ARL-TR-9350 • Dec 2021

DEVCOM ARMY RESEARCH LABORATORY

Multi-Agent Coordination for Strategic Maneuver with a Survey of Reinforcement Learning

by Rolando Fernandez, Derrik E Asher, Anjon Basak, Piyush K Sharma, Erin G Zaroukian, Christopher D Hsu, Michael R Dorothy, Christopher M Kroninger, Luke Frerichs, John Rogers, and John Fossaceca

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

ARL-TR-9350 ● DEC 2021

Multi-Agent Coordination for Strategic Maneuver with a Survey of Reinforcement Learning

by Rolando Fernandez, Derrik E Asher, Piyush K Sharma, Erin G Zaroukian, Michael R Dorothy, John Rogers, and John Fossaceca
Computational and Information Sciences Directorate, DEVCOM Army Research Laboratory

Christopher D Hsu and Christopher M Kroninger
Weapons and Materials Research Directorate, DEVCOM Army Research Laboratory

Anjon Basak
Oak Ridge Associated Universities

Luke Frerichs
Booz Allen Hamilton

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED (From - To)
December 2021	Technical Report	July–September 2021

4. TITLE AND SUBTITLE
Multi-Agent Coordination for Strategic Maneuver with a Survey of Reinforcement Learning

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Rolando Fernandez, Derrik E Asher, Anjon Basak, Piyush K Sharma, Erin G Zaroukian, Christopher D Hsu, Michael R Dorothy, Christopher M Kroninger, Luke Frerichs, John Rogers, and John Fossaceca

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
DEVCOM Army Research Laboratory
ATTN: FCDD-RLC-IS
Aberdeen Proving Ground, MD 21005-5066

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TR-9350

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
primary author's email: <rolando.fernandez1.civ@army.mil>.

14. ABSTRACT
One promising avenue for implementing strategic maneuver to gain superiority over adversaries is through coordination of multi-agent systems (MAS) in future military operations. Recent work exploring the coordination of MAS has focused on the identification, classification, validation, implementation, and operationalization of emergent coordination through multi-agent reinforcement learning. Reinforcement Learning (RL) approaches can illuminate emergent behaviors through the exploration and exploitation of selected actions in a given environment, potentially leading to the inhibition of adversarial coordination which, in turn, can provide windows of opportunity across various intelligence, surveillance, target acquisition, and reconnaissance tasks. In this report, we present a brief overview of salient work in the RL domain and its potential applications for collaborative MAS for autonomous strategic maneuver.

15. SUBJECT TERMS
Multi-Agent Systems, Reinforcement Learning, Multi-Domain Operations, Coordination, Military Scenario, Strategic Maneuver

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Rolando Fernandez
Unclassified	Unclassified	Unclassified	UU	33	19b. TELEPHONE NUMBER (Include area code) 301-956-4098

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

Acknowledgments

This project was supported in part by an appointment to the Science Education Programs at National Institutes of Health, administered by Oak Ridge Associated Universities through the US Department of Energy Oak Ridge Institute for Science and Education.

1. Introduction

The nascent surge in the US Army modernization is motivated by the threat adversaries pose to the nation in multiple domains (e.g., land, sea, air, cyber, electromagnetic, and space),[1–3] which threatens the national interest in ways beyond conventional warfare. It is expected that future battles will be fought in these complex multi-domain environments,[4–6] where artificial intelligence (AI) will guide the tactics, techniques, and procedures (TTPs) of robotic agents working alongside human Soldiers. Such robots will aggregate to form intelligent multi-agent teams coordinating efficiently with human Soldiers to complete the mission.

The US Army Combat Capabilities Development Command (DEVCOM) Army Research Laboratory's (ARL) Essential Research Programs (ERPs) constitute a specific programmatic path for developing and implementing intelligent multi-agent systems (MAS). Such Army programs provide US defense operations with answers to critical research questions that converge to support the Army Futures Command modernization efforts. Artificial Intelligence for Autonomous Maneuver and Mobility (AIMM)[7] and Emerging Overmatch Technologies (EOT) are example ERPs that explicitly focus on enabling the Next-Generation Combat Vehicles with autonomous sensing, learning, reasoning, planning, and maneuvering capabilities. These future autonomous systems will predict and plan in collaboration with human agents and provide support to the Soldier through autonomous maneuver (AIMM) and protection (EOT) on the battlefield. This report focuses on the autonomous collaboration that needs to take place in order to have multi-agent systems (i.e., human, agent, or human and agent) succeed in future military operations.

Integrated and coordinated MAS will require technological advancements focused on collaborative strategic maneuvers beyond our current capabilities to effectively deal with equivalently equipped adversaries (peer or near-peer). One immediate challenge is to develop teams of agents that can work autonomously and intelligently in a well-coordinated manner. This capability demands that agents observe, orient, decide, and act (OODA-Loop) alongside Soldiers in mission-critical tasks. While novel efforts have contributed to a general understanding of *intelligence* in multi-agent paradigms, the current interpretation of intelligence is not well-defined.[8,9] Recent literature suggests that *Reinforcement Learning (RL)*–based approaches may provide a viable path towards such technological advances, evi-

denced by a body of work introduced here.

In this report, contributions in the RL domain are presented along with their potential applications in military environments— specifically on coordination through strategic team maneuver to inhibit adversarial coordination for battlefield overmatch. The minimization, limitation, or complete inhibition of coordination in adversarial multi-agent behaviors is a means of exploring and executing strategic maneuvers derived through RL experimentation in simulated situations. Moreover, collaborative strategic maneuvers can be learned by various RL approaches to inform a defense force of potential avenues for creating windows of opportunity or superiority.

To achieve MAS coordination through strategic maneuver with RL approaches in simulation environments, we first introduce some of the most prominent RL implementations of recent years. These recent advancements in the RL domain (e.g., alphago[10]) have facilitated the use of more complex multi-agent reinforcement learning (MARL) algorithmic approaches towards eventual real-world application. Further, there have been some frameworks to implement multi-agent coordination in recent years.[11–15] Together, these efforts may provide a path towards developing and implementing multi-agent coordination for strategic maneuver in multi-robot systems designed for the future battlefield.

In the following sections, a taxonomy and overview of salient RL approaches in recent years are presented, and these approaches are shown to align with current research and development programs at the DEVCOM Army Research Laboratory. Specifically, this report focuses on determining the advantages and disadvantages of select algorithmic approaches for strategic maneuver. Further, selected classes of RL approaches are classified to give insight on potential implementations for strategic maneuver with intelligence, surveillance, target acquisition, and reconnaissance (ISTAR) tasks in mind.

