Refining Minimax Regret for Unsupervised Environment Design

Michael Beukman [*1] **Samuel Coward** [*1] **Michael Matthews** [1]
Mattie Fellows [1] **Minqi Jiang** [2] **Michael Dennis** [3] **Jakob Foerster** [1]

Abstract

In unsupervised environment design, reinforcement learning agents are trained on environment configurations (levels) generated by an adversary that maximises some objective. Regret is a commonly used objective that theoretically results in a minimax regret (MMR) policy with desirable robustness guarantees; in particular, the agent's maximum regret is bounded. However, once the agent reaches this regret bound on all levels, the adversary will only sample levels where regret cannot be further reduced. Although there may be possible performance improvements to be made outside of these regret-maximising levels, learning stagnates. In this work, we introduce *Bayesian level-perfect MMR* (BLP), a refinement of the minimax regret objective that overcomes this limitation. We formally show that solving for this objective results in a subset of MMR policies, and that BLP policies act consistently with a Perfect Bayesian policy over all levels. We further introduce an algorithm, *ReMiDi*, that results in a BLP policy at convergence. We empirically demonstrate that training on levels from a minimax regret adversary causes learning to prematurely stagnate, but that ReMiDi continues learning.

1. Introduction

Unsupervised environment design (UED) is an approach to automatically generate a training curriculum of environments for deep reinforcement learning (RL) agents (Dennis et al., 2020; Jiang et al., 2021a). Regret-based UED trains an adversary to select environment configurations (referred to as *levels*) that maximise the agent's *regret*, i.e., the difference in performance between an optimal policy on that level and the agent. In other words, regret measures how much better a particular agent could perform on a particular level. Empirically, training on these regret-maximising levels has been shown to improve generalisation to out-of-distribution levels in challenging domains (Dennis et al., 2020; Jiang et al., 2021a; Parker-Holder et al., 2022; Samvelyan et al., 2023; Team et al., 2023). Furthermore, at equilibrium, UED methods theoretically result in a *minimax regret policy* (Dennis et al., 2020; Jiang et al., 2021a), meaning that the policy's worst-case regret is bounded. Minimax regret (MMR) works well when the agent can simultaneously perform optimally on all levels: at convergence, the MMR policy would achieve zero regret for each level. However, this is not possible in all environments (Sukhbaatar et al., 2018). As an example, consider the T-mazes in Figure 1: these levels have different optimal behaviours but, due to partial observability, are indistinguishable to the agent. In this case, the agent cannot simultaneously perform optimally on both levels, and therefore suffers some *irreducible regret*. Since MMR-based UED methods prioritise sampling the highest regret levels, these two levels will continually be sampled for training—even though they provide nothing more for the agent to learn.

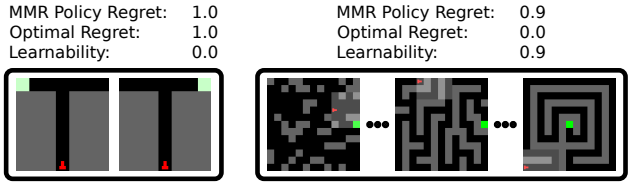

Figure 1. An illustration of the regret stagnation problem of minimax regret that our work addresses. In the T-mazes, the reward for reaching the goal is 1.0 and -1.0 for failing. The reward for the mazes is 0.9 for reaching the goal, and zero otherwise. Regret-based UED methods gravitate towards sampling high regret environments (T-mazes in this case with a regret of 1.0), even if the agent cannot improve on these levels. This is despite the existence of non-high-regret levels (the mazes, with regret upper-bounded by 0.9) on which the agent can still improve.

This phenomenon is problematic if we have a subset of levels that (a) are distinguishable from the irreducible regret levels and (b) have lower, but reducible, regret; for instance, the set of all simultaneously solvable mazes in Figure 1

[*]Equal contribution [1]University of Oxford [2]University College London [3]UC Berkeley. Correspondence to: Michael Beukman <mbeukman@robots.ox.ac.uk>, Samuel Coward <scoward@robots.ox.ac.uk>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

(which have regret of 0.9 or lower). A theoretically-sound UED method that implements MMR will converge to sampling each T-maze with 50% probability, and fail to sample any of the solvable mazes. While this does technically satisfy the MMR objective, there is no guarantee we will achieve a policy that is effective at solving normal mazes, as the agent would only rarely see these, if at all. This shows a weakness of MMR—which we call *regret stagnation*—because we know there exists a policy that obtains optimal regret on T-mazes and normal mazes.

We theoretically address this *regret stagnation problem* by proposing a refinement of the MMR objective, which we call *Bayesian level-perfect MMR* (BLP). Our objective aims to be minimax-regret over non-MMR levels, under the constraint that the policy must act according to the MMR policy in all trajectories that are consistent with MMR levels. This process is iteratively repeated, each time over a smaller subset of levels. In this way, a BLP policy retains minimax regret guarantees, and iteratively improves its worst-case regret on the remainder of the levels. We further show that any BLP policy acts consistently with a Perfect Bayesian policy on all levels. Finally, we develop an algorithm, ReMiDi, that results in a BLP policy at convergence.

Our contributions are as follows:

1. We theoretically introduce and characterise the regret stagnation problem in minimax regret UED.
2. We propose BLP, a refinement of minimax regret for UED, that retains global minimax regret, and additionally obtains minimax regret-like guarantees under trajectories that do not occur in high-regret levels.
3. We introduce a proof-of-concept algorithm, ReMiDi, that solves our new objective and returns a BLP policy.
4. We empirically demonstrate that, in settings with high irreducible regret, ReMiDi significantly outperforms standard regret-based UED.

By solving this problem, we empower the use of UED in larger and more open-ended settings, where irreducible regret is likely, as a BLP policy can be robust even outside these (potentially very rare) highest-regret levels.

2. Background

2.1. UPOMDPs

We consider an underspecified partially-observable Markov decision process (Dennis et al., 2020, UPOMDP) $\mathcal{M} = \langle A, O, \Theta, S, P_S, P_O, \mathcal{R}, \gamma \rangle$. Here A is the action space, O is the observation space, and S is the state space. Θ is the space of underspecified parameters commonly referred to as *levels*, $P_S : S \times A \times \Theta \to \Delta(S)$[1] is the level-conditional transition distribution. We denote the initial state distribution as $P_0 : \Theta \to \Delta(S)$. In the partially observable setting, the agent does not directly observe the state, but an observation variable $o \in O$ that is correlated to the underlying state. $\mathcal{R} : S \times A \to \mathbb{R}$ is the scalar reward function, we denote instances of reward at time t as $r_t = \mathcal{R}(s_t, a_t)$ and γ is the discount factor. Each set of underspecified parameters $\theta \in \Theta$ indexes a particular POMDP called a *level*. In our maze example in Figure 1, the level determines the location of the goal and obstacles but dynamics such as navigating and the reward function remain shared across all levels.

At time t the agent observes an action-observation history (or trajectory) $\tau_t = \langle o_0, a_0, \ldots, o_{t-1}, a_{t-1}, o_t \rangle$ and chooses an action according to a trajectory-conditioned policy $a_t \sim \pi(\tau_t)$. We denote the set of all trajectory-conditioned policies as $\Pi \doteq \{\pi | \pi : \mathcal{T} \to \Delta(A)\}$ where $\mathcal{T}$ denotes the set of all possible trajectories. For any level θ, the agent's goal is to maximise the expected discounted return (called *utility*), which we denote as:

$$U_\theta(\pi) \doteq \mathbb{E}_{\pi,\theta}\left[\sum_{t=0}^{T} \gamma^t r_t\right],$$

where $\mathbb{E}_{\pi,\theta}$ denotes the expected value on θ if the agent follows policy π (Sutton & Barto, 2018). We denote an optimal policy for level θ as $\pi^\star_\theta \in \arg\max_{\pi'} U_\theta(\pi')$. Finally, similar to prior work (Dennis et al., 2020), for all proofs, we restrict our attention to finite and discrete UPOMDPs.

2.2. Unsupervised Environment Design

Unsupervised Environment Design (UED) is posed as a two-player game, where an adversary Λ selects levels θ for an agent π to train on (Dennis et al., 2020). The adversary's goal is to choose levels that maximise some utility function, e.g., a constant utility for each level corresponds to domain randomisation (Tobin et al., 2017). One commonly-used objective is to maximise the agent's regret (Savage, 1951; Bell, 1982; Loomes & Sugden, 1982). Formally, the *regret* of policy π with respect to an optimal policy $\pi^\star_\theta$ on a level θ is equal to how much better $\pi^\star_\theta$ performs than π on θ, $\text{Regret}_\theta(\pi) \doteq U_\theta(\pi^\star_\theta) - U_\theta(\pi)$.

If regret is used as the payoff, at Nash equilibrium of this two-player zero-sum game, the policy satisfies minimax regret (Osborne et al., 2004; Dennis et al., 2020):[2]

$$\pi_{\text{MMR}} \in \Pi^\star_{\text{MMR}} \doteq \arg\min_{\pi}\{\max_{\theta \in \Theta}\{\text{Regret}_\theta(\pi)\}\}. \quad (1)$$

Constraining policies to the set of MMR policies $\Pi^\star_{\text{MMR}}$ has several advantages: when deploying our policy, our regret

[1] $\Delta(X)$ is the set of all probability distributions over the set X.

[2] A Nash equilibrium always exists in finite games (Nash, 1950).

can never be higher than the minimax regret bound, so the policy has a certain degree of robustness. Using minimax regret also results in an adaptive curriculum that increases in complexity over time, leading to the agent learning more efficiently (Dennis et al., 2020; Parker-Holder et al., 2022).

Further, choosing levels based on maximising regret avoids sampling levels that are too easy (as the agent already performs well on these) or impossible (where the optimal policy also does poorly). This is in contrast to standard minimax, which tends to choose impossible levels that minimise the agent's performance (Pinto et al., 2017; Dennis et al., 2020).

A drawback of using minimax regret is the assumption of having access to the optimal policy $\pi^\star_\theta$ per level, which is generally unavailable. To circumvent this issue, most methods approximate regret in practice. We discuss several commonly-used heuristics in Appendix C.3. A more serious issue with using minimax regret in isolation is that there is no formal method to choose between policies in $\Pi^\star_{\text{MMR}}$. Typically it is chance and initialisation that determines the policy an algorithm converges to. While all minimax regret policies protect against the highest-regret outcomes, these events may be rare and there may be significant differences in the utility of policies in $\Pi^\star_{\text{MMR}}$ in more commonly encountered levels. We discuss this issue further in Section 3. Finally, we refer the interested reader to Appendix C for additional background on decision theory and UED.

