Utility of Doctrine with Multi-agent RL for Military Engagements

Anjon Basak[1], Erin G. Zaroukian[3], Kevin Corder[1,2], Rolando Fernandez[3], Christopher D. Hsu[3], Piyush K. Sharma[3], Nicholas R. Waytowich[3], and Derrik E. Asher[3]

[1]Army Research Laboratory-Research Associateship Program, US
[2]University of Delaware, US
[3]DEVCOM Army Research Laboratory, US

October 2021

ABSTRACT

Artificial intelligence (AI), or more specifically deep learning approaches to AI, have led to astonishing results in recent times, which makes them a prime candidate for guiding agent actions in military domains. However, it is often difficult to train multiple agents with deep learning approaches when a task is sufficiently complex, or the state space is huge, as is often the case in military domains. One possible way to alleviate the difficulties associated with military tasks is to leverage military doctrine to assist in the guidance of multi-agent systems. Military doctrine is a guide to action rather than hard rules for the execution of military campaigns, operations, exercises, and engagements. Doctrine, written by experts in their respective domains, is used to make sure that each task associated with an engagement, for example, is executed according to military standards. Such standards ensure coordination between different tasks, resulting in a greater likelihood of Mission success. In addition, the efficacy of combining doctrine with deep learning must be tested to determine any realized benefit for AI driven military engagements under adversarial conditions. Further, the inherent complexities associated with military engagements demand coordination between heterogeneous resources and teams of Soldiers which are often geospatially separated. In this work, we establish a baseline of doctrine-based maneuvers for a military engagement scenario with a multi-agent system (MAS) embedded in the StarCraft Multi-Agent Challenge (SMAC) simulation environment, now a standard test environment for Multi-Agent Reinforcement Learning (MARL). We introduce a hybrid training approach that combines MARL with doctrine (MARDOC) to test whether doctrine-informed MARL policies produce more realistic behaviors and/or improved performance in a simple military engagement task. We compare this hybrid approach to both doctrine-only (i.e., supervised learning to mimic doctrine) and MARL-only approaches to evaluate the efficacy of the proposed MARDOC approach. Our experiments show that MARDOC approaches produce desired behavior and improved performance over supervised approaches or MARL alone. In summary, the experimental results suggest that MARDOC approaches provide a sufficient advantage over MARL alone due to agent doctrinal guidance of MARL exploration to overcome the complexities in military domains.

1. INTRODUCTION

Searching a large state space is a fundamental problem in the multi-agent reinforcement learning (MARL) domain. Agents need to explore the state space to learn effective policies, and the larger the state space, the more time it will take agents to learn an optimal and appropriate behavior. This indicates that the amount of training steps required for effective AI agent behavior (i.e., a learned RL policy) grows proportionally to the state space size within a given environment and associated task. This large state space issue is exacerbated in military domains because agents and tasks are inherently distributed across large spatial areas (e.g., a city) and typically involve many dimensions and variables that all can have a critical impact on the outcome of adversarial interactions. Even an abstracted simulation of a military engagement in a game environment (e.g., StarCraft) is a challenging problem to overcome with MARL agents because the training time can quickly become untenable. Even with supercomputers, training time can take weeks to months to develop an effective policy requiring 10s to 100s of millions of training iterations.[1]

To overcome the extensive amount of iterations required to develop an effective policy with MARL agents in military engagements, it is possible to include doctrinal knowledge into the learning process. Doctrinal knowledge is used in this work to represent information used to guide the behavior of agents that would otherwise be unavailable in the learning environment. Specifically, the doctrinal knowledge referred to in this work is an implementation of military doctrine that guides the behavior of the multi-agent system (MAS) to perform an envelopment maneuver in an adversarial engagement.[2]

Given that domain knowledge (e.g., imitation or demonstration learning) can allow AI systems to rapidly learn tasks irrespective of state space size,[3–7] it is reasonable to assume that an AI agent guided by doctrine may learn a task much more rapidly regardless of task complexity or state space size. To give a specific example, if domain knowledge drives humans to work as a group, that group might then more quickly learn to maneuver in a group formation to relevant places in an environment and coordinate with other groups, resulting in relevant, interpretable, and desired behavior. In the US Army domain, coordination across a team or group of cooperative agents within a MAS has been an important topic of research.[8–18] These efforts aimed to measure and observe emergent behaviors associated with explicitly working together, instead of focusing on a typically low dimensional representations of group performance. Therefore, it would be of substantial interest to combine doctrine with AI to guide coordinated MAS in military domains.

Minimizing exploration in MARL training can minimize the time that agents need to converge on a policy. However, this may result in undesirable behavior or poor team performance because exploration is a critical component to learning. As MARL agents randomly explore an environment, the reward function will guide agents towards exploitable behaviors, which should eventually result (given enough training or exploration time) in a good set of policies that yield desirable agent behaviors. Therefore, learning good policies comes at the cost of sufficient exploration time. In order to minimize the cost of learning (i.e., minimizing exploration time) agents must be guided by something other than random exploration.

The inclusion of doctrinal knowledge, e.g. team behavior or role, even guided placement or maneuver in state space, can allow agents to explore more task relevant parts of a state space. Exploitation of doctrinal knowledge in combination with experiences gained through exploration can rapidly guide agents towards task-relevant behavior in a given environment. Although, even state-of-the-art algorithms need a sufficient training signal to guide agents behavior through exploration of a state space. Most MARL approaches suffer with insufficient training signals (e.g., sparse reward, small reward, or vanishing reward), and as a result, agents either waver around randomly or stick to a primitive policy which is associated with undesirable behaviors.[19] In this work, agents are provided with a sufficient learning signal as demonstrated in literature.[20–23]

MARL exploration approaches in current literature can be divided into two categories: *directed* and *undirected*.[24,25] *Directed* exploration takes into account, 1) the learning process, and 2) the agent history, where learned behavior can influence future exploration. Most of the directed approaches take motivation from cognitive science, using some form of intrinsic reward to motivate an agent to explore novel states.[26–29] However, many of the directed methods suffer from vanishing intrinsic rewards. To overcome this issue, the intrinsic reward can be modulated with state novelty and an entropy regularizer.[16,21,30] For undirected exploration (e.g., ϵ-greedy), most methods utilize some type of probability distribution, ignoring agent history, which results in selected actions based on the chosen probability distribution parameters. Such techniques can learn an optimal policy in small tabular settings, but they are not typically scalable to large state spaces. Game theory in MAS mostly uses domain knowledge to merge different states together[31,32] to tackle the scalability issue. In game theory it is called abstraction. However, this technique is not followed in MARL since deep NNs use function approximation to combat the scalability issue.

In this paper, the MARL + doctrine (MARDOC) approach can be described as directed exploration. Specifically, the MARDOC approach reduces the cost of agent exploration by following task relevant maneuvers guided by US Army doctrine (i.e., give agents a head-start in learning using doctrinal maneuvers). The experimental results show that doctrine guided agents to explore more efficiently, leading to militarily relevant behaviors, in comparison to a state-of-the-art MARL algorithm, which converged to a non-militarily relevant policy within the same training duration.

2. METHODS

In this work, a modified version of the *8m* map from the StarCraft Multi-agent challenge (SMAC)[33] environment was utilized for all experiments. The map was modified to facilitate an *envelopment maneuver* (section 2.2).[2] Two doctrine-only baseline algorithms were utilized (with full state space and partial state space observability) to compare the different learning approaches against. Three MARL + doctrine approaches paired the RODE[34,35] algorithm with elements from military doctrine e.g. envelopment maneuver and Fog of War (FOW).

FOW is the uncertainty in the state space, or the information that lies outside of agents' partial observations in an environment. FOW may refer to the uncertainty a particular agent or the collective group, and can include unseen agents, behaviors, intents, capabilities, etc. In our experiment FOW is implemented over the Adversarial forces by making them aware of only the Allied force (e.g. Alpha or Bravo) that triggers the sensor in a Trigger Region. For the purposes of these experiments, doctrine is effectively used to define initial positions (with and without a FOW, MARDOC(-)FOW and MARDOC(-) respectively) and fixed behaviors (with the FOW, MARDOC(+)FOW) taken from the envelopment doctrinal maneuver document.[2] The **MARDOC(-)FOW** terminology is used to depict, MARL + doctrine was utilized ("MARDOC"), with a minimal amount of doctrine (the minus symbol "-"). Similarly, **MARDOC(-)** indicates MARL + doctrine with a minimal amount of doctrine, and no fog of war implemented (i.e., Adversarial forces would attack both Allied teams if any Allied agent trigger the sensor in a Trigger Region). The term **MARDOC(+)FOW**, indicates MARL + doctrine, with maximal doctrine implemented (the plus symbol "+") in the form of a fixed set of movements, and fog of war over the Adversarial forces. In MARDOC(+)FOW the adversary is aware of the Allied force (e.g. Allied Alpha) that triggers the sensor. The Adversarial force cannot pursue the other Allied force (Allied Bravo) until their task is finished with the current engaged Allied force (Allied Alpha).

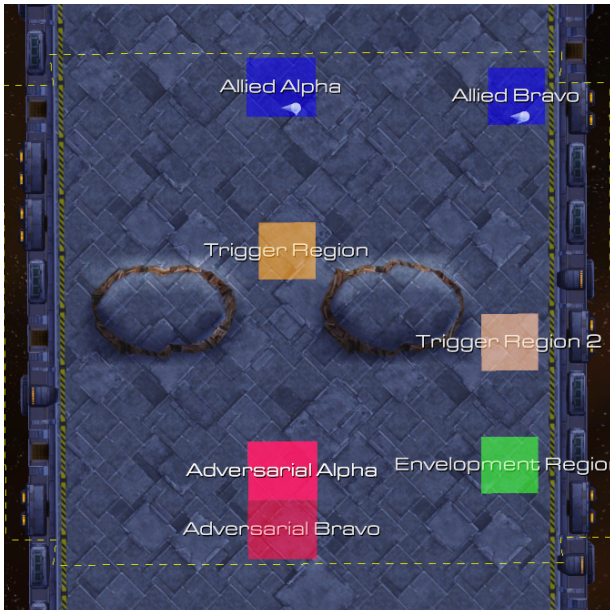
In all the learning algorithms the Allied forces are divided into two teams (Allied Alpha and Allied Bravo) based on envelopment doctrine. In the FOW implementations (MARDOC(-)FOW, MARDOC(+)FOW, and MARLFOW), the Adversarial forces are only aware of the the Allied force that enters a trigger region, and the other Allied force remains invisible until any agent within that team triggers the sensor in a trigger region. In contrast, for MARDOC(-), the Adversarial forces are aware of both Allied forces when any agent from either team enters a trigger region. For *MARDOC(+)FOW*, the Allied forces use a fixed policy to maneuver according to the envelopment doctrine, and MARL (specifically a standard implementation of the RODE algorithm, see Section 2.3) takes over when they trigger the sensors on the terrain. For MARDOC(-) and MARDOC(-)FOW, the Allied forces use MARL from the onset with initial positions following envelopment doctrine. Finally, *MARLFOW* uses MARL for the entire episode.