2. Strategic Maneuver with Multi-Agent Systems in Multi-Domain Operations

In simple terms, strategic maneuver can be interpreted as a set of agents coordinating their actions to achieve a common goal by overcoming an adversary. Disruption, which is a special case of strategic maneuver, can be represented as the inhibition of an adversary's coordinated strategic maneuver. Therefore, the use of the terms strategic maneuver implies that there exists at least two opposing or adversarial sides that are in a dynamic struggle to gain superiority over each other by limiting, inhibiting, or otherwise disrupting their opponent's coordination or tactics, and imposing their own coordinated tactics.

In this section, an adversarial engagement scenario is provided that is centered on the use of selected long-range assets that inherently disrupt friendly force engagement. A legend is shown in Fig. 1 to describe the military symbology for selected assets and forces associated with the described Multi-Domain Operation (MDO) scenario. According to MDO[16] doctrine, in an armed conflict, adversarial long-range anti-access and area-denial (A2AD) fire systems can be used to deny friendly forces freedom of maneuver in a theater of operations (see Fig. 1). This is accomplished by combining intelligence, surveillance, and reconnaissance (ISR) assets with both lethal and nonlethal fires to attack friendly command structures, sustainment capabilities, and troop formations in the Strategic and Operational Support Areas. These areas are the traditional staging ground for assets (e.g., troops and equipment) oper-

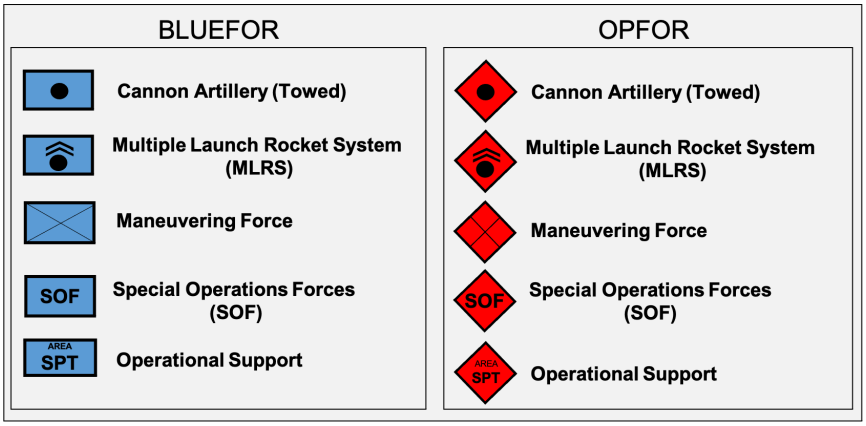

Fig. 1 Assets and resources for the friendly (BLUEFOR, left) and opposition (OPFOR, right) forces. In the described MDO scenario, it is assumed that all assets are autonomy-enabled formations for both BLUEFOR and OPFOR.

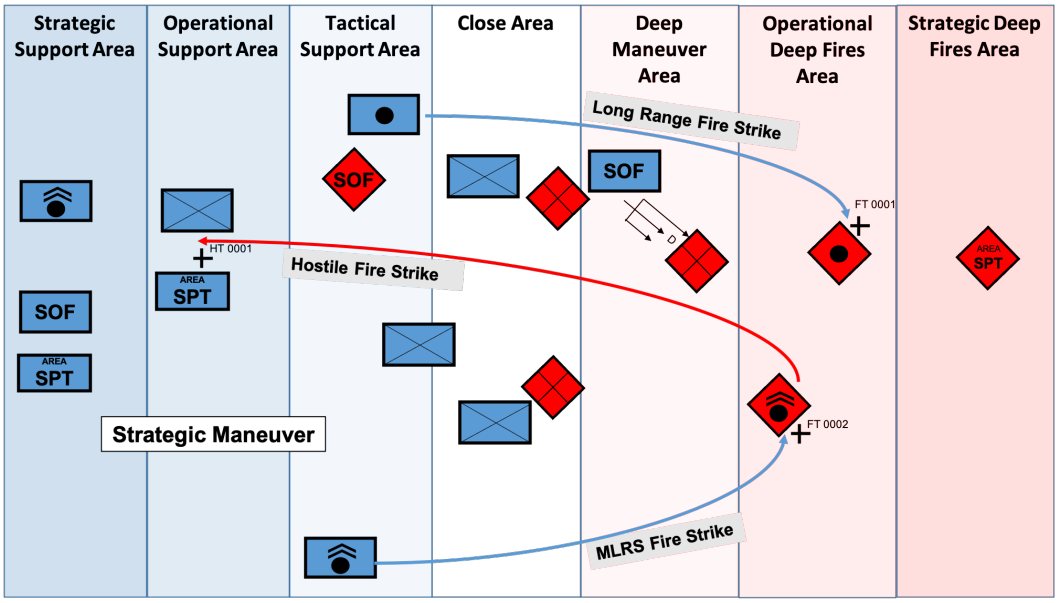

Fig. 2 Adversarial forces (OPFOR) use long-range missile and rocket fire to disrupt or destroy sustainment operations in the friendly (BLUEFOR) Strategic Support Area, which prevents friendly forces from engaging enemy maneuver elements in the Close Area on favorable terms. To counter this strategy, BLUEFOR conducts counterfire missions to destroy OPFOR long-range fire systems located in the Deep Fires Area (blue arrow). The three-pronged arrow emanating from the BLUEFOR SOF in the Deep Maneuver Area represents a "Disrupt" tactic which breaks up the adversary's formation and tempo.[17]

ating in the Close Area (see Fig. 2). The adversary's ability to identify and engage targets deep behind friendly lines causes those entities to be geographically separated from the Tactical Support and Close Areas, which effectively raises the attrition rate of friendly forces, referred to as *stand-off*. Given that the forward force is separated from strategic and operational maneuver support, adversarial forces can take advantage of this friendly force isolation and destroy them.

The MDO[16] doctrine lays down a plan to defeat adversarial A2AD capability (i.e., stand-off) so that strategic and operational maneuvers can enable forward-deployed friendly forces to engage the adversary on favorable terms (i.e., penetrate and disintegrate A2AD systems to exploit freedom of maneuver). Here we focus only on the penetration and disintegration portions of an engagement* with adversarial A2AD systems by friendly (BLUEFOR) field army and corps that may entail the use of

*There are six joint functions from the doctrinal phases of a joint operation. These functions are Command and Control (C2), Intelligence, Fires, Movement and Maneuver, Protection, and Sustainment. Penetration and disintegration are part of the Fires and Movement and Maneuver joint functions.

