CONTINUOUSLY DISCOVERING NOVEL STRATEGIES VIA REWARD-SWITCHING POLICY OPTIMIZATION

Zihan Zhou$^{*\dagger\ 1\flat}$, **Wei Fu**$^{*\ 2\sharp}$, **Bingliang Zhang**^{2}, **Yi Wu**$^{23\natural}$
[1] CS Department, University of Toronto, [2] IIIS, Tsinghua University, [3] Shanghai Qi Zhi Institute
$^{\flat}$ footoredo@gmail.com, $^{\sharp}$ fuwth17@gmail.com, $^{\natural}$ jxwuyi@gmail.com

ABSTRACT

We present Reward-Switching Policy Optimization (RSPO), a paradigm to discover diverse strategies in complex RL environments by iteratively finding novel policies that are both locally optimal and sufficiently different from existing ones. To encourage the learning policy to consistently converge towards a previously undiscovered local optimum, RSPO switches between extrinsic and intrinsic rewards via a trajectory-based novelty measurement during the optimization process. When a sampled trajectory is sufficiently distinct, RSPO performs standard policy optimization with extrinsic rewards. For trajectories with high likelihood under existing policies, RSPO utilizes an intrinsic diversity reward to promote exploration. Experiments show that RSPO is able to discover a wide spectrum of strategies in a variety of domains, ranging from single-agent navigation tasks and MuJoCo control to multi-agent stag-hunt games and the StarCraft II Multi-Agent Challenge.

1 INTRODUCTION

The foundation of deep learning successes is the use of stochastic gradient descent methods to obtain a local minimum for a highly non-convex learning objective. It has been a popular consensus with theoretical justifications that most local optima are very close to the global optimum (Ma, 2020). Consequently, algorithms for most classical deep learning applications only focus on the final performance of the learned local solution rather than *which* local minimum is discovered.

However, this assumption can be problematic in reinforcement learning (RL), where different local optima in the policy space can correspond to substantially different strategies. Therefore, discovering a diverse set of policies can be critical for many RL applications, such as producing natural dialogues in chatbot (Li et al., 2016), improving the chance of finding a targeted molecule (Pereira et al., 2021), generating novel designs (Wang et al., 2019) or training a specialist robot for fast adaptation (Cully et al., 2015). Moreover, in the multi-agent setting, a collection of diverse local optima could further result in interesting emergent behaviors (Liu et al., 2019; Zheng et al., 2020; Baker et al., 2020) and discovery of multiple Nash equilibria (Tang et al., 2021), which further help build strong policies that can adapt to unseen participating agents in a zero-shot manner in competitive (Jaderberg et al., 2019; Vinyals et al., 2019) and cooperative games (Lupu et al., 2021).

In order to obtain diverse strategies in RL, most existing works train a large population of policies in parallel (Pugh et al., 2016; Cully et al., 2015; Parker-Holder et al., 2020b). These methods often adopt a soft learning objective by introducing additional diversity intrinsic rewards or auxiliary losses. However, when the underlying reward landscape in the RL problem is particularly non-uniform, policies obtained by population-based methods often lead to visually identical strategies (Omidshafiei et al., 2020; Tang et al., 2021). Therefore, population-based methods may require a substantially large population size in order to fully explore the policy space, which can be computationally infeasible. Moreover, the use of soft objective also results in non-trivial and subtle hyper-parameter tuning to balance diversity and the actual performance in the environment, which largely prevents these existing methods from discovering *both diverse and high-quality* policies in practice (Parker-Holder et al., 2020b; Lupu et al., 2021; Masood & Doshi-Velez, 2019). Another type of methods directly explores diverse strategies in the reward space by performing multi-objective optimization over

[*] Equal Contribution.
[†] Work done as a residence researcher at Shanghai Qi Zhi Institute.

human-designed behavior characterizations (Pugh et al., 2016; Cully et al., 2015) or random search over linear combinations of the predefined objectives (Tang et al., 2021; Zheng et al., 2020; Ma et al., 2020). Although these multi-objective methods are particularly successful, a set of well-defined and informative behavior objectives may not be accessible in most scenarios.

We propose a simple, generic and effective *iterative* learning algorithm, *Reward-Switching Policy Optimization (RSPO)*, for continuously discovering novel strategies under a single reward function without the need of any environment-specific inductive bias. RSPO discovers novel strategies by solving a filtering-based objective, which restricts the RL policy to converge to a solution that is sufficiently different from a set of locally optimal reference policies. After a novel strategy is obtained, it becomes another reference policy for future RL optimization. Therefore, by repeatedly running RSPO, we can quickly derive diverse strategies in just a few iterations. In order to strictly enforce the novelty constraints in policy optimization, we adopt rejection sampling instead of optimizing a soft objective, which is adopted by many existing methods by converting the constraints as Lagrangian penalties or intrinsic rewards. Specifically, RSPO *only* optimizes extrinsic rewards over trajectories that have sufficiently low likelihood w.r.t. the reference policies. Meanwhile, to further utilize those rejected trajectories that are not distinct enough, RSPO ignores the environment rewards on these trajectories and only optimizes diversity rewards to promote effective exploration. Intuitively, this process adaptively switches the training objective between extrinsic rewards and diversity rewards w.r.t. the novelty of each sampled trajectory, so we call it the *Reward Switching* technique.

We empirically validate RSPO on a collection of highly multi-modal RL problems, ranging from multi-target navigation (Mordatch & Abbeel, 2018) and MuJoCo control (Todorov et al., 2012) in the single-agent domain, to stag-hunt games (Tang et al., 2021) and the StarCraft II Multi-Agent Challenge (SMAC) (Rashid et al., 2019) in the multi-agent domain. Experiments demonstrate that RSPO can reliably and efficiently discover surprisingly diverse strategies in all these challenging scenarios and substantially outperform existing baselines. The contributions can be summarized as follows:

1. We propose a novel algorithm, *Reward-Switching Policy Optimization*, for continuously discovering diverse policies. The iterative learning scheme and reward-switching technique both significantly benefit the efficiency of discovering strategically different policies.
2. We propose to use cross-entropy-based diversity metric for policy optimization and two additional diversity-driven intrinsic rewards for promoting diversity-driven exploration.
3. Our algorithm is both general and effective across a variety of single-agent and multi-agent domains. Specifically, our algorithm is the first to learn the optimal policy in the stag-hunt games without any domain knowledge, and successfully discovers 6 visually distinct winning strategies via merely 6 iterations on a hard map in SMAC.