2.1 SMAC Map

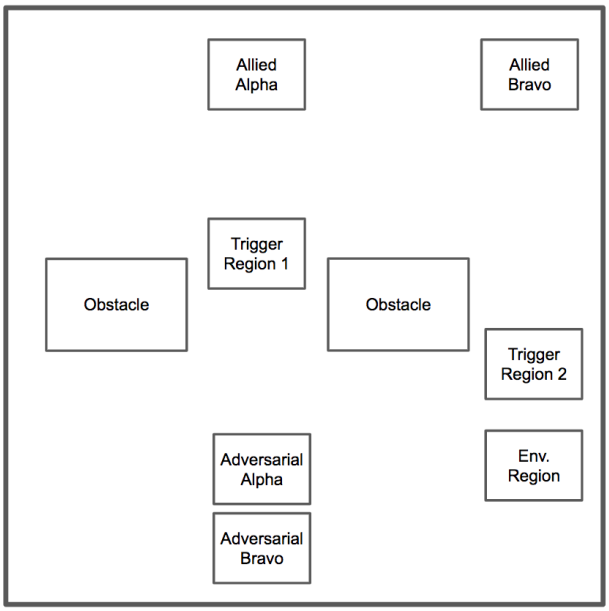
For the experiments, a modified version of the *8m* map from the SMAC maps[34] was utilized, with the addition of two obstacles (hills) as can be seen in Figure 1. Modifications to the original SMAC map (not shown) were implemented to illustrate that doctrinal knowledge with MARL can guide agents to find more militarily relevant or desirable behavior. Otherwise in most cases, agents simply learned to immediately engage face to face without exhibiting any sophisticated maneuver. In the simulation environment (see Figure 1), obstacles act as impassable barriers (i.e., the agents must move around the obstacles and cannot attack over or through them). Therefore, the obstacles can act as protection and create funneled points of engagement between the Allied and Adversarial forces (i.e., left of both obstacles, between the obstacles, or to the right of the obstacles). Further, the original SMAC map did not have Trigger Regions (i.e., a pseudo-sensor area of the map that alerted or informed the Adversarial teams that Allied forces were in that region), an Envelopment Region, or a separated initial position for the Allied Bravo team (see Figure 1). These SMAC map modifications facilitated experiments with doctrinal maneuver for single envelopment (see Figure 2).

Each side (Allied and Adversarial) started with 8 *marine* agents. Only the Allied force is controlled by either MARL, doctrine, or MARDOC in the set of experiments. The Adversarial agents are exclusively controlled by the default StarCraft II heuristic built-in game AI.

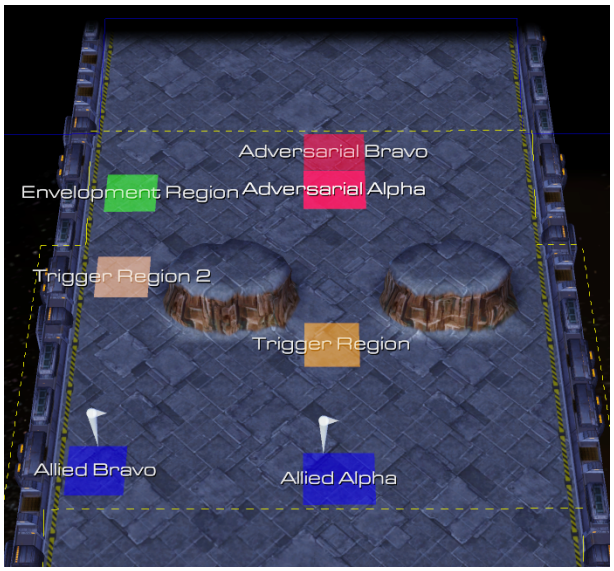
The envelopment maneuver was first implemented by dividing the Allied force into Allied Alpha and Allied Bravo. Similarly the Adversarial force was separated into two teams (Adversarial Alpha and Adversarial Bravo).



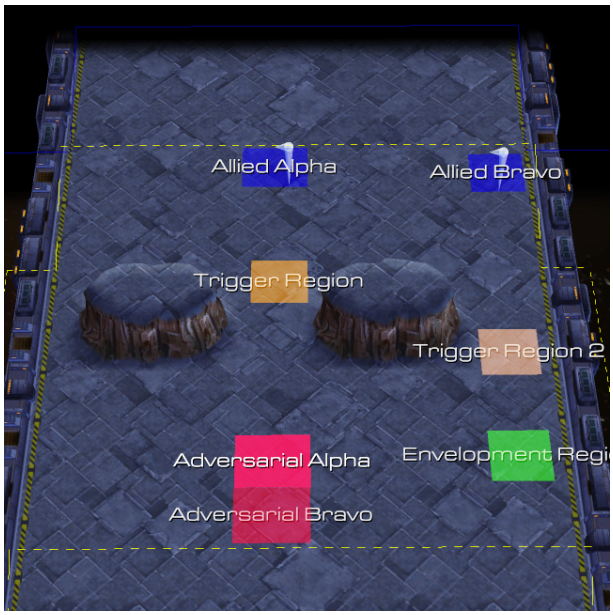
(a) Top view of actual SMAC map.



(b) Simplified top view of the SMAC map.



(c) View from Allied side.



(d) View from Adversarial side.

Figure 1: Modified SMAC map with doctrinal implementations of an envelopment maneuver for Allied Alpha and Bravo initialized positions. The SMAC images show the starting locations for the Allied agents (blue), Adversarial agents (red), the Trigger Regions (orange is region 1, and beige is region 2), and the envelopment region (green). (a) Top view of the actual SMAC map used in the experiments, (b) Simplified doctrinal implementations, (c) SMAC map from Allied initial position perspective, and (d) SMAC map from the Adversarial initial position perspective.

The *Allied Alpha* and *Allied Bravo* regions represent the initialized locations for the Allied Alpha and Allied Bravo forces. Similarly, the *Adversarial Alpha* and *Adversarial Bravo* regions represent respective initialization locations. *Trigger Region 1* and *Trigger Region 2* represent sensors placed on the map that give the Adversarial forces alerts that an Allied agent has entered the respective trigger region, resulting in Adversarial agent engagement. The Envelopment Region represents a region on the SMAC map that initiates the envelopment maneuver by the Allied force.

In our experiments, all the learning algorithms were trained for 1m time steps. Each episode lasts a maximum of 120 time steps. For killing any one Adversarial agent, the Allied forces get 10 reward. The Allied forces get an additional 200 reward if they win by killing all of the Adversarial forces. The default hyperparameters of the RODE[35] algorithms were unmodified for our experiments.

2.2 Doctrinal Envelopment Maneuver

In this work, we utilize the envelopment doctrine maneuver[2] where the commander organizes the ally force to perform two primary tasks: fixing the enemy force in its current location and conducting the envelopment. Envelopment is defined as "a form of maneuver in which an attacking force seeks to avoid the principal enemy defenses by seizing objectives behind those defenses that allow the targeted enemy force to be destroyed in their current positions".[36] Often times, the focus of an *envelopment maneuver* entails seizing terrain, destroying specific enemy forces, or interdicting enemy withdrawal routes. In these experiments, we model the single envelopment maneuver where the Allied fixing force (or Allied Alpha in Figure 2), with sufficient combat power, ascertains the enemy's attention by conducting a frontal attack on the bulk of the enemy's forces, where 1) the enemy is strongest, 2) the enemy's attention is focused, and 3) the enemy's fires are most easily concentrated. While the Allied fixing force (or Allied Alpha in Figure 2) conducts a front assault on the enemy's main force, the Allied envelopment force (axis hook or Allied Bravo in Figure 2) (see Figure 2, left image) performs the decisive operation. The envelopment force avoids the bulk of the enemy force's front by maneuvering around to the enemy's flank and sometimes catching the enemy unawares due to the enemy's own forward movement to engage the fixing allied force. Attacking forces of allies need to be agile enough to concentrate and amass combat power before the enemy can reorient their defense. The envelopment doctrine maneuver is a decisive tactic that can change the tide of a battle.

2.3 Algorithms

Learning was only applied to the Allied forces throughout all experiments where MARL was utilized. The MARL approach utilized for learning cases is called the RODE[35] algorithm, and was selected as a state-of-the-art algorithm that typically provides excellent performance in SMAC environments. The reward function awarded the Allied forces with 10 points for eliminating each Adversarial unit, and 200 additional points upon winning a battle (i.e., eliminating all Adversarial forces in an episode). The Adversarial forces used the built-in StarCraft II game AI for all experiments. Performance of four different algorithmic approaches was compared: a) doctrine + heuristic AI, b) doctrine + heuristic AI w/ limit, c) MARL, and d) MARL + doctrine (MARDOC).

The MARL + doctrine (MARDOC) approaches were divided into three different cases (i.e., MARDOC(-)FOW, MARDOC(+)FOW, and MARDOC(-)) to demonstrate the impact on various dimensions of performance from levels of doctrine integration As stated previously, the envelopment maneuver from the US Army doctrine[2] was used to guide the agents in all cases (see Figure 2 for a description of the doctrine implementation).

2.4 Doctrine + Heuristic AI

This scenario is utilized as an upper bound on agent performance, where full state-space knowledge is provided. Doctrine + Heuristic AI should be compared against the MARL and MARDOC cases, to show how simple agents can perform with full state space knowledge.

