MAESTRO: OPEN-ENDED ENVIRONMENT DESIGN FOR MULTI-AGENT REINFORCEMENT LEARNING

Mikayel Samvelyan[12] **Akbir Khan**[2] **Michael Dennis**[3] **Minqi Jiang**[12] **Jack Parker-Holder**[4] **Jakob Foerster**[4] **Roberta Raileanu**[1] **Tim Rocktäschel**[2]

[1]Meta AI [2]University College London [3]UC Berkeley [4]University of Oxford

samvelyan@meta.com

ABSTRACT

Open-ended learning methods that automatically generate a curriculum of increasingly challenging tasks serve as a promising avenue toward generally capable reinforcement learning (RL) agents. Existing methods adapt curricula independently over either environment parameters (in single-agent settings) or co-player policies (in multi-agent settings). However, the strengths and weaknesses of co-players can manifest themselves differently depending on environmental features. It is thus crucial to consider the dependency between the environment and co-player when shaping a curriculum in multi-agent domains. In this work, we use this insight and extend Unsupervised Environment Design (UED) to multi-agent environments. We then introduce *Multi-Agent Environment Design Strategist for Open-Ended Learning* (MAESTRO), the first multi-agent UED approach for two-player zero-sum settings. MAESTRO efficiently produces adversarial, joint curricula over both environments and co-players and attains minimax-regret guarantees at Nash equilibrium. Our experiments show that MAESTRO outperforms a number of strong baselines on competitive two-player environments, spanning discrete and continuous control.[1]

1 INTRODUCTION

The past few years have seen a series of remarkable achievements in producing deep reinforcement learning (RL) agents with expert (Vinyals et al., 2019; Berner et al., 2019; Wurman et al., 2022) and superhuman (Silver et al., 2016; Schrittwieser et al., 2020) performance in challenging competitive games. Central to these successes are adversarial training processes that result in curricula creating new challenges at the frontier of an agent's capabilities (Leibo et al., 2019; Yang et al., 2021). Such automatic curricula, or autocurricula, can improve the sample efficiency and generality of trained policies (Open Ended Learning Team et al., 2021), as well as induce an open-ended learning process (Balduzzi et al., 2019; Stanley et al., 2017) that continues to endlessly robustify an agent.

Autocurricula have been effective in multi-agent RL for adapting to different *co-players* in competitive games (Leibo et al., 2019; Garnelo et al., 2021; Baker et al., 2019; Bansal et al., 2018; Feng et al., 2021), where it is crucial to play against increasingly stronger opponents (Silver et al., 2018) and avoid being exploited by other agents (Vinyals et al., 2019). Here, algorithms such as self-play (Silver et al., 2018; Tesauro, 1995) and fictitious self-play (Brown, 1951; Heinrich et al., 2015) have proven especially effective. Similarly, in single-agent RL, autocurricula methods based on Unsupervised Environment Design (UED, Dennis et al., 2020) have proven effective in producing agents robust to a wide distribution of *environments* (Wang et al., 2019; 2020; Jiang et al., 2021a; Parker-Holder et al., 2022). UED seeks to adapt distributions over environments to maximise some metrics of interest. Minimax-regret UED seeks to maximise the *regret* of the learning agent, viewing this process as a game between a teacher that proposes challenging environments and a student that learns to solve them. At a Nash equilibrium of such games, the student policy provably reaches a minimax-regret policy over the set of possible environments, thereby providing a strong robustness guarantee.

However, prior works in UED focus on single-agent RL and thus ignore the dependency between the environment and the behaviours of other agents within it. In multi-agent domains, the behaviour of other agents plays a critical role in modulating the complexity and diversity of the challenges faced by a learning agent. For example, an empty environment that has no blocks to hide behind might be

[1]Videos of MAESTRO agents are available at https://sites.google.com/view/maestro-ued.

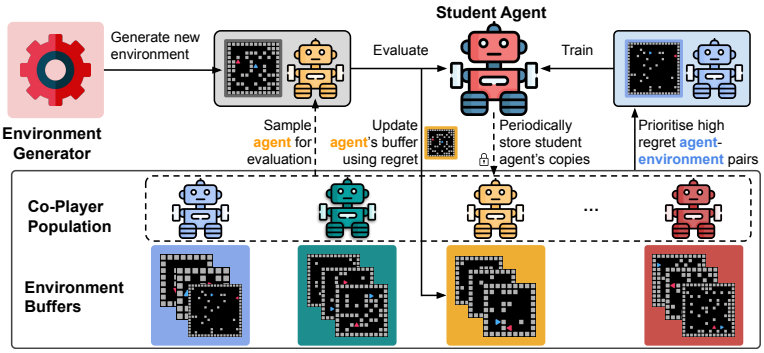


Figure 1: **A diagram of MAESTRO.** MAESTRO maintains a population of co-players, each having an individual buffer of high-regret environments. When new environments are sampled, the student's regret is calculated with respect to the corresponding co-player and added to the co-player's buffer. MAESTRO continually provides high-regret environment/co-player pairs for training the student.

most challenging when playing against opponent policies that attack head-on, whereas environments that are full of winding hallways might be difficult when playing against defensive policies. Robust RL agents should be expected to interact successfully with a wide assortment of other rational agents in their environment (Yang et al., 2021). Therefore, to become widely applicable, UED must be extended to include multi-agent dynamics as part of the environment design process.

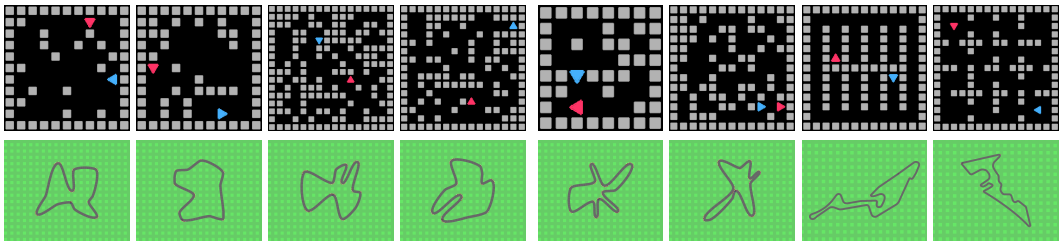
We formalise this novel problem as an *Underspecified Partially-Observable Stochastic Game* (UPOSG), which generalises UED to multi-agent settings. We then introduce ***M**ulti-**A**gent **E**nvironment Design **Str**ategist for **O**pen-Ended Learning* (MAESTRO), the first approach to train generally capable agents in two-player UPOSGs such that they are robust to changes in the environment and co-player policies. MAESTRO is a replay-guided approach that explicitly considers the dependence between agents and environments by jointly sampling over environment/co-player pairs using a regret-based curriculum and population learning (see Figure 1). In partially observable two-player zero-sum games, we show that at equilibrium, the MAESTRO student policy reaches a Bayes-Nash Equilibrium with respect to a regret-maximising distribution over environments. Furthermore, in fully observable settings, it attains a Nash-Equilibrium policy in every environment against every rational agent.

We assess the curricula induced by MAESTRO and a variety of strong baselines in two competitive two-player games, namely a sparse-reward grid-based LaserTag environment with discrete actions (Lanctot et al., 2017) and a dense-reward pixel-based MultiCarRacing environment with continuous actions (Schwarting et al., 2021). In both cases, MAESTRO produces more robust agents than baseline autocurriculum methods on out-of-distribution (OOD) human-designed environment instances against unseen co-players. Furthermore, we show that MAESTRO agents, trained only on randomised environments and having never seen the target task, can significantly outperform *specialist* agents trained directly on the target environment. Moreover, in analysing how the student's regret varies across environments and co-players, we find that a joint curriculum, as produced by MAESTRO, is indeed required for finding the highest regret levels, as necessitated by UED.

In summary, we make the following core contributions: (i) we provide the first formalism for multi-agent learning in underspecified environments, (ii) we introduce MAESTRO, a novel approach to jointly learn autocurricula over environment/co-player pairs, implicitly modelling their dependence, (iii) we prove MAESTRO inherits the theoretical property from the single-agent setting of implementing a minimax-regret policy at equilibrium, which corresponds to a Bayes-Nash or Nash equilibrium in certain settings, and (iv) by rigorously analysing the curriculum induced by MAESTRO and evaluating MAESTRO agents against strong baselines, we empirically demonstrate the importance of the joint curriculum over the environments and co-players.

2 Problem Statement and Preliminaries

In single-agent domains, the problem of *Unsupervised Environment Design* (UED) is cast in the framework of an underspecified POMDP (Dennis et al., 2020), which explicitly augments a standard POMDP with a set of *free parameters* controlling aspects of the environment subject to the design process. We extend this formalism to the multi-agent setting using stochastic games (Shapley, 1953).



(a) Start of training (b) Middle of training (c) End of training (d) Zero-shot evaluation

Figure 2: **Emergent complexity of autocurricula induced by MAESTRO**. Example environments provided to the MAESTRO student agent at (a) start, (b) middle, and (c) end of training. Levels become more complex over time. LaserTag levels (top row) increase in wall density and active engagement between the student and opponent. MultiCarRacing tracks (bottom row) become increasingly more challenging with many sharp turns. (d) Example held-out human-designed LaserTag levels and Formula 1 benchmark tracks (Jiang et al., 2021a) used for OOD evaluation. For the full list of evaluation environments see Appendix C.

An ***Underspecified Partially Observable Stochastic Game (UPOSG)*** is given by the tuple $\mathcal{M} = \langle n, \mathcal{A}, \mathcal{O}, \Theta, S, \mathcal{T}, \mathcal{I}, \mathcal{R}, \gamma \rangle$. $\mathcal{A}$, $\mathcal{O}$, and S denote the action, observation, and state spaces, respectively. Θ is the set of the environment's free parameters, such as possible positions of walls in a maze or levels of a game. These parameters can be distinct at every time step and are also incorporated into the transition function $\mathcal{T} : S \times \mathbf{A} \times \Theta \to \mathbf{\Delta}(S)$, where $\mathbf{A} \equiv \mathcal{A}^n$ is the joint action of all agents. Each agent draws individual observations according to the observation function $\mathcal{I} : S \times N \to \mathcal{O}$ and obtains reward according to the reward function $\mathcal{R} : S \times \mathbf{A} \times N \to \mathbb{R}$, where $N = \{1, \dots, n\}$. The discount factor is denoted by γ. Each configuration of environment parameters $\theta \in \Theta$ defines a specific instantiation of the environment $\mathcal{M}_\theta$, which is often referred to as a *level* (Jiang et al., 2021a; Parker-Holder et al., 2022). In this work, we will simply refer to a specific instance θ as an *environment* when clear from the context. We use $-i$ to notate all agents *except* i: $\boldsymbol{\pi}_{-i} = (\pi_1, \dots, \pi_{i-1}, \pi_{i+1}, \dots, \pi_N)$. For an agent i with a stochastic policy $\pi_i : \mathcal{O} \times \mathcal{A} \to [0, 1]$, we define the value in $\mathcal{M}_\theta$ with co-players $\boldsymbol{\pi}_{-i}$ as $V^\theta(\pi_i, \boldsymbol{\pi}_{-i}) = \mathbb{E}[\sum_{t=0}^{T} \gamma^t r_t^i]$ where r_t^i are the rewards achieved by agent i when following policy π_i in $\mathcal{M}_\theta$. The goal is to sequentially provide values for Θ and co-players $\boldsymbol{\pi}_{-i}$ during training such that the resulting policy π_i of agent i is robust to any possible environment parameterisation and co-player policies, i.e., $\pi_i = \arg\max_{\pi_i} \min_{\theta, \boldsymbol{\pi}_{-i}} V^\theta(\pi_i, \boldsymbol{\pi}_{-i})$. UPOSGs are general by nature and allow for cooperative, competitive, and mixed scenarios. Furthermore, Θ can represent different reward functions, game layouts, changes in observations, and environment dynamics. When $n = 1$, UPOSGs are identical to UPOMDPs for single-agent tasks. In the remainder of this work, we concentrate on competitive settings with $n = 2$ agents.

2.1 UED APPROACHES IN SINGLE-AGENT RL

A curriculum over the environment parameters θ can arise from a teacher maximising a utility function $U_t(\pi, \theta)$ based on the student's policy π. The most naive form of UED is *domain randomisation* (DR, Jakobi, 1997; Sadeghi & Levine, 2017), whereby environments are sampled uniformly at random, corresponding to a constant utility $U_t^U(\pi, \theta) = C$. Recent UED approaches use *regret* as the objective for maximisation (Dennis et al., 2020; Gur et al., 2021): $U_t^R(\pi, \theta) = \max_{\pi^* \in \Pi}\{\text{REGRET}^\theta(\pi, \pi^*)\} = \max_{\pi^* \in \Pi}\{V_\theta(\pi^*) - V_\theta(\pi)\}$, where π^* is the optimal policy on θ.

Empirically, regret-based objectives produce curricula of increasing complexity that result in more robust policies. Moreover, if the learning process reaches a Nash equilibrium, the student provably follows a minimax-regret policy (Dennis et al., 2020): $\pi \in \arg\min_{\pi_A \in \Pi}\{\max_{\theta, \pi_B \in \Theta, \Pi}\{\text{REGRET}^\theta(\pi_A, \pi_B)\}\}$, where Π and Θ are the strategy sets of the student and the teacher, respectively. While this is an appealing property, the optimal policy π^* is unknown in practice and must be approximated instead.

