NPS-CS-23-002

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

OPTIMIZING NAVAL MOVEMENT USING DEEP REINFORCEMENT LEARNING

by

Armon Barton, Chris Darken, and Joseph Coble

September 2023

Approved for public release. Distribution is unlimited.

Prepared for: Naval Surface Warfare Center Crane Division
This research is supported by funding from the Naval Postgraduate School Naval Research Program (PE 0605853N/2098)
NRP Project ID: NPS-23-N059-C

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED	
		START DATE	END DATE
September 2023	Technical Report	October 2022	September 2023

4. TITLE AND SUBTITLE
Optimizing Naval Movement Using Deep Reinforcement Learning

5a. CONTRACT NUMBER	5b. GRANT NUMBER	5c. PROGRAM ELEMENT NUMBER
		PE 0605853N/2098
5d. PROJECT NUMBER	**5e. TASK NUMBER**	**5f. WORK UNIT NUMBER**
NPS-23-N059-C		

6. AUTHOR(S)
Armon Barton, Chris Darken, and Joseph Coble

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
1 University Circle
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER
NPS-CS-23-002

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)	11. SPONSOR/MONITOR'S REPORT NUMBER(S)
Naval Surface Warfare Center, Crane Division; Naval Postgraduate School, Naval Research Program	NSWC-CD; NRP	NPS-23-N059-C

12. DISTRIBUTION/AVAILABILITY STATEMENT

Distribution Statement A: Approved for public release. Distribution is unlimited.

13. SUPPLEMENTARY NOTES
The views expressed in this document are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

14. ABSTRACT
In a rapidly evolving maritime warfare landscape, the U.S. Navy and its allies require their crews to quickly identify optimal strategies for vessel engagements to ensure freedom of the seas. This necessity becomes more pronounced given the potential grave consequences of sub-optimal maneuvers, as illustrated by the cases of the *USS John McCain* and *USS Fitzgerald*. Recent advancements in Machine Learning (ML) and Artificial Intelligence (AI) offer a promising solution. There have been significant strides in implementing AI to outperform human experts in complex games such as Chess, Poker, and StarCraft, which now have the potential to also benefit real-time decision-making and wargaming in the naval domain. This study explores the potential for Reinforcement Learning (RL) techniques to be applied to naval contexts, which could provide valuable decision-support tools to ship captains and their staff by suggesting optimal movement strategies in complex maritime scenarios. In this study, exemplar naval scenarios were designed and modeled within a combat simulation environment, AI agents (consisting of a mix of rule-based, method-based, and value-based approaches) were designed, and the performances of these agents were evaluated and compared. The aim was to assess the agents' ability to identify optimal movements against a rule-based adversary, while also comparing these performances against human-level play. The insights drawn from this study contribute to ongoing research aimed at developing effective decision aids for ship captains in real-world operations.

15. SUBJECT TERMS
Reinforcement Learning, Wargaming.

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
a. REPORT	b. ABSTRACT	C. THIS PAGE		
Unclassified	Unclassified	Unclassified	UU	46

19a. NAME OF RESPONSIBLE PERSON
Armon Barton

19b. PHONE NUMBER
8316566567

STANDARD FORM 298 (REV. 5/2020)
Prescribed by ANSI Std. Z39.18

THIS PAGE INTENTIONALLY LEFT BLANK

NAVAL POSTGRADUATE SCHOOL
Monterey, California 93943-5000

Ann E. Rondeau
President

Scott Gartner
Provost

The report entitled "Optimizing Naval Movement Using Deep Reinforcement Learning" was prepared for Naval Surface Warfare Center (NSWC) Crane Division and funded by the Naval Postgraduate School Naval Research Program (NRP) (PE 0605853N/2098).

Further distribution of all or part of this report is authorized.

This report was prepared by:

BARTON.ARMON.C.1591693324 Digitally signed by BARTON.ARMON.C.1591693324 Date: 2023.10.27 08:22:27 -07'00'

Armon Barton
Assistant Professor

DARKEN.CHRISTIAN.J.1251898797 Digitally signed by DARKEN.CHRISTIAN.J.1251898797 Date: 2023.10.27 15:38:05 -07'00'

Christian Darken
Associate Professor

Reviewed by:

SINGH.GURMINDER.1258791815 Digitally signed by SINGH.GURMINDER.1258791815 Date: 2023.11.14 14:08:26 -08'00'

Gurminder Singh, Chairman
Computer Science

Released by:

SMITH.KEVIN.B.1230026366 Digitally signed by SMITH.KEVIN.B.1230026366 Date: 2023.12.02 06:45:53 -08'00'

Kevin B. Smith
Vice Provost for Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

In a rapidly evolving maritime warfare landscape, the U.S. Navy and its allies require their crews to quickly identify optimal strategies for vessel engagements to ensure freedom of the seas. This necessity becomes more pronounced given the potential grave consequences of sub-optimal maneuvers, as illustrated by the cases of the *USS John McCain* and *USS Fitzgerald*. Recent advancements in Machine Learning (ML) and Artificial Intelligence (AI) offer a promising solution. There have been significant strides in implementing AI to outperform human experts in complex games such as Chess, Poker, and StarCraft, which now have the potential to also benefit real-time decision-making and wargaming in the naval domain. This study explores the potential for Reinforcement Learning (RL) techniques to be applied to naval contexts, which could provide valuable decision-support tools to ship captains and their staff by suggesting optimal movement strategies in complex maritime scenarios. In this study, exemplar naval scenarios were designed and modeled within a combat simulation environment, AI agents (consisting of a mix of rule-based, method-based, and value-based approaches) were designed, and the performances of these agents were evaluated and compared. The aim was to assess the agents' ability to identify optimal movements against a rule-based adversary, while also comparing these performances against human-level play. The insights drawn from this study contribute to ongoing research aimed at developing effective decision aids for ship captains in real-world operations.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

