COORDINATING TEAM TACTICS FOR SWARM-VS.-SWARM ADVERSARIAL GAMES

A Dissertation
Presented to
The Academic Faculty

By

Laura G. Strickland

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in Robotics
School of Interactive Computing

Georgia Institute of Technology

August 2022

COORDINATING TEAM TACTICS FOR SWARM-VS.-SWARM ADVERSARIAL GAMES

Approved by:

Prof. Matthew Gombolay
School of Interactive Computing, College of Computing
Georgia Institute of Technology

Dr. Jeremy Reed
Sensors and Electromagnetic Applications Laboratory (SEAL), Intelligence, Surveillance, and Reconnaissance Division
Georgia Tech Research Institute

Dr. Charles Pippin
Aerospace, Transportation, and Advanced Sciences (ATAS) Laboratory, Robotics and Autonomous Systems Division
Georgia Tech Research Institute

Prof. Frank Dellaert
School of Interactive Computing, College of Computing
Georgia Institute of Technology

Prof. Seth Hutchinson
School of Interactive Computing, College of Computing
Georgia Institute of Technology

Date Approved: July 2, 2022

Kein Operationsplan reicht mit einiger Sicherheit
über das erste Zusammentreffen mit der feindlichen Hauptmacht hinaus.

(No plan of operation is sure to withstand
initial contact with the adversary's primary force.)

– Helmuth Karl Bernhard Graf von Moltke, *Über Strategie* [1]

ACKNOWLEDGEMENTS

There are many individuals who have been great supporters to me throughout the long journey of my Ph.D. To provide context, I started my Ph.D. at Georgia Tech in 2013, in a new lab within the George W. Woodruff School of Mechanical Engineering. This lab began with a focus on the dynamics of a variety of robots and robot teams, but as the lab grew, its focus shifted towards aeronautical-engineering-related projects. When my non-flying robot dissertation project lost funding, I, a mechanical engineer with no background in the dynamics of flight, was no longer a good fit. Thus, I accepted a Graduate Research Assistantship at Georgia Tech Research Institute (GTRI), where, ironically, I began working on projects that involved swarms of aircraft (albeit with a strong focus on autonomy rather than flight dynamics). These projects eventually became the basis for much of the work in my dissertation. I sincerely appreciate all that I learned in that first lab and those in that lab who taught me so much. My time at Georgia Tech has been a long and winding road, but I would not change a minute of the journey—I have learned and grown so much and found so many amazing friends and mentors along the way, and my experiences make me who I am today.

First and foremost, I would like to thank Dr. Matthew Gombolay, my advisor. Matthew, despite me being an "old" student (my words) when I interviewed you/interviewed with you in the process of finding a new advisor in May of 2019, you graciously welcomed me into your lab, helped me define and refine the problem upon which I was endeavoring to build my dissertation, and mentored me through many challenging situations, both professionally and personally. You were—and continue to be—so patient when we discover that my professional and academic background did not completely prepare me for some aspect of my work. Your challenging and encouraging me to expand my areas of competence to fill those holes in my knowledge, as well as your patient, careful explanations when I ask questions that reveal that my efforts to increase my understanding have instead confused it, are a vivid illustration in my mind of the type of mentor I wish to have and wish to be to others. Your support and advice, and your honesty and candidness in sharing with

those of us in your lab (with appropriate anonymity) how you face your own professional challenges, have greatly influenced who I have become today. Matthew, it meant so much to me every time you defended me in front of the huge body of faculty present at the Interactive Computing Ph.D. student review meetings. You invested so much time into me, working with me and helping me navigate the requirements of both the Robotics Ph.D. program and Interactive Computing and pushing me to challenge myself in taking steps towards finishing my degree. Your agility and resourcefulness in helping all of us in Cognitive Optimization and RElational Robotics Laboratory (CORE) lab adjust and shift to being able to work from home when the COVID-19 pandemic hit was a lighthouse to all of us at that confusing time, and seeing you roll with the punches as you led our lab has been a continual source of inspiration. Your efforts to individually advise your students in the manner most fit for them rather than in one unmoving management style is a true testament to how much you care about the students you mentor, and it shows in the efforts and enthusiasm all of us in CORE lab pour into our work. Thanks to your mentoring and feedback, I am a far stronger person today than I was when I first joined your lab, and I wholeheartedly thank you.

I first met Dr. Charles Pippin at the dissertation defense of Dr. Michael "Misha" Novitzky, but did not know that, just a few months later, Charles would have a profound influence on both my dissertation and my professional direction. When I first interviewed at GTRI in 2016, I did so knowing that I would need to leave behind the bespoke-hardware-dependent dissertation topic that could no longer be funded in my first lab due to that first lab's shift in focus. At GTRI, I was welcomed with open arms. Charles became a de-facto dissertation advisor and helped me find a new direction for my research, and in an area that I have become quite passionate about. His constant reminders to extract myself from the weeds of the problem I was working on and re-examine the big picture still echo helpfully in my head today, and his faith in myself and all of my GTRI colleagues encouraged all of us. Charles' gracious responses when I reported various mistakes of even my own creation—appreciation of me finding and reporting the mistake, and then focusing on how to fix the issue and learn from it rather than assigning any blame—have made me much more willing to try new things research-wise rather than sticking to a known-working-

but-less-innovative status quo. His leadership encouraged this gracious, ever-learning collaborative and constructive attitude to be a part of the culture of Robotics and Autonomous Systems Division (RASD), Aerospace, Transportation and Advanced Systems Laboratory (ATAS), and GTRI as a whole, and the world is better for it. When I found myself spinning my wheels, you gave me your time to help me decide which faculty members at Georgia Tech to approach about becoming my new advisor, and supported my new bimodal work location schedule. You even showed support when I needed to leave GTRI in 2020 to both complete my TA requirement and focus my research efforts more completely on my dissertation, which is a continual source of encouragement for me.

I most sincerely thank the other members of my committee as well: Dr. Seth Hutchinson, who believed in me and the work I had done towards my dissertation, who was far more invested in helping me in my search for a new advisor in 2019 than I ever expected, and who encouraged me to find ways to motivate the story I had to tell through both my proposal and my defense and dissertation to best showcase my work. Dr. Frank Dellaert, who also encouraged me in structuring the narrative I was weaving for my proposal and dissertation as well as opened my eyes to the value of unit tests and unit-test-like mechanisms for verifying the functionality of small chunks of code to foster the correctness and understanding of the whole codebase. And Dr. Jeremy Reed, who, despite joining my committee towards the end of my degree, provided a great deal of encouragement as well as a number of practical, helpful suggestions during my preparations for my dissertation writing and my final defense presentation. You all took significant amounts of time out of your busy lives to invest in me, and I appreciate that so much!

Next, a sincere thank-you to my colleagues and friends in CORE lab—the list is a bit too long to name all of you personally, but you all have been great friends and encouragers to me along the way. I'd like to specifically thank Letian "Zac" Chen for his cheerful willingness to answer questions, all the times he helped me debug my Reinforcement Learning (RL) setup (and teaching me so much about the practical-implementation side of deep RL in the process!), for encouraging me throughout the aforementioned debugging sessions, and for taking—and even enforcing—my reservations for some of the lab's High-Performance Computing (HPC) resources at all hours of

day and night. I also want to specifically thank Erin Hedlund-Botti for being a sympathetic ear during some of the more challenging parts of my Ph.D. journey and for her thoughtfulness and great suggestions in helping me iterate on versions of my defense.

