DGPO: Discovering Multiple Strategies with Diversity-Guided Policy Optimization

Wentse Chen[1], **Shiyu Huang**[2], **Yuan Chiang**[3], **Tim Pearce**[4], **Wei-Wei Tu**[2], **Ting Chen**[3], **Jun Zhu**[3]
[1]Carnegie Mellon University, Pittsburgh, USA
[2]4Paradigm Inc., Beijing, China
[3]Tsinghua University, Beijing, China
[4]Microsoft Research, Cambridge, United Kingdom
wentsec@andrew.cmu.edu, huangsy1314@163.com, yjennice2001@gmail.com, tim.pearce@microsoft.com,
tuweiwei@4paradigm.com, tingchen@tsinghua.edu.cn, dcszj@tsinghua.edu.cn

Abstract

Most reinforcement learning algorithms seek a single optimal strategy that solves a given task. However, it can often be valuable to learn a diverse set of solutions, for instance, to make an agent's interaction with users more engaging, or improve the robustness of a policy to an unexpected perturbance. We propose Diversity-Guided Policy Optimization (DGPO), an on-policy algorithm that discovers multiple strategies for solving a given task. Unlike prior work, it achieves this with a shared policy network trained over a single run. Specifically, we design an intrinsic reward based on an information-theoretic diversity objective. Our final objective alternately constraints on the diversity of the strategies and on the extrinsic reward. We solve the constrained optimization problem by casting it as a probabilistic inference task and use policy iteration to maximize the derived lower bound. Experimental results show that our method efficiently discovers diverse strategies in a wide variety of reinforcement learning tasks. Compared to baseline methods, DGPO achieves comparable rewards, while discovering more diverse strategies, and often with better sample efficiency.

Introduction

Reinforcement Learning (RL) has pioneered breakthroughs in various domains ranging from video games (Vinyals et al. 2019; Berner et al. 2019; Huang et al. 2019a, 2021) to robotics (Raffin et al. 2018; Yu et al. 2021b). While its achievements are remarkable, RL is not devoid of challenges. A paramount issue is the innate pursuit of RL algorithms for a singular optimal solution, even when a myriad of equally viable strategies exists. This tunnel vision for optimization can inadvertently introduce weaknesses.

For instance, RL algorithms are known for "overfitting" tasks. By zeroing in on just one strategy, they often miss out on exploring a wealth of high-quality alternative solutions. This over-specialization renders the agent vulnerable to unpredictable environmental changes, as it lacks the robustness multiple strategies could have offered (Kumar et al. 2020). In competitive arenas, predictability can be an Achilles heel, with adversaries exploiting the agent's inflexibility. A diversified approach would obfuscate the agent's strategies, improving its competitive edge (Lanctot et al. 2017). Furthermore, in domains like dialogue systems, monotony can dull user interactions, whereas varied responses could significantly enhance user experience (Li et al. 2016; Gao et al. 2019; Pavel, Budulan, and Rebedea 2020; Xu et al. 2022; Chow et al. 2022).

For instance, RL algorithms are known for "overfitting" tasks. By zeroing in on just one strategy, they often miss out on exploring a wealth of high-quality alternative solutions. This over-specialization renders the agent vulnerable to unpredictable environmental changes, as it lacks the robustness multiple strategies could have offered (Kumar et al. 2020). In competitive arenas, predictability can be an Achilles heel, with adversaries exploiting the agent's inflexibility. A diversified approach would obfuscate the agent's strategies, improving its competitive edge (Lanctot et al. 2017). Furthermore, in domains like dialogue systems, monotony can dull user interactions, whereas varied responses could significantly enhance user experience (Li et al. 2016; Gao et al. 2019; Pavel, Budulan, and Rebedea 2020; Xu et al. 2022; Chow et al. 2022).

We identify two key scenarios where multiple strategies are beneficial: 1. The margin for error is inconsequential to the task's success. In such cases, agents can operate optimally without adhering strictly to the best strategy, enabling a spread of near-optimal strategies (Zahavy et al. 2021). 2. The task inherently allows multiple optimal solutions, such as a maze offering two equally efficient paths (Osa, Tangkaratt, and Sugiyama 2021; Zhou et al. 2022).

Crafting an algorithm that harnesses diverse high-reward solutions efficiently is intricate. With the Diversity-Guided Policy Optimization (DGPO) we propose, we aim to address several requisites for such an algorithm:

Strategy Representation: the diversity of a policy suite is evaluated in DGPO using a metric grounded on the mutual information between states and the latent variable, implemented through a learned discriminator.

Diversity Evaluation: The diversity of a policy suite is evaluated in DGPO using a metric grounded on the mutual information between states and the latent variable, implemented through a learned discriminator.

Diversity Exploration: DGPO embarks on exploration that encourages deviation from familiar strategies while

safeguarding performance. This is achieved using a constrained optimization method that harmonizes performance and diversity.

Sample Efficiency: Unlike predecessors like RSPO (Zhou et al. 2022) and RPG (Tang et al. 2021), which necessitated multiple networks and training phases, DGPO employs a shared network for concurrent learning of strategies, resulting in superior sample efficiency.

In encapsulation, this work makes three pivotal contributions: (1) We introduce a structured approach to discover diverse high-reward policies by framing it as two constrained optimization problems, coupled with tailored diversity rewards to guide the policy learning. (2) DGPO, a novel on-policy algorithm, is unveiled, designed to seamlessly uncover a diverse suite of high-quality strategies. (3) Our empirical evaluations elucidate that DGPO not only holds its own against benchmarks but frequently surpasses them in terms of diversity, performance, and sample efficiency.

Related Works

In this section, we provide an overview of prior research that encompasses two main aspects: the representation of reinforcement learning (RL) as a probabilistic graphical model (PGM), and the explicit integration of diversity learning with RL.

