Multi-Agent Diagnostics for Robustness via Illuminated Diversity

Mikayel Samvelyan[12*] **Davide Paglieri**[1*] **Minqi Jiang**[12] **Jack Parker-Holder**[1] **Tim Rocktäschel**[1]
[1]University College London [2]Meta AI

Abstract

In the rapidly advancing field of multi-agent systems, ensuring robustness in unfamiliar and adversarial settings is crucial, particularly for those systems deployed in real-world scenarios. Notwithstanding their outstanding performance in familiar environments, these systems often falter in new situations due to overfitting during the training phase. This is especially pronounced in settings where both cooperative and competitive behaviours are present, encapsulating a dual nature of overfitting and generalisation challenges. To address this issue, we present *Multi-Agent Diagnostics for Robustness via Illuminated Diversity* (MADRID), a novel approach for systematically generating diverse adversarial scenarios that expose strategic vulnerabilities in pre-trained multi-agent policies. Leveraging the concepts from open-ended learning, MADRID navigates the vast space of adversarial settings, employing a target policy's regret to gauge the vulnerabilities of these settings. We evaluate the effectiveness of MADRID on the 11 vs 11 version of Google Research Football, one of the most complex environments for multi-agent reinforcement learning. Specifically, we employ MADRID for generating a diverse array of adversarial settings for TiZero, the state-of-the-art approach which "masters" the game through 45 days of training on a large-scale distributed infrastructure. Using MADRID, we expose key shortcomings in TiZero's tactical decision-making, underlining the crucial importance of rigorous evaluation in multi-agent systems.[2]

1 Introduction

In recent times, multi-agent systems, particularly those designed to interact with humans, have emerged as a primary model for AI deployment in real-world scenarios [31, 1, 2, 45]. Although there have been significant successes in simulated environments, as evidenced by deep reinforcement learning (RL) in complex multi-agent games [41, 38, 47, 4, 52], the transfer from simulation to reality (sim2real) continues to pose challenges [19, 54]. Specifically, while these models demonstrate proficiency in known environments, they become highly susceptible to faulty behaviors in unfamiliar settings and adversarial situations [36]. Given their critical roles in human-centric applications, understanding and mitigating these susceptibilities becomes paramount for fostering more effective, reliable, and transparent deployment of multi-agent AI systems in the future.

The Achilles' heel of these multi-agent systems, contributing to their lack of robustness, is often their overfitting to the specific settings encountered during training [24]. This overfitting becomes notably evident in two-team zero-sum settings where both cooperative and competitive dynamics intertwine. A primary manifestation of the overfitting between cooperative agents, especially when all agents in the group share the same set of network parameters (i.e., parameter sharing [14]), is in the agents becoming too accustomed to their training environments, leading to a detailed coordination

*Equal contribution. Correspondance to samvelyan@meta.com and d.paglieri@cs.ucl.ac.uk.
[2]Visuals are available at https://sites.google.com/view/madrid-marl.

2nd Workshop on Agent Learning in Open-Endedness (ALOE) at NeurIPS 2023.

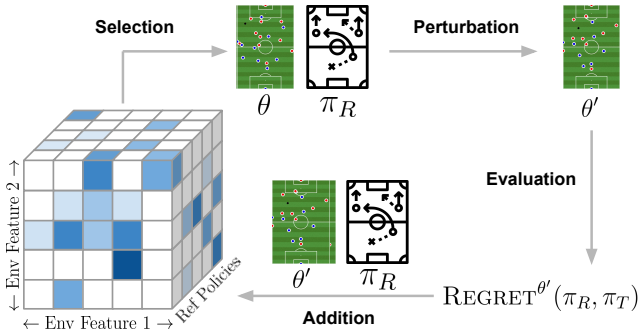


Figure 1: **Overview of MADRID**. Operating on a discretised grid with an added dimension for reference policies, MADRID archives environment variations (or levels) characterized by representative features, e.g., (x, y) coordinates of the ball position in football. During each iteration, MADRID mutates a selected level, computes regret using its associated reference policy, and reincorporates levels with higher regret into the archive, effectively generating diverse collection adversarial levels.

tailored to these specific conditions. As a consequence, when introduced to unfamiliar settings, their performance tends to falter. Concurrently, there is also an overfitting to specific opponent teams they have trained against. Instead of developing a flexible strategy that can withstand a variety of opponents, their strategies might be overly optimised to counteract the strategies of familiar adversaries. These dual forms of overfitting—both to the environment and to opponents—render such settings as perfect platforms to probe for vulnerabilities [46]. Furthermore, it is crucial to pinpoint a diverse set of adversarial scenarios for a holistic diagnostic of robustness, shedding light on possible shortcomings from various perspectives.

Given these challenges, we introduce ***M**ulti-**A**gent **D**iagnostics for **R**obustness via **I**lluminated **D**iversity* (MADRID), a novel method for systematically generating a diverse collection of adversarial settings where pre-trained multi-agent policies make strategic mistakes. To this end, MADRID employs approaches from quality-diversity [25, 8], a family of evolutionary algorithm that aim to generate a large collection of high-performing solutions each with their own unique characteristics.

MADRID incorporates MAP-Elites [30], a simple and effective QD approach, to systematically explore the vast space of adversarial settings. By discretising the search space, MADRID iteratively performs selection, mutation, and evaluations steps, endlessly refining and expanding the repertoire of high-performing adversarial scenarios within its archive (see Figure 1). A crucial feature of MADRID is its employment of the target policy's *regret*— the gap in performance between the optimal and target policy—to quantify the quality of adversarial settings. It has been previously shown that regret is an effective metric for identifying situations where RL agents underperform in both single-agent [11, 22, 33, 29] and multi-agent [36] domains. MADRID estimates a lower-bound on the true regret by utilising a collection of reference policies [47, 17], which are not necessarily required to be high-performing. MADRID identifies situations where these reference policies surpass the target one, thereby providing a clear illustration of superior performance in given situations.

To evaluate MADRID, we concentrate specifically on one of the most challenging multi-agent domains, namely the fully decentralised 11 vs 11 variation of Google Research Football [GRF, 23]. This simulated environment is based on the popular real-world sport of football (a.k.a. soccer) and requires two teams of agents to combine short-term control techniques with coordinated, long-term global strategies. GRF represents a unique combination of characteristics not present in other RL environments [27], namely multi-agent cooperation (within each team), competition (between the two teams), sparse rewards, large action and observation spaces, and stochastic dynamics. While many of the individual challenges in GRF, including multi-agent coordination [34, 53], long-term planning [12] and non-transitivity [3, 9], have been studied extensively in isolation, learning highly-competitive GRF policies has long remained outside the reach of RL methods. TiZero [27], a recent multi-agent RL approach, learned to "master" the fully decentralised variation of GRF from scratch for the first time, using a hand-crafted curriculum, reward shaping, and self-play. Experimentally, TiZero has shown impressive results and outperformed previous methods by a large margin after an expensive training lasting 45 days on a large-scale distributed training infrastructure.