As the U.S. Navy and its allies continue to ensure freedom of navigation worldwide, it has become increasingly necessary for sailors and decision-makers to quickly identify the optimal strategies when engaging other, potentially hostile vessels. In today's "gray zone" [1] conflicts or activities, ship captains and their crews are expected to make real-time decisions on the best actions to take in each and every interaction they encounter. This study adapts a combat simulation environment to accommodate naval scenarios and develops artificial intelligence (AI) agents designed to output optimal actions in any given scenario. The objective is to establish a foundational tool capable of accommodating a wide range of variables, thus facilitating the exploration of novel approaches to address emerging maritime challenges. Using this framework, the U.S. Navy can test and evaluate potential solutions without incurring the risks and costs associated with real-world experimentation—ultimately improving decision-making and strategic planning in various operational contexts, which include real-time naval engagements and operational wargaming.

A. DECISION MAKING IN REAL-TIME NAVAL ENGAGEMENTS

Naval interactions of high consequences are very common, in which even the slightest margins of error can be unacceptable. The ability for a ship captain, or Officer of the Deck (OOD), to identify the optimal path—not just for their ship but the adversary as well—can easily result in the damage of costly equipment and, even worse, the loss of human lives. These decisions are often made by qualified officers who stand in for the captain of the ship when needed. Unfortunately, a considerable amount of discrepancy exists in the skills and knowledge of these decision-makers, which is predominantly influenced by their past experiences in ship navigation and the extent of their training.

Examples of historic naval encounters where the lack of decision-support may have contributed to major casualties include: the collision between the *USS Fitzgerald* Guided Missile Destroyer (DDG 62) and the *ACX Crystal* which resulted in the loss of seven servicemembers [2]; and the collision between the *USS John S. McCain* Guided

Missile Destroyer (DDG 56) and the tanker *Alnic MC* which resulted in two damaged ships [3].

B. NAVAL WARGAMING

For over a century, the U.S. Navy has utilized wargaming as a crucial approach to evaluate theories and gain valuable insights during a peacetime environment [4]. The Naval War College's wargaming department is dedicated to conducting these games to gain a better understanding as to how future battles might be played out [5]. Moreover, the desire from leadership service-wide to implement AI into these wargames has been increasing, as the technology advances and decreases in cost [6].

Because wargames usually involve two competing sides, these simulated conflicts provide an opportunity to leverage intelligent agent behavior development, due to the zero-sum relationship between the sides (i.e., one side's victory is the other side's loss), which provides more distinct feedback in the learning process. This study uses these types of wargaming scenarios to investigate whether, in the highest-risk scenarios, AI can provide a forceful backup (i.e., an entity that is able to anticipate future issues and present corrective actions before a catastrophic event occurs) to the ship commanders.

This ability to replace or augment human decision-makers with AI agents presents an opportunity to create tools that could be used in certain scenarios—such as congested area navigation or maritime combat—to enhance the speed and quality of decisions across the fleet.

C. PROBLEM STATEMENT

While implementing different types of AI to achieve optimal performance for wargaming is not new, the wargaming scenarios used thus far have consisted predominately of land forces with a very broad scope. However, it is important to note that land forces and naval forces operate very differently and at different scales. Whereas land forces are typically organized and operate in hierarchical structures, naval forces—on the other hand—tend to operate more independently. Thus, there exists a need to

investigate the use of computer, simulation-based wargaming and the ability to leverage AI agents that can find the optimal outcome of a given naval scenario to aid or augment human decision-making in complex maritime interactions.

In this study, after adapting a combat simulation environment to allow for naval operations, five different AI agents—derived from a combination of rule-based, method-based, and value-based approaches—are developed and evaluated for effectiveness.

THIS PAGE INTENTIONALLY LEFT BLANK

II. REINFORCEMENT LEARNING

Reinforcement learning (RL) is a type of machine learning where an agent serves as a computational entity that interacts with its environment by executing actions and examining the resulting outcomes. The environment encompasses the external setting in which the agent functions, while states signify the distinct conditions of the environment at any given moment. Actions pertain to the choices accessible to the agent, leading to alterations in the environment and subsequent state transitions. Rewards constitute the feedback signals acquired by the agent following an action, providing an indication of the action's desirability [7].

Extending RL is deep reinforcement learning (DRL), which leverages deep neural networks to better manage the high-dimensional information typically characteristic of complex problems. DRL has enabled significant advancements in AI applications, particularly in areas where traditional reinforcement learning methods struggle to perform. DRL algorithms are categorized into four primary types, each with unique characteristics and applications [7]:

Policy-Based Algorithms. Policy-based algorithms, such as REINFORCE and Proximal Policy Optimization (PPO), directly optimize the agent's policy, which dictates the actions the agent should take in a given state by learning the optimal policy without explicitly estimating the value function.

Value-Based Algorithms. Value-based algorithms, such as Q-learning and Deep Q-Networks (DQN), seek to estimate the optimal value function, which represents the expected cumulative reward an agent can obtain from a particular state.

Model-Based Algorithms. Encompassed by method-based algorithms are model-based algorithms. These include algorithms such as Model Predictive Control (MPC) and Monte Carlo tree search (MCTS), which construct an internal model of the environment that is then used to simulate the environment's dynamics and predict future states and rewards.

Combined Methods. Combined methods, such as Actor-Critic and AlphaZero, integrate elements from policy-based, value-based, and model-based approaches to harness their respective strengths and overcome some of their individual limitations.

