Algorithmically identifying strategies in multi-agent game-theoretic environments

Conference Paper · April 2019

DOI: 10.1117/12.2518609

8 authors, including:

Erin Zaroukian
Army Research Laboratory
32 PUBLICATIONS **36** CITATIONS
SEE PROFILE

Sebastian Samuel Rodriguez
University of Illinois, Urbana-Champaign
3 PUBLICATIONS **19** CITATIONS
SEE PROFILE

Sean Linnaeus Barton
U.S. Army Research Laboratory
28 PUBLICATIONS **65** CITATIONS
SEE PROFILE

James Schaffer
Sysco Corporation
26 PUBLICATIONS **97** CITATIONS
SEE PROFILE

Algorithmically identifying strategies in multi-agent game-theoretic environments

Erin Zaroukian[a], Sebastian S. Rodriguez[b], Sean L. Barton[a], James A. Schaffer[a], Brandon Perelman[a], Nicholas R. Waytowich[a], Blaine Hoffman[a], Derrik E. Asher[a]
[a]CCDC Army Research Laboratory, 2800 Powder Mill Rd., Adelphi, MD USA 20783;
[b]Department of Computer Science, University of Illinois at Urbana-Champaign, 201 N Goodwin Ave., Urbana, IL USA 61801

ABSTRACT

Artificial intelligence (AI) has enormous potential for military applications. Fully realizing the conceived benefits of AI requires effective interactions among Soldiers and computational agents in highly uncertain and unconstrained operational environments. Because AI can be complex and unpredictable, computational agents should support their human teammates by adapting their behavior to the human's elected strategy for a given task, facilitating mutually-adaptive behavior within the team. While some situations entail explicit and easy-to-understand human top-down strategies, more often than not, human strategies tend to be implicit, *ad hoc*, exploratory, and difficult to describe. In order to facilitate mutually-adaptive human-agent team behavior, computational teammates must identify, adapt, and modify their behaviors to support human strategies with little or no *a priori* experience. This challenge may be achieved by training learning agents with examples of successful group strategies. Therefore, this paper focuses on an algorithmic approach to extract group strategies from multi-agent teaming behaviors in a game-theoretic environment: predator-prey pursuit. Group strategies are illuminated with a new method inspired from Graph Theory. This method treats agents as vertices to generate a timeseries of group dynamics and analytically compares timeseries segments to identify group coordinated behaviors. Ultimately, this approach may lead to the design of agents that can recognize and fall in line with strategies implicitly adopted by human teammates. This work can provide a substantial advance to the field of human-agent teaming by facilitating natural interactions within heterogeneous teams.

Keywords: predator-prey pursuit, group dynamics, group strategies, adaptive agents, human-agent teaming, mutually-adaptive

1. INTRODUCTION

1.1 Strategy

While artificial intelligence (AI) has enormous potential for military applications, AI can be complex and unpredictable. Computational agents therefore should support their human teammates by adapting their behavior to the human's elected strategy for a given task in order to facilitate mutually-adaptive behavior within the team. While there are situations where human strategies are top-down, explicit, and easy to understand, human strategies are often implicit and *ad hoc*, which requires the computational teammates to identify, adapt, and modify their behaviors to support human strategies with little or no *a priori* experience.

Collaborative computational teammates can be designed by relying on black-box learning methods [1]. However, such learning often requires massive amounts of training, and the strategies deemed optimal by standard AI measures may not be strategies that humans would find or utilize naturally. Further, by identifying and labeling the implicit and *ad hoc* strategies humans use naturally, we will provide transparency, promote trust (e.g., [2]), and provide a better understanding of how humans work together and how computational teammates can be trained to fit into a human-human dynamic.

We chose to focus on a game-theoretic environment that facilitates multi-agent teaming: the predator-prey pursuit task. In our specification of this task (see Section 2.1 for more details), three predators chase a single prey. When a predator comes in contact with the prey, all predators are rewarded and the prey is punished. While this task may not *require* collaboration, strategies adopted in such predator-prey tasks are often collaborative [3]. While it is important to consider the individual [4], coherent group strategies will require observations at the group level. It has been shown that optimal

group strategies do not necessarily arise from individual optimal behaviors [5]. For these reasons, it is important to consider how suboptimal individual performance can contribute towards optimal group performance. In summary, group optimal behaviors, or group strategies should emerge from cooperative, collaborative coordination (C3) between agents with a joint goal.

At lowest levels of force organization, Soldiers are trained to conduct Battle Drills, well-rehearsed collaborative maneuvers in response to specific situations [6], which remove the need for the formulation of emergent strategies. Many military tasks, however, can be characterized as predator-prey pursuit tasks, where Soldier's rarely act in isolation. From initial planning phases through to execution and after-action reviews, maneuvers are group efforts involving various roles. At a general level, strategies are top-down, but teams/squads execute using decisions made in the field and evolving situations.

1.2 Strategy from movement data

Movement patterns and coordination have long been important in the field of ecology for understanding hunting, mating, migration, conservation, and more. For example, in an experiment on collaborative hunting that exposed pairs of yellow saddle goatfish to a prey, one fish would initiate a chase, and the other fish would either pursue the prey directly or by a more circuitous route, depending on its initial distance from the initiator. These actions show that collaborative strategies can result from simple, ostensibly selfish rules [7][6]. As geolocation data became more readily available, ecologists developed computational techniques to characterize dispersed movement patterns among groups of animals. These methods, however, often focus on broad patterns of large geographic and time scales, such as migration, or they identify simple predefined patterns such as flocking or trendsetting [8].

At a more abstract level, algorithms have been developed to learn discrete descriptions like linear temporal logic [9] and probabilistic context free grammars for tasks like setting a table [10] or parking in a parking lot [11]. These, however, require abstraction when using continuous movement data in order to arrive at workable atomic units, which can be difficult to determine *a priori*. For example, navigation through a parking lot with the grid movements of left, right, forward, and backward, may suffice to describe some tasks, but it may not have the resolution to capture the important nuances of parallel parking. Continuous environments allow for greater movement flexibility with the increased degrees of freedom, which is important for more complex strategies and even harder to discretize *a priori*. Another limitation for learning discrete descriptions from input is that these methods often do not allow for generalization, describing only what they've seen and making it difficult or impossible to categorize novel movement data into classifications of strategy based on prior observations.