2 Related work

Searching for diverse solutions in a highly multi-modal optimization problem has a long history and various block-box methods have been proposed (Miller & Shaw, 1996; Deb & Saha, 2010; Kroese et al., 2006). In reinforcement learning, one of the most popular paradigms is population-based training with multi-objective optimization. Representative works include the family of qualitative diversity (QD) (Pugh et al., 2016) algorithms, such as MAP-Elites (Cully et al., 2015), which are based on genetic methods and assume a set of human-defined behavior characterizations, and policy-gradient methods (Ma et al., 2020; Tang et al., 2021), which typically assume a distribution of reward function is accessible. There are also some recent works that combine QD algorithms and policy gradient algorithms (Cideron et al., 2020; Nilsson & Cully, 2021). The DvD algorithm (Parker-Holder et al., 2020b) improves QD by optimizing *population diversity (PD)*, a KL-divergence-based diversity metric, without the need of hand-designed behavior characterizations. Similarly, Lupu et al. (2021) proposes to maximize *trajectory diversity*, i.e., the approximated Jensen-Shannon divergence with action-discounting kernel, to train a diversified population.

There are also works aiming to learn policies iteratively. PSRO (Lanctot et al., 2017) focuses on learning Nash equilibrium strategies in zero-sum games by maintaining a strategy oracle and repeatedly adding best responses to it. Various improvements have been made upon PSRO by using different metrics to promote diverse oracle strategies (Liu et al., 2021; Nieves et al., 2021). Hong et al. (2018) utilizes the KL-divergence between the current policy and a past policy version as an exploration bonus, while we are maximizing the diversity w.r.t a fixed set of reference policies, which is more stable and will not incur a cyclic training process. Diversity-Inducing Policy Gradient (DIPG) (Masood & Doshi-Velez, 2019) utilizes maximum mean discrepancy (MMD) between policies as a soft learning objective to iteratively find novel policies. By contrast, our method utilizes a filtering-based objective via reward switching to strictly enforce all the diversity constraints. Sun et al. (2020) adopts a conceptually similar objective by early terminating episodes that do not incur sufficient novelty. However, Sun et al. (2020) does not leverage any exploration technique for those rejected samples and may easily suffer from low sample efficiency in challenging RL tasks we consider in this paper. There is another concurrent work with an orthogonal focus, which directly optimizes diversity with reward constraints (Zahavy et al., 2021). We remark that enforcing a reward constraint can be problematic in multi-agent scenarios where different Nash Equilibrium can have substantially different pay-offs. In addition, the ridge rider algorithm (Parker-Holder et al., 2020a) proposes to follow the eigenvectors of the Hessian matrix to discover diverse local optima with theoretical guarantees, but Hessian estimates can be extremely inaccurate in complex RL problems.

Another stream of work uses unsupervised RL to discover diverse skills without the use of environment rewards, such as DIYAN (Eysenbach et al., 2019) and DDLUS (Hartikainen et al., 2020). However, ignoring the reward signal can substantially limit the capability of discovering strategic behaviors. SMERL (Kumar et al., 2020) augments DIYAN with extrinsic rewards to induce diverse solutions for robust generalization. These methods primarily focus on learning low-level locomotion while we tackle a much harder problem of discovering strategically and visually different policies.

Finally, our algorithm is also conceptually related to exploration methods (Zheng et al., 2018; Burda et al., 2019; Simmons-Edler et al., 2019), since it can even bypass inescapable local optima in challenging RL environments. Empirical comparisons can be found in Section 4. However, we emphasize that our paper tackles a much more challenging problem than standard RL exploration: we aim to discover as many *distinct local optima* as possible. That is, even if the global optimal solution is discovered, we still want to continuously seek for sufficiently distinct local-optimum strategies. We remark that such an objective is particularly important for multi-agent games where finding all the Nash equilibria can be necessary for analyzing rational multi-agent behaviors (Tang et al., 2021).

3 Method

3.1 Preliminary

We consider environments that can be modeled as a Markov decision process (MDP) (Puterman, 1994) $M = (\mathcal{S}, \mathcal{A}, R, P, \gamma)$ where $\mathcal{S}$ and $\mathcal{A}$ are the state and action space respectively, $R(s,a)$ is the reward function, $P(s'|s,a)$ is the transition dynamics and γ is the discount factor. We consider a stochastic policy π_θ paramterized by θ. Reinforcement learning optimizes the policy w.r.t. the expected return $J(\pi) = \mathbb{E}_{\tau\sim\pi}[\sum_t \gamma^t r_t]$ over the sampled trajectories from π, where a trajectory τ denotes a sequence of state-action-reward triplets, i.e., $\tau = \{(s_t, a_t, r_t)\}$. Note that this formulation can be naturally applied to multi-agent scenarios with homogeneous agents with shared state and action space, where learning a shared policy for all the agents will be sufficient.

Rather than learning a single solution for $J(\theta)$, we aim to discover a diverse set of M policies, i.e, $\{\pi_{\theta^k} | 1 \le k \le M\}$, such that all of these polices are locally optimized under $J(\theta)$ and mutually distinct w.r.t. some distance measure $D(\pi_{\theta^i}, \pi_{\theta^j})$, i.e.,

$$\max_{\theta^k} J(\theta^k) \ \ \forall 1 \le k \le M, \quad \text{subject to } \ D(\pi_{\theta^i}, \pi_{\theta^j}) \ge \delta, \quad \forall 1 \le i < j \le M. \tag{1}$$

Here $D(\cdot,\cdot)$ measures how different two policies are and δ is the novelty threshold. For conciseness, in the following content, we omit θ and use π_k to denote the policy with parameter θ^k.

3.2 Iterative constrained policy optimization

Directly solving Eq. (1) suggests a population-based training paradigm, which requires a non-trivial optimization technique for the pairwise constraints and typically needs a large population size M. Herein, we adopt an iterative process to discover novel policies: in the k-th iteration, we optimize a single policy π_k with the constraint that π_k is sufficiently distinct from previously discovered policies $\pi_1, \dots, \pi_{k-1}$. Here, the term "iteration" is used to denote the process of learning a new policy. Formally, we solve the following iterative constrained optimization problem for iteration $1 \le k \le M$:

$$\theta_k = \arg\max_\theta J(\theta), \quad \text{subject to } \ D(\pi_\theta, \pi_j) \ge \delta, \quad \forall 1 \le j < k. \tag{2}$$

Eq. (2) reduces the population-based objective to a standard constrained optimization problem for a single policy, which is much easier to solve. Such an iterative procedure does not require a large population size M as is typically necessary in population-based methods. And, in practice, only a few iterations could result in a sufficiently diverse collection of policies. We remark that, in theory, directly solving the constraint problem in Eq. (2) may lead to a solution that is not a local optimum w.r.t. the unconstrained objective $J(\theta)$. It is because a solution in Eq. (2) can be located on the boundary of the constraint space (i.e., $D(\pi_\theta, \pi_j) = \delta$), which is undesirable according to our original goal. However, this issue can be often alleviated by properly setting the novelty threshold δ.