3. The Limits of Minimax Regret

To elucidate the issues with using minimax regret in isolation, we analyse the set of minimax regret policies $\Pi^\star_{\text{MMR}}$ introduced in Section 2.2. For any $\pi_{\text{MMR}} \in \Pi^\star_{\text{MMR}}$ and $\vartheta \in \Theta$, it trivially holds that:

$$\text{Regret}_\vartheta(\pi_{\text{MMR}}) \leq \min_{\pi'} \left\{ \max_{\theta \in \Theta} \{\text{Regret}_\theta(\pi')\} \right\}.$$

However, it is unclear whether all policies in $\Pi^\star_{\text{MMR}}$ are equally desirable across all levels. In the worst case, the minimax-regret game will converge to an agent policy that only performs as well as this bound, *even if further improvement is possible* on other (non-minimax regret) levels. In addition, the adversary's distribution will not change at such Nash equilibria, by definition. Thus, at equilibrium, the agent will not be presented with levels outside the support of Λ and as such will not have the opportunity to improve further—despite the possible existence of other MMR policies with lower regret outside the support of Λ.

This observation, and the concrete example in Figure 1 demonstrate that minimax regret does not always correspond to *learnability*: there could exist UPOMDPs with high regret on a subset of levels on which an agent is optimal (given the partial observability constraints), and low regret on levels in which it can still improve. *Our key insight is that optimising solely for minimax regret can result in the agent's learning to stop prematurely, preventing further improvement across levels outside the support of MMR levels*. We summarise this *regret stagnation problem* of minimax regret as follows:

1. The minimax regret game is indifferent to which MMR policy is achieved on convergence; and
2. Upon convergence to a policy in $\Pi^\star_{\text{MMR}}$, no improvements occur on levels outside the support of Λ.

4. Refining Minimax Regret

Having described the regret stagnation problem of minimax regret, we now introduce a new objective to address it. Concretely, we propose *Bayesian level-perfect Minimax Regret* (BLP), our refinement of the MMR decision rule applied to UED. To describe this objective succinctly, we first introduce the notion of a *realisable* trajectory, and the *refined MMR game*. The refined game fixes a policy π and set of levels Θ', and restricts the solution to act consistent with π in all trajectories possible given π and Θ', where behaviour for other trajectories can be chosen arbitrarily.

Definition 4.1. (Realisable Trajectory): For a set Θ' and policy π, $\mathcal{T}_\pi(\Theta')$ denotes the set of all trajectories that are possible by following π on any $\theta \in \Theta'$. We call a trajectory τ realisable under π and Θ' iff $\tau \in \mathcal{T}_\pi(\Theta')$.

Definition 4.2. Refined Minimax Regret game: Given a UPOMDP with level space Θ, suppose we have some policy π and some subset of levels $\Theta' \subseteq \Theta$. We introduce the **refined minimax regret game under π and Θ'**, a two-player zero-sum game between an agent and adversary where:

- the agent's strategy set is all policies of the form
$$\pi'(a|\tau) = \begin{cases} \pi(a|\tau) \text{ if } \tau \in \mathcal{T}_\pi(\Theta') \\ \bar{\pi}(a|\tau) \text{ otherwise,} \end{cases}$$
where $\bar{\pi}$ is an arbitrary policy;
- the adversary's strategy set is $\overline{\Theta} \doteq \Theta \setminus \Theta'$;
- the adversary's payoff is $\text{Regret}_\theta(\pi')$.

In other words, π' represents the set of policies that perform identically to π in any trajectory possible under π and Θ'. At Nash equilibrium of the refined game, the agent will converge to a policy that performs identically to π under all levels in Θ' (by definition), but otherwise will perform minimax regret optimally over $\overline{\Theta}$ with respect to these constraints. The adversary will converge onto a minimax regret equilibrium distribution with support only on levels in $\overline{\Theta}$.

We note that the solution to the refined game is not guaranteed to obtain absolute best worst-case regret over $\overline{\Theta}$, since

it is constrained to act according to π when observing certain trajectories. However, we can guarantee that the Nash solution must obtain optimal worst-case regret compared to all other policies that also have this constraint.

We now state and prove an important property of the equilibrium policy in the refined game: it must monotonically improve upon π in the highest-regret $\theta \in \Theta \setminus \Theta'$, while maintaining π's regret over Θ'.

Theorem 4.3. *Suppose we have a UPOMDP with level space Θ. Let π be some policy and $\Theta' \subseteq \Theta$ be some subset of levels. Let (π', Λ') denote a policy and adversary at Nash equilibrium for the refined minimax regret game under π and Θ'. Then, (a) for all $\theta \in \Theta'$, $Regret_\theta(\pi') = Regret_\theta(\pi)$; and (b) we have,*

$$\max_{\theta \in \Theta \setminus \Theta'} \{Regret_\theta(\pi')\} \leq \max_{\theta \in \Theta \setminus \Theta'} \{Regret_\theta(\pi))\}.$$

Proof. (a) By definition, π' must act according to π on all trajectories possible under π and Θ'. Therefore, the performance (and thus regret) of π' must be identical to π for all levels in Θ'. (b) In the refined MMR game, π is trivially in the agent's strategy set. So regardless of what level the adversary plays, the agent can always play π. Thus the agent's best response can never be worse than π. □

Theorem 4.4 next shows the benefits of iteratively refining an initial minimax regret policy.

Theorem 4.4. ***(Minimax Regret Refinement Theorem):***

Let $\langle \pi_1, \Lambda_1 \rangle$ be in Nash equilibrium of the minimax regret game. Let $\langle \pi_i, \Lambda_i \rangle$ with $1 < i$ denote the Nash equilibrium solution to the refined minimax regret game under π_{i-1} and $\Theta'_i = \bigcup_{j=1}^{i-1} \mathrm{Supp}(\Lambda_j)$.[3] Then, for all $i \geq 1$, (a) π_i is minimax regret and (b) we have

$$\max_{\theta \in \Theta \setminus \Theta'_i} \{Regret_\theta(\pi_i)\} \leq \max_{\theta \in \Theta \setminus \Theta'_i} \{Regret_\theta(\pi_{i-1}))\}. \quad (2)$$

Finally, (c) for $1 \leq j < i$ and $\theta \in Supp(\Lambda_j)$, $Regret_\theta(\pi_i) = Regret_\theta(\pi_j)$. In other words, iteratively refining a minimax regret policy (a) retains minimax regret guarantees; (b) monotonically improves worst-case regret on the set of levels not already sampled by any adversary; and (c) retains regret of previous refinements on previous adversaries.

This result holds due to inductively applying Theorem 4.3. At every iteration of the refined game, we must perform equally on all previously chosen levels, by definition. On the non-chosen levels, we must either maintain or improve the regret bound of the previous step, by Theorem 4.3. For space reasons, the formal proof is in Appendix A.1.

[3] $\mathrm{Supp}(\Lambda)$ denotes the support of Λ, i.e., the set of all environments that it samples with nonzero probability.

We have now defined the refined game, and shown that iteratively solving it retains minimax regret guarantees *and* monotonically improves worst-case regret in non-MMR levels. We next define the solution concept we propose to use instead of standard minimax regret, which we call *Bayesian level-perfect Minimax Regret* (BLP MMR). Intuitively, a BLP policy solves all of the refined games until all levels have been sampled.

Definition 4.5. Bayesian level-perfect Minimax Regret: Let $\langle \pi_1, \Lambda_1 \rangle$ be in Nash equilibrium of the minimax regret game. Let $\langle \pi_i, \Lambda_i \rangle$, $1 < i$ denote the solution to the refined game under π_{i-1} and $\bigcup_{j=1}^{i-1} \mathrm{Supp}(\Lambda_j)$. Policy π_j is a *Bayesian level-perfect minimax regret policy* if $\bigcup_{k=1}^{j} \mathrm{Supp}(\Lambda_k) = \Theta$.[4]

Next, Theorem 4.6 shows that BLP is similar to a Bayes perfect refinement of minimax regret, in that it acts consistently with a Perfect Bayesian policy under a minimax regret prior over levels. Its proof can be found in Appendix A.2.

Theorem 4.6. *A Bayesian level-perfect minimax regret policy π' acts consistently with a Perfect Bayesian policy on all realisable trajectories under π' and Θ.*

5. ReMiDi

Having introduced our refinement of minimax regret optimality, we now introduce our proof-of-concept method ***Re**fining **Mi**nimax Regret **Di**stributions* (ReMiDi, see Algorithm 1) that learns a *Bayesian level-perfect minimax regret policy*. Figure 2 illustrates how our approach works.

Algorithm 1 Refining **Mi**nimax Regret **Di**stributions

Require: Level space Θ
Initialise agent $\mathcal{S} = \{\}$ and adversary buffers $\mathcal{A} = \{\}$
$i \leftarrow 1$
while $\bigcup_{j=1}^{i} \mathrm{Supp}(\Lambda_j) \neq \Theta$ **do**
  Initialise adversary Λ_i and policy π_i.
  while (π_i, Λ_i) is not at Nash equilibrium **do**
    Sample level $\theta \sim \Lambda_i$
    $\pi' \leftarrow \texttt{Combine}(\mathcal{S} \cup \{\pi_i\}, \mathcal{A})$
    Sample trajectory τ using π' and θ.
    if $\tau \notin \mathcal{T}_\pi(\cup_{j=1}^{i-1} \mathrm{Supp}(\Lambda_j))$ **then**
      Update π_i and Λ_i using θ and τ
    end if
  end while
  $\mathcal{S} \leftarrow \mathcal{S} \cup \{\pi_i\}$
  $\mathcal{A} \leftarrow \mathcal{A} \cup \{\Lambda_i\}$
  $i \leftarrow i + 1$
end while
return $\texttt{Combine}(\mathcal{S}, \mathcal{A})$

[4] We refer the interested reader to Appendix B, where we discuss simpler, but ultimately flawed alternatives to BLP.