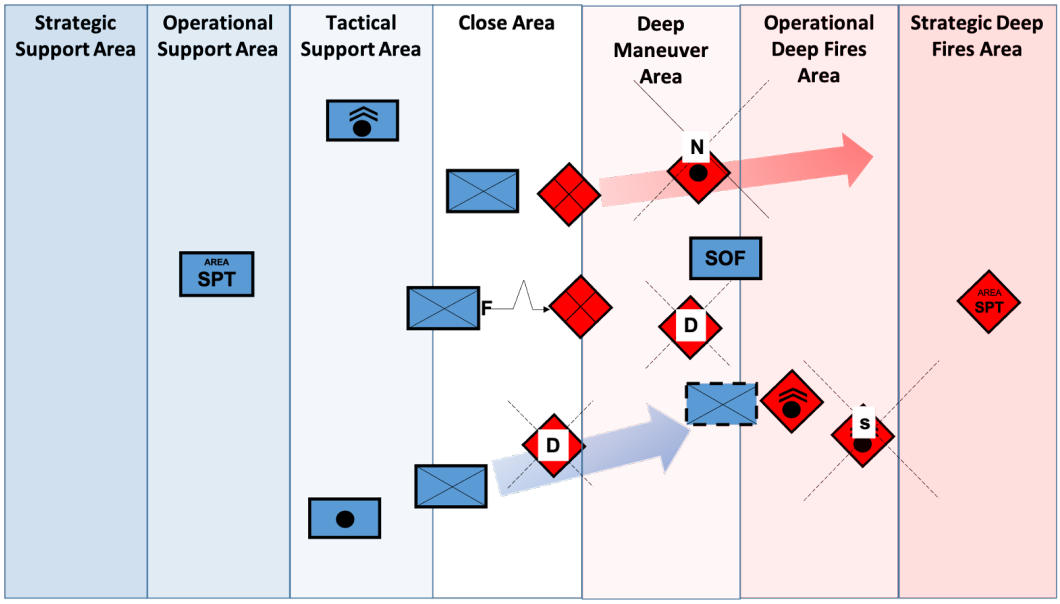

Fig. 3 Suppressing (S) or neutralizing (N) the enemy long-range fire systems and ISR assets enables friendly forces to penetrate the adversary's A2AD umbrella. This allows friendly forces to defeat the enemy in the Close Area and gives the maneuver commander the ability to exploit their success by rapidly moving forces into the Deep Maneuver Area to destroy (D) vulnerable enemy assets and pursue retreating enemy forces. The F indicates "Fix" which effectively slows the adversary's movement. The thick arrows represent the direction of troop movements.[17]

autonomous MAS in future battles. Further, it is speculated that all of the symbols shown in Fig. 1 for both friendly (BLUEFOR) and adversary (OPFOR) forces will contain autonomy-enabled formations (e.g., Robotic Combat Vehicles, automatic targeting systems, ground and aerial robotic ISR assets). Scenario diagrammatics for strategic maneuver utilizing this symbology with the autonomy-enabled formations are shown, respectively, in Figs. 2 and 3.

The adversarial A2AD fire systems create stand-off by attacking the Strategic and Operational Support Areas as shown in Fig. 2. The friendly fires and air defense forces receive targeted intelligence from space and high-altitude surveillance (not shown) to strike high-value targets (i.e., Multiple Launched Rocket System [MLRS]) within narrow time windows to reduce adversarial position adjustments. In addition to surveillance, strategic *stimulation-see-strike* can be employed to penetrate and disintegrate adversarial long-range fire systems.[16]

MARL can be used to strategically illuminate and track the locations of adversarial targets in ISTAR tasks by exploiting adversarial doctrine and local observa-

tions from adversarial actions. Further, MARL-trained autonomy-enabled formations with a combination of highly mobile and distributed air and ground fires can begin to overwhelm adversarial long-range air defenses. Friendly forces may utilize MARL approaches trained to exploit adversarial TTPs for strategic maneuver of air-defense and ground fires. These autonomy-enabled formations choose geographic locations based on surveillance data collected from strategic air-based stimulation. As adversarial long-range fire systems are neutralized, strategic and operational support units are able to advance (maneuver) towards the forward OPFOR (see Fig. 2).

Adversarial forces identify friendly assets in the Operational Support Area using ISR assets and engage friendly forces with long-range fire systems (i.e., MLRS) from the Operational Deep Fires Area. These hostile fires disrupt the friendly force's ability to conduct traditional support operations in that area, which in turn causes such activities to take place farther back from the forward line of troops. This creates geographical stand-off by extending the battlefield and straining supply lines. Further, this permits the hostile maneuver forces to engage friendly forces in the Close Area on terms favorable to the adversary *fait accompli*.* According to MDO doctrine, to eliminate stand-off, friendly artillery systems must identify, engage, and destroy hostile fires and ISR assets before they can be deployed. Friendly SOF assist this effort by disrupting supply and command and control (C2) nodes and providing targeting data for Joint Fires. This creates gaps in the enemy A2AD umbrella, which can be exploited by maneuver commanders. **Under this coverage, friendly maneuver penetrates and then exploits gaps in the Close and Deep Maneuver Areas**.

Strategic formations of joint forces in the Close and Deep Areas, starting from the Operational Area, may be autonomous-enabled formations (i.e., MAS) utilizing MARL-trained policies to exploit adversarial TTPs (from doctrine), local observations, and ISR-gathered information. Joint forces will coordinate between their ISR and long-range precision fires capabilities to provide support for the forward deployed BLUEFOR forces as shown in Fig. 2. With support from strategic and operational units, forward forces with autonomous-enabled formations can coordinate in Close and Deep Areas to isolate and defeat adversarial assets. This leads to

*fait accompli is a term used here to describe an event that has already been decided before those affected hear about it, leaving them with no option but to accept it.

the elimination of the adversarial forward-maneuvering forces (OPFOR), leaving long-range fire systems vulnerable to ground attack (disintegration), as shown in Fig. 2.

Joint Fires (i.e., friendly forces or BLUEFOR) suppress or neutralize adversarial long-range fire systems, allowing friendly Maneuver Forces to move in and defeat OPFOR in the Close Area (see Fig. 3). Friendly Maneuver Forces then exploit this advantage by destroying adversarial enablers in the Deep Maneuver Area (see **D** in Fig. 3). This causes the remaining adversarial maneuver formations to withdraw from the Close Area and establish a new front in the Deep Maneuver Area. This process repeats until strategic objectives are met or OPFOR is defeated. These coordinated activities could in theory be achieved in a collaboration between human Soldiers and an autonomous multi-agent system. Further, given that there is active research in the development and deployment of such autonomous systems, it is expected that battlefields of the future will need to consider scenarios like this for planning of strategic maneuver.

This section has provided a scenario where autonomous-enabled formations trained by MARL approaches can be applied; however, the specific RL approaches to perform in such complex MDO environments have not been tested or may not yet exist. The following section illuminates some of the challenges associated with utilizing RL approaches to train MAS for future MDO engagements.

3. Challenges

In this work, we narrow our focus to RL approaches that can guide a MAS to overcome the challenges associated with strategic maneuver in military defense MDO. Technically, RL is a branch of machine learning (ML) that goes beyond building accurate predictions from data by demonstrating learning with the production of actions in an environment. This demonstration of learning can be considered a form of decision-making but is more accurately described as *strategic action selection* through state-space exploration.