Over the last few years, there have been several key successes applying DRL to games requiring strategic thinking. Specific examples in the open literature include Go, StarCraft II, and Dota 2 [8]. Most real-time strategy games, such as Dota 2 and StarCraft II involve environments and state spaces the same as those used for wargaming, where long-term goal planning needs to be conducted while making short-term tactical decisions—all in an imperfect-information environment [8]. This research area seems promising as there have been successful cases in demonstrating that RL agents can replicate the desired combat behaviors necessary for wargaming [9]-[12]. In this study, the following specific algorithms are investigated:

DQN. DQN is an extension of Q-Learning and employs deep neural networks to approximate the action-value function [13]. By leveraging the expressive power of neural networks, DQN can handle high-dimensional state spaces and complex problems that traditional Q-Learning cannot efficiently address [7]. Key innovations, such as experience replay and target networks, help stabilize the learning process and improve the performance of the algorithm [7]. DQN has demonstrated remarkable success in various domains, such as game playing, that could inform the necessary approach for wargaming.

MCTS. MCTS is a heuristic search algorithm that combines the principles of tree search with Monte Carlo simulations to make decisions in complex environments [14]. MCTS iteratively builds a search tree by performing random simulations and incrementally refining the tree, based on the results of those simulations. MCTS has been widely applied to various domains, including games and planning, where it has demonstrated significant improvements over traditional search techniques.

AlphaZero. AlphaZero is a reinforcement learning algorithm developed by DeepMind that achieved superhuman performance in the games of chess, shogi, and Go [15]. It represents a significant departure from traditional game-playing algorithms in that it learns solely through self-play, without any reliance on human-generated data or domain-specific knowledge. AlphaZero employs a combination of deep neural networks, MCTS, and RL to iteratively improve its gameplay. Its success demonstrates the potential

of general-purpose learning algorithms to excel in complex and strategic tasks—offering new possibilities for AI research and applications.

THIS PAGE INTENTIONALLY LEFT BLANK

III. SIMULATION ENVIRONMENT

The combat simulation environment used in this study is Atlatl. Atlatl was developed at the Modeling, Virtual Environments, and Simulation (MOVES) Institute, Naval Postgraduate School (NPS) [16]. The simulation is implemented using the Python programming language, while much of the user interface is written in JavaScript and HyperText Markup Language (HTML). Through a web browser interface, the human player can view and interact with the game visually. The communication between the program and the AI is handled via JavaScript Object Notation (JSON) messages. Atlatl allows for the simulation of combat between two factions or sides, typically labeled "blue" for friendly forces, and "red'' for adversary forces. The simulation is turn-based and uses a hexagonal board where each unit takes up one space. The program allows play of human vs. human, human vs. AI, and AI vs. AI. An example scenario of land combat is depicted in Fig. 1.

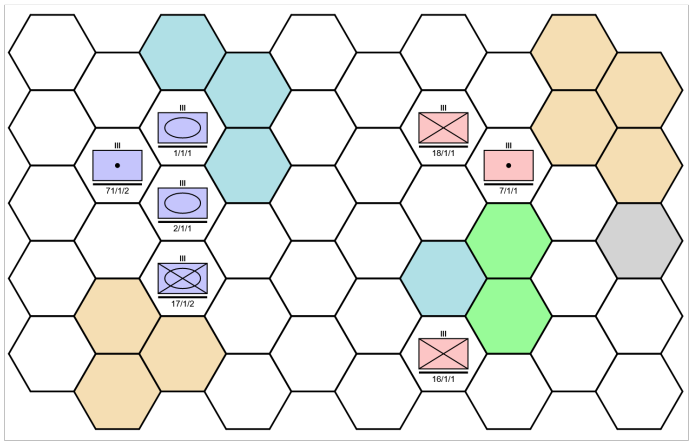

Figure 1. Atlatl Hexagon Gameboard. An Atlatl scenario as it appears in the browser based human interface showing multiple unit and terrain types. Source: [16].

A. TERRAIN REPRESENTATION

While Atlatl was originally designed to model and simulate land units on different types of land terrain, this study required converting the land-specific terrain to naval-

specific terrain, as shown in Figure 2. Each terrain type affects the mobility and defense of the units that occupy them. Table 1 shows the mobility costs for each terrain type in the new naval scenarios created for this study. The mobility cost for the Coastal Defense Cruise Missile (CDCM) is set to infinite to represent fixed coastal batteries that cannot readily move. The mobility cost of the Destroyer is set to 50 in deep water to allow it to move two hexagons per turn and set to 100 in coastal areas to limit its movement to only one hexagon per turn in shallow waters—thus simulating the restriction of movement that ships typically encounter when they approach the shore.

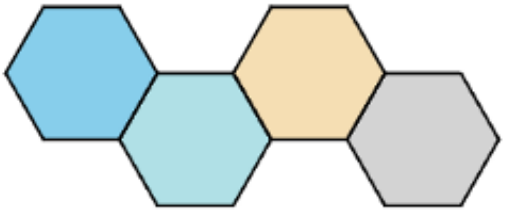

Figure 2. Maritime Terrain Representations. From left to right, hexagons of these colors represent water, coast (or shallow water), land, and urban.

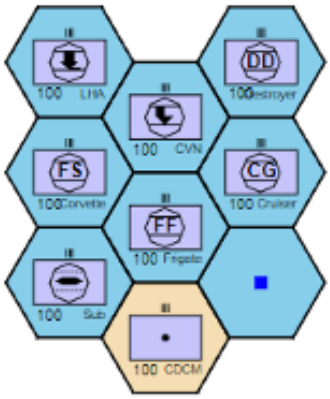

Figure 3. Naval Units Representations

Table 1. Mobility Table for Naval Atlatl

	Water	Coast	Rough	Urban
Destroyer	50	100	Infinite	100
CDCM	Infinite	Infinite	Infinite	Infinite

Table 2. Damage Multiplier for Naval Atlatl

	Water	Coast	Rough	Urban
Destroyer	1	1	1	0.5
CDCM	1	1	1	1

As shown in Table 2, the terrain damage multiplier for urban terrain is set to 0.5, while the rest is set to 1. This means that units occupying an urban hexagon, when attacked, only receive half the damage that the same unit would receive if it were occupying any other hexagon. The urban terrain is designed to incentivize capturing the city outside of just the points gained.

