Shaping the Behavior Space with Counterfactual Agents in Multi-Objective Map Elites

Anna Nickelson, Nicholas Zerbel, Gaurav Dixit, and Kagan Tumer

Collaborative Robotics and Intelligent Systems (CoRIS) Institute, Oregon State University, Corvallis, OR, USA

{*nickelsa, zerbeln, dixitg, kagan.tumer*}*@oregonstate.edu*

Keywords: Counterfactuals, Quality Diversity, Multi-Objective, Evolutionary Learning

Abstract: Success in many real-world problems cannot be adequately defined under a single objective and instead requires multiple, sometimes competing, objectives to define the problem. To perform well in these environments, autonomous agents need to have a variety of skills and behaviors to balance these objectives. The combination of Multi-Objective Optimization (MOO) and Quality Diversity (QD) methods, such as in Multi-Objective Map Elites (MOME), aim to provide a set of policies with diverse behaviors that cover multiple objectives. However, MOME is unable to diversify its search across the behavior space, resulting in significantly reduced coverage of the global Pareto front. This paper introduces Counterfactual Behavior Shaping for Multi-Objective Map Elites (C-MOME), a method that superimposes counterfactual agents onto the state space of a learning agent to more richly define the diversity of agent behaviors. Counterfactuals explicitly introduce new forms of diversity in agent behaviors, resulting in C-MOME's effective coverage of behavioral niches; this provides a broader set of Pareto optimal policies. We show that C-MOME covers more than twice as much of the behavior space compared to MOME while increasing the hypervolume of the global Pareto front by over 40%.

1 INTRODUCTION

When optimizing over a single objective, learning-based approaches seek to find the best solution available that represents the global optimum. Although some problems can be adequately defined under a single objective, many real-world problems are complex and define success using multiple competing objectives that need to be balanced. For example, engineering design tasks often aim to maximize safety while minimizing costs (Coello Coello, 2006) and in multi-robot exploration tasks, designers try to maximize exploration while minimizing energy consumption, travel time, or distance (Burgard et al., 2005; Alitappeh and Jeddisaravi, 2022). While many Multi-Objective Optimization (MOO) algorithms have been introduced to find well-balanced solutions to these problems, they typically consider objective trade-offs under a particular set of operating conditions. For long-term deployments and dynamic environments, autonomous agents need to be able to adapt to changes in the environment while continuing to balance multiple goals. This is a challenging problem for agents as they need to learn what tasks to accomplish, how to accomplish the tasks, how different tasks correlate with different objectives, and what the best balance between objectives is for the current system conditions.

One way to provide a stepping stone to this level of reasoning is to train a diverse set of policies or skills for the agent to enact in different scenarios. Quality Diversity (QD) methods can be used to address this training step as they intentionally diversify the search by focusing on policies that cover a wide range of behaviors in order to provide a breadth of policies (Mouret and Clune, 2015). Popular QD approaches, such as Multi-dimensional Archive of Phenotypic Elites (MAP-Elites) (Mouret and Clune, 2015), aim to find policies that fill different behavioral niches, but these methods still typically focus on finding policies that solve for a singular objective. Multi-Objective Map Elites (MOME) addresses this issue by maintaining a local Pareto front within each behavioral niche to cover multiple objectives (Pierrot et al., 2022). However, MOME is unable to fully diversify its search across the behavior space, leading to poor coverage across the Pareto front and behavior space.

In this work we propose Counterfactual Behavior Shaping for Multi-Objective Map Elites (C-MOME).

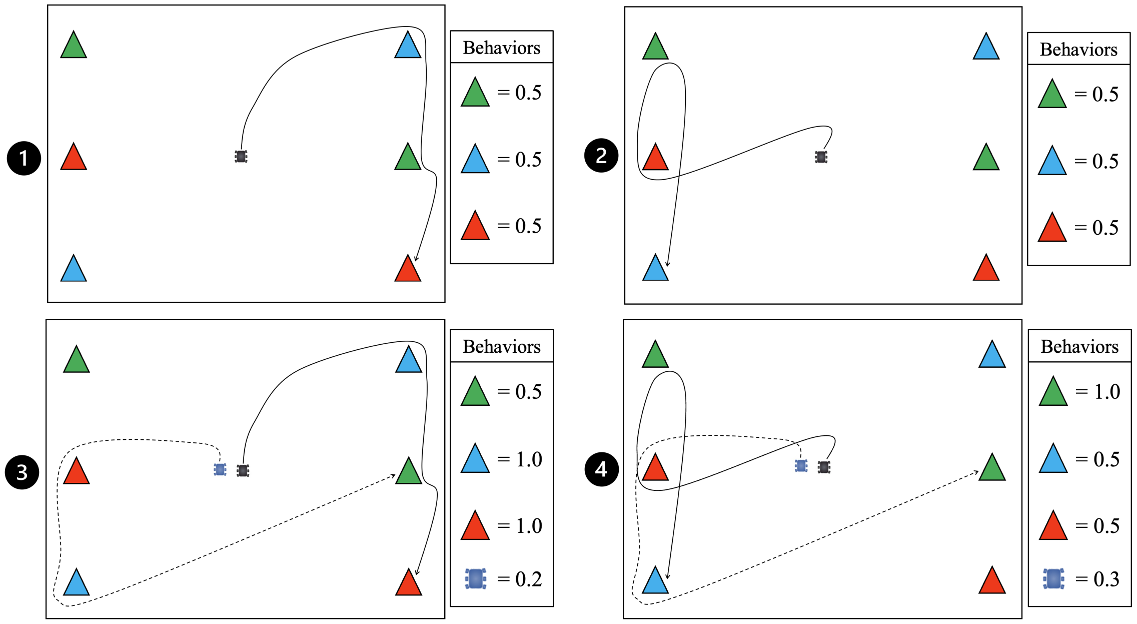


Figure 1: In sub-figures (1) and (2), we show two distinct agent behaviors resulting from different policies in a multi-objective tasks observation problem. In (1) and (2) behaviors are defined by the percentage of each tasks type an agent observes. However, this definition of behaviors does not effectively capture the distinctiveness of the two agent behaviors. By injecting a counterfactual agent, we redefine the learning agent's behaviors to more accurately capture this diversity. In this scenario, adding a counterfactual agent also adds a new behavioral measure that describes tasks exploration overlap between agents. Applying these counterfactual behaviors we see that agent behavior (1) can be redefined as (3) and agent behavior (2) can be redefined as (4). While MOME would sort the policies for behaviors (1) and (2) into the same behavioral niche, C-MOME sorts these policies into separate niches. This helps C-MOME diversify its search across both the behavior and objective spaces.