Prioritized Level Replay (PLR, Jiang et al., 2021a;b) continually curates an *environment buffer* containing the environments generated under domain randomisation with the highest learning potential, e.g., as measured by estimated regret. PLR alternates between evaluating new environments for learning potential and performing prioritised training of the agent on the environments with the highest learning potential so far found. By training the student only on curated high-regret environments, PLR provably results in a minimax-regret student policy at Nash equilibrium. MAESTRO makes use of PLR-style curation of environment buffers to discover high-regret environments during training.

2.2 Co-Player Autocurricula in Competitive Multi-Agent RL

The choice of co-players plays a crucial role for training agents in multi-agent domains (Leibo et al., 2019). A simple but effective approach is *self-play* (SP, Silver et al., 2018), whereby the agent always plays with copies of itself. In competitive two-player games, SP produces a curriculum in which opponents match each other in skill level. However, SP may cause cycles in the strategy space when agents forget how to play against previous versions of their policy (Garnelo et al., 2021).

Fictitious Self-Play (FSP, Heinrich et al., 2015) overcomes the emergence of such cycles by training an agent against a uniform mixture of all previous policies. However, FSP can result in wasting a large number of interactions against significantly weaker opponents.

Prioritized Fictitious Self-Play (PFSP, Vinyals et al., 2019) mitigates this potential inefficiency by matching agent A with a frozen opponent B from the set of candidates $\mathcal{C}$ with probability $\frac{f(\mathbb{P}[A \text{ beats } B])}{\sum_{C \in \mathcal{C}} f(\mathbb{P}[A \text{ beats } C])}$, where f defines the exact curriculum over opponents. For example, $f_{hard}(x) = (1-x)^p$, where $p \in \mathbb{R}_+$, forces PFSP to focus on the hardest opponents.

3 Method

3.1 An Illustrative Example

To highlight the importance of curricula over the joint space of environments and co-players, we provide an illustrative example of a simple two-player game in Table 1. Here, the goal of the regret-maximising teacher is to select an environment/co-player pair for the student. Ignoring the co-player and selecting the highest regret environment leads to choosing θ_3. Similarly, ignoring environments and selecting the highest regret co-player leads to π_C. This yields a suboptimal pair (θ_3, π_C) with $\text{Regret}(\theta_3, \pi_C) = 0.4$, whereas a teacher over the joint space yields the optimal pair (θ_1, π_A) with $\text{Regret}(\theta_1, \pi_A) = 0.6$. Thus, naively treating the environment and co-player as independent can yield a sub-optimal curriculum. Such overspecialisation toward a subset of environmental challenges at the expense of overall robustness commonly emerges in multi-agent settings (Garnelo et al., 2021).

Table 1: **An illustrative two-player game.** Rows correspond to co-player policies and $\Pi = \{\pi_A, \pi_B, \pi_C\}$. Columns indicate different environments and $\Theta = \{\theta_1, \theta_2, \theta_3, \theta_4\}$. The payoff matrix represents the regret of the student on pair (θ, π).

	θ_1	θ_2	θ_3	θ_4	$\mathbb{E}_{\theta \sim \Theta}$
π_A	0.6	0.1	0.4	0.2	0.325
π_B	0.1	0.5	0.4	0.3	0.325
π_C	0.2	0.4	0.4	0.4	0.35
$\mathbb{E}_{\pi \sim \Pi}$	0.3	0.33	0.4	0.3	

3.2 Multi-Agent Environment Design Strategist for Open-Ended Learning

In this section, we describe a new multi-agent UED approach called ***M**ulti-**A**gent **E**nvironment Design **Str**ategist for **O**pen-Ended Learning* (Maestro) which induces a regret-based autocurricula jointly over environments and co-players.

Maestro is a replay-guided approach (Jiang et al., 2021a) that relies on an environment generator to continuously create new environment instances. Rather than storing a single environment buffer, as done by PLR, Maestro maintains a population of policies $\mathfrak{B}$ with each policy assigned its own environment buffer. In each environment θ produced by the generator, Maestro evaluates the student's performance against a non-uniform mixture of co-player policies in $\mathfrak{B}$. High-regret environment/co-player pairs are then stored, along with the student agent's regret estimate for that pair, in co-player's corresponding environment buffer.

Algorithm 1: Maestro

Input: Environment generator Θ
Initialise: Student policy π, co-player population $\mathfrak{B}$
Initialise: Environment buffers $\forall \pi' \in \mathfrak{B}, \mathbf{\Lambda}(\pi') := \emptyset$.
for $i = \{1, 2, \dots\}$ **do**
  for *many episodes* **do**
    $\pi' \sim \mathfrak{B}$ ▷ Sample co-player via Eq 1
    Sample replay decision ▷ see Section 3.2.2
    if *replaying* **then**
      $\theta \sim \mathbf{\Lambda}(\pi')$ ▷ Sample a replay environment
      Collect trajectory τ of π using (θ, π')
      Update π with rewards $\boldsymbol{R}(\tau)$
    else
      $\theta \sim \Theta$ ▷ Sample a random environment
      Collect trajectory τ of π using (θ, π')
    Compute regret score $S = \widetilde{Regret}(\theta, \pi')$
    Update $\mathbf{\Lambda}(\pi')$ with θ using score S
  $\mathfrak{B} \leftarrow \mathfrak{B} \cup \{\pi_i^{\perp}\}, \mathbf{\Lambda}(\pi_i^{\perp}) := \emptyset$ ▷ frozen weights

MAESTRO maintains a dynamic population of co-players of various skill levels throughout training (Czarnecki et al., 2020). Figure 1 presents the overall training paradigm of MAESTRO and Algorithm 1 provides its pseudocode. We note that when applied to a fixed singleton environment, MAESTRO becomes a variation of PFSP (Vinyals et al., 2019), where the mixture of policies from the population is computed based on the agent's regret estimate, rather than the probability of winning.

3.2.1 MAINTAINING A POPULATION OF CO-PLAYERS

A key issue of using replay-guided autocurricula for multi-agent settings is nonstationarity. Specifically, using PLR with SP results in inaccurate regret estimates over environment/co-player pairs, as the co-player policies evolve over training. MAESTRO overcomes this issue by maintaining a population of past policies (Lanctot et al., 2017). This approach confers several key benefits. First, re-encountering past agents helps avoid cycles in strategy space (Balduzzi et al., 2019; Garnelo et al., 2021). Second, MAESTRO maintains accurate regret estimates in the face of nonstationarity by employing a separate environment buffer for each policy in the population. Third, MAESTRO always optimises a *single* policy throughout training, rather than a set of distinct policies which can be computationally expensive.

3.2.2 CURATING THE ENVIRONMENT/CO-PLAYER PAIRS

To optimise for the global regret over the joint environment/co-player space, a MAESTRO student is trained to best respond to a non-uniform mixture of policies from a population by prioritising training against co-players with high-regret environments in their buffers:

$$\text{CO-PLAYER}^{\text{HR}} \in \arg\max_{\pi' \in \mathfrak{B}} \{ \max_{\theta \in \mathbf{\Lambda}(\pi')} \widetilde{Regret}(\theta, \pi') \}, \tag{1}$$

where $\mathfrak{B}$ is the co-player population, $\mathbf{\Lambda}(\pi')$ is the environment buffer of agent π', and $\widetilde{Regret}$ is the estimated regret of student for the pair (θ, π'). To ensure that the student learns to best respond to high-regret co-players as well as to the entire population $\mathfrak{B}$, we enforce all members of $\mathfrak{B}$ to be assigned a minimum probability $\frac{\lambda}{N}$. For instance, if Equation (1) returns a single highest-regret co-player, then the resulting prioritised distribution assigns a weight of $\frac{N-\lambda(N-1)}{N}$ to the highest-regret co-player and weight of $\frac{\lambda}{N}$ to the remaining co-players. For each co-player $\pi' \in \mathfrak{B}$, MAESTRO maintains a PLR environment buffer $\mathbf{\Lambda}(\pi')$ with the top-K high-regret levels. Once the co-player is sampled, we make a replay decision: with probability p, we use a bandit to sample a training environment from $\mathbf{\Lambda}(\pi')$,[2] and with probability $1-p$, we sample a new environment for evaluation from the environment generator. Similar to Robust PLR (Jiang et al., 2021a), we only update the student policy on environments sampled from the environment buffer. This provides MAESTRO with strong robustness guarantees, which we discuss in Section 3.3.

3.3 ROBUSTNESS GUARANTEES OF MAESTRO

We analyse the expected behaviour of MAESTRO if the system reaches equilibrium: does MAESTRO produce a regret-maximising distribution over environments and co-players, and is the policy optimal with respect to these distributions? We cast this problem in terms of Bayesian Nash Equilibrium (BNE) behaviour on individual environments. BNE is an extension of Nash Equilibrium (NE) where each co-player j has an unknown *type* parameter θ^j, which affects the dynamics of the game and is only known to that player. The distribution over these parameters $\tilde{\Theta}^N$ is assumed to be common knowledge. Equilibria are then defined as the set of policies, conditioned on their unknown type, each being a best response to the policies of the other players. That is, for policy π_j of any player j,

$$\pi_j \in \arg\max_{\hat{\pi}_j \in \Pi_j} \{ \mathop{\mathbb{E}}_{\theta^N \in \tilde{\Theta}^N} [U(\hat{\pi}_j(\theta^j), \boldsymbol{\pi}_{-j}(\theta^{-j}))] \}. \tag{2}$$

In MAESTRO, we can assume each co-player is effectively omniscient, as each is co-evolved for maximal performance against the student on the environments it is paired with in its high-regret environment buffer. In contrast, the student is conferred no special advantages and has access to only the standard observations. We formalise this setting as a $-i$-knowing game. This game corresponds to the POSG with the same set of players, action space, rewards, and states as the original UPOSG, but with θ sampled at the first time step and provided to co-player agents $-i$ as part of their observation.

[2] Sampling is based on environment's regret score, staleness, and other features following (Jiang et al., 2021b).

Definition 1. *The -i-knowing-game of an UPOSG* $M = (N, A = \times_{i \in N} A_i, O = \times_{i \in N} O_i, \Theta, R = \times_{i \in N} R_i, S, I = \times_{i \in N} I_i, T)$ *with parameter distribution* $\tilde{\theta}$ *is defined to be the POSG* $K = (N' = N, A'_i = A_i, O'_i = O_i + \{\Theta \text{ if } i \in -i\}, R'_i = R_i, S' = S, I'_i = I_i + \{\theta \text{ if } \in -i\}, T' = T(\theta))$ *where* θ *is sampled from the distribution* $\tilde{\theta}$ *on the first time step.*

We thus arrive at our main theorem, followed by a convenient and natural corollary for fully observable settings. We include the full proofs in Appendix A.

Theorem 1. *In two-player zero-sum settings, the* MAESTRO *student at equilibrium implements a Bayes-Nash Equilibrium of the* $-i$*-knowing game, over a regret-maximising distribution of levels.*

Corollary 1. *In fully-observable two-player zero-sum settings, the* MAESTRO *student at equilibrium implements a Nash Equilibrium in each environment in the support of the environment distribution.*

Informally, the proof follows from the observation that the $-i$-knowing game in a fully observable setting is equivalent to the original distribution of environments, as there is no longer an information asymmetry between the student and co-players. Moreover, the NE strategy on this distribution of environment instances would be a NE strategy on each instance individually, given that they are fully observable. This argument is formalised in Appendix A.

4 EXPERIMENTAL SETTING

Our experiments aim to understand (1) the interaction between autocurricula over environments and co-players in multi-agent UED, (2) its impact on zero-shot transfer performance of student policies to unseen environments and opponents, and (3) the emergent complexity of the environments provided to the student agent under autocurricula. To this end, we evaluate methods in two distinct domains: discrete control with sparse rewards, and continuous control with dense rewards. We assess student robustness in OOD human-designed environments against previously unseen opponents. Given its strong performance and usage in related works, PPO (Schulman et al., 2017) serves as the base RL algorithm in our experiments. We provide full environment descriptions in Appendix C and detail our model architecture and hyperparameter choices in Appendix D.

Baselines and Ablations We compare MAESTRO against two key baselines methods producing autocurricula over environments: domain randomization (DR; Jakobi, 1997), and (Robust) PLR (Jiang et al., 2021a), a state-of-the-art UED baseline. For co-player curricula, we consider SP, FSP, and PFSP, popular methods that underlie breakthroughs such as AlphaGo (Silver et al., 2016) and AlphaStar (Vinyals et al., 2019). Since these baselines independently produce either curricula over environments or over co-players, our choice of baselines results in a combination of 6 joint curriculum baselines over the environment/co-player space. We present further ablations investigating the importance of MAESTRO's co-player selection mechanism in Appendix B.

4.1 ENVIRONMENTS

LaserTag is a grid-based, two-player zero-sum game proposed by Lanctot et al. (2017) where agents aim to tag each other with laser beams. Success in LaserTag requires agents to master sophisticated behaviours, including chasing opponents, hiding behind walls, keeping clear of long corridors, and maze solving. Each agent observes the 5×5 grid area in front of it and can turn right or left, move forward, and shoot. Upon tagging an opponent, the agent and the opponent receive a reward of 1 and -1, respectively, and the episode terminates. LaserTag training environments are generated by randomly sampling grid size, wall locations, and the initial locations and directions of agents.

MultiCarRacing (MCR, Schwarting et al., 2021) is a high-dimensional, pixel-based continuous control environment with dense rewards. Two agents compete by driving a full lap around the track. Each track consists of n tiles. Agents receive a reward of $1000/n$ or $500/n$ for reaching a tile first or second, respectively. Agents receive a top-down, egocentric $96 \times 96 \times 3$ partial observation and can collide, push, or block each other, allowing for complex emergent behaviour and non-transitive dynamics. The action space is 3-dimensional, controlling changes in acceleration, braking, and steering. All training tracks used for training agents are generated by sampling 12 random control points defining a Bézier curve forming the track.