In the Doctrine + Heuristic AI scenario, Allied Alpha and Allied Bravo forces advance towards *Trigger Region 1* and *Env. Region* respectively using a fixed policy as shown in Figure 2. As the Allied Alpha force progresses towards the nearest trigger region or 'sensor' labeled *Trigger Region 1*, the Adversarial Alpha and Bravo forces advance towards the Allied Alpha force to engage (see Figure 2, right image, arrows labeled '1' and '3'). While the Allied Alpha force engages the Adversarial Alpha and Bravo forces, the Allied Bravo force progresses towards

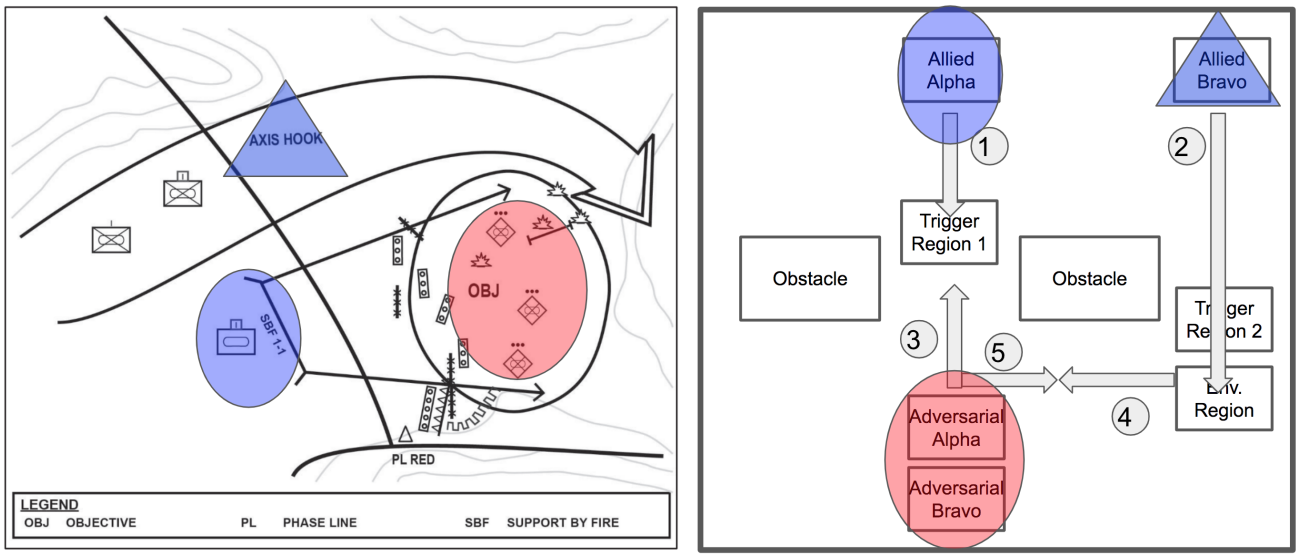


Figure 2: Maneuver for single envelopment from doctrine (left) and SMAC implementation (right). The left image was adopted from US Army Combined Arms Battalion[2] doctrine showing doctrinal maneuver for single envelopment with the allied forces starting in two spatially separated regions (blue oval and blue triangle), maneuvering towards an objective region where adversaries reside (red oval). The right image extends Figure 1b with numbered arrows indicating the order of actions for the implementation of the single envelopment maneuver. Note that numbered arrows 1 and 2 are executed roughly simultaneously. The Allied Alpha force reaches Trigger Region 1 before Allied Bravo reaches Trigger Region 2, which is why the Adversarial forces begin to engage the Allied Alpha force (number 3 arrow) before Allied Bravo begins the envelopment action (number 4 arrow), followed by the Adversarial force engaging Allied Bravo (number 5 arrow). The Allied Alpha (fixing force) force keeps the Adversarial force engaged with enough firepower so that when the Allied Bravo (envelopment force) attacks the Adversarial force cannot get enough time to reorient themselves and properly defend the attack.

the 'sensor' labeled *Trigger Region 2* and continue maneuvering towards the *Env. Region* (see Figure 2, right image, arrow labeled '2'). *Trigger Region 2* informs both Adversarial forces of Allied Bravo's location. The built-in game AI controlling the Adversarial forces maintains a queue to handle multiple triggers. Next, the Allied Bravo force maneuvers towards the Adversarial forces that are engaging with Allied Alpha, and attacks from the side (see Figure 2, right image, arrow labeled '4'). The built-in game heuristic AI takes over control of Allied Alpha and Bravo forces once they have reached *Trigger region 1* and *Env. Region* respectively. The heuristic AI finds the closest opponent and starts shooting.

2.5 Doctrine + Heuristic AI w/ Limit

This scenario is essentially same as the Doctrine + Heuristic AI scenario except the Allied forces have a range limit in their action space: an Allied agent range of shot is smaller than that of the Adversarial agent and requires closer engagement. This limit was imposed to match the limitations placed on the MARL and MARDOC implementations, and serves as a baseline or lower bound for learning agent performance, in the absence of learning.

2.6 MARL

For this scenario, the two Allied agent forces (Alpha and Bravo) start in the same location (i.e., *Alpha* region), (see Figures 1 and 2 for reference to the starting region). As stated previously, Allied forces utilized the RODE[35] MARL algorithm to guide agent behavior. In contrast, Adversarial force behavior was guided by the built-in StarCraft II game AI. In this work, the MARL algorithm is labeled "MARLFOW" to indicate that the MARL approach had the same fog of war implementation on Adversarial agent action as two of the MARDOC

approaches. It is important to note that although the MARLFOW approach had both Allied Alpha and Allied Bravo forces start in the same location, the two teams still existed. This implies that the "FOW" designation only permits the Adversarial forces to engage members of the Allied force that entered a trigger region (e.g., if one agent from Allied Alpha entered a trigger region, the Adversarial forces would pursue and engage all members of the Allied Alpha force and completely ignore the Allied Bravo force.)

In other words, although the Allied forces start in the same location (both Alpha and Bravo forces start in the Alpha location), the trigger region sensors still handle the Alpha and Bravo forces separately. This indicates that the the Adversarial forces attack the triggering Allied force. Further, the corresponding group (e.g. Allied Alpha or Allied Bravo) is added to a queue and maintained by the built-in AI in the case that multiple sensors were triggered by the Allied forces in a first-in-first-out (FIFO) order.

2.7 MARL + Doctrine (MARDOC)

In this paper, we present two stages or levels of doctrinal implementation based on the degree to which doctrine is integrated with MARL. The first level is a simple implementation of the initialized location (for MARDOC(-) and MARDOC(-)FOW) of the Allied agents in two spatially separated areas on the map (Allied Alpha and Allied Bravo). The second level of doctrinal integration effectively is an implementation of the envelopment doctrinal maneuver (for MARDOC(+)FOW) with Allied Alpha and Allied Bravo forces (see Figure 2, right image).

It is important to note that MARDOC(-) is the same as MARDOC(-)FOW with a difference in the implementation of the trigger regions. The trigger regions in MARDOC(-)FOW will only alert the Adversarial forces of the Allied force that triggers the sensor in the trigger region (e.g., if Allied Alpha triggers the sensor in Trigger Region 1 or Trigger Region 2, the Adversarial force will begin to pursue and engage only Allied Alpha, while completely ignoring Allied Bravo). For MARDOC(-), when Allied Alpha or Allied Bravo forces triggers (sensor) in either trigger region (i.e., Trigger Region 1 or Trigger Region 2), the Adversarial forces begin to pursue and engage both Allied Alpha and Bravo forces. Entering a trigger region immediately alerts the Adversary.

A summary of the 3 MARDOC approaches is provided below:

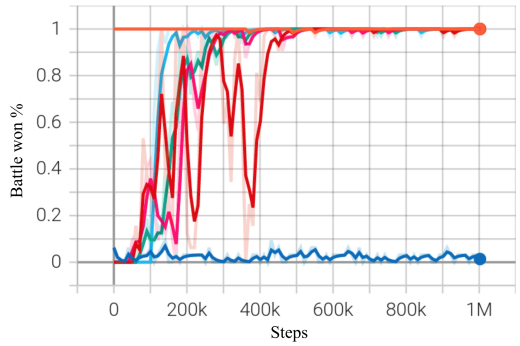
(1) **MARDOC(-)FOW**: Allied Alpha (4 agents) and Allied Bravo (4 agents) forces start each episode in the Allied Alpha and Allied Bravo regions, shown in Figure 1. Every time step after the initialization is controlled by MARL (RODE algorithm). Adversarial forces can only attack the Allied force that has triggered the sensor in the trigger region.

(2) **MARDOC(+)FOW**: doctrine is used as a fixed policy to guide Allied Alpha and Bravo forces' behavior as is shown in Figure 2. Allied forces (Alpha and Bravo) maneuver to *Trigger Region 1* and the envelopment region (*Env Region*) respectively before MARL (RODE algorithm) takes over. Identical to MARDOC(-)FOW, Adversarial forces can only pursue the Allied force that has triggers the sensor in a trigger region. If there are multiple triggers then the triggers are handled in a FIFO order.

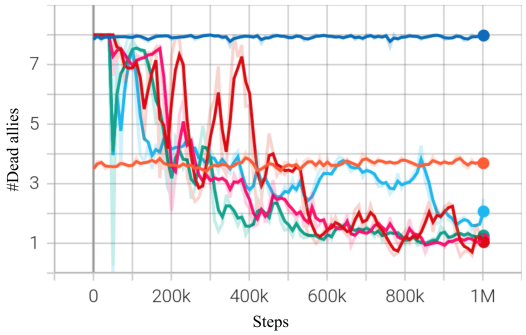
(3) **MARDOC(-)**: same as MARDOC(-)FOW, except, when one of the Allied forces triggers either sensor (i.e., *Trigger Region 1* or *Trigger Region 2*), the Adversarial forces become aware of both Allied forces (i.e., no fog of war).

3. RESULTS

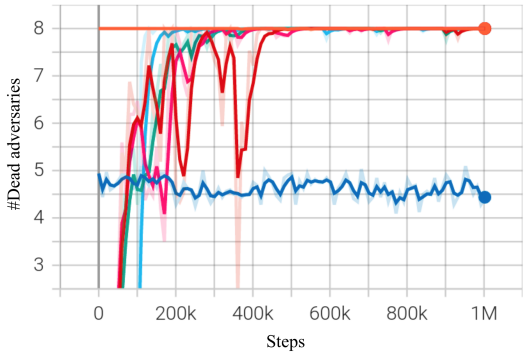
The primary purpose of the experiments described in this work was to investigate the impact of integrating components of doctrinal knowledge (specifically from an envelopment maneuver) with a MARL algorithm (RODE in this case). The performance across 4 dimensions (proportion of battles won, Allied casualties, Adversarial casualties, and episode length) was compared between six algorithmic approaches: i) Doctrine + Heuristic AI, ii) Doctrine + Heuristic AI w/ limit, iii) MARDOC(-)FOW, iv) MARDOC(+)FOW, v) MARDOC(-), and vi) MARLFOW. 10 independent models were trained for each learning approach (MARDOC and MARL), but not for the Doctrine + Heuristic AI approaches because learning was absent. The best performing model for each



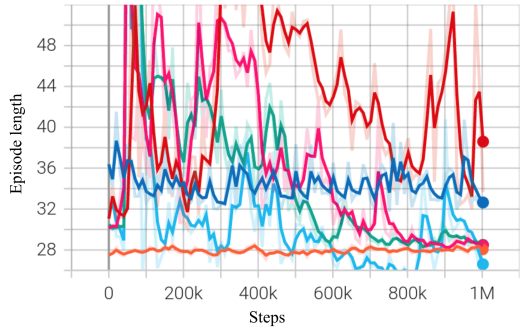
(a) Normalized proportion of battles won by Allied force (1 = 100% of battles won).