C-MOME superimposes counterfactual states containing agent representations onto the state space of learning agents. Counterfactual agents inject structured dynamics into agent behaviors, which enables a broader definition of diversity within the behavior space of MAP-Elites. This enables C-MOME diversify its search across the behavior space; in turn, C-MOME provides significantly greater coverage of effective policies along the global Pareto front and across the behavior space.

To illustrate how counterfactuals redefine agent behaviors, we consider a multi-objective observation task within a confined environment involving multiple types of targets. In this example, the behavior space is defined by the types of targets the agent visits and how it chooses to interact with them. As in many QD problems, this definition is constrained by the environment. As a result, MOME is unable to effectively expand its search across the entire behavior space. This constricts down the search resulting in a lack of coverage of the global Pareto front. By introducing new behaviors defined through counterfactual agents, C-MOME more richly defines agent behaviors and expands the search to cover the entire behavior space. We illustrate this in Figure 1.

The contributions of this work are to:

1. Refine the behavior space of an agent by injecting counterfactual agents into the state space of a learning agent,
2. Provide a greater coverage of policies along the global Pareto front and behavior space in multi-objective problems,
3. Analyze how counterfactuals impact the quality and diversity of agent policies.

In a Multi-Objective tasks observation problem, we show that C-MOME improves coverage of both the behavior space by more than 2x and the global Pareto front by more than 40% compared with MOME. We compare the performance of MOME and C-MOME to other state-of-the-art MOO algorithms (namely Nondominated Sorting Genetic Algorithm II (NSGA-II)). We also demonstrate the impact of different counterfactuals on agent policies by comparing the policies found along the Pareto front with overall coverage of niches within the behavior space.

2 BACKGROUND

In many real-world problems, success is seldom defined under a singular objective. To solve these problems, autonomous agents typically need to learn a wide range of different policies that can balance trade-offs between multiple different objectives in various ways. In this section we discuss several areas of study relevant to this problem including Multi-Objective Optimization and Quality Diversity. We also discuss related works such as MOME and the use of counterfactuals in machine learning.

2.1 Multi-Objective Optimization

Problems that do not have a single, optimal solution and are classified as Multi-Objective Optimization (MOO) problems. Many problems use multiple different objectives to define success, many of which are conflicting. When the preferred balance between objectives is known in advance, the objectives can be scalarized and an optimal solution can be found with traditional single-objective searches (Das and Dennis Jr, 1996; Drugan and Nowe, 2013; Van Moffaert et al., 2013). In many cases, the preferred balance may not be known or may change based on the situation. Most strategies to MOO aim to provide a set of solutions that optimize different trade-offs between objectives. The goal of MOO methods is to find coverage of the optimal region of the objective space, known as the Pareto front, in order to present a set of solutions that provide different balances between objectives.

Evolutionary search-based methods are a natural fit for this problem, as they are able to find, compare, and iterate over a population of policies to find incrementally better solutions. Two of the most prominent evolutionary multi-objective methods are Strength Pareto Evolutionary Algorithm 2 (SPEA2) (Zitzler et al., 2001) and Nondominated Sorting Genetic Algorithm II (NSGA-II) (Deb et al., 2002). NSGA-II uses a non-dominated sorting method for ranking and sorting each solution, and it also includes a measure of crowding distance among solutions to force the search to diversify across the objective space. SPEA2 computes a strength value for each solution found, and it stores stores previously found non-dominated solutions. This set of stored solutions is updated after every generation. The fitness of each solution is calculated based on the number of solutions that dominate it and that are dominated by it.

2.2 Quality Diversity

Where MOO aims to solutions that provide coverage over the objective space, Quality Diversity (QD) seeks find diversified policies that capture a broad set of behaviors. This builds on previous techniques in novelty search that focus on finding substantially different solutions (Eysenbach et al., 2019; Lehman and Stanley, 2011). QD aims to diversify across a pre-defined "behavior space", a reduced-dimensional summary of the actions or behaviors of each policy. QD methods separate the behavior space into many local regions, called niches, then keeps the best policy found in each niche. One notable example of QD methods is Multi-dimensional Archive of Phenotypic Elites (MAP-Elites) (Mouret and Clune, 2015). MAP-Elites is an evolutionary method that aims to find many different solutions for a given task. Specifically, MAP-Elites aims to find a diverse set of policies that do not necessarily solve a task optimally, but solve a task using a variety of unique behaviors.

2.2.1 Multi-Objective Map Elites (MOME):

Multi-Objective Map Elites (MOME) (Pierrot et al., 2022) incorporates the principles of both MOO and QD search by building on the MAP-Elites framework. Instead of a singular policy, MOME keeps the set of policies in each niche that are locally Pareto optimal. In doing so, they are able to provide solutions that are diverse across both the objective and behavior spaces. However, combining these two methods leads to extensive time spent searching and optimizing in unhelpful or sub-optimal regions of the behavior space. By keeping locally Pareto optimal policies, this method keeps many policies that result in no reward as there are many niches where zeroes across all objectives is locally Pareto optimal. These policies are then used for mutation and crossover; without another mechanism to add back diversity, the search is weighed down and has a difficult time escaping these local minima.

2.3 Counterfactuals in Agent Learning

Counterfactual thought is a strategy people often use to analyze alternative outcomes to real events and learn from "what-if" scenarios (Boninger et al., 1994; Byrne, 2016). This concept has also been adopted in agent-based learning paradigms, most notably in applications of reward shaping that aim to address the issue of credit assignment within multiagent learning problems. For example, Difference Evaluations compare rewards generated from counterfactual states with the actual rewards agents received to provide individuals with finely-tuned reward feedback in cooperative multiagent learning problems (Agogino and Tumer, 2004). In a similar work, Counterfactual Multi-Agent (COMA) policy gradients rely on a counterfactual baseline in reinforcement learning contexts to single out individual agent actions while keeping the actions of others fixed so more informative rewards can be evaluated (Foerster et al., 2018).