I also would like to thank my colleagues at GTRI. When I started at GTRI, I was introduced to SCRIMMAGE, the simulator in which I would do my dissertation experiments, by being asked to get it to compile and run. Throughout the process, and through the rest of my student time at GTRI, I met a number of research engineers and scientists who I am proud to call both colleagues and friends. Dr. Eric Squires, thank you for taking so much time to help me with a variety of problems, for being a prime example of being considerate of one's colleagues' time, and for your encouragement and advice along the way. It was so inspiring seeing you work hard towards your Ph.D. while working full-time and supporting your family, all while retaining your optimistic outlook and your sense of humor along the way. And I'm still using a fork of your linux-config repository for my own Linux configuration setup—thank you for noticing my fledgling efforts to utilize vim and showing me so many tools to make it work more effectively with my workflow. Dr. Kevin DeMarco, thank you so much for your advice and encouragement—your perspective as a new Ph.D. Research Scientist whose student years were recent memory was illuminating and encouraging. You invested a lot of time into me, helping me debug while teaching me all sorts of things about CMake, multi-agent simulation, and priorities as a researcher constantly fighting the battle between "perfection" and "good enough." Eric and Kevin, I'm not only a better C++ programmer because of you, but also a better researcher and better collaborator. There are many additional people at GTRI who I would like to thank—too many to include here—but Michael Day, thank you for your excellent management on several projects and for your incredible balance between productive practicality and dry, good-natured humor helping us all not take minor setbacks too seriously. Michael Matthews, you helped me navigate a number of unfamiliar and confusing systems while I was at GTRI, both when I was first getting started, to the times when I saw a ceiling tile leaking and had no idea who to report it to, as well as the framework used to get agent autonomy code from simulation to flying on actual aircraft; you introduced a number of students to GTRI's Unmanned

Aerial Vehicle (UAV) frameworks, and we all appreciate the time you spent showing us the big picture as well as helping us navigate the details within. Lastly, I'd like to thank Rusty Roberts and Dr. Don Davis, who, despite their impressive leadership positions and influence, took the time to get to know, invest in, and encourage even us student workers at GTRI. Rusty, I'd like to offer extra thanks to you for taking the time to attend not only my proposal presentation, but also staying in the room when I was asked to leave for my committee to deliberate. I did not hear what you said then, but I have a feeling it helped to shape the remaining semesters of my degree—thank you.

I also sincerely appreciate a number of student groups who helped me meet friends and colleagues, especially RoboGrads, RoboWomen, and the ever-changing group of people attending CPL Drinks for dinner and conversation. Thank you for introducing me to a number of good friends, giving me places to vent, and in the cases of RoboWomen and RoboGrads, giving me a chance to help the robotics-researching grad students at Georgia Tech make the Robotics Ph.D. program even better in some small way. I'd like to thank my good friend Professor Tesca Fitzgerald for her encouragement, optimism, sense of humor, and willingness to listen to my venting and frustrations. Tesca, whatever part of a room isn't lit up by your intelligence is lit up by your cheerful demeanor, and the world is a better place for it—you're an incredible inspiration. I know you will be an excellent mentor, advisor, and friend to your students, and I am so blessed to call you my friend! Dr. Vivian Chu, thank you for being such a good friend, hosting so many wonderfully-memorable gatherings of friends, and letting me vent and providing encouragement. Thank you for providing your perspective in a number of different situations; especially when I was a brand-new Ph.D. student, your willingness to share your wisdom from your experience during your M.S. was so helpful. It's amazing watching you run a successful startup, and I am blessed to be your friend. Dr. Andrew Price, thank you for being a great friend, all the advice you have offered in difficult situations, and for teaching me a great many things. During some of my first group projects at Georgia Tech, you taught me how to use git (as you patiently untangled some git-related mess I had created), and you taught me just enough about vim to get me started using it. (And thank you for appreciating the humor in the situation when you learned that I had adopted vim as my primary

text editor instead of as a stopgap command line tool for quick edits!) Thank you for encouraging me to take some chances, both academically as well as personally, and for showing how to not only accept points of view that differ from your own, but also how to be curious about them. Dr. Brian Goldfain, you were one of the first students I met at Georgia Tech, and you were so welcoming; you showed me the project you were continuing from your M.S. that eventually became your dissertation work, and really made me feel like I was a friend, not an outsider. You getting me connected to the CPL Drinks group and to RoboGrads led me to meeting many of my current friends. Your gift for hospitality and your willingness to try new things and encourage others to do so has made for many wonderful memories, and your encouragement through challenges throughout my degree program was more impactful than you know. To the rest of the members of the Ramen Night Crew and/or Game Night Crew who I have not already mentioned here—especially Dr. Shan Tie, Dr. Paul Drews, Dr. Amrita Gupta, Shelley Bagchi—you all are wonderful friends! Thank you for the many fun memories, and here's to many more fun times together in the future. To Dr. Michael "Misha" Novitzky, thank you for all of the pep talks you gave me during my first few years at Georgia Tech—your encouragement has buoyed me through a lot of impostor's syndrome.

I'd also like to thank my parents, Dr. Kim Strickland and Glenda Strickland, who have stood by me and encouraged me every step of the way through my education journey, from my first days of preschool onwards. Mom and Dad, you fostering my interest in computers when I was little and encouraging me during all phases of my studies were instrumental in getting me to where I am today. Your willingness to love and support me regardless of what decisions I made about my future—even through the complicated logistics of studying abroad or deciding to pursue a career in research and work towards my Ph.D.—has been such a blessing. Mom and Dad, I can't tell you how much I love you!

Lastly, but above all, I thank God for all He has done in my life, and for not only giving me grace through Jesus' death on the cross and subsequent resurrection, but also for all of the opportunities He has placed in my life and all of the wonderful people He has sent to help me walk the road I'm on. He is my strength in all areas of my life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

While swarms of UAVs have received much attention in the last few years, adversarial swarms (i.e., competitive, swarm-vs.-swarm games) have been less well studied. In this dissertation, I investigate the factors influential in team-vs.-team UAV aerial combat scenarios, elucidating the impacts of force concentration and opponent spread in the engagement space. Specifically, this dissertation makes the following contributions:

1. **Tactical Analysis:** Identifies conditions under which either explicitly-coordinating tactics or decentralized, greedy tactics are superior in engagements as small as 2-vs.-2 and as large as 10-vs.-10, and examines how these patterns change with the quality of the teams' weapons;

2. **Coordinating Tactics:** Introduces and demonstrates a deep-reinforcement-learning framework that equips agents to learn to use their own and their teammates' situational context to decide which pre-scripted tactics to employ in what situations, and which teammates, if any, to coordinate with throughout the engagement; the efficacy of agents using the neural network trained within this framework outperform baseline tactics in engagements against teams of agents employing baseline tactics in N-vs.-N engagements for N as small as two and as large as 64; and

3. **Bio-Inspired Coordination:** Discovers through Monte-Carlo agent-based simulations the importance of prioritizing the team's force concentration against the most threatening opponent agents, but also of preserving some resources by deploying a smaller defense force and defending against lower-penalty threats in addition to high-priority threats to maximize the remaining fuel within the defending team's fuel reservoir.

