ECAI 2025
I. Lynce et al. (Eds.)

doi:10.3233/FAIA251201

Multiagent Quality-Diversity for Effective Adaptation

Siddarth Iyer[a,*], Ayhan Alp Aydeniz[a], Gaurav Dixit[a,1] and Kagan Tumer[a,1]

[a]Oregon State University

Abstract.

Robust adaptation in multiagent settings requires learning not just a single optimal behavior, but a repertoire of high-performing and diverse team behaviors that can succeed under environmental contingencies. Traditional multiagent reinforcement learning methods typically converge to a single specialized team behavior, limiting their adaptability. Recent approaches like Mix-ME promote behavioral diversity but rely solely on evolutionary operators, often resulting in sample-inefficiency and uncoordinated team composition. This work introduces Multiagent Sample-Efficient Quality-Diversity (MASQD), a learning framework that produces an archive of diverse, high-performing multiagent teams. MASQD builds on the Cross-Entropy Method Reinforcement Learning algorithm and extends it to the multiagent setting by representing teams as parameter-shared neural networks, directing exploration from previously discovered behaviors, and guiding refinement through a descriptor-conditioned critic. Through this coupling of anchored exploration and targeted exploitation, MASQD produces functional diversity: teams that are not only behaviorally distinct but also robust and effective under varied conditions. Experiments across four Multiagent MuJoCo tasks show that MASQD outperforms state-of-the-art baselines in both team fitness and functional diversity.

1 Introduction

Multiagent learning is critical for success in complex, real-world domains such as disaster response [35], autonomous robotics [26], and planetary exploration [22]. In these domains, teams of agents must not only perform well in the conditions they were trained for but also adapt effectively to unforeseen contingencies in their environment [4, 28, 38]. For instance, during a planetary mission, a team of agents may need to traverse previously unseen terrain features or continue operating despite partial damage to some members [40]. Success in such settings requires teams to learn and catalog diverse behaviors, enabling them to rapidly test and deploy alternative solutions as conditions change [1].

In cooperative settings, Multiagent Reinforcement Learning (MARL) methods train teams of agents to coordinate on complex tasks, using gradient-based optimization to refine team behavior efficiently [14, 41]). Because agents are trained together, MARL produces coordinated teams that perform well in the environments they were trained for. However, the resulting solution typically represents a single, highly specialized behavior. When the environment conditions change, the team cannot adapt without extensive retraining, significantly limiting its applicability in dynamic multiagent settings.

Quality-Diversity (QD), a family of diversity-first methods, offers a promising alternative by developing and preserving an archive of high-performing and behaviorally diverse policies [15, 34, 37]. QD has been particularly successful in single-agent settings, such as rapid adaptation through damage recovery [8]. Extending QD to multiagent settings, however, introduces unique challenges: maintaining coordinated yet diverse teams, efficiently exploring high-dimensional joint policy spaces, and improving sample-efficiency in settings where agent interactions are complex and costly to evaluate [10, 11]. Mix-ME, a recent multiagent QD method, inherits these limitations [23]. While it produces an archive of diverse teams via crossover-based team recombination, its reliance on evolutionary operators leads to sample-inefficiency. Moreover, significant computational effort is spent evaluating poorly coordinated teams generated through uninformed recombination.

This work introduces Multiagent Sample-Efficient Quality-Diversity (MASQD)[2], a learning framework that generates an archive of high-performing and diverse multiagent teams capable of adapting to contingencies in the environment. MASQD performs an informed search through the behavior space, facilitating the discovery of multiple high-quality regions that yield distinct, coordinated team behaviors. It builds on the Cross-Entropy Method Reinforcement Learning (CEM-RL) algorithm [33] which decouples exploration (via CEM) from exploitation (via reinforcement learning), by extending it to the multiagent QD setting through three key components: (1) a team-level neural network architecture with parameter sharing that promotes tight coordination, (2) a periodic sampling stage that uses previously discovered behaviors as anchors in the behavior space to guide exploration, and (3) a descriptor-conditioned critic that refines policies toward specific target behaviors, improving performance and expanding local coverage in the behavior archive.

Our key insight is that achieving *functional diversity*—not merely discovering a wide range of behaviors, but developing an archive of effective and adaptable ones—requires targeted exploration and refinement. MASQD continually expands high-performing regions of the behavior space by anchoring exploration and using a descriptor-conditioned critic to guide policies toward nearby target behaviors. The tight coupling of anchored exploration and targeted exploitation allows MASQD to reuse and improve prior solutions, forming localized regions of high-performing teams rather than isolated solutions. The result is an archive of diverse, high-quality teams that can be deployed for robust adaptation to environmental changes.

Across four Multiagent MuJoCo tasks, MASQD consistently produces archives of functionally diverse teams that outperform multiagent extensions of single-agent QD methods and Mix-ME by over an order of magnitude, while discovering a comparable proportion

* Corresponding Author. Email: viswansi@oregonstate.edu.
[1] Equal supervision.

[2] Code: https://github.com/siddiyer/masqd

of behaviors. Beyond these metrics, MASQD adapts over 100% better than its baselines in challenging variants of each environment, where agents must compensate for changes in gravity, recover from leg dysfunction, and traverse steep terrain. These results highlight MASQD's ability to generate diverse, high-performing teams that remain robust under contingencies.

2 Background and Related Work

2.1 Problem Formulation

We consider a team of N agents that interact with an environment using their individual observations o at each time step t for an episode length of T, modeled as a decentralized partially observable Markov Decision Process (Dec-POMDP) [30]. In the Multiagent QD setting, the objective is to learn a joint policy $\pi = (\pi_1, \ldots, \pi_N)$ that maximizes the expected discounted reward of the system for each discretized behavior d in the behavior space, $\mathcal{D}$. The behavior descriptor function $\mathcal{Z} : \tau \to \mathcal{D}$ can be hand-designed [15, 32] or a black-box [9, 17, 20] that characterizes the behavior of a team over a trajectory.

2.2 Multiagent Reinforcement Learning

Multiagent Reinforcement Learning (MARL) addresses the challenge of decision-making in cooperative settings. To mitigate the instability of fully decentralized training, most approaches adopt the Centralized Training with Decentralized Execution (CTDE) paradigm [14, 41], which allows agents to access global information during training while acting independently at deployment. In our setting, we use a Multi-Headed Actor (MHA) policy architecture [2, 27], where all agents share trunk layers that process common features, while each agent has its own independent output head. This structure improves coordination, reduces the number of trainable parameters, and allows an entire team to be represented by a single neural network. Despite these advances, most MARL algorithms focus on learning a single high-performing behavior characterized by a descriptor d. In contrast, MASQD aims to learn a diverse archive of high-performing behaviors.