RL agents learn (or are trained) based on a reward function that ultimately determines which is the best action for an agent to select given the current situation (i.e., state of that agent situated in an environment). For example, an RL agent can interact with an environment to produce experiences that are tied to rewards, which

will result in a learned policy (i.e., a series of state-action pairs). However, in later sections it is highlighted that current RL approaches may not yet be mature enough to overcome the challenges associated with human-like adaptability for intelligent decision-making in novel situations or environments. Although RL algorithms have their shortcomings,[18] they appear to be one of the most promising avenues moving forward towards achieving coordinated MAS performing strategic maneuver in military defense MDO.

Coordination is typically ill-defined in multi-agent tasks and is often used to indicate that a group of agents has performed successfully in some cooperative task domain.* In prior work, various novel methods were developed and employed to measure the interdependence between agent actions while performing cooperative tasks, to confirm that these agents had in fact learned to coordinate.[19–26] The confirmation of coordination is a precursor to establishing that a MAS is capable of working with its partners instead of simply taking actions that result in some measure of optimality. Whereas optimal behavior may be desirable in some circumstances, an agent that is simply acting optimal may result in catastrophic losses on a battlefield if the mission has changed in some unforeseen way. Therefore, it is critical that MAS for future defense operations has the capability to explicitly coordinate.

For the remainder of this section, some of the challenges associated with developing MAS capable of strategic maneuver are described, where timescales, capabilities, and local goals may be vastly different (e.g., MDO), but some level of coordination is required. Further, it is assumed that a greater degree of flexible coordination can result in improved (e.g., faster, less losses, nonintuitive strategies, effectively handle changing capabilities/team composition) mission execution.

As an environment changes in response to a dynamic battlefield, the two (at least) adversarial sides may need to reuse plans and predictions in order to either 1) keep up with, or 2) stay ahead of the planning and predictions of an opponent. An RL-trained MAS may be able to learn this dynamic planning and prediction cycle. Alternatively, this can be achieved if the learning agents build an appropriate model of their opponent's coordinated actions and then take actions to disrupt that coordination.

*More explicitly, coordination is a measurable quantity that describes the interdependent relationship between two or more entities, whereas cooperation is inferred and describes the alignment of goals in a given task domain.

In an ideal case, an algorithm selected to guide behavior of a MAS would learn to deal with changes in the environment, adversary tactics and capabilities, own capabilities (gain new ones or lose previous ones), team composition (e.g., changing out cooperative partners), and local goals. However, most state-of-the-art (sota) approaches are limited by experiences (as is the case with many RL approaches).[27] Moreover, team capabilities and composition are typically fixed in most simulations and do not provide sufficient data for an algorithm to operate on and handle any of the aforementioned feature changes. Therefore, in selecting an algorithm to guide behavior of a MAS, intended for producing strategic maneuver, it is important that novel or dynamic events, behaviors, assets, and entities be considered.

In summary, current algorithmic approaches fall short of the required capability in a complex military defense MDO environment. Current shortcomings can be divided into three categories: 1) data requirements, where either data is limited due to the novelty of a situation, a data set is insufficient to produce accurate predictions, or the data is polluted in some way (e.g., noisy, dirty, or adversarial alteration), 2) limited computational resources, and 3) algorithms do not generalize to situations other than what was encountered during training (e.g., different goals, altered capabilities, or modified team composition) resulting in a narrow or brittle MAS solution.

In the next section, we address RL shortcomings in greater detail to illuminate how overcoming such issues may produce solutions for military defense MDO environments. To do so, an existing taxonomy of RL algorithms is introduced. This effort should provide a better insight into promising RL techniques that may help determine viable avenues for eventual application in US defense MDO.

4. RL Techniques and Approaches

Scalability of a learning algorithm is one of the major concerns for military tasks in MDO, especially since such tasks may require a large number of agents to complete an objective. In addition, military tasks can involve multiple subtasks each with their own subgoals, further complexifying the scenario. In MDO, it is expected that a subgoal consists of a myriad of complex strategic maneuvers that would require fast computation for MAS, and an optimal (or at least sufficient) strategy with the use of minimal computational resources (e.g., computing at the tactical edge). Therefore, a scalable RL algorithm must account for 1) environmental and task complexities and 2) number of agents (partners and adversaries) so each agent can properly select

actions as experiences are collected through the RL learning process.

Environmental complexity (i.e., the size of an agent's state and action spaces) can refer to the number of states available in an environment's state space along with the number and actions available to agents in that environment. Scalability of an RL algorithm is the capability of computing an optimal policy within reasonable time and computational power for a sufficiently complex state and action space*. Environmental complexity also entails the inclusion of additional agents (e.g., expanding to a MAS), where the state space is scaled up to account for the extra agents, and the size of the action space is multiplied by that number of agents.

It is not practical to tackle the scalability issue in RL by using a table for state-action pairs because continuous domains would make the table untenable, and simultaneously updating the entries of a table for all agents is infeasible within a reasonable amount of time. Even with sufficiently large computational resources (e.g., excessive computer memory) to contain all states, learning across each state-action pair would be too slow. In contrast to utilizing a table for tracking state-action pairs, a solution is to use a nonparametric function approximator (e.g., deep neural network where the weights are the parameters) that approximates values across the entire state space. However, a function approximator must be differentiable, such that a gradient can be calculated to provide the direction of parameter adjustments.

There are two approaches to train value function approximators: 1) incremental methods and 2) batch methods. Incremental methods use a stochastic gradient to adjust the approximator's parameters in the direction of the gradient to minimize the error between the estimated and target values. However, the incremental approach is not sample efficient, and therefore does not lend itself to scalability. In contrast, batch methods save the data from a set of experiences and use them to compute the error between function approximator estimation and the target value. Batch methods share commonalities with traditional supervised learning, where the outcome is known (e.g., data is labeled) and an error is calculated between the approximator's estimate value and the actual outcome value. This type of batch learning is typically referred to as experience replay. Repeating this process will lead to a least-square error solution. A recent successful example of experience replay was demonstrated with deep Q-networks (DQN)[28] playing Atari games. Although

*Sufficiently complex is used here as an arbitrary term that can be identified post-hoc through trial and error methods (e.g., varying state or action spaces).

function approximator methods have shown success in complex environments, it is unlikely that this approach alone will be sufficient to train a MAS for MDO scenarios without accounting for the inclusion of additional agents (i.e., non-stationarity or partial-observability).

Compared to value function approximation, policy learning methods rely on policy gradient (PG) calculations to explicitly optimize a policy rather than indirectly on a value function. PG has better convergence properties over function approximator methods*. The main reason PG methods are used over value approximation methods is their ability to be effective in high dimensional and continuous action spaces (i.e., scalable in complex environments). In Monte Carlo (MC) policy gradient (e.g., REINFORCE algorithm[29]), the actual return (from selecting an action) is multiplied with a score function to calculate the gradient. That gradient is used for policy adjustment (by changing the values of parameters) to find the greatest rewarding action. An MC policy gradient has high variance and is slow to converge, since it uses the entire trajectory of an agent's state-action pairs across time to get a return value. An alternate solution that may surpass the shortcomings of traditional function approximator methods is to utilize Actor-Critic approaches.