CHAPTER 1
INTRODUCTION

1.1 Motivation

Unmanned Aerial Vehicles (UAVs) are growing more popular and more available in the consumer market. With this consumer interest, there is likewise growing attention in the defense and research communities. A number of groups around the world are displaying ever-growing interest in their abilities to construct, maintain, and deploy large swarms of UAVs [2, 3]. With this international increase in interest, and with financial and logistical obstacles to obtaining and deploying UAVs decreasing, there is increasing concern of the threat of UAVs operated by nefarious or negligent actors [4–8]. Indeed, as recently as May 2021, Israel employed a surface-to-air missile to shoot down a single suspected-hostile UAV [9]; swarms of hostile UAVs are likely to become even more serious threats in the coming years. Programs such as DARPA's AlphaDogfight [10] and Air Combat Evolution (ACE) [11] demonstrate that the United States is working to prepare for such an eventuality. AlphaDogfight explored 1-vs.-1 aerial combat between UAVs, and ACE seeks to build upon AlphaDogfight's foundation in both practical application and scaling up to application in larger engagements. As I demonstrate in Chapter 2, however, the perspectives and approaches with respect to small (1-vs.-1, 2-vs.-2, etc.) engagements and with respect to larger engagements in the literature are very different, and the literature that explores the effectiveness of tactical behaviors as engagement sizes scale from small to large is scarce. To my knowledge, as of the time of writing of this dissertation, no literature apart from my own work (Chapter 4 and its predecessor [12]) addresses leveraging Reinforcement Learning (RL) in the selection of aerial combat tactics in small and large engagements. To work towards filling this gap in the literature and human understanding, this dissertation explores the scaling of engagements from small to large, examining what factorrs influence engagement outcomes of various sizes. I present a bio-inspired close-in site defense

scheme utilizing multirotors, and analyze within-visual-range (WVR) fixed-wing aerial combat tactics for the interception of evaders further away from their target. I also introduce and demonstrate a reinforcement learning architecture that equips a team of homogeneous fixed-wing UAVs to switch between aerial combat tactical doctrines mid-engagement for greater combat effectiveness in engagements of a large variety of sizes than the individual tactical doctrines alone. In these contexts, I elucidate the key components of leveraging tactical advantage and distribution of force concentration during an engagement to disable the opponent team and preserve the protagonist team, and discuss how the protagonist team's performance changes as team sizes grow.

1.2 Research Overview

In this dissertation, I present studies of tactics and coordination factors in engagements between UAVs leveraging human-pilot-inspired aerial combat tactics. I then introduce and demonstrate an algorithm for training agents to use their own and their teammates' situational context to make appropriate tactical and teamworking decisions to achieve tactical advantage within aerial combat engagements. I then conclude by motivating these observations on force concentration and force distribution in swarm-vs.-swarm engagements with a demonstration of how they apply in a different scenario—in a defense scenario between heterogeneous teams of multirotors. The fixed-wing and multirotor scenarios may be seen as two components of a comprehensive site defense scheme, where fixed-wing UAVs are deployed to intercept as many opponent agents as possible before the opponents can reach a targeted location, and multirotors are employed for close-in last-mile defense of the targeted location to defeat any adversaries that evaded the fixed-wing intercept.

1.2.1 Summary of Contributions

The contributions presented in this work explore team-tactical approaches to swarm defense scenarios and UAV aerial combat. This body of work is towards the overall aim of comprehensive site defense, with fixed-wing UAVs intercepting adversary agents far from the defended location and multirotors executing close-in last-mile defense. Firstly, I investigate greedy and explicitly-

coordinated tactical maneuvers inspired by tactical doctrine employed by human fighter pilots. I demonstrate the dependence of tactics that require precise maneuvering with a teammate to aim on the quality of their weapons and the availability of non-hostile space in the engagement, as well as the resilience of the decentralized, greedy team's abilities to differences in weapon quality and its strength in dense regions of engagements. Next, I present and demonstrate an approach to train a neural network that performs distributed, agent-centric evaluation of individual and teaming-based tactics to determine the most tactically-effective scripted maneuver an agent can use at the current moment. In doing so, these trained agents determine which, if any, teammate they should partner with to execute the maneuver and how to maneuver when none of the pre-scripted tactics are situationally appropriate. I then examine a bio-inspired defense scenario and draw conclusions regarding the corce concentration implications of how differences between the abilities of agents on a heterogeneous team to identify and attack a heterogeneous team of enemies impact the scheduling of different defender roles and fuel required for the defense scheme.

1.2.2 Detailed Overview

Tactical Analysis: What Affects Swarm-vs.-Swarm Engagements – Chapter 3 explores the effects of opponent spread in engagements between teams of fixed-wing aircraft, where the aircraft must maneuver precisely to successfully aim at their opponents [13]. These maneuvers are components of aerial combat tactics inspired by tactics employed by human fighter pilots. My interest in engagements that leverage these tactics is in how well the tactics' effectiveness and ability to create and exploit force concentration advantages changes as the size of the engagement changes, and whether weapon quality has any impact on these relationships. As shown in the literature in Chapter 2, most Basic Fighter Maneuvers (BFMs) and most of the pairwise tactics human fighter pilots have employed historically are focused upon achieving a position relative to the opponent that prevents the opponent from being able to fire at the protagonist team member(s), and that allows one or more of the protagonist team members to have an unhindered firing opportunity against the antagonist [14]. These tactics (with the exception of BFM, which are for single aircraft) primarily

depend on two or more aircraft on the protagonist team maneuvering in a coordinated fashion. Some tactics are effective with no explicit coordination or communication, while other explicitly-coordinating tactics may perform best with explicit communication, e.g. to clarify which opponent a pair of aircraft are targeting, or to establish when to begin a specific phase of a timing-dependent multi-stage coordinated maneuver. Chapter 3 discusses the characteristics of two different tactical behaviors—a fully decentralized, greedy tactic; and a tactical behavior that leverages explicit coordination to execute pairwise-coordinated maneuvers that target specific opponents. Of particular interest are the conditions under which these two tactical behaviors are effective, as well as how varying the ability of an aircraft's fire to attrit an opponent changes the effectiveness of these tactics.[1] The maneuvers used by the explicitly-coordinating teams focus on achieving firing opportunities against one opponent at a time, highlighting again the importance of local force concentration superiority within the overall engagement. In engagements of 2-vs.-1, 2-vs.-2, and 4-vs.-4, when against a team employing non-coordinating greedy aiming tactics, the coordinated teams generally attrit more opponents than their team loses. In larger engagements, such as 10-vs.-10, the decentralized, greedy tactical behavior is more effective under some weapon quality conditions than its explicitly-coordinating counterpart. In those scenarios, the denser conditions created by the larger teams in the same engagement zone greatly reduce the non-hostile space and available maneuvering time the coordinated teams need to perform the maneuvers upon which their aiming procedures rely, and thus, with a low-accuracy weapon, the explicitly-coordinating teams perform more poorly than when they are equipped with accurate weapons. This trend is especially apparent when comparing the explicitly-coordinating team's performance in dense engagements and with poor weapon quality to that of their decentralized counterparts, who have no dependence upon complex maneuvers to aim. These experiments show that the teams using the explicitly-coordinating maneuvers excel in low-density engagements, but the coordinated behavior's dependence on these aiming maneuvers is a liability in larger, denser engagements, where the