The natural choice for measuring the policy difference is KL divergence, as done in the trust-region constraint (Schulman et al., 2015; 2017). However, in our setting where the difference between policies should be maximized, using KL as the diversity measure would inherently encourage learning a policy with small entropy, which is typically undesirable in RL problems (see App. F for a detailed derivation). Therefore, we adopt the accumulative cross-entropy as our diversity measure, i.e.,

$$D(\pi_i, \pi_j) := \mathcal{H}(\pi_i, \pi_j) = \mathbb{E}_{\tau \sim \pi_i}\left[-\sum_t \log \pi_j(a_t \mid s_t)\right] \quad (3)$$

3.3 Trajectory Filtering for enforcing diversity constraints

A popular approach to solve the constrained optimization problem in Eq. (2) is to use Lagrangian multipliers to convert the constraints to penalties in the learning objective. Formally, let $\beta_1, \dots, \beta_{k-1}$ be a set of hyperparameters, the soft objective for Eq. (2) is defined by

$$J_{\text{soft}}(\theta) := J(\pi_\theta) + \sum_{j=1}^{k-1} \beta_j D(\pi_\theta, \pi_j). \quad (4)$$

Such a soft objective substantially simplifies optimization and is widely adopted in RL applications. However, in our setting, since cross-entropy is a particularly dense function, including the diversity bonus as part of the objective may largely change the reward landscape of the original RL problem, which could make the final solution diverge from a locally optimal solution w.r.t $J(\theta)$. Therefore, it is often necessary to anneal the Lagrangian multipliers β_j, which is particularly challenging in our setting with a large number of reference policies. Moreover, since $D(\pi_\theta, \pi_j)$ is estimated over the trajectory samples, it introduces substantially high variance to the learning objective, which becomes even more severe as more policies are discovered.

Consequently, we propose a *Trajectory Filtering* objective to alleviate the issues of the soft objective. Let's use $\text{NLL}(\tau;\pi)$ to denote the negative log-likelihood of a trajectory τ w.r.t. a policy π, i.e., $\text{NLL}(\tau;\pi) = -\sum_{(s_t,a_t)\sim\tau} \log \pi(a_t|s_t)$. We apply rejection sampling over the sampled *trajectories* $\tau \sim \pi_\theta$ such that we train on those trajectories satisfying *all* the constraints, i.e., $\text{NLL}(\tau;\pi_j) \geq \delta$ for each reference policy π_j. Formally, for each sampled trajectory τ, we define a filtering function $\phi(\tau)$, which indicates whether we want to reject the sample τ, and use $\mathbb{I}[\cdot]$ to denote the indicator function, and then the trajectory filtering objective $J_{\text{filter}}(\theta)$ can be expressed as

$$J_{\text{filter}}(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\left[\phi(\tau)\sum_t \gamma^t r_t\right], \quad \text{where} \;\; \phi(\tau) := \prod_{j=1}^{k-1} \mathbb{I}[\text{NLL}(\tau;\pi_j) \geq \delta]. \quad (5)$$

We call the objective in Eq. (5) a *filtering* objective. We show in App. G that solving Eq. (5) is equivalent to solving Eq. (2) with an even stronger diversity constraint. In addition, we also remark that trajectory filtering shares a conceptually similar motivation with the clipping term in Proximal Policy Optimization (Schulman et al., 2017).

3.4 Intrinsic rewards for diversity exploration

The main issue in Eq. (5) is that trajectory filtering may reject a significant number of trajectories, especially in the early stage of policy learning since the policy is typically initialized to the a random policy. Hence, it is often the case that most of the data in a batch are abandoned, which leads to a severe wasting of samples and may even break learning due to the lack of feasible trajectories.

Can we make use of those rejected trajectories? We propose to additionally apply a novelty-driven objective on those rejected samples. Formally, we use $\phi_j(\tau)$ to denote whether τ violates the constraint of π_j, i.e., $\phi_j(\tau) = \mathbb{I}[\text{NLL}(\tau, \pi_j) \geq \delta]$. Then we have the following *switching* objective:

$$J_{\text{switch}} = \mathbb{E}_{\tau \sim \pi_\theta} \left[\phi(\tau) \sum_t \gamma^t r_t + \lambda \sum_j \left(1 - \phi_j(\tau)\right) \text{NLL}(\tau, \pi_j) \right] \tag{6}$$

The above objective simultaneously maximizes the extrinsic return on accepted trajectories and the cross-entropy on rejected trajectories. It can be proved that solving Eq. (6) is also equivalent to solving Eq. (2) with a stronger diversity constraint (see App. G).

Furthermore, Eq. (6) can be also interpreted as introducing additional cross-entropy intrinsic rewards on rejected trajectories (i.e., $\phi_j(\tau) = 0$). More specifically, given $\text{NLL}(\tau; \pi) = -\sum_{(s_t, a_t) \in \tau} \log \pi(a_t | s_t)$, an intrinsic reward $r^{\text{int}}(s_t, a_t; \pi_j) = -\log \pi_j(a_t \mid s_t)$ is applied to each state-action pair (s_t, a_t) from every rejected trajectory τ. Conceptually, this suggests an even more general paradigm for diversity exploration: we can optimize extrinsic rewards on accepted trajectories while utilizing novelty-driven intrinsic rewards on rejected trajectories for more effective exploration, i.e., by encouraging the learning policy π_θ to be distinct from a reference policy π_j.

Hence, we propose two different types of intrinsic rewards to promote diversity exploration: one is *likelihood-based*, which directly follows Eq. (6) and focuses more on behavior novelty, and the other is *reward-prediction-based*, which focuses more on achieving novel states and reward signals.