Reinforcement Learning as Probabilistic Graphical Model

PGM's have proven to be a useful way of framing the RL problem (Ziebart et al. 2008; Furmston and Barber 2010; Levine 2018). The soft actor-critic (SAC) algorithm (Haarnoja et al. 2018b) formalizes RL as probabilistic inference and maximizes an evidence lower bound by adding an entropy term to the training objective, encouraging exploration. PGM's also serve as useful tools for studying partially observable Markov decision processes (POMDP's) (Igl et al. 2018; Huang et al. 2019b; Lee et al. 2019). Haarnoja et al. (Haarnoja et al. 2018a) use PGM's to construct a hierarchical reinforcement learning algorithm. Hausman et al. (Hausman et al. 2018) optimize a multi-task policy through a variational bound, allowing for the discovery of multiple solutions with a minimum number of distinct skills. In this work, we modify the PGM of a Markov decision process (MDP) by introducing a latent variable to induce diversity into the MDP. We then derive the evidence lower bound of the new PGM, allowing us to construct a novel RL algorithm.

Diversity in Reinforcement Learning

Achieving diversity has been studied in various contexts in RL (Mouret and Doncieux 2009; Mohamed and Rezende 2015; Eysenbach et al. 2018; Osa, Tangkaratt, and Sugiyama 2021; Derek and Isola 2021). Eysenbach et al. (Eysenbach et al. 2018) proposed DIAYN to maximize the mutual information between states and skills, which results in a maximum entropy policy. Osa et al. (Osa, Tangkaratt, and Sugiyama 2021) proposed a method that can learn infinitely many solutions by training a policy conditioned on a continuous or discrete low-dimensional latent variable. Their method can learn diverse solutions in continuous control tasks via variational information maximization. There is also a growing corpus of work on diversity in multi-agent reinforcement learning (Mahajan et al. 2019; Lee, Yang, and Lim 2019; He, Shao, and Ji 2020). Mahajan et al. (Mahajan et al. 2019) proposed MAVEN, a method that overcomes the detrimental effects of QMIX's (Rashid et al. 2018) monotonicity constraint on exploration by maximizing the mutual information between latent variables and trajectories. However, their method can not find multiple diverse strategies for a specified task. He et al. (He, Shao, and Ji 2020) investigated multi-agent algorithms for learning diverse skills using information bottlenecks with unsupervised rewards. However, their method operates in an unsupervised manner, without external rewards. More recently, RSPO (Zhou et al. 2022) was proposed to derive diverse strategies. However, it requires multiple training stages, which results in poor sample efficiency – our method trains diverse strategies simultaneously which reduces sample complexity.

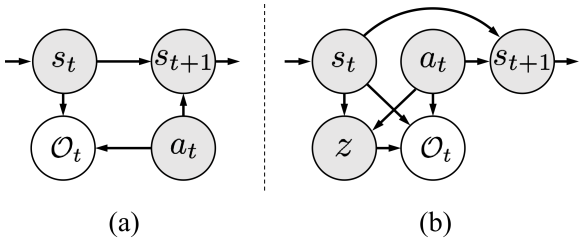


Figure 1: (a) The graphical model of MDPs. (b) The graphical model of diverse MDPs. Grey nodes are observed, and white nodes are hidden. As introduced in (Levine 2018), $\mathcal{O}_t$ is a binary random variable, where $\mathcal{O}_t = 1$ denotes that the action is optimal at time t, and $\mathcal{O}_t = 0$ denotes that the action is not optimal.

Preliminaries

RL can be formalized as an MDP. An MDP is a tuple $(\mathcal{S}, \mathcal{A}, P, r, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ represent state and action space respectively, $P(s, a) : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is the transition probability density, $r(s, a) : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is a reward function, and $\gamma \in [0, 1]$ is the discount factor.

Latent conditioned policy: We consider a policy π_θ that is conditioned on latent variable z to model diverse strategies, where θ represents policy parameters. For compactness, we will omit θ in our notation. We denote the latent-conditioned policy as $\pi(a|s, z)$ and a latent conditioned critic network as $V^\pi(s, z)$. For each episode, a single latent variable is sampled, $z \sim p(z)$ from a categorical distribution with n_z categories. In our work, we choose $p(z)$ to be a uniform distribution. The agent then conditions on this latent code z, to produce a trajectory τ_z.

Discounted state occupancy: The discounted state occupancy measure for policy π is defined as $\rho^\pi(s) = (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t P_t^\pi(s)$, where $P_t^\pi(s)$ is the probability that policy π visit state s at time t. The goal of the RL agent is to

train a policy π to maximize the discounted accumulated reward $J(\theta) = \mathbb{E}_{z\sim p(z), s\sim\rho^{\pi}(s), a\sim\pi(\cdot|s,z)}[\sum_t \gamma^t r(s_t, a_t)]$.

RL as probabilistic graphical model: An MDP can be framed as a probabilistic graphical model as shown in Fig. 1(a) and the optimal control problem can be solved as a probabilistic inference task (Levine 2018). In this paper, we propose a new probabilistic graphical model, denoted as the diverse MDP, as shown in Fig. 1(b). We introduce a binary random variable $\mathcal{O}_t$ and an integer random variable z into the model. $\mathcal{O}_t = 1$ denotes that action a_t is optimal at time step t and $\mathcal{O}_t = 0$ denotes it is not. Previous work(Levine 2018) has defined, $p(\mathcal{O}_t = 1|s_t, a_t, z) \propto \exp(r(s_t, a_t))$. The evidence lower bound (ELBO) is given by:

$$\begin{aligned}
&\log p(\mathcal{O}_{1:T}) \\
&\geq \mathbb{E}_{\tau\sim\mathcal{D}_\pi}[\log p(\mathcal{O}_{1:T}, a_{1:T}, s_{1:T}, z) \\
&\quad - \log \pi(a_{1:T}, s_{1:T}, z)] \\
&= \mathbb{E}_{\tau\sim\mathcal{D}_\pi}[\log p(\mathcal{O}_t|s_t, a_t, z) \\
&\quad + \log p(z|s_t, a_t) - \log \pi(a_t|s_t, z)],
\end{aligned} \tag{1}$$

where the trajectory $\tau = \{a_{1:T}, s_{1:T}, z\}$ is sampled from a trajectory dataset $\mathcal{D}_\pi$. The proof of Eq. 1 can be found in Appendix C. Note that the introduction of z gives rise to the term $p(z|s_t, a_t)$, which represents how identifiable the latent code is from the current policy behavior. This will be a crucial ingredient of DGPO, guiding the policy to explore and discover a set of diverse strategies.