In similarly constrained problems it is possible to infer strategies and heuristics from planned routes, rather than actual movement. For example, route planning strategies were tested in a graph, and the heuristics used by subjects changed with increasing memory loads [12]. Similarly, a graph-based approach was used to determine human ability to solve problems where the solution does not fit the optimum (i.e., the shortest route) [13]. Critically, these graph-based approaches require highly constrained stimuli, each with a relatively small number of potential solutions, to successfully infer the agents' strategies. In these cases, the experimenters designed the stimuli specifically to produce several different types of solutions, each with its own implication for human spatial decision-making. Such a technique is useful for inferring generalizable findings about human behavior, but it is not suited for the inferential strategies formed and utilized by multi-agent or human teams in real-time.

While human strategies are often inferred in highly constrained environments, such as the table setting, parking, and route planning examples above, they can also be inferred using retrospective verbal report. These approaches assume that the humans have declarative access to the types of strategies that they employ during spatial problem solving. Several prior studies have investigated human spatial decision-making strategies using retrospective subjective reporting). For example, participants have been asked to plan a vacation route in a fictional location then asked to write about their strategies in planning the route [14]. Behavioral and subjective response data were then both analyzed to infer different decision-making strategies used by the participants. In some cases, the employed strategies were based on global characteristics of the trajectories themselves. In other cases, participants employed neighborhood-based strategies in which their trajectories were formed around local goals. Forming and subsequently inferring strategies from

subjective report requires a great deal of ground truth knowledge about the environment, and hence may not generalize to novel environments or conditions with substantial uncertainty.

To address the requirement for *a priori* ground truth knowledge of the environment, data-driven approaches have been developed for characterizing strategies in less constrained environments. A shape-based similarity algorithm (the Algorithm for finding the Least Cost Areal Mapping between Paths; ALCAMP [16]) and multidimensional scaling can be applied to differentiate two distinct strategies in a simulated aerial search and rescue task [15]. More recently, a similar approach has been used to investigate distinct spatial mental models in a route planning task [17]. These approaches are useful for inferring agents' strategies in unconstrained environments, however methodologically they still require multiple observations across agents in a single environment. That is, all agents must solve the same set of spatial problems, and they are therefore not suited to the simulation approaches that randomly generate environments and agents' starting locations for each trial. A more general problem space should include diversity, e.g., sufficient variation in initial conditions through learning to avoid encountering only a subset of strategies dependent on the selected subset of initial conditions, cf. [7].

The challenge with identifying any strategy employed to achieve a goal is that a strategy is not observable (i.e., not directly measurable), whereas the behaviors that lead to an achieved goal are observable and measurable. Thus, the objective of identifying group strategies ultimately must emerge from individual behaviors. Many methods, however, only detect predefined strategies, highly repeatable strategies, or strategies built from predefined units, making them ill-suited for discovering strategies in unconstrained environments.

In this paper, we opt to interpret movement data as timeseries data and use comparison algorithms to determine similarity between timeseries. Dynamic Time Warping (DTW) measures the similarities between two data sequences over time (i.e., timeseries), without the requirement of having an equal number of samples. By defining a cost function (usually the absolute difference between two points as a function of time, i.e., $c(t) = |x(t) - y(t)|$), DTW warps a sequence to the length of the comparing sequence, using the cost function to determine their similarity. DTW has been used in signal processing and speech recognition [18] and is actively used in satellite image analysis [19]. In the current study, the timeseries are generated from multi-agent movement data by allowing the area of a simple polygon (the agents are the vertices) to measure the relative distances between agents, which can reveal group movement-based strategies. Additionally, while not pursued within this paper, methods such as Dynamic Feature Extraction can be used to identify additional pieces of information.

The proposed methodology was initially motivated by an application of coordinated behaviors in multi-robot systems [20]. The coordinated behaviors displayed by agents (or robots) in their work, was driven by an implementation of graph theory that treats agents as vertices with exclusive edges that only connect to other agents within some local vicinity. Similarly, our approach treats agents as vertices, and assumes that the area enclosed by the agents is measurable over time to produce a timeseries showing some representation of group coordinated behaviors within the context of reinforcement learning (RL) agents within a multi-agent system.

2. METHODS

As presented above and discussed in greater detail below, in order to investigate group movement strategies, we use simulated multi-agent movement data from a predator-prey pursuit task. The agents' locations are used to calculate a measure of group configuration at a given point in time, the area of the smallest polygon that can be made using the agents' positions as vertices, which is then used to represent an episode of the predator-prey pursuit task as a univariate timeseries. Techniques for discovering break points and for determining similarity between timeseries can then be applied to this group movement data in an attempt to discover recurring group coordinated behaviors or strategies.

2.1 The predator-prey pursuit task and data simulation

In a military context, problems are often considered in terms of the interactions within and between groups of allies and adversaries. While human ally-adversary dynamics are likely highly complex and context dependent, a useful model of these interactions can be found in the interactions between predator and prey animals engaged in a pursuit-avoidance

task. Such a task pits predators and prey against one another with competing mutually exclusive goals, and the outcome of such a task depends critically on both the capabilities of the individuals involved and the group strategies employed by each side. As discussed above, models of predator-prey pursuit behavior are common within the field of ecology, and methods used to identify behavioral strategies among groups of animals (such as the fish in [7][6]) may be useful in identifying behavioral strategies between allies and adversaries in general.

Due to the obvious advantage of cooperation between predators in a pursuit task, simulations of predator-prey pursuit problems provide a remarkably useful means of studying the development of group strategies and collaborative behavior amongst individuals [21]. While this task is historically presented with discretized states, more recent variations have begun to look at the emergence of collaborative behavior in continuous versions of predator-prey pursuit simulation experiments [22], [23]. In this study, we utilized a continuous version of the predator-prey pursuit task developed in [22] as part of their study on deep reinforcement learning for inter-agent collaboration.