[1] "Attrit" means the act of causing a fighting force to experience attrition, generally in the sense of a specific agent being removed from the active members of its fighting force; e.g. "agent A attrits agent B" means that agent A fires at agent B, the shot connects, and agent B is removed from the engagement.

number of aircraft in the engagement makes isolating opponents to obtain the non-hostile space and maneuvering time needed to employ these explicitly-coordinating maneuvers to pick off individual opponents difficult. The decentralized, greedy teams, in contrast, are less effective in small engagements against teams using their same tactics or using the explicitly-coordinating tactics, but their tactics' invariance with respect to the proximity of agents other than their target allows them to more easily overwhelm weapon-disadvantaged opponents in larger engagements with denser conditions. These two very different tactical approaches to aerial combat engagements are both effective, but in different scenarios. Thus, in this dissertation, I equip agents to switch tactics during an engagement based on what is happening around themselves and their teammates.

Learning To Leverage Tactics – Chapter 4 demonstrates a team tactic coordination algorithm that directs the behaviors employed by the members of the protagonist team so that team members employ either greedy or explicitly-coordinating tactics when appropriate, and when neither option is a tactically-favorable choice for a particular agent, makes intelligent maneuvering choices directly by dictating the agent's yaw rate. My hypothesis in Chapter 4 is that these agents, equipped with the ability to learn to switch tactics based on their surroundings, will perform better (attrit more opponents and keep more own-team agents alive) than teams employing the hand-crafted tactics of Chapter 3. The work introduced above (Chapter 3) emphasizes the importance of achieving favorable force concentration against groups of opponent team members, as well as the importance of leveraging the tactical behavior best suited to the current situation. In the literature (see Chapter 2), the primary emphases of research regarding fixed-wing UAVs in aerial combat is split into works that focus primarily on the actions of individual or small groups of agents and how those tactical action choices have an impact on the overall engagement, and works that focus more on team-wide strategies for engagements [14]. Chapter 4's contribution enables a team to make decisions about which tactic each team member should employ with respect to the local situational contexts of the other members of the team, and when to coordinate explicitly with one of those teammates. Agents trained using this algorithm are capable of outperforming teams that employ one of the baseline hand-crafted tactics discussed in Chapter 3, despite only being trained against teams employing

one of the baseline tactics. The learner agents are trained in relatively small engagements, yet perform increasingly well in N-vs.-N engagements even when N increases beyond the engagement size in which the team was trained.

Bio-Inspired Coordination: Force Allocation Prioritization on a Heterogeneous Team – Approaching the close-in defense problem, Chapter 5 demonstrates the effectiveness of leveraging force concentration in a different setting than fixed-wing aerial combat, instead investigating its effectiveness in a defense scenario between two heterogeneous teams of multirotors. Chapter 5 explores the hypotheses that the ability for the guarding force to mainly encounter enemies one at a time and to engage the most threatening opponents multiple times, combine to ultimately create advantageous conditions for the protagonist team [15]. The bio-inspired simulation detailed in Chapter 5 is a defense scenario in which a heterogeneous swarm of multirotors counters a heterogeneous force of attackers that approach the defended location one by one, and examine the some of the mechanics of why this bio-inspired guarding structure is effective. In these experiments, two types of defensive agents guard a High-Value Target (HVT), which is their team's energy source, and two types of adversary agents attempt to break through the guard ranks to reach the HVT to deduct a type-specific amount of energy from it. To initialize each guard, a role-specific amount of energy is deducted from the HVT energy store, with one guard role deducting more energy than the other. The more expensive guards, which specialize in identifying the more-costly attackers, have the benefit of being able to re-engage an escaped opponent agent; this capability gives them greater fighting strength against their opponents[2] providing the defenders countering the high-threat attackers with local numerical superiority over the attackers that approach the defended location one by one. The less-expensive guards, which specialize in identifying the cheaper attackers, cannot re-engage escaped attackers, but the low penalty of the opponents against which they guard reduces the impact of this limitation. The team guarding resources from these attackers generally maximize

[2]Lanchester's Square Law [16–18] illustrates how the fighting strength of a force employing aimable weaponry against its opponents is directly proportional to the difference between the *squares* of the force sizes. As is shown in Section 5.5, concentration of force and the ability for the guards countering the high-threat attackers to re-engage escaped adversaries are why the expensive guards are as effective as they are against the high-penalty attackers. Employing multiple of these expensive guards spreads the approaching attackers out across the guard force,

the amount of resource remaining after all attackers have approached the defended location with fewer guards than there are attackers in a given simulation, indicating that expecting and accepting some loss of resources is more economical in terms of preserving the defended resource than using the defended resource to create an impenetrably-large guard force.

1.3 Objectives and Contributions

In this work, I investigate the problem of developing coordination strategies for adversarial swarm-vs.-swarm scenarios. Informed by my investigations into aerial combat tactics, I develop a novel computational approach that, by leveraging the advantages of expert-defined tactics and the flexibility of deep reinforcement learning, outperforms hand-crafted baseline tactics. Specifically, this dissertation makes the following contributions:

1. **Tactical Analysis:** Identifies the conditions under which either explicitly-coordinating tactics or decentralized, greedy tactics are tactically superior in engagements as small as 2-vs.-2 and as large as 64-vs.-64, and examines how these patterns change as the quality of a team's weapon changes [13];

2. **Coordinating Tactics:** Introduces and demonstrates a deep-RL framework that equips agents to learn to use their own and their teammates' situational context to make decisions about which pre-scripted tactics to employ in what situations, and which teammates, if any, to coordinate with throughout the engagement; the efficacy of agents using the neural network trained within this framework outperform the baseline tactics introduced in Chapter 3 in engagements against teams of agents employing such tactics in N-vs.-N engagements for N as small as two and as large as 64; and

3. **Bio-Inspired Coordination** Discovers through Monte-Carlo agent-based simulations the importance of prioritizing the team's force concentration against the most threatening opponent agents, but also preserving some resources by deploying a smaller defense force and

defending against lower-penalty threats in addition to high-priority threats to maximize the remaining fuel within the defending team's fuel reservoir [15].

1.4 Outline

I first examine related literature in Chapter 2, then explore the aforementioned implicitly- and explicitly-coordinating team tactics, the situations in which each is effective, and how weapon quality affects their performance in Chapter 3. I then introduce and demonstrate the training and testing of an RL scheme for switching tactics based on pairs of an agent's situational context with the situational contexts of its teammates in Chapter 4. I demonstrate the implications of advantageous force concentration in a multirotor bio-inspired defense scenario in Chapter 5. Finally, I address the limitations of these approaches and experiments and potential future work in Chapter 6, and conclude in Chapter 7.