With Actor-Critic approaches, a PG equation is modified to use the value function approximation instead of using the true action-value function multiplied by the score (as is done with the REINFORCE algorithm[29]). This indicates that the actor adjusts the policy in the direction that the critic is pointing so that total cumulative rewards can be maximized. This policy evaluation step by the critic can be done by using combined value approximation methods (i.e., MC, Temporal Difference - TD(0) and TD(λ)). To reduce the variance in the policy gradient, an advantage function can be used.[30] The advantage function tells us how much better one action is over another (Q-value) compared to a general state value function. This implies that the critic must estimate the Q-value. An efficient way to do this is to use TD-error, which is an unbiased sample of the advantage function where the critic approximates one set of parameters. TD(λ) eligibility traces can also be used for the critic to estimate the value across different time steps. Interestingly, MC (high variance) and TD methods can be used with the actor to modify the policy over time (i.e., collected experiences).

*Policy convergence is an important property that implies an algorithmic approach reaches a stable level of reward and performance after a sufficient number of training iterations.

Since MDO involves military tasks where an RL algorithm must have the capability to coordinate with many other agents for optimal strategic maneuvers, an algorithm for MAS must be able to scale with a large number of agents and heterogeneous assets. Another important capability of an algorithm is the ability to process the vast observations of complex state-spaces (i.e., many agents) and multi-domain environments. In the next sections, we discuss the implication of using different kinds of RL algorithms in MDO for strategic maneuver.

Model-free algorithms can be divided into off-policy and on-policy algorithms where state-action space can be either continuous or discrete. In this section, the strengths and weaknesses of the model-free algorithms are discussed, along with how they might align with strategic maneuver, leading to the goals of MDO. The purpose of this analysis is to provide direction towards finding potential algorithmic approaches that may achieve strategic maneuver in MDO environments.

4.1 Deep Q-Network

Deep Q-Network (DQN)[28] is a single RL agent algorithm that was trained to play Atari 2600 games[31] where the action space was discrete and the state space was continuous. DQN uses a convolutional neural network trained with Q-learning[32] to learn from high-dimensional input (sequential images).

The DQN algorithm is a sample efficient approach since it makes use of all the collected experiences to extract the maximum amount of information possible. DQN is robust enough to be trained using the same hyperparameters* to play six different Atari games, where the agent performed better than human experts in three of these games.

However, a drawback with DQN is that there are no theoretical guarantees of a trained neural network achieving a stable Q-Value prediction (i.e., potentially high variance in the trained policies across independent models).

Given that DQN is inherently a single RL agent model, it should be insufficient for strategic maneuver in MDO. In MDO, multi-agent RL algorithms may be more suitable due to the typical decentralization of the agents during execution time, allowing for agents to operate independent from one another. In addition, the original

*In machine learning, a hyperparameter is a parameter whose value is used to control the learning process. By contrast, the values of other parameters (typically node weights) are derived via training.

implementation of DQN only utilizes a sequence of four observations to learn the Q-value, which is insufficient for strategic maneuver in MDO. Strategic maneuvers of multiple assets cannot typically be captured within such short time intervals. In fact, this is the primary reason that DQN did not perform well compared to humans in three of the Atari games evaluated (i.e., Q*bert, Seaquest, and Space Invaders). However, there exist some variations of DQN to address this and other weaknesses.

Bootstrap DQN[33] is one such variation that learns an ensemble of Q-Networks to improve sample efficiency and overcome the shortcomings of traditional DQN. Action elimination is another method used with DQN to tackle large action spaces.[34] DQN with a type of memory (i.e., recurrent neural network) can be used to handle partial observability as well.[35] This approach is particularly useful if an agent needs to navigate an environment for task completion. Alternatively, distributional DQN[36,37] returns a distribution that can be used to evaluate policy risk and to reduce variance or noise around an optimal solution.

Although DQN and its modified variants are promising for tackling tasks more complicated than simple Atari games, the DQN method inherently lacks a multi-agent prediction mechanism to conduct coordinated tactics, which are required for strategic maneuver in MDO. Further, DQN is most often too computationally intensive to be used in militarily relevant environments. Finally, DQN algorithmic approaches lack sufficient adaptability for unseen examples (e.g., novel behaviors of partners or entities/obstacles emerge in an environment).

4.2 Deep Deterministic Policy Gradient

In the real world, most regular tasks involve continuous state and action spaces. However, DQN only considers discrete state spaces and low-dimensional action spaces. An alternative approach to DQN, where continuous state and action spaces are handled, is the Deep Deterministic Policy Gradient (DDPG) method. DDPG advances the progress from DQN approaches by combining value function approximation and deterministic policy gradient (DPG).[38] DDPG utilizes an actor-critic approach,[39] which can overcome the complexities of continuous spaces. This model-free, off-policy prediction and control algorithm can perform physical control tasks (e.g., cart pole, dexterous manipulation, legged locomotion, or car driving).

Another approach that uses a deep neural network is trust region policy optimiza-

tion (TRPO). This method constructs a stochastic policy directly[40] without the need for an actor-critic model (not to be confused with an environment model which would make this a model-based approach). Similar to TRPO, guided policy search (GPS)[41] is void of an actor-critic model and uses trajectory-guided supervised policy learning along with some additional techniques (e.g., reduction of dimension from visual features, additional information on robot configuration dynamics at the first layer of the network). As a result, GPS is data-efficient and can be adapted to DDPG if required. PILCO,[42] on the other hand, learns a probabilistic model first, then finds an optimal policy. PILCO is highly data efficient in some problem domains; however, it is computationally demanding. Further, D4PG[43] proposed some improvements over the DDPG algorithm: distributional critic update, distributed parallel actors, N-step returns, and prioritization of the experience replay to achieve a more stable and better solution for a different category of tasks.

From the perspective of strategic maneuver, the primary drawback of the DDPG algorithm is that it was designed as a fully decentralized single agent algorithm (i.e., independent learners). As such, the DDPG algorithm does not facilitate coordination in multi-agent scenarios. Consequently the resulting strategic maneuvers using DDPG will not result in coordinated team behaviors. Moreover, DDPG is not equipped to handle role-based tasks with multiple objectives, which might be a requirement for strategic maneuver in military operations.

4.3 Multi-Agent Deep Deterministic Policy Gradient

RL agent interaction is crucial to AI systems for strategic maneuver where different agents may need to form teams to effectuate strategic collaboration against an adversary or inhibit the adversary's coordination. Q-Learning and PG approaches alone suffer from nonstationarity* and high variance, respectively. To overcome these issues, the Multi-Agent Deep Deterministic Policy Gradient (MADDPG)[44] algorithm extends an actor-critic approach, which allows it to work for multi-agent systems by centralizing agent training. The MADDPG framework adopts a centralized critic for training and deploys decentralized actors during test time. A critic (one for each agent) receives the policy of every agent, which allows for the development of dependent policies with potentially different reward functions (e.g.,

*Nonstationarity generally refers to an environment where sudden concept/goal/task drift can occur due to dynamic and unknown probability data distribution functions associated with other agents taking actions that change local goals.