We apply MADRID on GRF by targeting TiZero to diagnose a broad set of scenarios in which it commits tactical mistakes. Our extensive evaluations reveal diverse settings where TiZero exhibits a poor performance, where weaker policies can outperform it. Specifically, MADRID discovers

instances where TiZero is ineffective at near the opponent's goal, demonstrates a marked inability to comprehend the offside rule effectively, and even encounters situations of scoring accidental own goals. These findings highlight the latent vulnerabilities within even highly trained models and demonstrate that there is much room for improving the their robustness. Our analysis showcases the value of identifying such adversarial settings in offering new insights into the hidden weaknesses of pretrained policies that may otherwise appear undefeatable.

2 Background

Underspecified Stochastic Games In this work, we consider *Underspecified Stochastic Games (USG)*, i.e., stochastic games [40] with underspecified parameters of the environment. An USG game for N agent environment is defined by a set of states $\mathcal{S}$, actions $\mathcal{A}_1, ..., \mathcal{A}_N$ and a set of observations $\mathcal{O}_1, ..., \mathcal{O}_N$ for each of the agents. Each agent i select actions using a stochastic policy $\pi_i : \mathcal{O}_i \times \mathcal{A}_i \mapsto [0, 1]$. Θ defines the *free parameters* of the environments which are incorporated into the transition function $\mathcal{T} : \mathcal{S} \times \Theta \times \mathcal{A}_1 \times ... \times \mathcal{A}_N \mapsto \mathcal{S}$ which produces the next state based on the actions of all agents. Each agent i receives observations $\mathbf{o}_i : \mathcal{S} \mapsto \mathcal{O}_i$ correlated with the current state and reward $r_i : \mathcal{S} \times \mathcal{A}_i \mapsto \mathbb{R}$ as a function of the state and agent's action. The goal of each agent i is to maximise its own total expected return $R_i = \sum_{t=0}^{T} \gamma^t r_i^t$ for the time horizon T, where γ is a discount factor.

Each configuration of the free parameter $\theta \in \Theta$, which is often called a *level* [22, 33], defines a specific instantiation of the environment $\mathcal{M}_\theta$. For example, this can correspond to different positions of the walls in a maze, or locations of players and the ball in a football game. USG is a multi-agent variation of Underspecified POMDPs [11] and fully observable variant of UPOSGs [36].

Quality-Diversity Quality-diversity (QD) is a family of methods that are used to find a *diverse collection* of solutions that are *performant* and span a meaningful spectrum of solution characteristics [25, 8]. The performance of solution $x \in \mathcal{X}$ is measure using the *fitness* : $\mathcal{X} \mapsto \mathbb{R}$ function. The diversity of solutions is typically measured using the *feature_descriptor* : $\mathcal{X} \mapsto \mathbb{B}$ function that maps a solution into the feature space $\mathbb{B} = \mathbb{R}^K$ that describes specific characteristics of the solution, such as behavioral properties or visual appearance.

2.1 MAP-Elites

MAP-Elites is a simple and effective QD method [30]. Here, the descriptor space $\mathbb{B}$ is discretised and represented as an initially empty $N < K$ dimensional grid (archive). The algorithm starts by generating an arbitrary collection of candidate solutions. In each iteration, a solution is randomly selected among those in the grid. A new solution is obtained by mutating the selected solution, which is then evaluated and mapped to a cell of the grid based on its feature descriptor. The solution is then placed in the corresponding cell of the grid if it has a higher fitness than the current occupant, or if the cell if it is empty. This cycle of selection, mutation, and evaluation is repeated, progressively enhancing both the diversity (coverage) and the quality (fitness) of the collection. The pseudo-code of MAP-Elites is presented in Algorithm 1.

Algorithm 1: MAP-Elites [30]

Initialise: N-dimensional grids for solutions X and performances $\mathcal{P}$
Initialise: n cells of X with random solutions and corresponding cells of P with their fitness
for $i = \{1, 2, \dots\}$ **do**
  Sample solution x from X
  Get solution x' from x via random mutation
  $p' \leftarrow fitness(x')$
  $b' \leftarrow feature\_descriptor(x')$
  if $\mathcal{P}(b') = \emptyset$ *or* $\mathcal{P}(b') < p'$ **then**
    $\mathcal{P}(b') \leftarrow p'$
    $X(b') \leftarrow b'$

3 MADRID

In this section, we describe ***M**ulti-**A**gent **D**iagnostics for **R**obustness via **I**lluminated **D**iversity* (MADRID), a novel method for automatically generating diverse adversarial settings for a *target* pre-trained policy π_T. These are settings that either deceive the policy, forcing it to produce incorrect behaviour, or where the policy inherently performs poorly, deviating from the optimal behaviour. For

USGs, these settings correspond to particular environment levels $\theta \in \Theta$ that have been procedurally generated.

For quantifying adversarial levels, we make use the target policy's *regret* in level θ, i.e., the difference in utility between the optimal policy π^* and π_T :

$$\textsc{Regret}^{\theta}(\pi^*, \pi_T) = V^{\theta}(\pi^*, \pi_T) - V^{\theta}(\pi_T, \pi_T),$$

where $V_\theta(\pi_A, \pi_B) = \mathbb{E}[\sum_{t=0}^{T} \gamma^t r_t^A]$ is the value of a policy π_A against policy π_B in θ.[3]

Regret is a suitable metric for evaluating adversarial examples in pre-trained models. It provides a measure that directly quantifies the suboptimality of a model's decisions. While a high regret value serves as a glaring indicator of how far off a model's behavior is from the optimal choice, a low regret indicates the model's decisions are closely aligned with the optimal choice. The importance of regret becomes even more pronounced when considering the varied scenarios in which a model might be deployed. Therefore, by investigating regret across *diverse* situations, we can not only pinpoint specific vulnerabilities of a model but also ensure the robustness in previously unseen scenarios.

Since the optimal policy is usually unavailable, MADRID relies on utilising a collection of *suboptimal* policies $\Pi_R = \bigcup_{i=1}^{M} \pi_i$ for estimating the lower bound on true regret. Specifically, the goal is to find adversarial levels that maximise the gap in utility acquired through a *reference* policy $\pi_i \in \Pi_R$ and target policy π_T. Utilising a collection of diverse reference policies can be advantageous in the absence of a true optimal policy, since each of these reference policies may excel in a unique set of levels [36].