Behavior-driven exploration. The behavior-driven intrinsic reward $r^{\text{int}}_{\mathbf{B}}$ is defined by

$$r^{\text{int}}_{\mathbf{B}}(a, s; \pi_j) = -\log \pi_j(a \mid s). \tag{7}$$

$r^{\text{int}}_{\mathbf{B}}$ encourages the learning policy to output different actions from those reference policies and therefore to be more likely to be accepted. Note that $r^{\text{int}}_{\mathbf{B}}$ can be directly interpreted as the Lagrangian penalty utilized in the soft objective $J_{\text{soft}}(\theta)$.

Reward-driven exploration. A possible limitation of behavior-driven exploration is that it may overly focus on visually indistinguishable action changes rather than high-level strategies. Note that in RL problems with diverse reward signals, it is usually preferred to discover policies that can achieve different types of rewards (Simmons-Edler et al., 2020; Wang* et al., 2020). Inspired by the curiosity-driven exploration method (Pathak et al., 2017), we adopt a model-based approach for predicting novel reward signals. In particular, after obtaining each reference policy π_j, we learn a reward prediction function $f(s, a; \psi_j)$ trained by minimizing the expected MSE loss $\mathcal{L}(\psi_j) = \mathbb{E}_{\tau \sim \pi_j, t}\left[|f(s_t, a_t; \psi_j) - r_t|^2 \right]$ over the trajectories generated by π_j. The reward prediction function $f(s, a; \psi_j)$ is expected to predict the extrinsic environment reward more accurately on state-action pairs that are more frequently visited by π_j and less accurately on rarely visited pairs. To encourage policy exploration, we adopt the reward prediction error as our reward-driven intrinsic reward $r^{\text{int}}_{\mathbf{R}}(a, s; \pi_j)$. Formally, given the transition triplet (s_t, a_t, r_t), $r^{\text{int}}_{\mathbf{R}}(a, s; \pi_j)$ is defined by

$$r^{\text{int}}_{\mathbf{R}}(s_t, a_t; \pi_j) = |f(s_t, a_t; \psi_j) - r_t|^2. \tag{8}$$

We remark that reward-driven exploration can be also interpreted as approximately maximizing the f-divergence of joint state occupancy measure between policies (Liu et al., 2021). By combining these two intrinsic rewards together, we approximately maximize the divergence of *both actions and states* between policies to effectively promote diversity. By default, we use behavior-driven intrinsic reward for computational simplicity and optionally augment it with reward-driven intrinsic reward in more challenging scenarios (see examples in Section 4.2).

3.5 Reward-Switching Policy Optimization

We define the RSPO function r_t^{RSPO} by

$$r_t^{\text{RSPO}} = \phi(\tau) r_t + \lambda \sum_j (1 - \phi_j(\tau)) r^{\text{int}}(a_t, s_t; \pi_j), \tag{9}$$

where λ is a scaling hyper-parameter. Note that extrinsic rewards and intrinsic rewards are mutually exclusive, i.e., a trajectory τ may be either included in $J_{\text{filtering}}$ or be rejected to produce exploration bonuses. Conceptually, our method is adaptively "switching" between extrinsic and intrinsic rewards during policy gradients, which is so-called *Reward-Switching Policy Optimization (RSPO)*. We also remark that the intrinsic reward will constantly push the learning policy towards the feasible policy space and the optimization objective will eventually converge to $J(\theta)$ when no trajectory is rejected.

In addition to the aforementioned RSPO algorithm, we also introduce two implementation enhancements for better empirical performances, especially in some performance-sensitive scenarios.

Automatic threshold selection. We provide an empirical way of adjusting δ. In some environments, δ is sensitive to each reference policy. Instead of tuning δ for each reference policy, we choose its corresponding threshold by $\delta_j = \alpha \cdot D(\pi^{\text{rnd}}, \pi_j)$, where π^{rnd} is a fully random policy and α is a task-specific hyperparameter. We remark that α is a constant parameter across training iterations and is much easier to choose than manually tuning δ, which requires subtle variation throughout multiple training iterations. We use automatic threshold selection by default. Detailed values of α and the methodology of tuning α can be found in App. D.1 and App. B.3 respectively.

Smoothed-switching for intrinsic rewards. Intrinsic rewards have multiple switching indicators, i.e., $\phi_1, \ldots, \phi_{k-1}$. Moreover, for different trajectories, different subsets of indicators will be turned on and off, which may result in a varying scale of intrinsic rewards and hurt training stability. Therefore, in some constraint-sensitive cases, we propose a smoothed switching mechanism which could further improve practical performance. Specifically, we maintain a running average $\tilde{\phi}_j$ over all the sampled trajectories for each indicator $\phi_j(\tau)$, and use these smoothed indicators to compute intrinsic rewards defined in Eq. (9). Smoothed-switching empirically improves training stability when a large number of reference policies exist, such as in stag-hunt games (see Section 4.2).

4 Experiments

To illustrate that our method can be applied to *general* RL applications, we experiment on 4 domains that feature multi-modality of solutions, including a single-agent navigation problem in the particle-world (Mordatch & Abbeel, 2018), 2-agent Markov stag-hunt games (Tang et al., 2021), continuous control in MuJoCo (Todorov et al., 2012), and the StarCraft II Multi-Agent Challenge (SMAC) (Vinyals et al., 2017; Rashid et al., 2019). In particle world and stag-hunt games, all the local optima can be precisely calculated, so we can quantitatively evaluate the effectiveness of different algorithms by measuring how many distinct strategy modes are discovered. In MuJoCo control and SMAC, we qualitatively demonstrate that our method can discover a large collection of visually distinguishable strategies. Notably, we primarily present results from purely RL-based methods which do not require prior knowledge over possible local optima for a fair comparison. We also remark that when a precise feature descriptor of local optima is feasible, it is also possible to apply evolutionary methods (Nilsson & Cully, 2021) to a subset of the scenarios we considered. For readers of further interest, a thorough study with discussions can be found in App. B.4.

Our implementation is based on PPO (Schulman et al., 2017) on a desktop machine with one CPU and one NVIDIA RTX3090 GPU. All the algorithms are run for the same number of total environment steps and the same number of iterations (or population size). More details can be found in appendix.