MADDPG permits training of adversarial teams with opposing reward functions). Conversely, the actors (i.e., policy networks) only have local knowledge during training and testing. The actor improves the policy iteratively (through training) in a direction consistent with the critic's evaluation.

A major weakness of MADDPG is that the input to the Q-function increases with the number of agents of the environment (not scalable). This poses a problem for strategic maneuver in MDO. If agents need to be replaced, added, modified, or removed, retraining may need to take place. In strategic maneuver, agents may need to switch roles or change capabilities periodically, which poses a major challenge towards adapting MADDPG to military domains. In addition, frequent retraining would make rapid strategic maneuver unlikely. Reducing training time will reduce the computational load on the edge and make rapid strategic maneuver possible. MADDPG cannot accommodate such extreme cases. For military applications, a robust model of opponents or agents is desired so that the operational period is maximized (i.e., enough time to execute strategic maneuver).

A potential modification to MADDPG to address its scalability issues is to form clusters of agents and learn a policy for the clusters instead of each agent individually. In the case of a new event, the need for retraining can be postponed because, in theory, a cluster of agents would have a set of variable capabilities to handle dynamic situations. Further, this would avoid increasing the input space for the Q-function as agents are modified or new agents are introduced. However, the question arises: How can we decompose a task into partially independent subtasks with minimum degradation of an optimal group policy?

While MADDPG can lead to a set of heterogeneous multi-agent policies capable of diverse tasks, this approach does not scale well beyond a dozen agents. As the number of agents grows, the variance of the policy gradient grows exponentially. Therefore, this approach is not well suited for strategic maneuver in MDO where more than 40 heterogeneous agents must be accounted for in adversarial contexts. A method for overcoming this scalability issue is the Mean Field Multi-agent RL algorithm,[45] which computes a mean estimation for the neighborhood agents' Q-value that may result in high error margin when the nearby interaction between agents gets complex. Further, the Evolutionary Population Curriculum[46] algorithm was designed to make MADDPG scalable by combining genetic algorithmic ap-

proaches with RL. With advances upon MADDPG and the successes shown with the approach, it is conceivable that these algorithmic advances could lead to robust demonstrations of strategic maneuver within MDO in simulation experiments.

Distinct from MADDPG, the Counterfactual Multi-Agent (COMA)[47] approach uses a single centralized critic for all agents but is designed for discrete action spaces. COMA is more scalable than MADDPG, but it may result in a homogeneous set of policies that could fail with sufficiently different agent capabilities, different local goals, or different reward functions. Similar to MADDPG, Minmax Multi-Agent DDPG (M3DDPG)[48] adds an improvement over the original version of MADDPG by allowing agents to develop more robust policies against adversaries (i.e., competitive games with opposing reward structures). However, M3DDPG is still unable to handle scenarios when heterogeneous agents are introduced into the system.

Implementing algorithms into environments with continuous state and action spaces sometimes requires utilizing common techniques to manipulate the inputs or outputs, such as discretizing the state and action spaces or converting the discrete policy output to a continuous output. One example of converting the policy output is the implementation of MADDPG in the OpenAI Multi-Agent Particle Environment. In this example, the discrete policy components are utilized to compute continuous actions. In another perspective, the multi-agent transformer soft double Q-learning algorithm[49] discretizes the continuous action space into a set of velocity and angular velocity controls which can then be used in a motion model. Although these techniques permit the use of such algorithms in continuous environments, these algorithmic approaches do not train with continuous information, which could limit their efficacy in physical environments for strategic maneuver.

4.4 Value Based

A recent family of value-based MARL algorithms[50] has proven to be quite successful in the very complex Starcraft 2 simulation environment[?] where a centralized joint action-value Q_{tot} is learned based on the agents' local Q_a values. A decentralized policy is then extracted from the Q_a by taking the linear argmax operator. This very simple but efficient factorization approach avoids learning the joint action-value which, does not scale very well. If new agents are added or agents are replaced with new capabilities, retraining still has to be done. However, it is more scalable compared to MADDPG because the individual Q-values are learned from

local observation only which avoid learning the joint-action value by learning a factorized Q_{tot}. Still, the scalability of this family of algorithms can be challenged when there are more than 40 agents. To make it more scalable, the role-based algorithm RODE[51] has been proposed where the agents' roles are determined by clustering their actions based on their effect on the environment. The algorithm has shown very promising results for a large number of agents.

For strategic maneuver, the RODE algorithm is very promising since groups of agents can be assigned to different roles, where roles can be based on their action and effects on the environment or any other fixed behaviors (for ally or even enemy). The algorithm then can be used for strategic role switching for different groups. Since the action space of different roles is restricted, the algorithm converges very quickly. This algorithm is also fit for strategic use of role-based techniques, which may be investigated in future work. Even though RODE is very scalable, it is not clear how we can adapt it for when new agents will be added to the environment; a centralized policy needs to be learned for optimal coordination.

In contrast to the RODE algorithm, a scalable multi-agent reinforcement learning method[49] deploys an entropy-regularized off-policy method for learning a stochastic value function policy that has been experimentally shown to be able to scale to over 1000 agents. As discussed previously, scalable RL algorithms are concerned with the complexity of the environment—the more agents in the system or team, the larger the state space. RODE is limited as it uses a centralized policy that must be retrained when more agents are introduced into the environment. The algorithm, multi-agent transformer soft double Q-learning, is a centrally trained off-policy learning algorithm (i.e., sharing a central experience replay buffer) with decentralized execution (i.e., each agent makes its own control decisions based on its local observations) not from a central controller. Due to this decentralization scheme, when agents are added or removed from the system, the team is unaffected and continues to execute their policy.

With respect to scalability, training a large MAS (i.e., many agents) is difficult, and it has been shown that even state-of-the-art algorithms fail to learn performant policies for complex MARL tasks. Multi-agent transformer soft double Q-learning alleviates this scalability issue by utilizing a heuristic during training that allows for the policy to be trained on a smaller set of agents (e.g., four agents tracking four

targets in a target-tracking scenario), and the policy has been shown to work with many more agents in execution without any adaptation needed (i.e., tested and evaluated with 1000 agents). The heuristic used during training and execution allows the algorithm to address a dramatic distribution shift in the number of agents: it essentially scales down the large complex observation space at test time into something that is close to what the agent policy was originally trained for. In the military perspective, this formulation is ideal for strategic maneuver as agents in the field might be lost or gained in-situ and might have to account for additional strategic information. A flexible and scalable algorithm provides the capabilities needed to be robust in MDO.