Algorithm 2: MADRID

Input: Target policy π_T
Input: A collection of reference policies Π_R
Input: $level\_descriptor : \Theta \mapsto \mathbb{R}^N$ function
 # Initialise a discretised grid, with an added dimension for Π_R, to archive levels and regret scores.
Initialise: $N+1$-dimensional grids for levels X and regret estimates $\mathcal{P}$
Initialise: n cells of X with randomly generated levels and corresponding estimated regret in $\mathcal{P}$
for $i = \{1, 2, \dots\}$ **do**
  # Sample a level θ and corresponding reference policy π_R from X.
  $\theta, \pi_R \sim X$
  # Perform level mutation.
  $\theta' \leftarrow \theta + \mathcal{N}(0,\, \sigma^2)$
  # Estimate the regret of π_T on θ' using π_R.
  $\widetilde{r}' \leftarrow V^{\theta'}(\pi_R, \pi_T) - V^{\theta'}(\pi_T, \pi_T)$
  $b' \leftarrow level\_descriptor(\theta')$
  if $\mathcal{P}(b', \pi_R) = \emptyset$ *or* $\mathcal{P}(b', \pi_R) < \widetilde{r}'$ **then**
    $\mathcal{P}(b', \pi_R) \leftarrow \widetilde{r}'$
    $X(b', \pi_R) \leftarrow b'$

MADRID casts the task of generating a diverse array of adversarial levels for each reference policy as a QD search problem. Specifically, MADRID uses MAP-Elites to systematically generate levels from Θ by discretising the feature space of levels into an N-dimensional grid, with an additional dimension representing the corresponding reference policy from Π_T. Using a discretised grid of MAP-Elites provides interpretability to the adversarial examples found in MADRID given that each cell defines specific environment parameters, alongside a reference policy which outperforms the target under these parameters.

MADRID starts by populating the grid with initial levels for each reference policy. During the iterative process, levels are selected from the grid to undergo mutation, followed by regret estimation. Each mutated level is then mapped to a specific cell in the grid based on its features and replaces the existing occupant if the mutated level has higher regret or the corresponding cell is unoccupied. This procedure ensures a thorough exploration and exploitation of the environment design space, allowing

[3]Note that here, for the simplicity of the notation, we assume a two-team zero sum setting. π_T and π_R describe the policies for groups of agents, either through a centralised controller or decentralised policies that employ parameter sharing. However, MADRID can be applied for more general multi-agent settings.

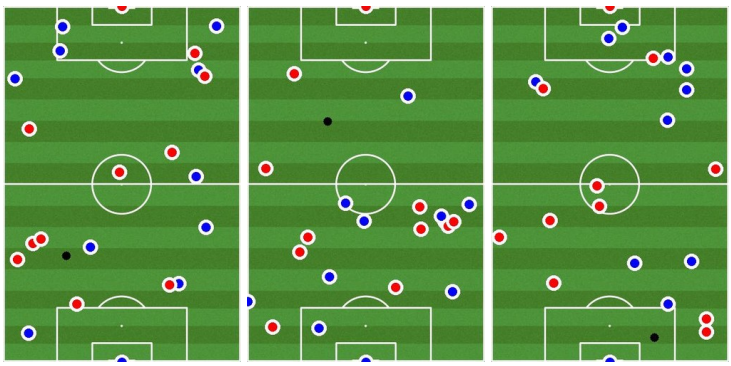


Figure 2: Examples of randomly generated levels on Google Research Football.

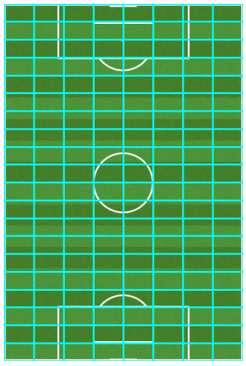


Figure 3: Dividing the field in 160 grids using the ball (x, y) coordinates.

MADRID to generate levels that are both diverse and high-regret. Figure 1 illustrates this process. Algorithm 2 provides the pseudocode of the method.

4 Experimental Setting

Our experiments seek to (1) showcase the effectiveness of MADRID in generating diverse adversarial settings for a target state-of-the-art pre-trained RL model, (2) analyse the adversarial settings generated by MADRID to find key weaknesses of the target model, (3) validate the design choices of MADRID by comparing it to two ablated baselines. To this end, we evaluate MADRID on Google Research Football [GRF, 23]. Given its strong performance and usage in related works, Covariance Matrix Adaptation MAP-Elites [CMA-ME, 16] serves as the base MAP-Elites method in our experiments. We provide full environment descriptions in Appendix B and implementation details in Appendix C.

Baselines We compare MADRID against two baselines: The *targeted baseline* uses a MAP-Elites archive but randomly samples levels from scratch, rather then evolving previously discovered high-regret levels from the grid. Consequently, it does not leverage the stepping stones to the optimisation problem [25]. The *random baseline* samples levels randomly from scratch without maintaining an archive of high-regret levels.

4.1 Environment

We use MADRID to find adversarial scenarios for TiZero, the state-of-the-art model for GRF. TiZero was trained via a complex regime on large-scale distributed infrastructure [27] over 45 days. In particular, we aim to generate adversarial levels whereby the decentralised agents in TiZero make a diverse array of strategic errors, as highlighted by better behaviours of the reference policy.

GRF is a complex open-source RL environment designed for training and evaluating agents to master the intricate dynamics of football, one of the world's most celebrated sports. It offers a physics-based 3D simulation that tasks the RL policy with controlling a team of players to penetrate the opponent's defense, while passing the ball among teammates, in order to score goals. GRF is a two-team zero-sum environment that has long been considered one of the most complex multi-agent RL benchmarks due to a unique combination of challenges [20, 50, 27], such as multi-agent cooperation, multi-agent competition, sparse rewards, large action and observation spaces, and stochastic dynamics.[4]

In this work, we focus on the fully decentralised 11 vs 11 version of the environment where each of the 10 RL agents on both side controls an individual player on the field.[5] Following [27], each agents receives as observation a 268-dimensional feature vector include own player information, player IDs, as well as information about the ball, player of the own and opponents teams, as well as

[4]Highlighting the stochasticity of the GRF environment, a shot from the top of the box can lead to various outcomes, underscoring that not every action results in a predictable outcome.

[5]The goalkeepers are controlled by the game AI.

general match details. The action space of agents consists of 19 discrete actions, such as moving in 8 direction, sprinting, passing, shooting, etc.

To apply MADRID on GRF, we utilise procedurally generated levels each represented as a vector consisting of (x, y) coordinates of 20 players [6] and the ball. The position of the ball on the field serves as a convenient descriptor for levels in GRF because it accommodates diverse scenarios, ranging from attacking to defending on both field halves. Therefore, we use the x and y coordinates of the ball as the first two environment features in MADRID. This leads to a categorisation of levels into 160 uniformly spaced cells across the football field, as illustrated in Figure 3. Given that we are interested in evaluating TiZero in specific adversarial levels, we restrict the episode length to 128 steps in our experiments.

The third axis for the MAP-Elites archive indexes the reference policies Π_R. In our experiments, we make use of 48 checkpoints of TiZero saved throughout its training [27], as well as three built-in bots in GRF with varying difficulties (easy, medium, hard). For each reference policy, we initialise the grid with randomly sampled levels that assign random locations to players and the ball. Figure 2 illustrates some of the randomly generated levels.