4.1 Single-agent particle-world environment

We consider a sparse-reward navigation scenario called *4-Goals* (Fig. 1). The agent starts from the center and will receive a reward when reaching a landmark. We set up 3 difficulty levels. In the easy mode, the landmark locations are fixed. In the medium mode, the landmarks are randomly placed. In the hard mode, landmarks are not only placed randomly but also have different sizes and rewards. Specifically, the sizes and rewards of each landmark are $2\times$, $1\times$, $0.5\times$, $0.25\times$ and $1\times$, $1.1\times$, $1.2\times$, $1.3\times$ of the normal one respectively. We remark that in the hard mode, the landmark size decreases at an exponential rate while the reward gain is only marginal, making it exponentially harder to discover policies towards those smaller landmarks. We compare RSPO with several baselines, including PPO with restarts (PG), Diversity-Inducing Policy Gradient (DIPG) (Masood & Doshi-Velez, 2019), population-based training with cross-entropy objective (PBT-CE), DvD (Parker-Holder et al., 2020b), SMERL (Kumar et al., 2020), and Random Network Distillation (RND) (Burda et al., 2019). RND is designed to explore the policy with the highest reward, so we only evaluate RND in the hard mode.

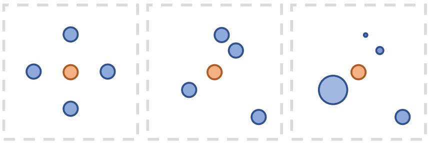
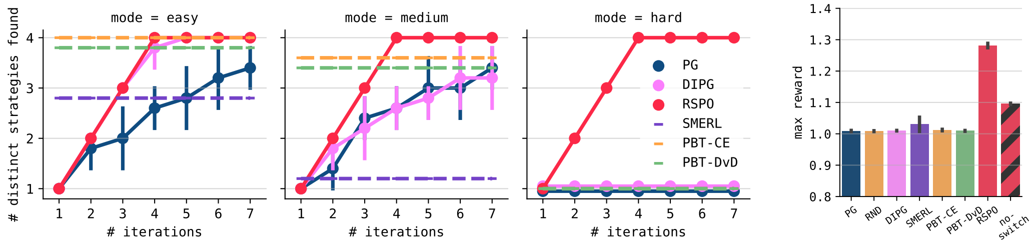

(a) *Easy* (b) *Medium* (c) *Hard*

Figure 1: The agent (orange) and landmarks (blue) in *4-Goals*.

The number of distinct local optima discovered by different methods is presented in Fig. 2a. RSPO consistently discovers all the 4 modes within 4 iterations even without the use of any intrinsic rewards over rejected trajectories (i.e., $r_t^{\text{int}} = 0$). DIPG finds 4 strategies in 4 out of the 5 runs in the easy mode but performs no better than PG in the two harder modes. Fig. 2b shows the highest expected return achieved over the policy population in the hard mode. RSPO is the only algorithm that



(a) Mean number of distinct strategies found on *4-Goals*.

(b) Rewards on *hard* mode.

Figure 2: Experiment results on *4-Goals* for $M = 7$ iterations averaged over 5 random seeds. Error bars are 95% confidence intervals.

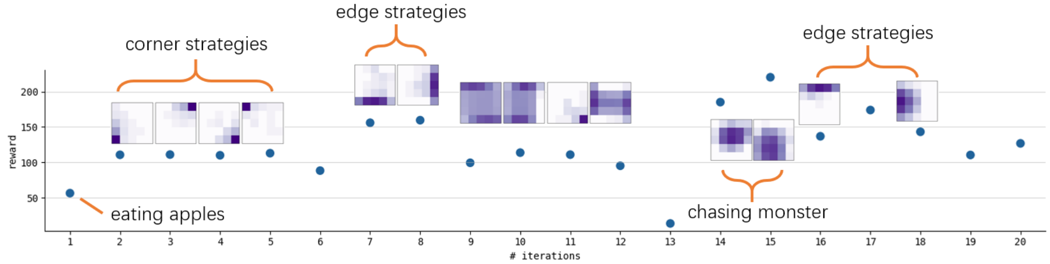


Figure 4: Different strategies found in a run of 20 iterations diversity setting RSPO in *Monster-Hunt*. We plot the heatmap of the two agents' meeting point to indicate the type of the found strategies.

successfully learns the optimal policy towards the smallest ball. We also report the performance of RSPO optimizing the soft objective in Eq. (4) with the default behavior-driven intrinsic reward $r_{\mathbf{B}}^{\text{int}}$ (*no-switch*), which was able to discover the policy towards the second largest landmark. Comparing *no-switch* with $r_t^{int} = 0$, we could conclude that the filtering-based objective can be critical for RSPO to discover sufficiently different modes. We also remark that the *no-switch* variant only differs from DIPG by the used diversity metric. This suggests that in environments with high state variance (e.g. landmarks with random sizes and positions), state-based diversity metric may be less effective.

4.2 2-Agent Markov Stag-Hunt Games

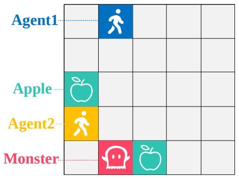


Figure 3: *Monster-Hunt*

We further show the effectiveness of RSPO on two grid-world stag-hunt games developed in Tang et al. (2021), *Monster-Hunt* and *Escalation*, both of which have very distinct Nash Equilibria (NEs) for self-play RL methods to converge to. Moreover, the optimal NE with the highest rewards for both agents in these games are *risky cooperation*, i.e., a big penalty will be given to an agent if the *other* agent stops cooperation. This makes most self-play RL algorithms converge to the safe non-cooperative NE strategies with lower rewards. It has been shown that *none* of the state-of-the-art exploration methods can discover the global optimal solution without knowing the underlying reward structure (Tang et al., 2021). We remark that since there are enormous NEs in these environments as shown in Fig. 4 and 7b, population-based methods (PBT) require a significantly large population size for meaningful performances, which is computationally too expensive to run. Therefore, we do not include the results of PBT baselines. We also apply the *smoothed-switching* heuristic for RSPO in this domain. Environment details can be found in App. C.2.

The *Monster-Hunt* game. The *Monster-Hunt* game (Fig. 3) contains a monster and two apples. When a single agent meets the monster, it gets a penalty of -2. When both agents meet the monster at the same time, they "catch" the monster and both get a bonus of 5. When a player meets an apple, it gets a bonus of 2. The optimal strategy, i.e., both agents move towards the monster, is a risky cooperative NE since an agent will receive a penalty if the other agent deceives. The non-cooperative NE for eating apples is a safe NE and easy to discover but has lower rewards.