Methodology

In this section, we will introduce our algorithm in detail. Our algorithm can be divided into two stages. In the first stage, we will focus on improving the agent's performance while maintaining its diversity. In the second stage, we will focus more on enhancing diverse strategies. Finally, we will introduce our final algorithm, denoted as Diversity-Guided Policy Optimization (DGPO), that unifies the two-stage training processes and also the implementation details.

Diversity Measurement

In this section, we present a diversity score capable of evaluating the diversity of a given set of policies. We then derive a diversity objective from this score to facilitate exploration. Eysenbach et al. (Eysenbach et al. 2018) proposed a diversity score based on mutual information between states and latent codes z,

$$I(s; z) = \mathbb{E}_{z\sim p(z), s\sim\rho^{\pi}(s)}[\log p(z|s) - \log p(z)], \tag{2}$$

where $I(\cdot, \cdot)$ stands for mutual information. As we can not directly calculate $p(z|s)$, we approximate it with a learned discriminator $q_\phi(z|s)$ and derive the ELBO as $\mathbb{E}_{z\sim p(z), s\sim\rho^{\pi}(s)}[\log q_\phi(z|s) - \log p(z)]$, where ϕ are the parameters of the discriminator network.

Mutual information is equal to the KL distance between the state marginal distribution of one policy and the average state marginal distribution, i.e., $I(s; z) = \mathbb{E}_{z\sim p(z)}[D_{KL}(\rho^{\pi}(s|z)||\rho^{\pi}(s))]$ (Eysenbach, Salakhutdinov, and Levine 2021). This means that $I(s; z)$ captures the diversity averaged over the *whole set* of policies. In DGPO, we wish to ensure that *any given pair* of strategies is different, rather than on average. As such, we define a novel, stricter diversity score,

$$\mathrm{DIV}(\pi_\theta) = \mathbb{E}_{z\sim p(z)}[\min_{z'\neq z} D_{KL}(\rho^{\pi_\theta}(s|z)||\rho^{\pi_\theta}(s|z'))]. \tag{3}$$

Instead of comparing policy with the average state marginal distribution, we compare it with the nearest policy in $\rho^{\pi}(s)$ space. In this way, setting $\mathrm{DIV}(\pi_\theta) \geq \delta$ means that each pair of policies have at least δ distance in terms of expectation. In order to optimize Eq. 3, we first derive a lower bound,

$$\mathrm{DIV}(\pi_\theta) \geq \mathbb{E}_{z,s}\left[\min_{z'\neq z} \log \frac{p(z|s)}{p(z|s) + p(z'|s)}\right]. \tag{4}$$

The proof is given in Appendix D. To maximize this lower bound, we first assume we can learn a latent code discriminator, $q_\phi(z|s)$, to approximate $p(z|s)$. We then define an intrinsic reward,

$$r_t^{in} = \min_{z'\neq z} \log \frac{q_\phi(z|s_{t+1})}{q_\phi(z|s_{t+1}) + q_\phi(z'|s_{t+1})}. \tag{5}$$

This allows us to define our final diversity objective,

$$J_{Div}(\theta) = \mathbb{E}_{z\sim p(z), s\sim\rho^{\pi}(s), a\sim\pi(\cdot|s,z)}[\sum_t \gamma^t r_t^{in}]. \tag{6}$$

A straightforward way to incorporate the diversity metric into a PGM is defining elements in Eq. 1 as,

$$\begin{aligned}
&p(\mathcal{O}_t = 1|s_t, a_t, z) = \exp\left(r(s_t, a_t)\right), \\
&p(z|s_t, a_t) = \exp\left(r_t^{in}\right).
\end{aligned} \tag{7}$$

However, the simple combination of extrinsic and intrinsic rewards may lead to poor performance. In the following paragraph, we formulate the algorithm as constrained optimization problems and mask elements in Eq. 7 based on constraints to guide the policy to explore.

Stage 1: Diversity-Constrained Optimization

Strategies that solve a given RL task may be very distinct. One can think of these as a set of discrete points in $\rho^{\pi}(s)$ space – if the distance between the points is large, perturbing around one single solution may not allow the discovery of all optimal strategies. Thus, we formulate the policy optimization process as a diversity-constrained optimization problem,

$$\max_{\pi_\theta} J(\theta), \text{ s.t. } J_{Div}(\theta) \geq \delta. \tag{8}$$

Under this objective, individual policies are constrained to keep a certain distance δ apart from each other, and systematically explore their own regions. We can introduce a Lagrange multiplier λ to tackle this constrained optimization problem,

$$\begin{aligned}
&\max_{\pi_\theta} \min_{\lambda\geq 0} J(\theta) + \lambda(J_{Div}(\theta) - \delta) \\
&\geq \max_{\pi_\theta} \mathbb{E}_{z\sim p(z), s\sim\rho^{\pi}(s)}[\min_{\lambda\geq 0} \mathbb{E}_{a\sim\pi(\cdot|s,z)} \\
&\quad [\sum_t \gamma^t r(s_t, a_t)] + \lambda(\sum_t \gamma^t r_t^{in} - \delta)].
\end{aligned} \tag{9}$$