2.3 Quality-Diversity

Quality-Diversity (QD) algorithms aim to learn a repertoire of diverse, high-performing policies by optimizing in a pre-defined behavior space [3, 7, 24]. A policy, characterized by a behavior descriptor d, is stored in an archive $\mathcal{A}$ and discovered through a loop of sampling, perturbation, and insertion based on novelty and fitness. While this improves behavior coverage [39], QD methods are often sample-inefficient and struggle to refine policies without drifting from their original behaviors.

Descriptor-conditioned critics (DCG) address both issues by learning Q-values for state-action pairs conditioned on a target descriptor d'. This allows gradient-based refinement toward specific behaviors. DCG [12, 13] demonstrates that descriptor conditioning improves policy optimization compared to regular critics, as the same transition may be useful for one behavior but harmful for another. Prior approaches include using successor features with constrained optimization [18, 19] or modifying the critic loss directly [12, 13]. We adopt the latter due to its simplicity and compatibility with standard reinforcement learning algorithms (Algorithm 1). To satisfy the continuity hypothesis of the Universal Approximation Theorem [21], MASQD uses a similarity function $S(d, d') = \exp(-\|d - d'\|/l)$ to down-weight transitions that deviate from the target. Training requires both positive ($d = d'$) and negative ($d \neq d'$) samples, which MASQD logs into $\mathcal{R}$ during rollouts.

Algorithm 1: Descriptor-Conditioned TD3

Function `DC_TD3` (π_θ*: solution,* N*: number of agents*) **:**
  for $i = 1 \to N$ *(each agent repeats* ξ *times)* **do**
    Sample K transitions, $(s, a, r(s, a), s', d, d')$ from $\mathcal{R}$
    Sample smoothing noise ϵ
    $\sigma \leftarrow$ S(d, d') $\leftarrow \exp \frac{-||d-d'||}{l}$
    $y \leftarrow \sigma\, r(s, a) + \gamma \min_{i=1,2} Q_{\theta_i'}(s', \pi_{\phi'}(s' \mid d') + \epsilon \mid d')$
    $\nabla_{\theta_i} \frac{1}{K} \sum (Q_{\theta_i}(s, a) - y)^2$
    if *t mod* Δ **then**
      Actor update:
      $\frac{1}{N} \sum \nabla_a Q_{\theta_1}(s, a \mid d')\big|_{a=\pi_\phi(s)} \nabla_\phi \pi_\phi(s)$
      Soft update target networks $Q_{\theta_i'}$ and $\pi_{\phi'}$

Multiagent Quality-Diversity is a nascent but growing area challenged by the difficulty of defining behavior spaces for teams. Two relevant works are the Asymmetric Island Model (AIM) [10], which decomposes the task into agent-level QD spaces (at the cost of egocentric diversity and requiring agent-specific rewards), and Mix-ME [23], which maintains team-level diversity by storing n-tuples in each niche. However, Mix-ME forms teams by randomly sampling individual policies from across niches, preventing cohesive team optimization. While highly diverse, this design yields low-performing teams and remains sample-inefficient due to repeated rollouts.

2.4 Combining Evolution and Reinforcement Learning

Evolution Strategies (ES) are black-box optimization methods that update a parameterized distribution to maximize fitness via estimated gradients [39, 5].

The Cross-Entropy Method (CEM) is an effective ES that samples a population from a Gaussian distribution (μ, Σ), evaluates all individuals, and updates the distribution using the top K_e performers (Algorithm 2) [36].

Algorithm 2: CEM Update

Function `CEM_Update` ($\{\theta_i\}_{i=1}^N$*: solutions,* K_e*: elites*) **:**
  Sort $\{\theta_i\}_{i=1}^N$ in descending order of fitness
  Select top K_e elite solutions: $\{\theta_i\}_{i=1}^{K_e}$
  $\lambda_i \leftarrow \frac{\log(1+K_e)/i}{\sum_{j=1}^{K_e} \log(1+K_e)/j} \quad \forall i \in [1, K_e]$
  $\mu_{\text{new}} \leftarrow \sum_{i=1}^{K_e} \lambda_i \theta_i$
  $\Sigma_{\text{new}} \leftarrow \sum_{i=1}^{K_e} \lambda_i (\theta_i - \mu_{\text{old}})(\theta_i - \mu_{\text{old}})^T + \epsilon \mathcal{I}$
  return $\mu_{\text{new}}, \Sigma_{\text{new}}$

Evolutionary Reinforcement Learning (ERL) combines the parameter space search of ES with the sample-efficiency of RL [2, 25, 29]. CEM-RL merges CEM with TD3, splitting the population into two halves: one refined via TD3 gradients, the other directly evaluated [33, 16]. Transitions from both are stored in a shared replay buffer, improving critic training due to the rich diverse experiences collected by CEM and TD3. However, CEM-RL was designed for single-agent settings, focusing solely on maximizing performance for a single policy without regard to behavioral diversity.

Our method builds on CEM-RL to support multiagent QD, integrating behavior-conditioned refinement and archive-based sampling to learn a repertoire of diverse, high-performing teams.

3 Multiagent Sample-Efficient Quality-Diversity

Multiagent Sample-Efficient Quality-Diversity (MASQD) is a learning framework for training and archiving a population of high-performing and behaviorally diverse multiagent teams capable of adapting to environmental contingencies. It builds on the Cross-Entropy Method Reinforcement Learning (CEM-RL) algorithm to simultaneously discover policies that exhibit novel behaviors and improve their performance. MASQD introduces three core mechanisms that enable this diversity-driven optimization: (1) a team-level neural network architecture with shared parameters to foster coordination, (2) periodic sampling that leverages previously discovered behaviors to anchor and guide exploration (Section 3.2), and (3) a descriptor-conditioned critic that enables fine-grained policy refinement toward specific target behaviors (Section 3.4).

MASQD proceeds in two stages: 1) A warm-up phase initializes the archive and replay buffers using random policies sampled from an initial distribution. This bootstraps the behavior space and provides the initial training data used by the subsequent optimization phase. 2) In the optimization phase, half of the population is used to drive CEM updates that support exploration, while the remaining half is refined using a descriptor-conditioned critic to target behaviors near previously sampled regions from the archive. This split in evaluation enables MASQD to achieve both quality exploration, by discovering new high-performing regions, and targeted exploitation, by improving known behaviors and locally expanding previously discovered regions of the behavior space. Algorithm 3 provides a high-level sketch of MASQD.