At each iteration of MADRID, we sample a level and reference policy pair (θ, π_R). The level is then mutated by adding Gaussian noise to the (x, y) positions of the players and the ball in the field. The fitness of each solution is estimated by computing TiZero's regret, which is the difference in performance between the selected reference policy π_R and TiZero's policy π_T. In both cases, we estimate the regret against the TiZero policy on the level θ as:

$$\widetilde{Regret}(\theta, \pi_T, \pi_R) = V^{\theta}(\pi_R, \pi_T) - V^{\theta}(\pi_T, \pi_T), \tag{1}$$

which corresponds to the difference of cross-play and self-play values between the reference and target policies. The performance on a given level θ between two policies π_A and π_B is the reward for scoring a goal:

$$V^{\theta}(\pi_A, \pi_B) = \begin{cases} 1 & \text{if } \pi_A \text{ scores} \\ 0 & \text{if no goal is scored} \\ -1 & \text{if } \pi_B \text{ scores} \end{cases} \tag{2}$$

Upon scoring a goal by either of the sides, the level terminates. Given the non-deterministic nature of GRF, we account for variability by calculating the average regret across 4 repetitions of the same pair of level θ and reference policy π_R.

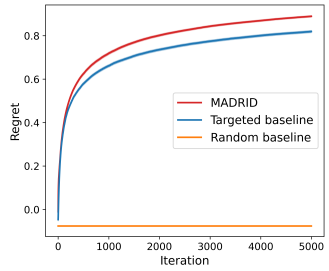
5 Results and Discussion

In our analysis of targeting TiZero on GRF, we closely examine the performance of MADRID and baselines. Figure 4a displays the average estimated regret values for all 160 cells within the MAP-Elites grid across the entire collection of reference policies. Here, MADRID outperforms both baseline methods. The *random baseline* exhibits a negative value close to 0, as TiZero proves to be a stronger policy than all the reference policies on entirely random game levels. On the other hand, the *targeted baseline* performs well, closely resembling MADRID's performance at the early stages of iterations. However, as the iterations continue, it lags behind due to its failure to capitalise on previously identified high-regret levels that serve as stepping stones for next iterations.

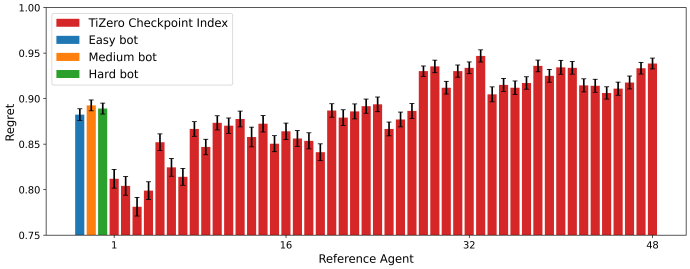
In Figure 4b, the variation in estimated final regret scores across the reference policies is illustrated. Here, the regret increases as we move to higher-ranked agents. The heuristic bots display regret levels that are on par with the intermediate checkpoints of TiZero.

As we approximate the regret using the difference between cross-play (XP) and self-play (SP) between reference and TiZero policies (see Equations 1 and 2), a regret estimate of 1 for an adversarial level θ can be achieved in two situations. First, the reference policy scores against TiZero in XP, while TiZero cannot score in its SP. Second, TiZero concedes a goal in SP in θ. Intriguingly, our findings reveal that for around 90% of the adversarial levels generated by MADRID, a nominally weaker reference policy outperforms TiZero. This emphasises MADRID's capability in exposing adversarial levels where even state-of-the-art policies be prone to missteps.

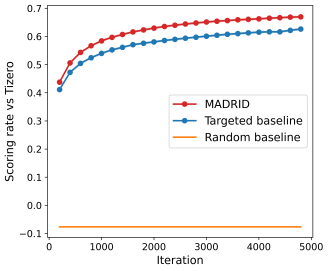
[6] The goalkeepers position positions are always near their own goals.



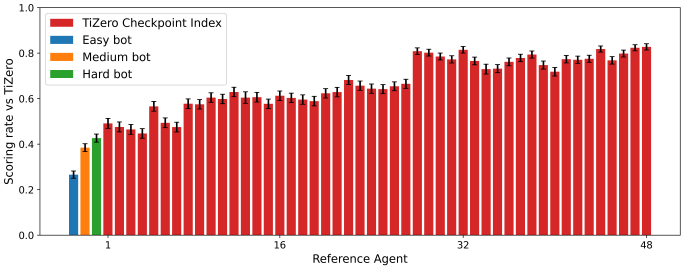
(a) Estimated regret at each iteration.



(b) Final estimated regret of TiZero over reference policies using MADRID.

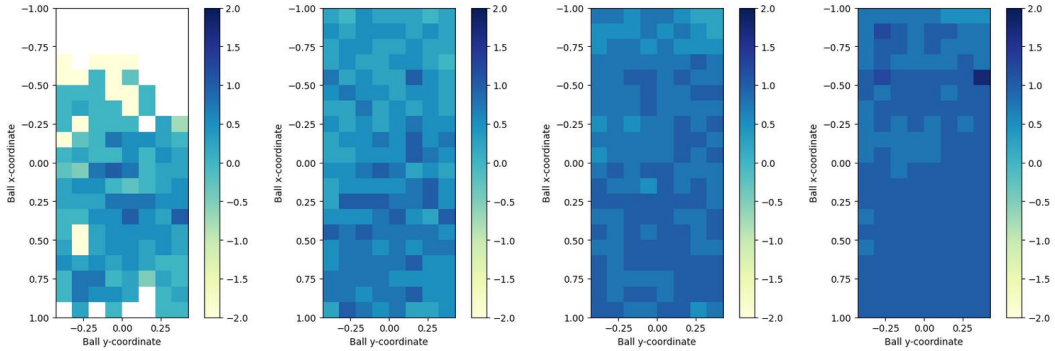


(c) Scoring rate vs TiZero at each iteration.



(d) Final estimated scoring rate vs TiZero over reference policies using MADRID.

Figure 4: The estimated regret and goal score rate against TiZero in Google Research Football. Illustrated throughout each iteration for 51 reference agents (a) and (c), as well as final values in (b) and (d). Standard error over 3 random seeds is shown.



(a) 25 iterations. (b) 200 iterations. (c) 1000 iterations. (d) 5000 iterations.

Figure 5: The coverage and regret values of TiZero in the grid at various iterations of MADRID with respect to TiZero-048 reference policy.

Figure 4c and Figure 4d illustrate the estimated rate of goals scored against TiZero by the reference policies on adversarial levels produced by MADRID and baselines. We can see that in approximately 70% of the time across all reference policies, the reference policy scored a goal against TiZero in a short period of time.[7] It should be noted that within the remaining 30%, the majority of instances resulted in no goals due to the nondeterministic dynamics of the environment.

Figure 22 in Appendix D highlights the difference in performance for selected reference policies. Notably, the higher-rank checkpoints of TiZero, saved at the later stages of its training, can be used to identify more severe vulnerabilities, as measured using the regret estimate.

Figure 5 shows the evolution of MADRID's archive for a specific reference policy, illustrating its search process over time. Initially, the grid is sparsely filled with low-regret levels. However, as

[7]The levels last only 128 steps, which is a short episode compared to the 3000 steps for the full game.

(a) Initial player and ball positions in the level. TiZero is about to pass the ball to a teammate.

(b) The receiving player is clearly in offside, thus a freekick is awarded to the opponents team.

(c) Reference policy does not pass to offside player and directly runs towards the goal to score.