In order to understand the development of behavioral strategies within the context of an adversary-ally task like predator-prey pursuit, we used a modified version of the predator-prey simulation and learning environment presented in [22]. We trained three predator agents to try to capture a single prey agent using a deep reinforcement learning paradigm. The behavior of both the predators and prey were dictated by a 2-layer fully-connected feed forward network that processed perceptual information about the environment (in the form of positions, velocities, and actions of all agents) and used this information to select an action in this continuous space. The networks were trained using a modified off-policy deterministic policy gradient (DDPG) reinforcement learning algorithm that utilized a centralized critic between all predator and prey agents to train decentralized behavioral policy networks. This multiagent-DDPG algorithm (MADDPG) was first documented in [22] and has shown promise as a means for training agents to modify their behaviors with respect to the actions of both partners and adversaries within an environment. We utilized the meta-parameters provided in [22]. Our only modifications were that 1) a hard boundary was placed around the task environment to prevent agent escape/divergent trajectories, 2) obstacles were removed from the environment, and 3) one predator's behavior was periodically replaced with a fixed behavioral strategy (see below) for the purposes of measuring coordination [23].

The agents were trained for 100,000 episodes, and each episode lasted for 25 timesteps. To minimize the potential of an agent learning behavior specific to the starting configuration, the position of each agent at the beginning of an episode was randomized. Predator agents received a fixed positive reward anytime *any* predator made contact with the prey agent. The prey received a fixed negative reward (i.e., a punishment) any time it collided with a predator. Agents were not provided any reward based on their distance from one another, and predators were not penalized for collisions with one another. The authors in [22] showed performance convergence at 40,000 episodes and no significant behavioral or performance deviations after an additional 30,000 episodes. They observed robust prey pursuit behavior from their predator agents, as well as intelligent avoidance behavior from the prey after 70,000 episodes. We similarly found that our agents demonstrated intelligent pursuit-avoidance behaviors consistent with learning to perform the task. As such, we assumed that after 100,000 training episodes agent behavioral strategies and performance had stabilized.

Agent learning was "turned-off" after 100,000 episodes and the trained policy networks were used to generate behavioral trajectories (i.e., test episodes). These test episodes (i.e., episodes where no learning took place) were extended to 250 timesteps (from the 25 timesteps during training) to ensure sufficient data to identify behavioral patterns. On a subset of test episodes, the trained policy of one predator was replaced with a fixed-policy that did not pursue the prey and instead followed the dynamics of double pendulum with a randomly initialized acceleration and position [3], [23], [24]. This modified behavior was used as a marker for a substantially different behavioral strategy from any that may have been learned through training and served as a baseline comparison for strategy shifting. We collected 8 test episodes, of which 4 had one predator following the modified behavioral strategy (i.e., the double pendulum dynamics). As in training, the starting position for each agent was randomized to provide agents an opportunity to demonstrate a diverse range of behaviors.

2.2 Timeseries methodology

To conduct the timeseries analysis, timeseries were constructed to test the effectiveness of using change point detection (CPD) to identify shifts in strategies as continuous agent actions occurred. Data was generated as described in Section 2.1. A 500 timestep episode was synthesized by interleaving segments ranging from 50 to 250 timesteps in length, which we will refer to as *ground truth segments*, from two episodes. The interleaved ground truth segments came from episodes that either had 1) all trained-policy agents or, 2) one predator agent using double pendulum dynamics (fixed-

policy) and the other three agents (two predators and one prey) were trained-policy agents. The interleaved segments were connected by linearly interpolating the edge points over 4 steps.

In order to distill multi-agent movement data into a timeseries, we propose a single composite measure for the relative position of all agents – the area of the smallest polygon that can be made using the agents' positions as vertices (see Figure 1). Agents' Cartesian coordinates within the task environment were used to calculate the area of the polygon they enclosed using a modified version of the *shoelace formula* [25]. Polygon vertices were ordered by computing the polygon centroid and then sorting each vertex by its tangent to the centroid. An example polygon is show in Figure 1, and an example time series is shown in Figure 2.

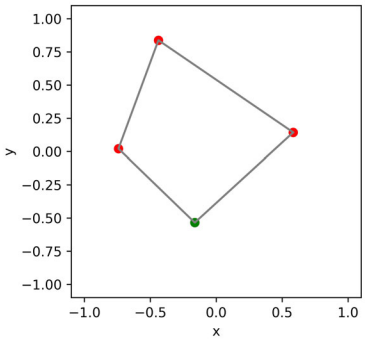

Figure 1. A single time step of the predator-prey pursuit task with three predators (red dots) and one prey (green dot). The polygon area used in timeseries analysis is visualized here through the lines connecting the four agents. The x- and y-axes show the simulation environment centered on (0, 0) and ranging [-1, 1] in both x and y dimensions.

CPD can be approached as an offline analysis where all data points are collected before analysis, the sample size is fixed, and no new data can be considered. As a conservative approach, a combination of various cost functions from different distributions of data was utilized to determine change points in the timeseries. In general, CPD considers a variety of statistical metrics, including, but not limited to, mean, variance, covariance, rank, and density [26]. The Pruned Exact Linear Time (PELT) method is then used to optimally search for timeseries sections while maintaining linear computational complexity (i.e., it produces results in a finite amount of time) [27].

In the current study, CPD was used to identify boundaries between different group strategies. The interleaved ground truth segment boundaries (see Figure 2, switching between colors) served as a validation for the CPD method selected.

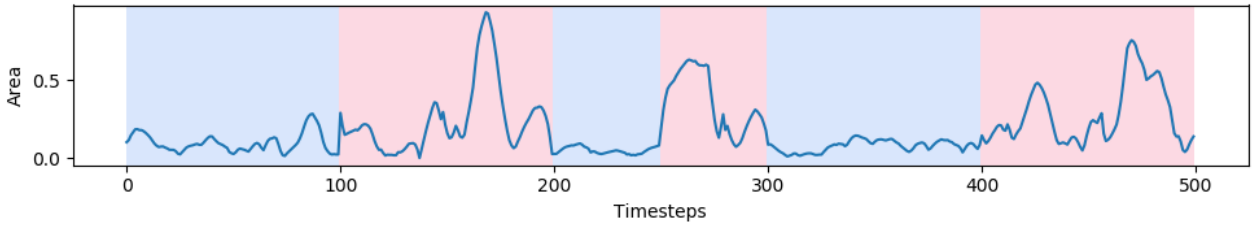

Figure 2. Polygon area timeseries for Episode 2. The different shaded regions of the plot represent the 2 different interleaved ground truth timeseries segments from one predator agent using double pendulum dynamics (fixed policy) in red and all learning agents in blue. The interleaving ground truth segments break at time steps 100, 200, 250, 300, and 400. The x-axis shows timesteps and the y-axis shows polygon area.