CHAPTER 2
RELATED WORK

2.1 UAV Tactics

Aerial combat tactics for fixed-wing aircraft[1] have been discussed extensively in the literature, with much of aerial engagement research focused on small engagements (1-vs.-1, 2-vs.-1, or 2-vs.-2). Works on small engagements typically examine how the maneuvering of the individual agents involved in the engagement impacts the engagement outcome. Works pertaining to larger engagements primarily focus on weapon-target assignment algorithms, cognitive architectures for human-pilot-like decisionmaking, and high-level engagement analysis. I review these works and elucidate the need for study in what properties of a tactical behavior make it capable of scaling well between small and large engagements, and what role within-team coordination plays in engagements of all sizes.

2.1.1 Small Engagements

The selection of literature pertaining to small aerial combat engagements is vast, but all center around the crucial BFM tenet of achieving a position of advantage against one's opponent where one can fire at the opponent without risk of the opponent being able to return fire, and preventing the opponent from achieving this same advantage against one's own aircraft [14].

Popular methods for achieving this desired advantageous positioning include utilizing expert systems [19–24], control laws for pursuit and/or evasion [25–30], differential-game-theoretical

[1]This dissertation primarily focuses on fixed-wing aircraft specifically due to their restricted motion. Rotorcraft can change their yaw independent of their flight path, which, assuming a boresight-fixed weapon, allows them to aim their fire at a moving opponent without difficulty. Conversely, fixed-wing aircraft using a boresight-fixed weapon must maneuver carefully to aim their fire, and may do so more easily with assistance from teammates moving in a way that entices opponent aircraft into predictable, tactically-disadvantageous locations. The discussions on coordinated maneuvers in this document are centered on this assistance in achieving favorable positioning against opponents, hence the focus on fixed-wing aircraft.

approaches to analysis and tactical maneuvering in zero-sum 1-vs.-1 scenarios [24, 26, 27, 29, 31–36], machine learning approaches [37–41], and hybrid approaches [42–54]. I expand on some of these contributions below.

Differential Game Theory – Differential games generally consist of two players with known system dynamics, both of whom make a sequence of decisions until some terminating condition is reached (e.g. one player captures the other by coming within a specific range) [31]. Some differential games can be solved exactly, such as the Homicidal Chauffeur problem [31, 55, 56], or the somewhat-more-complex Game of Two Identical Cars [31, 57–59]. In differential game theory, "solved," means that methods exist by which one can compute whether the pursuer will be able to capture the evader—the solution to the "game of kind"— and, when capture is possible, how quickly or to what degree the ultimate objective of one player or the other may be achieved—the "game of degree"—assuming that both players select their optimal actions throughout the game [31]. The majority of the aerial combat literature in this area formulates these differential game scenarios as "zero-sum games," meaning that results occurring during the game that are good for one player are equally bad for its opponent, and vice-versa. Variations of some of these and other scenarios have been used to analyze various 1-vs.-1 aerial combat scenarios in such a way that the scenario can be solved [32, 34, 60–62], but the problem becomes more complex as assumptions are dropped; for instance, solving 2D 1-vs.-1 scenarios with aircraft that can adjust their speeds and turn radii during the game requires numerical solutions [34, 61].

Machine Learning – A number of machine learning approaches have also been used to approach the problem of 1-vs.-1 aerial engagements [37–41]. In particular, McGrew [37] and McGrew, How, Williams, and Roy [38] present work in structuring 1-vs.-1 aerial combat in the horizontal plane as an Markov Decision Process (MDP), then use Approximate Dynamic Programming (ADP) to learn the optimal policy against specific opponent policies. McGrew, himself a former fighter pilot [37], discusses features that human pilots are attentive to during engagements, and employs ADP to learn which of these features increase the performance of the learned aerial combat maneuvering

policies the most [37, 38]. McGrew's approach, however, and even the work of Ma, Xia, and Zhao that extends it to deep learning [39], would face the curse of dimensionality if applied to larger team sizes, as the features the presented algorithms depend upon during the training process are measurements specifically between two aircraft—the protagonist and the opponent. The work presented in Chapter 4 simplifies the features from which the agents make decisions to an embedding that is agnostic to the number of aircraft in the engagement, thus allowing the algorithm presented in Chapter 4 to scale gracefully between small and large engagement sizes.

Hybrid Approaches – A particularly interesting approach to 1-vs.-1 and 2-vs.-1 engagement autonomy is Dynamic Scripting (DS), which uses a genetic algorithm to evolve expert systems for aerial combat [47–54]. These expert systems' responses to decisions consist of pre-defined scripted maneuvers the aircraft can execute, a somewhat similar concept to the work in Chapter 4, albeit with a different decisionmaking algorithm and different pre-scripted maneuvers. An enhancement of DS, Dynamic Scripting + Coordination (DS+C) [48], considers communication between aircraft to be a part of each aircraft's selected script-action, and received messages are added to an aircraft's state. By considering both messages and aircraft physical states in the decisionmaking process, the aircraft develop associations between messages their partner sends and actions they should take, causing the fittest systems to be capable of communicating effectively both within the context of their own and their partner's actions and with respect to the overall scenario. To my knowledge, however, the DS+C approach has not been applied to engagements larger than 2-vs.-2, and would face the curse of dimensionality if scaled to larger engagements, in contrast to the approach demonstrated in Chapter 4.

2.1.2 Many-vs-Many

The primary components of team tactics addressed in the many-vs.-many aerial combat literature are weapon-target assignment algorithms and cognitive architectures for creating teams of cooperating agents.

Operations Research: Weapon-Target Assignment – Weapon-target assignment has been addressed extensively in the literature (e.g. [63–71]). More directly related to UAV aerial combat engagements are the theses of Day and Gaertner [4, 72]. The UAVs in Gaertner's work [72] collaborate with their teammates by sharing positions of sensed enemies and allocating which team members should attack which enemies. Similarly, Day investigates the differences between several centralized and decentralized task assignment algorithms in allocating UAV team members to enemy targets [4]. Unlike these approaches to whole-team assignment problems, the work in Chapter 3 ([13]) examines how specific hand-scripted tactics with their own decentralized target-selection logic employed by individuals or pairs of agents on a team affect the outcomes of engagements of sizes between 2-vs.-1 and 10-vs.-10. Furthermore, the work presented in Chapter 4 shows how agents who are trained to select which of the two tactics employed by entire teams in Chapter 3 ([13]) to leverage at a given moment based on the situational context of an agent and its teammates can be more effective at attriting opponents and preserving own-team UAVs than are teams employing one of either of the hand-scripted tactics alone.

Ernest, Cohen, et al. introduce and demonstrate Genetic Fuzzy Treess (GFTs)—trainable trees of expert systems that make fuzzy decisions—and their application to controlling simulated aircraft in air-vs-mixed-force engagements [73–75] and beyond visual range (BVR) air-to-air [76] engagements. Once trained, the GFT aircraft act in a decentralized manner, coordinating with teammates (when communication is possible) to allocate tasks and roles [75]. In contrast to my work, these papers on GFTs do not discuss the tactics the aircraft learn to execute or the decisions they learn to make in detail, instead focusing primarily upon the general learning mechanism. Moreover, this GFT work on BVR aerial combat, while novel and interesting, is employed to tackle a different problem than what I explore in this dissertation; WVR aerial combat with cannon weaponry requires faster reaction time and more precise aim than is required for the BVR engagement scenarios for which the GFTs in these papers are trained. As such, it is currently unknown how well GFTs trained for WVR scenarios similar to those investigated in this document would perform.