Figure 2 shows example LaserTag and MCR environments, including human-designed OOD test environments. MCR test environments are the Formula 1 CarRacing tracks from (Jiang et al., 2021a).

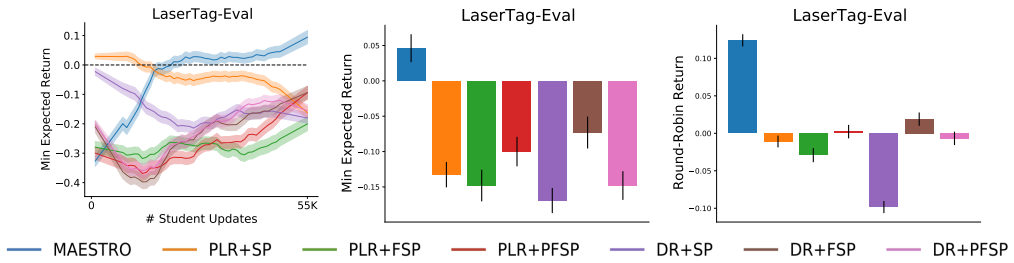


Figure 3: **LaserTag Cross-Play Results**. (Left) MER on LaserTag evaluation environments during training. (Middle) MER and (Right) round-robin returns at the end of training (mean and standard error over 10 seeds).

5 Results and Discussion

5.1 Cross-Play Results

To assess the robustness of the approaches, we evaluate all pairs of methods in cross-play on all OOD human-designed environments evaluation (13 LaserTag levels and 20 Formula 1 tracks for MCR). Agents are trained for 55000 PPO updates (≈1B steps) on LaserTag and 4500 updates (≈40M steps) on MCR. Full results on individual environments are included in Appendix E.2.

Figure 3 shows the cross-play results on LaserTag throughout training, as well as at the end of training. We use the minimum expected return (MER) against any baseline to estimate the exploitability of each method against all other baselines. Less exploitable agents should attain higher MER across all other agents. We can see that MAESTRO outperforms all baselines in MER, being the only method with positive MER, i.e., it is not exploited by any other baseline. Although SP-based methods achieve an early performance lead due to more frequent policy updates, population-based methods (FSP, PFSP, and MAESTRO) eventually outperform them.

Figure 4 reports the MCR results on challenging F1 tracks, as well as the procedurally generated training tracks. Here, MAESTRO outperforms all baselines individually across all benchmark tracks, while also being the least exploitable policy. PLR-based baselines outperform DR, underscoring the benefits of a curriculum over environments on this task. Nonetheless, MAESTRO produces the most robust agents, highlighting the importance of a joint curriculum over environments and co-players.

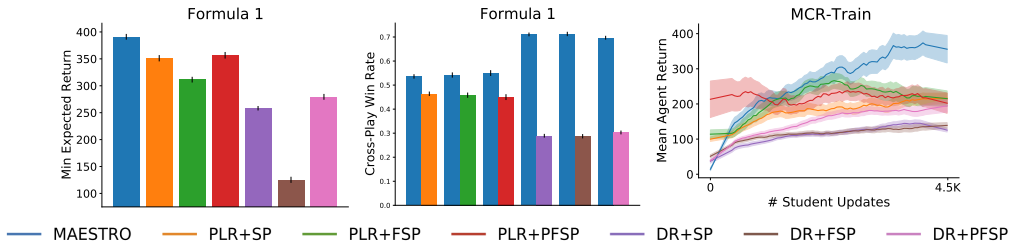


Figure 4: **MultiCarRacing Cross-Play Results**: (Left) MER on Formula 1. (Middle) Cross-play win rate between MAESTRO and baselines on Formula 1. (Right) Mean agent return of methods throughout training. Plots show the mean and standard error across 5 training seeds.

5.2 Importance of the curriculum over the environment/co-player space

Figure 5 shows the co-player×environment regret landscape of a MAESTRO student agent on randomly-sampled environments against each co-player from MAESTRO's population at the end of training (as done in the motivating example in Section 3.1). We observe that high regret estimates depend on both the choice of the environment and the co-player. Most importantly, maximising mean regrets over environments and co-players independently does not lead to maximum regret over the joint space, highlighting the importance of the curricula over the joint environment/co-player space.

5.3 Evaluation Against Specialist Agents

While we show that MAESTRO outperforms baselines trained on a large number of environments, we are also interested in evaluating MAESTRO against *specialist* agents trained only on single, *fixed* environments. Figure 6 shows that, despite having never seen the target environments, the MAESTRO agent beats specialist agents trained exclusively on them for the same number of updates. This is

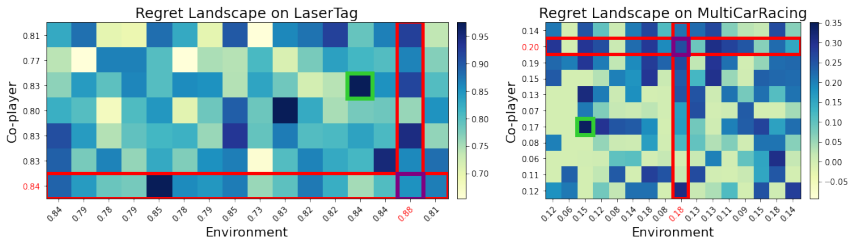


Figure 5: **Regret landscapes on LaserTag and MultiCarRacing.** Shown are regret estimates of the student on 16 random environment (columns) against each policy from MAESTRO's co-player population (rows). Highlighted in red are the regret estimates of the co-player and environment with the highest mean values when considered in isolation, whereas in green we highlight the overall highest regret environment/co-player pair.

possible because the MAESTRO agent was trained with an autocurriculum that promotes the robustness of the trained policy, allowing it to transfer better to unseen opponents on OOD environments. These results demonstrate the potential of multi-agent UED in addressing key problems in multi-agent RL, such as exploration (Leonardos et al., 2021) and co-player overfitting (Lanctot et al., 2017).

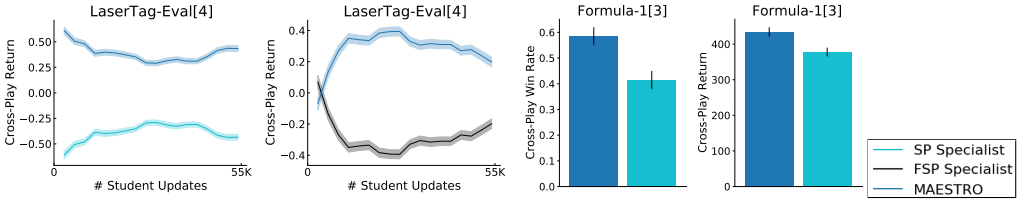


Figure 6: **Cross-Play vs Specialists**. Cross-play evaluation against specialist agents trained directly on the target task. We use different specialist agents for each fixed environment. Shown are averaged metrics across 4 target LaserTag levels and 3 Formula 1 tracks. Full results on individual environments are included in Appendix E.1.

5.4 EMERGENT CURRICULUM ANALYSIS

We analyse the curriculum over environments induced by MAESTRO and compare it with the curricula generated by the baselines. Figure 7 demonstrates that, as training progresses, MAESTRO provides agents with LaserTag environments with an increasing density of walls, which contributes to a gradual increase in environment complexity. Initially, the size of the grid also increases as the teacher challenges the student to navigate the mazes. However, after more adept opponent agents enter the population, larger mazes yield lower regret. Instead, MAESTRO starts prioritising smaller grid sizes compared to other baselines. Smaller environments with high wall density challenge the student to navigate maze structures while succeeding in more frequent multi-agent interactions against increasingly more capable opponents throughout an episode. Figure 2c illustrates such challenging environments, which, due to the smaller grid, require the student to engage with the opponent earlier in the episode. PLR-based methods also prioritise high wall density but do not shrink the grid size to prioritise competitive interactions to the same extent as MAESTRO.

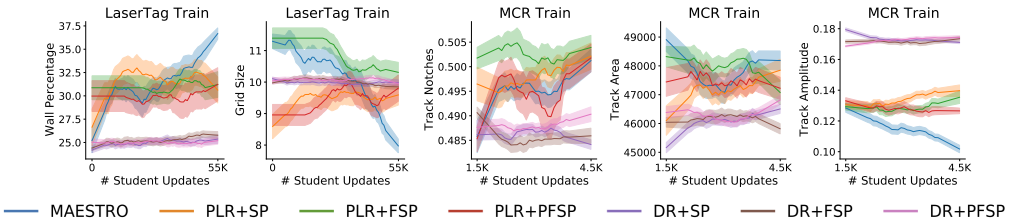


Figure 7: **Characteristics of emergent autocurricula in LaserTag and MultiCarRacing.**

We observe similar emergent complexity in the MCR domain. Here the PLR-based baselines and MAESTRO all prioritise tracks that have a high number of notches (i.e. the non-convex parts of the track) and enclosed area. Here, as in the LaserTag domain, MAESTRO gradually shrinks the track amplitude, which corresponds to a lower intensity of the winding portions of the track. Such tracks may contain more segments where competitive interactions can play out without sacrificing overall episodic return, thus inducing policies more adept at the competitive aspects of this domain. Figure 2 illustrates randomly sampled environments curated by MAESTRO at different stages of training.

6 Related Work

Unsupervised Environment Design (UED, Dennis et al., 2020) is a family of methods that provide an agent with a sequence of environments for training robust policies. The simplest UED approach is *Domain Randomisation* (DR, Jakobi, 1997; Sadeghi & Levine, 2017) which has demonstrated strong empirical gains in domains such as robotics (Tobin et al., 2017; James et al., 2017) and magnetic control of tokamak plasmas (Degrave et al., 2022). PAIRED (Dennis et al., 2020; Gur et al., 2021) trains an environment generator that maximises the student's regret, approximated as the difference in return between the student and an antagonist agent. *Prioritized Level Replay* (PLR, Jiang et al., 2021a;b) curates environment instances (i.e., levels) for training, by performing a random search of domain randomised levels for those with high learning potential, e.g. as measured by estimated regret. *ACCEL* (Parker-Holder et al., 2022) is a replay-guided UED approach that extends PLR by making edits to previously high-regret environments. Several methods generate curricula by adapting the environment parameters in response to the agent's performance (Portelas et al., 2019; Matiisen et al., 2020; Klink et al., 2019; Eimer et al., 2021). This adaptation is largely heuristic-driven, without the robustness guarantees shared by minimax-regret UED methods. Notably, all these methods focus on single-agent RL, while MAESTRO is designed for the two-player multi-agent setting.

Many prior works study curricula over opponents in two-player zero-sum settings. The most naive approach, self-play (SP), consists of pitting the agent against a copy of itself. Combined with search, SP has led to superhuman performances in board games such as Backgammon (Tesauro, 1995), Chess and Go (Silver et al., 2016). Zinkevich et al. (2007) uses self-play with regret minimisation for achieving Nash-Equilibrium, an approach that led to superhuman performance in Poker (Brown & Sandholm, 2018; 2019). *Fictitious self-play* (FSP) learns a best-response to the uniform mixture of all previous versions of the agent (Brown, 1951; Leslie & Collins, 2006; Heinrich et al., 2015). *Prioritised fictitious self-play* (PFSP, Vinyals et al., 2019) trains agents against a non-uniform mixture of policies based on the probability of winning against each policy. PFSP is a practical variant of *Policy-Space Response Oracles* (PSRO, Lanctot et al., 2017), a general population learning framework, whereby new policies are trained as best responses to a mixture of previous policies following a meta-strategy solver. MAESTRO is related to PSRO but adapted for UPOSGs. In MAESTRO, the population meta-strategy is based on the student's regret when playing against policies on environments observed during training. Unlike our work, these prior autocurricula methods for competitive multi-agent environments do not directly consider variations of the environment itself.

Several prior works have applied DR in multi-agent domains. Randomly modifying the environment has proven critical for the emergence of complex behaviours in Hide&Seek (Baker et al., 2019) and Capture the Flag (Jaderberg et al., 2019). In XLand (Open Ended Learning Team et al., 2021), a curriculum is provided over both environments and tasks to create general learners. This work differs from ours in multiple aspects. Open Ended Learning Team et al. (2021) uses handcrafted heuristics and rejection sampling for selecting environments for training and evaluating agents, while MAESTRO automatically selects environments based on regret rather than hand-coded heuristics. Further, unlike the autocurricula used in XLand, MAESTRO does not rely on population-based training, a computationally expensive algorithm for tuning the autocurriculum hyperparameters.

7 Conclusion and Future Work

In this paper, we provided the first formalism for multi-agent learning in underspecified environments and introduced MAESTRO, an approach for producing a joint autocurriculum over both environments and co-players. Moreover, we proved that MAESTRO attains minimax-regret robustness guarantees at Nash equilibrium. Empirically MAESTRO produces agents that are more robust to the environment and co-player variations than a number of strong baselines in two challenging domains. MAESTRO agents even outperform specialist agents in these domains. Our work opens up many interesting directions for future work. MAESTRO could be extended to n-player games, as well as cooperative or mixed settings. Furthermore, MAESTRO could be combined with search-based methods in order to further improve sample efficiency and generalisation. Another interesting open-question is identifying conditions whereby such an algorithm can provably converge to Nash equilibrium in two-player zero-sum settings. Finally, this work is limited to training only a single policy for each of the multi-agent UED approaches. Concurrent and continued training of several unique policies in underspecified multi-agent problems could be a generally fruitful research direction.