DTW was used to compare similarities between pairs of timeseries subsets discovered through CPD, which we refer to as *CPD fragments*. We can visualize DTW as a gradient matrix with an outlined shortest path between start and end points of two timeseries (see Figure 3). Dark areas signify similar values between the two timeseries. To successfully warp the timeseries, DTW must fulfill three constraints: 1) the first and last indices of one timeseries must match with the first and last indices from another, 2) the indices must be monotonically increasing (e.g., if index 3 is matched, index 2 can no longer be matched), and 3) every index must be matched with at least one index from the other series. DTW

will provide a solution of similarity (similarity = 1 – distance) if these three constraints are met. The solution is a measure of the distance between the two compared timeseries.

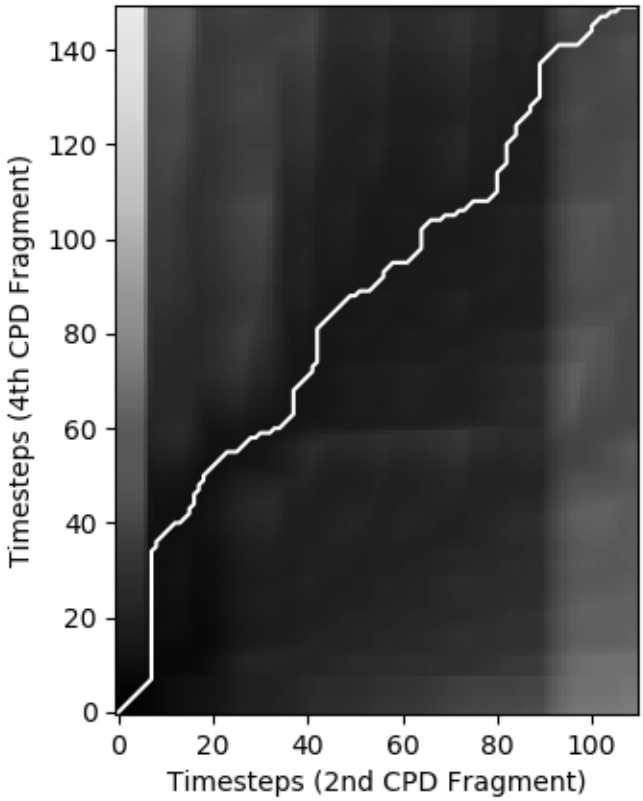

Figure 3. Dynamic Time Warping matrix for Episode 3 comparing the 2nd and 4th CPD fragments. The line represents the shortest path between start and end, fulfilling all DTW constraints while yielding the highest similarity.

3. RESULTS

3.1 Change Point Detection

CPD can be evaluated using three metrics: precision and recall, the Rand index, and the Hausdorff distance. Precision and recall allow us to determine how well CPD determined the correct breakpoints taking into account Type I and Type II errors. Since these three metrics are aggregations of a binary event, we allow a leniency margin of 20 timesteps (i.e., if the breakpoint is within 20 timesteps of the truth, it is considered a true positive). The Rand index determines the similarity between the generated breakpoints to the ground truth, effectively serving as a proxy to accuracy. The Hausdorff distance measure is the farthest distance between generated breakpoints and the ground truth (i.e., how far the worst generated breakpoint is from the truth). Because the cost function can be defined, we can adjust the sensitivity of CPD to our needs. Low costs make the PELT method too sensitive to change in the timeseries, generating excessive and inaccurate breakpoints (Figure 4). High costs instead prevent recognition of breakpoints, treating the entire series as a single segment. A moderate cost yielded the highest similarity between the CPD-generated breakpoints and the ground truth.

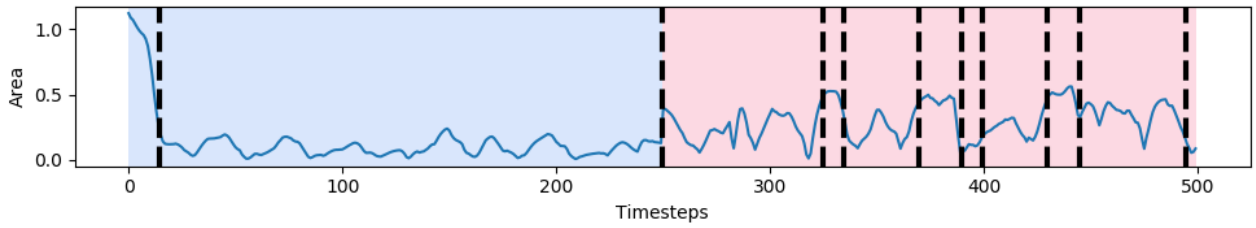

Figure 4. Episode 1 timeseries with penalty = 5 (dotted lines) overlaid the ground truth segments (blue = trained/red = fixed). Excessive generated breakpoints (dotted lines) can be seen in the later half.

With a moderate cost, most generated breakpoints are adequate for analysis with DTW (Figures 5-8). Figure 5 shows Episode 1 with moderate-cost generated breakpoints. Note that the second breakpoint identified exactly the transition from trained- to fixed-policy strategy ground truth segments, each 250 timesteps in length. One extraneous breakpoint was generated at the beginning of the episode. Because agents were randomly assigned starting positions, this small fragment, while not a separate ground truth segment, represents the predators' initial convergence on the prey.