Cognitive Architectures – The literature covering tactical coordination for large-scale cooperative-competitive engagements is largely focused on cognitive architectures that imitate the decisionmaking of human pilots [77–83]. Tidhar et al. [78, 79] use the dMARS cognitive architecture to model teams of pilots in many-vs.-many aerial engagements. The earlier of these papers [78] describes a hierarchical role assignment arrangement similar to that described by Shaw [14] as being used by human pilots for role allocation; two individual aircraft partner together as the leader and wingman of a section, two sections partner together in a similar fashion, and so on. The leaders of sub-teams in this arrangement can choose to either directly control their subordinates or permit them to act in a decentralized manner (autonomously, according to their role on the team). The later paper [79] discusses an enhancement that allows for agents to be on multiple sub-teams and to change team or sub-team membership mid-engagement if it seems advantageous to do so. Similar approaches to dynamic team switching and engagement role allocation are presented in the works of Laird, Jones, and Nielsen [80] and Tambe et al. [81–83]. Rather than focusing on imitating the reasoning process of human fighter pilots, Chapter 4 focuses on training agents to recognize, from their own situational context and the situational contexts of their teammates, when specific pre-scripted known-good tactics are likely to be effective with respect to each teammate and themselves, and if selecting the hand-scripted tactic requiring explicit teammate coordination, with which teammate to coordinate. This loose partner-switching arrangement, while simpler than the sub-team and role switching structures presented in the works discussed in this paragraph, shows itself to be very effective in aerial combat against teams using one of either of the baseline hand-scripted tactics alone. Additionally, while agents employing this learning framework require the ability to train against a team of the opponents employing the tactics against which the protagonist team will be tested, the protagonist team can learn more specifically how to counter the particular opponent tactics and strategy with the hand-scripted-tactic building blocks it has available, rather than using inflexible non-learning maneuvering and role-switching logic.

Multi-Agent Reinforcement Learning – Multi-agent reinforcement learning, and Actor-Critic methods in particular, have seen increasing interest in the literature in recent years [84–87]. Actor-Critic

methods train one or more critic networks to assess the utility of states across the team while the actor networks simultaneously learn from the critic network(s) what actions they should take in each state. The critic network(s) can access more information than the actors' networks can, and so provide individualized, big-picture feedback to the actors during training. The actors then operate without their critics at test-time. One key algorithm of this kind, MADDPG [84], trains a unique policy for each agent on a team, and provides the critics that train these agents with full access to all of the other agents' critics and action-selection distributions—including, for adversarial scenarios, those of the adversary team. While MADDPG may handle large numbers of agents far more gracefully than the 1-vs.-1 works discussed earlier, it unfortunately is not structured to account for agent attrition. In Chapter 4, the learner agents only have access to opponent state information that the agent can observe directly, not the internal decision information of the opponents. Additionally, the inputs to the learning framework introduced in Chapter 4 are agnostic to team size, allowing for agents to continue legeraging the network outputs for decisionmaking even in engagements larger or smaller than those the agents experienced during training, as well as when team sizes change mid-engagement. Training the policy network for the approach in Chapter 4 also does not require the training or use of a critic, and, as it is trained via a policy gradient algorithm, its training setup does not employ a replay buffer.

An additional multi-agent learning approach for cooperative-competitive scenarios is presented in the 2018 work of Hoang et al. [88], which leverages the GDICE [89] algorithm for solving MacDec-POMDPs. Hoang's approach sets a team of learning agents against a team of adversary agents that switch policies. The learner agents train specific countering policies against specific adversary policies, then learn to recognize when an opponent has switched its policy, which dictates when the protagonist agents switch to a different countering policy. The approach in Chapter 4 is somewhat similar in that it learns when and where agents on the protagonist team should use different behaviors, but is more modular in that it allows the protagonist team to leverage known-good pre-existing behaviors.

Zhang et al. [90] leverage deep multiagent RL to train policy, critic, and target prediction net-

works for a pursuit-evasion game between five multirotor pursuers and one evader of unknown motion model. Aside from the environment and team size differences between Zhang et al.'s game and that of the engagements in the experiments presented in this dissertation, the primary differences between Zhang et al.'s learning structure and that that I introduce in Chapter 4 are that (a) Zhang et al. employ an actor-critic off-policy method to train actor and critic networks for each individual agent, while the work in Chapter 4 employs a simple policy network employed by all agents on the trained team and that is trained with an on-policy algorithm; and (b) while the actor and critic networks of the agents in Zhang et al. can be constructed to handle any desired number of team members, it does not appear that these networks in Zhang et al.'s work can scale to engagements with different-sized teams than the size of the team with which the agents are trained, as the actor and critic networks have exactly twice as many inputs as there are agents on the pursuer team. The training framework presented in Chapter 4 may be used in engagements with different numbers of agents than are in the engagements upon which the network is trained, and, as the dimensions of the inputs to the network do not change regardless of how many agents are present on a team, the network may continue to be used in an engagement even as agents are attrited. Additionally, the agents in Zhang et al.'s paper cannot leverage pre-existing tactics in their coordination, and it is not known how well agents trained with their method would perform against multiple evaders or an aggressive opponent team, drawbacks that the work I present in Chapter 4 addresses.

Seraj et al. [91] frame multi-agent communication between heterogeneous agents in a wildfire firefighting environment as a MAH-POMDP and develop a Graph Attention Network scheme to learn facilitate effective inter-agent communication between the two types of robots in the swarm. The two classes of robots have different sensors, state configurations, and can perform different actions, and this HetGAT-based communication learning framework produces a "translator" between the two types of agents to help them effectively collaborate in the performances of their tasks. I examine heterogeneous teams in Chapter 5, but those agents do not communicate with one another or learn to share information with one another in any way. Regarding inter-agent communication on a team, MAGIC [92] also learns how to enable agents on a team to communicate effectively, albeit on

homogeneous teams. In the work in Chapter 3 [13] and Chapter 4, the agents communicating and coordinating with one another are homogeneous, but do not have complex communication requirements in their current form. The pairwise-coordinating tactic employed in Chapter 3 establishes agent tactic pairings via a central authority (which is assumed to be capable of communicating with all agents), and the agents themselves only communicate simple internal state messages with their maneuver partner. More complex communication could be utilized in a version of the work presented in Chapter 4 in communicating agent states between agents, but the simulation framework currently stands as a proof-of-concept that learning from and making decisions based on pairs of ego-centric agent states is fruitful and feasible; incorporating a more complex fully-decentralized system into that learning and evaluation framework is a possible avenue of future work, but is outside of the scope of this document.