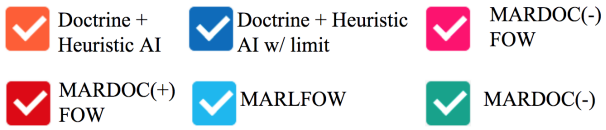
(b) Allied force casualties.



(c) Adversarial force casualties.



(d) Episode length.



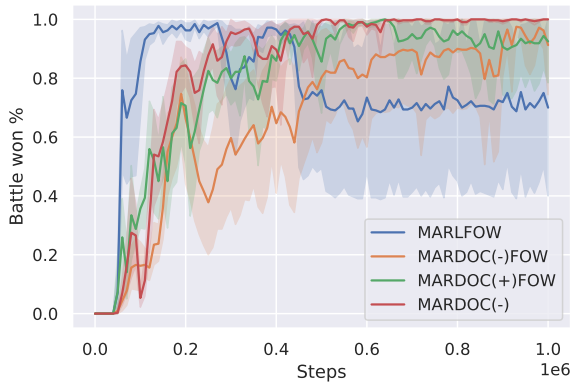
(e) Legend for comparing the 6 different approaches in subplots a - d.

Figure 3: Metrics comparing the best performing model among 6 different algorithmic approaches with doctrinal maneuver implemented into a SMAC simulation environment. (a) Normalized proportion of battles won by the Allied force across 1M time steps of training (for best performing MARL and MARDOC models) or testing (for Doctrine + Hueristic AI with and without limit). (b) and (c) show the number of causalities (max of 8) for the Allied and Adversarial forces respectively. (d) The number of time steps per episode, averaged over the time steps since the previous data point.

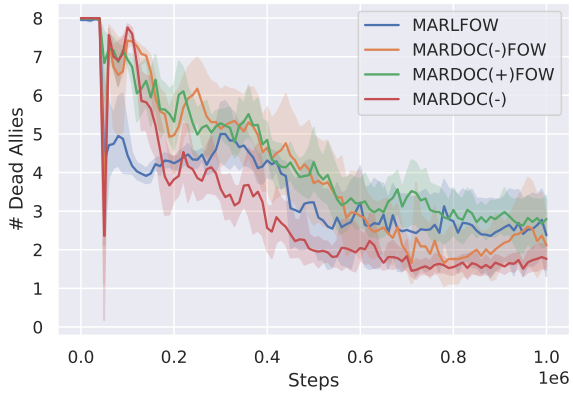
algorithm is shown in Figure 3. Note that Doctrine + Heuristic AI has a 100% battle win rate and Adversarial force casualties from the beginning (see Figure 3a, x-axis at 0, orange line plot), since this approach allows the Allied forces to utilize the full state space information (i.e., there is no limit on the sight range or weapon range for the Allied forces when attacking an Adversarial agent).

It can be observed that the Allied force suffers a higher casualty rate after about 500k steps when learning is involved (compare MARDOC and MARL to Doctrine + Heuristic AI in Figure 3b). The Doctrine + Heuristic AI w/ limit performed the poorest in all performance metrics since the Allied forces could not see or shoot an Adversarial agent unless it was within range. Therefore, this approach should be evaluated as a lower bound or 2nd bounding baseline for learning approach metrics.

Similar to Doctrine + Heuristic AI, MARDOC(-)FOW, MARDOC(+)FOW, and MARDOC(-) algorithmic

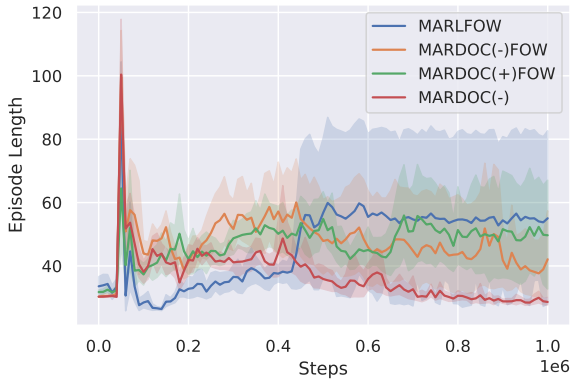


(a) Normalized proportion of battles won by Allied force (1 = 100% of battles won).



(b) Allied force casualties.

(c) Adversarial force casualties.



(d) Episode length.

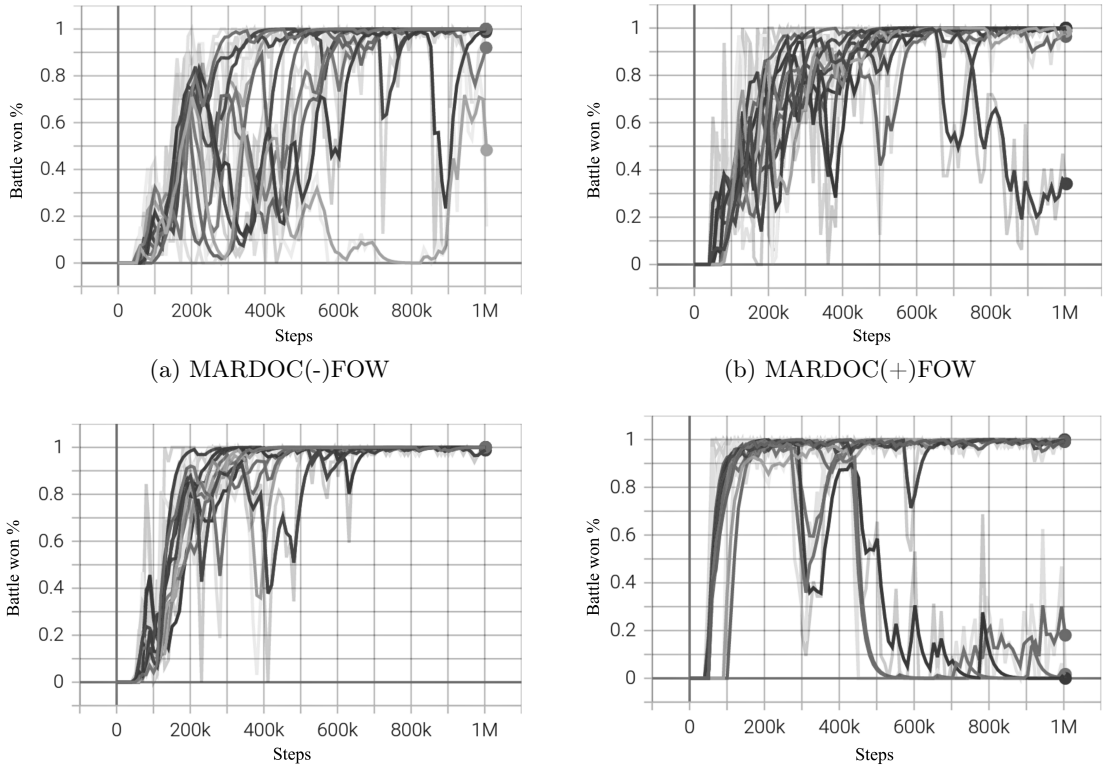
Figure 4: Metrics showing the mean values (solid lines) with a 95% confidence interval (corresponding color shaded regions) across the four learning approaches (MARDOC(-)FOW, MARDOC(+)FOW, MARDOC(-), and MARLFOW) implemented into the SMAC simulation environment. (a) Normalized proportion of battles won by the Allied force across 1M time steps of training. (b) and (c) show the number of causalities (max of 8) for the Allied and Adversarial forces respectively. (d) The number of time steps per episode, averaged over the time steps since the previous data point.

approaches achieved 100% battle win rate and Adversarial force casualties (see Figure 3a and 3c). It is important to note that although Allied force casualties were not accounted for in the reward function, all learning approaches (MARLFOW included) yielded lower casualties than Doctrine + Heuristic AI (see Figure 3b). Another important observation was that there is evidence to support greater exploration of the state space by MARDOC approaches over MARLFOW alone (see Figure 3d).

Comparisons between the 4 learning approaches (MARDOC(-)FOW, MARDOC(+)FOW, MARDOC(-), and MARLFOW) are shown across the same four dimensions displayed in Figure 3 for normalized proportion of battles won (see Figure 5), Allied force casualties (Figure 6), Adversarial force casualties (Figure 7), and episode length (Figure 8). The displayed curve is the best performing model among all 10 independently trained models per algorithmic approach, chosen by the highest mean battle win proportion in the final episodes. Figure 4 shows the mean curves among these 10 trained models with the 95% confidence interval as the shaded region.

It is important to note the strong similarities between Figures 5 and 7 as these two performance metrics were directly accounted for in the reward function, and the Allied forces cannot win a battle without eliminating all Adversarial forces. Thus these two metrics are directly dependent and are meant to provide a comprehensive

assessment of the different approaches.



(a) MARDOC(-)FOW

(b) MARDOC(+)FOW

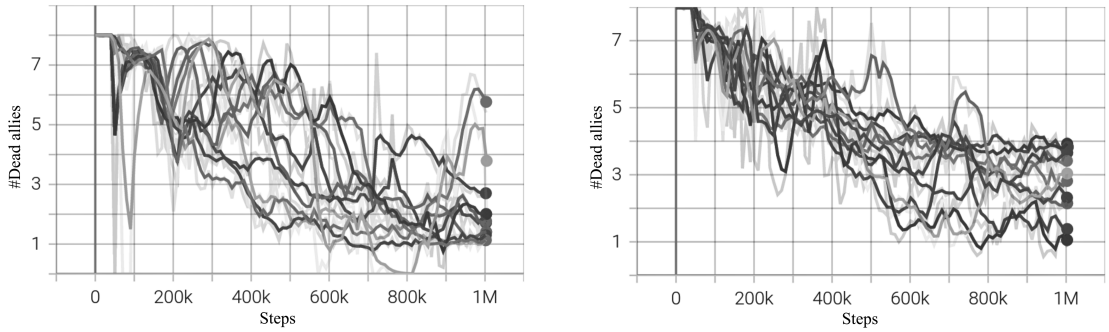
(c) MARDOC(-)

(d) MARLFOW

Figure 5: Normalized proportion of battles won across 10 independent models per algorithmic approach. The x-axes show number of total training or learning steps ranging between [0, 1M] and the y-axes show the normalized proportion of battles won ranging between [0, 1], with 1 equal to 100% of battles won. (a) MARDOC(-)FOW with a MEAN = 0.9125 and STD = 0.2659 at 1M time steps. (b) MARDOC(+)FOW with a MEAN = 0.9250 and STD = 0.2156 at 1M time steps. (c) MARDOC(-) with a MEAN = 1 and STD = 0 at 1M time steps. (d) MARLFOW with a MEAN = 0.7000 and STD = 0.4831 at 1M time steps.