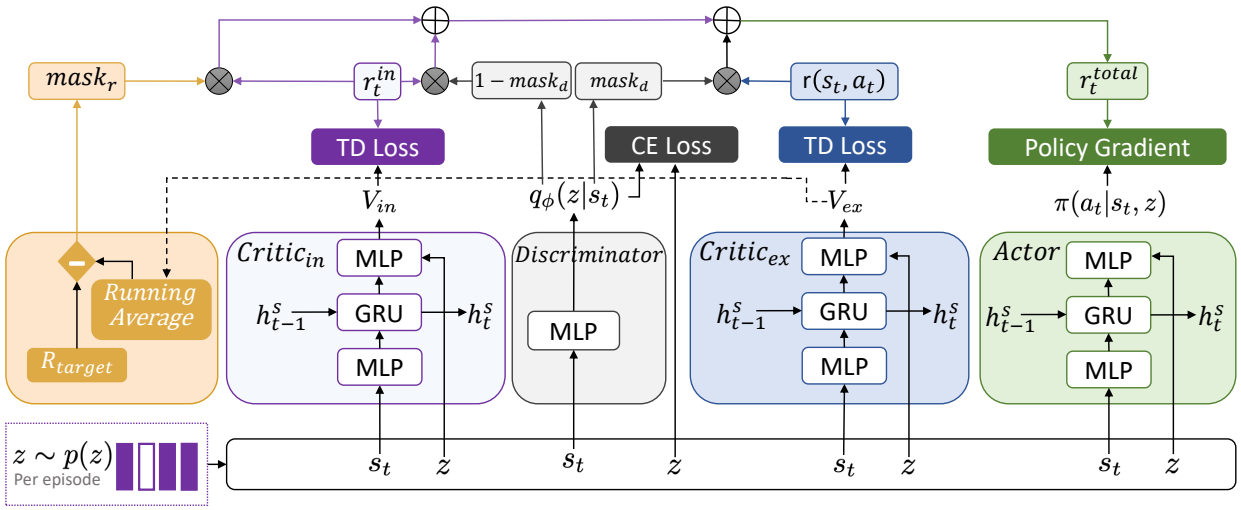


Figure 2: The overall framework of the DGPO algorithm. The top illustrates the way of calculating r_t^{total}, where $mask_r = \mathbb{I}[J(\theta) \geq R_{target}]$ and $mask_d = \mathbb{I}[J_{Div}(\theta) \geq \delta]$. The center shows the network structure and the data flow of the DGPO algorithm. The bottom shows the latent variable sampling process.

The proof can be found in Appendix E. Eq. 9 provides a lower bound on the Lagrange multiplier objective, which can be optimized more easily than the original problem. Eq. 9 can be interpreted as optimizing the extrinsic-rewards, $J(\theta)$, when diversity is greater than some threshold. Otherwise, the intrinsic rewards objective $J_{Div}(\theta)$ are optimized. From another perspective, one can think of masking out terms in elements in Eq. 1 based on a diversity metric. The updated objective can be written,

$$\begin{aligned} p(\mathcal{O}_t = 1|s_t, a_t, z) &= \exp\left(\mathbb{I}[J_{Div}(\theta) \geq \delta] r(s_t, a_t)\right), \\ p(z|s_t, a_t) &= \exp\left((1 - \mathbb{I}[J_{Div}(\theta) \geq \delta]) r_t^{in}\right), \end{aligned} \tag{10}$$

where $\mathbb{I}[\cdot]$ is the indicator function.

Stage 2: Extrinsic-Reward-Constrained Optimization

The objective developed in the previous section can return a set of discrete optimal points. However, sometimes two strategies may still converge to the same sub-optimal point. This can destabilize the training process since both strategies are attracted to the same optimal point but simultaneously repelled by each other. To stabilize the training process and improve diversity, we relax the definition of "optimal" and assume that a policy with accumulated extrinsic rewards greater than some target value R_{target} is an optimal policy. Thus, the policy optimization process can be formulated as an extrinsic-reward-constrained optimization problem,

$$\max_{\pi_\theta} J_{Div}(\theta), \text{ s.t. } J(\theta) \geq R_{target}. \tag{11}$$

This means that if two strategies try to converge to the same sub-optimal point. They are allowed to find their destiny in the neighborhood of the optimal point to further maximize the level of diversity. On the other hand, for those policies that are already sufficiently distinct, the diversity objective serves as an intrinsic reward to encourage exploration. Similar to how we deal with diversity-constrained optimization. We implement Eq. 11 by injecting an extrinsic-rewards-constraint into the framework. The updated elements in Eq. 1 can be defined as:

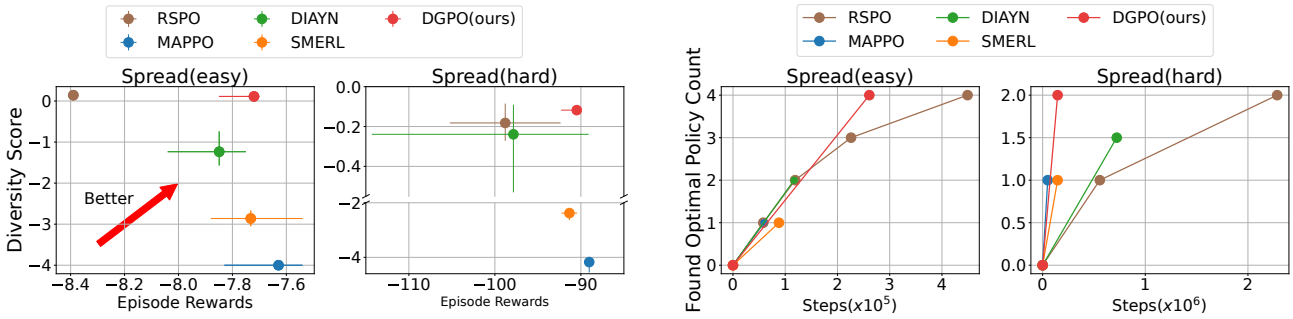
$$\begin{aligned} p(\mathcal{O}_t = 1|s_t, a_t, z) &= \exp(r(s_t, a_t)), \\ p(z|s_t, a_t) &= \exp(\mathbb{I}[J(\theta) \geq R_{target}] r_t^{in}). \end{aligned} \tag{12}$$