Counterfactuals have also been used in multiagent coordination problems to produce "stepping stone" rewards to make learning easier in situations where task completion depends on simultaneous action inputs from multiple different agents. For example, D++ compares the reward given by the current state of a system with a reward generated from a counterfactual state where *n* partner agents were available to cooperate with a specific individual (Rahmattalabi et al., 2016). Counterfactuals have also been extended beyond reward shaping in applications such as explainability in AI (Verma et al., 2020; Olson et al., 2021), inference and decision making (Johansson et al., 2016; Kusner et al., 2017), and ethics in machine learning (Kasirzadeh and Smart, 2021).

Regardless of the application, counterfactuals in agent-based learning seek to shape available information to provide richer learning signals or more transparent decision making. In this work we apply counterfactuals to quality diversity within multi-objective problems to shape the state and behavior spaces using counterfactual agents.

3 COUNTERFACTUAL BEHAVIOR SPACE SHAPING

In this work, we introduce C-MOME, a novel extension to MOME that superimposes counterfactual agents onto a learning agent's state space and shapes the behavior space used for policy search in MAP-Elites and MOME. By injecting counterfactual agents exhibiting different behaviors into the state space, we more richly define the behaviors of a learning agent. This provides structured diversity at the behavior level enabling substantially better coverage the behavior space. In turn, C-MOME finds a more useful and diverse set of policies along the global Pareto front.

To illustrate how counterfactuals can be used to redefine the behavior space of an agent, we present an example of an agent observing different types of tasks in Figure 1. In this example, we define behaviors as the percentage of each task type observed by an agent. In Figure 1-1 and Figure 1-2, we see two different behaviors encoded within different agent policies. Although the outcomes are similar in terms of observed tasks, the policies exhibit two distinctive agent behaviors. However, based on the definition of the behavior space, these behaviors appear as non-distinct and would be sorted into the same behavioral niche within MOME. By injecting counterfactual agents, we can redefine the behavior space to better capture the differences between agent behaviors (Figure 1-3 and Figure 1-4). Using these redefined behaviors, the differences in the learning agent's behaviors are more distinct allowing C-MOME to sort the underlying policies into more more representative behavioral niches. By providing this form of structured diversity, C-MOME builds a set of Pareto optimal policies that provide a greater coverage along the global Pareto front.

Using counterfactual agents also allows the learning agent to learn different behaviors in a single agent formulation while being sensitive to future learning with real agent partners. This implementation of C-MOME is designed to be the first step in a curriculum learning solution for multiagent teaming, which we discuss in the future work section (Section 6).

3.1 Counterfactual Representation

Counterfactuals are "counter to the facts". They are represented as non-learning agents and exhibit behaviors that impact the state space and re-define the behavior space for the learning agent. For example, counterfactuals can be static or dynamic. The counterfactual agents can also impact the tasks within the state space or have no impact on tasks. A key factor is that counterfactuals *do not* impact any of the agent rewards or the completion of objectives directly. Learning agents merely perceive that counterfactual agents are impacting task completion and act according to their state perception and policies. This limits the impact of the counterfactuals to the state space and behavior space of the agent to improve the fidelity of the policy search within the behavior space.

Counterfactuals impact the agent's perception in three distinct ways. In the state space, counterfactuals can have a direct impact by (1) appearing as agents; they can also indirectly impact it by (2) 'completing' tasks. Finally, counterfactuals can be used to (3) redefine the behavior space to decouple it from a stationary environment. Each of these changes in the agent's perception injects structured diversity to improve the overall policy search.

3.2 Counterfactual State Space

Counterfactuals impact the agent's state space in two ways. First, they are perceived by the learning agent's sensors as agents in the environment. This allows the agent to learn the impact of simple behaviors, such as moving closer to or further from other agents. Counterfactuals can also impact the completeness of tasks in the environment but only through the perceptions of the learning agent. This leads to the definition of the learning agent's state as described in Equation 1 where the state, s, is defined by the ground truth state of the world, s_w, and the superimposed counterfactual state, s_c.

$$s = s_w + s_c \tag{1}$$

Agent sensors provide a summary of the distance, d_a, to any real agents, $a \in A$, within the sensor range in a given region, r, as shown in equation 2. Counterfactual agents, $c \in CF$, are added in the same manner, as shown in the highlighted portion of Equation 2. In this work, we do not consider other non-counterfactual agents, a.

$$d_{rA} = \sum_{a \in r} \frac{1}{d_a + 1} + \sum_{c \in r} \frac{1}{d_c + 1} \tag{2}$$

Agent sensors also summarize the distance, d, and completion percentage, p, for tasks of a given group, $t \in T_g$, in a region, r. This is shown in Equations 3 and 4, respectively. Counterfactuals, $c \in CF$, can impact the completeness of each task, shown in the highlighted region of Equation 4. Once again, the counterfactual agents impact on task completion is only reflected in the learning agent's state space; counterfactual agents cannot directly contribute to the rewards earned by the learning agent.

$$d_{rg} = \sum_{t \in r} \frac{1}{d_t + 1} \tag{3}$$

$$p_{rg} = \sum_{t \in r} p_{ta} + \sum_{t \in r} p_{tc} \tag{4}$$

where $t \in T_g$, $a \in A$, $c \in CF$

Collectively, equations 2, 3, and 4 combine to create the agent's state, s. The highlighted portions of these equations denoted with the subscript c form the counterfactual state, s_c, while the rest makes up the world state, s_w as seen in Equations 6 and 5, respectively. In Equation 5, b is a state variable representing the agent's current battery level. This provides a constraint on the agent, forcing it to be strategic about how it competes tasks.

$$s_w = \left[\sum_{a \in r} \frac{1}{d_a + 1} \quad \left[\sum_{t \in r} \frac{1}{d_t + 1} \quad \sum_{t \in r} p_{ta} \right]_g \quad b \right] \tag{5}$$

$$s_c = \left[\sum_{c \in r} \frac{1}{d_c + 1} \quad \left[0 \quad \sum_{t \in r} p_{tc} \right]_g \quad 0 \right] \tag{6}$$

3.3 Counterfactual Behavior Space

Counterfactual agents enable C-MOME to more richly define agent behaviors allowing for broader behavioral diversity. By incorporating counterfactuals into the state space, agents are able to detect and react to the perceived actions and behaviors of counterfactual agents. The behavior space is defined to capture the diverse ways in which the learning agent can react to these counterfactuals.

