to accumulate regret for the l-suffixes of the current history.

It is not clear at first which notion of regret CRM minimizes. This is because we are only updating the regret of the experts corresponding to the l-suffixes of the current history. However, we can show that for the histories that the agent visits at a non-vanishing rate, the associated regret is minimized.

Theorem 1 (Correctness of CRM)**.** *Let $L \in \mathbb{N}_0$, $h \in \mathcal{H}^{\leq L}$ be a history and $C^T_{|h}$ be the number of times h was an $|h|$-suffix of a history of CRM until step $T \in \mathbb{N}$. Then, for any sequence of rewards, the regret of CRM conditioned on h satisfies $|\boldsymbol{R}^T_{|h}|_\infty \in O(\sqrt{T})$. Furthermore,*

$$|\boldsymbol{R}^T_{|h}|_\infty \in \Omega(C^T_{|h}) \implies C^T_{|h} \in O(\sqrt{T}).$$

In particular, the visit rate $C^T_{|h}/T$ goes to zero as $T \to \infty$.

Proof. We proceed by induction over the step number T. For the induction base, we have $T = 0$ and $\boldsymbol{R}^0_{|h} = \boldsymbol{0}$ for every $h \in \mathcal{H}^{\leq L}$, and thus $|\boldsymbol{R}^T_{|h}|_\infty \in O(\sqrt{T})$. For the induction step, our induction hypothesis gives $|\boldsymbol{R}^T_{|h}|_\infty \in O(\sqrt{T})$ for every $h \in \mathcal{H}^{\leq L}$. This implies $|\boldsymbol{R}^T_{\mathcal{S}}|_\infty \in O(\sqrt{T})$, since each coordinate of $\boldsymbol{R}^T_{\mathcal{S}}$ achieves a maximum for some history from $\mathcal{H}^{\leq L}$. CRM chooses the strategy $\boldsymbol{\sigma}^{T+1}$ with respect to $\boldsymbol{R}^T_{\mathcal{S}}$ such that for any reward $\boldsymbol{x}^{T+1} \in [-1,1]^{|\mathcal{A}|}$ we have $|\boldsymbol{R}^T_{\mathcal{S}} + \boldsymbol{r}(\boldsymbol{\sigma}^{T+1}, \boldsymbol{x}^{T+1})|_\infty \in O(\sqrt{T+1})$. This follows from the properties of the regret minimizer m which CRM internally uses. Our choice of $\boldsymbol{R}^T_{\mathcal{S}}$ gives $\boldsymbol{R}^T_{\mathcal{S}}(a) \geq \boldsymbol{R}^T_{|h}(a)$ for every $a \in \mathcal{A}$ and $h \in \{\mathcal{S}_l(h^{T-1})\colon l \in \{0, \ldots, L\}\}$. Thus, we have,

$$|\boldsymbol{R}^{T+1}_{|h}|_\infty = |\boldsymbol{R}^T_{|h} + \boldsymbol{r}(\boldsymbol{\sigma}^{T+1}, \boldsymbol{x}^{T+1})|_\infty \leq |\boldsymbol{R}^T_{\mathcal{S}} + \boldsymbol{r}(\boldsymbol{\sigma}^{T+1}, \boldsymbol{x}^{T+1})|_\infty \in O(\sqrt{T+1})$$

for every $h \in \{\mathcal{S}_l(h^{T-1})\colon l \in \{0, \ldots, L\}\}$. For the histories $h \in \mathcal{H}^{\leq L}$ which are not a suffix of h^{T-1}, we have $\boldsymbol{R}^{T+1}_{|h}(a) = \boldsymbol{R}^T_{|h}(a)$ for every $a \in \mathcal{A}$, which in combination with the induction hypothesis implies

$$|\boldsymbol{R}^T_{|h}|_\infty = |\boldsymbol{R}^{T+1}_{|h}|_\infty \in O(\sqrt{T}).$$

Altogether, we obtain $|\boldsymbol{R}^{T+1}_{|h}|_\infty \in O(\sqrt{T+1})$ for every history $h \in \mathcal{H}^{\leq L}$.