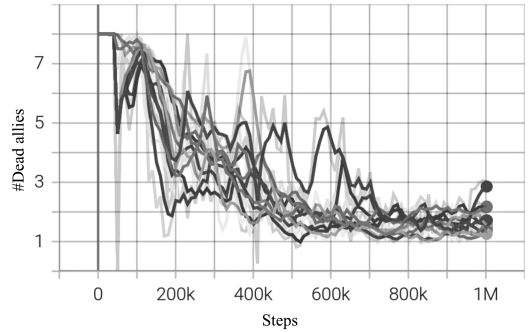
In a comparison between MARLFOW and MARDOC, the results show that 3 MARLFOW models (out of 10) were not able to achieve 100% battle win rate, whereas, out of the 3 MARDOC approaches only 4 of 30 models did not reach 100% win rate, and all 10 of the MARDOC(-) models reached 100% battle win rate (see Figure 5) after 1M time steps of training. In addition, it can be observed from the results that MARDOC(-)FOW took more time steps (about 700k) to reach the maximum battle win rate compared to MARDOC(+)FOW (about 600k) and MARDOC(-) (about 500k) for successful models (compare Figure 5a on the x-axis [500k, 700k] to b and c in the same range). This result indicates that progressing the Allied teams to the *Trigger Regions* (for MARDOC(+)FOW) from initialized locations (MARDOC(-)FOW and MARDOC(-)) had a large impact on battle win rate convergence (for *Allied Alpha* and *Allied Bravo* initial location reference, see Figures 1 and 2). In summary, it can be observed that for successful MARLFOW and MARDOC models, doctrine had a differential impact on convergence time (MARLFOW < MARDOC(-) < MARDOC(+)FOW < MARDOC(-)FOW), compare Figure 5d to a, b, and c at 400k on the x-axes.

In Figure 6, we can observe that most of the of MARLFOW trained models had greater Allied force casualties (7 out of 10 models averaged 4 lost Allied agents) compared to any of the MARDOC approaches (compare Figure 6d to a, b, and c). Interestingly, the results show that 3 of the MARLFOW models appeared to outperform all MARDOC approaches (best achieving an average of 0 Allied force casualties) with respect to Allied force casualties. However, upon further inspection, these 3 MARLFOW models actually just learned to 'do nothing.'

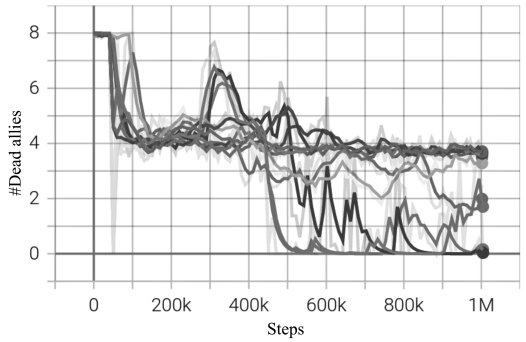


(a) MARDOC(-)FOW

(b) MARDOC(+)FOW



(c) MARDOC(-)



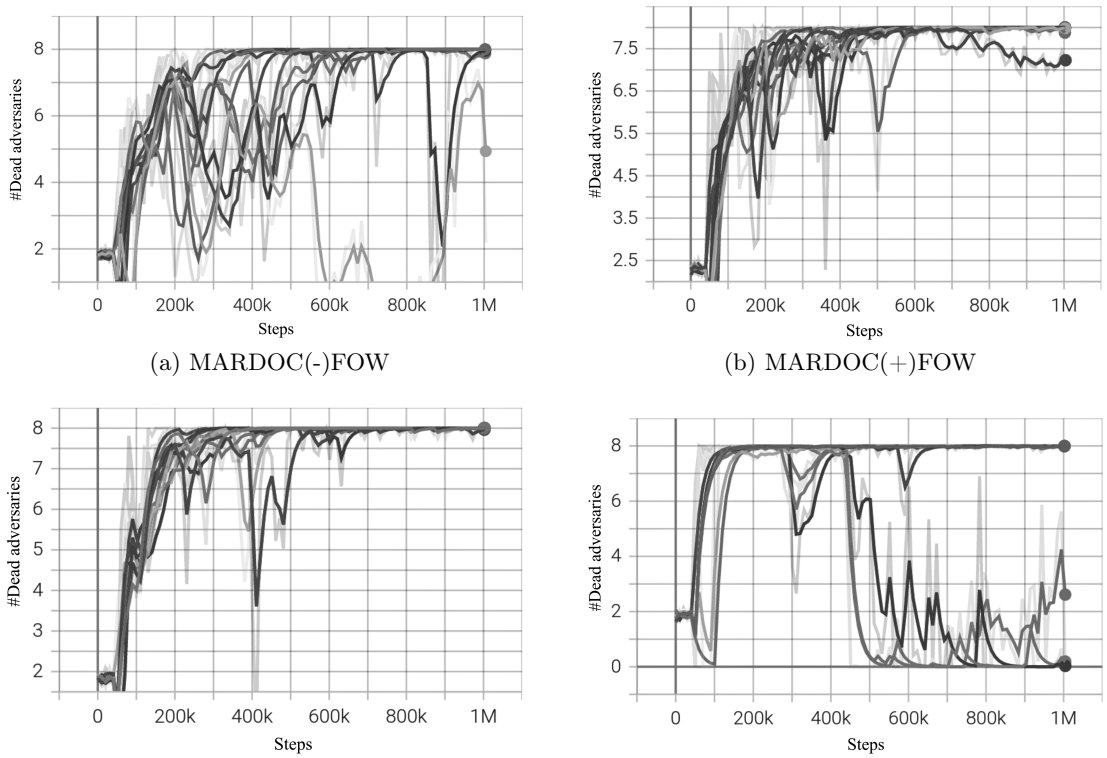
(d) MARLFOW

Figure 6: Allied force casualties across 10 independent models per algorithmic approach. The x-axes show number of total training or learning steps ranging between [0, 1M] and the y-axes show the number of Allied agents lost ranging between [0, 8] with 8, equivalent to a battle lost. (a) MARDOC(-)FOW with a MEAN = 2.12 and STD = 1.24 at 1M time steps. (b) MARDOC(+)FOW with a MEAN = 2.80 and STD = 1.13 at 1M time steps. (c) MARDOC(-) with a MEAN = 1.76 and STD = 0.57 at 1M time steps. (d) MARLFOW with a a MEAN = 2.37 and STD = 1.63 at 1M time steps.

This evidence is supported by the very low win rate (between 0% and 20%) for these 3 models shown in Figure 5d), minimal Adversarial casualties (between 0 and 3 in Figure 7d and the maxed out episode duration (see Figure 8d). In contrast, MARDOC(-) averaged less than 2 Allied agent losses across all 10 models, with only 1 model averaging 3 losses. Note, of the 2 levels of doctrinal implementation the results suggest that MARDOC(-)FOW and MARDOC(-) performed about the same for successful models (i.e., ignore the 2 MARDOC(-)FOW models with 4 and 6 average Allied casualties at 1M time steps), whereas, MARDOC(+)FOW had greater variance in Allied force casualties (ranging from 1 to 4 at 1M time steps). Finally, MARDOC(-) should be interpreted as the best performing approach (utilizing the same amount of doctrine as MARDOC(-)FOW) with respect to Allied agents lost (compare Figure 6c to a, b, and d).

Similar to the results for battle win rate in Figure 5, the Adversarial force casualties metric reveals that (Figure 7) MARLFOW converged on max Adversarial casualties before MARDOC(-), then MARDOC(+)FOW, followed by MARDOC(-)FOW. Additionally, these results show that only MARDOC(-) was able to eliminate all 8 of the Adversarial agents across the 10 models. However, an important difference between the results shown in Figures 5 and 7 is that a lost battle can occur when either all Allied forces have been eliminated or the episode had reach the 120 time step limit and at least 1 Adversarial force was not eliminated. As an example, Figure 5b (MARDOC(+)FOW) has one model that only reached about 30% win rate, but Figure 7b (MARDOC(+)FOW) shows that this model achieved around 7 average defeated Adversaries, which indicates that although this model did not win frequently, it did almost win on average but ran out of time (see Figure 8b).

Although the line plots in Figure 8 are fairly noisy, the results reveal 2 important differences between MARL-



(a) MARDOC(-)FOW

(b) MARDOC(+)FOW

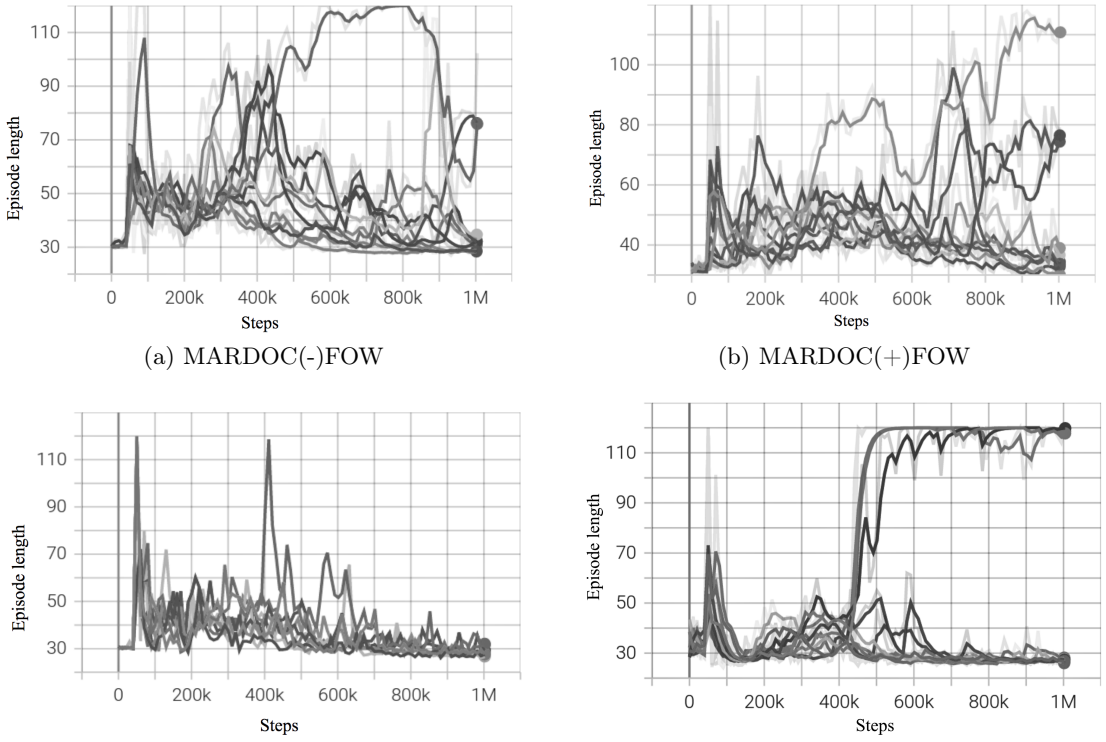
(c) MARDOC(-)

(d) MARLFOW

Figure 7: Adversarial force casualties across 10 independent models per algorithmic approach. The x-axes show number of total training or learning steps ranging between [0, 1M] and the y-axes show the number of Adversarial agents defeated ranging between [0, 8], with 8 equivalent to a battle won. (a) MARDOC(-)FOW with a MEAN = 7.41 and STD = 1.83 at 1M time steps. (b) MARDOC(+)FOW with a MEAN = 7.89 and STD = 0.25 at 1M time steps. (c) MARDOC(-) with a MEAN = 8 and STD = 0 at 1M time steps. (d) MARLFOW with a a MEAN = 5.62 and STD = 3.84 at 1M time steps.