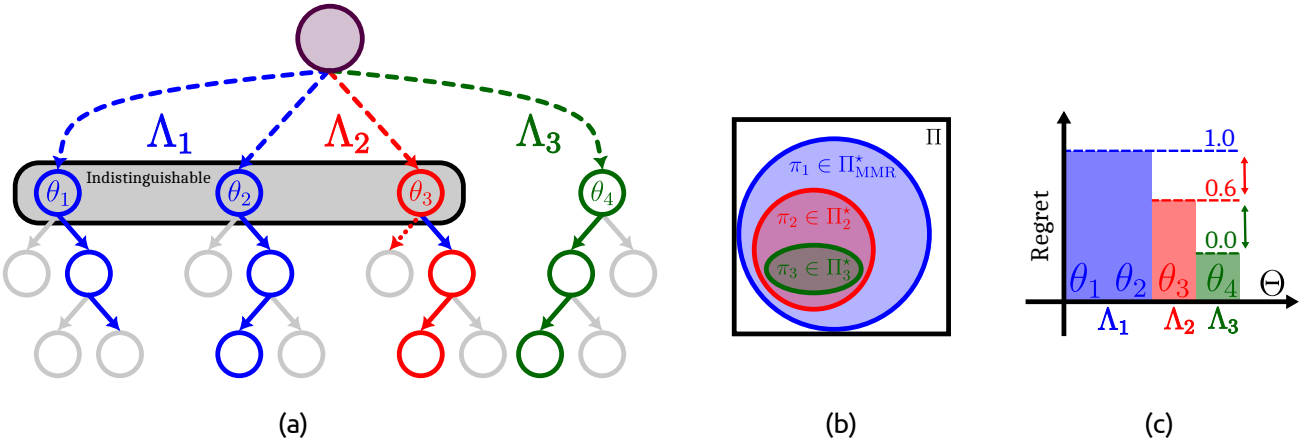

Figure 2. The BLP solution concept iteratively restricts the sets of policies by altering behaviour only in certain trajectories. **(a)** Each node corresponds to a level-trajectory pair, and the root node indicates that the adversary samples some level θ. MMR results in adversary Λ_1 and policy π_1, reaching the nodes in blue. MMR would terminate at step 1, but we instead refine our policy further. In step 2, we learn Λ_2 and π_2. Following π_1 in θ_3 leads to a trajectory that never happens under Λ_1 and π_1. In step 3, we fill in behaviour for θ_4, as these trajectories are never reached under any of the previous MMR adversaries. We terminate after all environments have been sampled by an adversary. **(b)** Iterative refinement reduces the set of policies we consider, improving upon the initial MMR policy π_1. **(c)** If we have a minimax regret adversary Λ_1, we are only guaranteed that the regret on all other levels must be at or below the regret of levels in the support of Λ_1 (indicated by the dashed blue line). Refining our policy improves the bound on all levels except those sampled by Λ_1. We iterate this process until all levels have been sampled, monotonically improving the regret bound on all non-previously-sampled levels.

In Algorithm 1, $\texttt{Combine}(\mathcal{S}, \mathcal{A})$ returns π'_n, with $n = |\mathcal{S}|$, which is constructed as follows, with $\pi'_1 = \pi_1$ being minimax regret:

$$\pi'_n(a|\tau) = \begin{cases} \pi'_{n-1}(a|\tau) & \text{if } \tau \in \mathcal{T}_{\pi'_{n-1}}(\bigcup_{i=1}^{n-1} \text{Supp}(\Lambda_i))) \\ \pi_n(a|\tau) & \text{otherwise.} \end{cases}$$

Outer Loop. Algorithm 1 is a direct implementation of the successive refined MMR games. Thus, at convergence, it will return a BLP policy. The outer loop continues until all levels have been sampled by an adversary. In practice, this is both infeasible and excessive. Thus, one may choose to only compute a fixed number of outer iterations. Lines 9-11 ensure that the current adversary only contains levels with trajectories inconsistent with any previous adversary.

Checking for convergence in the inner loop. Line 5 requires convergence to Nash equilibrium, which is not guaranteed to occur in practice; even if equilibrium is reached, determining when this has happened is also non-trivial. One way to approximately determine convergence is to measure the expected regret over the adversary, and if this has plateaued, we can terminate the optimisation and continue to the next level. We use another, simpler, technique of having a fixed number of timesteps that each adversary is trained for, and assuming that (approximate) convergence is reached after training for this number of timesteps.

Choice of adversary. Algorithm 1 is agnostic to the choice of the adversary; it could be an antagonist, level generator pair as used in PAIRED (Dennis et al., 2020), or a curated level buffer (e.g., PLR (Jiang et al., 2021b)). In the case of PAIRED, only one antagonist is required for all generators.

6. Experiments

6.1. Experimental Setup

In this section, we empirically demonstrate that the problems identified in Section 3 do occur, and that ReMiDi alleviates these issues. First, in Section 6.2, we illustrate some of the failure cases of ideal UED in a simple tabular setting. Next, in Section 6.3, we experiment in the canonical Minigrid domain. In Section 6.4, we consider a different setting where regret-based UED results in a policy that performs poorly over a large subset of levels. Finally, we evaluate on a robotics task in Section 6.5. To compare against an ideal version of UED, we use perfect regret as our score function. All plots show the mean and standard deviation over 10 seeds, whereas the tabular experiments use 50.[5]

For the latter experiments, we compare against Robust PLR (Jiang et al., 2021a, PLR$^\perp$), which is based on curating randomly-generated levels into a buffer of high-regret levels. At every step, the agent is either trained on a sample of levels from the buffer, or evaluated on a set of randomly-

[5]We publicly release our code at https://github.com/Michael-Beukman/ReMiDi.

generated levels. These randomly generated levels replace existing levels in the buffer that have lower regret scores. In robust PLR, the agent does not train on randomly generated levels. Our second baseline is a minimax return adversary, which represents a large class of robust RL and worst-case methods (Iyengar, 2005; Pinto et al., 2017). We implement this baseline as PLR$^\perp$ with a different score function: the negative of the agent's return. The buffer therefore contains levels that the agent obtains a low return on, and implicitly optimises agent performance on the worst-case levels. Our ReMiDi implementation maintains multiple buffers, and we perform standard PLR$^\perp$ on the first buffer for a certain number of iterations. We then perform PLR$^\perp$ again, but reject levels that have complete trajectory overlap with levels in a previous buffer. Instead of explicitly maintaining multiple policies, we have a single policy that we update only on the parts of trajectories that are distinguishable from levels in previous buffers, approximately maintaining performance on previous adversaries. Algorithm 1 assumes knowledge of whether a trajectory is possible given a policy and a set of levels, and we can compute this exactly in each environment. We use JaxUED (Coward et al., 2024) for our experiments, and Appendix D contains more implementation details.

6.2. Exact Settings

We consider a one-step tabular game, where we have a set of N levels $\theta_1, \ldots, \theta_N$. Each level i corresponds to a particular initial observation τ_i, such that the same observation may be shared by two different environments. Each level also has an associated reward for each action $a_j, 1 \leq j \leq M$.

We model the adversary as a N-arm bandit, implemented using tabular Actor-Critic. Each action corresponds to a different level θ_i. For ReMiDi, we have a sequence of adversaries, each selecting levels where the observations are disjoint with any previous adversary. In both cases, the agent is also a tabular Actor-Critic policy, with different action choices for each observation (equivalent to a trajectory) τ.

6.2.1. When Minimax Regret is Sufficient

In Figure 3, we first consider a case where MMR has none of the problems discussed in Section 3. Here, each θ has a unique initial observation τ, thus the level can be deduced solely from this observation. Minimax regret succeeds and converges to the globally optimal policy. Convergence occurs because a single policy can be simultaneously optimal over the set Θ, as for every observation, there is one optimal action. The MMR policy is therefore also unique.

6.2.2. When Minimax Regret Fails

We next examine a UPOMDP where a single policy can no longer be simultaneously optimal over all levels. The setup is the same as the previous experiment, except that $\tau_2 = \tau_1$, $\tau_4 = \tau_3$, etc., meaning that there is some irreducible regret. Figure 4 shows that regret-based UED rapidly obtains minimax regret, but fails to obtain optimal regret on the non-regret-maximising levels. By contrast, ReMiDi obtains optimal regret on all levels. It does this by first obtaining global minimax regret, at which point it restricts its search over levels to those that are distinguishable from minimax regret levels. Since the agent's policy is not updated on these prior states, it does not lose MMR guarantees.

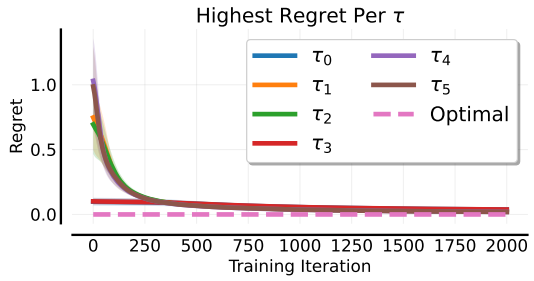

Figure 3. Plotting the regret of MMR UED throughout training for each of the 6 trajectories. Here the optimal regret is 0 for each trajectory, and minimax regret achieves this.

We analyse this further in Figure 5 by plotting the probability of each level being sampled over time. Regret-based UED rapidly converges to sampling only the highest-regret levels (θ_4 and θ_5), and shifts the probability of sampling the other levels to zero. By contrast, our multi-step process first samples these high-regret levels exclusively. Thereafter, these are removed from the adversary's options and it places support on all other levels. This shows that, while we could improve the performance of regret-based UED by adding stronger entropy regularisation (Mediratta et al., 2023), or making the adversary learn slower, the core limitation remains: when regret does not correspond to learnability, minimax regret UED will sample inefficiently.

6.3. Maze

We next consider Minigrid, a common benchmark in UED (Dennis et al., 2020; Jiang et al., 2021a; Parker-Holder et al., 2022). We discuss two distinct experimental settings. The first is an implementation of the T-maze example in Section 1. Here the adversary can sample T-mazes or normal mazes. The reward of T-mazes is $+1$ or -1 depending on whether the agent reaches the goal or not, and the standard maze reward is the same as is used in prior work (Dennis et al., 2020; Jiang et al., 2021a). The second experiment is where the adversary has the choice of blindfolding the agent; in other words, it can zero out the agent's observation. In both cases, we evaluate on a standard set of held-out mazes.

Figure 6a shows that PLR$^\perp$ with perfect regret as its score function results in poor performance on actual mazes. The

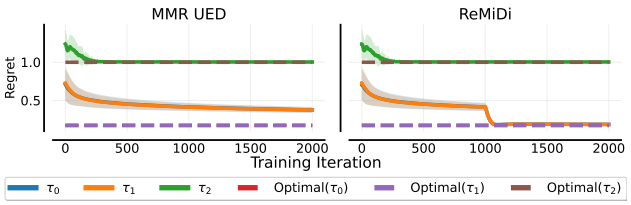

Figure 4. The greatest regret on each trajectory τ for (left) Standard Minimax Regret UED; and (right) ReMiDi. ReMiDi obtains optimal regret on all levels, whereas MMR does not.

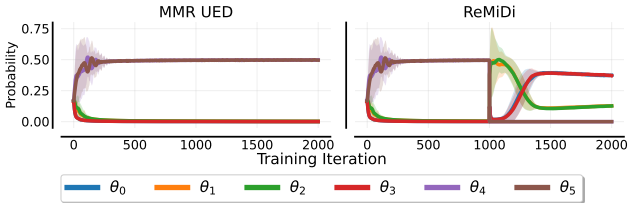

Figure 5. The Adversary's probability of sampling each environment θ for (left) Standard Minimax Regret UED; and (b) ReMiDi. MMR UED exclusively samples the irreducible-regret levels.

reason for this is that it trains almost exclusively on T-mazes, and not on actual mazes (see Figure 7a). ReMiDi, by contrast, samples T-mazes initially, and thereafter does not, as they have identical observations with previous MMR levels. This results in improved performance on actual mazes.