3.1 Warm-up Phase

MASQD begins with a warm-up phase that populates the behavior archive $\mathcal{A}$ and replay buffers $\mathcal{R}_{n=1}^{N}$ (one per agent) with initial policies and corresponding trajectories. A multi-headed actor network π_μ, representing a team policy, is randomly initialized to serve as the mean of the parameterized distribution $\mathcal{N}(\pi_\mu, \Sigma)$ used by CEM, with covariance $\Sigma = \sigma_{\text{init}}\mathcal{I}$ (lines 2-5). For ω iterations, MASQD samples teams (multi-head policies) from $\mathcal{N}(\pi_\mu, \Sigma)$ and performs rollouts. Each team's policy, fitness, and behavior descriptor are added to the archive, while the resulting trajectories are inserted into the replay buffers $\mathcal{R}$ (lines 7-10). The warm-up phase bootstraps and stabilizes the critic with diverse early experiences and seeds the archive with a broad initial slice of the behavior space [12].

3.2 Anchoring Exploration

The optimization phase begins by resetting the mean and covariance of MASQD's sampling distribution to re-center exploration around a previously discovered region of the behavior space (every $resample$ iterations; line 12). A policy-descriptor pair (π_μ, d) is sampled from the archive $\mathcal{A}$, and the descriptor is perturbed with Gaussian noise to yield a new target d_{target}. The mean of the distribution is set to π_μ, and the covariance matrix is reinitialized to $\Sigma = \sigma_{\text{init}}\mathcal{I}$ to encourage local exploration around the anchor (lines 13-15). This distribution is used to sample new teams in subsequent stages of the optimization phase, enabling MASQD to revisit, refine, and locally expand previously discovered behaviors.

Algorithm 3: Multiagent Sample-Efficient Quality-Diversity

1 **Function** `MASQD` (*P: population size, N: num of agents*) **:**
2 Initialize replay buffers $\mathcal{R} = \{\mathcal{R}_1, \ldots, \mathcal{R}_N\}$
3 Initialize archive $\mathcal{A}$
4 Initialize mean actor π_μ randomly
5 Initialize covariance matrix $\Sigma = \sigma_{\text{init}}\mathcal{I}$
6 $steps \leftarrow 0$
7 *Warmup:* **for** $i = 1 \rightarrow \omega$ **do**
8 Sample $\Pi = \{\pi_1, \ldots, \pi_P\} \sim \mathcal{N}(\pi_\mu, \Sigma)$
9 $steps$ += `rollout` $(\Pi, \mathcal{R}, \mathcal{A})$
10 $(\pi_\mu, \Sigma) \leftarrow$ `CEM_Update`$(\Pi, P/2)$
11 *Optimization:* **while** $steps <$ *max_steps* **do**
12 **if** $resample$ **then**
13 $(\pi_\mu, d) \sim \mathcal{A}$
14 $d_{\text{target}} \leftarrow d + \mathcal{N}(0, \sigma_d\mathcal{I})$
15 $\Sigma \leftarrow \sigma_{\text{init}}\mathcal{I}$
16 Sample $\Pi = \{\pi_1, \ldots, \pi_P\} \sim \mathcal{N}(\pi_\mu, \Sigma)$
17 $\Pi_1,\ \Pi_2 \leftarrow \Pi[:50\%],\ \Pi[50\%:]$
18 $steps$ += `rollout` $(\Pi_1, \mathcal{R}, \mathcal{A})$
19 **if** *ready to update critic* **then**
20 `DC_TD3` (*copy of each* $\pi \in \Pi_2$, N)
21 Sample transitions $\tau = (s, a, r, s', d, d') \sim \mathcal{R}$
22 **Do** k gradient ascent steps/agent $\forall \pi \in \Pi_2$ using τ
23 $steps$ += `rollout` $(\Pi_2[:80\%], \mathcal{R}, \mathcal{A}, d)$
24 $steps$ += `rollout` $(\Pi_2[20\%:], \mathcal{R}, \mathcal{A}, d_{\text{target}})$
25 $(\pi_\mu, \Sigma) \leftarrow$ `CEM_Update`$(\Pi, P/2)$
26 **Function** `rollout` $(\Pi, \mathcal{R}, \mathcal{A})$, *$[d']$*:
27 $steps \leftarrow 0$
28 **foreach** $\pi \in \Pi$ **do**
29 $(f, d, T, s) \leftarrow \mathcal{F}(\pi)$
30 Insert (π, f, d) into $\mathcal{A}$
31 $d' \leftarrow d' \vee d$ `// Fallback to` d `if` d' `unset`
32 Insert $T_n \cup (d, d')$ into $\mathcal{R}_n$ for $n = 1, \ldots, N$
33 $steps$ += s
34 **return** $steps$

3.3 Exploration with CEM

MASQD samples a population Π of P team policies from the sampling distribution $\mathcal{N}(\pi_\mu, \Sigma)$ (line 16). This population is divided into two equal halves (line 17): the first half, Π_1, is evaluated directly (line 18), while the second half, Π_2, is first refined using the descriptor-conditioned critic (Section 3.4) and then evaluated (lines 21-24). All evaluated teams contribute to the archive and the subsequent update of the sampling distribution. Each evaluated team's policy, fitness, and behavior descriptor are inserted into the behavior archive $\mathcal{A}$, and their trajectories are stored in the replay buffers $\mathcal{R}$. The top-performing $P/2$ policies (based on fitness) are then used to update the distribution parameters (line 25). Specifically, the CEM update computes a weighted average of the top policies to update the new mean π_μ and covariance Σ, as described in Algorithm 2.

This update balances exploitation of high-performing solutions with continued exploration: by adapting the sampling distribution based on top-performing teams, MASQD gradually concentrates search in productive regions of the policy space while maintaining enough variance to discover new behaviors.

3.4 Refinement via Descriptor-Conditioned Critic

MASQD uses a descriptor-conditioned critic to refine team policies to more precisely express target behaviors (with descriptor d_{target}; line 14). This mechanism enables targeted exploitation of the behavior space: when a behavior descriptor d_{target} is sampled during re-anchoring (Section 3.2), policies in the second half of the population are refined using the critic to express behaviors near this anchor.