We adopt both behavior-driven and reward-driven intrinsic rewards in RSPO to tackle *Monster-Hunt*. Fig. 4 illustrates all the discovered strategies by RSPO over 20 iterations, which covers a wide range of human-interpretable strategies, including the non-cooperative apple-eating strategy as well as the

Table 1: Types of strategies discovered by each methods in *Monster-Hunt* over 20 iterations.

		Apple	Corner	Edge	Chase
Ablation	RSPO	✓	✓	✓	✓
	- No switch	✓	✓		
	- No r^{int}	✓			
	- $r^{\text{int}}_{\mathbf{B}}$ only	✓	✓	✓	
Baseline	RPG	✓	✓		✓
	MAVEN	✓	✓		
	PG/DIPG/RND	✓			

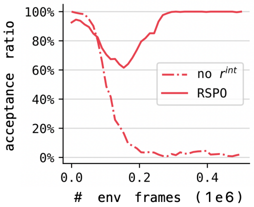


Figure 5: Sample acceptance ratio when learning a policy distinct from *Apple* NE. The intrinsic reward is critical.

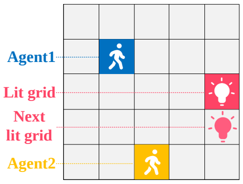
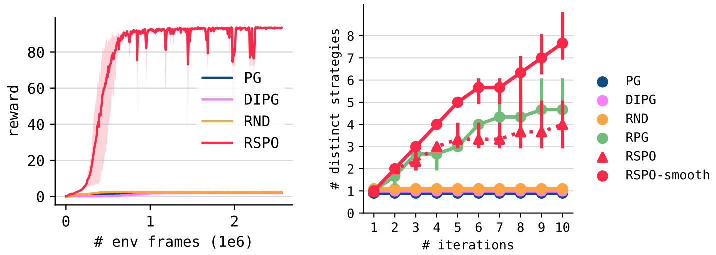


Figure 6: *Escalation*: two agents need to keep stepping on the light simultaneously.



(a) Reward in iteration 2.

(b) Number of distinct strategies found.

Figure 7: Results on *Escalation* averaged over 3 random seeds. Shaded area and error bars are 95% confident intervals.

optimal strategy, where both two agents stay together and chase the monster actively. By visualizing the heatmap of where both agents meet, we observe a surprisingly diverse sub-optimal cooperation NEs, where both agents move to a corner or an edge simultaneously, keep staying there, and wait for the monster coming. We remark that due to the existence of such a great number of passive waiting strategies, which all have similar accumulative environment rewards and states, it becomes critical to include the reward-driven intrinsic reward to quickly bypass them and discover the optimal solution.

We perform ablation studies on RSPO by turning off reward switching (*No Switch*) or intrinsic reward (*No* r^{int}) or only using the behavior-driven intrinsic reward ($r^{int}_{\mathbf{B}}$ *only*), and evaluate the performances of many baseline methods, including vanilla PG with restarts (PG), DIPG, RND, a popular multi-agent exploration method MAVEN (Mahajan et al., 2019) and reward-randomized policy gradient (RPG) (Tang et al., 2021). We summarize the categories of discovered strategies by all these baselines in Table 1. *Apple* denotes the non-cooperative apple-eating NE; *Chase* denotes the optimal NE where both agents actively chase the monster; *Corner* and *Edge* denote the sub-optimal cooperative NE where both agents passively wait for the monster at a corner or an edge respectively. Regarding the baselines, PG, DIPG, and RND never discover any strategy beyond the non-cooperative *Apple* NE. For RPG, even using the domain knowledge to change the reward structure of the game, it never discovers the *Edge* NE. Regarding the RSPO variants, both reward switching and intrinsic rewards are necessary. Fig. 5 shows that when the intrinsic reward is turned off, the proportion of accepted trajectories per batch stays low throughout training. This implies that the learning policy failed to escape the infeasible subspace. Besides, as shown in Table 1, using behavior-driven exploration alone fails to discover the optimal NE, which suggests the necessity of reward-driven exploration to maximize the divergence of both states and actions in problems with massive equivalent local optima.

The *Escalation* game. *Escalation* (Fig. 6) requires the two players to interact with a static *light*. When *both* players step on the light simultaneously, they both receive a bonus of 1. Then the light moves to a random adjacent grid. The game continues only if both players choose to follow the light. If only one player steps on the light, it receives a penalty of $-0.9L$, where L is the number of previous cooperation steps. For each integer L, there is a corresponding NE where both players follow the light for L steps then simultaneously stop cooperation. We run RSPO with both diversity-driven intrinsic rewards and compare it with PG, DIPG, RND and RPG. Except for RPG, none of the baseline methods discover any cooperative NEs while RSPO directly learns the optimal cooperative NE (i.e., always cooperate) in the second iteration as shown in Fig. 7a. We also measure the total number of discovered NEs by different methods over 10 iterations in Fig. 7b. Due to the existence of many spiky local optima, the *smoothed-switching* technique can be crucial here to stabilize the

Table 2: *Population Diversity* scores in *MuJoCo*.

	H.-Cheetah	*Hopper*	*Walker2d*	*Humanoid*
PG	0.033 (0.013)	0.418 (0.125)	0.188 (0.079)	0.965 (0.006)
DIPG	0.051 (0.009)	0.468 (0.054)	0.179 (0.056)	0.996 (0.000)
PBT-CE	0.160 (0.078)	0.620 (0.294)	0.512 (0.032)	**0.999 (0.000)**
DvD	0.275 (0.164)	0.656 (0.523)	0.542 (0.103)	**1.000 (0.000)**
SMERL	0.003 (0.002)	0.674 (0.389)	0.669 (0.152)	N/A
RSPO	**0.359 (0.058)**	**0.989 (0.009)**	**0.955 (0.039)**	**0.999 (0.000)**

Table 3: Number of visually distinct policies over 4 iterations in SMAC.

	2c64zg	*2m1z*
PG	2	2
DIPG	2	2
PBT-CE	2	3
TrajDiv	3	1
RSPO	**4**	**4**

training process. We remark that even without the *smoothed-switching* technique, RSPO achieves comparable performance with RPG — note that RPG requires a known reward function while RSPO does not assume any environment-specific domain knowledge.