REFERENCES

Bowen Baker, Ingmar Kanitscheider, Todor Markov, Yi Wu, Glenn Powell, Bob McGrew, and Igor Mordatch. Emergent tool use from multi-agent autocurricula, 2019. URL https://arxiv.org/abs/1909.07528.

David Balduzzi, Marta Garnelo, Yoram Bachrach, Wojciech Czarnecki, Julien Perolat, Max Jaderberg, and Thore Graepel. Open-ended learning in symmetric zero-sum games. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 434–443. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/balduzzi19a.html.

Chris Bamford, Shengyi Huang, and Simon Lucas. Griddly: A platform for ai research in games, 2020. URL https://arxiv.org/abs/2011.06363.

Christopher Bamford, Minqi Jiang, Mikayel Samvelyan, and Tim Rocktäschel. Griddlyjs: A web ide for reinforcement learning, 2022. URL https://arxiv.org/abs/2207.06105.

Trapit Bansal, Jakub Pachocki, Szymon Sidor, Ilya Sutskever, and Igor Mordatch. Emergent complexity via multi-agent competition. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Sy0GnUxCb.

Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemyslaw Debiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, Rafal Józefowicz, Scott Gray, Catherine Olsson, Jakub Pachocki, Michael Petrov, Henrique Pondé de Oliveira Pinto, Jonathan Raiman, Tim Salimans, Jeremy Schlatter, Jonas Schneider, Szymon Sidor, Ilya Sutskever, Jie Tang, Filip Wolski, and Susan Zhang. Dota 2 with large scale deep reinforcement learning. *CoRR*, abs/1912.06680, 2019.

George W Brown. Iterative solution of games by fictitious play. *Activity analysis of production and allocation*, 13(1):374–376, 1951.

Noam Brown and Tuomas Sandholm. Superhuman ai for heads-up no-limit poker: Libratus beats top professionals. *Science*, 359(6374):418–424, 2018.

Noam Brown and Tuomas Sandholm. Superhuman ai for multiplayer poker. *Science*, 365(6456): 885–890, 2019. doi: 10.1126/science.aay2400. URL https://www.science.org/doi/abs/10.1126/science.aay2400.

Wojciech M Czarnecki, Gauthier Gidel, Brendan Tracey, Karl Tuyls, Shayegan Omidshafiei, David Balduzzi, and Max Jaderberg. Real world games look like spinning tops. *Advances in Neural Information Processing Systems*, 33:17443–17454, 2020.

Jonas Degrave, Federico Felici, Jonas Buchli, Michael Neunert, Brendan D. Tracey, Francesco Carpanese, Timo Ewalds, Roland Hafner, Abbas Abdolmaleki, Diego de Las Casas, Craig Donner, Leslie Fritz, Cristian Galperti, Andrea Huber, James Keeling, Maria Tsimpoukelli, Jackie Kay, Antoine Merle, J-M. Moret, Seb Noury, Federico Pesamosca, David G. Pfau, Olivier Sauter, Cristian Sommariva, Stefano Coda, B. Duval, Ambrogio Fasoli, Pushmeet Kohli, Koray Kavukcuoglu, Demis Hassabis, and Martin A. Riedmiller. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, 602:414 – 419, 2022.

Michael Dennis, Natasha Jaques, Eugene Vinitsky, Alexandre Bayen, Stuart Russell, Andrew Critch, and Sergey Levine. Emergent complexity and zero-shot transfer via unsupervised environment design. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Theresa Eimer, André Biedenkapp, Frank Hutter, and Marius Lindauer. Self-paced context evaluation for contextual reinforcement learning. In *The International Conference on Machine Learning*. 2021.

Xidong Feng, Oliver Slumbers, Ziyu Wan, Bo Liu, Stephen McAleer, Ying Wen, Jun Wang, and Yaodong Yang. Neural Auto-Curricula in Two-Player Zero-Sum Games. In *Advances in Neural Information Processing Systems*, volume 34, pp. 3504–3517. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/hash/1cd73be1e256a7405516501e94e892ac-Abstract.html.

Marta Garnelo, Wojciech Marian Czarnecki, Siqi Liu, Dhruva Tirumala, Junhyuk Oh, Gauthier Gidel, Hado van Hasselt, and David Balduzzi. Pick your battles: Interaction graphs as population-level objectives for strategic diversity, 2021. URL https://arxiv.org/abs/2110.04041.

Izzeddin Gur, Natasha Jaques, Yingjie Miao, Jongwook Choi, Manoj Tiwari, Honglak Lee, and Aleksandra Faust. Environment generation for zero-shot compositional reinforcement learning. In *Advances in Neural Information Processing Systems*, 2021.

Johannes Heinrich, Marc Lanctot, and David Silver. Fictitious self-play in extensive-form games. In *International conference on machine learning*, pp. 805–813. PMLR, 2015.

Max Jaderberg, Wojciech M. Czarnecki, Iain Dunning, Luke Marris, Guy Lever, Antonio Garcia Castañ eda, Charles Beattie, Neil C. Rabinowitz, Ari S. Morcos, Avraham Ruderman, Nicolas Sonnerat, Tim Green, Louise Deason, Joel Z. Leibo, David Silver, Demis Hassabis, Koray Kavukcuoglu, and Thore Graepel. Human-level performance in 3d multiplayer games with population-based reinforcement learning. *Science*, 364(6443):859–865, may 2019.

Nick Jakobi. Evolutionary robotics and the radical envelope-of-noise hypothesis. *Adaptive Behavior*, 6(2):325–368, 1997.

Stephen James, Andrew J. Davison, and Edward Johns. Transferring end-to-end visuomotor control from simulation to real world for a multi-stage task. In *1st Conference on Robot Learning*, 2017.

Minqi Jiang, Michael Dennis, Jack Parker-Holder, Jakob Foerster, Edward Grefenstette, and Tim Rocktäschel. Replay-guided adversarial environment design. In *Advances in Neural Information Processing Systems*. 2021a.

Minqi Jiang, Edward Grefenstette, and Tim Rocktäschel. Prioritized level replay. In *The International Conference on Machine Learning*. 2021b.

Pascal Klink, Hany Abdulsamad, Boris Belousov, and Jan Peters. Self-paced contextual reinforcement learning. In *Conference on Robot Learning*. 10 2019.

Marc Lanctot, Vinicius Zambaldi, Audrunas Gruslys, Angeliki Lazaridou, Karl Tuyls, Julien Perolat, David Silver, and Thore Graepel. A unified game-theoretic approach to multiagent reinforcement learning, 2017. URL https://arxiv.org/abs/1711.00832.

Joel Z. Leibo, Vinicius Zambaldi, Marc Lanctot, Janusz Marecki, and Thore Graepel. Multi-agent reinforcement learning in sequential social dilemmas, 2017. URL https://arxiv.org/abs/1702.03037.

Joel Z. Leibo, Edward Hughes, Marc Lanctot, and Thore Graepel. Autocurricula and the emergence of innovation from social interaction: A manifesto for multi-agent intelligence research. *ArXiv*, abs/1903.00742, 2019.

Stefanos Leonardos, Georgios Piliouras, and Kelly Spendlove. Exploration-exploitation in multi-agent competition: Convergence with bounded rationality. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=OSLVL-tIBei.

David S Leslie and Edmund J Collins. Generalised weakened fictitious play. *Games and Economic Behavior*, 56(2):285–298, 2006.

Xiaoteng Ma. Car racing with pytorch. 2019. URL https://github.com/xtma/pytorch_car_caring.

Tambet Matiisen, Avital Oliver, Taco Cohen, and John Schulman. Teacher-student curriculum learning. *IEEE Trans. Neural Networks Learn. Syst.*, 31(9):3732–3740, 2020.

Open Ended Learning Team, Adam Stooke, Anuj Mahajan, Catarina Barros, Charlie Deck, Jakob Bauer, Jakub Sygnowski, Maja Trebacz, Max Jaderberg, Michaël Mathieu, Nat McAleese, Nathalie Bradley-Schmieg, Nathaniel Wong, Nicolas Porcel, Roberta Raileanu, Steph Hughes-Fitt, Valentin Dalibard, and Wojciech Marian Czarnecki. Open-ended learning leads to generally capable agents. *CoRR*, abs/2107.12808, 2021.

Jack Parker-Holder, Minqi Jiang, Michael Dennis, Mikayel Samvelyan, Jakob Foerster, Edward Grefenstette, and Tim Rocktäschel. Evolving curricula with regret-based environment design, 2022. URL https://arxiv.org/abs/2203.01302.

Rémy Portelas, Cédric Colas, Katja Hofmann, and Pierre-Yves Oudeyer. Teacher algorithms for curriculum learning of deep RL in continuously parameterized environments. In Leslie Pack Kaelbling, Danica Kragic, and Komei Sugiura (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.

Fereshteh Sadeghi and Sergey Levine. CAD2RL: real single-image flight without a single real image. In Nancy M. Amato, Siddhartha S. Srinivasa, Nora Ayanian, and Scott Kuindersma (eds.), *Robotics: Science and Systems XIII, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, July 12-16, 2017*, 2017.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy Lillicrap, and David Silver. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588 (7839):604–609, dec 2020.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *ArXiv*, abs/1707.06347, 2017.

Wilko Schwarting, Tim Seyde, Igor Gilitschenski, Lucas Liebenwein, Ryan Sander, Sertac Karaman, and Daniela Rus. Deep latent competition: Learning to race using visual control policies in latent space, 2021.

L. S. Shapley. Stochastic games. *Proceedings of the National Academy of Sciences*, 39(10):1095–1100, 1953. doi: 10.1073/pnas.39.10.1095. URL https://www.pnas.org/doi/abs/10.1073/pnas.39.10.1095.

David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Vedavyas Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy P. Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529:484–489, 2016.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018. doi: 10.1126/science.aar6404.

Kenneth O Stanley, Joel Lehman, and Lisa Soros. Open-endedness: The last grand challenge you've never heard of. *While open-endedness could be a force for discovering intelligence, it could also be a component of AI itself*, 2017.

Gerald Tesauro. Temporal difference learning and td-gammon. *Commun. ACM*, 38(3):58–68, mar 1995. ISSN 0001-0782.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2017, Vancouver, BC, Canada, September 24-28, 2017*, pp. 23–30. IEEE, 2017.

Oriol Vinyals, Igor Babuschkin, Wojciech M. Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H. Choi, Richard Powell, Timo Ewalds, Petko Georgiev, Junhyuk Oh, Dan Horgan, Manuel Kroiss, Ivo Danihelka, Aja Huang, Laurent Sifre, Trevor Cai, John P. Agapiou, Max Jaderberg, Alexander Sasha Vezhnevets, Rémi Leblond, Tobias Pohlen, Valentin Dalibard, David

Budden, Yury Sulsky, James Molloy, Tom L. Paine, Çaglar Gülçehre, Ziyu Wang, Tobias Pfaff, Yuhuai Wu, Roman Ring, Dani Yogatama, Dario Wünsch, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy P. Lillicrap, Koray Kavukcuoglu, Demis Hassabis, Chris Apps, and David Silver. Grandmaster level in starcraft II using multi-agent reinforcement learning. *Nat.*, 575(7782): 350–354, 2019. doi: 10.1038/s41586-019-1724-z.

Rui Wang, Joel Lehman, Jeff Clune, and Kenneth O. Stanley. Paired open-ended trailblazer (POET): endlessly generating increasingly complex and diverse learning environments and their solutions. *CoRR*, abs/1901.01753, 2019.

Rui Wang, Joel Lehman, Aditya Rawal, Jiale Zhi, Yulun Li, Jeffrey Clune, and Kenneth Stanley. Enhanced POET: Open-ended reinforcement learning through unbounded invention of learning challenges and their solutions. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9940–9951. PMLR, 13–18 Jul 2020.

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.

Yaodong Yang, Jun Luo, Ying Wen, Oliver Slumbers, Daniel Graves, Haitham Bou-Ammar, Jun Wang, and Matthew E. Taylor. Diverse auto-curriculum is critical for successful real-world multiagent learning systems. In Frank Dignum, Alessio Lomuscio, Ulle Endriss, and Ann Nowé (eds.), *AAMAS '21: 20th International Conference on Autonomous Agents and Multiagent Systems, Virtual Event, United Kingdom, May 3-7, 2021*, pp. 51–56. ACM, 2021.

Martin Zinkevich, Michael Johanson, Michael Bowling, and Carmelo Piccione. Regret minimization in games with incomplete information. In J. Platt, D. Koller, Y. Singer, and S. Roweis (eds.), *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007. URL `https://proceedings.neurips.cc/paper/2007/file/08d98638c6fcd194a4b1e6992063e944-Paper.pdf`.

A THEORETICAL RESULTS

It is useful to understand the long-term behaviour of MAESTRO if each student and teacher agent reaches optimality. In this section, we will formally characterise this equilibrium behaviour, showing MAESTRO achieves a Bayes-Nash equilibrium in a modified game, which we call the $-i$-knowing game, in which the other player has prior knowledge of the environment and can design their policy accordingly. We will do this by first defining the $-i$-knowing game, showing that the policies the MAESTRO student learns in equilibria represent Bayes-Nash equilibrium strategies in this game, and then specialising this result to a corollary focused on fully observable games where MAESTRO finds a Nash Equilibrium for all environments in support of the teacher distribution.