FOW and MARDOC approaches. First, none of the MARDOC approaches converged on a max or near max episode length (around 120 time steps) as the results show for 3 of the 10 MARLFOW models (see Figure 8d from [500k, 1M]). Second, 2 of the 3 MARDOC approaches showed substantially more exploration than MARLFOW, evidenced by the greater average episode length in the later stages of training (compare Figure 8b and c to d between [800k, 1M]). It is important to note that MARDOC(-) and MARLFOW were quite similar in this metric for successful models (those that achieved high win rates), but it appears that MARDOC(-) had greater agreement across the 10 models with a converged episode length of about 29 time steps with little standard deviation (about 2 time steps). In contrast, MARLFOW had 3 degenerate models that converged on the max episode length (120 time steps). In addition, the results suggest that all approaches had models that either achieved or were tending to an average episode length of 30 time steps. Finally, although the learning approaches show agreement with episode length, the emergent behaviors or learned policies were observably different, as is shown in the next section.

In the next section, a preliminary interpretation of the emergent learned behaviors is presented with respect to each of the best performing models from the 4 learning approaches (MARLFOW, MARDOC(-)FOW, MARDOC(+)FOW, and MARDOC(-)). These interpretations were extracted manually from visual observation of the respective converged policies. Convergence was determined from the results shown in Figures 5 and 7, which directly tie to the reward function used to train the RODE algorithm in all approaches.



(a) MARDOC(-)FOW

(b) MARDOC(+)FOW

(c) MARDOC(-)

(d) MARLFOW

Figure 8: Average episode length across the 4 learning approaches. Each data point show the number of time steps per episode, averaged over the time steps since the previous data point. The x-axes show the number of training time steps ranging between [0, 1M], and the y-axes show the average episode duration ranging between [27, 120]. (a) MARDOC(-)FOW with a MEAN = 42.11 and STD = 24.92 at 1M time steps. (b) MARDOC(+)FOW with a MEAN = 49.68 and STD = 29.07 at 1M time steps. c) MARDOC(-) with a MEAN = 28.63 and STD = 1.96 at 1M time steps. (d) MARLFOW with a MEAN = 55.04 and STD = 44.86 at 1M time steps.

4. LEARNED POLICY INTERPRETATIONS

In the experiments conducted for this paper, doctrinal maneuver allowed the Allied force agents to explore targeted parts of the state space, which effectively focused the RODE algorithm exploration to smaller portions of the state space, leading to more desired behavior and improved performance over a state-of-the-art MARLFOW approach.

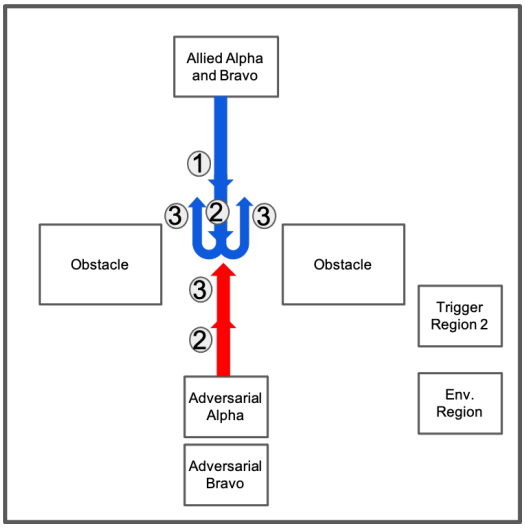
In general, a MARL converged policy is heavily dependent on the Adversarial agents' strategy or behavior, and the terrain (i.e., the locations of obstacles, sensor or trigger regions, force divisions, and initial positions) within an environment. In the SMAC environment, the default built-in AI Adversarial force agent's strategy was to prioritize the order of Allied force engagement based on the order of sensors (i.e., Trigger Regions 1 and 2) triggered during an episode. This indicates that the Adversarial agent's strategy was to pursue and engage the nearest Allied force agent associated withe the most recent Trigger Region sensor, which can become a highly exploitable strategy. With this exploit in mind, the following subsections describe interpretations of emergent strategies for the best performing policies from each of the 4 learning approaches (see Figures 9, 10, and 11).

In this section, we present policy (or learned behavior) interpretations of the best performing model for each algorithmic approach. The interpretation is based on visual inspection of the replay video of roll-outs (i.e., video replays of episodes to capture agent behavior). In Figures 9, 10, and 11, *Trigger Region 1* is removed to alleviate clutter and emphasize the demonstrated strategies. Finally, the factors that might have led to the observed behaviors are discussed, and what can be done in future experiments to improve this current thrust of

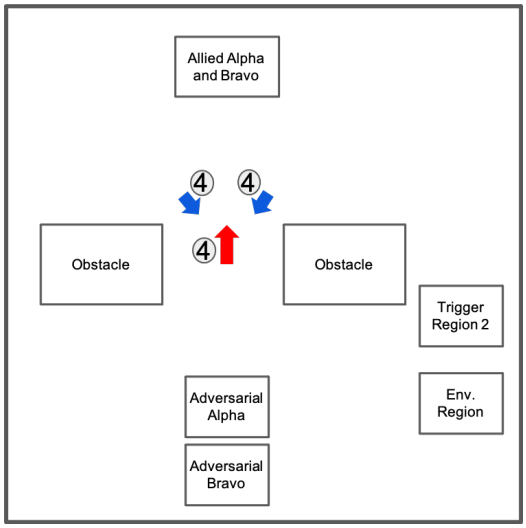
work (i.e., utilizing MARL with doctrine to overcome the difficulty of training models in large, militarily relevant state spaces).

4.1 MARLFOW

The scenario and the settings are described in Section 2.6. In short the Allied forces start in the same location at the beginning of each episode, the Adversarial forces handle the sensor triggers in first-in-first-out (FIFO) order, and an Allied Alpha or Bravo force remains invisible if no member of either Allied Alpha or Allied Bravo force enters either trigger region. Figure 9 shows a phased (phase 1 and phase 2) interpretation of the best performing MARLFOW policy to help describe the observed behavior.



(a) First phase of the MARLFOW policy. The Allied forces advanced, retreated, and split to draw the Adversarial forces out.



(b) Second phase of the MARLFOW policy. The Allied forces engaged with concentrated fire power, overwhelming the Adversarial forces as they drew near.

Figure 9: MARLFOW policy interpretation for the best performing MARLFOW model. (a) Phase 1 shows the Allied forces advance to Trigger Region 1 (blue arrow numbered 1) and continue advancing a distance into the trigger region (blue overlapping arrow numbered 2), triggering the Adversarial forces to advance towards the Allied forces (red arrow numbered 2), which resulted in the Allied forces to turn around (blue arrow numbered 3) as the Adversarial forces continued to pursue (red arrow numbered 3). (b) Phase 2 shows the Adversarial forces continue to advance between the obstacles (red arrow numbered 4) towards the split Allied forces (blue arrows numbered 4). Note that Trigger Region 2 or the Envelopment Region (Env. Region) do not get activated by the Allied forces.

For the best performing model of the MARLFOW algorithm, the learned policy is described here and in Figure 9. The first phase of the policy interpretation in Figure 9a shows, (1) blue arrow indicates that both of the Allied Alpha and Bravo forces advance towards *Trigger Region 1* to trigger the sensor. (2) The red and blue arrows indicate that the Adversarial and Allied forces advance towards each other. (3) As the Adversarial forces continue forward (red arrow), the Allied forces begin to retreat (blue hooked arrows), which effectively draws the Adversarial forces to follow. In Figure 9b, the second phase of the policy interpretation is shown. (4) The Adversarial forces engage the Allied forces according to the trigger order, which causes the Adversarial forces to pursue Allied agents that have moved to the rear of the Alpha and Bravo teams, resulting in a barrage of attacks before the built-in AI is able to attack the agents in the FIFO queue.

Since the Adversarial forces handle the triggers in a FIFO order, the Allied forces partially learned to position themselves in reverse order to make the Adversarial force highly exploitable. However, this policy (observed emergent behavior) only occurred in 1 of the 10 trained models. Further, of the 10 MARLFOW models, none

learned to utilize or exploit "Trigger Region 2" (i.e., behavior where Allied agents divide into two teams and execute doctrine-like ambush or envelopment maneuvers was not observed). In addition, an interesting available exploit where Allied forces could learn to avoid triggering the sensor, making them effectively "invisible" from the Adversarial forces' perspective was not observed. This observation is potentially due to ineffective exploration of the state space (possibly due to the need for additional training steps), or lack of sufficient reward to explore unexplored states.

The observations associated with the interpretation of the best performing MARLFOW model imply that MARLFOW alone may not be able to to find or demonstrate team configurations, resulting in different capabilities (e.g., encounter available exploits) even in a relatively small state space, which further reinforces that the integration of doctrinal knowledge might be critical for the deployment of learning approaches in militarily relevant scenarios. In future experiments, it would be of interest to evaluate the impact of changing the order of Adversarial force handling of triggers (e.g., random or last-in-first-out [LIFO]) on the trained Allied force policy.

4.2 MARDOC(-)FOW and MARDOC(-)

Given the differences in performance between MARDOC(-)FOW and MARDOC(-) shown in Figures 5, 6, 7, 8, it is surprising that the best performing models from these 2 approaches converged upon the same policy (see Figure 10). Although, it is important to note that all 10 (100%) of the MARDOC(-) models converged upon the policy described in Figure 10, whereas, only 7 of the 10 (70%) MARDOC(-)FOW models demonstrated this behavior.

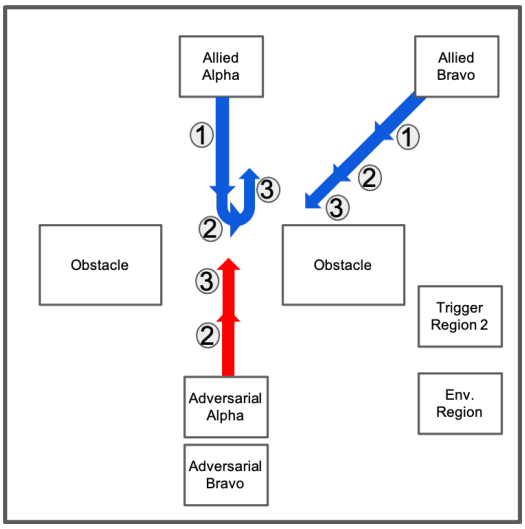
To begin each episode, the Allied forces (Allied Alpha and Bravo) were initialized to spatially separated locations as a part of the *envelopment doctrinal maneuver*, with no additional fixed policy doctrinal maneuver implemented (details for the similarities and differences are described for MARDOC(-)FOW and MARDOC(-) in Section 2.7). Figure 10 represents an interpretation of the learned behavior or strategy that emerged after 1M time steps of training.