Diversity-Guided Policy Optimization

In this section, we will introduce our final algorithm which unifies the two-stage training processes into one unified algorithm We develop a new variation of PPO (Schulman et al. 2017) by considering policy network and critic network that are conditioned on latent variable z, i.e., $\pi(a_t|s_t, z)$. The critic network is divided into two parts, i.e., $V_{\psi_{ex}}^\pi(o_{1:t}, z)$ and $V_{\psi_{in}}^\pi(o_{1:t}, z)$, where ψ_{ex} and ψ_{in} are their parameters. The parameters of critic networks can be trained by a temporal difference (TD) loss (Sutton and Barto 2018):

$$\begin{aligned} L(\psi_{ex}) &= MSE\Big(V_{\psi_{ex}}^\pi(o_{1:t}, z), \\ &\sum_{t'=t}^{\infty} \gamma^{t'-t} \mathbb{I}[J_{Div}(\theta) \geq \delta] r(s_{t'}, a_{t'})\Big), \\ L(\psi_{in}) &= MSE\Big(V_{\psi_{in}}^\pi(o_{1:t}, z), \sum_{t'=t}^{\infty} \gamma^{t'-t} \\ &[(1 - \mathbb{I}[J_{Div}(\theta) \geq \delta]) + \mathbb{I}\left[J(\theta) \geq R_{target}\right]] r_{t'}^{in}\Big), \end{aligned} \tag{13}$$

where $L(\cdot)$ stands for loss function and $MSE(\cdot)$ stands for mean square error. We maintain a running average of $V_{\psi_{ex}}^\pi(s_t, z)$ to approximate $\mathbb{E}_{s\sim\rho^\pi(s), a\sim\pi(\cdot|s)}[\sum_t \gamma^t r(s_t, a_t)]$. We also construct a



(a) (b)

Figure 3: Experimental results in two MPE scenarios – Spread (easy) and Spread (hard) – each with multiple optimal solutions. (a) Plot showing extrinsic reward performance vs. how diverse the set of discovered strategies are. Positions in the upper–right corner are preferred – DGPO is located here. (b) Plot showing at which point in training each optimal strategy is discovered. Results show that only DGPO and RSPO can find all the solutions. But DGPO achieved over $1.7\times$ and $15\times$ speedup in convergence speed compared to RSPO in the Spread (easy) and Spread (hard) scenarios, respectively.

discriminator $q_\phi(z|s_t)$ that takes the state as input and predict the probability of latent variable z, where ϕ is the parameter of the discriminator network. And the discriminator can be trained in a supervised manner:

$$L(\phi) = \mathbb{E}_{(s_t,a_t,z)\sim\mathcal{D}_\pi}[CE(q_\phi(s_t), z)], \tag{14}$$

where $CE(\cdot,\cdot)$ stands for cross entropy loss. We implement DGPO by incorporating diversity-constrained optimization and extrinsic-reward-constrained optimization into the same framework, i.e., the total value of the state and the total reward can be defined as below:

$$\begin{aligned} &V^\pi_{total}(o_{1:t}, z) = V^\pi_{\psi_{in}}(o_{1:t}, z) + V^\pi_{\psi_{ex}}(o_{1:t}, z), \\ &r^{total}_t = \mathbb{I}[J_{Div}(\theta) \geq \delta] r(s_{t'}, a_{t'}) \\ &+ [(1 - \mathbb{I}[J_{Div}(\theta) \geq \delta]) + \mathbb{I}\left[J(\theta) \geq R_{target}\right]] \, r^{in}_{t'}. \end{aligned} \tag{15}$$

In theory, it is not feasible to simultaneously conduct diversity-constrained optimization and extrinsic-reward-constrained optimization. As a result, the aforementioned implementation serves as an approximation to the original objective. We posit that this approximation is reasonable, as the empirical findings demonstrate that the two training stages are not concurrent. Fig. 2 shows the overall framework of the DGPO algorithm. The detailed training process of DGPO can be found in the Appendix G.

Experiments

In this section, we evaluate our algorithm on several RL benchmarks – Multi-agent Particle Environment (MPE) (Mordatch and Abbeel 2018), StarCraft Multi-Agent Challenge (SMAC) (Samvelyan et al. 2019), and Atari (Bellemare et al. 2013). We compare our algorithm to four baseline algorithms:

MAPPO (Yu et al. 2021a): MAPPO adapts the single-agent PPO (Schulman et al. 2017) algorithm to the multi-agent setting by using a centralized value function with shared team-based rewards.

DIAYN (Eysenbach et al. 2018): DIAYN trains agents with a mutual-information based intrinsic reward to discover a diverse set of skills. In our setup, these intrinsic rewards are combined with extrinsic rewards.

SMERL (Kumar et al. 2020): SMERL maximizes a weighted combination of intrinsic rewards and extrinsic rewards when the return of extrinsic reward is greater than some given threshold.

RSPO (Zhou et al. 2022): RSPO is an iterative algorithm for discovering a diverse set of quality strategies. It toggles between extrinsic and intrinsic rewards based on a trajectory-based novelty measurement.

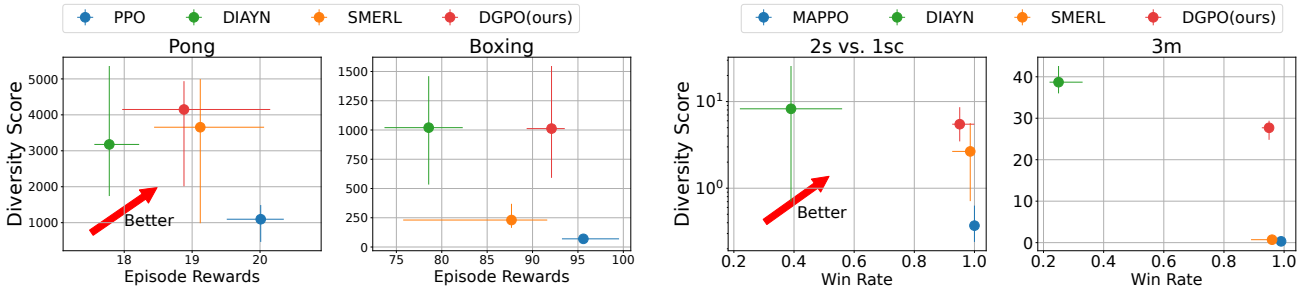
As far as possible, all methods use the same hyper-parameters. However, there is some difference for RSPO, for which we use the open-source implementation. (Full hyper-parameters are listed in the Appendix B.) All experiments were performed on a machine with 128 GB RAM, one 32-core CPU, and one GeForce RTX 3090 GPU.