We will define the $-i$-knowing-game, given a UPOSG M, with a parameter distribution $\tilde{\theta}$ as POSG constructed as a modification of the original game where the distribution over parameters is determined by $\tilde{\theta}$, and the agents other than i know the true parameters of the world. This simulates the setting where the co-player is a specialist for the particular environment, who has played the game many times. More formally:

Definition 1. *The -i-knowing-game of an underspecified partially observable stochastic game* $M = (N, A = \times_{i \in N} A_i, O = \times_{i \in N} O_i, \Theta, R = \times_{i \in N} R_i, S, I = \times_{i \in N} I_i, T)$ *with parameter distribution* $\tilde{\theta}$ *is defined to be the partially observable stochastic game* $K = (N' = N, A'_i = A_i, O'_i = O_i + \{\Theta \text{ if } i \in -i\}, R'_i = R_i, S' = S, I'_i = I_i + \{\theta \text{ if } \in -i\}, T' = T(\theta))$ *where* θ *is sampled from the distribution* $\tilde{\theta}$ *on the first time step. That is, a POSG with the same set of players, action space, rewards, and states as the original UPOSG, but with* θ *sampled once at the beginning of time, fixed into the transition function and given to the agents* $-i$ *as part of their observation.*

We use $U^K(\pi_i; \pi^K_{-i}; \tilde{\theta})$ to refer to the utility function in the -i-knowing game, which can be written in terms of the utility function of the original UPOSG as

$$U^K(\pi_i; \pi^K_{-i}; \tilde{\theta}) = \mathop{\mathbb{E}}_{\theta \sim \tilde{\theta}}[U(\pi_i; \pi^K_{-i}(\theta), \theta)],$$

where $\pi^K_{-i}(\theta)$ is the policy for players $-i$ in the -i-knowing game conditioned on θ. Given this definition, we can prove the main theorem, that the equilibrium behaviour of MAESTRO represents a Bayes-Nash Equilibrium of this game.

Theorem 1. *In two-player zero-sum settings, the* MAESTRO *student at equilibrium implements a Bayes-Nash Equilibrium of the* $-i$*-knowing game, over a regret-maximising distribution of levels.*

Proof. Let $\pi_i, \tilde{\theta}^M$ be a pair which is in equilibrium in the MAESTRO game. That is:

$$\pi_i \in \arg\max_{\pi_i \in \Pi_i} \{ \mathop{\mathbb{E}}_{\theta, \pi_{-i} \sim \tilde{\theta}^M} [U(\pi_i; \pi_{-i}; \theta)] \} \tag{3}$$

$$\tilde{\theta}^M \in \arg\max_{\tilde{\theta}^M \in \Delta(\Theta \times \Pi_{-i})} \{ \mathop{\mathbb{E}}_{\theta, \pi_{-i} \sim \tilde{\theta}^M} [U(\pi^*_i; \pi_{-i}; \theta) - U(\pi_i; \pi_{-i}; \theta)] \} \tag{4}$$

where π^*_i is an optimal policy for player i given π_{-i} and θ, while $\Delta(S)$ denotes the set of distributions over S. Then we can define D^{Regret} to be the marginal distribution over θ from samples $\theta, \pi_{-i} \sim \tilde{\theta}^M$. Define $\pi^K_{-i}(\theta)$ as the marginal distribution over π_{-i} sampled from $\tilde{\theta}^M$ conditioned on θ for θ in the support of $\tilde{\theta}^M$ and π_{-i} a best response to θ and π_i otherwise.

We will show that $(\pi_i; \pi^K_{-i})$ is a Bayes-Nash equilibrium on -i-knowing game of M with a regret-maximizing distribution over parameters D^{Regret}. We can show both of these by unwrapping and re-wrapping our definitions.

First to show $\pi_i \in \arg\max_{\pi_i \in \Pi_i} \{ \mathop{\mathbb{E}}_{\tilde{\theta} \sim D^{Regret}} [U^K(\pi_i; \pi^K_{-i}; \tilde{\theta})] \}$:

$$\pi_i \in \arg\max_{\pi_i \in \Pi_i} \{ \mathop{\mathbb{E}}_{\tilde{\theta} \sim D^{Regret}} [U^K(\pi_i; \pi^K_{-i}; \tilde{\theta})] \} \tag{5}$$

$$\iff \pi_i \in \arg\max_{\pi_i \in \Pi_i} \{ \mathop{\mathbb{E}}_{\theta, \pi_{-i} \sim \tilde{\theta}^M} [U(\pi_i; \pi^K_{-i}(\theta); \theta)] \} \tag{6}$$

$$\iff \pi_i \in \arg\max_{\pi_i \in \Pi_i} \{ \mathop{\mathbb{E}}_{\theta, \pi_{-i} \sim \tilde{\theta}^M} [U(\pi_i; \pi_{-i}; \theta)] \} \tag{7}$$

Which is known by the definition of Nash-Equilibrium in the MAESTRO game, in Equation 3.

Similarly, we show that we have $\pi^K_{-i} \in \arg\max_{\pi^K_{-i} \in \Pi^K_{-i}} \{ \mathbb{E}_{\tilde{\theta} \sim D^{Regret}} [U^K(\pi_i; \pi^K_{-i}; \tilde{\theta})] \}$ by:

$$\pi^K_{-i} \in \arg\max_{\pi^K_{-i} \in \Pi^K_{-i}} \{ \mathbb{E}_{\tilde{\theta} \sim D^{Regret}} [U^K(\pi_i; \pi^K_{-i}; \tilde{\theta})] \} \tag{8}$$

$$\iff \pi^K_{-i} \in \arg\max_{\pi^K_{-i} \in \Pi^K_{-i}} \{ \mathbb{E}_{\theta, \pi_{-i} \sim \tilde{\theta}^M} [U(\pi_i; \pi^K_{-i}(\theta); \theta)] \} \tag{9}$$

$$\iff \pi^K_{-i}(\theta) \in \arg\max_{\pi^K_{-i} \in \Pi^K_{-i}} \{ \mathbb{E}_{\theta, \pi_{-i} \sim \tilde{\theta}^M} [U(\pi_i; \pi_{-i}; \theta)] \}. \tag{10}$$

The final line of which follows from the fact that π_{-i} is a best-response π_i for each θ. More concretely, this can be seen in Equation 4 by noting that θ and π_{-i} conditioned on a specific θ can be independently optimised and holding θ fixed. □

Using this theorem, we can also prove a natural and intuitive corollary for the case where the environment is fully observable:

Corollary 1. *In fully-observable two-player zero-sum settings, the* MAESTRO *student at equilibrium implements a Nash Equilibrium in each environment in the support of the environment distribution.*

Proof. From Theorem 1 we have:

$$\pi_i \in \arg\max_{\pi_i \in \Pi_i} \{ \mathbb{E}_{\tilde{\theta} \sim D^{Regret}} [U^K(\pi_i; \pi^K_{-i}; \tilde{\theta})] \}$$

However, since the world is fully observable, the policy π^K_{-i} can be made independent of the additional observation θ in the $-i$-knowing game since that can be inferred from the information already in the agent's observations. As such, π^K_{-i} can be interpreted as a policy in the original game, giving:

$$\pi_i \in \arg\max_{\pi_i \in \Pi_i} \{ \mathbb{E}_{\theta \sim D^{Regret}} U(\pi_i; \pi^K_{-i}(\theta); \theta) \}$$

Moreover, since the environment is fully observable, π_i can condition on θ, so for it to be optimal for the distribution, it must be optimal for each level in the support of the distribution. Giving, for each θ be in the support of $\tilde{\theta}^M$:

$$\pi_i \in \arg\max_{\pi_i \in \Pi_i} \{ U(\pi_i; \pi^K_{-i}(\theta); \theta) \}$$

Showing that π is a best-response to $\pi^K_{-i}(\theta)$ on θ. The same arguments can be followed to show that $\pi^K_{-i}(\theta)$ is a best response to π on θ. Since each policy is a best response to the other, they are in a Nash Equilibrium as desired. □

Thus, if MAESTRO reaches an equilibrium in a fully observable two-player zero-sum setting it behaves as expected by achieving a Nash Equilibrium in every environment in support of the curriculum distribution $\tilde{\theta}^M$.

B ABLATION STUDY

We perform an ablation study to evaluate the effectiveness of co-player selection in MAESTRO from a population based on their per-environment regret scores, as described in Equation (1). We consider different methods for selecting a co-player in MAESTRO (line 6 in Algorithm 1). MAESTRO-R samples the co-player uniformly at random, whilst MAESTRO-P uses PFSP's win rate heuristic for opponent prioritisation. Figure 8 illustrates that MAESTRO outperforms both variants in terms of sample-efficiency and robustness.

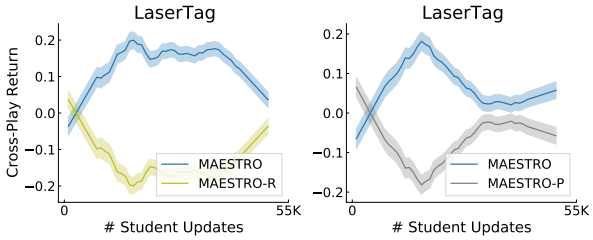


Figure 8: **Ablation Study Results**. Comparing MAESTRO against two variants: MAESTRO-R and MAESTRO-P. Plots show the mean and standard error across 5 training seeds.

C Environment Details

This section describes the environment-specific details used in our experiments. For both LaserTag and MultiCarRacing, we outline the process of environment generation, present held-out evaluation environments, as well as other relevant information.

C.1 LaserTag

LaserTag is a two-player zero-sum grid-based game, where two agents aim to tag each other with a light beam under partial observability. It is inspired by prior singleton variation used in (Lanctot et al., 2017; Leibo et al., 2017) and developed using the Griddly sandbox framework (Bamford et al., 2020; 2022). LaserTag challenges the agent to master various sophisticated behaviour, such as chasing opponents, hiding behind walls, keeping clear of long corridors, maze solving, etc. Each agent can only observe a 5×5 area of the grid in front of it. The action space includes the following 5 actions: turn right, turn left, move forward, shoot, and no-op. Upon tagging an opponent, an agent receives a reward of 1, while the opponent receives a -1 penalty, after which the episode is restarted. If the episode terminates while the two agents are alive, neither of the agents receives any reward.

All environment variations (or levels) that are used to train agents are procedurally generated by an environment generator. Firstly, the generator samples the size of the grid (from 5×5 to 15×15) and the percentage of the walls it is in (from 0% to 50%) uniformly at random. Then the generator samples random locations for the walls, followed by the locations and directions of the two agents. Figure 2 illustrates some levels sampled from the level generator. Note that the generator can generate levels where agents are unreachable.

Upon training the agents on randomly generated levels, we assess their robustness on previously unseen human-designed levels shown in Figure 9 against previously unseen agents. For this environment, we recognise the agent with a higher episodic return as the winner of that episode.

C.2 MultiCarRacing

MultiCarRacing is a continuous control problem with dense rewards and pixel-based observations (Schwarting et al., 2021). Each track consists of n tiles, with cars receiving a reward of $1000/n$ or $500/n$, depending on if they reach the tile first or second respectively. An additional penalty of -0.1 is applied at every timestep. Episodes finish when all tiles have been driven over by at least one car. If a car drives out of bounds of the map (rectangle area encompassing the track), the car "dies" and the episode is terminated. Each agent receives a $96 \times 96 \times 3$ image as observation at each timestep. The action space consists of 3 simultaneous moves that change the gas, brakes, and steering direction of the car.

All tracks used to train student agents are procedurally generated by an environment generator, which was built on top of the original MultiCarRacing environment (Schwarting et al., 2021). Each track consists of a closed loop around which the agents must drive a full lap. In order to increase the expressiveness of the original MultiCarRacing, we reparameterized the tracks using Bézier curves. In our experiments, each track consists of a Bézier curve based on 12 randomly sampled control points within a fixed radius of $B/2$ of the centre O of the playfield with $B \times B$ size.

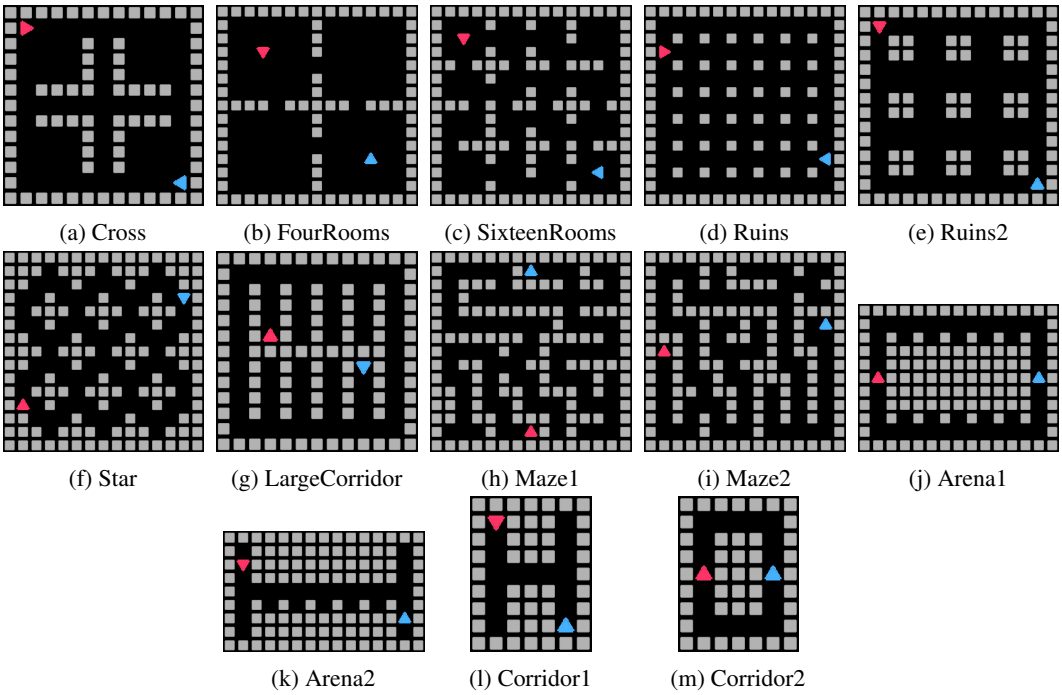


Figure 9: Evaluation environments for LaserTag.