Konan, Seraj, and Gombolay [93] present a method for employing a multi-agent RL policy gradient method called InfoPG to solve MAF-Dec-POMDP-framed problems. The agents maximize mutual information between agents working towards the team's goal, and are able to quickly learn how to act in cases where one of the agents acts unreliably. The training and testing framework I present in Chapter 4 is a more strongly adversarial scenario, but the teams in the experiments presented in both Chapter 3 and Chapter 4 contain no agents that act traitorously towards their team and do not attempt to explicitly discern the opponent or opponents' policy or reasoning.

2.2 Additional Background: Lanchester's Laws

Throughout this document, I reference the seminal work of Frederick Lanchester [16] and others who expanded upon his work (e.g. [17, 18]) in discussions of tactical advantage. Lanchester's Laws model the fighting strength of two forces, red and blue, in terms of each force's size at a given time ($R(t)$, $B(t)$, respectively, or simply R, B); as well as each force's attrition-rate coefficient (k_R, k_B), which approximates the rate at which one firer of the force denoted by the coefficient's subscript attrits members of the other team [17]. While I do not conduct rigorous Lanchestrian analysis of the tactics described in this document, Lanchester's Laws are helpful tools for clarifying how force

concentration creates tactical advantage for portions of one team or the other in various scenarios.

For what Lanchester referred to as "ancient warfare," in which the fighting forces utilize spears, swords, and other one-to-one weapons, the ratio of the change in size of the blue force to the change in size of the red force is approximately constant, and is defined as E, the exchange ratio, given in Equation (2.1).[2]

$$\frac{\frac{dB}{dt}}{\frac{dR}{dt}} = E. \tag{2.1}$$

Lanchester's Linear Law, Equation (2.2), comes from rearranging and integrating Equation (2.1) [17, 18].

$$B_0 - B_f = E\left(R_0 - R_f\right) \tag{2.2}$$

B_0 and R_0 are the initial sizes of the blue and red force, respectively, and B_f and R_f are their final sizes. For the linear law, if $E = 1$, the difference in fighting strength between the red and blue forces is simply $R - B$ [16]. Figure 2.1 demonstrates the linear law for $E = 1$ in a scenario where the blue team starts with 100 agents and the red team starts with 75. After ten seconds, the blue team has attrited the entire red team, but is left with 25 agents remaining—the blue team loses as many own-team members as it attrits of the red team.

In so-called "modern warfare" scenarios, where fire may be aimed at opponents from a distance and any member of one team may aim at any member of the opposing team, the ratio in the rates of change of force sizes exhibits a different relationship [16]. The change in the size of the red team over time is given in Equation (2.3), and the change in size of the blue team over time is shown in Equation (2.4).

$$\frac{dR}{dt} = -k_B B, \tag{2.3}$$

[2] See Chapter 2 of [17] and Section 4.2.2 of [18] for discussion on how k_R and k_B relate to E.

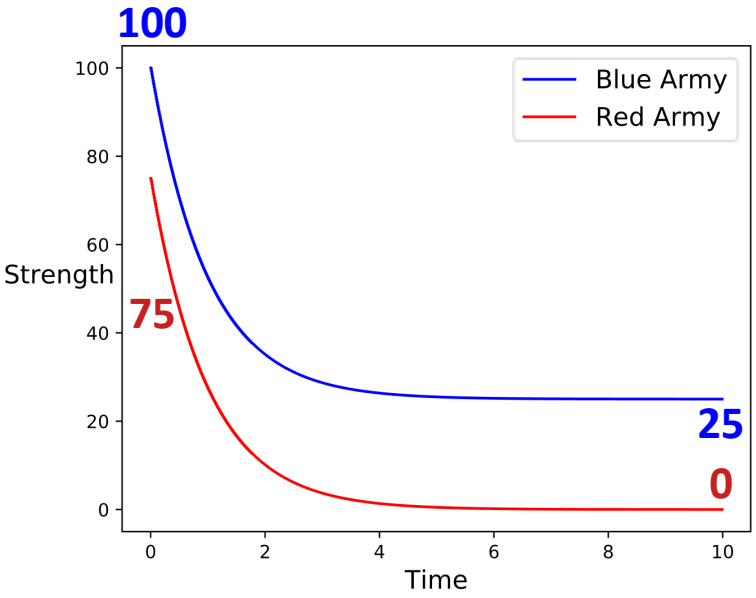


Figure 2.1: In this demonstration of Lanchester's Linear Law [16, 94], the blue army contains 100 agents at time $t = 0$, and the red army starts with 75 agents. Both teams are equally effective at attriting one another; i.e. $E = 1.0$. By the time that the blue team has attrited the entire red team, the blue team's remaining size is only 25 agents—both teams lose the same number of agents.

$$\frac{dB}{dt} = -k_R R \tag{2.4}$$

Divide Equation (2.3) by Equation (2.4), then re-arrange terms and integrate to obtain Lanchester's Square Law, shown in Equation (2.5).

$$k_B \left(B_0^2 - B_f^2\right) = k_R \left(R_0^2 - R_f^2\right) \tag{2.5}$$

That is, the fighting strengths of red and blue are equivalent when the square of each team's size multiplied by that team's attrition coefficient is the same for both teams [16]. By this logic, if both teams' attrition coefficients are equal ($k_R = k_B$), then the team with the numerically-superior force (e.g. R) has a higher fighting strength than its opponent, (B), not by $R - B$ as in the linear law case, but by $R^2 - B^2$. This difference in fighting strength is demonstrated in Figure 2.2, where the blue force starts with 100 agents and the red force with 75 agents, as in Figure 2.1. Both armies

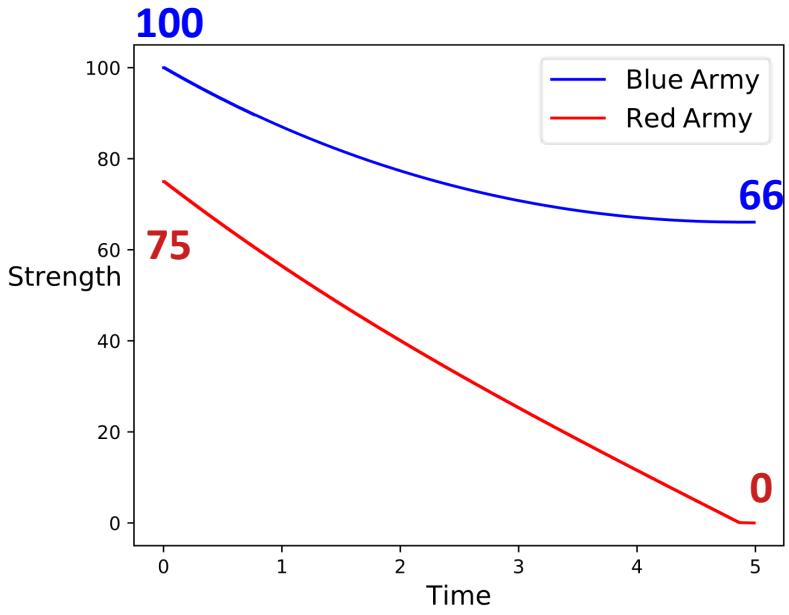


Figure 2.2: This plot demonstrates Lanchester's Square Law [94]. As in Figure 2.1, the blue army starts with 100 agents at time $t = 0$, and the red army begins with 75 agents. Both teams are equally effective at attriting one another; in this case, $k_R = k_B = 0.2$. Due to the blue team's square law advantage, the blue team has reduced the size of the red team to 0 agents after less than $5\,\mathrm{s}$, and the blue team's size at that point is 66 agents. Simply using aimable weaponry and having a numerical advantage in this battle preserved 41 of the blue team's agents compared to the comparable linear law scenario shown in Figure 2.1.

have the same effectiveness, $k_B = k_R = 0.2$. Under the square law, by the time the blue force has reduced the size of the red force to 0, the blue force contains 66 agents—far more than survived in the blue force in the linear law example above.