The policy interpretation for MARDOC(-)FOW and MARDOC(-) was broken into 2 phases with colored and numbered arrows to help describe the emergent policy that unfolded over time. For the first phase of the policy interpretation (see Figure 10a): (1) the Allied Alpha force advanced towards (and triggered the sensor) "Trigger Region 1." In response to the sensor being triggered, (2) the Adversarial forces moved to engage the Allied Alpha force. (3) As the Adversarial forces approached, the Allied Alpha force retreated to draw the Adversarial forces past the Obstacles, while simultaneously, the Allied Bravo force advanced towards the edge of the nearest Obstacle to ambush the Adversarial forces. In the second phase of the policy interpretation (see Figure 10b), it was observed that the Allied Alpha force turned back towards the Adversarial forces and engaged, while the Allied Bravo force ambushed or effectively engaged in a side flanking maneuver to eliminate the Adversarial forces.

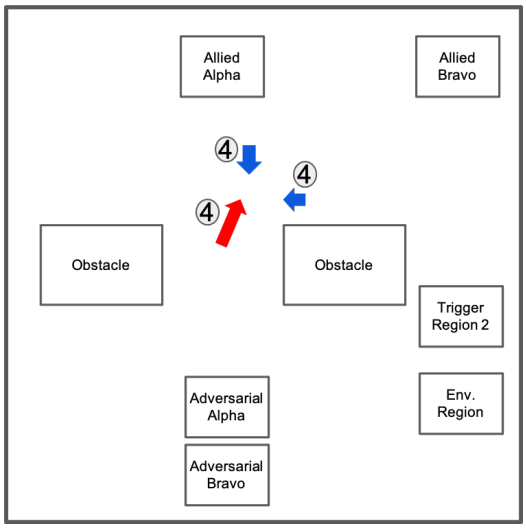
In the policy interpretation for MARDOC(-)FOW and MARDOC(-), the Allied forces learned to lure the Adversarial forces into an ambush. In addition, the Allied Bravo force learned to avoid triggering any sensors. The doctrinal implementation of dividing the Allied forces into teams, resulted in: 1) coordination between the 2 Allied forces, and 2) different but complementary behaviors or roles emerged for each force (Allied Alpha and Bravo). The Allied Alpha force learned to lure the Adversarial forces to a vulnerable area, while the Allied Bravo force waited in a secure location to ambush. The results from these experiments validate that there is benefit from integrating doctrine into learning paradigms, leading to interpretable outcomes.

4.3 MARDOC(+)FOW

The scenario and the settings of MARDOC(+)FOW are described in Section 2.7. Figure 11 represents the policy interpretation from the best performing model (see Figure 3, red line plot in a-d). For this approach, Allied Alpha and Bravo forces were controlled with a fixed maneuver behavior until they reached "Trigger Region 1" and the "Envelopment Region" respectively (see Figure 11a, blue arrows labeled 1 for Allied Alpha and 3 for Allied Bravo). The policy interpretation progression can be described as: (1) Using a fixed policy, the Allied Alpha force approached "Trigger Region 1" and the Allied Bravo force approached the "Envelopment Region". (2) As the respective Allied forces trigger the sensors, the Adversarial forces approached the Allied Alpha force first, because the envelopment maneuver was conducted in such a way that the Allied Alpha force would trigger



(a) First phase of the MARDOC(-)FOW and MARDOC(-) policy. The Allied Alpha force drew the Adversarial forces back towards the edge of an Obstacle where the Allied Bravo force was ready to engage.

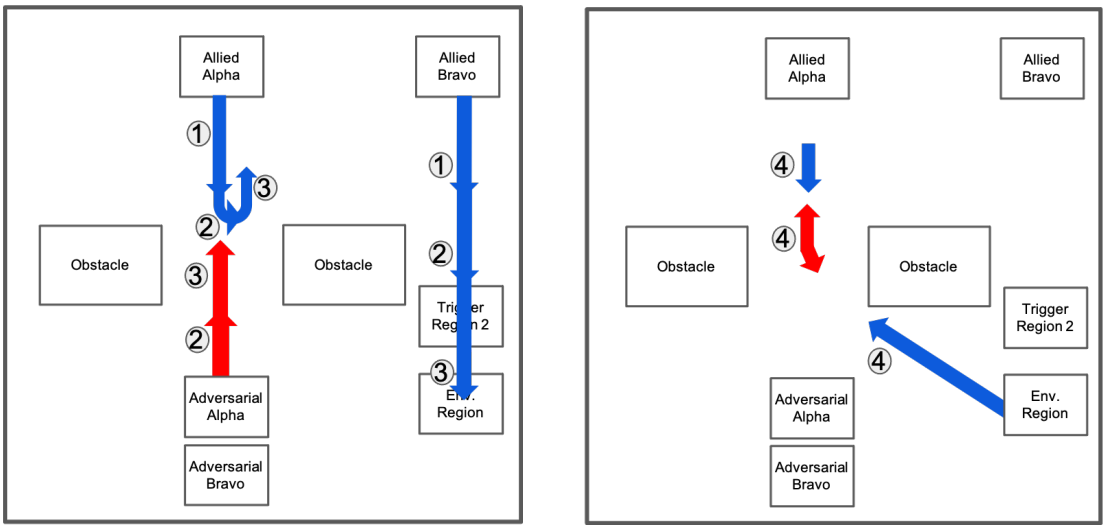


(b) Second phase of the MARDOC(-)FOW and MARDOC(-) policy. The Allied Alpha force engaged the Adversarial forces while the Allied Bravo force ambushed from the edge of the nearest Obstacle.

Figure 10: MARDOC(-)FOW and MARDOC(-) policy interpretation for the majority of models per approach. (a) Phase 1 shows the Allied Alpha force advance to Trigger Region 1 (blue arrow numbered 1) while Allied Bravo force advanced towards the edge of the nearest Obstacle. The Adversarial forces advanced towards the Allied Alpha force (red arrows numbered 2 and 3) causing the Allied Alpha force to retreat (blue curved arrows numbered 2 and 3), as the Allied Bravo force continued towards the Obstacle edge (diagonal blue arrows numbered 2 and 3). (b) Phase 2 shows the Allied Alpha force turned back towards the advancing Adversarial forces to engage, just past the Obstacle where the Allied Bravo force was ready to ambush or flank (red and blue arrows numbered 4).

a sensor before the Allied Bravo force. This was done to enforce the envelopment doctrinal maneuver, where the fixing force (Allied Alpha in this case) kept the enemy engaged, so the attacking force (Allied Bravo) could neutralize the enemy from behind (or flank from the side) before the Adversarial forces could reorients. (3) As the Adversarial forces got closer, the Allied Alpha force backed up to keep the Adversarial force pursuing, while at the same time, the Allied Bravo force started to engage (or attack) from the Envelopment Region. (4) The Allied Alpha force kept the Adversarial force engaged from the front, while the Allied Bravo force attacked from the behind before the Adversarial forces could effectively reorient themselves to handle the triggering of the second sensor.

In this approach, the Allied Alpha force learned to draw the Adversarial forces back past the Obstacles, to keep them engaged as the Allied Bravo force attacked the from the Envelopment Region (effectively from behind or the side), before they were able to effectively reorient themselves. This policy verifies the effectiveness of the envelopment doctrinal maneuver in this particular scenario, because there was only an average of 1 Allied force lost (see Figure 3b, red line) and a perfect win rate (see Figures 3b and c, red lines). Further, the Allied force casualties were the same as MARDOC(-)FOW (compare Figure 6a to b). However, for all 10 models, the bound for the Allied force casualty rate was tighter for MARDOC(-)FOW compared to MARDOC(+)FOW as shown in Figure 6. The differences observed between MARDOC(-)FOW and MARDOC(+)FOW must have arisen from the fixed policy implementation over Allied Bravo, since the behavior of Allied Alpha was very similar in both approaches (compare phase 1 of Figure 10a to Figure 11a). For MARDOC(+)FOW, the doctrinal maneuver might be the cause of having a less tight bound on Allied force casualty rate (i.e., greater variance between the 10 trained models), since the fixed maneuver guided the Allied forces to explore a different part of the state space. This implies that different degrees (or levels) of doctrinal guidance for exploration can have a dramatic impact



(a) First phase of the MARDOC(+)FOW policy. The Allied Alpha force drew the Adversarial forces to the far side of the Obstacle where the Allied Bravo force advanced towards the Envelopment Region.

(b) Second phase of the MARDOC(+)FOW policy. The Allied Alpha force engaged the Adversarial forces while the Allied Bravo force ambushed from the Envelopment Region.

Figure 11: Policy interpretation divided into 2 phases to help describe the behaviors learned by best performing MARDOC(+)FOW model. (a) The first phase of the policy interpretation for MARDOC(+)FOW began with the fixed policy (i.e., advance towards selected regions on the map), which transitioned to control by the RODE algorithm at blue arrows numbered '2' for Allied Alpha and number '3' for Allied Bravo. As the Adversarial forces advanced towards Trigger Region 1 (red arrows numbered 2 and 3), the Allied Alpha force began to turn and retreat (blue curved arrows numbered 2 and 3), while the Allied Bravo force advanced towards the Envelopment Region (blue straight arrows labeled 2 and 3). (b) The second phase has the Allied Alpha force turn to engage the pursuing Adversarial forces while the Allied Bravo force engaged the Adversarial forces from behind.

on the RODE algorithm's ability to effectively explore state spaces in search of novel and better strategies.

In future work, it would be of interest to observe the emergent Allied force behavior from the MARDOC(+)FOW approach when the fog of war is removed (i.e., MARDOC(+)). The primary reason this approach was not compared in this research was due to resource constraints. The next steps are to run and evaluate the MARDOC(+) approach to those shown in this paper. Finally, we would predict that the MARDOC(+) models would outperform MARDOC(+)FOW as was the case with MARDOC(-) over MARDOC(-)FOW.

5. CONCLUSIONS

One of the greatest challenges facing the US Army is and will continue be, an ever-increasingly connected world with an untenable state space for AI agents to learn within (or optimize over action selection and decision making). In the military domain, the 'states' (of an environment) are inherently distributed across large spatial areas (e.g., a city, mountain, cyber, space). In addition to a vast state space, a lack of methodical policy analysis (or behavioral analysis) mechanisms, leads to a superficial evaluation of MARL algorithms, which typically only evaluate how quickly they converge to a solution.