The blindfold experiment (Figure 6b) shows a similar result. ReMiDi performs better than PLR$^{\perp}$; again, this is because PLR$^{\perp}$ trains the agent almost exclusively on blindfold levels (Figure 7b), as these have high irreducible regret. Interestingly, PLR$^{\perp}$, despite training almost exclusively on blind levels, still manages to solve many test mazes, a phenomenon investigated by Wijmans et al. (2023).

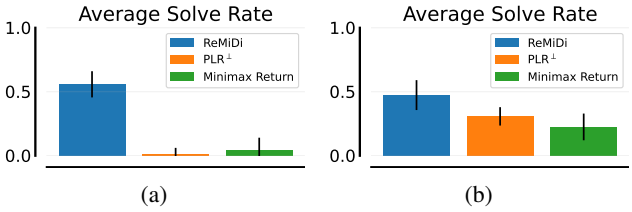

Figure 6. Solve rate on a set of held-out test mazes for the (a) T-maze and (b) blindfold experiments. ReMiDi is more effective at solving mazes than either PLR$^{\perp}$ or Minimax return.

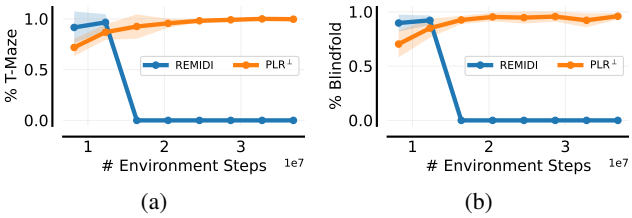

Figure 7. Fraction of replay levels that are (a) T-mazes or (b) blindfolded. PLR$^{\perp}$ converges to almost never sampling normal mazes.

6.4. Lever Game

In this environment, inspired by Hu et al. (2020), there are 64 levers to pull, one of which is correct (reward of $+1$), and pulling a wrong lever results in a reward of -1. The adversary can make the correct lever known or unknown to the agent. In the latter case, the reward is multiplied by 10

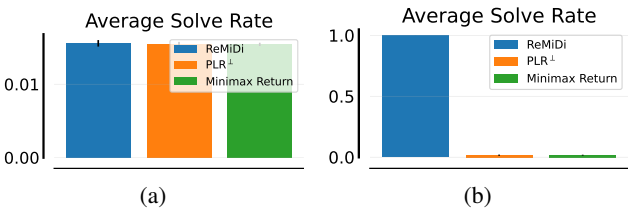

Figure 8. Solve rate in the lever game for two subsets of evaluation levels: (a) invisible; and (b) visible answers. ReMiDi can solve both types, whereas PLR$^{\perp}$ cannot, as it trained almost exclusively on invisible-answer levels.

(to simulate a harder problem having a higher reward). Our analysis in Section 3 suggests that regret-based UED should solely sample levels where the correct answer is unknown, and the best option for the agent is to guess randomly (as this induces irreducible regret). Training solely on these levels, however, would cause the agent to perform poorly when it observes the correct answer. Indeed, Figure 8 shows that on levels where the correct lever is not given—and the best policy guesses randomly—PLR$^{\perp}$ performs the same as ReMiDi, with a solve rate of around $\frac{1}{64} \approx 0.015$ (since there are 64 levers). On levels where the correct answer is given, however, ReMiDi performs perfectly (as it is possible to obtain 100% accuracy), but PLR$^{\perp}$ fails as it almost never trained on these types of levels. Importantly, this result shows that both PLR$^{\perp}$ and ReMiDi satisfy minimax regret, but PLR$^{\perp}$ results in a policy that is effectively random, whereas ReMiDi learns a much more useful policy.

6.5. Robotics

Our final experimental domain is robotics, using Brax (Todorov et al., 2012; Freeman et al., 2021) and a horizon of 100 timesteps. There are two types of levels that the adversary can choose, type A or type B. In type A levels, the task is standard: to walk forward. The levels represent disturbances to the agent's actions; the agent's action $a \in \mathbb{R}^n$ is multiplied by the level $\theta \in \mathbb{R}^n$ before being applied to the simulation. Type B levels require the agent to perform a specific, but arbitrary, action that it cannot observe. The agent receives a large positive or negative reward,

depending on whether the action is correct or not, and the episode terminates. After training, we evaluate performance when no disturbances are applied. For these experiments, we use an episode length of 100, and consider four separate morphologies, Ant, Half-Cheetah, Hopper and Walker.

We present the results in Table 1. MMR and minimax return both fall victim to irreducible regret, training primarily on type B levels, causing the agent to fail when presented with type A ones. ReMiDi, by contrast, does not suffer from this problem, and is able to learn an effective policy.

Table 1. Brax results; mean $\pm$ standard deviation over 10 seeds.

Method	Ant	Cheetah	Hopper	Walker
PLR$^{\perp}$	-50 $\pm$ 2	-19 $\pm$ 14	67 $\pm$ 19	126 $\pm$ 68
Minimax Return	-49 $\pm$ 4	-16 $\pm$ 3	86 $\pm$ 15	148 $\pm$ 67
ReMiDi	**257** $\pm$ 35	**321** $\pm$ 16	**193** $\pm$ 2	**344** $\pm$ 21

7. Related Work

UED and Adaptive Curricula. Many recent works aim to find an adaptive curriculum for an RL agent to train on, with different methods using different metrics to choose training levels. The learning potential (Oudeyer et al., 2007) of an agent on a particular level or training example is a measure of how much an agent's loss or reward will improve after training on this data point. Therefore, a level with a high learning potential is a promising one to train on, as the agent can still improve on it, but the level is not too difficult for the agent's current capability. While methods based on learning potential have shown promise empirically (Florensa et al., 2018; Portelas et al., 2019; Matiisen et al., 2020), techniques that aim for robustness adversarially sample levels; for instance, adversarial minimax trains the agent in levels that minimise the agent's performance (Pinto et al., 2017; Wang et al., 2019). However, these worst-case levels may be impossible, and therefore provide no learning benefit (Dennis et al., 2020). Minimax regret provides stronger robustness guarantees (Dennis et al., 2020; Jiang et al., 2021a; Parker-Holder et al., 2022) and also alleviates the problem of minimax choosing impossible levels. Implicit in these methods is that regret corresponds to some notion of learnability (Dennis et al., 2020). However, we have shown that regret does not always coincide with learnability, and that minimax regret may therefore cause learning to stagnate, a problem ReMiDi addresses. Some works in UED also discuss problems with minimax regret; for instance, Gur et al. (2021) describe situations in which a minimax regret adversary is incentivised to generate levels that are too difficult for the agent, and would provide no learning. Garcin et al. (2024) also show that minimax regret may not be a suitable objective if there is a particular distribution of interest we desire the agent to perform well over. They address this problem by using a small sample from this distribution to inform the level-generation process.

Decision Theory. From another perspective, regret-based UED implements a minimax regret decision rule—i.e., minimising the worst-case regret (Savage, 1951; Luce & Raiffa, 1957; Peterson, 2017; Dennis et al., 2020). This decision rule provides an ordering over policies, preferring policies with lower worst-case regret. However, any two policies that have the same worst-case regret are treated as equally good by this decision rule. Our BLP objective imposes further ordering, breaking ties according to worst-case regret over all other levels that are distinguishable from MMR levels.

Game Theory and Equilibrium Refinements. Our work is also related to the rich literature of Nash equilibrium refinements in game theory (Selten, 1975b;a; Kreps & Wilson, 1982; Fudenberg & Tirole, 1991; Osborne et al., 2004; Bonanno, 2015). In particular, our BLP solution concept is similar to the notion of Perfect Bayesian (or sequential) equilibria (Selten, 1975a; Kreps & Wilson, 1982; Fudenberg & Tirole, 1991), where a strategy must be optimal at all nodes of the game tree (given some belief), regardless of whether these nodes are reached under equilibrium or not.

SAMPLR. Jiang et al. (2022) also focus on the limitations of standard minimax regret. They identify the problem of an automatic curriculum shifting the distribution of environment parameters away from the ground truth, which is problematic in partially-observable settings. The authors propose a solution to this problem that involves training on fictitious transitions based on the ground-truth distribution of unseen parameters. The regret stagnation problem we address—where degenerate environments with irreducible regret are continually trained on—is also a side-effect of using minimax regret in partially observable settings. Importantly, one of SAMPLR's core assumptions is that the adversary and ground truth share support; however, as we have shown, this is not a given, as the adversary can place all of its probability mass on irreducible regret levels.

Worst-Case and Robust RL. There are also other formalisms outside of the UED field, such as robust MDPs and robust control in general (Iyengar, 2005; El Ghaoui & Nilim, 2005; Xu & Mannor, 2010; Wiesemann et al., 2013; Lim et al., 2013; Shen et al., 2020; Nakao et al., 2021; Yang et al., 2022; Bäuerle & Glauner, 2022; Wang et al., 2023; Bhardwaj et al., 2023; Ye et al., 2023; Goyal & Grand-Clément, 2023). This field generally has a different overall objective to UED. In particular, the Robust MDP literature often focuses on the case where the model of an MDP has some statistical or estimation errors, and the task is to learn a robust policy with respect to the real dynamics. UPOMDPs, on the other hand, are rather aimed at modelling transfer as a decision under ignorance problem (Milnor, 1951; Peterson, 2017; Dennis et al., 2020), where there is extreme

uncertainty about the environment, and the goal is to obtain an agent that obtains minimax regret over this set of environments. Furthermore, Robust RL methods tend to focus on worst-case objectives (Iyengar, 2005; El Ghaoui & Nilim, 2005) which—as our results and prior work (Dennis et al., 2020) have shown—often prioritises unsolvable levels.

8. Limitations & Future Work

One limitation of our work—indeed, of the UED field in general—is that all of our theoretical results hold only at convergence to Nash equilibrium. While a Nash equilibrium always exists in finite games (Nash, 1950), there is no guarantee that we will converge to it. Despite this, UED methods are still empirically useful (Dennis et al., 2020; Parker-Holder et al., 2022; Samvelyan et al., 2023; Team et al., 2023). Using more sophisticated algorithms for the agent and adversary—such as magnetic mirror descent (Sokota et al., 2023)—could be a way to alleviate this problem.

Furthermore, since we build on prior UED work, we use the same set of assumptions, that is, that our UPOMDPs are finite and discrete. In a continuous setting, we would require some other condition that ensures equilibria exist (for instance, closed convex continuous strategy spaces). Despite this, our algorithm can directly be applied to continuous settings, as our robotics experiments have shown.