Figure 6: Adversarial example of offsides.

Figure 7: Adversarial example of an own goal. TiZero gets tricked and shoots in its own goal.

iterations progress, MADRID generates high-regret levels that progressively populate the entire grid. This shows that MADRID can discover high-regret levels anywhere on the football field. On average, we notice that higher-level scenarios tend to be located towards the positive x coordinates. These correspond to situations where the ball is close to the opponent's goal from the perspective reference policy. While most regret scores tend to have uniform values around in similar positions on the field, in Figure 5d the grid also includes an adversarial level with estimated regret of 1.75. This indicates that MADRID found a level where the reference policy scores against TiZero in XP, while TiZero concedes a goal in SP.

5.1 Qualitative Analysis

Next we conduct a qualitative analysis of the adversarial levels identified by MADRID on GRF by visualising the highest ranking levels in the archive across all reference policies. We provide a selection of these examples below, with a comprehensive list available in Appendix A. Full videos of all identified vulnerabilities can be found at `https://sites.google.com/view/madrid-marl`.

Offsides Despite its strong performance under standard evaluations, TiZero frequently falls victim to erroneously passing the ball to players unmistakably in offside positions, as shown in Figure 6 This highlights TiZero's lack of a deep understanding of the rules of the game. In contrast, the reference policies abstain from passing the ball to offside players, resulting in successful scoring outcomes.[8]

Unforced Own Goals Perhaps the most glaring adversarial behaviour discovered are instances where TiZero agents inexplicably shoot towards their own goal, resulting in unforced own goals (See Figure 7). In contrast, when starting from identical in-game positions, the reference policies manage to counterattack effectively, often resulting in successful scoring endeavors.

Slow running opponents The TiZero agents always choose to sprint throughout the episode. However, this makes them weak on defense against opponents who move slower with the ball. Instead of trying to tackle and take the ball, TiZero's main defensive strategy is to try and block opponents. Opponents can take advantage of this by using deceptive moves, especially when moving slowly, making it hard for TiZero's defenders to stop them. This is illustrated in Figure 8.

[8] A player is offside when it is in the opponents' half and any part of their body is closer to the opponents' goal line than both the ball and the second-last opponent. Usually one of the two opponents is the goalkeeper. When this happens a free kick is awarded to the opponent's team.

Figure 8: Adversarial example of a slow running opponent. Three TiZero-controlled defenders are not able to stop a simple slow running opponent, who walks past them and scores.

(a) Initial player and ball positions in the level.

(b) TiZero shoots from a narrow angle and is blocked by the goalkeeper

(c) Reference policy goes to shoot from a better position and scores

Figure 9: Adversarial example of better shooting positioning.

Suboptimal Ball Positioning for Shooting When trying to score a goal, TiZero agents often choose a suboptimal positioning, such as shooting from a narrow angle. In contrast, the reference policies often make subtle adjustments to optimally position the ball before initiating a shot (e.g., move towards the centre of the goals Figure 9).

Passing to Better Positioned Players A notable shortcoming in TiZero's policy, when compared to the built-in heuristic, is its reluctance to pass the ball to teammates who are in more favorable positions and have a higher likelihood of scoring, as illustrated in Figure 10. In contrast, heuristic bots—whether easy, medium, or hard—demonstrate a consistent pattern of passing to optimally positioned players, enhancing their goal-scoring opportunities. This effective passing strategy seems unfamiliar to TiZero, causing it difficulty in overcoming a successful defense.

6 Related Work

Quality Diversity

Quality Diversity (QD) is a category of open-ended learning methods aimed at discovering a collection of solutions that are both highly diverse and performant[25, 8]. Two commonly used QD algorithms are Novelty Search with Local Competition [NSLC, 25] and MAP-Elites [30, 7]. These two approaches differ in the way they structure the archive; novelty search completely forgoes a grid and opts instead for growing an unstructured archive that dynamically expands, while MAP-Elites

(a) Initial player and ball positions in the level.

(b) TiZero (blue) runs towards the goal and shoots, getting blocked by the goalkeeper.

(c) Reference policy (blue) passes the ball to a better positioned player who scores.

Figure 10: Adversarial example of passing.

adopts a static mapping approach. Although MADRID leverages MAP-Elites as its diversity mechanism, it can be adapted to use NSLC. One of the most effective versions of MAP-Elites is CMA-ME [16]. CMA-ME combines MAP-Elites with the evolutionary optimization algorithm Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [18], improving the selection of the fittest solutions which will be perturbed to generate new elites. Mix-ME [21] extends MAP-Elites to multi-agent domains, but is limited to fully cooperative settings.

Multi-Agent RL

Recent advancements in the field of cooperative multi-agent RL [15, 34, 10, 28] have shown remarkable success in tackling complex challenges in video games, such as StarCraft II [37, 13]. Google Research Football [GRF, 23] stands as one of the most complex multi-agent RL benchmarks, as a two-team zero-sum game with sparse reward and requiring significant amount of coordination between co-players. Most of the prior work on addresses the toy settings of the GRF only involved a few agents (e.g., 3 vs 1 scenario). Multi-Agent PPO [MAPPO, 53] uses PPO [39] with a centralised critic to play on toy settings. CDS [26] analyses the importance of diversity between policies in GRF. Multi-Agent Transformer [MAT, 50] models GRF as a sequence problem using the self-attention mechanism. TiKick [20] attempts to solve the full 11 vs 11 game using demonstrations from single-agent trajectories. SPC [48] uses an adaptive curriculum on handcrafted environments for overcoming the sparse reward issue in GRF. TiZero is the first method that claims to have mastered the full 11 vs 11 game of GRF from scratch [27] following 45 days of training with large amount of computational resources. To achieve this, TiZero uses a hand-crafted curricula over environment variations, self-play, augmented observation space, reward shaping, and action masking.

Adversarial Attacks on Multi-Agent Policies

Deep neural networks, such as image classifiers, are known to be sensitive to adversarial attacks [42, 5, 35]. Such susceptibility has also been demonstrated in multi-agent RL. [49] attacks the leading Go-playing AI, KataGo [51], by training adversarial policies and achieving >97% win rate against it. Such adversarial agents are not expert Go-playing bots at all and are easily defeated by amateur human players, instead they simply trick KataGo into making serious blunders. Similarly, [43] introduce ISMCTS-BR, a search-based deep RL algorithm that learns a best response to a given agent. Both of these solutions find exploitability using RL and expensive Monte-Carlo tree search [6], whereas MADRID is a fast, gradient-free, training-free method that finds adversarial settings using QD. Furthermore, unlike the previous methods, MADRID is not restricted to any concrete agent architecture and is more general in nature. MAESTRO [36] crafts adversarial curricula for training robust agents in 2-player settings by jointly sampling environment/co-player pairs, emphasizing the interplay between agents and environments.