B. UNIT REPRESENTATION

Units are represented with either their U.S. Military Standard 2525 (MIL-STD-2525) or North Atlantic Treaty Organization (NATO) designations, as shown in Fig. 3. For simplicity, only the destroyer class of ships, as well as the CDCM, are used in this study.

Destroyers can attack adversary units within a range of two hexagons, while the CDCM can only attack adversary units within a range of one hexagon. Of note, the CDCM is only used in one scenario in our study, one which seeks to test the agent's ability to recognize this firing range limitation of the CDCM and maneuver around this range to get to the urban port.

C. SCORING MECHANISM

The scoring system within Atlatl is flexible and can be adjusted according to user desires [16]. Game scores are from the perspective of the blue player. For this study, scoring predominantly hinges on two facets: combat effectiveness and control of urban areas (or cities). In combat, each "strength point'' loss inflicted on the red force translates into a positive point for the blue force. The blue force also incurs a penalty of

-1 point for each of its own strength points lost or rendered ineffective in combat. Each unit begins with an initial 100 strength points and is removed from the game (i.e., deemed ineffective) once they drop below 50, with the remaining strength points going to the other team.

In addition to combat outcomes, the control of cities plays a significant role in shaping the player's score. At the start of each scenario, cities are not controlled by any faction. Control shifts only when a unit enters the city, with the controlling faction awarded a score of 24 points divided by the total number of cities being controlled, awarded each phase. Importantly, once a city is occupied, it remains under that faction's control, even if the unit vacates the hexagon, up until an opponent's unit occupies it.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. EXPERIMENTS

The performances of five different agents are compared in these experiments, all fighting against the scripted *Pass-Agg* agent as the red AI. The name Pass-Agg is derived from the terms "passive" and "aggressive." This agent uses decision trees, based on the current state of the game. The core decisions revolve around the collective strength of its force compared to the adversary force. If the Pass-Agg agent has more overall force strength, it adopts an "aggressive" (i.e., offensive) posture and moves to attack the adversary. Of note, when a Pass-Agg agent is within attacking range of multiple enemy units, it uses a uniform distribution to decide which adversary unit to attack. On the other hand, if an agent has less overall force strength than its adversary, it will in turn adopt a "passive" (i.e., defensive) posture and wait for the enemy to attack them. The agents assess their force ratio each time they are called to take any action and, if the force ratio changes due to changes in the environment (e.g., a unit is damaged), then the agent switches to the corresponding posture.

The blue agent behavior models developed and/or trained for this study include: a scripted agent (i.e., Pass-Agg) that is used as the baseline; two MCTS agents (one with a 30K iteration budget and one with a 50K iteration budget); an RL DQN agent; and an AlphaZero-based agent. Each agent is tested across a range of six different exemplar naval scenarios and the results analyzed.

A. NAVAL SCENARIOS

The following six scenarios, as depicted in Fig. 4, represent real-life scenarios that naval units could plausibly encounter while conducting maritime operations. Each has unique aspects that test the agent's ability to overcome challenges or use the environments differently to obtain the optimal score.

Of note, to ascertain the optimal score a human player could receive from each of these scenarios, a researcher familiar with Atlatl—having thorough knowledge of the game and the adversary Pass-Agg behavior model—played each scenario presented here

10 times. Of these 10 games, the best score achieved was selected as the optimal human-level baseline score with which to assess the AI agents.

Scenario 1, Middle Island, is designed to test an agent's ability to maneuver around both sides of the island and attack the adversary, ideally from both directions simultaneously. The coastal waters hinder the destroyer's two-hexagon movement which, in an all-out assault, should stagger the blue unit's approaches and send them into the red force's range in a single file. This lack of formation should result in a loss for the blue forces, even with superior numbers. The optimal move found by the human player is to move all the units outside of the range of one of the enemies, then move two units within range of the other enemy unit, and finally destroy this unit on their next turn. After this first unit is destroyed, the friendly unit that is damaged should not advance with the other two blue units towards the final enemy. The optimal human-level score is found to be 100.

Scenario 2, Mainland and Island, represents a close adversarial force that can quickly get to the island and require the blue forces to position themselves to attack while minimizing losses efficiently. The optimal human strategy here is to leave a unit within range of the city, while the other two units sail around the island. This strategy allows the unit in range of the city to soften up the red unit when it reaches the city and also allows faster recapture to minimize the loss of points from the red force controlling the city. This scenario uses a 10 x 10 map (as opposed to the 7 x 7 map used in all the other scenarios) to provide enough distance between the forces and the city so the red force does not have an insurmountable lead before the blue force can move into their positions. The optimal human-level score is found to be 271.

Scenario 3, Multi-Island, represents small island chains that adversarial forces have occupied. This scenario simulates the restricted movement of units near islands while facing a defensive enemy force. The optimal human strategy for this scenario is to split the blue forces into individual units, circumvent the islands, and simultaneously attack the red force in the middle from three different axes. The optimal human-level score is found to be 100.

Scenario 4, Land Approach, is unique in that it utilizes a CDCM-type unit that cannot move but can fire at units one hexagon away. This scenario also contains a city hexagon that should be quickly controlled by the red forces. The optimal human strategy in this scenario is to quickly attack the city, while remaining outside of the firing range of the CDCM. This strategy forces the blue units to attack the destroyers from a distance and move up the right side of the coast around the CDCM to enter the city, once the adversary ships are defeated. The optimal human-level score is found to be 121.75.