The critic is periodically updated after a fixed number of new transitions have been collected (lines 19-20). During training, it receives batches of transitions (s, a, r, s', d, d'), where d is the descriptor of the behavior expressed during the rollout, and d' is the intended descriptor. A similarity score $\sigma = S(d, d') = \exp(-\|d - d'\|/l)$ scales the reward, reducing the influence of mismatched behaviors. The critic is trained to minimize the regression loss over these Q-values. Every Δ iterations, copies of actors from Π_2 perform gradient ascent to maximize predicted Q-values under target descriptors d' from sampled transitions. These updates improve the descriptor-conditioned critic and are not retained (Algorithm 1). The updated critic, now conditioned on the anchor's target descriptor d_{target}, is then used to refine each actor in Π_2 via k steps of gradient ascent per agent (Algorithm 3, line 22). This refinement process allows policies sampled from the anchored distribution (Section 3.2) to improve performance on d_{target} and discover new, nearby behaviors that extend the local region of the archive.

Training the critic requires both positive and negative examples. A sample is treated as *positive* when the descriptor d of the expressed behavior is assumed to match the intended target descriptor d_{target}, and as *negative* when it does not. To ensure both types are present, MASQD splits transitions from the refined population into two batches. For the first 80%, the target is retroactively set to the behavior actually expressed ($d_{\text{target}} \coloneqq d$), providing the critic with positive examples that reinforce the current policy's behavior (line 23). For the remaining 20%, the original target d_{target} is retained, allowing the critic to see negative examples where the expressed behavior, characterized by descriptor d, may deviate from the intended behavior (line 24). These labeled transitions are inserted into the agent-specific replay buffers $\mathcal{R}_n$.

Through simultaneous exploration guided by a re-anchored sampling distribution and refinement using a descriptor-conditioned critic, MASQD builds a diverse archive of multiagent teams capable of adapting across a spectrum of environmental contingencies.

4 Experimental Setup

4.1 Compared Baselines

We compare MASQD against two baselines:

- **Mix-ME**: the current state-of-the-art multiagent QD method. We reimplement Mix-ME based on the original paper [23], as the provided code relies on unavailable dependencies. Hyperparameters are kept consistent with the original implementation when possible. Mutation rates are tuned to improve performance in our setup, as MIX-ME's JAX-based [6] environment has different episode lengths and only positive timestep-level rewards.
- A multiagent extension of **DCG**, a single-agent QD method that uses evolutionary operators and a descriptor-conditioned critic [12]. We choose DCG to isolate the contribution of descriptor-conditioned training in a multiagent setting. To extend DCG to multiagent settings, we use the same multi-head actor (MHA) architecture as MASQD and follow the same critic training procedure as in Algorithm 1. All hyperparameters are retained from the original DCG [12], with the critic batch size increased from 100 to 256, which yields improved performance in multiagent settings.

We also perform two additional ablations of MASQD to isolate key design choices. **Regular critic (RC)**, which replaces the descriptor-conditioned critic with a standard critic conditioned only on state-action pairs. This ablation tests whether conditioning on behavior descriptors is necessary for local expansion in the behavior space. **NoRL**, which removes reinforcement learning entirely. The full population is evaluated without refinement, isolating the role of reinforcement learning in enabling sample-efficient and behaviorally diverse policy improvement.

4.2 Evaluation Metrics

We evaluate each algorithm using three standard metrics from the QD literature to gauge both the quality and diversity of solutions in the archive: (1) **QD Score**, the sum of fitness values across all policies in the archive. This metric accounts for *functional diversity* as poor-performing policies act as a penalty to the QD Score, (2) **Archive Size**, which is the proportion of the behavior space occupied by discovered solutions. This measures each method's ability to effectively explore and cover diverse behaviors, and (3) **Maximum Policy Fitness**, the highest fitness attained by any policy in the archive, which captures each method's ability to optimize its solutions.

4.3 Environment

We evaluate MASQD, DCG, and Mix-ME across four multiagent continuous control tasks from the partially observable multiagent MuJoCo (MaMuJoCo) suite [31]: Half Cheetah-2x3 (2 agents), Walker2d-2x3 (2 agents), Hopper-3x1 (3 agents), and Half Cheetah-6x1 (6 agents). In all tasks, a single robot is partitioned into multiple components, each controlled by a separate agent. Agents must coordinate to propel the robot forward as far as possible over 1000-timestep episodes. In Walker2d and Hopper, episodes terminate if the robot falls. In both Half Cheetah variants, however, episodes persist for the full 1000 timesteps regardless of robot failure. Consequently, sample-efficiency is especially crucial in Half Cheetah as agents may spend many timesteps in uninformative or unrecoverable states.

To assess adaptability, we evaluate the behavior archives produced by each method in a suite of few-shot adaptation tasks, which introduce previously unseen environmental perturbations including altered gravity, leg dysfunction, and steep terrain. No retraining is allowed, allowing us to test the robustness of learned behaviors under unseen environmental contingencies.

The behavior descriptor used across all environments captures the robot's gait, defined as the average ground contact time for each foot over an episode. Formally, it is calculated as:

$$bd = \frac{1}{T}\sum_{t=1}^{T}\begin{pmatrix}\mathbb{I}[\text{foot 1 touches ground}]\\ \vdots \\ \mathbb{I}[\text{foot X touches ground}]\end{pmatrix} \tag{1}$$

where T is the episode length, X is the the number of robot feet (2 in all tasks except Hopper which has 1), and $\mathbb{I}$ is an indicator function that returns 1 when a foot is touching the ground and 0 otherwise.

4.4 MASQD Learning Parameters

In this section, we include MASQD parameters that appear in Algorithm 3, and include all MASQD and baseline parameters in the Appendix. The population size P we use is 10, based on previous work [33]. The warm-up phase is performed for $\omega = 5$ iterations, in which 50 team policies ($P * \omega$) are evaluated, which is consistent with the number of evaluations DCG and Mix-ME perform in their warm up phases. The covariance matrix of CEM, Σ, is initialized with $\sigma_{init} = 0.001$, consistent with CEM-RL [33]. A new policy and its corresponding behavior, d, are sampled every $resample = 5$ iterations, and the d is perturbed by $\sigma_d = 0.0004$ to produce d_{target}, consistent with DCG [12]. We have empirically seen that increasing or decreasing this value increases or decreases the level of exploration our method performs, respectively. Half of the population is refined using the descriptor-conditioned critic for $k = 150$ steps of gradient ascent per agent, consistent with DCG [12].