Multi-Agent Particle Environment

We evaluate on two scenarios shown per Fig. 5, Spread (easy) and Spread (hard). In Spread (easy), there are four landmarks and one agent. The agent starts from the center and aims to reach one of the landmarks, giving four optimal solutions. In Spread (hard), there are three agents and three landmarks. Agents cooperate to cover all the landmarks and avoid colliding with others, giving two optimal solutions. Model weights are shared across agents.

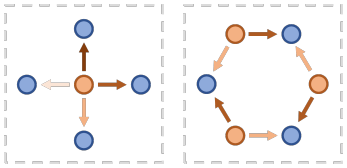
Similar to (Parker-Holder et al. 2020), we introduce a metric to quantitatively evaluate the diversity score of the given set of policies, Π: $M_{Div}(\Pi) = \frac{1}{n_z}\sum_{i=1}^{n_z}\sum_{j=i+1}^{n_z} \ln(\|\Phi(\pi_i) - \Phi(\pi_j)\|_2)$, where $\Phi(\pi)$ is the behavior embedding of the policy π. For MPE, $\Phi(\pi)$ is represented by the concatenated agents' positions over an episode.

We set $n_z = 4$ in Spread (easy) and $n_z = 2$ in Spread (hard) to test whether an algorithm can discover all optimal



(a) (b)

Figure 4: Plots showing extrinsic reward performance vs. the diversity of the set of discovered strategies. (a) In two Atari games. (b) In two SMAC scenarios.



(a) Easy (b) Hard

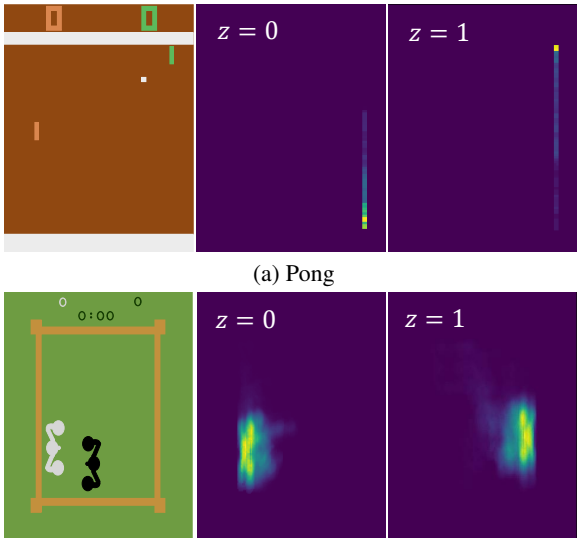
Figure 5: The initial state of Spread (easy) and Spread (hard). In both scenarios, Agents (orange dots) aim to reach one of the destinies (blue dots). We highlight the optimal solutions with arrows of different colors.

solutions. The performance, diversity score, and steps required for convergence for all algorithms are given in Fig. 3. Results are averaged over five seeds.

DGPO provides a favorable combination of high diversity, and high reward. It also exhibits rapid convergence. MAPPO only recovers a single solution in each setting. DIAYN only finds 2/4 optimal solutions in Spread (easy) and often only 1/2 in Spread (hard). This supports our earlier claim that a naive combination of extrinsic and intrinsic rewards is insufficient. SMERL is only able to discover new strategies by perturbing around a discovered global optimal. Thus, it is unable to find more than one optimal strategy. RSPO is the only other method also to recover all optimal strategies. However, relative to DGPO, RSPO achieves lower overall reward and slower convergence.

Atari

We evaluate DGPO on the Atari games Pong and Boxing, to test the performance for tasks with image observations. DGPO's diversity metric is defined similarly to the one used in MPE (full details in Appendix F). In each environment, we set $n_z = 2$. Results are summarized in Fig. 4(a), averaged over five seeds. We also reported the experimental results for $n_z = 10$ in Appendix I. DGPO again delivers a favorable trade-off between external reward and strategy diversity. Fig. 6(a) visualizes the results of the two strategies obtained by DGPO on Pong. In this game, the agent controls the green paddle. When $z = 0$ the agent holds the paddle at the bottom of the screen until the ball is near, while when $z = 1$, the default position of the paddle is at the top of the screen. Fig. 6(b) similarly shows the different strategies obtained by DGPO on Boxing, where the agent controls the white character, and is rewarded for punching the black opponent. The heatmap shows DGPO learns to attack from different sides. Other baseline algorithms tend to remain in a single corner.



(a) Pong

(b) Boxing

Figure 6: Screen shots and heat maps of agents' trajectories on Pong and Boxing. (a) In the Pong game, our agent controls the paddle (green block) to hit the ball (white dot). (b) In the Boxing game, our agent (in white) has a boxing match with the opponent (in black).

Figure 7: Visualization results of the three strategies obtained by DGPO on *3m* map. The green arrows show the trajectories of our agents (in green).

StarCraft II

We conduct experiments on two StarCraft II maps (*2s_vs._1sc* & *3m*) from SMAC. We set $n_z = 3$ and measure the mean win rates over five seeds. Fig. 4(b) shows that, relative to other algorithms, DGPO discovers sets of strategies that are both diverse and achieve good win rates. Fig. 7 visualizes three strategies obtained by DGPO on *3m* map. In this map, we control the three green agents to combat the red built-in agents. We visualize the trajectories of our agents with green arrows. When $z = 0$, the policy produces an aggressive strategy, with agents moving directly forward to attack the enemies. When $z = 1$, the agents display a kiting strategy, alternating between attacking and moving. This allows them to attack enemies while limiting the taken damage. When $z = 2$, the policy produces another kiting strategy but now with a downwards, rather than upwards drift.