Scenario 5, Chokepoint, is designed to assess if the agent can use the restriction of movement through the channel to defeat the adversary. The optimal human tactic in this scenario is to position the blue forces in a way that allows them to attack each red unit as they transit the channel. The optimal human-level score is found to be 250.

Scenario 6, Tight Channel, tests the agent's ability to efficiently use the space given, while not entering the restricted maneuvering hexagons near the coast. Additionally, this scenario evaluates the agent's ability to spread their forces into a formation that would allow all units to attack at once. The optimal human strategy for this scenario is to use the three-hexagon wide channel, while staying out of the coastal hexagons and forcing a three-on-one engagement against the red units. The optimal human-level score is found to be 100.

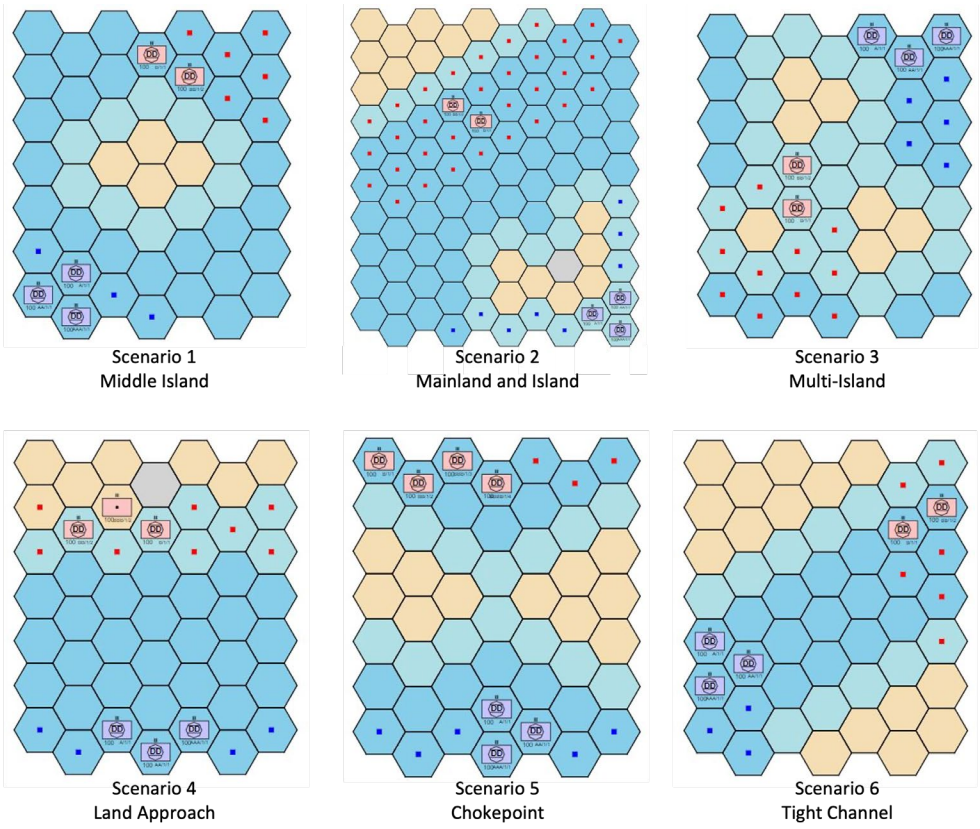

Figure 4. Scenarios Used

B. COMPARISON OF AI AGENTS

In this study, a total of five agents are compared. These agents consist of a combination of rule-based, method-based, and value-based AI algorithms. The first agent is the rule-based agent Pass-Agg described previously. The second and third agents are method-based agents that employ the MCTS algorithm. Two different MCTS agents are used: one with an iteration limit of 30,000 (MCTS 30K), and one with an iteration limit of 50,000 (MCTS 50K). This was designed to analyze the utility of number of simulations versus performance.

The fourth agent used in this study is a value-based RL agent that employs the DQN algorithm from the Stable-Baselines3 (SB3) library [17], using the OpenAI Gym training environment.

The reward function for the DQN agent is derived from the remaining strength of friendly forces, damage to adversary forces, occupation of the urban hexagon, and a terminal bonus value. The terminal bonus value is implemented to help overcome the natural tendency of RL to be risk-averse and encourages the agent to explore without the significant loss of points that can result from getting within range of the adversary. Additionally, the penalty for losing a city in the reward function is only ten percent of the positive bonus of capturing the city. Having immediate rewards following the capture of a city or damage to the enemy enhances the agent's ability to more accurately identify which specific actions are most valuable in each scenario.

The action space of the DQN agent includes 37 discrete actions the unit can take. This action space consists specifically of 18 hexagons the unit can move to, 18 hexagons the unit can fire on, and 1 pass (i.e., the unit chooses to take no action). A total of 5 million training steps each is used to train separate agents for each scenario.

An AlphaZero-based agent is the fifth and final agent implemented. AlphaZero is a combination of a method-based and a value-based approach in that it is an RL version of MCTS. The specific Atlatl agent developed in this study is based on [15] and [18] and uses the MCTS approach and the self-play method to find the best estimated or terminal move. The resulting reward from the action selected is then passed back to the neural network, along with the game winner, to refine a loss-minimizing policy.

THIS PAGE INTENTIONALLY LEFT BLANK

V. RESULTS

The results of the experiments are presented below.