5. Insights and Conclusions

US adversaries are becoming more advanced due to a number of factors, including scientific and technological progress made through proxy conflicts that test novel technologies. Coordinated strategic maneuver can be used by defense forces to give certain advantages over an adversary in future MAS autonomous warfare. In this article, some of the most prominent RL algorithms were discussed to uncover viable candidates for training MAS that can effectively perform strategic maneuver towards opening windows of opportunity in potential future military operations. A taxonomy of RL approaches was described with an overview for the most prominent RL algorithms. It was found that most RL algorithms lack the capability of handling the complexities associated with potential future conflicts due to differences in training and test factors.

DEVCOM ARL ERPs provide a programmatic path for developing and implementing intelligent MAS. Given that Army research programs provide US defense operations with answers to critical research questions, the AIMM and EOT ERPs specifically enable research that can provide a path towards coordinated autonomous MAS that can overcome the challenges associated with 1) an environment, 2) adversary tactics and capabilities, 3) own capabilities (i.e., gain new capabilities, lose previous capabilities, or have capabilities altered), 4) team composition (e.g., adding, removing, or swapping of teammates), 5) strategic team positioning, entry, navigate (maneuver) to support forces and overwhelm an adversary, and 6) mission objectives. Recent work in this domain by the AIMM and EOT ERPs has illuminated a means of measuring coordination in MAS[13,19,20,22–24] and allowed the development

for a framework that trains and tests the coordination of MAS performing various tasks, in addition to evaluating novel algorithmic approaches utilizing an array of centralized training techniques.[8]

Further, additional investigation is needed to illuminate military strategy that facilitates the utilization of MAS in ISTAR tasks and other engagement scenarios. In obvious cases, it is desirable to send fully autonomous MAS into high-risk situations (i.e., where expected causality rates are high); however, it is insufficient to simply expect that a MAS will be capable of achieving the mission in the absence of human oversight or intervention due to current technological limitations. Therefore, in future work, research to identify a robust set of engagement scenarios will be pursued. Finally, this line of work will lead to the eventual integration of autonomous MAS for coordinated strategic maneuver where possible in future military operations.

6. References

1. Sharma PK, Dennison M, Raglin A. IoT solutions with multi-sensor fusion and signal-image encoding for secure data transfer and decision making; 2020. http://www.ibai-publishing.org/html/proceedings_2020/pdf/proceedings_book_MDA-AI&PR_2020.pdf

2. Sharma PK, Raglin A. Image-audio encoding to improve C2 decision-making in multi-domain environment. In: 25th International Command and Control Research and Technology Symposium.

3. Sharma PK. Heterogeneous noisy short signal camouflage in multi-domain environment decision-making. Transactions on Mass-Data Analysis of Images and Signals. 2020;11(1):3–26.

4. Sharma PK, Raglin A. Image-audio encoding for information camouflage and improving malware pattern analysis. In: 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA); p. 1059–1064.

5. Sharma PK, Raglin A. IoT in smart cities: exploring information theoretic and deep learning models to improve parking solutions. In: The 5th IEEE International Conference on Internet of People (IoP).

6. Sharma PK, Raglin A. Iot: smart city parking solutions with metric-Chisini-Jensen-Shannon divergence based kernels. In: IEEE International Conference on Military Communications Conference (MILCOM).

7. Autonomous maneuver and mobility. https://www.arl.army.mil/opencampus/?q=AIMM;.

8. Sharma PK, Zaroukian E, Fernandez R, Basak A, Asher DE. Survey of recent multi-agent reinforcement learning algorithms utilizing centralized training. In: Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications III; Vol. 11746; p. 117462K.

9. Army researchers develop innovative framework for training AI; 2021. https://www.army.mil/article/247261.

10. Silver D, et al. Mastering the game of go with deep neural networks and tree search. Nature. 2016;529(7587):484–489.

11. Pierpaoli P, Ravichandar H, Waytowich N, Li A, Asher D, Egerstedt M. Inferring and learning multi-robot policies by observing an expert; 2019. arXiv preprint arXiv:1909.07887.

12. Barton SL, Asher D. Reinforcement learning framework for collaborative agents interacting with soldiers in dynamic military contexts. In: Next-Generation Analyst VI; Vol. 10653; p. 1065303.

13. Barton SL, Waytowich NR, Asher DE. Coordination-driven learning in multi-agent problem spaces; 2018. arXiv preprint arXiv:1809.04918.

14. Caylor JP, Barton SL, Zaroukian EG, Asher DE. Classification of military occupational specialty codes for agent learning in human-agent teams. In: Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications; Vol. 11006; p. 110060W.

15. Rodriguez SS, Chen J, Deep H, Lee JJ, Asher DE, Zaroukian E. Measuring complacency in humans interacting with autonomous agents in a multi-agent system. In: Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications II; Vol. 11413; p. 114130U.

16. Army Training and Doctrine Command (US). The U.S. Army in Multi-Domain Operations 2028; 2021 [accessed 2021 Jun 16]. https://www.army.mil/article/243754/the_u_s_army_in_multi_domain_operations_2028

17. Headquarters, Department of the Army. Army doctrine publication ADP 1-02: terms and military symbols. Createspace Independent Publishing Platform; 2018.

18. Dulac-Arnold G, Mankowitz D, Hester T. Challenges of real-world reinforcement learning; 2019. arXiv preprint arXiv:1904.12901.

19. Asher D, Garber-Barron M, Rodriguez S, Zaroukian E, Waytowich N. Multi-agent coordination profiles through state space perturbations. In: 2019 International Conference on Computational Science and Computational Intelligence (CSCI); p. 249–252.

20. Asher D, Barton S, Zaroukian E, Waytowich N. Effect of cooperative team size on coordination in adaptive multi-agent systems. In: Artificial Intelligence and

Machine Learning for Multi-Domain Operations Applications; Vol. 11006; p. 110060Z.

21. Asher DE, Zaroukian E, Perelman B, Perret J, Fernandez R, Hoffman B, Rodriguez SS. Multi-agent collaboration with ergodic spatial distributions. In: Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications II; Vol. 11413; p. 114131N.

22. Barton SL, Waytowich NR, Zaroukian E, Asher DE. Measuring collaborative emergent behavior in multi-agent reinforcement learning. In: International Conference on Human Systems Engineering and Design: Future Trends and Applications; p. 422–427.

23. Fernandez R et al. Multi-agent collaboration in an adversarial turret reconnaissance task. In: International Conference on Intelligent Human Systems Integration; p. 38–43.

24. Fernandez R, Zaroukian E, Humann J, Perelman B, Dorothy M, Rodriguez S, Asher D. Emergent heterogeneous strategies from homogeneous capabilities in multi-agent systems. Springer International Publishing; 2021. p. 491–498.