Ablation Study

We performed ablation studies on MPE Spread (hard) tasks, systematically removing each element of our algorithm to assess its impact on diversity. The empirical result is shown in Fig. 8(a). Throughout this section, we set $n_z = 3$ and the result is averaged over 5 seeds. **Change-Diversity-Measurement** uses mutual information as shown in Eq. 2 as diversity metric. Experimental results indicate that in the Spread (hard) scenario, the diversity score of Change-Diversity-Measurement is lower than that of DGPO. While optimizing three policies simultaneously, the limited number of optimal solutions (only two) leads to one policy behaving differently while the other two exhibit similar behavior. Consequently, the diversity score based on mutual information becomes artificially high (as it reflects the overall diversity level of the policy set), causing the policy to stop optimizing diversity, even though two policies continue to behave similarly. **No-Diversity-Constrained-Optimization** excludes diversity-constrained optimization. Empirical results reveal that it only identifies one optimal solution. This suggests that utilizing the diversity metric as a constraint, rather than blending it with extrinsic rewards during the initial stages of training, significantly enhances the algorithm's effectiveness. **No-Extrinsic-Reward-Constrained-Optimization** omits extrinsic-reward-constrained optimization. While DGPO continues to enhance the diversity score as the expected return surpasses R_{target}, No-Extrinsic-Reward-Constrained-Optimization fails to do so. Consequently, it ultimately attains a lower diversity score compared to DGPO.

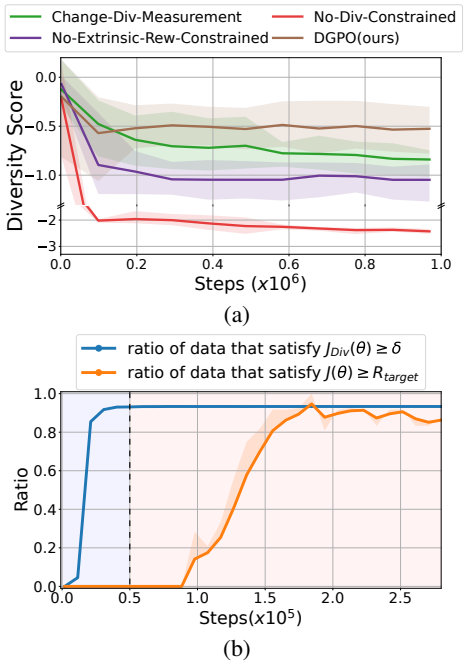


Figure 8: (a) The impact of each component of our algorithm on diversity scores in the MPE Spread (hard) scenario. (b) DGPO can be distinctly divided into two stages: diversity-constrained optimization and extrinsic-reward-constrained optimization.

Our algorithm can be divided into two distinct stages, as depicted in Fig. 8(b). In the initial stage, we focus on diversity-constrained optimization until policies exhibit sufficient behavioral differences. Once the expected return reaches R_{target}, we transition to the extrinsic-reward-constrained optimization stage. Here, we maintain a fixed performance level at R_{target} while maximizing the diversity score. These stages are clearly separated and non-overlapping.

Conclusions

In conclusion, this paper introduced the Diversity-Guided Policy Optimization (DGPO) algorithm, which demonstrates its capability to efficiently uncover diverse strategies that yield high rewards. By framing the training process as a pair of constrained optimization problems and solving them through probabilistic inference, DGPO stands out as a promising on-policy algorithm. Through experiments, we observed that DGPO strikes a favorable balance between diversity scores and rewards, all while exhibiting improved sample efficiency. Moving forward, we envision delving deeper into its potential to handle challenges such as exploration, self-play, and robustness.

References

Bellemare, M. G.; Naddaf, Y.; Veness, J.; and Bowling, M. 2013. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47: 253–279.

Berner, C.; Brockman, G.; Chan, B.; Cheung, V.; Debiak, P.; Dennison, C.; Farhi, D.; Fischer, Q.; Hashme, S.; Hesse, C.; et al. 2019. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*.

Chow, Y.; Tulepbergenov, A.; Nachum, O.; Ryu, M.; Ghavamzadeh, M.; and Boutilier, C. 2022. A Mixture-of-Expert Approach to RL-based Dialogue Management. *arXiv preprint arXiv:2206.00059*.

Derek, K.; and Isola, P. 2021. Adaptable Agent Populations via a Generative Model of Policies. *Advances in Neural Information Processing Systems*, 34: 3902–3913.

Eysenbach, B.; Gupta, A.; Ibarz, J.; and Levine, S. 2018. Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*.

Eysenbach, B.; Salakhutdinov, R.; and Levine, S. 2021. The Information Geometry of Unsupervised Reinforcement Learning. *arXiv preprint arXiv:2110.02719*.

Furmston, T.; and Barber, D. 2010. Variational methods for reinforcement learning. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 241–248. JMLR Workshop and Conference Proceedings.

Gao, J.; Bi, W.; Liu, X.; Li, J.; and Shi, S. 2019. Generating multiple diverse responses for short-text conversation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 6383–6390.

Haarnoja, T.; Hartikainen, K.; Abbeel, P.; and Levine, S. 2018a. Latent space policies for hierarchical reinforcement learning. In *International Conference on Machine Learning*, 1851–1860. PMLR.

Haarnoja, T.; Zhou, A.; Hartikainen, K.; Tucker, G.; Ha, S.; Tan, J.; Kumar, V.; Zhu, H.; Gupta, A.; Abbeel, P.; et al. 2018b. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*.

Hausman, K.; Springenberg, J. T.; Wang, Z.; Heess, N.; and Riedmiller, M. 2018. Learning an embedding space for transferable robot skills. In *International Conference on Learning Representations*.

He, S.; Shao, J.; and Ji, X. 2020. Skill Discovery of Coordination in Multi-agent Reinforcement Learning. *arXiv preprint arXiv:2006.04021*.