A. PASS-AGG AGENT

The Pass-Agg agent is evaluated by playing each scenario 500 times and capturing the mean scores and their respective standard deviations. The results are depicted in Tab. 3. The breakdown of the scores show that Pass-Agg seems to struggle to understand the environment as a whole. Whereas it is able to correctly assess its combat power ratio and knows where it can move to, it appears to fail to understand these concepts within the context of the given naval scenario. Visual replays show that the Pass-Agg agent would proceed directly towards the red agents, when it was superior in combat power without concern for how it arrived, which often meant it arrived in a vulnerable formation.

B. MCTS AGENT

The MCTS 30K and MCTS 50K agents both generally outperformed the Pass-Agg baseline agent when looking at the overall mean or max scores, as shown in Tab. 4. Additionally, as shown in Fig. 5, human benchmark scores were also compared against the best MCTS 30K and MCTS 50K agents for each scenario. While human play generally outperformed MCTS, the MCTS 50K outperformed human play in Scenarios 2 and 5.

Scenario 2 highlighted the benefits of allowing the MCTS agent more iterations as it was able to discover better, more rare paths that resulted in higher scores—demonstrating the MCTS algorithm's ability to effectively explore and exploit the decision space. The additional simulations appeared to have allowed the agent to build a more thorough decision tree, leading to the achievement of a higher score. Scenario 5 demonstrated how the MCTS agent was able to better manage limited resources and

ultimately take advantage of the geographical chokepoint, which restricted the red forces' movements.

Overall, the MCTS agent showed encouraging results throughout the scenarios presented. The solutions it produced were generally a clear improvement over the Pass-Agg baseline; however, the results varied quite significantly across the six scenarios. Counterintuitively, in Scenarios 1, 3, and 6, the MCTS 30K agent actually outperformed the MCTS 50K agent. Although more systematic investigation needs to be conducted, we postulate that this phenomenon may be due to the "horizon effect" [18], in which the algorithm might fail to foresee the consequences that lie just beyond the depth reach. In these specific scenarios, it may be possible that the short-term tactics were more relevant than the longer-term strategies encountered with the larger iteration budget. Nevertheless, these results provide promising insights into the utility of MCTS algorithms for wargaming decision-making and offer invaluable information for further optimization efforts.

Table 3. Results of Pass-Agg vs. Pass-Agg

	Scores		
	Mean	Std Dev	Max
Scenario 1	-125.000	0.000	-125.000
Scenario 2	-352.000	0.000	-352.000
Scenario 3	-101.375	69.180	-24.000
Scenario 4	-802.875	25.940	-783.500
Scenario 5	115.500	79.200	225.000
Scenario 6	22.500	25.090	62.500
Overall	-207.208	45.389	-107.417

Table 4. Results of MCTS vs. Pass-Agg

	Scores					
	MCTS 30K			MCTS 50K		
	Mean	Std Dev	Max	Mean	Std Dev	Max
Scenario 1	-16.250	36.520	50.000	-3.750	16.770	0.000
Scenario 2	-11.580	146.670	175.000	-354.650	184.340	298.500
Scenario 3	-6.250	26.750	50.000	-11.250	27.480	0.000
Scenario 4	-464.500	90.590	-333.000	-450.130	68.610	-333.000
Scenario 5	168.130	96.440	250.000	214.060	51.130	275.000
Scenario 6	-27.500	47.920	50.000	-29.380	38.320	12.500
Overall	-59.658	-85.015	40.333	-105.850	85.447	42.167

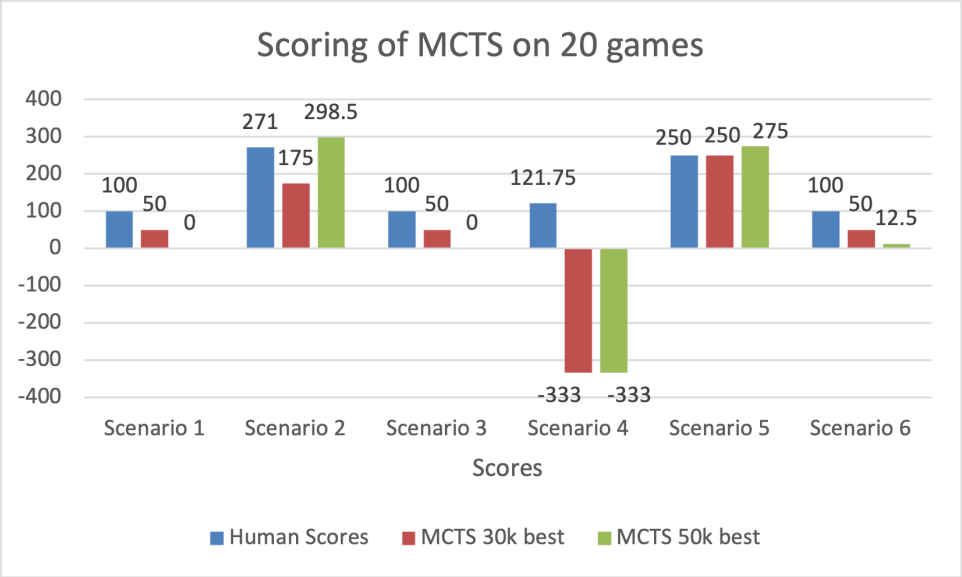

Figure 5. MCTS Results. Comparison of best scores achieved by humans, MCTS with 30K iterations, and MCTS with 50K iterations in 20 games.

C. DEEP Q-NETWORKS AGENT

The results of the DQN agent's performance are shown in Tab. 5 and Fig. 6. Tab. 5 shows the DQN agent's mean and max scores, and Fig. 6 shows the DQN agent's breakdown of wins, losses, and draws in each of the scenarios across 1,000 games. Of

note, the DQN agent achieved all positive mean scores and an overwhelming number of wins over draws and losses.