We also note that ReMiDi requires finding the Nash equilibrium of multiple MMR games, which is theoretically more computationally challenging than just finding the equilibrium of one game. However, since existing UED methods merely run for a fixed number of environment timesteps, this problem falls away in practice. Furthermore, to obtain a BLP policy, ReMiDi potentially has to perform a large number of outer loop iterations (at most $|\Theta|$). However, in practice, we similarly approximate this by running for a small (e.g. 4) number of outer-loop iterations.

We intentionally designed our experiments to exhibit irreducible regret, but we note that in many prior benchmarks, learning stagnation due to irreducible regret levels has not been a major problem. One reason for this is if a policy can do well on all levels, then irreducible regret does not occur. Another reason is that irreducible regret is less of a problem when using imperfect measures of regret, as current methods do (Dennis et al., 2020; Jiang et al., 2021a). Nevertheless, as our regret measures improve, and as UED is applied to more complex environments with more room for irreducible regret, our analysis will become increasingly relevant. We would further like to extend our work to the more general case where the score of a level does not correspond to its *learnability*. Moreover, we believe that the connection (and difference) between regret and learnability merits further investigation. This would improve understanding of, and potentially disentangle the separate robustness (Iyengar, 2005; Pinto et al., 2017) and curriculum (Matiisen et al., 2020; Florensa et al., 2018) aspects of UED.

A final limitation is that our algorithm requires knowing the likelihood of a trajectory under a set of environments a priori, but this may be impractical in practice. Learning a belief model (Hu et al., 2021; Jiang et al., 2022) could alleviate this issue and make our algorithm more practical. Relatedly, we would like to investigate more practical ways of not degrading performance on previous adversaries, for instance, periodically training on these prior buffers.

There are several promising avenues for future work, such as attempting to refine our solution concept further, to move towards true leximinimax regret (Peterson, 2017). Ties among minimax regret policies would then be broken by considering the second-worst case environments, and so on (instead of considering only the second-worst case levels that are *distinguishable* from MMR levels). Another avenue that would merit further investigation is using constrained optimisation approaches (e.g. Ghosh et al. (2022)), where we train the policy on new levels with the constraint that its performance on past levels cannot degrade.

9. Conclusion

In this work, we show that the minimax regret decision rule has a significant limitation in partially-observable settings with high irreducible regret: the minimax regret policy may perform poorly on a large subset of levels. We propose the Bayesian level-perfect objective, a refinement of minimax regret that iteratively solves smaller games, obtaining minimax regret-like performance over more environments than is guaranteed by regret-based UED, alleviating this problem of MMR. We also develop a proof-of-concept algorithm, ReMiDi, that results in a BLP policy at convergence.

We provide theoretical justification for our approach, and show that it retains minimax regret guarantees while monotonically improving worst-case regret on non-highest-regret levels. We perform experiments in several settings with high irreducible regret and demonstrate that regret-based UED methods suffer from the regret stagnation problem we describe, causing the resulting policies to perform poorly on a large number of levels. ReMiDi, by contrast, is less susceptible to environment distributions that contain degenerate high-irreducible regret environments.

Ultimately, we believe that UED has great potential to develop robust, generalisable and effective RL agents. Our BLP solution concept is an improved objective that obtains stronger robustness guarantees than minimax regret, and avoids its pathologies in irreducible-regret environments—paving the way for UED to be applied to more open-ended and complex task spaces.

Acknowledgements

MB is supported by the Rhodes Trust. SC is funded by the EPSRC Centre for Doctoral Training in Autonomous Intelligent Machines and Systems. This work was supported by UK Research and Innovation and the European Research Council, selected by the ERC, and funded by the UKRI [grant number EP/Y028481/1].

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Bäuerle, N. and Glauner, A. Distributionally robust markov decision processes and their connection to risk measures. *Math. Oper. Res.*, 47(3):1757–1780, 2022. doi: 10.1287/MOOR.2021.1187. URL https://doi.org/10.1287/moor.2021.1187. 8

Bell, D. E. Regret in decision making under uncertainty. *Operations Research*, 30(5):961–981, 1982. ISSN 0030364X, 15265463. URL http://www.jstor.org/stable/170353. 2

Bhardwaj, M., Xie, T., Boots, B., Jiang, N., and Cheng, C. Adversarial model for offline reinforcement learning. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/0429ececfb199efc93182990169e73bb-Abstract-Conference.html. 8

Bonanno, G. Game theory: Parts i and ii-with 88 solved exercises. an open access textbook. Technical report, Working Paper, 2015. 8, 16

Coward, S., Beukman, M., and Foerster, J. Jaxued: A simple and useable ued library in jax. *arXiv preprint*, 2024. 6

Dennis, M., Jaques, N., Vinitsky, E., Bayen, A. M., Russell, S., Critch, A., and Levine, S. Emergent complexity and zero-shot transfer via unsupervised environment design. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/985e9a46e10005356bbaf194249f6856-Abstract.html. 1, 2, 3, 5, 6, 8, 9, 18, 19

El Ghaoui, L. and Nilim, A. Robust solutions to markov decision problems with uncertain transition matrices. *Operations Research*, 53(5):780–798, 2005. 8, 9

Florensa, C., Held, D., Geng, X., and Abbeel, P. Automatic goal generation for reinforcement learning agents. In Dy, J. G. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1514–1523. PMLR, 2018. URL http://proceedings.mlr.press/v80/florensa18a.html. 8, 9

Freeman, C. D., Frey, E., Raichuk, A., Girgin, S., Mordatch, I., and Bachem, O. Brax - a differentiable physics engine for large scale rigid body simulation, 2021. URL http://github.com/google/brax. 7

Fudenberg, D. and Tirole, J. Perfect bayesian equilibrium and sequential equilibrium. *journal of Economic Theory*, 53(2):236–260, 1991. 8

Garcin, S., Doran, J., Guo, S., Lucas, C. G., and Albrecht, S. V. Dred: Zero-shot transfer in reinforcement learning via data-regularised environment design. 2024. URL https://doi.org/10.48550/arXiv.2402.03479. 8

Ghosh, A., Zhou, X., and Shroff, N. Provably efficient model-free constrained rl with linear function approximation. *Advances in Neural Information Processing Systems*, 35:13303–13315, 2022. 9

Goyal, V. and Grand-Clément, J. Robust markov decision processes: Beyond rectangularity. *Math. Oper. Res.*, 48 (1):203–226, 2023. doi: 10.1287/MOOR.2022.1259. URL https://doi.org/10.1287/moor.2022.1259. 8

Gur, I., Jaques, N., Miao, Y., Choi, J., Tiwari, M., Lee, H., and Faust, A. Environment generation for zero-shot compositional reinforcement learning. In Ranzato, M., Beygelzimer, A., Dauphin, Y. N., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, pp. 4157–4169, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/218344619d8fb95d504ccfa11804073f-Abstract.html. 8

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735. 19

Hu, H., Lerer, A., Peysakhovich, A., and Foerster, J. N. "other-play" for zero-shot coordination. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 4399–4410. PMLR, 2020. URL http://proceedings.mlr.press/v119/hu20a.html. 7

Hu, H., Lerer, A., Cui, B., Pineda, L., Brown, N., and Foerster, J. N. Off-belief learning. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 4369–4379. PMLR, 2021. URL http://proceedings.mlr.press/v139/hu21c.html. 9

Iyengar, G. N. Robust dynamic programming. *Mathematics of Operations Research*, 30(2):257–280, 2005. 6, 8, 9

Jiang, M., Dennis, M., Parker-Holder, J., Foerster, J. N., Grefenstette, E., and Rocktäschel, T. Replay-guided adversarial environment design. In Ranzato, M., Beygelzimer, A., Dauphin, Y. N., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, pp. 1884–1897, 2021a. URL https://proceedings.neurips.cc/paper/2021/hash/0e915db6326b6fb6a3c56546980a8c93-Abstract.html. 1, 5, 6, 8, 9, 18, 19

Jiang, M., Grefenstette, E., and Rocktäschel, T. Prioritized level replay. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 4940–4950. PMLR, 2021b. URL http://proceedings.mlr.press/v139/jiang21b.html. 5

Jiang, M., Dennis, M., Parker-Holder, J., Lupu, A., Küttler, H., Grefenstette, E., Rocktäschel, T., and Foerster, J. Grounding aleatoric uncertainty for unsupervised environment design. *Advances in Neural Information Processing Systems*, 35:32868–32881, 2022. 8, 9

Jiang, M., Dennis, M., Grefenstette, E., and Rocktäschel, T. minimax: Efficient baselines for autocurricula in jax. In *Agent Learning in Open-Endedness Workshop at NeurIPS*, 2023. 19

Kreps, D. M. and Wilson, R. Sequential equilibria. *Econometrica: Journal of the Econometric Society*, pp. 863–894, 1982. 8

Lim, S. H., Xu, H., and Mannor, S. Reinforcement learning in robust markov decision processes. *Advances in Neural Information Processing Systems*, 26, 2013. 8

Loomes, G. and Sugden, R. Regret theory: An alternative theory of rational choice under uncertainty. *The Economic Journal*, 92(368):805–824, 1982. ISSN 00130133, 14680297. URL http://www.jstor.org/stable/2232669. 2

Lu, C., Kuba, J., Letcher, A., Metz, L., Schroeder de Witt, C., and Foerster, J. Discovered policy optimisation. *Advances in Neural Information Processing Systems*, 35: 16455–16468, 2022. 20

Luce, R. D. and Raiffa, H. *Games and Decisions: Introduction and Critical Survey*. Wiley, New York, 1957. 8

Matiisen, T., Oliver, A., Cohen, T., and Schulman, J. Teacher-student curriculum learning. volume 31, pp. 3732–3740, 2020. doi: 10.1109/TNNLS.2019.2934906. URL https://doi.org/10.1109/TNNLS.2019.2934906. 8, 9

Mediratta, I., Jiang, M., Parker-Holder, J., Dennis, M., Vinitsky, E., and Rocktäschel, T. Stabilizing unsupervised environment design with a learned adversary. In *Conference on Lifelong Learning Agents*, pp. 270–291, 2023. 6

Milnor, J. Games against nature. Technical report, RAND PROJECT AIR FORCE SANTA MONICA CA, 1951. 8

Nakao, H., Jiang, R., and Shen, S. Distributionally robust partially observable markov decision process with moment-based ambiguity. *SIAM J. Optim.*, 31(1):461–488, 2021. doi: 10.1137/19M1268410. URL https://doi.org/10.1137/19M1268410. 8

Nash, J. F. Equilibrium points in n-person games. *Proceedings of the national academy of sciences*, 36(1):48–49, 1950. 2, 9, 17

Osborne, M. J. and Rubinstein, A. *A course in game theory*. MIT press, 1994. 15, 16

Osborne, M. J. et al. *An introduction to game theory*, volume 3. Oxford university press New York, 2004. 2, 8, 17