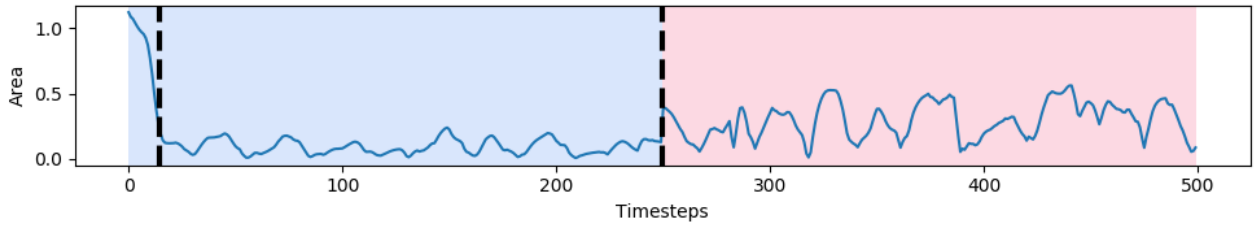

Figure 5. Episode 1 with Generated breakpoints with penalty = 10 (dotted lines) overlaid the ground truth segments (blue = trained/red = fixed).

Figure 6 shows Episode 2 with moderate-cost breakpoints. Episode 2 was built from shorter ground truth segments, 50 and 100 timesteps in length, than Episode 1, and CPD correspondingly did a fair job identifying all the ground truth segment breakpoints (i.e., the first ground truth segment breakpoint is missed by 40 timesteps in Figure 6).

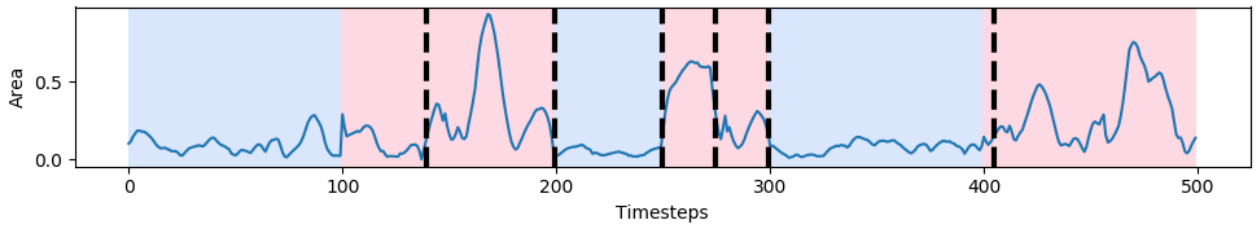

Figure 6. Episode 2 with Generated breakpoints with penalty = 10 (dotted lines) overlaid the ground truth segments (blue = trained/red = fixed).

Figure 7 shows Episode 3 with moderate-cost breakpoints. The ground truth segments are 100 and 150 timesteps in length, intermediate between those identified in Episodes 1 and 2 (see Figures 5and 6, respectively). The first breakpoint is off by 10 timesteps (within our leniency margin for precision and recall), and the second is off by 20 (at our leniency margin).

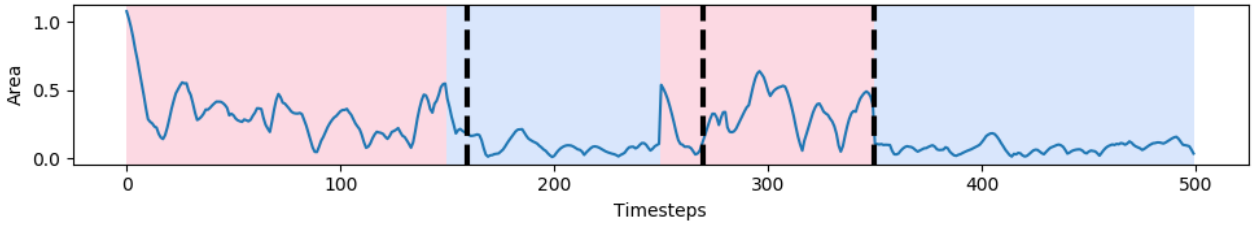

Figure 7. Episode 3 with Generated breakpoints with penalty = 10 (dotted lines) overlaid the ground truth segments (blue = trained/red = fixed).

Figure 8 shows Episode 4 with moderate-cost breakpoints. Similar to Episode 2, the ground truth segments are 50 and 100 timesteps in length. The fourth breakpoint is off by 5 timesteps, the fifth by 5 timesteps, the sixth by 15 timesteps, and the seventh and final by 25 timesteps.

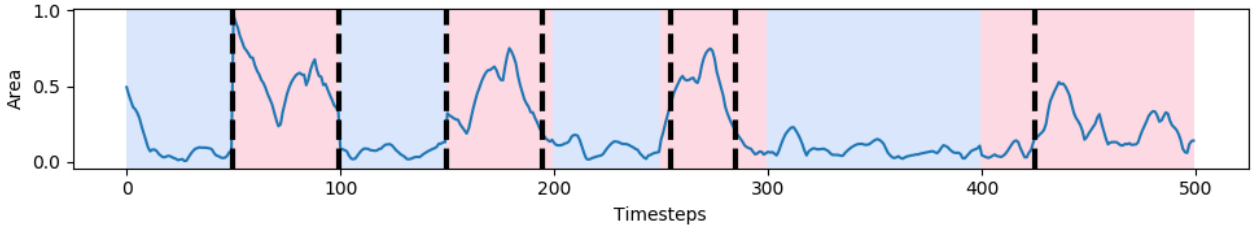

Figure 8. Episode 4 with Generated breakpoints with penalty = 10 (dotted lines) overlaid the ground truth segments (blue = learned/red = fixed).

Evaluation metrics for the similarity between ground truth and the CPD-generated breakpoints using various costs are shown for all episodes in Tables 1-3. Table 1 shows the precision and recall for the breakpoints vs ground truth, allowing a leniency margin of 20 timesteps. Unsurprisingly, a compromise between precision and recall is achieved with a moderate cost.

Table 1. Precision (P) and Recall (R) results in fabricated series compared to ground truth (leniency margin = 20 timesteps)

Episode/Cost	**C(pen) = 1** P, R	**C(pen) = 5** P, R	**C(pen) = 10** P, R	**C(pen) = 15** P, R	**C(pen) = 20** P, R
Episode 1	0.02, 1.0	0.10, 1.0	0.50, 1.0	0.50, 1.0	1.0, 1.0
Episode 2	0.12, 1.0	0.33, 1.0	0.67, 0.80	0.00, 0.00	0.00, 0.00
Episode 3	0.08, 1.0	0.22, 0.67	0.67, 0.67	0.67, 0.67	0.67, 0.67
Episode 4	0.16, 0.86	0.46, 0.85	0.85, 0.85	0.85, 0.85	0.00, 0.00

Table 2 shows the Rand similarity index for each episode. Unsurprisingly, episodes with fewer ground truth segments (Episodes 1 and 3) have a higher similarity index for greater costs, whereas those with more ground truth segments (Episodes 2 and 4) have their highest similarity index for the more moderate cost used in Figures 5-8.