Overall, the DQN agent exhibited relatively strong performances across all scenarios. Its consistent, superior performances across each and every scenario, in comparison to both those of the MCTS agents and human benchmarks, show potential for being able to perform optimally in complex environments. These strong performances are indicative that an RL-based approach may be capable of handling complexity better than the other approaches examined. Thus, the DQN agent's score maximization and robust performances across various game instances highlight the potential advantages of integrating an RL-approach into naval wargaming.

Table 5. Results of DQN vs. Pass-AGG

	Scores		
	Mean	Std Dev	Max
Scenario 1	12.230	78.190	50.000
Scenario 2	-132.740	138.090	175.000
Scenario 3	25.040	57.850	50.000
Scenario 4	72.320	279.220	559.000
Scenario 5	231.140	82.260	300.000
Scenario 6	23.653	66.180	50.000
Overall	82.854	140.023	197.333

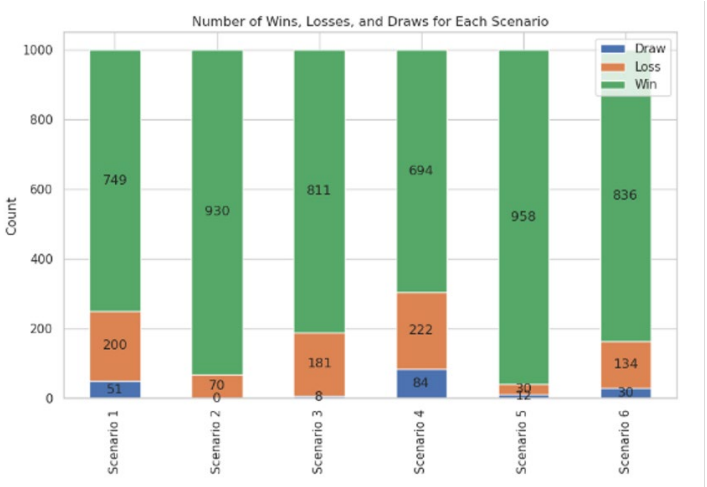

Figure 6. DQN Agent's Wins/Loss/Draws Across 1,000 Games.

D. ALPHAZERO AGENT

As shown in Tab. 6, the AlphaZero agent struggled to optimize its play against the red force. In scenarios where the red force adopted a defensive stance—because it had a negative force-ratio—the AlphaZero agent became very passive and made no significant moves. This resulted in many draws in the game. This might be due to AlphaZero's expectation of action from the opponent, as part of its learned policy strategy. For example, if an opponent is passive and does not move at all, AlphaZero may struggle to learn an effective policy because there needs to be more variety in the opponent's actions to inform AlphaZero's policy updates. This underscores a potential limitation in AlphaZero's self-play approach in that it is optimized for games where both players are actively making moves but struggles when facing a highly passive opponent.

In scenarios with an active opponent, the AlphaZero agent seems to have learned to take offensive actions, such as shooting back once the adversary agents come within range, as in Scenario 5. Nevertheless, these actions were not sufficient to win the game. For instance, in Scenarios 2 and 4, the AlphaZero agent allowed the opponent to capture the city, indicating that it had not learned an effective policy that could prevent the red force from gaining points from occupying the urban hexagon.

Table 6. Results of AlphaZero vs. Pass-Agg

	Scores		
	Mean	Std Dev	Max
Scenario 1	0.000	0.000	0.000
Scenario 2	-834.500	15.188	0.000
Scenario 3	0.000	0.000	0.000
Scenario 4	-996.000	5.590	0.000
Scenario 5	-298.750	5.590	0.000
Scenario 6	0.000	0.000	0.000
Overall	-354.875	6.607	0.000

E. SUMMARY OF RESULTS

Tab. 7 shows the mean scores for each agent across each scenario. Overall, the DQN agent emerged as a consistently high performer across all scenarios. The other agents (Pass-Agg, MCTS 30K, MCTS 50K, and AlphaZero) all struggled to score consistently well, though some agents did manage positive scores in certain scenarios.

Table 7. All Results Summarized

	Mean Scores				
Scen	Pass-Agg	MCTS30K	MCTS50K	DQN	AlphaZero
1	-125.0	-16.3	-3.8	12.2	0.0
2	-352.0	-11.6	-354.7	132.7	-834.5
3	-101.4	-6.3	-11.3	25.0	0.0
4	-802.9	-464.5	-450.1	72.3	-996.0
5	115.5	168.1	214.1	231.1	-298.8
6	22.5	-27.5	-29.4	23.7	0.0
Overall	-207.2	-59.7	-105.9	82.9	-354.9

THIS PAGE INTENTIONALLY LEFT BLANK

VI. CONCLUSION AND FUTURE WORK

This research resulted in a more comprehensive understanding of how AI agents might be used in naval-centric combat simulations. It found benefits and limitations of using different AI approaches that can be influenced by the specific scenarios themselves. Furthermore, this study found that integrating AI into wargaming to drive agent behaviors is a promising approach.

A significant finding is the exceptional performance of the DQN agent over all the other AI agents assessed in this study. The DQN agent demonstrated a promising ability to identify optimal strategies in different situations—outperforming human players in some of the scenarios presented. DQN's robustness and adaptive nature allowed it to generalize and adapt to different operational contexts, thus making it an asset to investigate further for application in the U.S. Navy's decision-making processes.

The MCTS agent, while not as consistently dominant as the DQN agent, showed itself to be effective in certain scenarios. It excelled in complex strategic situations in which exploration and exploitation of the decision space were key—even surpassing human-level performance in one instance. However, it also highlighted the delicate balance between exploration and exploitation, as an overemphasis on either led to a drop in performance. Despite this, the MCTS agent's ability to deliver superior results in certain scenarios suggests it has potential as a strategic tool in naval wargames.