In terms of the visit count $C^T_{|h}$ of each history h, we can thus accumulate even superlinear regret in the number of visits, but only if the visit count grows sufficiently slowly. In particular, since $|\boldsymbol{R}^T_{|h}|_\infty \in O(\sqrt{T})$ and since we are assuming $|\boldsymbol{R}^T_{|h}|_\infty \in \Omega(C^T_{|h})$, we obtain $C^T_{|h} \in O(\sqrt{T})$. □

Corollary 1. *Since the root of the regret tree, corresponding to the empty history, is a suffix of any h^T, the algorithm minimizes the standard external regret, similar to applying a regret minimizer along the trajectory.*

4.1 Incorporating Policy Gradient

While CRM can work with any regret minimization algorithm m, one particular interesting class is the neural predictive regret matching (NPRM) (Sychrovský et al., 2024). NPRM is an extension of predictive regret matching (Farina et al., 2021), which employs a predictor about future rewards. More precise predictions provable result in lower regret (Farina et al., 2021). However, importantly, the algorithm maintains regret minimization guarantees for *arbitrary* predictions. NPRM uses predictor parametrized by a neural network. This allows NPRM to work with arbitrary additional, even continuous, context—something most other regret minimization algorithms cannot do. When the predictor is trained to minimize the external regret $R^{\text{ext},T}$, then it is equivalent to maximizing the expected reward $\sum_{t=1}^{T} \langle \boldsymbol{\sigma}^t, \boldsymbol{x}^t \rangle$ (Sychrovský et al., 2024). We refer to CRM which uses NPRM as the regret minimizer m and trains to maximize the reward as NPCRM.

5 Experiments

We illustrate the differences between RL, un-contextual regret minimization, and CRM in a set of small non-Markovian learning tasks. In particular, we use the *policy gradient* (PG) algorithm (Sutton et al., 1998) to represent RL, and *regret matching* (RM) (Blackwell et al., 1956) as a regret minimizer. Both CRM and NPCRM use the depth of the regret tree $L = 3$. All algorithms are trained online with full-information feedback $\boldsymbol{x}^t$ about every expert without resetting the environment. After committing to a strategy, the environment uses it to sample an expert, which it uses to transition to the next state.

5.1 Multi-Agent Environment

First, we show that RL fails to minimize regret in a notorious example of a two-player zero-sum game. We use the rock-paper-scissors game where the opponent changes her strategy according

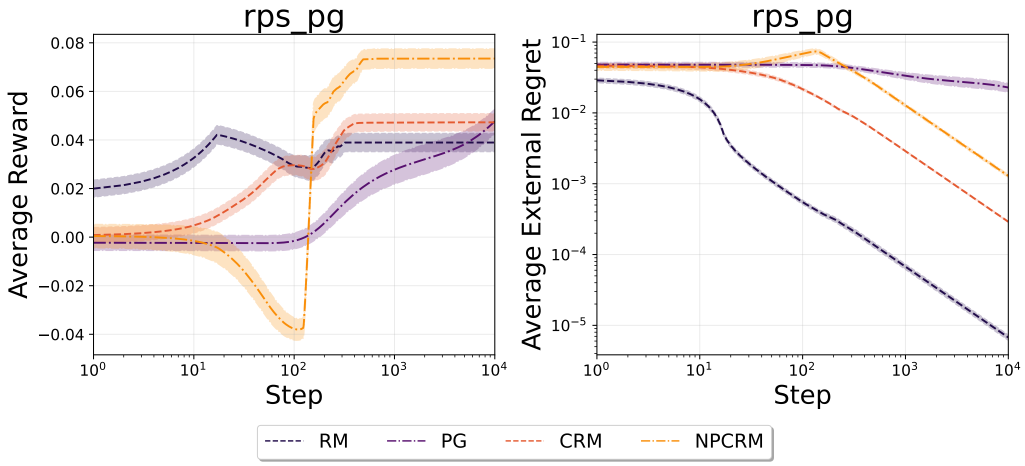

Figure 2: Comparison of the average reward and external regret of RM, PG, CRM, and NPCRM in `rps_pg`, see Section 5.1. In contrast to the regret minimization algorithms, the average external regret of RL remains high even after many environment steps.