7 Conclusion and Future Work

This paper introduced Multi-Agent Diagnostics for Robustness via Illuminated Diversity (MADRID), a novel approach aimed at systematically uncovering situations where pre-trained multi-agent RL agents display strategic errors in complex environments. MADRID leverages quality-diversity mechanisms and employs the concept of regret to identify and quantify a multitude of scenarios where agents enact suboptimal strategies, with a particular focus on the advanced TiZero agent within the Google Research Football environment. Our investigations using MADRID revealed several previously unnoticed vulnerabilities in TiZero's strategic decision-making, such as ineffective finishing and misunderstandings of the offside rule, highlighting the hidden strategic inefficiencies and latent vulnerabilities in even the most advanced RL agents.

Looking forward, we are excited to apply MADRID to a wider range of multi-agent domains and integrate it with more sophisticated evolutionary and learning-based approaches to further expand its capability in identifying strategic inefficiencies. Exploring interactions between varying adversarial scenarios and different RL models can provide deeper insights into the inherent strategic complexities and adaptive learning processes in multi-agent environments, thereby contributing to the evolution of more robust and sophisticated solutions in the field of RL. Furthermore, future research with MADRID can focus on optimising mitigation strategies to both identify and rectify strategic errors, advancing the development of more robust multi-agent systems.

References

[1] Anthropic. Introducing Claude, 2023. Accessed on Oct 6, 2023. 1

[2] Claudine Badue, Rânik Guidolini, Raphael Vivacqua Carneiro, Pedro Azevedo, Vinicius Brito Cardoso, Avelino Forechi, Luan Jesus, Rodrigo Berriel, Thiago Paixão, Filipe Mutz, Lucas Veronese, Thiago Oliveira-Santos, and Alberto Ferreira De Souza. Self-driving cars: A survey, 2019. 1

[3] 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, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 434–443. PMLR, 09–15 Jun 2019. 2

[4] 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. 1

[5] Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Madry, and Alexey Kurakin. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*, 2019. 10

[6] Rémi Coulom. Efficient selectivity and backup operators in monte-carlo tree search. In H. Jaap van den Herik, Paolo Ciancarini, and H. H. L. M. (Jeroen) Donkers, editors, *Computers and Games*, pages 72–83, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. 10

[7] Antoine Cully, Jeff Clune, Danesh Tarapore, and Jean-Baptiste Mouret. Robots that can adapt like animals. *Nature*, 521:503–507, 2015. 9

[8] Antoine Cully and Yiannis Demiris. Quality and diversity optimization: A unifying modular framework. *IEEE Transactions on Evolutionary Computation*, 22(2):245–259, 2018. 2, 3, 9

[9] 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. 2

[10] Christian Schroeder de Witt, Tarun Gupta, Denys Makoviichuk, Viktor Makoviychuk, Philip H. S. Torr, Mingfei Sun, and Shimon Whiteson. Is independent learning all you need in the starcraft multi-agent challenge?, 2020. 10

[11] 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. 2, 3

[12] Adrien Ecoffet, Joost Huizinga, Joel Lehman, Kenneth O. Stanley, and Jeff Clune. First return, then explore. *Nature*, 590:580 – 586, 2020. 2

[13] Benjamin Ellis, Skander Moalla, Mikayel Samvelyan, Mingfei Sun, Anuj Mahajan, Jakob N. Foerster, and Shimon Whiteson. Smacv2: An improved benchmark for cooperative multi-agent reinforcement learning, 2022. 10

[14] Jakob Foerster, Yannis M Assael, Nando de Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 2137–2145, 2016. 1

[15] Jakob N. Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press, 2018. 10

[16] Matthew C. Fontaine, Julian Togelius, Stefanos Nikolaidis, and Amy K. Hoover. Covariance matrix adaptation for the rapid illumination of behavior space. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, GECCO '20, page 94–102, New York, NY, USA, 2020. Association for Computing Machinery. 5, 10

[17] 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. 2

[18] Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evol. Comput.*, 9(2):159–195, jun 2001. 10

[19] Sebastian Höfer, Kostas Bekris, Ankur Handa, Juan Camilo Gamboa, Melissa Mozifian, Florian Golemo, Chris Atkeson, Dieter Fox, Ken Goldberg, John Leonard, et al. Sim2real in robotics and automation: Applications and challenges. *IEEE transactions on automation science and engineering*, 18(2):398–400, 2021. 1

[20] Shiyu Huang, Wenze Chen, Longfei Zhang, Shizhen Xu, Ziyang Li, Fengming Zhu, Deheng Ye, Ting Chen, and Jun Zhu. Tikick: towards playing multi-agent football full games from single-agent demonstrations. *arXiv preprint arXiv:2110.04507*, 2021. 5, 10

[21] Garðar Ingvarsson, Mikayel Samvelyan, Bryan Lim, Manon Flageat, Antoine Cully, and Tim Rocktäschel. Mix-me: Quality-diversity for multi-agent learning, 2023. 10

[22] 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*. 2021. 2, 3

[23] Karol Kurach, Anton Raichuk, Piotr Stańczyk, Michał Zając, Olivier Bachem, Lasse Espeholt, Carlos Riquelme, Damien Vincent, Marcin Michalski, Olivier Bousquet, and Sylvain Gelly. Google research football: A novel reinforcement learning environment, 2020. 2, 5, 10, 18

[24] 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. 1

[25] Joel Lehman and Kenneth O Stanley. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary computation*, 19(2):189–223, 2011. 2, 3, 5, 9

[26] Chenghao Li, Tonghan Wang, Chengjie Wu, Qianchuan Zhao, Jun Yang, and Chongjie Zhang. Celebrating diversity in shared multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 34:3991–4002, 2021. 10

[27] Fanqi Lin, Shiyu Huang, Tim Pearce, Wenze Chen, and Wei-Wei Tu. Tizero: Mastering multi-agent football with curriculum learning and self-play. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '23, page 67–76, Richland, SC, 2023. International Foundation for Autonomous Agents and Multiagent Systems. 2, 5, 6, 10, 18, 19

[28] Anuj Mahajan, Tabish Rashid, Mikayel Samvelyan, and Shimon Whiteson. Maven: Multi-agent variational exploration. *Advances in Neural Information Processing Systems*, 32, 2019. 10

[29] Ishita Mediratta, Minqi Jiang, Jack Parker-Holder, Michael Dennis, Eugene Vinitsky, and Tim Rocktäschel. Stabilizing unsupervised environment design with a learned adversary, 2023. 2

[30] Jean-Baptiste Mouret and Jeff Clune. Illuminating search spaces by mapping elites, 2015. 2, 3, 9

[31] OpenAI. GPT-4 Technical Report, 2023. 1

[32] OpenRL-Lab. Tizero. `https://github.com/OpenRL-Lab/TiZero`, 2023. GitHub repository. 19

[33] Jack Parker-Holder, Minqi Jiang, Michael Dennis, Mikayel Samvelyan, Jakob Foerster, Edward Grefenstette, and Tim Rocktäschel. Evolving curricula with regret-based environment design, 2022. 2, 3

[34] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 4295–4304. PMLR, 2018. 2, 10

[35] Kui Ren, Tianhang Zheng, Zhan Qin, and Xue Liu. Adversarial attacks and defenses in deep learning. *Engineering*, 6(3):346–360, 2020. 10