4.3 Continuous control in *MuJoCo*

We evaluate RSPO in the continuous control domain, including *Half-Cheetah*, *Hopper*, *Walker2d* and *Humanoid*, and compare it with baseline methods including PG, DIPG, DvD (Parker-Holder et al., 2020b), SMERL (Kumar et al., 2020) and population-based training with our cross-entropy objective (PBT-CE). All the methods are run over 5 iterations (a population size of 5, or have a latent dimension of 5) across 3 seeds. We adopt *Population Diversity*, a determinant-based diversity criterion proposed in Parker-Holder et al. (2020b), to evaluate the diversity of derived policies by different methods. Results are summarized in Table 2, where RSPO achieves comparable performance in *Hopper* and *Humanoid* and substantially outperforms all the baselines in *Half-Cheetah* and *Walker2d*. We remark that even with the same intrinsic reward, population-based training (PBT-CE) cannot discover sufficiently novel policies compared with iterative learning (RSPO). In *Humanoid*, SMERL achieves substantially lower return than other baseline methods and we don't report the population diversity score (more details can be found in App. B.6). We also visualize some interesting emergent behaviors RSPO discovered for *Half-Cheetah* and *Hopper* in App. B.1, where different strategy modes discovered by RSPO are visually distinguishable while baselines methods often converge to very similar behaviors despite of the non-zero diversity score.

4.4 Stacraft Multi-Agent Challenge

We further apply RSPO to the StarCraft II Multi-Agent Challenge (SMAC) (Rashid et al., 2019), which is substantially more difficult due to partial observability, long horizon, and complex state/action space. We conduct experiments on 2 maps, an easy map *2m_vs_1z* and a hard map *2c_vs_64zg*, both of which have heterogeneous unit types leading to a multi-modal solution space. Baseline methods include PG, DIPG, PBT-CE and trajectory diversity (TrajDiv) (Lupu et al., 2021). SMERL algorithm and DvD algorithm are not included because they were originally designed for continuous control domain and not suitable for SMAC (see App. B.6 and App. B.2.2). Instead, we include the TrajDiv algorithm (Lupu et al., 2021) as an additional baseline, which was designed for cooperative multi-agent games. We compare the number of visually distinct policies by training a population of 4 (or for 4 iterations), as shown in Table 3. While PBT-based algorithms tend to discover policies with slight distinctions, RSPO can effectively discover different *winning* strategies demonstrating intelligent behaviors in just a few iterations consistently across repetitions. We remark that there may not exist an appropriate quantitative diversity metric for such a sophisticated MARL game in the existing literature (see App. B.2.2). Visualizations and discussions can be found in App. B.1.

5 Conclusion

We propose *Reward-Switching Policy Optimization (RSPO)*, a simple, generic, and effective iterative learning algorithm that can continuously discover novel strategies. RSPO tackles a novelty-constrained optimization problem via adaptive switching between extrinsic and intrinsic rewards used for policy learning. Empirically, RSPO can successfully tackle a wide range of challenging RL domains under both single-agent and multi-agent settings. We leave further theoretical justifications and sample efficiency improvements as future work.

REFERENCES

Bowen Baker, Ingmar Kanitscheider, Todor Markov, Yi Wu, Glenn Powell, Bob McGrew, and Igor Mordatch. Emergent tool use from multi-agent autocurricula. In *International Conference on Learning Representations*, 2020.

Nolan Bard, Jakob N Foerster, Sarath Chandar, Neil Burch, Marc Lanctot, H Francis Song, Emilio Parisotto, Vincent Dumoulin, Subhodeep Moitra, Edward Hughes, et al. The hanabi challenge: A new frontier for ai research. *Artificial Intelligence*, 280:103216, 2020.

Yuri Burda, Harrison Edwards, A. Storkey, and Oleg Klimov. Exploration by random network distillation. *ICLR*, 2019.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

Geoffrey Cideron, Thomas Pierrot, Nicolas Perrin, Karim Beguir, and Olivier Sigaud. Qd-rl: Efficient mixing of quality and diversity in reinforcement learning. *arXiv preprint arXiv:2006.08505*, 2020.

Antoine Cully, Jeff Clune, Danesh Tarapore, and Jean-Baptiste Mouret. Robots that can adapt like animals. *Nature*, 521(7553):503–507, 2015.

K. Deb and Amit Saha. Finding multiple solutions for multimodal optimization problems using a multi-objective evolutionary approach. In *GECCO '10*, 2010.

Benjamin Eysenbach, A. Gupta, J. Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. *ICLR*, 2019.

Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, pp. 1587–1596. PMLR, 2018.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. PMLR, 2018.

Kristian Hartikainen, Xinyang Geng, T. Haarnoja, and Sergey Levine. Dynamical distance learning for semi-supervised and unsupervised skill discovery. *ICLR*, 2020.

Zhang-Wei Hong, Tzu-Yun Shann, Shih-Yang Su, Y. Chang, and Chun-Yi Lee. Diversity-driven exploration strategy for deep reinforcement learning. 2018.

Max Jaderberg, Wojciech M Czarnecki, Iain Dunning, Luke Marris, Guy Lever, Antonio Garcia Castaneda, Charles Beattie, Neil C Rabinowitz, Ari S Morcos, Avraham Ruderman, et al. Human-level performance in 3d multiplayer games with population-based reinforcement learning. *Science*, 364(6443):859–865, 2019.

Dirk P. Kroese, S. Porotsky, and R. Rubinstein. The cross-entropy method for continuous multi-extremal optimization. *Methodology and Computing in Applied Probability*, 8:383–407, 2006.

Saurabh Kumar, Aviral Kumar, Sergey Levine, and Chelsea Finn. One solution is not all you need: Few-shot extrapolation via structured maxent rl. *Advances in Neural Information Processing Systems*, 33, 2020.

Marc Lanctot, V. Zambaldi, A. Gruslys, A. Lazaridou, K. Tuyls, J. Pérolat, D. Silver, and T. Graepel. A unified game-theoretic approach to multiagent reinforcement learning. In *NIPS*, 2017.

Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. Deep reinforcement learning for dialogue generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1192–1202, 2016.

Siqi Liu, G. Lever, J. Merel, S. Tunyasuvunakool, N. Heess, and T. Graepel. Emergent coordination through competition. 2019.

Xiangyu Liu, Hangtian Jia, Ying Wen, Yaodong Yang, Yujing Hu, Yingfeng Chen, Changjie Fan, and Zhipeng Hu. Unifying behavioral and response diversity for open-ended learning in zero-sum games. *arXiv preprint arXiv:2106.04958*, 2021.