Similar to the state space, the behavior space, beh, is defined by the agent's behaviors in the world, beh_w, and the agent's behaviors in relation to the counterfactuals, beh_c.

$$beh = [beh_w, beh_c] \tag{7}$$

The counterfactual behavior space, beh_c, includes summary statistics of the distance from the agent to the counterfactual agents. This is summarized using three different variables: the closest the agent got to a counterfactual, the furthest an agent got away from a counterfactual, and the average distance to all counterfactuals across an episode.

$$beh_c = \left[min(d_c), max(d_c), \bar{d}_c \right] \tag{8}$$

4 EXPERIMENTAL SETUP

In this work, we test C-MOME in a continuous, multi-objective observation problem where an agent must observe different types of targets. Exploration of each type of target is considered a separate task (or objective). While the overall goal is for the agent to complete as many tasks as possible, the agent has to balance different, competing objectives by observing different task types during its mission.

Objectives are defined by how complete tasks are in each task group, T_g. The fitness calculation for each objective is the sum of the completion percentage of each task of a given type as described in Equation 9. In this work, we use two task groups and therefore have two objectives to optimize over. We note that it is not possible for the agent to complete 100% of the tasks within the time provided. This is an intentional quality of the environment, as the agent must be strategic about how to spend its time and balance the tasks it aims to accomplish.

$$F_g = \sum_{t \in T_g} p_t \tag{9}$$

4.1 Learning Agents

Learning agents have two sets of sensors, one set detects agents and the other detects tasks. Agents have four sensors of each type to collect information within four quadrants of a 360 degree circular area centered around the agent. The agent's state includes one input for each agent sensor that detects the distance to other agents in each region. The agent's state also include two inputs for each task sensor that detects task completeness and the distance to tasks; each task type has a separate set of sensors. One additional sensor input defines the current charge of the agent's battery (b). These collectively become the learning agent's state, s_w. The counterfactual state, s_c, is added to the learning agent's state in order to more richly define the agent's behavior. An agent's sensors collect state information defining s_w and s_c (Equations 5 and 6).

The learning agent is represented by a neural network policy with a single hidden layer (30 nodes). Using this policy, the agent selects which region to go toward, which task type to accomplish in that region, how much effort to put into that task, and its velocity, v. A separate navigation policy navigates the agent to the closest task of the chosen type in that region. In this work, we assume the agent has optimized navigation and control policies for travelling between different tasks in the environment. The battery is drained at a rate of the velocity squared, so the agent must balance speed and energy (Equation 10).

The agent outputs two variables that determine how much effort it puts in to the chosen task, e_0 and e_1. The battery, b, drains at the rate of their product (Equation 10), and the product of these two scalars adds to the task's completion percentage, p_t, up to 100% (Equation 11).

$$b' = b - v^2 - (e_0 * e_1) \tag{10}$$

$$p'_t = p_t + (e_0 * e_1) \tag{11}$$

4.2 Behavior Space

The tasks in this problem form the baseline for defining the behavior space used by MOME. The behaviors are direct outputs from the agent's policy, namely the velocity, v, and effort, e_0 and e_1. The behavior space is defined as the mean of each of variable across an episode, separated by task group g. For our implementation, we test with two task groups creating a 6-dimensional behavior space (Equation 12).

$$beh_{no} = beh_w = [[\bar{e}_{0,g}, \bar{e}_{1,g}, \bar{v}_g]\ \forall g] \tag{12}$$

For C-MOME, we reduce the world behavior space (beh_w) to two dimensions by taking the mean of all three variables for each group in (Equation 12). As a result, the behavior space with counterfactuals is 5-dimensional: two task groups in beh_w and three summary statistics in beh_c (Equation 8). We do this to make the behavior space dimensions more closely aligned and comparable in the cases with and without counterfactuals (5 vs 6 dimensions).

$$beh_{CF} = \begin{bmatrix} beh_w \\ beh_c \end{bmatrix} = \begin{bmatrix} [\,\overline{\bar{e}_{0,g}, \bar{e}_{1,g}, \bar{v}_g}\,]\ \forall g \\ min(d_c),\, max(d_c),\, \bar{d}_c \end{bmatrix} \tag{13}$$

4.3 Counterfactual Parameters

To gain a more thorough understanding of how counterfactual agents impact both the behavior space and objective space in C-MOME, we test a number of different parameters that dictate the creation of counterfactual agents. First, we test how different types of counterfactual agents impact performance (Section 5.2). We evaluate C-MOME with Static counterfactual agents that are present in the environment but do not move, Move counterfactuals where counterfactual agents move but do not effect task completion within the learning agent's state space, and Task counterfactuals where counterfactual agents move and impact task completion. Next, we analyze how the number of counterfactual agents impacts performance (Section 5.3). Finally, we test how adding counterfactuals to three different portions of the agent's perception impacts the overall search (Section 5.4). In the state, counterfactuals can (1) impact task completion or (2) appear as agents; they can also impact (3) the definition of the behavior space. In each experiment, we compare against MOME which has no counterfactual agents to form a baseline for comparison.

When there are no counterfactuals (MOME), the state is defined by the true world state, s_w, and the behavior space is defined by the agent's interactions with the world, beh_w (Equations 5 and 12). With counterfactuals (C-MOME), the counterfactual information is injected in the state space, $s_w + s_c$, and the behavior space $beh_w + beh_c$ (Equations 6 and 13) unless otherwise noted.

4.4 Experimental Parameters

We run experiments in the multi-objective exploration problem in a 20x20, 2D continuous environment. The learning agent starts at a randomized position within one unit of the center of the map. The agent can move at up to 2 units per time step; the speed at each time step is determined by the velocity output, v, multiplied by the max speed. Learning agents have a battery life of 30 units at the start of each episode and episodes last for 50 time steps. Counterfactuals move at one unit per time step. Each counterfactual has a randomly generated list of tasks they visit in order.