For training, additional reward shaping was introduced similar to (Ma, 2019): an additional reward penalty of -0.1 for driving on the grass, a penalty of -0.5 for driving backwards, as well as an early termination if cars spent too much time on grass. These are all used to help terminate less informative episodes. We utilise a memory-less agent with a frame stacking $= 4$ and with sticky actions $= 8$. After training the agents on randomly generated tracks, we assess their robustness on previously unseen 20 real-world Formula One (F1) tracks designed to challenge professional racecar drivers proposed by (Jiang et al., 2021a) and shown in Figure 10.

Given the poor performance of random opponents on the MultiCarRacing domain, agents are added to co-player populations in MAESTRO as well as FSP- and PFSP-based baselines only after 400 PPO updates. All baselines are trained using SP until that point.

D IMPLEMENTATION DETAILS

In this section, we detail the agent architectures, hyperparameter choices, and evaluation procedures used in our experiments discussed in Section 4. We use PPO to train the student agent in all experiments. Table 2 summarises our final hyperparameter choices for all methods.

All experiments are performed on an internal cluster. Each job (representing a seed) is performed with a single Tesla V100 GPU and 10 CPUs. For each method, we train 10 LaserTag agents for approximately 7 days and 5 MultiCarRacing agents for approximately 15 days.

D.1 LASERTAG

Agent Architecture: The student policy architecture is adapted from (Dennis et al., 2020; Jiang et al., 2021a). Our model encodes the partial grid observation using a convolution layer (3×3 kernel, stride length 1, 16 filters) followed by a ReLU activation layer over the flattened convolution outputs. This is then passed through an LSTM with hidden dimension 256, followed by two fully-connected layers, each with a hidden dimension of size 32 with ReLU activations, to produce the action logits over the 5 possible actions. The model does not receive the agent's direction as input.

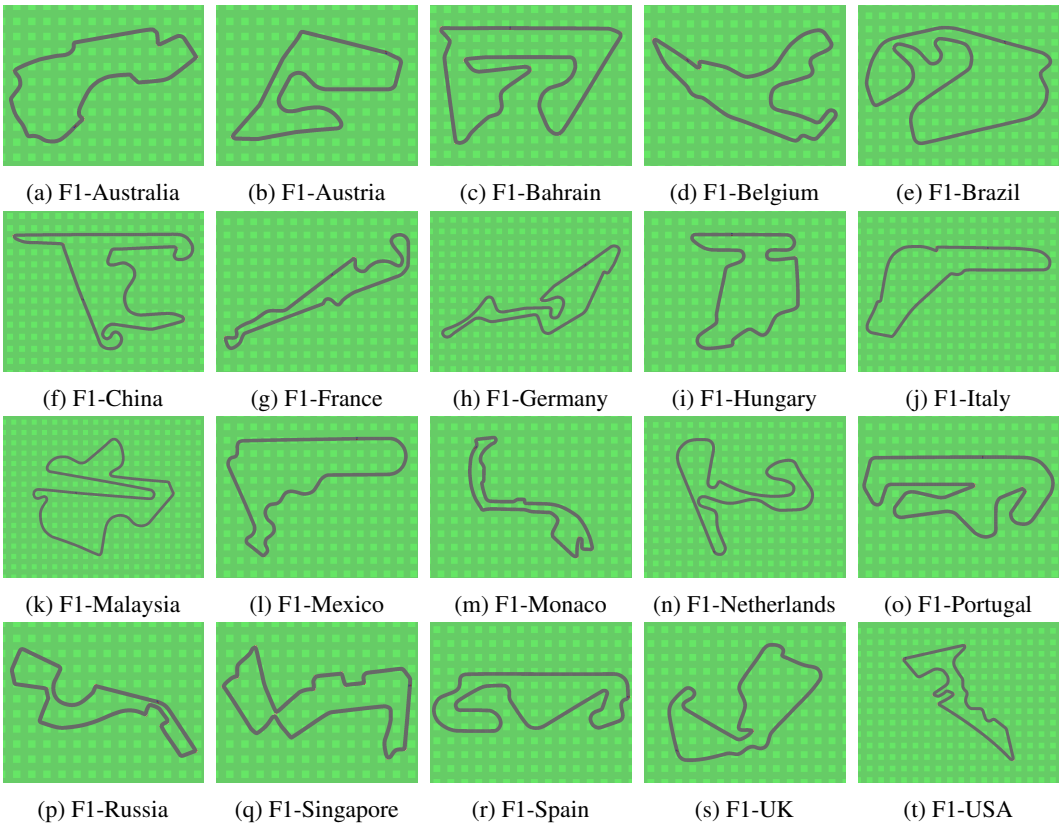


Figure 10: Evaluation Formula 1 tracks for MultiCarRacing.

Evaluation Procedure: For each pair of methods in LaserTag, we evaluate cross-play performance across all 5 training seeds of each method (25 possible pair combinations) over 4 episodes on 11 test environments, resulting in a total of 1100 evaluation episodes per method pair.

Choice of Hyperparameters: Many of our hyperparameters are inherited from previous works such as (Dennis et al., 2020; Jiang et al., 2021b;a; Parker-Holder et al., 2022) with some small changes. We selected the best performing settings based on the average return on the unseen validation levels against previously unseen opponents on at least 5 seeds.

We conducted a coarse grid search over student learning rate in $\{5 * 10^{-4}, 10^{-4}, 5 * 10^{-5}, 10^{-5}\}$, number of minibatches per epoch in $\{1, 2, 4\}$, entropy coefficients in $\{0, 10^{-3}, 5*10^{-3}\}$, and number of epochs in $\{2, 5, 10\}$. For agent storage, we tested adding a copy of the student agent in the storage after every $\{2000, 4000, 6000, 8000\}$ student update. For PFSP, we compute the win rate between agents in the last 128 episodes. We further conducted a grid search over the entropy parameter of f_{hard} in $\{1.5, 2, 3\}$ and a smoothing constant which adds a small value to each probability in PFSP with $\{0.1, 0.2\}$ values so that, at all previous checkpointed agents have a nonzero probability to be replayed again.[3] For the parameters of PLR, we conducted a grid search over level replay rate p in $\{0.5, 0.9\}$, buffer size in $\{4000, 8000, 12000\}$, staleness coefficient ρ in $\{0.3, 0.7\}$, as well as the level replay score functions in {MaxMC, PVL} (see Appendix D.4 for information on score functions in PLR). For MAESTRO, we evaluated the co-player exploration coefficients in $\{0.05, 0.1\}$, and per-agent environment buffer sizes in $\{500, 1000\}$.

D.2 MULTICARRACING

Agent Architecture: The student policy architecture is based on the PPO implementation in (Ma, 2019). The model utilises an image embedding module consisting of a stack of 2D convolutions with square kernels of sizes $2, 2, 2, 2, 3, 3$, channel outputs of $8, 16, 32, 64, 128, 256$, and stride lengths of $2, 2, 2, 2, 1, 1$ respectively, resulting in an embedding of size 256. This is then passed through a fully-connected layer with a hidden size of 100, followed by a ReLU nonlinearity. Then, the output is fed through two separate fully-connected layers, each with a hidden size of 100 and an output dimension equal to the action dimension, followed by softplus activations. We then add 1 to each component of these two output vectors, which serve as the α and β parameters respectively for the Beta distributions used to sample each action dimension. We normalize rewards by dividing rewards by the running standard deviation of returns so far encountered during the training.

Evaluation Procedure: For each pair of methods in MultiCarRacing, we evaluate cross-play performance between 5 training seeds of each method (random assignments between 5 pairs) over 2 episodes on 5 test environments, resulting in a total of 50 evaluation episodes per method pair.

Choice of Hyperparameters: Many of our hyperparameters are inherited from (Jiang et al., 2021a) with some small changes. We conducted a limited grid search over student learning rate in $\{10^{-4}, 3 * 10^{-4}\}$, number of actors in $\{16, 32\}$, PPO rollout length in $\{125, 256\}$. For agent storage, we tested adding a copy of the student agent in the storage after every $\{200, 400\}$ student update. For PFSP, we compute the win rate between agents in the last 128 episodes, while recognising the agent with a higher episodic return as the winner. For the parameters of PLR, we conducted a grid search over level buffer size in $\{4000, 6000, 8000\}$, staleness coefficient ρ in $\{0.3, 0.7\}$, as well as the level replay prioritisation in {rank, proportional} (Jiang et al., 2021a). For MAESTRO, we evaluated the co-player exploration coefficients in $\{0.05, 0.1\}$, and per-agent environment buffer sizes in $\{500, 1000\}$.

D.3 ALL HYPERPARAMETERS

Table 2 summarises our final hyperparameter choices for all methods.

D.4 REGRET APPROXIMATIONS

Jiang et al. (2021a) proposes the following two score functions to approximate regret in PLR.

[3] Otherwise, if the student agent wins all the episodes in their first encounter against opponent B, B will have 0 probability of being selected again.

Table 2: Hyperparameters used for training each method in the LaserTag and MultiCarRacing environments.

Parameter	LaserTag	MultiCarRacing
PPO		
γ	0.995	0.99
λ_{GAE}	0.95	0.9
PPO rollout length	256	125
PPO epochs	5	8
PPO mini-batches per epoch	4	4
PPO clip range	0.2	0.2
PPO number of workers	32	32
Adam learning rate	1e-4	1e-4
Adam ϵ	1e-5	1e-5
PPO max gradient norm	0.5	0.5
PPO value clipping	yes	no
Return normalization	no	yes
Value loss coefficient	0.5	0.5
Student entropy coefficient	0.0	0.0
PLR		
Replay rate, p	0.5	0.5
Buffer size, K	4000	8000
Scoring function	MaxMC	PVL
Prioritization	rank	rank
Temperature, β	0.3	1.0
Staleness coefficient, ρ	0.3	0.7
FSP		
Agent checkpoint interval	8000	400
PFSP		
f_{hard} entropy coef	2	2
Win rate episodic memory	128	128
MAESTRO		
λ coef	0.1	0.1
Buffer size for $\mathfrak{B}$ members	1000	1000

Positive Value Loss (PVL) estimates the regret by computing the difference between maximum achieved return and predicted return on an episodic basis. When GAE (Schulman et al., 2016) is used to estimate bootstrapped value targets, this loss takes the form of $\frac{1}{T}\sum_{t=0}^{T}\max\left(\sum_{k=t}^{T}(\gamma\lambda)^{k-t}\delta_k, 0\right)$, where λ and γ are the GAE and MDP discount factors respectively, and δ_t, the TD-error at timestep t.

Maximum Monte Carlo (MaxMC) mitigates some of the bias of the PVL by replacing the value target with the highest empirical return observed on the given environment variation throughout training. MaxMC ensures that the regret estimate does not depend on the agent's current policy. It takes the form of $(1/T)\sum_{t=0}^{T} R_{\max} - V(s_t)$.

E FULL RESULTS

E.1 MAESTRO VERSUS SPECIALISTS

Figures 11 and 12 show the cross-play performances between MAESTRO and specialist agents trained directly on the target environments for LaserTag and MultiCarRacing, respectively.

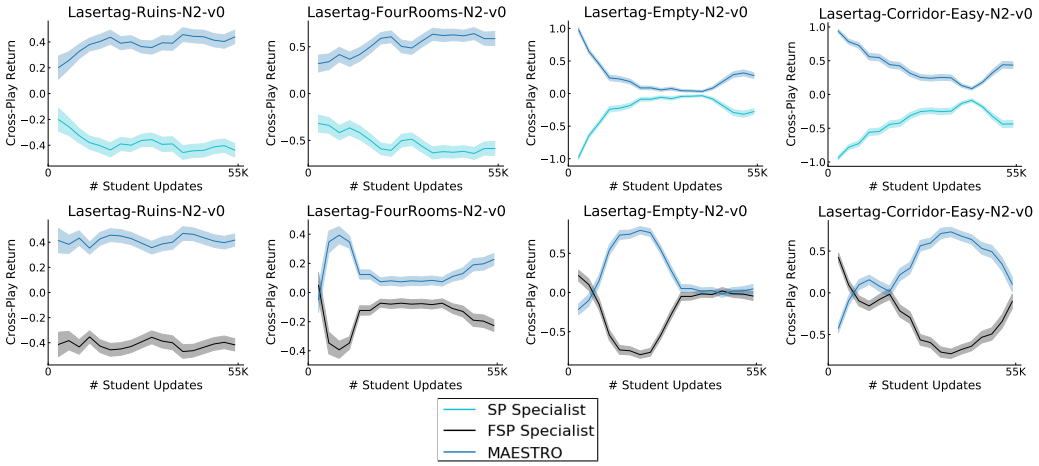


Figure 11: Cross-play between MAESTRO and specialist agents trained directly on the target environment in LaserTag.