[36] Mikayel Samvelyan, Akbir Khan, Michael D Dennis, Minqi Jiang, Jack Parker-Holder, Jakob Nicolaus Foerster, Roberta Raileanu, and Tim Rocktäschel. MAESTRO: Open-ended environment design for multi-agent reinforcement learning. In *International Conference on Learning Representations*, 2023. 1, 2, 3, 4, 10

[37] Mikayel Samvelyan, Tabish Rashid, Christian Schroeder de Witt, Gregory Farquhar, Nantas Nardelli, Tim GJ Rudner, Chia-Man Hung, Philip HS Torr, Jakob Foerster, and Shimon Whiteson. The starcraft multi-agent challenge. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 2186–2188, 2019. 10

[38] 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. 1

[39] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *ArXiv*, abs/1707.06347, 2017. 10

[40] L. S. Shapley. Stochastic games. *Proceedings of the National Academy of Sciences*, 39(10):1095–1100, 1953. 3

[41] 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. 1

[42] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 10

[43] Finbarr Timbers, Nolan Bard, Edward Lockhart, Marc Lanctot, Martin Schmid, Neil Burch, Julian Schrittwieser, Thomas Hubert, and Michael Bowling. Approximate exploitability: Learning a best response. In Lud De Raedt, editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 3487–3493. International Joint Conferences on Artificial Intelligence Organization, 7 2022. Main Track. 10

[44] Bryon Tjanaka, Matthew C Fontaine, David H Lee, Yulun Zhang, Nivedit Reddy Balam, Nathaniel Dennler, Sujay S Garlanka, Nikitas Dimitri Klapsis, and Stefanos Nikolaidis. Pyribs: A bare-bones python library for quality diversity optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '23, page 220–229, New York, NY, USA, 2023. Association for Computing Machinery. 19

[45] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov,

Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. 1

[46] Karl Tuyls, Shayegan Omidshafiei, Paul Muller, Zhe Wang, Jerome Connor, Daniel Hennes, Ian Graham, William Spearman, Tim Waskett, Dafydd Steel, Pauline Luc, Adria Recasens, Alexandre Galashov, Gregory Thornton, Romuald Elie, Pablo Sprechmann, Pol Moreno, Kris Cao, Marta Garnelo, Praneet Dutta, Michal Valko, Nicolas Heess, Alex Bridgland, Julien Pérolat, Bart De Vylder, S. M. Ali Eslami, Mark Rowland, Andrew Jaegle, Remi Munos, Trevor Back, Razia Ahamed, Simon Bouton, Nathalie Beauguerlange, Jackson Broshear, Thore Graepel, and Demis Hassabis. Game plan: What ai can do for football, and what football can do for ai. *J. Artif. Int. Res.*, 71:41–88, sep 2021. 2

[47] 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. 1, 2

[48] Rundong Wang, Longtao Zheng, Wei Qiu, Bowei He, Bo An, Zinovi Rabinovich, Yujing Hu, Yingfeng Chen, Tangjie Lv, and Changjie Fan. Towards skilled population curriculum for multi-agent reinforcement learning. *arXiv preprint arXiv:2302.03429*, 2023. 10

[49] Tony Tong Wang, Adam Gleave, Tom Tseng, Kellin Pelrine, Nora Belrose, Joseph Miller, Michael D Dennis, Yawen Duan, Viktor Pogrebniak, Sergey Levine, and Stuart Russell. Adversarial policies beat superhuman go AIs. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 35655–35739. PMLR, 23–29 Jul 2023. 10

[50] Muning Wen, Jakub Kuba, Runji Lin, Weinan Zhang, Ying Wen, Jun Wang, and Yaodong Yang. Multi-agent reinforcement learning is a sequence modeling problem. *Advances in Neural Information Processing Systems*, 35:16509–16521, 2022. 5, 10

[51] David J Wu. Accelerating self-play learning in go. *arXiv preprint arXiv:1902.10565*, 2019. 10

[52] 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. 1

[53] Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of PPO in cooperative multi-agent games. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. 2, 10

[54] Wenshuai Zhao, Jorge Peña Queralta, and Tomi Westerlund. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 737–744, 2020. 1

(a) Initial player and ball positions in the level. TiZero is about to pass the ball to a teammate.

(b) The receiving player is clearly in offside, thus a freekick is awarded to the opponents team.

(c) Reference policy does not pass to offside player and directly runs towards the goal to score.

Figure 11: Adversarial example of offsides.

Figure 12: Adversarial example of an own goal. TiZero gets tricked and shoots in its own goal.

A Adversarial Examples for Google Research Football

Below are 11 adversarial examples in TiZero we identifying using MADRID.

Offsides Despite its strong performance under standard evaluations, TiZero frequently falls victim to erroneously passing the ball to players unmistakably in offside positions, as shown in Figure 11 This observations highlights TiZero's lack of a deep understanding of the rules of the game. In contrast, the reference policies abstain from passing the ball to offside players, resulting in successful scoring outcomes.[9]

Unforced Own Goals Perhaps the most glaring adversarial behaviour discovered are instances where TiZero agents inexplicably shoot towards their own goal, resulting in unforced own goals (See Figure 12). In contrast, when starting from identical in-game positions, the reference policies manage to counterattack effectively, often resulting in successful scoring endeavors.

Slow running opponents The TiZero agents always choose to sprint throughout the episode. However, this makes them weak on defense against opponents who move slower with the ball. Instead of trying to tackle and take the ball, TiZero's main defensive strategy is to try and block opponents. Opponents can take advantage of this by using deceptive moves, especially when moving slowly, making it hard for TiZero's defenders to stop them. This is illustrated in Figure 13.

[9] A player is offside when it is in the opponents' half and any part of their body is closer to the opponents' goal line than both the ball and the second-last opponent. Usually one of the two opponents is the goalkeeper. When this happens a free kick is awarded to the opponent's team.

Figure 13: Adversarial example of a slow running opponent. Three TiZero-controlled defenders are not able to stop a simple slow running opponent, who walks past them and scores.

(a) Initial player and ball positions in the level.

(b) TiZero shoots from a narrow angle is blocked by the goalkeeper

(c) Reference policy goes to shoot from a better position and scores

Figure 14: Adversarial example of better shooting positioning.

(a) Initial player and ball positions in the level.

(b) TiZero (blue) runs towards the goal and shoots, getting blocked by the goalkeeper.

(c) Reference policy (blue) passes the ball to a better positioned player who scores.

Figure 15: Adversarial example of passing.

Suboptimal Ball Positioning for Shooting When trying to score a goal, TiZero agents often choose a suboptimal positioning, such as shooting from a narrow angle. In contrast, the reference policies often make subtle adjustments to optimally position the ball before initiating a shot (e.g., move towards the centre of the goals Figure 14).

Passing to Better Positioned Players A notable shortcoming in TiZero's policy, when compared to the built-in heuristic, is its reluctance to pass the ball to teammates who are in more favorable positions and have a higher likelihood of scoring, as illustrated in Figure 15. In contrast, heuristic bots—whether easy, medium, or hard—demonstrate a consistent pattern of passing to optimally positioned players, enhancing their goal-scoring opportunities. This effective passing strategy seems unfamiliar to TiZero, causing it difficulty in overcoming a successful defense.