The specific solution a MARL policy converges upon (i.e., different behaviors or strategies that yield the same average converged reward), may not be desirable, and is often ignored in most research. This can become a major issue in the military domain because a converged policy (or strategy) needs to be realistically implementable and not just an exploit of the reward function. Our experiments show that the MARL algorithm can converge frequently to infeasible strategies—ones that cannot be implemented in the real world due to terrain and capability constraints—in a vast and complex terrain that requires more sophisticated strategies where random exploration

alone does not work well. In this work, we showed that US Army doctrinal integration has a huge impact on the converged policy that can exhibit drastically different behavior depending on the degree of doctrinal integration. Our experiments clearly show that doctrine inspired knowledge can be used to bootstrap the learning of more sophisticated behavior (e.g., lure and ambush or envelopment maneuver); the integrated behavior from doctrine guides the exploration to relevant places to exploit desirable and feasible strategies. We believe that the guided exploration was the reason behind finding policies with low Allied force casualties that facilitated maneuver from different directions without requiring an exploration term in the reward function. The MARDOC trained agents were able to demonstrate sophisticated team behavior and intelligent role emergence. On the other hand, MARL alone was not able to learn any useful coordinated behavior relevant to the military.

For our future work we would like to extend the capability of the adversary by using more advanced and aggressive trigger handling mechanisms. We would also like to extend the capability of the Allied force by replacing the fixed policy with a trained network during long distance maneuvers since there is a possibility of an adversary attacking during that period. Since we observed that SMAC is capable of capturing battalion level doctrine up to a decent level of resolution where a sophisticated policy can emerge through doctrinal guidance, we believe this framework will allow us to refine or update doctrinal knowledge with new ideas through a scientific approach that is effective and appropriate for military domains, and may result in better strategies to exploit windows of superiority, and thus contribute to the military decision making process (MDMP).

Acknowledgement

All published reports, journal articles, or professional presentations that are based on the activities conducted during your appointment shall carry an acknowledgment such as the following: Research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-20-2-0209. 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] Goecks, V. G., Waytowich, N., Asher, D. E., Park, S. J., Mittrick, M., Richardson, J., Vindiola, M., Logie, A., Dennison, M., Trout, T., et al., "On games and simulators as a platform for development of artificial intelligence for command and control," *arXiv preprint arXiv:2110.11305* (2021).

[2] Army, U., "Combined arms battalion." `https://armypubs.army.mil/epubs/DR_pubs/DR_a/ARN32974-ATP_3-90.5-000-WEB-1.pdf` (Jul 2021). Accessed: 2022-02-18.

[3] Goecks, V. G., Waytowich, N., Watkins, D., and Prakash, B., "Combining learning from human feedback and knowledge engineering to solve hierarchical tasks in minecraft," *arXiv preprint arXiv:2112.03482* (2021).

[4] Bera, R., Goecks, V. G., Gremillion, G. M., Lawhern, V. J., Valasek, J., and Waytowich, N. R., "Gaze-informed multi-objective imitation learning from human demonstrations," *arXiv preprint arXiv:2102.13008* (2021).

[5] Schaefer, K. E., Perelman, B., Rexwinkle, J., Canady, J., Neubauer, C., Waytowich, N., Larkin, G., Cox, K., Geuss, M., Gremillion, G., et al., "Human-autonomy teaming for the tactical edge: The importance of humans in artificial intelligence research and development," in [*Systems Engineering and Artificial Intelligence*], 115–148, Springer (2021).

[6] Cèsar-Tondreau, B., Warnell, G., Stump, E., Kochersberger, K., and Waytowich, N. R., "Improving autonomous robotic navigation using imitation learning," *Frontiers in Robotics and AI* **8** (2021).

[7] Waytowich, N. R., Goecks, V. G., and Lawhern, V. J., "Cycle-of-learning for autonomous systems from human interaction," *arXiv preprint arXiv:1808.09572* (2018).

[8] Fernandez, R., Asher, D. E., Basak, A., Sharma, P. K., Zaroukian, E. G., Hsu, C. D., Dorothy, M. R., Kroninger, C. M., Frerichs, L., Rogers, J., et al., "Multi-agent coordination for strategic maneuver with a survey of reinforcement learning," tech. rep., US Army Combat Capabilities Development Command, Army Research Laboratory (2021).

[9] Fernandez, R., Zaroukian, E., Humann, J. D., Perelman, B., Dorothy, M. R., Rodriguez, S. S., and Asher, D. E., "Emergent heterogeneous strategies from homogeneous capabilities in multi-agent systems," in [*Advances in Artificial Intelligence and Applied Cognitive Computing*], 491–498, Springer (2021).

[10] Asher, D. E., Zaroukian, E., Perelman, B., Perret, J., Fernandez, R., Hoffman, B., and Rodriguez, S. S., "Multi-agent collaboration with ergodic spatial distributions," in [*Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications II*], **11413**, 114131N, International Society for Optics and Photonics (2020).

[11] Asher, D., Garber-Barron, M., Rodriguez, S., Zaroukian, E., and Waytowich, N., "Multi-agent coordination profiles through state space perturbations," in [*2019 International Conference on Computational Science and Computational Intelligence (CSCI)*], 249–252, IEEE (2019).

[12] Barton, S. L., Waytowich, N. R., Zaroukian, E., and Asher, D. E., "Measuring collaborative emergent behavior in multi-agent reinforcement learning," in [*International Conference on Human Systems Engineering and Design: Future Trends and Applications*], 422–427, Springer (2018).

[13] Zaroukian, E., Rodriguez, S. S., Barton, S. L., Schaffer, J. A., Perelman, B., Waytowich, N. R., Hoffman, B., and Asher, D. E., "Algorithmically identifying strategies in multi-agent game-theoretic environments," in [*Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*], **11006**, 1100614, International Society for Optics and Photonics (2019).

[14] Zaroukian, E., Basak, A., Sharma, P. K., Fernandez, R., and Asher, D. E., "Emergent reinforcement learning behaviors through novel testing conditions," in [*Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications III*], **11746**, 117460T, International Society for Optics and Photonics (2021).

[15] Fernandez, R., Basak, A., Howell, B., Hsu, C., Zaroukian, E., Perret, J., Humann, J., Dorothy, M., Sharma, P. K., Nivison, S., et al., "Multi-agent collaboration in an adversarial turret reconnaissance task," in [*International Conference on Intelligent Human Systems Integration*], 38–43, Springer (2021).

[16] Hsu, C. D., Jeong, H., Pappas, G. J., and Chaudhari, P., "Scalable reinforcement learning policies for multi-agent control," in [*2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*], 4785–4791, IEEE (2020).

[17] Barton, S. L., Zaroukian, E., Asher, D. E., and Waytowich, N. R., "Evaluating the coordination of agents in multi-agent reinforcement learning," in [*International Conference on Intelligent Human Systems Integration*], 765–770, Springer (2019).

[18] Barton, S. L., Waytowich, N. R., and Asher, D. E., "Coordination-driven learning in multi-agent problem spaces," *arXiv preprint arXiv:1809.04918* (2018).

[19] Sharma, P. K., Zaroukian, E., and Asher, D. E., "Emergent behaviors in multi-agent target acquisition," in [*Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications IV*], International Society for Optics and Photonics (2022).

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

[21] Liu, I.-J., Jain, U., Yeh, R. A., and Schwing, A., "Cooperative exploration for multi-agent deep reinforcement learning," in [*International Conference on Machine Learning*], 6826–6836, PMLR (2021).

[22] Peng, B., Rashid, T., Schroeder de Witt, C., Kamienny, P.-A., Torr, P., Böhmer, W., and Whiteson, S., "Facmac: Factored multi-agent centralised policy gradients," *Advances in Neural Information Processing Systems* **34** (2021).

[23] Mahajan, A., Rashid, T., Samvelyan, M., and Whiteson, S., "Maven: Multi-agent variational exploration," *Advances in Neural Information Processing Systems* **32** (2019).

[24] McFarlane, R., "A survey of exploration strategies in reinforcement learning," *McGill University* (2018).

[25] Sharma, P. K., Zaroukian, E., Fernandez, R., Basak, A., and Asher, D. E., "Survey of recent multi-agent reinforcement learning algorithms utilizing centralized training," in [*Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications III*], **11746**, 117462K, International Society for Optics and Photonics (2021).

[26] Bellemare, M., Srinivasan, S., Ostrovski, G., Schaul, T., Saxton, D., and Munos, R., "Unifying count-based exploration and intrinsic motivation," *Advances in neural information processing systems* **29** (2016).

[27] Wang, T., Wang, J., Wu, Y., and Zhang, C., "Influence-based multi-agent exploration," *arXiv preprint arXiv:1910.05512* (2019).

[28] Stadie, B. C., Levine, S., and Abbeel, P., "Incentivizing exploration in reinforcement learning with deep predictive models," *arXiv preprint arXiv:1507.00814* (2015).

[29] Tang, H., Houthooft, R., Foote, D., Stooke, A., Xi Chen, O., Duan, Y., Schulman, J., DeTurck, F., and Abbeel, P., "# exploration: A study of count-based exploration for deep reinforcement learning," *Advances in neural information processing systems* **30** (2017).

[30] Yuan, M., Pun, M.-o., Wang, D., Chen, Y., and Li, H., "Multimodal reward shaping for efficient exploration in reinforcement learning," *arXiv preprint arXiv:2107.08888* (2021).

[31] Basak, A., Fang, F., Nguyen, T. H., and Kiekintveld, C., "Abstraction methods for solving graph-based security games," in [*International Conference on Autonomous Agents and Multiagent Systems*], 13–33, Springer (2016).

[32] Basak, A., Fang, F., Nguyen, T. H., and Kiekintveld, C., "Combining graph contraction and strategy generation for green security games," in [*International Conference on Decision and Game Theory for Security*], 251–271, Springer (2016).

[33] Wang, T., "SMAC - starcraft multi-agent challenge." `https://github.com/oxwhirl/smac` (Oct 2020). Accessed: 2022-02-18.

[34] Wang, T., "RODE: Learning roles to decompose multi-agent tasks." `https://github.com/TonghanWang/RODE` (Oct 2020). Accessed: 2022-02-18.

[35] Wang, T., Gupta, T., Mahajan, A., Peng, B., Whiteson, S., and Zhang, C., "Rode: Learning roles to decompose multi-agent tasks," *arXiv preprint arXiv:2010.01523* (2020).

[36] Army, U., "Fm 3-90-1: Offense and defense volume 1." `https://armypubs.army.mil/epubs/DR_pubs/DR_a/NOCASE-FM_3-90-1-002-WEB-0.pdf` (March 2013). Accessed: 2022-02-18.