Table 2. Rand similarity index in generated breakpoints compared to ground truth (0 = no similarity, 1 = identical)

Episode/Cost	**C(pen) = 1**	**C(pen) = 5**	**C(pen) = 10**	**C(pen) = 15**	**C(pen) = 20**
Episode 1	0.52	0.77	0.97	0.97	1.0
Episode 2	0.85	0.89	0.94	0.18	0.18
Episode 3	0.79	0.89	0.95	0.95	0.95
Episode 4	0.89	0.91	0.94	0.93	0.14

Table 3 shows the Hausdorff distances, which represent the severity of the worst breakpoint within each episode. A moderate cost led to lower Hausdorff distances and permitted breakpoints.

Table 3. Hausdorff distance (in timesteps) in generated breakpoints compared to ground truth (NB = no breakpoints generated). Normalized distance (distance divided by episode length, 500 timesteps) is bolded.

Episode/Cost	C(pen) = 1	C(pen) = 5	C(pen) = 10	C(pen) = 15	C(pen) = 20
Episode 1	245 **(49%)**	245 **(49%)**	235 **(47%)**	235 **(47%)**	0 **(0%)**
Episode 2	90 **(18%)**	90 **(18%)**	40 **(8%)**	**NB**	**NB**
Episode 3	145 **(29%)**	120 **(24%)**	20 **(4%)**	20 **(4%)**	20 **(4%)**
Episode 4	90 **(18%)**	45 **(9%)**	25 **(5%)**	25 **(5%)**	**NB**

3.2 Dynamic Time Warping

DTW warps sequences very effectively, resulting in high similarities between CPD fragments, even with comparisons between differing strategies. Tables 4-7 provide the distances between segments and their similarity scores for Episodes 1-4 respectively. CPD fragments are classified in these tables as either trained-policy or fixed-policy based on the ground truth classification for the majority of the timesteps (this ranged from 71-100% of the CPD fragment). Table 4 shows distance and similarity scores for Episode 1. Surprisingly, the lowest similarity score is between same-strategy CPD fragments 1 and 2. Recall, however, that fragment 1 identifies the initial convergence of randomly positioned agents, so fragment 1 is indeed qualitatively different from fragment 2. The average same-strategy similarity score is 0.81, and the average different-strategy similarity score is 0.89.

Table 4 Distance matrix for Episode 1. Blue headers represent CPD fragments classified as trained policy, red headers as fixed policy. Cells comparing matching strategies are white, while cells with mismatched strategies are gray. Similarity is bolded.

Fragment	2	3
1	0.19 **(0.81)**	0.14 **(0.87)**
2		0.07 **(0.93)**

Table 5 shows distance and similarity scores for Episode 2. Recall that the first breakpoint was off by 40 timesteps, but despite this discrepancy, fragment 1 uniformly shows lower distances/higher similarity scores with same-strategy CPD fragments (3, 6) compared to different-strategy CDP fragments (2, 4, 5, 7). This reinforces the CPD assessment that included fixed-policy timesteps in fragment 1, either because the agent movements themselves more closely resembled a trained-policy strategy or because polygon area was unable to capture some nuance about their difference. Along the same lines, the greater similarity between fragments 5 and 6 (different strategies) than between fragments 5 and 7 suggests that CPD identified something important about fragment 5, whether it was the agent behavior or the way it was represented by polygon area. The average same-strategy similarity score is 0.95, and the average different-strategy similarity score is 0.90.

Table 5. Distance matrix for Episode 2. Blue headers represent CPD fragments classified as trained policy, red headers as fixed policy. Cells comparing matching strategies are white, while cells with mismatched strategies are gray. Similarity is bolded.

Fragment	2	3	4	5	6	7
1	0.07 **(0.93)**	0.04 **(0.97)**	0.14 **(0.86)**	0.04 **(0.96)**	0.02 **(0.98)**	0.05 **(0.95)**
2		0.14 **(0.86)**	0.08 **(0.92)**	0.09 **(0.91)**	0.10 **(0.90)**	0.03 **(0.97)**
3			0.21 **(0.79)**	0.06 **(0.94)**	0.01 **(0.99)**	0.13 **(0.87)**
4				0.14 **(0.86)**	0.16 **(0.84)**	0.08 **(0.93)**
5					0.04 **(0.97)**	0.08 **(0.92)**
6						0.09 **(0.91)**

Table 6 shows distance and similarity scores for Episode 3. Here, same-strategy CPD fragments uniformly measured more similar than different-strategy fragments. The average same-strategy similarity score is 0.97, and the average different-strategy similarity score is 0.91.

Table 6. Distance matrix for Episode 3. Blue headers represent CPD fragments classified as trained policy, red headers as fixed policy. Cells comparing matching strategies are white, while cells with mismatched strategies are gray. Similarity is bolded.

Fragment	2	3	4
1	0.09 **(0.91)**	0.05 **(0.95)**	0.10 **(0.90)**
2		0.08 **(0.92)**	0.01 **(0.99)**
3			0.09 **(0.91)**

Finally, Table 7 shows distance and similarity scores for Episode 4. Here again, same-strategy CPD fragments uniformly measured more similar than different-strategy fragments. The average same-strategy similarity score is 0.95, and the average different-strategy similarity score is 0.86.

Table 7. Distance matrix for Episode 4. Blue headers represent CPD fragments classified trained policy, red headers as fixed policy. Cells comparing matching strategies are white, while cells with mismatched strategies are gray. Similarity is bolded.