Oudeyer, P.-Y., Kaplan, F., and Hafner, V. V. Intrinsic motivation systems for autonomous mental development. *IEEE transactions on evolutionary computation*, 11(2): 265–286, 2007. 8

Parker-Holder, J., Jiang, M., Dennis, M., Samvelyan, M., Foerster, J., Grefenstette, E., and Rocktäschel, T. Evolving curricula with regret-based environment design. In *Proceedings of the International Conference on Machine Learning*, pp. 17473–17498. PMLR, 2022. URL https://proceedings.mlr.press/v162/parker-holder22a.html. 1, 3, 6, 8, 9, 18, 19

Peterson, M. *An introduction to decision theory*. Cambridge University Press, 2017. 8, 9, 17

Pinto, L., Davidson, J., Sukthankar, R., and Gupta, A. Robust adversarial reinforcement learning. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2817–2826. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/pinto17a.html. 3, 6, 8, 9

Portelas, R., Colas, C., Hofmann, K., and Oudeyer, P. Teacher algorithms for curriculum learning of deep RL in continuously parameterized environments. In Kaelbling, L. P., Kragic, D., and Sugiura, K. (eds.), *3rd Annual Conference on Robot Learning, CoRL 2019, Osaka, Japan, October 30 - November 1, 2019, Proceedings*, volume 100 of *Proceedings of Machine Learning Research*, pp. 835–853. PMLR, 2019. URL http://proceedings.mlr.press/v100/portelas20a.html. 8

Samvelyan, M., Khan, A., Dennis, M., Jiang, M., Parker-Holder, J., Foerster, J. N., Raileanu, R., and Rocktäschel, T. MAESTRO: open-ended environment design for multi-agent reinforcement learning. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/pdf?id=sKWlRDzPfd7. 1, 9, 18

Savage, L. J. The theory of statistical decision. *Journal of the American Statistical association*, 46(253):55–67, 1951. 2, 8

Selten, R. Reexamination of the perfectness concept for equilibrium points in extensive games. *International Journal of Game Theory*, 4, 1975a. 8

Selten, R. Spieltheoretische behandlung eines oligopolmodells mit nachfrageträgheit. *Zeitschrift für die gesamte Staatswissenschaft/Journal of Institutional and Theoretical Economics*, (H. 2):374–374, 1975b. 8

Shen, Q., Li, Y., Jiang, H., Wang, Z., and Zhao, T. Deep reinforcement learning with robust and smooth policy. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8707–8718. PMLR, 2020. URL http://proceedings.mlr.press/v119/shen20b.html. 8

Sokota, S., D'Orazio, R., Kolter, J. Z., Loizou, N., Lanctot, M., Mitliagkas, I., Brown, N., and Kroer, C. A unified approach to reinforcement learning, quantal response equilibria, and two-player zero-sum games. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/pdf?id=DpE5UYUQzZH. 9

Sukhbaatar, S., Lin, Z., Kostrikov, I., Synnaeve, G., Szlam, A., and Fergus, R. Intrinsic motivation and automatic curricula via asymmetric self-play. In *6th International Conference on Learning Representations*. OpenReview.net, 2018. URL https://openreview.net/forum?id=SkT5Yg-RZ. 1

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018. 2, 19

Team, A. A., Bauer, J., Baumli, K., Baveja, S., Behbahani, F. M. P., Bhoopchand, A., Bradley-Schmieg, N., Chang, M., Clay, N., Collister, A., Dasagi, V., Gonzalez, L., Gregor, K., Hughes, E., Kashem, S., Loks-Thompson, M., Openshaw, H., Parker-Holder, J., Pathak, S., Nieves, N. P., Rakicevic, N., Rocktäschel, T., Schroecker, Y., Sygnowski, J., Tuyls, K., York, S., Zacherl, A., and Zhang, L. Human-timescale adaptation in an open-ended task space. *CoRR*, abs/2301.07608, 2023. doi: 10.48550/arXiv.2301.07608. URL https://doi.org/10.48550/arXiv.2301.07608. 1, 9

Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., and Abbeel, P. Domain randomization for transferring deep neural networks from simulation to the real world. In *International Conference on Intelligent Robots and Systems*, pp. 23–30. IEEE, 2017. doi: 10.1109/IROS.2017.8202133. URL https://doi.org/10.1109/IROS.2017.8202133. 2

Todorov, E., Erez, T., and Tassa, Y. Mujoco: A physics engine for model-based control. In *International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012. doi: 10.1109/IROS.2012.6386109. URL https://doi.org/10.1109/IROS.2012.6386109. 7

Wang, R., Lehman, J., Clune, J., and Stanley, K. O. Paired Open-Ended Trailblazer (POET): Endlessly generating increasingly complex and diverse learning environments and their solutions. *CoRR*, abs/1901.01753, 2019. URL http://arxiv.org/abs/1901.01753. 8

Wang, S., Si, N., Blanchet, J. H., and Zhou, Z. On the foundation of distributionally robust reinforcement learning. *CoRR*, abs/2311.09018, 2023. doi: 10.48550/ARXIV.2311.09018. URL https://doi.org/10.48550/arXiv.2311.09018. 8

Wiesemann, W., Kuhn, D., and Rustem, B. Robust markov decision processes. *Mathematics of Operations Research*, 38(1):153–183, 2013. 8

Wijmans, E., Savva, M., Essa, I., Lee, S., Morcos, A. S., and Batra, D. Emergence of maps in the memories of blind navigation agents. *AI Matters*, 9(2):8–14, 2023. doi: 10.1145/3609468.3609471. URL https://doi.org/10.1145/3609468.3609471. 7

Xu, H. and Mannor, S. Distributionally robust markov decision processes. In Lafferty, J. D., Williams, C. K. I., Shawe-Taylor, J., Zemel, R. S., and Culotta, A. (eds.), *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada*, pp. 2505–2513. Curran Associates, Inc., 2010. URL https://proceedings.neurips.cc/paper/2010/hash/19f3cd308f1455b3fa09a282e0d496f4-Abstract.html. 8

Yang, R., Bai, C., Ma, X., Wang, Z., Zhang, C., and Han, L. RORL: robust offline reinforcement learning via conservative smoothing. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/96bbdd0ed2a9e7cd2fb7caf2fae15f3d-Abstract-Conference.html. 8

Ye, C., Yang, R., Gu, Q., and Zhang, T. Corruption-robust offline reinforcement learning with general function approximation. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/71b52a5b3fe2e9303433a174b60e160d-Abstract-Conference.html. 8

A. Proofs of Theoretical Results

A.1. Proof of the Minimax Regret Refinement Theorem (Theorem 4.4)

Theorem 4.4. Let $\langle \pi_1, \Lambda_1 \rangle$ be in Nash equilibrium of the minimax regret game. Let $\langle \pi_i, \Lambda_i \rangle$ with $1 < i$ denote the Nash equilibrium solution to the refined minimax regret game under π_{i-1} and $\Theta'_i = \bigcup_{j=1}^{i-1} \text{Supp}(\Lambda_j)$. Then, for all $i \geq 1$, (a) π_i is minimax regret and (b) we have

$$\max_{\theta \in \Theta \setminus \Theta'_i} \{\text{Regret}_\theta(\pi_i)\} \leq \max_{\theta \in \Theta \setminus \Theta'_i} \{\text{Regret}_\theta(\pi_{i-1}))\}. \tag{3}$$

Finally, (c) for $1 \leq j < i$ and $\theta \in \text{Supp}(\Lambda_j)$, $\text{Regret}_\theta(\pi_i) = \text{Regret}_\theta(\pi_j)$.

Proof. We first prove (a) inductively. For each k, we assume π_k is minimax regret and must show that π_{k+1} is also minimax regret.

Base Case: π_1 is trivially minimax regret by definition.

Inductive Case: Suppose π_k is minimax regret. By Theorem 4.3, π_{k+1} must have better worst-case regret than π_k for the set of levels outside the support of all previous adversaries. Given this, and since π_{k+1} is also constrained to behave identically to π_k under the support of these previous adversaries, it cannot perform worse over any previously-sampled level. Thus, π_{k+1} cannot decrease worst-case performance compared to π_k on the full level set Θ. Hence π_{k+1} is minimax regret optimal.

We prove (b) inductively again.

Base Case: The case of $i = 2$ follows directly from Theorem 4.3.

Inductive Case: Again, we can invoke Theorem 4.3 to show that π_i must monotonically improve worst-case regret of π_{i-1} over $\Theta \setminus \Theta'_i$.

(c) This follows directly from the definition of the refined game, as π_i cannot change behaviour for any level sampled by any previous adversary.

□

A.2. Proof of Bayesian Perfect Policy (Theorem 4.6)

Theorem 4.6. A Bayesian level-perfect minimax regret policy π' acts consistently with a Perfect Bayesian policy on all realisable trajectories under π' and Θ.

Proof. We denote Θ_τ as the information set of τ, consisting of all levels θ that could have generated τ.

Let π' be a Bayesian level-perfect MMR policy (as per Definition 4.5). To show that π' is a Perfect Bayesian policy, we must show (a) that at every trajectory, corresponding to an information set, π' acts optimally with respect to some distribution $\mu(\tau)$ over Θ_τ. Furthermore, (b) this belief must be updated using Bayes' rule wherever possible, i.e., we do not update the posterior on a 0 probability event.

(i): Equlibrium Paths We first consider on-equilibrium paths, i.e., those trajectories reachable by π_1 and Λ_1.

Define

$$W = \min_{\pi \in \Pi} \left\{ \max_{\theta \in \Theta} \{\text{Regret}_\theta(\pi)\} \right\}$$

to be the optimal worst-case regret.

Since π' and Λ_1 are in equilibrium of the minimax regret game, we have that, for each $\theta \in \text{Supp}(P_{\Lambda_1}(\theta))$, $\text{Regret}_\theta(\pi') = W$. Therefore, for any probability distribution $P'(\theta)$ such that $\text{Supp}(P'(\theta)) \subseteq \text{Supp}(P_{\Lambda_1}(\theta))$, we have that

$$\mathbb{E}_{\theta \sim P'(\theta)}[\text{Regret}_\theta(\pi')] = W.$$

Now, let τ be any trajectory along the equilibrium path. Then we know that $P(\theta|\tau)$ is a probability distribution that has support that is a subset of the support of Λ_1.

Assume, for the sake of contradiction, that there exists a policy $\bar{\pi}$ that obtains a lower expected regret over $P(\theta|\tau)$. If this is true, it must mean that $\bar{\pi}$ improves upon π' in some environments in the support of $P(\theta|\tau)$. Since $\text{Supp}(P(\theta|\tau)) \subseteq \text{Supp}(P_{\Lambda_1}(\theta))$, $\bar{\pi}$ must improve in performance compared to π' over the entire distribution $P_{\Lambda_1}(\theta)$. This provides the contradiction, since π' and Λ_1 are at Nash equilibrium—therefore, there does not exist a policy $\bar{\pi}$ that obtains a better utility while the adversary is fixed.