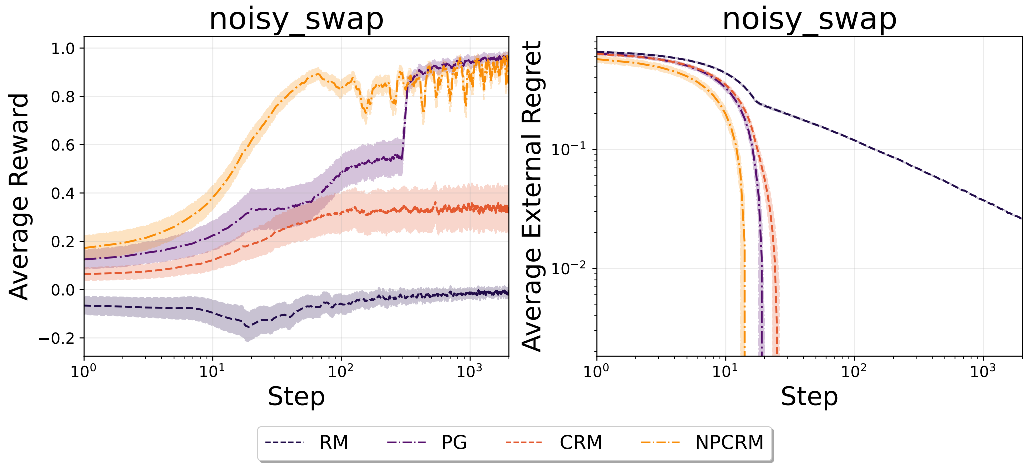

Figure 3: Comparison of the average reward and external regret of RM, PG, CRM, and NPCRM in `noisy_swap`, see Section 5.2. Both PG and NPCRM can achieve near-optimal average reward thanks to their ability to process observations.

to PG. We refer to this environment as `rps_pg` and compare performance of each algorithm in Figure 2. Unlike the remaining algorithms, PG fails to minimize regret. In this case, the regret minimization algorithms also perform better in terms of the average reward.

5.2 Continuous Observations

An advantage of PG is that it can efficiently work with large, or even continuous, observation spaces. In contrast, conditioning on such infinite space might result in no history being encountered more than once. We illustrate this advantage. We consider a simple environment where the observations are drawn from $\mathcal{U}(0, 1)$. The agent receives +1 reward for using the action with the index equal to the rounded observation, and -1 otherwise. On top of this reward signal, we add Gaussian noise to the rewards. We refer to this environment as `noisy_swap` and show our results in Figure 3.

Both PG and NPCRM can infer the correct action from the observation, and their average reward remains close to optimal. As a result, they also outperform all experts in the comparison class and

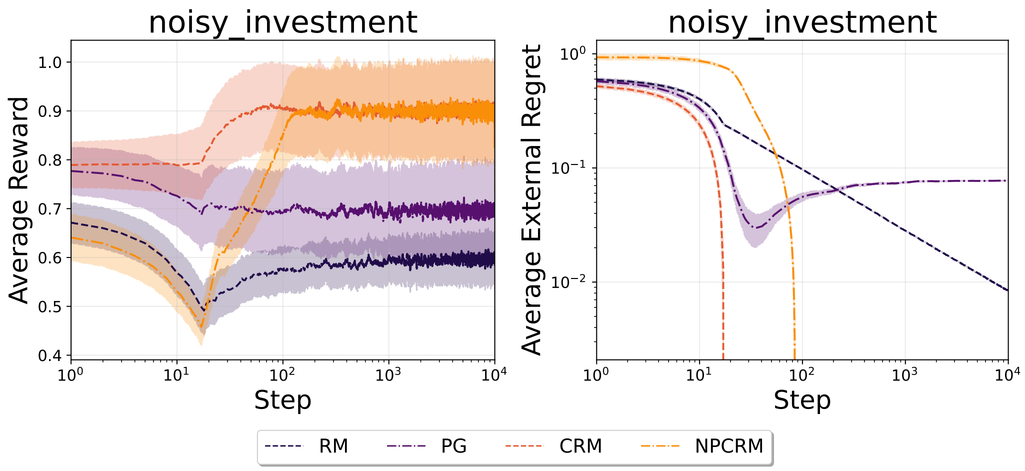

Figure 4: Comparison of the average reward and external regret of RM, PG, CRM, and NPCRM in `investment`, see Section 5.3. Both CRM and NPCRM achieve high reward and negative regret as they can condition their strategy on the actions selected in the last $L = 3$ steps.

have no regret. In this, both RM and CRM have no way to access the observation and thus fall behind. However, being able to condition on its last action, CRM can alternate between actions, bringing its expected reward up compared to RM.

5.3 History-Conditioned Feedback

An important feature of a continual learning environment is that it may respond differently depending on the strategy of the agent. To illustrate this, we use an environment with two actions and binary internal state. The state encodes the action the agent needs to use to enable the option of obtaining large reward in the following step. To get this reward, the agent needs to use the other action, which also flips the internal environment state. In addition, we perturb the rewards in each step with Gaussian noise.

We refer to this environment as `investment` and show the performance of each algorithm on this benchmark in Figure 4. While PG dominates RM in terms of the average reward, its regret remains high. In contrast, both CRM and NPCRM can achieve negative regret, outperforming all experts in the comparison class. Optimizing the expected reward in the way NPCRM does brings no benefit over CRM in this case.