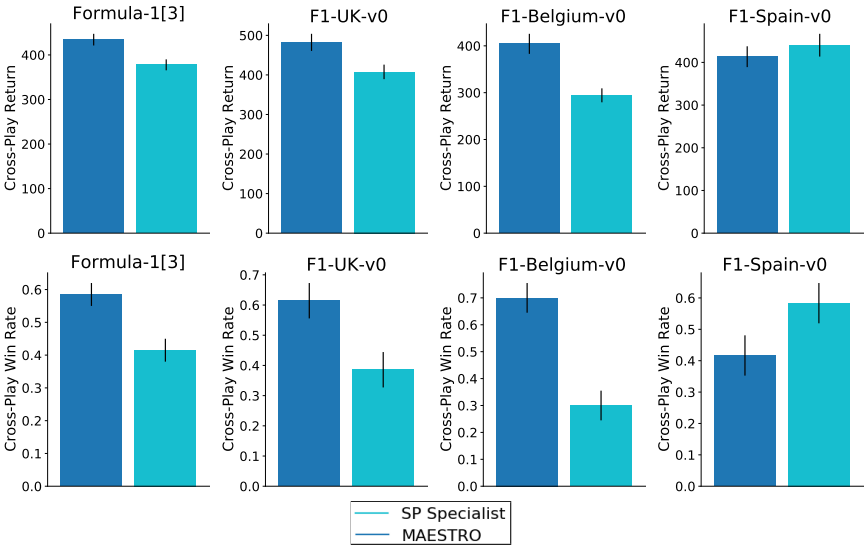


Figure 12: Cross-play between MAESTRO and specialist agents trained directly on the target environment in MultiCarRacing.

E.2 CROSS-PLAY RESULTS

E.2.1 LASERTAG CROSS-PLAY

Figure 13 shows the mean return of MAESTRO against each of the baselines throughout training. Figures 14 and 15 illustrate the minimum expected returns and round-robins returns between MAESTRO and other baselines throughout training on each held-out evaluation environment in LaserTag. 16 and 17 show the minimum expected return and round-robin returns between MAESTRO and other baselines after the training.

E.2.2 MULTICARRACING CROSS-PLAY

Figures 18 and 19 illustrate the min expected returns and round-robin returns between MAESTRO and other baselines on each track of the Formula 1 benchmark (Jiang et al., 2021a). Figures 20 and 21 show the win rates and returns in a round robin tournament between MAESTRO after the training on Formula 1 benchmark.

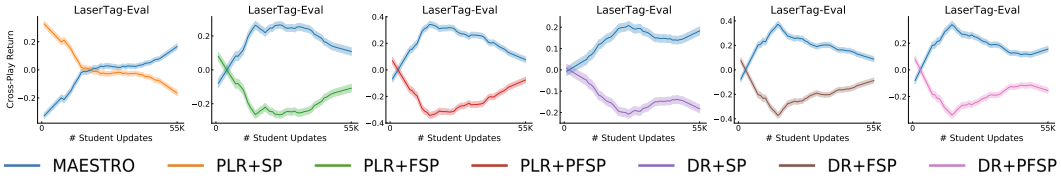


Figure 13: Mean return of MAESTRO versus each of the baselines throughout training on LaserTag evaluation environments. Plots show the mean and standard error across 10 training seeds.

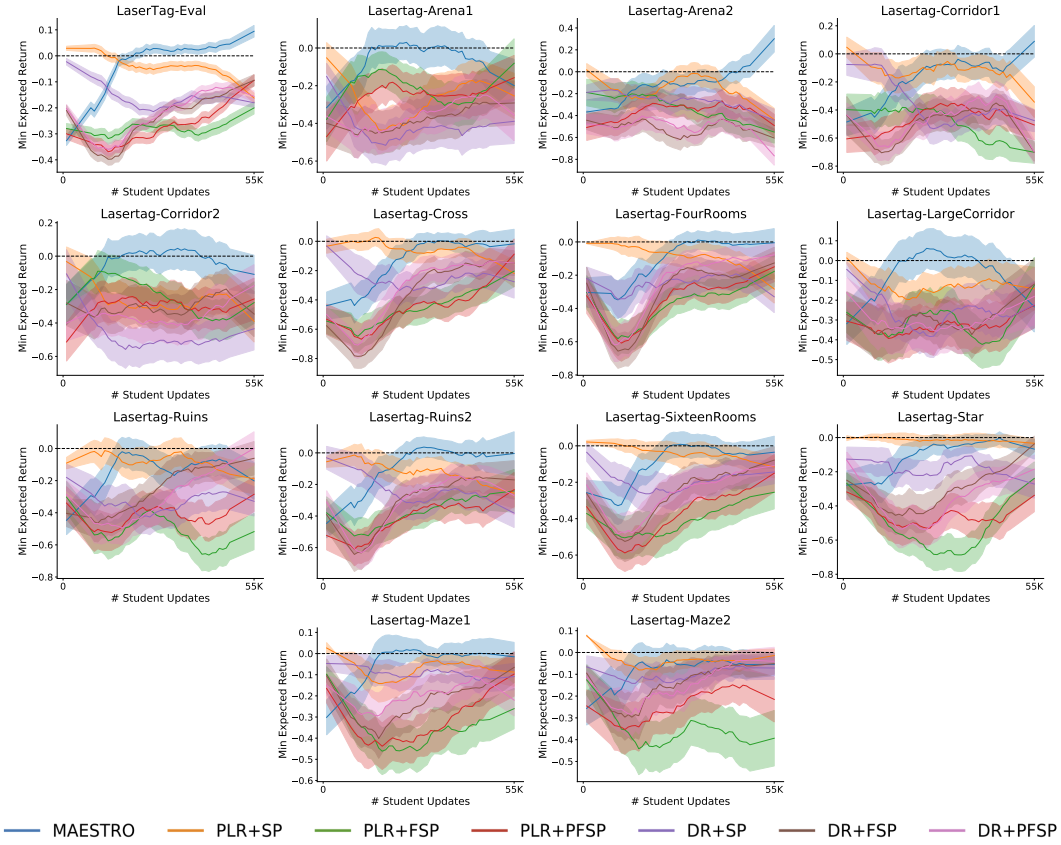


Figure 14: Minimum expected return in cross-play between MAESTRO and 6 baselines on all LaserTag evaluation environments throughout training (combined and individual). Plots show the mean and standard error across 10 training seeds.

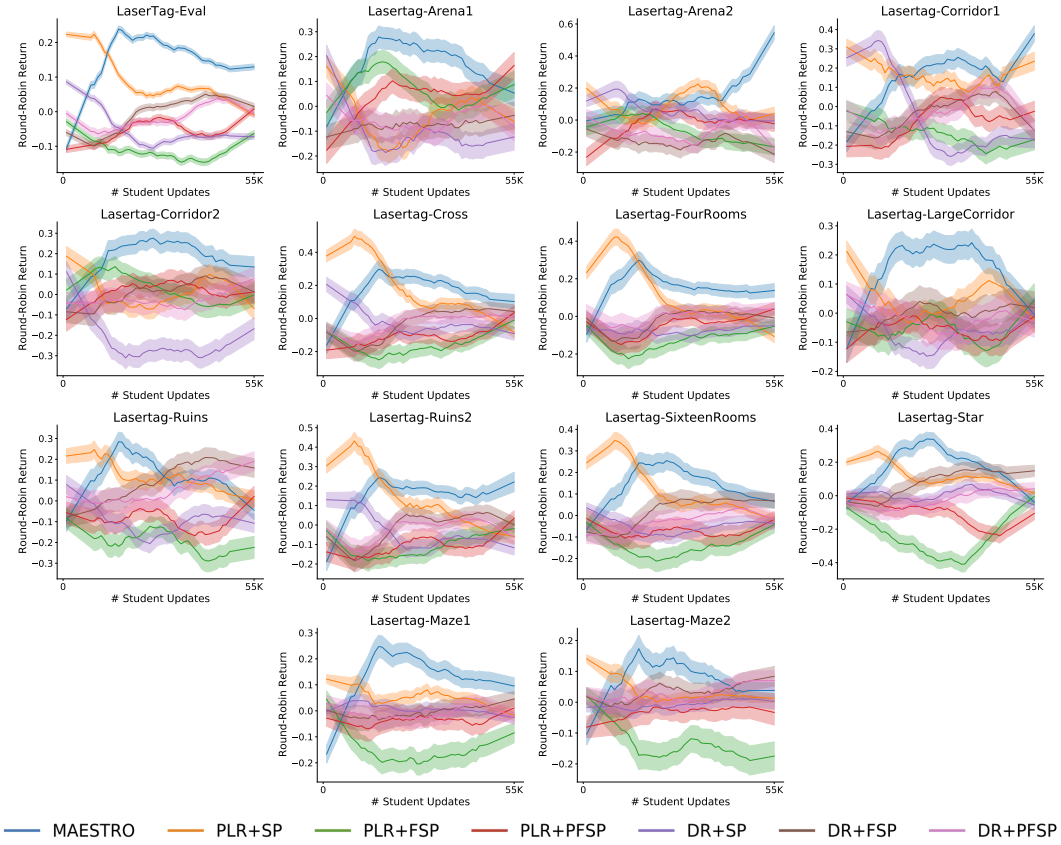


Figure 15: Round-robin return in cross-play between MAESTRO and 6 baselines on all LaserTag evaluation environments throughout training (combined and individual). Plots show the mean and standard error across 10 training seeds.

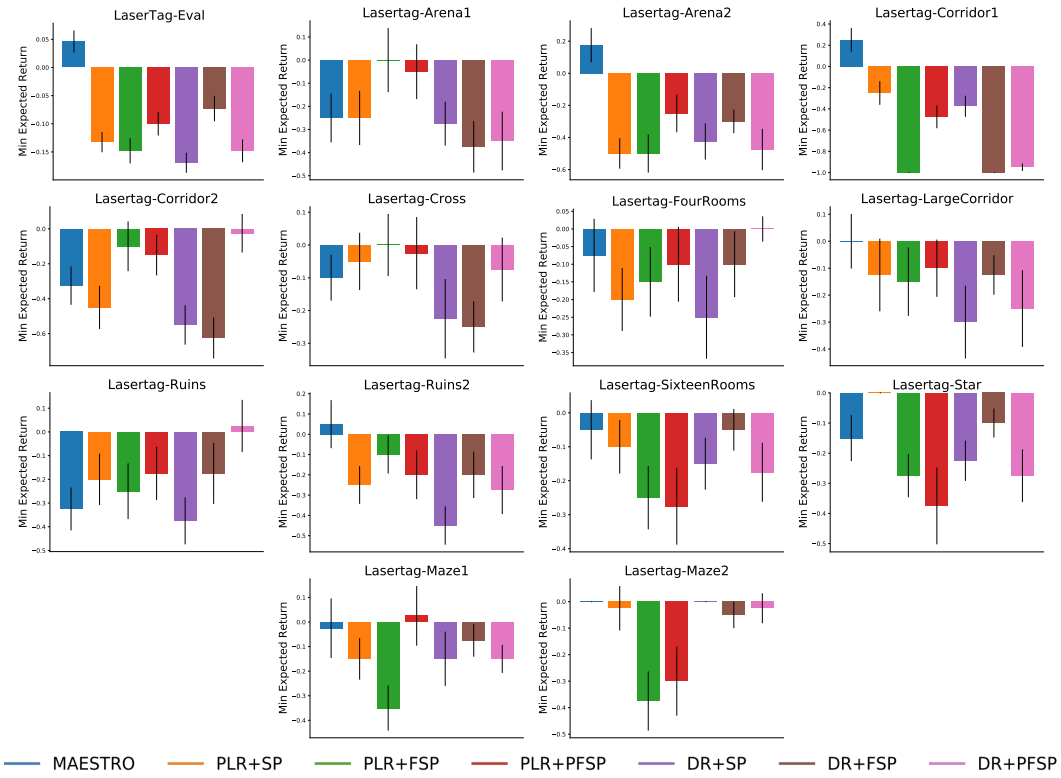


Figure 16: Minimum expected returns in cross-play between MAESTRO and 6 baselines on all LaserTag evaluation environments after the training (combined and individual). Plots show the mean and standard error across 10 training seeds.

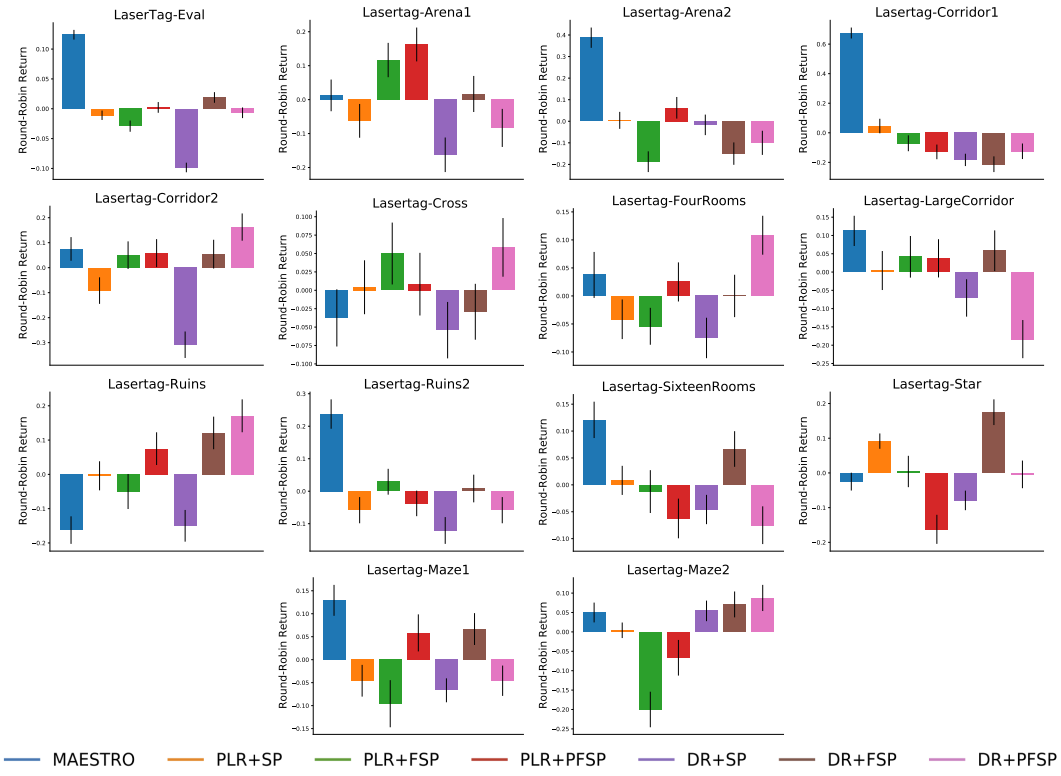


Figure 17: Returns in a round-robin tournament between MAESTRO and 6 baselines on all LaserTag evaluation environments (combined and individual). Plots show the mean and standard error across 10 training seeds.