(ii): Off-Equilibrium Paths Let τ be a realisable trajectory that is not reached under Λ_1 and π_1. Let i be the smallest integer such that τ is reached under Λ_i and π'_i. Then, using Λ_i as the prior and performing Bayesian updating using π'_i will result in a well-defined probability distribution over Θ_τ. If there exists a policy that performs better starting at τ than π'_i over this distribution, then—similarly to case (i)—that contradicts π'_i being a Nash equilibrium of the i-th refined game. Therefore, by contradiction, π' satisfies (a) on off-equilibrium paths.

(iii): Non-Realisable Paths Consider a trajectory that does not occur under any Λ_i, π'_i. This must mean that the trajectory can happen in some environment θ, but π'_i never acts such that this trajectory occurs. Let τ denote one of these non-realisable trajectories. If π' is strictly dominated over the information set Θ_τ, let $\bar{\pi}$ be any policy that weakly dominates π' and is itself not strictly dominated. This must exist, as some strategies must survive iterated deletion of strictly-dominated strategies (Osborne & Rubinstein, 1994, Proposition 61.2).

We can modify π''s actions in all descendants τ' of τ to act according to $\bar{\pi}$. We call this modified policy π_{bayes}. This policy acts according to π' in all realisable trajectories, and therefore obtains the same regret in all environments.

If π' is not strictly dominated, we can let π_{bayes} act according to π' on all descendants of τ.

Then, in either case, we have that π_{bayes} is not strictly dominated over Θ_τ, and therefore must be optimal over some distribution $\mu(\Theta_\tau)$ (See Lemma 60.1 of Osborne & Rubinstein (1994)).

By (i) and (ii), we have that π_{bayes} satisfies optimality over some belief (which is updated using Bayes' rule) on all realisable paths. By construction, we must have that π_{bayes} also satisfies this on all non-realisable paths. Therefore, π_{bayes} is a Perfect Bayesian policy. However, since π' acts consistently with π_{bayes} over *all* realisable trajectories (by construction), π' acts consistently with some Perfect Bayesian policy on these realisable trajectories. This proves the result.

□

B. On the Temporal Inconsistency of Minimax Regret

In this section we expand upon some simpler, but ultimately flawed alternatives to our BLP formulation.

Global Regret One possible approach is to consider each trajectory τ, and the associated set of levels consistent with it Θ_τ. We could aim to find the policy that satisfies MMR over all of these subsets. In other words, a policy in the intersection of $\Pi^\star_{\text{MMR}}$ for all Θ_τ:

$$\pi \in \Pi_{\mathcal{T}} \doteq \bigcap_{\tau \in \mathcal{T}} \arg\min_{\pi \in \Pi} \left\{ \max_{\theta \in \Theta_\tau} \{\text{Regret}_\theta(\pi)\} \right\}. \tag{4}$$

However, a simple counterexample suffices to show that this set is not guaranteed to be non-empty in general. Consider a two-step MDP that has a single initial observation but multiple levels θ_1 and θ_2, each having a unique second observation τ_1 or τ_2, respectively. Finally, suppose that θ_1 and θ_2 have different optimal actions in the initial state. Then, $\Theta_{\tau_1} = \{\theta_1\}$ and $\Theta_{\tau_2} = \{\theta_2\}$, and the minimax regret policy over Θ_{τ_1} and Θ_{τ_2} must perform different actions given the shared initial observation. Therefore, the set of minimax regret policies over Θ_{τ_1} and Θ_{τ_2} must be disjoint. This means that the policy in Equation (4) is not always guaranteed to exist.

Local Regret Next, if we use a local form of regret, where regret at a trajectory τ is defined as the performance difference between the optimal agent and the current agent, given that both are initialised to τ.

We propose the following environment as a counterexample to this: A two-step MDP with two levels θ_A and θ_B. These share an initial state s_0, and there are two possible next states, s_A and s_B. There is only one allowed action in s_0, which

stochastically transitions to s_A (99% probability if the level is θ_A, 1% probability if the level is θ_B) or s_B (99% probability in θ_B and 1% in θ_A).

Once the agent is in either s_A or s_B, it must bet \$100 on whether it is in θ_A (action a_1) or θ_B (action a_2). If it wins the bet it gains \$100, otherwise it loses the \$100. Any policy can therefore be described by two actions, i, j, corresponding to the policy's actions in s_A and s_B respectively. Table 2 contains the utility for each policy, and Table 3 shows the regret for each policy. We note here we consider only deterministic policies, and stochastic policies can be obtained by using a mixed strategy.

Now let us consider the minimax regret policy for the entire MDP. It is clear that the policy must perform action a_1 in θ_A and a_2 in θ_B, with an overall regret of 2. Therefore, the policy a_1a_2 is MMR.

Table 2. The decision matrix for this MDP.

Utility	a_1a_1	a_1a_2	a_2a_1	a_2a_2
θ_A	$+100$	$+98$	-98	-100
θ_B	-100	$+98$	-98	$+100$

Table 3. Regret Matrix corresponding to Table 2.

Regret	a_1a_1	a_1a_2	a_2a_1	a_2a_2
θ_A	0	2	198	200
θ_B	200	2	198	0

Now let us consider the game starting from s_A, with utility and regret shown in Tables 4 and 5. Here we note that the policy can be described solely by one action, that which it takes in s_A.

In this case, the game is similar to matching pennies (Osborne & Rubinstein, 1994; Bonanno, 2015) and we have that the minimax regret policy must perform action a_1 and a_2, each with 50% probability. By symmetry, the same holds for the game starting at s_B. The resulting policy that satisfies minimax regret starting from both s_A and s_B is to act according to a_1a_1 50% of the time, and according to a_2a_2 50% of the time. If we apply this policy in the original game, we can see that it obtains a worst-case regret of 100—much higher than the global MMR policy's worst-case regret of 2. Therefore, *being minimax regret on both s_A and s_B causes a policy to not be minimax regret over the entire game.*

Table 4. The decision matrix for the MDP starting at s_A.

Utility	a_1	a_2
θ_A	$+100$	-100
θ_B	-100	$+100$

Table 5. The regret for the MDP starting at s_A.

Utility	a_1	a_2
θ_A	0	200
θ_B	200	0

This counterexample shows that if we aim to perform minimax regret using a local regret measure at a particular trajectory τ, then we can invalidate global minimax regret guarantees.

Some Intuition An intuitive explanation for why this happens is that the global policy can hedge its risk by committing 100% to θ_A in state s_A and 100% to θ_B in s_B; in this way, it has minimax regret globally because the environment can transition to either s_A or s_B. The local policy, by contrast, assumes it is *already* in s_A or s_B, and therefore has to hedge its risk in each case.

Summary The side-effect of these counterexamples is that we cannot aim to be minimax regret given any trajectory using a local regret measure, because if we change behaviour on any future trajectory, it may cause the agent to lose minimax regret guarantees. This is why our BLP formulation *only* alters behaviour of trajectories that never occur under the previous adversaries and policies. This allows us to circumvent this temporal inconsistency problem.

C. Additional Background

In this section, we provide additional background. Appendix C.1 discusses decisions under ignorance—an important foundational body of work for UED. Appendix C.2 discusses practical UED methods and Appendix C.3 describes common approximations for regret.

C.1. Decisions under ignorance

Decisions under ignorance (Peterson, 2017) is the setting where a particular agent has to make a decision, i.e., performing an action, which will have a particular outcome, depending on the true state of the world. The possible world states, actions and outcomes are known, but the probability of the world being in any particular state is unknown.

Generally, this can be formalised by letting S be all possible world states, A being all possible actions that the agent can take, and $U(s, a)$ being the *utility* of the agent performing action a in state s.

This in turn can be written as a matrix, where each row corresponds to a particular state s and each column corresponds to a particular action a. The entry in the matrix is the utility of performing action a in state s. There are several different decision rules of which we briefly discuss three, *minimax*, *leximin* and *minimax regret*.

Concretely, each rule r induces an ordering over actions $\geq_r$. The **minimax** decision rule chooses actions based on maximising the worst case utility, i.e., $a_i \geq_{\text{minimax}} a_j$ if and only if $\min_s U(s, a_i) \geq \min_s U(s, a_j)$, i.e., that the worst case performance of a_i is better than that of a_j.

However, if multiple actions have the same worst case utility, then the minimax decision rule is indifferent between these actions. **leximin** breaks ties by choosing, among those actions that have optimal worst-case utility, the one with the best second worst-case utility, and so on. Concretely, if we let $\min^n$ as the nth worst possible outcome. The leximin ordering is as follows:

$$a_i \geq_{\text{leximin}} a_j \text{ iff } \exists n > 0, s.t., min^n(a_i) > min^n(a_j) \text{ and } min^m(a_i) = min^m(a_j) \text{ for all } m < n.$$

Minimax regret. Finally, the **minimax regret** decision rule chooses actions based on minimising the maximum regret, i.e., $a_i \geq_{\text{minimax regret}} a_j$ if and only if (letting $U^*(s) = \max_a\{U(s, a)\}$):

$$\max_s\{U^*(s) - U(s, a_i)\} \leq \max_s\{U^*(s) - U(s, a_j)\}.$$

For instance, if we consider Table 6, a_1 and a_2 would both be minimax, since their worst case is the best. a_2 would be leximin, since its second worst case is better than a_1's second worst case. Both a_3 and a_4 would be minimax regret, since their worst-case regret is the lowest.

Table 6. An example decision matrix

	a_1	a_2	a_3	a_4
s_1	-100	-100	-101	-101
s_1	-90	-2	-1	1
s_2	-90	0	-1	-2

Table 7. The regret matrix of Table 6

	a_1	a_2	a_3	a_4
s_1	0	0	1	1
s_1	91	3	2	0
s_2	90	0	1	2

C.1.1. Two-Player zero-sum games

Here we briefly connect the above setting to two-player zero-sum games. While we have explained the above in the context of a fixed, but unknown world state s, the world state could also be chosen by another agent, with its payoff being the negative of the first agent's payoff.

If these players are in a Nash equilibrium, it means that neither player can improve their payoff by unilaterally changing their strategy. If each strategy set is finite, this game has a Nash equilibrium (Nash, 1950), and the minimax theorem applies directly (Osborne et al., 2004), implying that the payoff is $\min_a\{\max_s\{\text{Payoff(s, a)}\}\}$.

C.2. Practical UED Methods

In practice, there are two primary classes of UED methods, generation- and curation-based. **Generation-based methods**, such as PAIRED (Dennis et al., 2020), train a level-editor adversary using reinforcement learning, and it learns how to generate levels that maximise the agent's regret. As an example, in a maze setting, the adversary performs N sequential actions, and each action places either the agent, the goal, or a wall. It is updated using RL with a reward of $-\text{Regret}_\theta(\pi)$.