Andrei Lupu, Brandon Cui, Hengyuan Hu, and Jakob Foerster. Trajectory diversity for zero-shot coordination. In *International Conference on Machine Learning*, pp. 7204–7213. PMLR, 2021.

Pingchuan Ma, Tao Du, and Wojciech Matusik. Efficient continuous pareto exploration in multi-task learning. In *International Conference on Machine Learning*, pp. 6522–6531. PMLR, 2020.

Tengyu Ma. Why do local methods solve nonconvex problems? *Beyond the Worst-Case Analysis of Algorithms*, pp. 465, 2020.

Anuj Mahajan, Tabish Rashid, Mikayel Samvelyan, and S. Whiteson. Maven: Multi-agent variational exploration. In *NeurIPS*, 2019.

M. A. Masood and Finale Doshi-Velez. Diversity-inducing policy gradient: Using maximum mean discrepancy to find a set of diverse policies. 2019.

B. Miller and Michael J. Shaw. Genetic algorithms with dynamic niche sharing for multimodal function optimization. *Proceedings of IEEE International Conference on Evolutionary Computation*, pp. 786–791, 1996.

Igor Mordatch and P. Abbeel. Emergence of grounded compositional language in multi-agent populations. In *AAAI*, 2018.

Nicolas Perez Nieves, Yaodong Yang, Oliver Slumbers, David Mguni, and Jun Wang. Modelling behavioural diversity for learning in open-ended games. *ICML*, 2021.

Olle Nilsson and Antoine Cully. Policy gradient assisted map-elites. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 866–875, 2021.

Shayegan Omidshafiei, Karl Tuyls, Wojciech M Czarnecki, Francisco C Santos, Mark Rowland, Jerome Connor, Daniel Hennes, Paul Muller, Julien Pérolat, Bart De Vylder, et al. Navigating the landscape of multiplayer games. *Nature communications*, 11(1):1–17, 2020.

Jack Parker-Holder, Luke Metz, Cinjon Resnick, H. Hu, A. Lerer, Alistair Letcher, Alexander Peysakhovich, Aldo Pacchiano, and Jakob Foerster. Ridge rider: Finding diverse solutions by following eigenvectors of the hessian. *NeurIPS*, 2020a.

Jack Parker-Holder, Aldo Pacchiano, Krzysztof Choromanski, and Stephen Roberts. Effective diversity in population-based reinforcement learning. *NeurIPS*, 2020b.

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International Conference on Machine Learning*, pp. 2778–2787. PMLR, 2017.

Tiago Pereira, Maryam Abbasi, Bernardete Ribeiro, and Joel P. Arrais. Diversity oriented deep reinforcement learning for targeted molecule generation. *Journal of Cheminformatics*, 13(1):21, Mar 2021. ISSN 1758-2946. doi: 10.1186/s13321-021-00498-z. URL https://doi.org/10.1186/s13321-021-00498-z.

Justin K. Pugh, L. B. Soros, and K. Stanley. Quality diversity: A new frontier for evolutionary computation. *Frontiers Robotics AI*, 3:40, 2016.

M. Puterman. Markov decision processes: Discrete stochastic dynamic programming. In *Wiley Series in Probability and Statistics*, 1994.

Tabish Rashid, Philip HS Torr, Gregory Farquhar, Chia-Man Hung, Tim GJ Rudner, Nantas Nardelli, Shimon Whiteson, Christian Schroeder de Witt, Jakob Foerster, and Mikayel Samvelyan. The Starcraft multi-agent challenge. volume 4, pp. 2186–2188. International Foundation for Autonomous Agents and Multiagent Systems, 2019.

John Schulman, Sergey Levine, P. Abbeel, Michael I. Jordan, and P. Moritz. Trust region policy optimization. *ICML*, 2015.

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

Riley Simmons-Edler, Ben Eisner, Daniel Yang, Anthony Bisulco, Eric Mitchell, Sebastian Seung, and Daniel Lee. Qxplore: Q-learning exploration by maximizing temporal difference error. 2019.

Riley Simmons-Edler, Ben Eisner, Daniel Yang, Anthony Bisulco, Eric Mitchell, Sebastian Seung, and Daniel Lee. Reward prediction error as an exploration objective in deep rl. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pp. 2816–2823, 7 2020.

Hao Sun, Zhenghao Peng, Bo Dai, Jian Guo, Dahua Lin, and Bolei Zhou. Novel policy seeking with constrained optimization. *arXiv preprint arXiv:2005.10696*, 2020.

Zhenggang Tang, C. Yu, Boyuan Chen, Huazhe Xu, Xiaolong Wang, Fei Fang, S. Du, Yu Wang, and Yi Wu. Discovering diverse multi-agent strategic behavior via reward randomization. *ICLR*, 2021.

E. Todorov, T. Erez, and Y. Tassa. Mujoco: A physics engine for model-based control. *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033, 2012.

Oriol Vinyals, Timo Ewalds, Sergey Bartunov, Petko Georgiev, Alexander Sasha Vezhnevets, Michelle Yeo, Alireza Makhzani, Heinrich Küttler, John Agapiou, Julian Schrittwieser, et al. Starcraft ii: A new challenge for reinforcement learning. *arXiv preprint arXiv:1708.04782*, 2017.

Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

Rui Wang, Joel Lehman, Jeff Clune, and Kenneth O Stanley. Poet: open-ended coevolution of environments and their optimized solutions. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 142–151, 2019.

Tonghan Wang*, Jianhao Wang*, Yi Wu, and Chongjie Zhang. Influence-based multi-agent exploration. In *International Conference on Learning Representations*, 2020.

Chao Yu, Akash Velu, Eugene Vinitsky, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of mappo in cooperative, multi-agent games. *arXiv preprint arXiv:2103.01955*, 2021.

Tom Zahavy, Brendan O'Donoghue, Andre Barreto, Volodymyr Mnih, Sebastian Flennerhag, and Satinder Singh. Discovering diverse nearly optimal policies with successor features. *arXiv preprint arXiv:2106.00669*, 2021.

Stephan Zheng, Alexander Trott, Sunil Srinivasa, Nikhil Naik, Melvin Gruesbeck, David C Parkes, and Richard Socher. The AI economist: Improving equality and productivity with AI-driven tax policies. *arXiv preprint arXiv:2004.13332*, 2020.

Zeyu Zheng, Junhyuk Oh, and Satinder Singh. On learning intrinsic rewards for policy gradient methods. *Advances in Neural Information Processing Systems*, 31:4644–4654, 2018.