5 Results

5.1 Quality-Diversity Performance Curves

Figure 1 demonstrates that MASQD consistently achieves the highest QD Score and maximum fitness across all environments. Although Mix-ME discovers a greater number of unique behaviors, MASQD's exploration remains competitive, matching Mix-ME in archive size on Hopper and both Half Cheetah variants, and equaling DCG in Walker2d (within the margin of standard error). This indicates that MASQD's exploration is always on par with a baseline *while* also producing higher-performing policies.

Interestingly, the multiagent extension of DCG outperforms Mix-ME in QD Score on all environments except Hopper, where Mix-ME performs marginally better. This discrepancy is likely due to the reduced dimensionality of the behavior space in Hopper, from 2D (10,000 behaviors) to 1D (100 behaviors), which increases the likelihood of crossover-based strategies forming high-performing teams. This interpretation is supported by the similar maximum policy fitness scores achieved by both Mix-ME and DCG in Hopper, suggesting that Mix-ME's team-level variation performs comparably to DCG's critic-guided policy refinement in this setting.

5.2 Behavior Archives

To capture the distribution of learned behaviors in the archive, we use a heat map to visualize the behavior space of each method. Each square that is filled represents a stored policy that exhibits the behavior with its color representing the policy's fitness. For each method, we visualize the archive that achieves the highest QD Score.

The first column of Figure 2 is a behavior space visualization for 2 agent Half Cheetah. MASQD's archive contains the highest-performing policies, with nearly every policy being highly-optimized, (these cells are illuminated with warmer colors). Moreover, MASQD's behaviors are spread across the archive, providing several options for adaptation. In comparison, DCG produces a fewer number of high-performing policies which are concentrated in the bottom left corner of the archive. The remaining policies in the archive produce fitnesses closer to 0. Mix-ME's behavior archive contains the lowest performing policies, but covers the widest range of behaviors. However, for effective adaptation, higher-performing solutions would be necessary.

The Walker2d visualizations are displayed in the second column of Figure 2. We see once again that MASQD contains the highest performing policies found in a large cluster. Although all methods find similarly performing policies in the top right corner, MASQD discovers (with DCG closely following behind) more policies with better performance around that region, displaying the descriptor-conditioned critic's ability to iteratively perform local expansion in promising regions of the behavior space. Mix-ME fills nearly all cells in the archive, uniformly exploring the behavior space. But in spite of its superior exploration, its QD Score remains the lowest among the three as its team crossover operator randomly samples teammates rather than directly optimizing its current solutions like DCG or MASQD. It is important to note that the behavior descriptor is a low-dimensional characterization of a policy's behavior and cannot capture several latent variables. As such, while all methods discover the same behaviors in the bottom left corner of the maps, MASQD's policies exhibit higher fitness.

In Hopper, which use a 1-dimensional behavior descriptor, MASQD discovers the highest performing region around 0.4 with another strong cluster around 0.8. All methods discover the same behaviors in the archive, but DCG finds better performing policies near 0.2 in comparison to MASQD and Mix-ME. Both DCG and Mix-ME receive very similar QD Scores in Hopper, while MASQD receives the highest through its refinement of policies around 0.4. Interestingly, all methods discover a gait associated with 0.0, which means the robot's leg never touches the ground. However, it is impossible to optimize such a behavior since the episode terminates as soon as the hopper falls, so all methods have a low performing solutions there.

The behavior archives for the 6-agent variant of Half Cheetah are presented in the last column of Figure 2. Even in this setting, which requires tight coordination, MASQD discovers the highest performing policies and forms clusters around these areas. DCG discovers higher-performing policies than Mix-ME, and their behaviors are located in the same general vicinity. Crucially, MASQD instead hones in on and fine-tunes a region of the behavior space that neither Mix-ME nor DCG discovers, while also discovering high fitness policies in the same areas as DCG and Mix-ME. MASQD also discovers a similar percent of the behavior space compared to Mix-ME, suggesting that the informed exploration performed by CEM is robust to the number of agents on a team.

5.3 Ablations

Table 1. Ablation Studies of MASQD

Environment	Method	QD Score (1e6)	Max Fitness (1e3)	Archive Size (%)
Half Cheetah (2x3)	MASQD	**3.94±0.12**	3.12±0.19	**35.8±1.46**
	RC	2.60±0.29	**3.96±0.19**	14.2±1.90
	NoRL	1.73±0.12	1.82±0.14	27.7±1.74
Walker2d (2x3)	MASQD	**5.43±0.12**	**3.83±0.096**	**71.4±2.14**
	RC	2.90±0.22	3.08±0.32	39.9±3.77
	NoRL	2.30±0.072	1.13±0.0089	69.0±3.27

We perform ablation studies in Half Cheetah 2x3 and Walker2d 2x3 to show the importance of the descriptor-conditioned critic by comparing MASQD against **RC** where a regular critic is used instead and capture CEM's ability to explore promising regions of the behavior space via **NoRL** where we remove RL training. Table 1 shows the QD Score, maximum policy fitness, and archive size of each method on both environments.

In Half Cheetah, MASQD receives the highest QD Score and Archive Size, demonstrating that the conditioned critic and CEM play important roles in refining existing solutions and exploration,