There are 16 tasks in the environment that are laid out in a circle with a radius of 6 units with alternating types; the position of each task is randomly perturbed by up to 1 unit in the x and y-directions. Agents can interact with any task less than 2 units away, and counterfactual agents can interact with any task up to 1 unit away. All results are averaged across 20 statistical runs unless otherwise stated. Error is reported as the standard error in the mean.

5 RESULTS

In this work, we compare C-MOME against two baseline methods representing the state-of-the-art in multi-objective and QD search: Multi-Objective Map Elites (MOME) (Pierrot et al., 2022) and Nondominated Sorting Genetic Algorithm II (NSGA-II) (Deb et al., 2002). C-MOME builds on the MOME framework by using counterfactual agents to shape the state and behavior spaces of the search. By comparing C-MOME with MOME, we characterize the impact counterfactuals have on both the coverage of behavioral niches within the behavior space and the coverage of policies along the global Pareto front in the objective space. Comparing C-MOME with NSGA-II characterizes the performance of this method against an algorithm that is considered state-of-the-art in multi-objective search. We expand our study on the positive impact of counterfactual states by adding counterfactual agents to the agent's state space in NSGA-II.

5.1 Performance Measures

To test the overall performance of each method, we compare them in both the objective space and the behavior space. The work introducing MOME, (Pierrot et al., 2022), uses the hypervolume of the Pareto front to test the quality of the multi-objective search. The hypervolume measures the volume of the geometric shape created by the points on the Pareto front along with a reference point; we use [0, 0] for our comparisons. Note that in a 2D objective space, this equates to the area of the polygon created by the Pareto front and the origin. For the behavior space, we compare the percent of niches filled by each method. When plots are shown for individual runs, such as for the Pareto front, we chose a run that was closest to the mean.

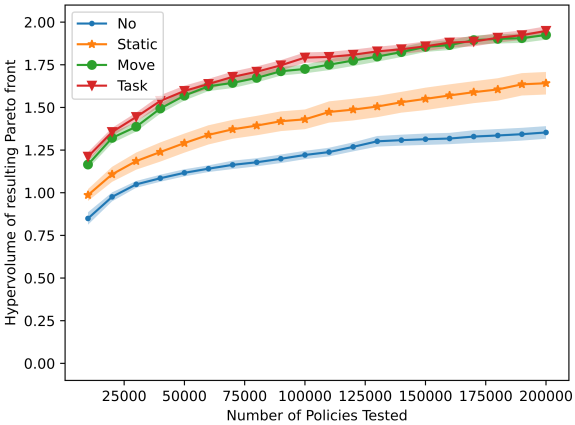
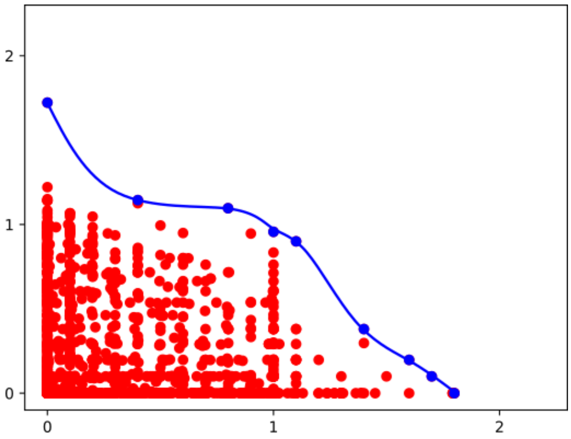


Figure 2: This compares the effects of different types of counterfactual agents on the hypervolume of the Pareto front. Results show the final Pareto hypervolume found after testing 200,000 policies per run, averaged over 20 stat runs. Move and Task counterfactuals result in a greater hypervolume compared to Static counterfactuals and MOME (No cf).

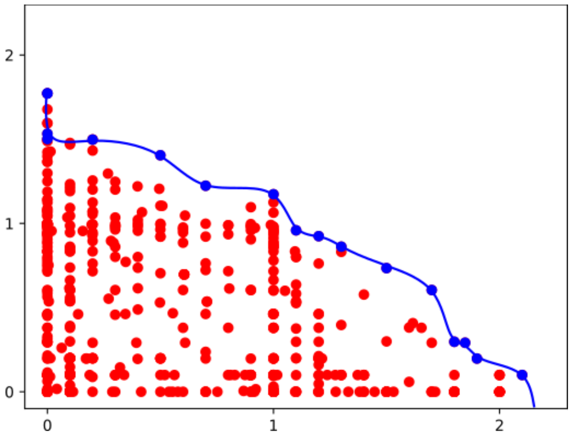
5.2 Counterfactual Types

First, we examine how different types of counterfactuals perform in C-MOME compared with MOME. As outlined in Section 4.3, we test Static, Move, and Task counterfactual types. For this test, we use 9 counterfactual agents for each type. These results are summarized in Figure 2.

The results in Figure 2 demonstrate that Static counterfactual agents have a small impact on performance. This is a positive result, as Static counterfactuals do not express unique behaviors that can effectively redefine the behaviors of a learning agent. This result shows that the change in behavior space definition, beh_{no} to beh_{cf}, and the inclusion of agents in the state provides some benefit. Similarly, we see that the hypervolume impact of Task counterfactuals compared with Move counterfactuals is also negligible. However, we see a noticeable increase in performance between having no counterfactuals or static counterfactuals and counterfactuals that move. The Move and Task counterfactuals both show significant improvement of the final Pareto hypervolume, increasing it by over 40%, each with approximately 3% standard error



(a) MOME Pareto Front



(b) C-MOME Pareto Front

Figure 3: Representative global Pareto fronts found after searching through 200,000 policies for MOME and C-MOME with Task counterfactuals. Policies that are Pareto dominant are represented as blue dots while non-Pareto dominant policies are in red. The counterfactual agents used in C-MOME result in a greater coverage of Pareto optimal policies compared with MOME and keeps significantly fewer policies in unhelpful regions, near [0, 0]. These graphs were chosen as representatives because they are closest to the mean for their population.

after 20 stat runs.