Huang, S.; Chen, W.; Zhang, L.; Li, Z.; Zhu, F.; Ye, D.; Chen, T.; and Zhu, J. 2021. TiKick: Towards Playing Multi-agent Football Full Games from Single-agent Demonstrations. *arXiv preprint arXiv:2110.04507*.

Huang, S.; Su, H.; Zhu, J.; and Chen, T. 2019a. Combo-action: Training agent for fps game with auxiliary tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 954–961.

Huang, S.; Su, H.; Zhu, J.; and Chen, T. 2019b. SVQN: Sequential Variational Soft Q-Learning Networks. In *International Conference on Learning Representations*.

Igl, M.; Zintgraf, L.; Le, T. A.; Wood, F.; and Whiteson, S. 2018. Deep variational reinforcement learning for POMDPs. In *International Conference on Machine Learning*, 2117–2126. PMLR.

Kumar, S.; Kumar, A.; Levine, S.; and Finn, C. 2020. One solution is not all you need: Few-shot extrapolation via structured maxent rl. *Advances in Neural Information Processing Systems*, 33: 8198–8210.

Lanctot, M.; Zambaldi, V.; Gruslys, A.; Lazaridou, A.; Tuyls, K.; Pérolat, J.; Silver, D.; and Graepel, T. 2017. A unified game-theoretic approach to multiagent reinforcement learning. *Advances in neural information processing systems*, 30.

Lee, A. X.; Nagabandi, A.; Abbeel, P.; and Levine, S. 2019. Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model. *arXiv preprint arXiv:1907.00953*.

Lee, Y.; Yang, J.; and Lim, J. J. 2019. Learning to coordinate manipulation skills via skill behavior diversification. In *International Conference on Learning Representations*.

Levine, S. 2018. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*.

Li, J.; Monroe, W.; Ritter, A.; Galley, M.; Gao, J.; and Jurafsky, D. 2016. Deep reinforcement learning for dialogue generation. *arXiv preprint arXiv:1606.01541*.

Mahajan, A.; Rashid, T.; Samvelyan, M.; and Whiteson, S. 2019. Maven: Multi-agent variational exploration. *arXiv preprint arXiv:1910.07483*.

Mohamed, S.; and Rezende, D. J. 2015. Variational information maximisation for intrinsically motivated reinforcement learning. *arXiv preprint arXiv:1509.08731*.

Mordatch, I.; and Abbeel, P. 2018. Emergence of grounded compositional language in multi-agent populations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Mouret, J.-B.; and Doncieux, S. 2009. Overcoming the bootstrap problem in evolutionary robotics using behavioral diversity. In *2009 IEEE Congress on Evolutionary Computation*, 1161–1168. IEEE.

Osa, T.; Tangkaratt, V.; and Sugiyama, M. 2021. Discovering diverse solutions in deep reinforcement learning. *arXiv preprint arXiv:2103.07084*.

Parker-Holder, J.; Pacchiano, A.; Choromanski, K. M.; and Roberts, S. J. 2020. Effective diversity in population based reinforcement learning. *Advances in Neural Information Processing Systems*, 33: 18050–18062.

Pavel, C.; Budulan, S.; and Rebedea, T. 2020. Increasing Diversity with Deep Reinforcement Learning for Chatbots. In *RoCHI*, 123–128.

Raffin, A.; Hill, A.; Traoré, R.; Lesort, T.; Díaz-Rodríguez, N.; and Filliat, D. 2018. S-RL Toolbox: Environments,

Datasets and Evaluation Metrics for State Representation Learning. *arXiv preprint arXiv:1809.09369*.

Rashid, T.; Samvelyan, M.; Schroeder, C.; Farquhar, G.; Foerster, J.; and Whiteson, S. 2018. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *International Conference on Machine Learning*, 4295–4304. PMLR.

Samvelyan, M.; Rashid, T.; De Witt, C. S.; Farquhar, G.; Nardelli, N.; Rudner, T. G.; Hung, C.-M.; Torr, P. H.; Foerster, J.; and Whiteson, S. 2019. The starcraft multi-agent challenge. *arXiv preprint arXiv:1902.04043*.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

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

Tang, Z.; Yu, C.; Chen, B.; Xu, H.; Wang, X.; Fang, F.; Du, S.; Wang, Y.; and Wu, Y. 2021. Discovering Diverse Multi-Agent Strategic Behavior via Reward Randomization. *arXiv preprint arXiv:2103.04564*.

Vinyals, O.; Babuschkin, I.; Czarnecki, W. M.; Mathieu, M.; Dudzik, A.; Chung, J.; Choi, D. H.; Powell, R.; Ewalds, T.; Georgiev, P.; et al. 2019. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782): 350–354.

Xu, F.; Xu, G.; Wang, Y.; Wang, R.; Ding, Q.; Liu, P.; and Zhu, Z. 2022. Diverse dialogue generation by fusing mutual persona-aware and self-transferrer. *Applied Intelligence*, 1–14.

Yu, C.; Velu, A.; Vinitsky, E.; Wang, Y.; Bayen, A.; and Wu, Y. 2021a. The Surprising Effectiveness of MAPPO in Cooperative, Multi-Agent Games. *arXiv preprint arXiv:2103.01955*.

Yu, C.; Yang, X.; Gao, J.; Yang, H.; Wang, Y.; and Wu, Y. 2021b. Learning Efficient Multi-Agent Cooperative Visual Exploration. *arXiv preprint arXiv:2110.05734*.

Zahavy, T.; O'Donoghue, B.; Barreto, A.; Flennerhag, S.; Mnih, V.; and Singh, S. 2021. Discovering diverse nearly optimal policies with successor features. In *ICML 2021 Workshop on Unsupervised Reinforcement Learning*.

Zhou, Z.; Fu, W.; Zhang, B.; and Wu, Y. 2022. Continuously Discovering Novel Strategies via Reward-Switching Policy Optimization. *arXiv preprint arXiv:2204.02246*.

Ziebart, B. D.; Maas, A. L.; Bagnell, J. A.; and Dey, A. K. 2008. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, 1433–1438. Chicago, IL, USA.