The AlphaZero-based agent, while having demonstrated successes in other domains, showed fewer promising results in the context of this study. Despite its ability to operate on highly complex tasks, its performance was suboptimal compared to the DQN agent. AlphaZero's policy and value networks appeared to not assign a high-enough expected reward to proactive movements toward the enemy units, leading to a more reactive and defensive strategy. Additionally, the asymmetrical nature of the forces in self-play may have posed a significant challenge for an algorithm like AlphaZero. One of the assumptions in AlphaZero's training approach is that games are symmetrical, meaning that the same rules and opportunities apply to both players. This assumption is inherent in games like chess or Go, for which AlphaZero was initially designed. In the scenarios used in this study, however, the forces were not symmetrical, which was a

purposeful design choice to limit the possibility of a draw being the optimal outcome. This suggests a potential need for further fine-tuning or customization of AlphaZero's learning structure to better adapt to asymmetrical environments.

Future work will continue to extend the findings of this study within a naval context. Furthermore, it will inform similar ongoing studies seeking to leverage different behavior models based on specific states of the game.

To conclude, this research serves as a foundational step towards the seamless integration of AI in naval warfare and operations, thus paving the way for more efficient, adaptable, and strategic decision-making tools. Our results indicate that RL-based AI agents, with additional refinement and customization for naval context, could offer substantial advantages for both training simulations and real-world naval operations. Although more work still needs to be done in this domain, the promising outcomes of this study suggest a potential role for AI in providing robust support to decision-makers in the U.S. Navy and beyond. This could ultimately elevate the U.S. Navy into a more secure and effective force, empowered by advanced AI decision-support tools.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] Forward Defense Experts, "Today's wars are fought in the 'gray zone.' Here's everything you need to know about it.," Atlantic Council, Feb. 23, 2022. https://www.atlanticcouncil.org/blogs/new-atlanticist/todays- wars-are-fought-in-the-gray-zone-heres-everything-you-need-to-know-about-it/

[2] National Transportation Safety Board, "Collision between U.S. Navy Destroyer Fitzgerald and Philippine-flag container ship ACX Crystal, Sagami Nada Bay, off Izu Peninsula, Honshu Island, Japan, July 17, 2017," Washington, DC, USA, Rep. NTSB/MAR-20/02 PB2020-10100, 2017 [Online]. Available: https://maritimecyprus.com/wp-content/uploads/2020/09/Collision-Fitzgerald-ACX-Crystal c.pdf

[3] National Transportation Safety Board, "Collision between U.S. Navy Destroyer John S McCain and Tanker Alnic MC," Washington, DC, USA, Rep. NTSB/MAR-19/01 PB2019-100970 [Online]. Available: https://s3.documentcloud.org/documents/6243999/MAR1901.pdf

[4] P. Dunn, Sea battle games. Hemel Hempstead: Model and Allied Publications, 1970

[5] Department of the Navy, "U.S. Naval War College." Accessed Apr. 17, 2023 [Online]. Available: https://usnwc.edu/

[6] P. K. Davis and P. Bracken, "Artificial Intelligence for wargaming and modeling," The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology, 2022. doi:10.1177/15485129211073126

[7] L. Graesser and W. L. Keng, Foundations of Deep Reinforcement Learning: Theory and Practice in Python, 1st ed. Addison-Wesley Professional, 2019.

[8] Goodman, J., Risi, S., and Lucas, S., "AI and Wargaming," Tech. Rep. arXiv:2009.08922, arXiv (Sept. 2020). arXiv:2009.08922 [cs] type: article.

[9] Rood, P. R., "Scaling Reinforcement Learning Through Feudal Muti-Agent Hierarchy," PhD thesis, Naval Postgraduate School, Monterey, CA (Sept. 2022).

[10] J. Boron and C. Darken, "Developing combat behavior through reinforcement learning in wargames and simulations," 2020 IEEE Conf. on Games, Osaka, Japan, 2020, pp. 728–731, doi: 10.1109/CoG47356.2020.9231609.

[11] C. T. Cannon and S. Goericke, "Using convolution neural networks to develop robust combat behaviors through reinforcement learning," M.S. thesis, Naval Postgraduate School, Monterey, CA, USA, 2020.

[12] J. Allen, “Enlisting AI in course of action analysis as applied to naval freedom of navigation operations,” M.S. thesis, Naval Postgraduate School, Monterey, CA, USA, 2022.

[13] A. G´eron, Hands-On Machine Learning with Scikit-Learn and Tensor-flow: Concepts, Tools, and Techniques to Build Intelligent Systems. Sebastopol, CA: O’Reilly Media, 2017.

[14] G. Chaslot, “Monte-Carlo tree search,” PH.D. dissertation, Universiteit Maastricht, Maastricht, Netherlands, 2010.

[15] D. Silver et al., “Mastering Chess and Shogi by self-play with a general reinforcement learning algorithm.” 2017 [Online]. Available: ArXiv. /abs/1712.01815

[16] C. Darken, “Atlatl.” Monterey, CA, USA [Online]. Available: https://gitlab.nps.edu/cjdarken/atlatl

[17] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, “Stable-Baselines3: Reliable Reinforcement Learning Implementations,” Journal of Machine Learning Research, vol. 22, no. 268, pp. 1–8, 2021.

[18] S. Thakoor, S. Nair, and M. Jhunjhunwala, “Learning to play Othello without human knowledge.” Stanford University, Final Project Report, 2016.

[19] S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 3rd ed. Prentice Hall, 2010.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Research Sponsored Programs Office, Code 41
   Naval Postgraduate School
   Monterey, CA 93943

4. Anthony Tai
   Naval Surface Warfare Center (NSWC), Division Crane
   Crane, IN