While measuring the hypervolume characterizes the coverage of the Pareto front as a whole, we also show examples of typical Pareto fronts generated by MOME (no counterfactuals) and C-MOME with Task counterfactuals in Figures 3a and 3b, respectively. These plots provide a more detailed view of how the introduction of moving, counterfactual agents can redefine the behavior space resulting in more substantial policies along the global Pareto front. These also highlight how MOME spends significant time searching a sub-optimal region, near $[0,0]$, while C-MOME is able to spread the search across the objective space.

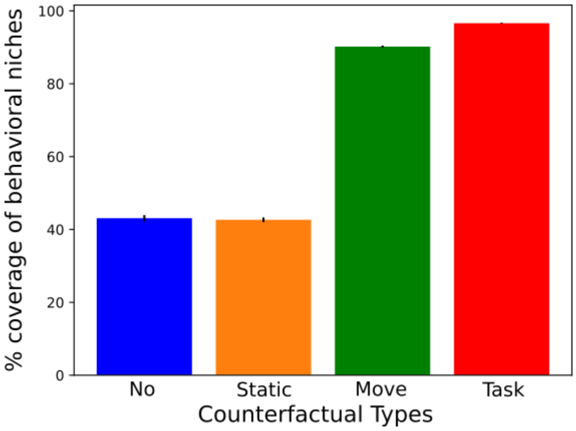


Figure 4: Comparing the effect of different types of counterfactual agents on the coverage of behavioral niches. We show that Move and Task counterfactuals successfully cover more than twice the area of the behavior space, with Task cfs achieving over 95% coverage.

Impact on Behavior Space: To characterize the impact counterfactuals have on the behavior space within MOME, we measure the total number of niches that contain policies. Note that for these results, we do not include counterfactuals in the behavior space and use beh_w for all cases. This is done to make the results directly comparable, as the counterfactual behavior space, beh_c, has significantly different constraints. By comparing the amount of "filled" niches to the total number of niches, we can estimate the total coverage of policies over the behavior space. We present these findings in Figure 4. The total number of niches filled by C-MOME is 2.3x the number of niches filled by MOME (with no counterfactuals). This result shows that C-MOME leverages the structured diversity provided by counterfactuals to provide significantly better coverage of both the behavior and objective spaces.

Compared against both MOME and C-MOME with Static counterfactuals, dynamic counterfactuals improve the overall search in the objective space. By comparing against the Static counterfactuals, we show that the redefinition of the behavior space and inclusion of additional information in the state is not what causes the improvement. The dynamic counterfactuals provide structured variability in the environment, which allows C-MOME to more richly define agent behaviors and effectively cover the behavior space as seen in Figure 4. This diversifies the search across the behavior space, which in turn improves the performance in the objective space.

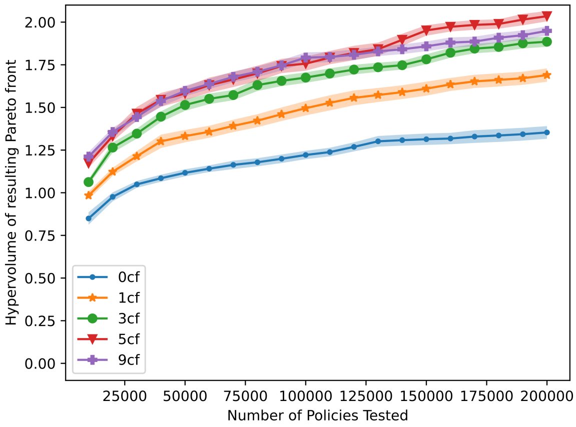


Figure 5: Comparing the impact of different numbers of Task counterfactual agents on the hypervolume of the global Pareto front. Increasing the number of counterfactual agents results in an increase in the total hypervolume. However, the gain in hypervolume becomes less significant as more agents are added beyond 5 agents.

5.3 Number of Counterfactual Agents

In testing the type of counterfactuals used in C-MOME, we found that dynamic counterfactual agents resulted in a marked increase in the hypervolume of the global Pareto front. To further understand the effect counterfactual agents have on learning and shaping the behavior space, we also examine the number of counterfactual agents used. Using Task counterfactuals we tested all cases from 0 to 9 Task counterfactual agents and found that hypervolume increases with more counterfactual agents; however, it does so incrementally. We show a representative sampling of these results in Figure 5 where, for simplicity, we present the results for 0, 1, 3, 5, and 9 counterfactuals.

In Figure 5, we see a larger initial increase in hypervolume when we compare MOME with no counterfactuals against C-MOME with 1 counterfactual. The hypervolume between 1 counterfactual and 5 counterfactuals is also relatively large; however, the rate of hypervolume increase for each added agent diminishes after 5 agents.

Impact on Behavior Space: We see a similar trend when comparing the impact of the number of counterfactual agents on the behavior space coverage. Figure 6 shows a visual representation of two dimensions of the behavior space for 0, 1, and 9 counterfactuals. The No cf case achieves 40.1% coverage of the behavior space, 1 CF achieves 67.8%, and the 9 Task CFs is able to cover 96.6%. We found this trend holds across all tests run from 0 to 9 agents.

5.4 Counterfactuals Shaping Perception

As presented in Section 3, there are three distinct ways in which counterfactuals impact agent perceptions. As part of the state, counterfactuals can impact the (1) percieved task completion or (2) appear as agents. Additionally, counterfactuals can impact (3) the definition of the behavior space. We explored the impact of these three factors on the performance of C-MOME. These results are presented in Figure 7 and Table 1.