6 Conclusion

In this paper, we focus on continual learning, a general framework in which a learning agent needs to constantly adapt. We argue that current reinforcement learning techniques are not well suited for this task, in particular due to the assumptions they make about the environment. In contrast, regret minimization algorithms are guaranteed to continuously adapt, without making any assumptions about the environment. We present a novel framework which extends the guarantees of the regret minimizer to recent history of the interaction. In particular, this allows the algorithm to model and adapt to the impart its own actions have on the environment.

Future Work We plan to extend our empirical evaluation, focusing on realistic scenarios where the effect of the agent's actions on the environment is significant. Offline pretraining from off-policy data can also be combined with NPCRM in an interesting way. When learning from such experiences, the agent can learn to take different action, but the simulator often cannot accurately replicate the environment feedback. However, such pretraining step can still improve empirical performance of NPCRM.

References

David Abel, André Barreto, Benjamin Van Roy, Doina Precup, Hado van Hasselt, and Satinder Singh. A definition of continual reinforcement learning. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Raman Arora, Ofer Dekel, and Ambuj Tewari. Online bandit learning against an adaptive adversary: from regret to policy regret. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, page 1747–1754, Madison, WI, USA, 2012. Omnipress. ISBN 9781450312851.

David Blackwell et al. An analog of the minimax theorem for vector payoffs. *Pacific Journal of Mathematics*, 6(1):1–8, 1956.

Michael Bowling and Esraa Elelimy. Rethinking the foundations for continual reinforcement learning, 2025. URL https://arxiv.org/abs/2504.08161.

Gabriele Farina, Christian Kroer, and Tuomas Sandholm. Faster game solving via predictive blackwell approachability: Connecting regret matching and mirror descent. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(6):5363–5371, May 2021. doi: 10.1609/aaai.v35i6.16676. URL https://ojs.aaai.org/index.php/AAAI/article/view/16676.

Saurabh Kumar, Hong Jun Jeon, Alex Lewandowski, and Benjamin Van Roy. The need for a big world simulator: A scientific challenge for continual learning, 2024. URL https://arxiv.org/abs/2408.02930.

Tor Lattimore and Csaba Szepesvári. Bandit algorithms. *preprint*, page 28, 2018.

Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V Vazirani. *Algorithmic Game Theory*. Cambridge University Press, New York, NY, USA, 2007.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Emmanouil Antonios Platanios, Abulhair Saparov, and Tom M. Mitchell. Jelly bean world: A testbed for never-ending learning. *CoRR*, abs/2002.06306, 2020. URL https://arxiv.org/abs/2002.06306.

Mark Bishop Ring. *Continual learning in reinforcement environments*. PhD thesis, USA, 1994. UMI Order No. GAX95-06083.

Richard S Sutton, Andrew G Barto, et al. *Introduction to reinforcement learning*, volume 135. MIT press Cambridge, 1998.

David Sychrovský, Michal Šustr, Elnaz Davoodi, Michael Bowling, Marc Lanctot, and Martin Schmid. Learning not to regret. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(14):15202–15210, Mar. 2024. doi: 10.1609/aaai.v38i14.29443. URL https://ojs.aaai.org/index.php/AAAI/article/view/29443.

Peter R. Wurman, Samuel Barrett, Kenta Kawamoto, James MacGlashan, Kaushik Subramanian, Thomas J. Walsh, Roberto Capobianco, Alisa Devlic, Franziska Eckert, Florian Fuchs, Leilani Gilpin, Piyush Khandelwal, Varun Kompella, HaoChih Lin, Patrick MacAlpine, Declan Oller, Takuma Seno, Craig Sherstan, Michael D. Thomure, Houmehr Aghabozorgi, Leon Barrett, Rory Douglas, Dion Whitehead, Peter Dürr, Peter Stone, Michael Spranger, and Hiroaki Kitano. Outracing champion Gran Turismo drivers with deep reinforcement learning. *Nature*, 602 (7896):223–228, February 2022. ISSN 1476-4687. doi: 10.1038/s41586-021-04357-7. URL https://doi.org/10.1038/s41586-021-04357-7.

Martin Zinkevich, Michael Johanson, Michael Bowling, and Carmelo Piccione. Regret minimization in games with incomplete information. In *Advances in neural information processing systems*, pages 1729–1736, 340 Pine Street, Sixth Floor. San Francisco. CA, 2007. Advances in Neural Information Processing Systems 20.