Fragment	2	3	4	5	6	7	8
1	0.18 **(0.82)**	0.03 **(0.97)**	0.12 **(0.88)**	0.03 **(0.97)**	0.11 **(0.89)**	0.02 **(0.98)**	0.05 **(0.95)**
2		0.24 **(0.76)**	0.08 **(0.92)**	0.18 **(0.82)**	0.07 **(0.93)**	0.195 **(0.81)**	0.12 **(0.88)**
3			0.18 **(0.82)**	0.02 **(0.98)**	0.21 **(0.79)**	0.02 **(0.98)**	0.09 **(0.91)**
4				0.11 **(0.89)**	0.03 **(0.97)**	0.14 **(0.86)**	0.05 **(0.95)**
5					0.18 **(0.82)**	0.02 **(0.98)**	0.06 **(0.94)**
6						0.17 **(0.83)**	0.07 **(0.93)**
7							0.04 **(0.96)**

DTW similarity scores between same-strategy CPD fragments vs. different-strategy CPD fragments were compared using an exact Wilcoxon-Mann-Whitney U test[1], which found that these scores were significantly different with a large effect size (Mann-Whitney $U = 675$, $n = 58$, $p < 0.001$, $r = 0.55$). The average same-strategy similarity score is 0.92, higher compared to the average different-strategy similarity score of 0.89.

4. DISCUSSION

The goal of this work is to facilitate the development of computational agents that support their human teammates by adapting their behavior to the human's elected strategy for a given task. We begin by identifying group strategies from movement data, with the aim of ultimately enabling computational agents to identify and adapt to these strategies bottom up. To identify group strategies from movement data, we generated multi-agent movement data for a preditor-prey pursuit task utilizing two visually distinct pursuit strategies. We then represented this movement as the area of a polygon with the agents are vertices and applied CPD to attempt to find transitions between strategies. While CPD was not perfect, it performed well with appropriate tuning. DTW was then used to compare CPD fragment pairs and classify them as using either the same strategy or different strategies with a good deal of success.

4.1 Limitations

The results presented in this paper provide a step on a path toward the goal of robustly identifying strategies from movement data to facilitate collaboration, but many roadblocks remain. The predator-prey environment used was free of obstacles, and while this serves as a simple starting point, it is not realistic for military applications. Obstacles can easily impact the type of strategies adopted in a predator-prey pursuit task, and polygon area may or may not do well capturing

[1] A non-parametric test was chosen because a Shapiro-Wilk test determined that the DTW distance data was not normally distributed ($W = 0.936$, $p = 0.004$).

these strategies. Similarly, a simplification was made in having all computational agent teammates. Human teammates may use different strategies, and they are susceptible to other forces, such as distraction.

As mentioned previously, the variable used in these timeseries, the area of the polygon enclosing the agents, is not necessarily the best measure. Similar to finding atomic units for discrete descriptions, information is lost here. For example, the same polygon areas are not completely unique (degenerate), as there are the cases when all agents have the same relative spacing but different orientations (e.g., rotation of the state space). Similarly, the same area can be calculated for, e.g., distant agents nearly lined up as for all agents clustered close together. However, this relative position approach serves as a reasonable approximation of group coordinated behavior and removes the troublesome issues associated with rotational and mirror invariances. Future exploration and work with Dynamic Factor Analysis may reveal other important measures (e.g., whether the prey was captured at a given time step, proximity to a boundary), which may change depending on nuances of the task (e.g., training, the number of agents in each role).

While we have attempted to justify our decisions, CPD functions on sensitive parameters, thus care must be taken when defining cost and sampling the sequence, as scarce data will prevent it from detecting adequate breakpoints. It is still unknown what range of sampling rates will be the most effective for CPD and will be a point of contention for research applications that do not include signal processing. Furthermore, what works for a one task, set of strategies, and movement representation may not work for another.

While DTW showed high similarity between CPD fragments representing the same strategy, it also showed rather high similarity between CPD fragments representing *different* strategies. This is not necessarily surprising, given that three of the four agents were using the trained policy in all episodes, with only one (fixed-policy) agent's movement fundamentally differing between strategies (qualitatively, the resulted in the fixed-policy agent largely hovering at the top of the environment). Without exploring more strategies, it is difficult to say whether these small differences will be robust enough to classify same/different strategies generally. If they are not, it may be determined that polygon area is not ultimately a reliable representation of group strategy, or multivariate timeseries can be explored.

4.2 Understanding and interpreting learned policies as strategies in deep learning

A central concern in the field of deep learning, and more specifically in the utilization of deep neural networks, is the question of understandability in the final behavior of a trained agent. Neural networks are inherently black-box functions, and as such the internal representation of environments, tasks, and action policies cannot be fully known. Network weights are non-linearly conditioned on the state of the world (which itself may be unpredictable or unknowable in certain dimensions), and as such knowing or extracting the strategies utilized by a learned policy is exceedingly difficult. The study and results presented here outline a potentially critical solution to this problem by demonstrating how shifts in strategies employed by an agent that could be representative of the underlying policy representation can be detected within the traces of agent behavior. By accepting the black-box nature of deep neural networks and employing methodologies utilized in biological and behavioral sciences, a measure of interpretability can be recovered in the policies of intelligent computational agents.

While interpreting the underlying representations inherent to network weights is difficult, it is not a fruitless endeavor. A complimentary approach to our methodology that is *not* available in behavioral studies of biological organisms is the ability to study how specific states of the world or specific behaviors of the agents are linked to the activations of the internal network that models the action policies under scrutiny. For example, t-distributed stochastic neighbor embedding (TSNE) [28] is a means of representing high-dimensional data in easily visualizable and understandable clusters in a low dimensional probabilistic mapping. This methodology has been utilized by researchers to understand how deep learning agents associate states of the game with desirable outcomes [29]. While this methodology is highly qualitative, it is nonetheless extremely informative about underlying representations embedded within the structure of the neural network.

An interesting extension of the work presented here would be to combine CPD/DTW of behavioral traces with qualitative analysis of network activation via TSNE to determine if behavioral changes in an agent's strategy or policy are represented by changes in activation of the neural network driving the policy. TSNE can be used to show how network states may represent persistent features of the environment via similarity in activation, and may also be informative about the strategy an agent is using to respond to such observations. CPD/DTW may be used to identify changes in an agent's behavior, and differences in activation detectible by TSNE could point to a shift in the underlying policy or strategy of the agent. Such an analysis would be instrumental in understanding if learning agents in a given task

are capable of learning and representing multiple strategies, or if apparent behavioral changes are an artifact of the structure of the task itself. Making such a determination however requires informative analysis of behavioral changes, such as those presented in this study.