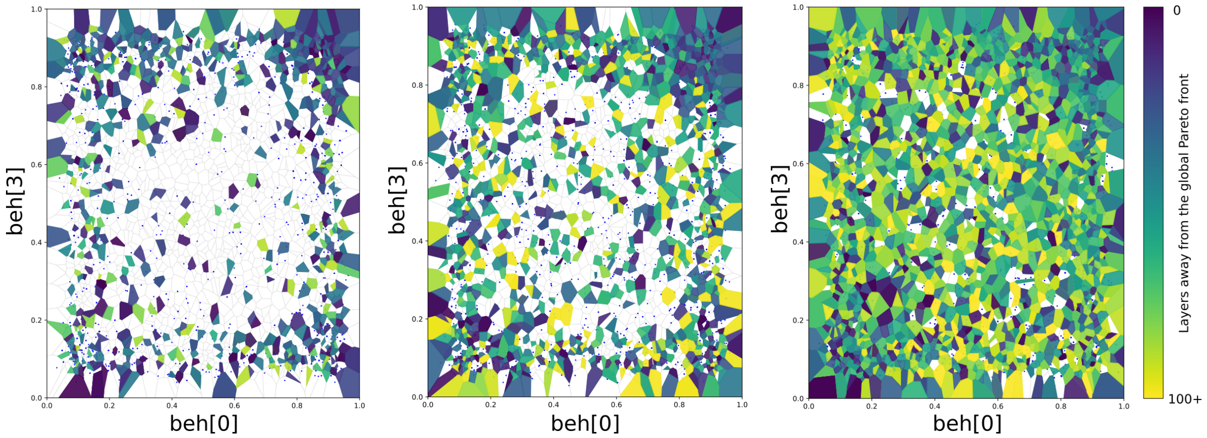
These results show that each of the three components have a moderate impact on the hypervolume of the pareto front. The most surprising result is that the behavior definition has a small impact alone, but when combined with either the task completion or agent state, it has a significantly better impact. Overall we show that the synergy of all three is what provides the best coverage of the objective space, and therefore the best policies.

5.5 Counterfactuals in MOME and NSGA-II

The previous results demonstrate the impact counterfactual agents have on shaping the behavior space in MOME. To gain a better understanding of overall performance within MOO, we also compare MOME, C-MOME, NSGA-II, and NSGA-II with counterfactuals. While the counterfactuals in C-MOME are primarily used to redefine the behavior space, NSGA-II does not rely on a behavior space. Therefore, the counterfactual states we add to NSGA-II only have an impact on the state space of the learning agent. Similar to the counterfactuals in C-MOME, the counterfactual agents do not impact the objectives.

We first explore the effect of counterfactuals in NSGA-II by testing the types of counterfactuals defined in Section 4.3. We plot the effect these counterfactuals have on NSGA-II hypervolume in Figure 8. Similar to the trends observed in C-MOME, we see that dynamic counterfactual agents lead to an increase in the hypervolume of the global Pareto front. However, unlike in C-MOME, the results indicate no change in hypervolume with static counterfactuals.

We also include the hypervolume comparison between MOME, C-MOME, and NSGA-II in Figure 8. Overall we see that NSGA-II outperforms MOME in this task with a substantial increase in the hypervolume of the global Pareto front. We also see that Task counterfactuals in both MOME and NSGA-II leads to a significant increase in hypervolume. It is worth noting that NSGA-II only solves half of the originally stated problem, as it provides diversity across the ob-



(a) No CF Behavior Space (b) 1 CF Behavior Space (c) 9 CFs Behavior Space

Figure 6: Representative behavior spaces found after searching through 200,000 policies for the No CF (Fig. 6a), 1 Task CF (Fig. 6b), and 9 Task CFs (Fig. 6c) cases. Niches that have a policy that is globally Pareto optimal are represented in purple; the color scales to yellow as the policies get further from the global Pareto front. All three cases use the beh_{no} behavior space in order to show a direct comparison. The No cf case achieves 40.1% coverage of the behavior space, 1 CF achieves 67.8%, and the 9 Task CFs is able to cover 96.6%.

Table 1: Hypervolume of the Pareto front and coverage of the niches in the behavior space, broken down by configuration. We present situations with 0 counterfactuals (MOME) or 9 counterfactuals (C-MOME). Note that we do not present niche coverage results for the configurations that have counterfactuals in the behavior space; the behavior space is significantly different and bounded, so the results are not comparable.

	Perception Shaping			Hypervolume		Niche coverage	
Num cf	(1) Task p	(2) Agent	(3) Beh.	Mean	St.Err.	Mean	St.Err
0	-	-	-	1.281	0.079	0.401	0.0069
	-	●	-	1.353	0.037	0.431	0.0084
9	-	-	-	1.298	0.025	0.420	0.0056
	●	-	-	1.636	0.042	0.884	0.0029
	-	●	-	1.709	0.027	0.902	0.0030
	-	-	●	1.414	0.032	-	-
	●	●	-	1.862	0.036	0.966	0.0015
	●	-	●	1.831	0.037	-	-
	-	●	●	1.925	0.028	-	-
	●	●	●	1.949	0.027	-	-

jective space but not across the behavior space.

Although NSGA-II does not use a behavior space for discovering new policies, the process of adding counterfactual agents to the learning agent's state space had a noticeable impact on the hypervolume. This result mirrors the study of the impact of counterfactuals in agent perceptions, outlined in Section 5.4. It also reinforces the power of counterfactuals, given that they do not have a direct impact on reward functions or objective measures. We believe this increase in performance is the result of structured dynamic elements added by the counterfactual agents. The added counterfactual dynamics in the state space helps NSGA-II avoid local minima when assessing the fitnesses of various policies during the policy search.

6 CONCLUSION

In this work we introduce C-MOME, a novel extension to MOME that uses counterfactual agents to better define agent behaviors thus shaping the behavior space. This shaped behavior space enables C-MOME to more richly define agent behaviors, diversify the search in the behavior and objective spaces, and bias the search toward more productive regions. We explored how different types of counterfactual agents can influence both the behavior space within MAP-Elites, and the coverage of policies along the global Pareto front. We found that dynamic counterfactual agents led to an increase in the hypervolume along the global Pareto front, filling over 96% of the niches of the behavior space; MOME is only

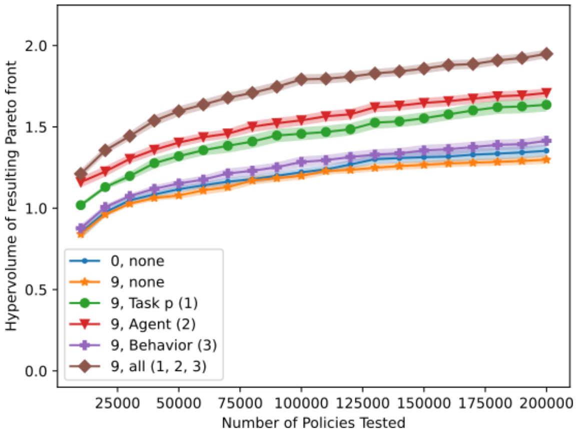


Figure 7: Comparing the impact of including counterfacutals in the learning agents' perceptions. In the state, counterfactuals can (1) impact task completion or (2) appear as agents; they can also impact (3) the definition of the behavior space.