25. Zaroukian E, Rodriguez SS, Barton SL, Schaffer JA, Perelman B, Waytowich NR, Hoffman B, Asher DE. Algorithmically identifying strategies in multi-agent game-theoretic environments. In: Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications; Vol. 11006; p. 1100614.

26. Zaroukian E, Basak A, Sharma PK, Fernandez R, Asher DE. Emergent reinforcement learning behaviors through novel testing conditions. In: Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications III; Vol. 11746; p. 117460T.

27. Nguyen TT, Nguyen ND, Nahavandi S. Deep reinforcement learning for multiagent systems: a review of challenges, solutions, and applications. IEEE transactions on cybernetics. 2020;50(9):3826–3839.

28. Mnih V, Kavukcuoglu K, Silver D, Graves A, Antonoglou I, Wierstra D, Riedmiller M. Playing Atari with deep reinforcement learning; 2013. arXiv preprint arXiv:1312.5602.

29. Williams RJ. Simple statistical gradient-following algorithms for connectionist reinforcement Learning. Machine learning. 1992;8(3):229–256.

30. Mnih V, Badia AP, Mirza M, Graves A, Lillicrap T, Harley T, Silver D, Kavukcuoglu K. Asynchronous methods for deep reinforcement learning. In: International conference on machine learning; p. 1928–1937.

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

32. Watkins CJ, Dayan P. Q-learning. Machine learning. 1992;8(3-4):279–292.

33. Osband I, Blundell C, Pritzel A, Van Roy B. Deep exploration via bootstrapped dqn. In: Advances in neural information processing systems; p. 4026–4034.

34. Zahavy T, Haroush M, Merlis N, Mankowitz DJ, Mannor S. Learn what not to learn: action elimination with deep reinforcement learning. In: Advances in Neural Information Processing Systems; p. 3562–3573.

35. Hausknecht M, Stone P. Deep recurrent q-learning for partially observable MDPS. In: 2015 AAAI Fall Symposium Series.

36. Bellemare MG, Dabney W, Munos R. A distributional perspective on reinforcement learning; 2017. arXiv preprint arXiv:1707.06887.

37. Dabney W, Ostrovski G, Silver D, Munos R. Implicit quantile networks for distributional reinforcement learning; 2018. arXiv preprint arXiv:1806.06923.

38. Silver D, Lever G, Heess N, Degris T, Wierstra D, Riedmiller M. Deterministic policy gradient algorithms. In: Deterministic policy gradient algorithms.

39. Heess N, Wayne G, Silver D, Lillicrap T, Erez T, Tassa Y. Learning continuous control policies by stochastic value gradients. In: Advances in Neural Information Processing Systems; p. 2944–2952.

40. Schulman J, Heess N, Weber T, Abbeel P. Gradient estimation using stochastic computation graphs. In: Advances in Neural Information Processing Systems; p. 3528–3536.

41. Levine S, Finn C, Darrell T, Abbeel P. End-to-end training of deep visuomotor policies. The Journal of Machine Learning Research. 2016;17(1):1334–1373.

42. Deisenroth M, Rasmussen CE. Pilco: a model-based and data-efficient approach to policy search. In: Proceedings of the 28th International Conference on machine learning (ICML-11); p. 465–472.

43. Barth-Maron G, Hoffman MW, Budden D, Dabney W, Horgan D, Tb D, Muldal A, Heess N, Lillicrap T. Distributed distributional deterministic policy gradients; 2018. arXiv preprint arXiv:1804.08617.

44. Lowe R, Wu YI, Tamar A, Harb J, Abbeel OP, Mordatch I. Multi-agent actor-critic for mixed cooperative-competitive environments. In: Advances in neural information processing systems; p. 6379–6390.

45. Yang Y, Luo R, Li M, Zhou M, Zhang W, Wang J. Mean field multi-agent reinforcement learning; 2018. arXiv preprint arXiv:1802.05438.

46. Long Q, Zhou Z, Gupta A, Fang F, Wu Y, Wang X. Evolutionary population curriculum for scaling multi-agent reinforcement learning; 2020. arXiv preprint arXiv:2003.10423.

47. Foerster J, Farquhar G, Afouras T, Nardelli N, Whiteson S. Counterfactual multi-agent policy gradients; 2017. arXiv preprint arXiv:1705.08926.

48. Li S, Wu Y, Cui X, Dong H, Fang F, Russell S. Robust multi-agent reinforcement learning via minimax deep deterministic policy gradient. In: Proceedings of the AAAI Conference on Artificial Intelligence; Vol. 33; p. 4213–4220.

49. Hsu CD, Jeong H, Pappas GJ, Chaudhari P. Scalable reinforcement learning policies for multi-agent control; 2021.

50. Whiteson Research Lab. Python MARL framework; 2019 [accessed 2021 June 17]. https://github.com/oxwhirl/pymarl

51. Wang T, Gupta T, Mahajan A, Peng B, Whiteson S, Zhang C. Rode: Learning roles to decompose multi-agent tasks; 2020. arXiv preprint arXiv:2010.01523.

List of Symbols, Abbreviations, and Acronyms

A2AD	Anti-access and Area-denial
AI	Artificial Intelligence
AIMM	Artificial Intelligence for Autonomous Maneuver and Mobility
ARL	Army Research Laboratory
C2	Command and Control
COMA	Counterfactual Multi-Agent
DDPG	Deep Deterministic Policy Gradient
DEVCOM	US Army Combat Capabilities Development Command
DPG	Deterministic Policy Gradient
DQN	Deep Q-Network
EOT	Emerging Overmatch Technologies
ERP	Essential Research Programs
GPS	Guided Policy Search
ISR	Intelligence, Surveillance, and Reconnaissance
ISTAR	Intelligence, Surveillance, Target Acquisition, and Reconnaissance
M3DDPG	Minmax Multi-Agent Deep Deterministic Policy Gradient
MADDPG	Multi-Agent Deep Deterministic Policy Gradient
MARL	Multi-agent Reinforcement Learning
MAS	Multi-agent System
MC	Monte Carlo
MDO	Multi-Domain Operation
ML	Machine Learning
MLRS	Multiple Launched Rocket System
OODA	Observe, Orient, Decide, and Act
PG	Policy Gradient
RL	Reinforcement Learning
SOF	Special Operations Forces
TD	Temporal Difference
TRPO	Trust Region Policy Optimization
TTP	Tactics, Techniques, and Procedures

1 (PDF)	DEFENSE TECHNICAL INFORMATION CTR DTIC OCA
1 (PDF)	DEVCOM ARL FCDD RLD DCI TECH LIB
10 (PDF)	DEVCOM ARL FCDD RLC J FOSSACECA FCDD RLC A L FRERICHS FCDD RLC IA J ROGERS FCDD RLC IS M DOROTHY R FERNANDEZ FCDD RLC IT D ASHER P SHARMA E ZAROUKIAN FCDD RLW C C KRONINGER FCDD RLW VB C HSU