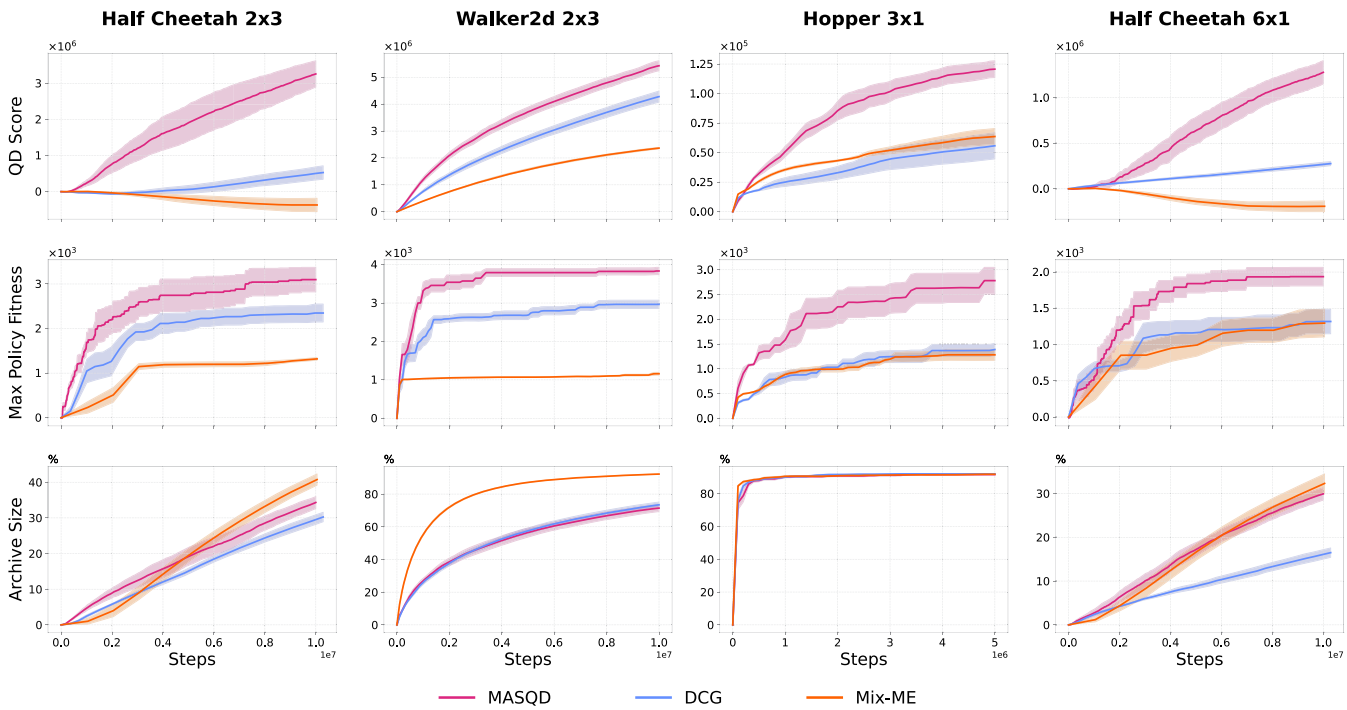

Figure 1. QD Score, maximum fitness, and coverage of MASQD, DCG, and Mix-ME across four environments. Each curve represents the mean over five random seeds, with shaded regions indicating standard error.

respectively. We see that RC produces a policy with a higher fitness than MASQD, but this is expected. However, balancing both quality and diversity is essential to produce an archive for effective adaptation—which is exactly what MASQD does. The critic in RC does not take behaviors into account and is instead solely optimizing for fitness. Coupled with the considerably smaller archive, RC is more likely to sample a high-performing solution and optimize it to achieve a higher fitness through the regular critic. While NoRL has the lowest QD Score compared to MASQD and RC, its overall performance is comparable to DCG with a higher QD Score and only marginally lower max fitness and archive size than DCG. Crucially, MASQD discovers approximately 8% more of the behavior space than NoRL even though CEM is designed for exploration. MASQD's added exploration comes from its iterative application of the descriptor-conditioned critic to expand learned regions of the behavior space with perturbed behavior descriptor targets.

We confirm our findings in Half Cheetah by performing an ablation study in Walker2d which yields similar results. In Walker2d, we find that MASQD performs best across all metrics: **QD Score**, **Max Fitness**, and **Archive Size**. In Walker2d, discovering new behaviors is easier than in Half Cheetah because episodes terminate when the robot falls. Agents can learn to keep the robot standing more quickly, while in Half Cheetah the robot can lay on its back and move, which is not captured by the behavior descriptor in Equation 1. As a result, while NoRL and MASQD's archive coverage is comparable, MASQD is still able to get roughly two percent more coverage through descriptor-conditioned training's local expansion, which can also be seen in the behavior space visualization in Figure 2. Also, the max fitness of MASQD is higher than RC, indicating that the iterative optimization of conditioning on a target descriptor can be even more effective than a regular critic in some settings.

5.4 *Adaptability*

We test each method's adaptability to several contingencies, including modifications to the environment's gravity, terrain angles, and partial damage to the robot's leg (Figure 3).

In Hopper, we vary the angle of the ground from -5 degrees (steepest incline) to 5 degrees (steepest decline) and add a scaling factor from 1.0 to 5.0 to the robot's foot joint, producing dysfunction in the joint. As seen in column 3 of Figure 2, all methods discover the same behaviors, but optimize different regions. MASQD remains the most robust, gracefully adapting to the environmental perturbations. This shows that policies from the region better refined by MASQD (0.3-1.0), are the most useful for adaptation in these settings. In the increasingly angled incline, agents must simultaneously account for the angle and cover as much distance as possible, proving to be extremely challenging. Even in the steepest decline, MASQD produces policies with fitnesses over 500 more than the baselines. MASQD similarly outperforms DCG and Mix-ME across all Hopper leg dysfunction levels.

In Walker2d, we add a scaling factor to the robot's left foot and leg joints from 0.0 to 4.0. Across all multipliers, MASQD outperforms DCG and Mix-ME. From 1.0 to 2.0, while other methods stay constant or drop slightly, MASQD's performance improves, demonstrating its resilience to damage and environmental changes.

In the 6-agent variant of Half Cheetah, the gravity of the environment is increased by multipliers of 1.0 to 5.0. MASQD remains the most robust to gravity changes, beginning and ending with the highest performance, and outperforms DCG across all multipliers. Mix-ME performs well, outperforming DCG and performing slightly better than MASQD at a multiplier of 3. This highlights the importance of discovering several different behaviors and that adaptation is not a 1-to-1 mapping: even though Mix-ME and MASQD focus on differ-