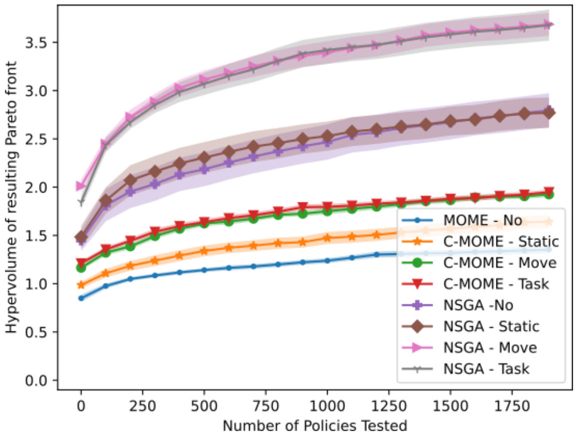


Figure 8: Comparing the effect of different counterfactual types on NSGA-II. Similar to C-MOME, Move and Task counterfactuals have a significant impact on hypervolume. Unlike C-MOME, the static counterfactuals have no impact on hypervolume instead of a minimal impact.

able to achieve 40% coverage. Finally, in comparing MOME and NSGA-II, we found that NSGA-II outperformed this implementation of MOME. However, adding dynamic counterfactual agents improved the performance of NSGA-II despite it relying on policy search through the objective space.

This work demonstrates the impact of adding counterfactual agents into a single agent MOO problem. C-MOME is intended to be the preliminary learning phase in a curriculum style approach to multiagent, multi-objective coordination. By reducing the learning of policies to a single agent problem, we decrease the noise typically associated with a multiagent learning environment, which simplifies learning enough to focus on the challenge of discovering a diverse set of policies that can balance multiple objectives. Although the counterfactual agents are representative of learning in an environment where multiple agents are acting, it cannot fully account for real agents that impact all aspects of the system. Future work will extend C-MOME via curriculum learning with the next step focusing on coordination and team formation.

ACKNOWLEDGEMENTS

This work was partially supported by the National Science Foundation under grant IIS-2112633 and the Air Force Office of Scientific Research under grant No. FA9550-19-1-0195.

REFERENCES

Agogino, A. and Tumer, K. (2004). Efficient evaluation functions for multi-rover systems. In *Genetic and Evolutionary Computation – GECCO 2004*, volume 3102, pages 1–11, Berlin, Heidelberg. Springer Berlin Heidelberg.

Alitappeh, R. J. and Jeddisaravi, K. (2022). Multi-robot exploration in task allocation problem. *Applied Intelligence*, 52(2):2189–2211.

Boninger, D. S., Gleicher, F., and Strathman, A. (1994). Counterfactual thinking: From what might have been to what may be. *Journal of personality and social psychology*, 67(2):297.

Burgard, W., Moors, M., Stachniss, C., and Schneider, F. E. (2005). Coordinated multi-robot exploration. *IEEE Transactions on robotics*, 21(3):376–386.

Byrne, R. M. (2016). Counterfactual thought. *Annual review of psychology*, 67:135–157.

Coello Coello, C. A. (2006). Evolutionary multi-objective optimization: A historical view of the field. *Comp. Intell. Mag.*, 1(1):28–36.

Das, I. and Dennis Jr, J. E. (1996). A closer look at drawbacks of minimizing weighted sums of objectives for pareto set generation in multicriteria optimization problems. Technical report.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *Trans. Evol. Comp*, 6(2):182–197.

Drugan, M. M. and Nowe, A. (2013). Designing multi-objective multi-armed bandits algorithms: A study. In *The 2013 international joint conference on neural networks (IJCNN)*, pages 1–8. IEEE.

Eysenbach, B., Gupta, A., Ibarz, J., and Levine, S. (2019). Diversity is all you need: Learning skills without a reward function. In *International Conference on Learning Representations*.

Foerster, J. N., Farquhar, G., Afouras, T., Nardelli, N., and Whiteson, S. (2018). Counterfactual multi-agent policy gradients. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence* *and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press.

Johansson, F. D., Shalit, U., and Sontag, D. (2016). Learning representations for counterfactual inference. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, page 3020–3029. JMLR.org.

Kasirzadeh, A. and Smart, A. (2021). The use and misuse of counterfactuals in ethical machine learning. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 228–236, New York, NY, USA. Association for Computing Machinery.

Kusner, M., Loftus, J., Russell, C., and Silva, R. (2017). Counterfactual fairness. page 4069–4079.

Lehman, J. and Stanley, K. O. (2011). Novelty search and the problem with objectives. *Genetic programming theory and practice IX*, pages 37–56.

Mouret, J.-B. and Clune, J. (2015). Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909*.

Olson, M. L., Khanna, R., Neal, L., Li, F., and Wong, W.-K. (2021). Counterfactual state explanations for reinforcement learning agents via generative deep learning. *Artificial Intelligence*, 295:103455.

Pierrot, T., Richard, G., Beguir, K., and Cully, A. (2022). Multi-objective quality diversity optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '22, page 139–147, New York, NY, USA. Association for Computing Machinery.

Rahmattalabi, A., Chung, J. J., Colby, M., and Tumer, K. (2016). D++: Structural Credit Assignment in Tightly Coupled Multiagent Domains. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4424–4429, Daejeon, South Korea. IEEE.

Van Moffaert, K., Drugan, M. M., and Nowé, A. (2013). Scalarized multi-objective reinforcement learning: Novel design techniques. In *2013 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, pages 191–199.

Verma, S., Boonsanong, V., Hoang, M., Hines, K. E., Dickerson, J. P., and Shah, C. (2020). Counterfactual explanations and algorithmic recourses for machine learning: a review. *arXiv preprint arXiv:2010.10596*.

Zitzler, E., Laumanns, M., and Thiele, L. (2001). Spea2: Improving the strength pareto evolutionary algorithm. *TIK-report*, 103.