ACKNOWLEDGMENTS

Many thanks to Jonathan Z. Bakdash for his help. This research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-18-2-0058. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

REFERENCES

[1] Jaderberg, M., Czarnecki, W. M., Dunning, I., Marris, L., Lever, G., Garcia Castaneda, A., Beattie, C., Rabinowitz, N. C., Morcos, A. S., Ruderman, A., Sonnerat, N., Green, T., Deason, L., Leibo, J. Z., Silver, D., Hassabis, D., Kavukcuoglu, K., and Graepel, T. "Human-level performance in first-person multiplayer games with population-based deep reinforcement learning." arXiv preprint arXiv:1807.01281 (2018).

[2] Gunning, D., "Explainable artificial intelligence (XAI)," Defense Advanced Research Projects Agency, DARPA/I20, (2017).

[3] Barton, S. L., Zaroukian, E., Asher, D. E. and Waytowich, N. R., "Evaluating the Coordination of Agents in Multi-agent Reinforcement Learning," International Conference on Intelligent Human Systems Integration, 765-770 (2019).

[4] Hackman, J. R., "Learning more by crossing levels: Evidence from airplanes, hospitals, and orchestras," Journal of Organizational Behavior: The International Journal of Industrial, Occupational and Organizational Psychology and Behavior 24(8), 905-922 (2003).

[5] Verbeeck, K., Nowé, A., Lenaerts, T. and Parent, J. "Learning to reach the Pareto optimal Nash equilibrium as a team," Australian Joint Conference on Artificial Intelligence, 407-418 (2002).

[6] Department of the Army, Infantry Platoon and Squad, ATP 3-21.8. DC: Washington, (2016).

[7] Steinegger, M., Roche, D. G., and Bshary, R., "Simple decision rules underlie collaborative hunting in yellow saddle goatfish," Proceedings of the Royal Society B: Biological Sciences 285(1871), (2018).

[8] Laube, P. and Imfeld, S., "Analyzing relative motion within groups of trackable moving point objects," Proc. of Geographic Information Science, Second International Conference, 132–144 (2002).

[9] Pnueli, A., "The temporal logic of programs," Proc. 18th IEEE Symp. Foundations of Computer Science, 46-57 (1977).

[10] Shah, A., Kamath, P., Li, S., and Shah, J., "Bayesian Inference of Temporal Task Specifications," Advances in Neural Information Processing Systems, 3808-3817 (2018).

[11] Geyik, S. C., Bulut, E. and Szymanski, B. K.. "Grammatical Inference for Modeling Mobility Patterns in Networks," IEEE Transactions on Mobile Computing, vol. 12(11), 2119-2131, (2013).

[12] Wiener, J. M., Ehbauer, N. N. and Mallot, H. A., "Planning paths to multiple targets: memory involvement and planning heuristics in spatial problem solving," Psychological Research PRPF 73(5), 644 (2009).

[13] Ragni, M., and Wiener, J., "Constraints, Inferences, and the Shortest Path: Which paths do we prefer?," Proceedings of the Annual Meeting of the Cognitive Science Society 34(34), (2012).

[14] Tenbrink, T. and Seifert, I., "Conceptual layers and strategies in tour planning," Cognitive Processing 12(1), 109-125 (2011).

[15] Perelman, B. S. and Mueller, S. T., "Identifying mental models of search in a simulated flight task using a pathmapping approach," Proceedings of the 18th International Symposium on Aviation Psychology, (2015).

[16] Mueller, S. T., Perelman, B. S., and Veinott, E. S., "An optimization approach for mapping and measuring the divergence and correspondence between paths," Behavior Research Methods 48(1), 53-71 (2016).

[17] Perelman, B. S., Evans, A. W. III, and Schaefer, K. E., "Mental model consensus and shifts during navigation system-assisted route planning," Proceedings of the Human Factors and Ergonomics Society, (2017).

[18] Sakoe, H. and Chiba, S., "Dynamic programming algorithm optimization for spoken word recognition," IEEE Trans Acoustics Speech Signal Process 26, 43-49 (1978).
[19] Wegner Maus, V., Câmara, G., Appel, M. and Pebesma, E., "dtwSat: Time-Weighted Dynamic Time Warping for Satellite Image Time Series Analysis in R," Journal of Statistical Software 88.5, 1-31 (2019).
[20] Cortés, J. and Egerstedt M., "Coordinated control of multi-robot systems: A survey," SICE Journal of Control, Measurement, and System Integration 10.6, 495-503 (2017).
[21] Benda, M., "On optimal cooperation of knowledge sources," Technical Report BCS-G2010-28, (1985).
[22] Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, O. P. and Mordatch, I., "Multi-agent actor-critic for mixed cooperative-competitive environments," Advances in Neural Information Processing Systems, 6382-6393 (2017).
[23] Barton, S. L., Waytowich, N. R., Zaroukian, E., and Asher, D. E., "Measuring collaborative emergent behavior in multi-agent reinforcement learning," International Conference on Human Systems Engineering and Design: Future Trends and Applications, 422-427 (2018).
[24] Barton, S. L., Waytowich, N. R., and Asher, D. E., "Coordination-driven learning in multi-agent problem spaces," CEUR Workshop Proceedings 2269 (2018).
[25] Pure, R. and Salman, D., "Computing exact closed-form distance distributions in arbitrarily-shaped polygons with arbitrary reference point," The Mathematica Journal 17, 1-27 (2015).
[26] Truong, C., Oudre, L. and Vayatis, N., "ruptures: change point detection in Python," arXiv preprint arXiv:1801.00826 (2018).
[27] Killick, R., Fearnhead, P. and Eckley, I. A., "Optimal Detection of Changepoints with a Linear Computational Cost," Journal of the American Statistical Association 107, 1590-1598 (2012).
[28] Van der Maaten, L. J. P. and Hinton, G. E., "Visualizing high-dimensional data using t-SNE," J. Mach. Learn. Res. 9, 2579–2605 (2008).
[29] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumara, D., Wierstra, D., Legg, S., Hassabis, D., "Human-level control through deep reinforcement learning," Nature 518(7540), 529-533 (2015).