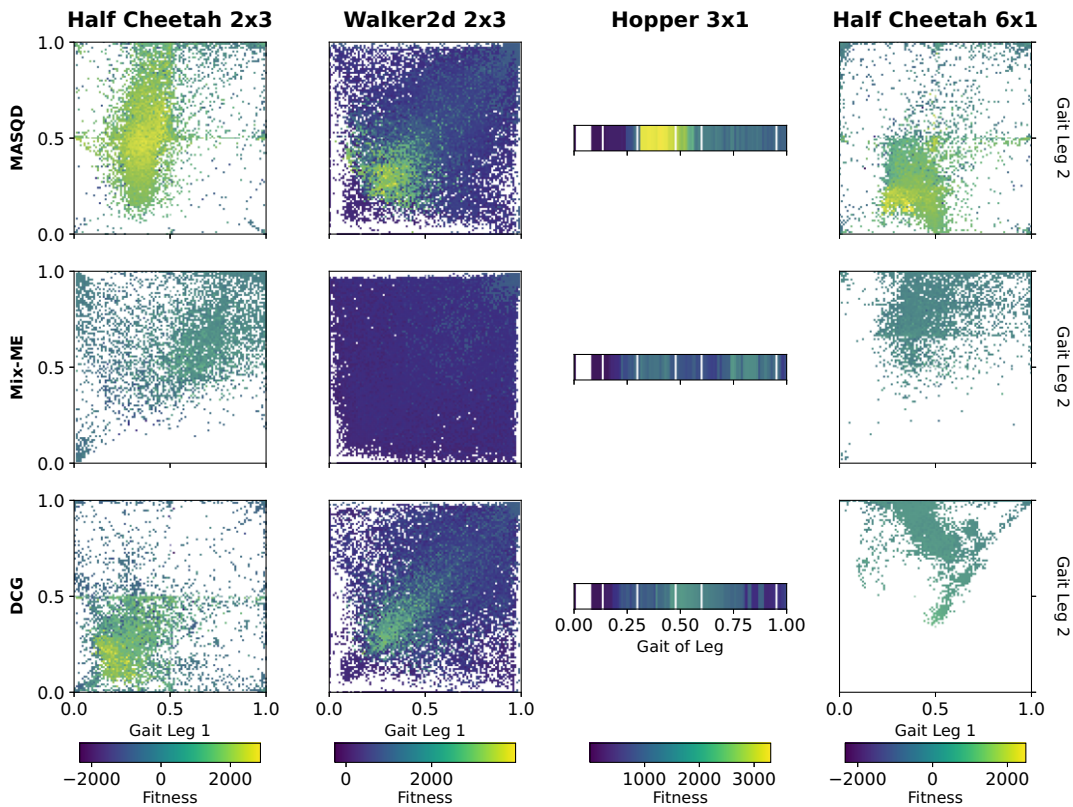

Figure 2. Behavior Archives of all methods on Half Cheetah 2x3, Walker2d 2x3, Hopper 3x1, and Half Cheetah 6x1. The first row is MASQD, second is Mix-ME, and third is DCG.

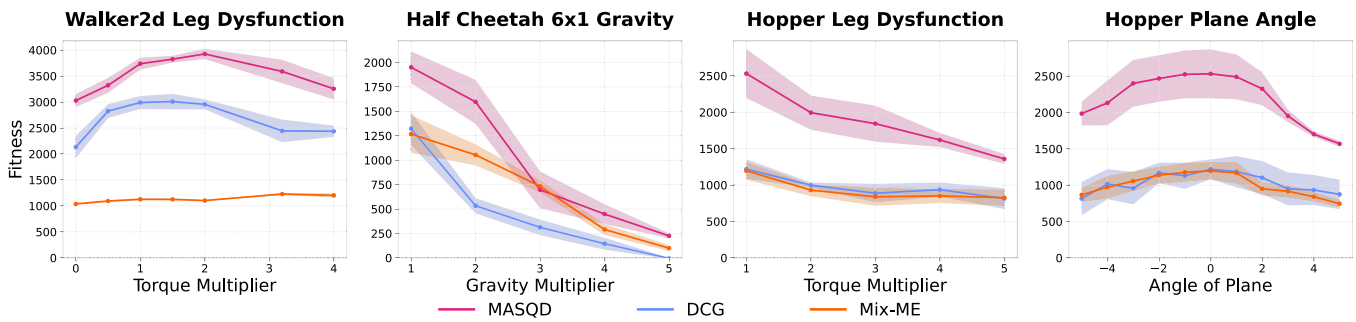

Figure 3. Adaptation experiments in Hopper, Half Cheetah 6x1, and Walker2d

ent areas of the behavior space, both produced similarly performing policies to this contingency.

6 Discussion

This work introduces Multiagent Sample-Efficient Quality-Diversity (MASQD), a learning framework that produces a population of high-performing and diverse teams. MASQD extends Cross-Entropy Method Reinforcement Learning (CEM-RL) to simultaneously promote *quality* and *diversity* by: 1) periodically re-anchoring exploration around known behaviors, facilitating local expansion of the behavior archive, and 2) using a descriptor-conditioned critic to refine policies toward high performance while preserving target behaviors. These mechanisms allow MASQD to adapt to environmental changes through an archive which contains a wide spectrum of functional and diverse team behaviors.

In this work, as in much of QD literature, the behavioral archive is defined using a low-dimensional descriptor derived from domain knowledge. In multiagent settings, capturing the full richness of team coordination with pre-defined descriptors can be challenging. As future work, we plan to investigate higher-dimensional behavior spaces that can be inferred based on the task and team dynamics. In such higher-dimensional behavior spaces, a single critic may struggle to generalize across the full diversity of behaviors. We propose training *local critics*, each specialized to sub-regions of the behavior space, enabling targeted and effective credit assignment for diverse teams.

Acknowledgements

This work was partially supported by the Office of Naval Research (ONR) under grant N00014-22-1-2114.

References

[1] E. Alonso, D. Kudenko, and D. Kazakov. *Adaptive agents and multi-agent systems: adaptation and multi-agent learning*, volume 2636. Springer, 2003.

[2] A. A. Aydeniz, R. Loftin, and K. Tumer. Novelty seeking multiagent evolutionary reinforcement learning. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 402–410, 2023.

[3] S. Batra, B. Tjanaka, M. C. Fontaine, A. Petrenko, S. Nikolaidis, and G. Sukhatme. Proximal policy gradient arborescence for quality diversity reinforcement learning. *arXiv preprint arXiv:2305.13795*, 2023.

[4] M. Bettini, R. Kortvelesy, and A. Prorok. Controlling Behavioral Diversity in Multi-Agent Reinforcement Learning, May 2024.

[5] H.-G. Beyer and H.-P. Schwefel. Evolution strategies–a comprehensive introduction. *Natural computing*, 1:3–52, 2002.

[6] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, et al. Jax: composable transformations of python+ numpy programs. 2018. URL https://github.com/jax-ml/jax.

[7] A. Cully and Y. Demiris. Quality and diversity optimization: A unifying modular framework. *IEEE Transactions on Evolutionary Computation*, 22(2):245–259, 2017.

[8] A. Cully, J. Clune, D. Tarapore, and J.-B. Mouret. Robots that can adapt like animals. *Nature*, 521(7553):503–507, 2015.

[9] G. Dixit and K. Tumer. Balancing teams with quality-diversity for heterogeneous multiagent coordination. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 236–239, 2022.