Our method is agnostic to the exact UED algorithm used, but our concrete implementation builds on PLR$^\perp$ (Jiang et al., 2021a).

The other broad class of methods are **curation-based** (Jiang et al., 2021a; Parker-Holder et al., 2022; Samvelyan et al., 2023). These methods generate levels using domain randomisation–effectively generating random levels by e.g. sampling maze tiles i.i.d. Then, the agent is evaluated on these levels and its approximate regret is measured. The levels that induce a high regret are then added to a buffer of levels, which the agent is later trained on. In this way, the UED algorithm only has to curate high-regret levels instead of generating them directly, which is quite challenging.

C.3. Regret Approximations in UED

While regret has desirable theoretical properties, it is intractable to compute in general. Therefore, UED methods have to instead approximate regret. PAIRED (Dennis et al., 2020) does this by concurrently training two agents, and using the difference in performance between these as a proxy for regret. PLR$^\perp$ (Jiang et al., 2021a) uses two different scoring functions, Positive Value Loss (PVL) and Maximum Monte Carlo (MaxMC). PVL approximates regret as the average of the value loss (the reward obtained minus the predicted reward) over all transitions that have positive value loss in an episode. This prioritises levels on which the agent can still improve. MaxMC approximates the optimal return on any level as the best performance the agent has ever achieved on this level.

D. Experimental Details

D.1. ReMiDi Implementation Details

We implement ReMiDi on top of PLR$^\perp$ as our base UED algorithm. The procedure is as follows. We first run PLR$^\perp$ normally for $N = 1000$ iterations. We then initialise a new adversary. When new levels are sampled, we perform the agent's action on every one of the levels in the previous buffer(s). Using the observations we obtain, we determine when the current trajectory is distinguishable from a trajectory from a prior adversary. If the trajectory has complete overlap with any level in the previous buffer, we do not add it to the new one. If it has partial or no overlap, we add it to the new buffer as normal (i.e., if the level has a higher score than any existing one). We continue to initialise a new adversary after every N steps. For computational reasons, we only have a fixed number of adversaries, and remain at the last one after we have iterated through all of the previous ones.

The above works since each of our environments is deterministic. In stochastic settings, one could either add the random seed to the level θ, in which case the environment becomes deterministic given θ. Alternatively, we can learn a belief model $P(\tau|\Lambda_i)$ concurrently with learning Λ_i. Then, if $P(\tau|\Lambda_i)$ is lower than some threshold for all previous adversaries, we can consider it as distinguishable.

Finally, we approximate Algorithm 1 in that we do not train separate policies for each adversary. However, we update π only on parts of trajectories that are distinguishable from previous adversaries. While we could periodically train the agent on levels from previous buffers to ensure it does not forget, we opt for the simpler option.

D.2. Environment Description

D.2.1. Tabular Setting

The tabular game consists of 6 environments $\theta_1, \ldots, \theta_6$. In the first experiment, each of these consists of a unique trajectory τ_i. In each state, the agent can choose between two actions, A and B. The environment terminates after one step, with rewards given by Table 8.

Table 8. The rewards of the first tabular setting in Section 6.2.

Environment	Trajectory	Reward A	Reward B
θ_1	τ_1	-0.1	$+0.1$
θ_2	τ_2	$+0.7$	-0.7
θ_3	τ_3	-0.7	$+0.7$
θ_4	τ_4	$+0.1$	-0.1
θ_5	τ_5	-1.0	$+1.0$
θ_6	τ_6	$+1.0$	-1.0

The second experiment used the same rewards, except that $\tau_1 = \tau_2$, $\tau_3 = \tau_4$ and $\tau_5 = \tau_6$.

In both cases, the adversary and agent were implemented using tabular actor-critic (Sutton & Barto, 2018). The procedure is similar to PAIRED (Dennis et al., 2020), except that we used perfect regret (corresponding to an antagonist that is optimal on each level and does not learn). Both the agent and adversary are effectively bandit agents, and the adversary was updated using regret as the reward, and the agent's reward was the negative of regret. When updating the agent, its reward is computed by taking the expectation over the adversary's distribution given the action the agent takes. Likewise, when computing the regret for updating the agent, we compute the agent's expected return over the environment selected by the adversary. At each iteration, we perform 5 updates of the adversary and then 5 updates of the agent.

We use $\gamma = 0.95$, entropy coefficient of 0.1, policy learning rate of 0.01 and value function learning rate of 1.0.

D.2.2. T-MAZE & MAZES

Every time we generate a new random level in this experiment, there is a 50% chance that it is a T-maze, and it is a normal maze otherwise. The T-maze's goal is invisible to the agent, and each T-maze looks like the levels in Figure 9. The boundaries are generated randomly. This has the effect that each T-maze in our buffer is unique, even though they play identically. The T-maze terminates once the agent moves with a reward of either $+1$ or -1, depending if the goal is reached or not.

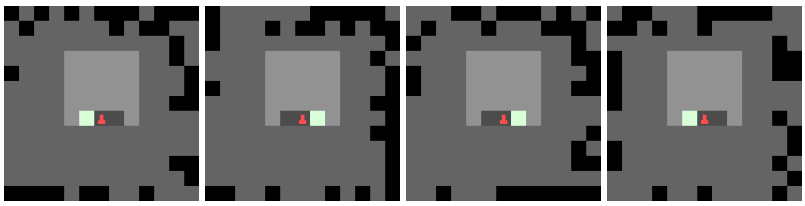

Figure 9. Several Examples of T-mazes

The mazes are generated using 25 walls sampled IID, similar to prior work (Jiang et al., 2021a). We use perfect Monte Carlo regret as the score. This is computed by first finding the shortest path from the agent's start location to the goal, and computing the number of steps that would take. The optimal return is then computed using this. The regret is then computed as the optimal return, minus the average return of the agent on this level.

Agent Architecture The agent architecture is similar to that used by prior work (Dennis et al., 2020; Jiang et al., 2021a). In particular, it observes the 5×5 window in front of and including the agent. The agent uses a 3×3 convolutional layer with 16 channels to process the input image. The agent's direction is embedded into a 5-dimensional vector. The image embedding is concatenated to the direction embedding and processed using a 256-feature LSTM (Hochreiter & Schmidhuber, 1997). The output of this is processed by two 32-hidden node fully-connected layers, one for the policy and one for the value estimate.

Evaluation Levels We evaluate the agent on a set of held-out standard test mazes used in prior work (Jiang et al., 2021a; Parker-Holder et al., 2022; Jiang et al., 2023). In particular, we use `SixteenRooms`, `SixteenRooms2`, `Labyrinth`,

`LabyrinthFlipped`, `Labyrinth2`, `StandardMaze`, `StandardMaze2`, `StandardMaze3`, `SmallCorridor` and `LargeCorridor`.

D.2.3. BLINDFOLD

In the blindfold experiment, levels are generated as normal, and then have a 50% chance of being "blindfold" levels, in which case the agent's observation is filled with zeros. The agent architecture and evaluation procedure are the same as in the T-maze case.

D.2.4. LEVER GAME

The lever environment is a one-step environment where there are 64 actions the agent can take. The observation consists of a 65-dimensional, one-hot-encoded vector. If the first dimension is active, that indicates that the correct answer is hidden. If any of the other i dimensions is activated, then the correct action is $i - 1$. We generate levels by first choosing a correct action and then, with 50% probability, choosing whether or not the correct answer is visible or not.

The reward for the visible case is $+1$, -1 for the correct and the incorrect answers respectively, and $+10$, -10 for the invisible case.

The agent's architecture consists of two 256 hidden node fully-connected layers, and then the same policy and value heads used in the other experiments. For hyperparameters, we use top-k prioritisation with $k = 32$, and a buffer size of 64 for PLR, where ReMiDi has two buffers of size 32 each.

D.2.5. BRAX

We use the Brax simulator, and restrict our episodes to only consist of 100 timesteps. The observation space is the standard Brax observations, with a one-hot encoded indication of whether the agent is in a type A or B level. In type B levels, since the joint positions do not matter, they are set to a dummy value of -1. The reward for type B levels is either $+10000$ or -10000 depending on if the agent performs the correct (but unobserved) action. Correct here means that the sign of the final action dimension is either positive or negative, depending on the level.

Since we cannot compute the optimal policy in a closed-form manner, we use the heuristic of setting the optimal return per level to be the same as obtained by a policy trained to convergence without disturbances. Since π_θ has access to the level, it can apply the necessary corrections to account for the particular disturbances of level θ.

Finally, during training, we divide the reward obtained by 10 (as it can become quite large), and use a running observation normalisation. We use the fully connected network architecture inspired by PureJaxRL (Lu et al., 2022). The network has two fully connected layers, then a recurrent cell, followed by two more fully connected layers for the actor and critic.

D.3. Hyperparameter Tuning

Table 9 contains the hyperparameters we used for our experiments.

We tuned PLR$^\perp$'s hyperparameters by training on standard mazes and choosing the hyperparameters that obtained the best performance over the evaluation levels. We used these for the minigrid experiments, and made slight changes for the simpler lever game. We performed a grid search over entropy coefficient of $\{0.01, 0.0\}$, learning rate of $\{0.001, 0.0001\}$, λ of $\{0.95, 0.98\}$, temperature of $\{0.3, 1.0\}$ and replay rate of $\{0.5, 0.8\}$. We searched over the same hyperparameters for the Brax experiments, separately for each environment. We use the same parameters for ReMiDi and minimax return as for PLR.

We used the same hyperparameters for ReMiDi, except that we have an inner buffer size of 256, and 4 outer buffers for the minigrid experiments. The inner buffer size of 256 performed better than a larger buffer of 1000 but is computationally more efficient.

Table 9. Hyperparameters

Parameter	Minigrid	Lever	Brax
PPO			
Number of Updates	30000	2500	200
γ	0.995		0.99
λ_{GAE}	0.95		0.95
PPO number of steps	256		100
PPO epochs	5		4
PPO minibatches per epoch	1		32
PPO clip range	0.2		
PPO # parallel environments	32		256
Adam learning rate	0.001		0.0001
Anneal LR	yes		
Adam ϵ	1e-5		
PPO max gradient norm	0.5		
PPO value clipping	yes		
return normalization	no		
value loss coefficient	0.5		
entropy coefficient	0.0		
PLR			
Replay rate, p	0.8	0.8	0.8
Buffer size, K	4000	64	512
Scoring function	Perfect	Perfect	Perfect
Prioritisation	Rank	Top K	Rank
Temperature, β	1.0	-	1.0
k	-	32	-
Staleness coefficient	0.3	0.3	0.3
ReMiDi			
Number of Adversaries	4	2	2
Inner Buffer Size	256	32	512
Number of Replays per adversary	1000	1000	25