Shooting while Running Capitalizing on another game mechanics, the reference policies exhibit stronger behaviours by halting their sprinting behaviour leading up to a shot, resulting in a notably higher success rate in goal realisation. TiZero's agents, in contrast, consistently maintain a sprinting stance, thereby frequently missing straightforward scoring opportunities in front of the opposing goalkeepers (Figure 16).

(a) Initial player and ball positions in the level.

(b) TiZero shoots while sprinting and the ball gets blocked by the goalkeeper.

(c) Reference policy doesn't run and is able to score.

Figure 16: Adversarial example of shooting while running.

(a) Initial player and ball positions in the level.

(b) TiZero aimlessly runs up and down from the same position in an endless loop.

(c) The reference policy attacks the opponent goal, often resulting in goal scoring endeavours.

Figure 17: Adversarial example of confused behaviour.

(a) Initial player and ball positions in the level.

(b) TiZero's defender runs along a suboptimal trajectory, giving space for the opponent to shoot and score.

(c) Reference policy instead runs towards the attacker to block the attempt.

Figure 18: Adversarial example of better defensive behaviour.

Confused Agent Behavior Another intriguing adversarial instance finds TiZero's ball-possessing player aimlessly sprinting back and forth in random areas of the field, thereby exhibiting a completely unproductive pattern of movement (Figure 17).

Improved Defensive Positioning TiZero shows several vulnerabilities in its defensive strategies, failing to close down on the opponent attacking trajectory and allowing them to score. In comparison, Figure 18 shows the reference policies closing down on the opponent striker and seizing the ball before they have the chance to shoot.

Erroneous Team Movement Several adversarial examples show the entirety of TiZero's team running in the wrong direction to defend their goal, while the ball is positioned favourably towards the opponents goal, leaving a solitary attacking player without support, who gets deceived and performs poorly. The reference policy instead doesn't get tricked and often manages to score despite the disarray (Figure 19).

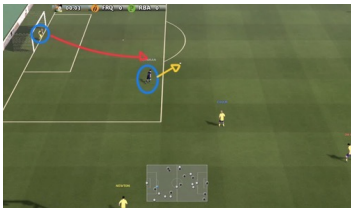
Hesitation Before Shooting The most common adversarial scenario encountered by the heuristic bots is situations in which TiZero hesitates before taking a shot, allowing the goalkeeper or defending

(a) Initial player and ball positions in the level.

(b) TiZero's team runs backwards, leaving a solitary attacker confused and unable to score.

(c) Reference policy instead doesn't get tricked, the attacker moves in a better position to score.

Figure 19: Adversarial example of erroneous team movement.



(a) Initial player and ball positions in the level.

(b) TiZero (in blue) hesitates before shooting, giving enough time for the goalkeeper to seize the ball

(c) Reference policy instead shoots without hesitation and scores.

Figure 20: Adversarial example of hesitation before shooting.

(a) Initial player and ball positions in the level.

(b) TiZero's attacker does not realise it can get to the ball before the goalkeeper, and runs backwards.

(c) Reference policy instead runs towards the ball, reaching it before the goal keeper does and scoring.

Figure 21: Adversarial example of missing a goal scoring opportunity.

players to seize the ball. In contrast, the inbuilt bot promptly recognizes the opportunity and shoots without hesitation, resulting in successful scoring (Figure 20).

Missing a Goal Scoring Opportunity TiZero often fails to acknowledge easy goal scoring opportunity, where it could get to the ball and score, but instead decides not to pursue it. Figure 21 shows how the reference policy capitalises on this kind of opportunity and scores.

B Environment Details

In our experiments with Google Research Football [23], we adopt a procedural generation method for level creation. For each player, as well as the ball, we randomly sample the (x, y) coordinates: the x-coordinate is sampled from the range $[-0.9, 0.9]$ and the y-coordinate from the range $[-0.4, 0.4]$. The settings employed during the generation are as follows:

- `deterministic`: set to `False`, implying that levels can have non-deterministic components.
- `offsides`: set to `True`, enforcing the offsides rule during gameplay.
- `end_episode_on_score`: set to `True`, which means the episode will terminate once a goal is scored.
- `end_episode_on_out_of_play`: set to `False`, indicating the episode will not end on ball out-of-play events.
- `end_episode_on_possession_change`: set to `False`, indicating the episode will not end when the ball changes possession from one team to another.

For the *easy* bot, the difficulty is set at 0.05. For the *medium* bot, it is set to 0.5, and for the *hard* bot, the difficulty is at 0.95. These values serve as the defaults in GRF, ensuring consistency across different game scenarios

We use the enhanced observation space as described in TiZero [27], consisting of 268-dimensional vector including information.

Table 1: Hyperparameters used for finding adversarial examples in Google Research Football.

Parameter	
Number of steps	5000
Game duration	128
Number of CMA-ME emitters	4
Number of repeats per level	4
Emitter gaussian noise σ	0.1
Ranker	improvement
QD score offset	-2

C Implementation Details

Hyperparameters of MADRID are provided in Table 1. We use the CMA-ME as implemented in pyribs [44]. For the TiZero and reference agents, we use the exact agent architecture as in the original paper [27] using TiZero's official open-source release [32]. Parameter sharing is applied to all agents in the team.

The policy network is made up of six different multi-layer perceptrons (MLPs), each having two fully-connected layers, including one specifically for the 'player ID', to encode every part of the observation individually. The MLP layers have a hidden size of 64. The hidden features extracted are brought together and then handled by an LSTM layer to give the agent memory, with the hidden size for this layer being 256. Every hidden layer is equipped with layer normalization and ReLU non-linearities. The orthogonal matrix is used for initializing parameters, and the learning process is optimized with the Adam optimizer. Similar to the original implementation, illegal actions are masked out by making their selection probability zero. The action output layer utilizes a softmax layer and is formed with a 19-dimension vector.

Experiments are conducted on an in-house cluster. Every task, denoted by a seed, uses one Tesla V100 GPU and 10 CPUs. For each of the 51 reference policies (48 TiZero checkpoints and 3 built-in bots), we use 3 random seeds, for each of the baselines. Runs last approximately 8.5 days for 5000 iterations of MADRID.

D Additional results

Figure 22 highlights the difference in performance for selected reference policies. Notably, the higher-rank checkpoints of TiZero, saved at the later stages of its training, can be used to identify more severe vulnerabilities, as measured using the regret estimate.

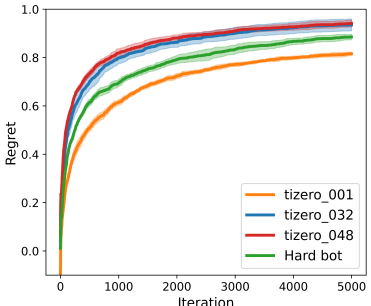


Figure 22: MADRID's estimated regret over different reference policies after each iteration GRF (mean and standard error over 3 seeds).