[10] G. Dixit and K. Tumer. Learning inter-agent synergies in asymmetric multiagent systems. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, pages 1569–1577, 2023.

[11] G. Dixit, E. Gonzalez, and K. Tumer. Diversifying behaviors for learning in asymmetric multiagent systems. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 350–358, 2022.

[12] M. Faldor, F. Chalumeau, M. Flageat, and A. Cully. Map-elites with descriptor-conditioned gradients and archive distillation into a single policy. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 138–146, 2023.

[13] M. Faldor, F. Chalumeau, M. Flageat, and A. Cully. Synergizing quality-diversity with descriptor-conditioned reinforcement learning. *ACM Transactions on Evolutionary Learning*, 5(1):1–35, 2025.

[14] J. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[15] M. Fontaine and S. Nikolaidis. Differentiable quality diversity. *Advances in Neural Information Processing Systems*, 34:10040–10052, 2021.

[16] S. Fujimoto, H. Hoof, and D. Meger. Addressing Function Approximation Error in Actor-Critic Methods. In *Proceedings of the 35th International Conference on Machine Learning*, pages 1587–1596. PMLR, July 2018.

[17] L. Grillotti and A. Cully. Unsupervised behavior discovery with quality-diversity optimization. *IEEE Transactions on Evolutionary Computation*, 26(6):1539–1552, 2022.

[18] L. Grillotti, M. Faldor, B. G. León, and A. Cully. Skill-conditioned policy optimization with successor features representations. In *Second Agent Learning in Open-Endedness Workshop*, 2023.

[19] L. Grillotti, M. Faldor, B. G. León, and A. Cully. Quality-Diversity Actor-Critic: Learning High-Performing and Diverse Behaviors via Value and Successor Features Critics. In *Forty-First International Conference on Machine Learning*, June 2024.

[20] S. Hedayatian and S. Nikolaidis. Autoqd: Automatic discovery of diverse behaviors with quality-diversity optimization. *arXiv preprint arXiv:2506.05634*, 2025.

[21] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[22] Y. Huang, S. Wu, Z. Mu, X. Long, S. Chu, and G. Zhao. A multi-agent reinforcement learning method for swarm robots in space collaborative exploration. In *2020 6th international conference on control, automation and robotics (ICCAR)*, pages 139–144. IEEE, 2020.

[23] G. Ingvarsson, M. Samvelyan, B. Lim, M. Flageat, A. Cully, and T. Rocktäschel. Mix-me: Quality-diversity for multi-agent learning. *arXiv preprint arXiv:2311.01829*, 2023.

[24] P. Kent, A. Gaier, J.-B. Mouret, and J. Branke. Bop-elites, a bayesian optimisation approach to quality diversity search with black-box descriptor functions. *arXiv preprint arXiv:2307.09326*, 2023.

[25] S. Khadka and K. Tumer. Evolution-Guided Policy Gradient in Reinforcement Learning. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[26] J. Kober, J. A. Bagnell, and J. Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013.

[27] S. Majumdar, S. Khadka, S. Miret, S. McAleer, and K. Tumer. Evolutionary reinforcement learning for sample-efficient multiagent coordination. In *International Conference on Machine Learning*, pages 6651–6660. PMLR, 2020.

[28] W. Mao, H. Qiu, C. Wang, H. Franke, Z. Kalbarczyk, R. Iyer, and T. Basar. Multi-agent meta-reinforcement learning: Sharper convergence rates with task similarity. *Advances in Neural Information Processing Systems*, 36:66556–66570, 2023.

[29] E. Marchesini, D. Corsi, and A. Farinelli. Genetic soft updates for policy evolution in deep reinforcement learning. In *International Conference on Learning Representations*, 2021.

[30] F. A. Oliehoek, C. Amato, et al. *A concise introduction to decentralized POMDPs*, volume 1. Springer, 2016.

[31] B. Peng, T. Rashid, C. Schroeder de Witt, P.-A. Kamienny, P. Torr, W. Boehmer, and S. Whiteson. FACMAC: Factored Multi-Agent Centralised Policy Gradients. In *Advances in Neural Information Processing Systems*, volume 34, pages 12208–12221. Curran Associates, Inc., 2021.

[32] T. Pierrot, V. Macé, F. Chalumeau, A. Flajolet, G. Cideron, K. Beguir, A. Cully, O. Sigaud, and N. Perrin-Gilbert. Diversity policy gradient for sample efficient quality-diversity optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '22, pages 1075–1083, New York, NY, USA, July 2022. Association for Computing Machinery. ISBN 978-1-4503-9237-2. doi: 10.1145/3512290.3528845.

[33] A. Pourchot and O. Sigaud. Cem-rl: Combining evolutionary and gradient-based methods for policy search. *arXiv preprint arXiv:1810.01222*, 2018.

[34] C. Qian, K. Xue, and R.-J. Wang. Quality-diversity algorithms can provably be helpful for optimization. *arXiv preprint arXiv:2401.10539*, 2024.

[35] S. D. Ramchurn, T. D. Huynh, F. Wu, Y. Ikuno, J. Flann, L. Moreau, J. E. Fischer, W. Jiang, T. Rodden, E. Simpson, et al. A disaster response system based on human-agent collectives. *Journal of Artificial Intelligence Research*, 57:661–708, 2016.

[36] R. Rubinstein. The cross-entropy method for combinatorial and continuous optimization. *Methodology and computing in applied probability*, 1:127–190, 1999.

[37] A. Salehi, A. Coninx, and S. Doncieux. Few-shot quality-diversity optimization. *IEEE Robotics and Automation Letters*, 7(2):4424–4431, 2022.

[38] G. Smith, J. W. Sanders, and K. Winter. Reasoning about adaptivity of agents and multi-agent systems. In *2012 IEEE 17th International Conference on Engineering of Complex Computer Systems*, pages 341–350. IEEE, 2012.

[39] D. Wierstra, T. Schaul, J. Peters, and J. Schmidhuber. Natural Evolution Strategies. In *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pages 3381–3387, Hong Kong, China, June 2008. IEEE. ISBN 978-1-4244-1822-0. doi: 10.1109/CEC.2008.4631255.

[40] D. Wolpert. Theory of collective intelligence. In *Collectives and the design of complex systems*, pages 43–106. Springer, 2003.

[41] C. Yu, A. Velu, E. Vinitsky, J. Gao, Y. Wang, A. Bayen, and Y. Wu. The surprising effectiveness of ppo in cooperative multi-agent games. *Advances in neural information processing systems*, 35:24611–24624, 2022.