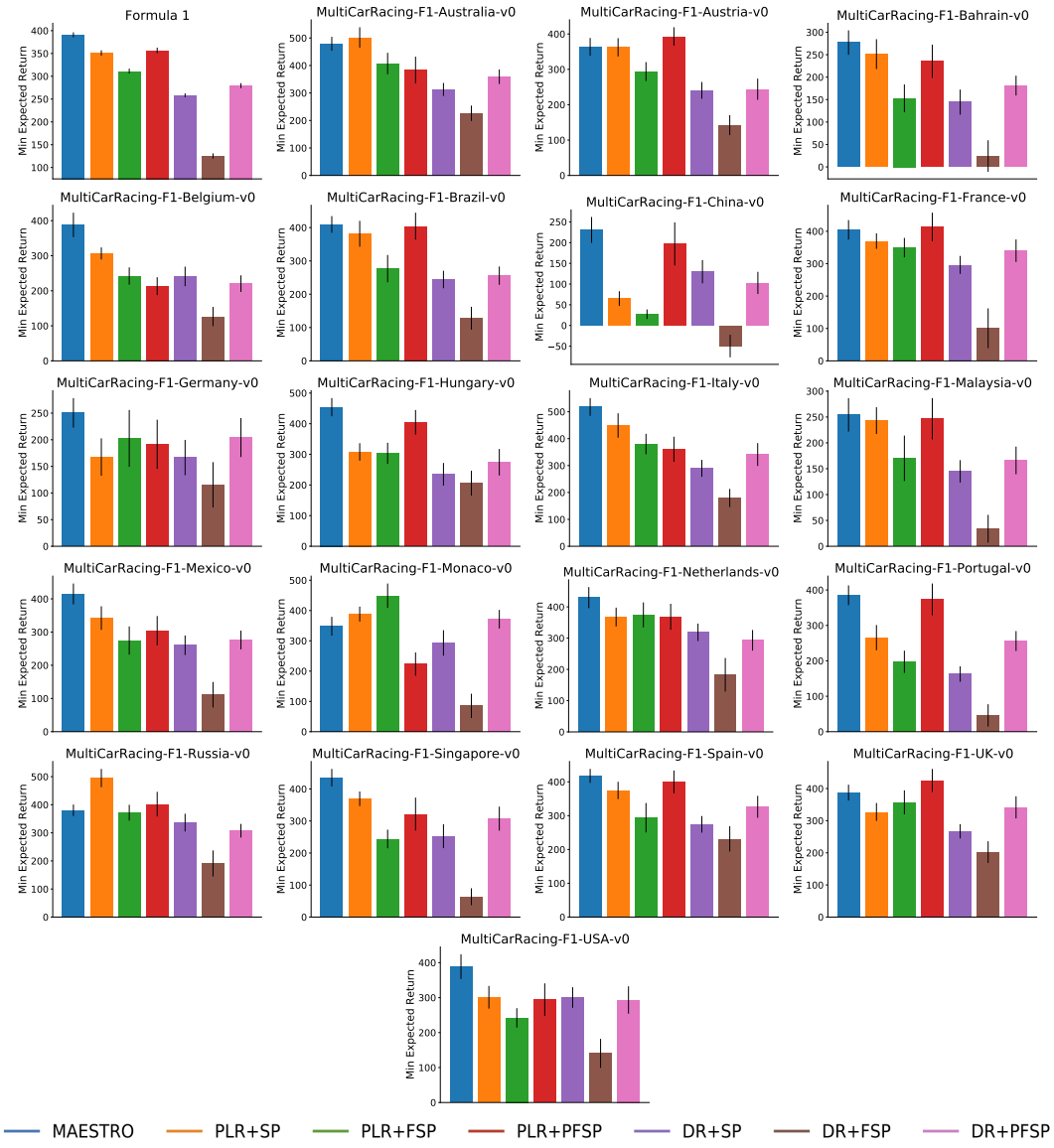


Figure 18: Minimum expected returns between MAESTRO and 6 baselines on all Formula 1 tracks (combined and individual). Plots show the mean and standard error across 5 training seeds.

Figure 19: Round-robin returns between MAESTRO and 6 baselines on all Formula 1 tracks (combined and individual). Plots show the mean and standard error across 5 training seeds.

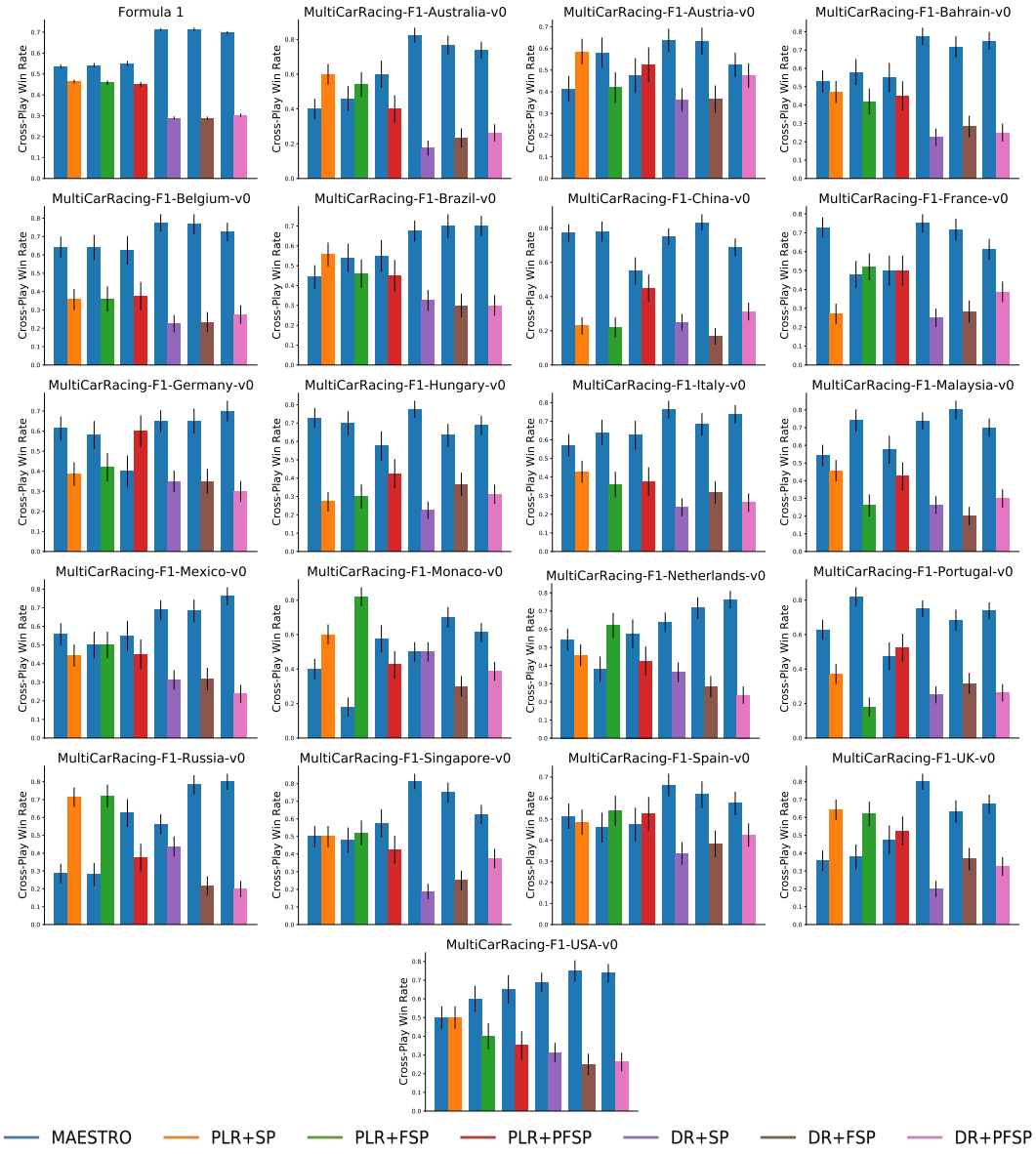


Figure 20: Win rates in cross-play between MAESTRO vs each of the 6 baselines on all Formula 1 tracks (combined and individual). Plots show the mean and standard error across 5 training seeds.

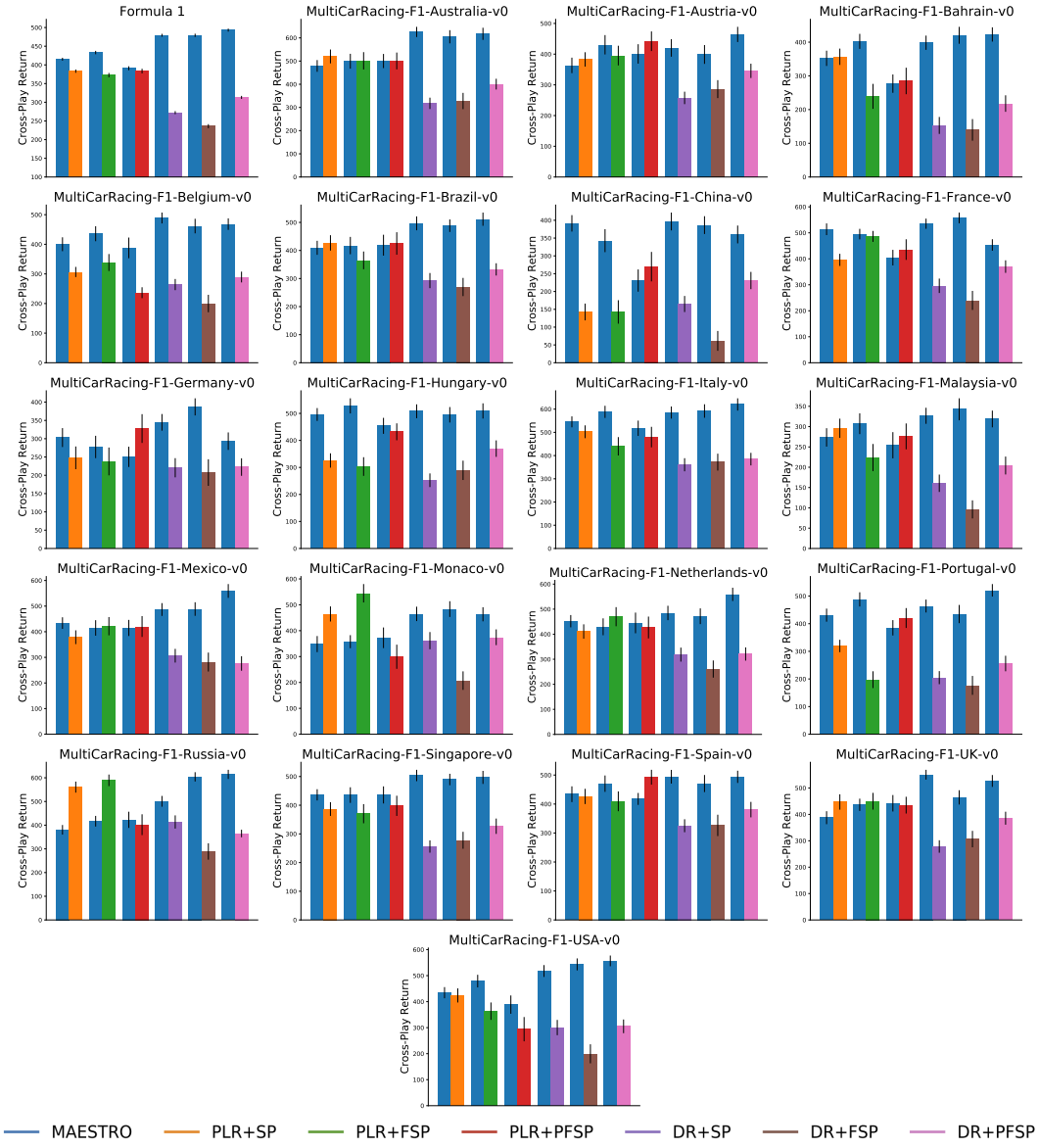


Figure 21: Returns in cross-play between MAESTRO vs each of the 6 baselines on all Formula 1 tracks (combined and individual). Plots show the mean and standard error across 5 training seeds.