To demonstrate the difference between the Linear Law and Square Law, if $R = B$ and $k_R = k_B = 1$, but the red team is broken up into two waves of sizes R_1 and R_2 that blue team engages sequentially, does team B have any advantage? Under linear-law battle conditions, neither team has an advantage, as $B - (R_1 + R_2) = 0$ [16]. If this is a square-law situation, however, blue has a significant advantage over red, as $B^2 > (R_1^2 + R_2^2)$ [16].

In this dissertation, I focus primarily on Lanchester's Laws in local application [16]. Lanchester clarified that the laws he defines in his seminal work are not only useful for analysis of battles between large red and blue forces, but also in small regions of larger engagements. i.e. if a portion of the protagonist team can achieve a force concentration advantage against some group of antagoinst

team agents, one may assume that I am implying a local Lanchester's Square Law advantage of the protagonist team's segment over the antagonist team's segment unless otherwise noted.

2.3 Key Takeaways

As discussed in the previous sections, much of the literature pertaining to UAV aerial engagements focuses on individual agent actions within small engagements, how to allocate agents to targets in large engagements, and how a team of aircraft can make human-pilot-like tactical decisions. A number of multi-agent RL approaches have been employed to learn policies for aircraft in small aerial combat engagements or other cooperative-competitive game scenarios, but the teams in these engagements are quite small, often with the largest team containing two or three agents at most. What I believe the swarm-vs.-swarm literature lacks is a comprehensive approach to multi-agent cooperative-competitive aerial combat engagements that leverages tactics that are known to be effective in small *and* large engagements, can learn when and with which teammates to employ these tactics, and can do so in a way that concentrates the team's force on the enemy aircraft effectively. My work fills this gap, first starting with a far-off defense scheme, exploring what affects aerial combat engagements between teams of fixed-wing UAVs employing known-good human-pilot-inspired hand-scripted aerial combat tactics (Chapter 3 [13]) in engagements of various sizes, then details a deep-RL scheme that equips agents on the protagonist team to switch between the hand-scripted tactics based on pairs of agent situational context representations (Chapter 4). I then shift focus to the close-in defense problem and present a bio-inspired close-in defense approach to protecting a fuel reservoir from a heterogeneous team of thieving attackers with a heterogeneous team of defenders fueled from the fuel reservoir, all while trying to maximize the fuel quantity remaining. This defense scenario investigates the force concentration advantages of the two types of guards on the defending team and explores the emergent prioritization of agents to guard roles in response to the nature of the attacking team, and draws comparisons to the biological scenario that inspired the defense scheme (Chapter 5 [15]).

CHAPTER 3

TACTICAL ANALYSIS

In this chapter, I compare the efficacy of a decentralized, greedy tactic and an explicitly-coordinating tactical behavior in swarm-vs.-swarm UAV engagements in simulation, as well as investigate the role of the structure of human-pilot-inspired aerial combat tactics [14] in obtaining favorable force concentration against the opponent team. More specifically, I investigate what affects engagements involving one or both of these two tactical behaviors, specifically with respect to engagement size and the effectiveness of each team's weapons. Both of the behaviors investigated in this chapter have strengths, but are also brittle and exploitable. Here, I refer to a tactical behavior as "brittle" if it is effective at attriting opponents and preserving own-team agents in specific situations, but ineffective at either or both of those aims in many other situations. Brittle can describe both a behavior's inability to adapt to the natural progression of the engagement as well as its inability to effectively counter opponents that do not fall within the tactic's parameters of competence.

3.1 Simulation Design

In these experiments, two teams of fixed-wing UAVs are initialized at opposite ends of a $10\,\mathrm{km}$-by-$10\,\mathrm{km}$ arena, facing each other. Individual team members' initial positions are chosen by sampling from a bivariate normal distribution, with $\mu_x = \pm 4\,\mathrm{km}$, $\mu_y = 0\,\mathrm{km}$, $\sigma_x = 0.001\,\mathrm{km}^2$, and $\sigma_y = 2\,\mathrm{km}^2$. These teams approach one another and act according to the behavior assigned to their team in an effort to attrit as many members of the other team as possible while preserving their own team's numbers. All UAVs are aerodynamically identical, with parameters defined in Table 3.1 [13]. Simulations are conducted in SCRIMMAGE [95], an open-source multi-agent simulation framework, and all aircraft operate under a version of SCRIMMAGE's SimpleAircraft

Table 3.1: Aircraft Model Parameters

Property	**Symbol**	**Value**
Sensing Range	r_s	$1{,}000.0\,\mathrm{m}$
Firing Range	d_f	$100.0\,\mathrm{m}$
Angular Firing Range	δ_f	$3°$
Max bank angle	ϕ_{max}	$45.0°$
Turn Radius	r_{tr}	$34.9\,\mathrm{m}$
Cruise velocity	v_{cruise}	$18.5\,\mathrm{m/s}$
Min velocity*	$v_{min,DA}$	$15.0\,\mathrm{m/s}$
Max velocity*	$v_{max,DA}$	$18.5\,\mathrm{m/s}$
**Only relevant to Double Attack*		

fixed-wing motion model[1] and its corresponding controller, SimpleAircraftControllerPID[2].

Evaluation – The experiments documented in this chapter compare the performance of two teams of UAVs who use either greedy tactics or explicitly-coordinating tactics against a team of greedy-tactic UAVs, and in a way that highlights the equal importance of own-team survival and opponent attrition in coordinating tactics. The hypotheses for these experiments are that agents employing explicitly-coordinating tactics in UAV aerial combat will attrit more opponents and have higher own-team survival than teams employing greedy tactics, but also that a low weapon effectiveness affects the teams that explicitly coordinate to aim more than the implicitly-coordinating agents. The primary metric that quantifies the success of team A in an engagement between teams A and B is the score, $S(\text{AvB})$, given in Equation (3.1). The score is the weighted sum of the percentage of team A UAVs that survive the engagement and the percentage of UAVs of team B that are attrited. In Equation (3.1), N_{A_o} and N_{B_o} are the starting number of UAVs on teams A and B, respectively.

[1] https://github.com/gtri/scrimmage/blob/8cfa91b254912796d25148456a4fac147fb7f362/src/plugins/motion/SimpleAircraft/SimpleAircraft.cpp

[2] https://github.com/gtri/scrimmage/blob/1761c081720692a9540d6791d574a10fb23e26d7/src/plugins/controller/SimpleAircraftControllerPID/SimpleAircraftControllerPID.cpp

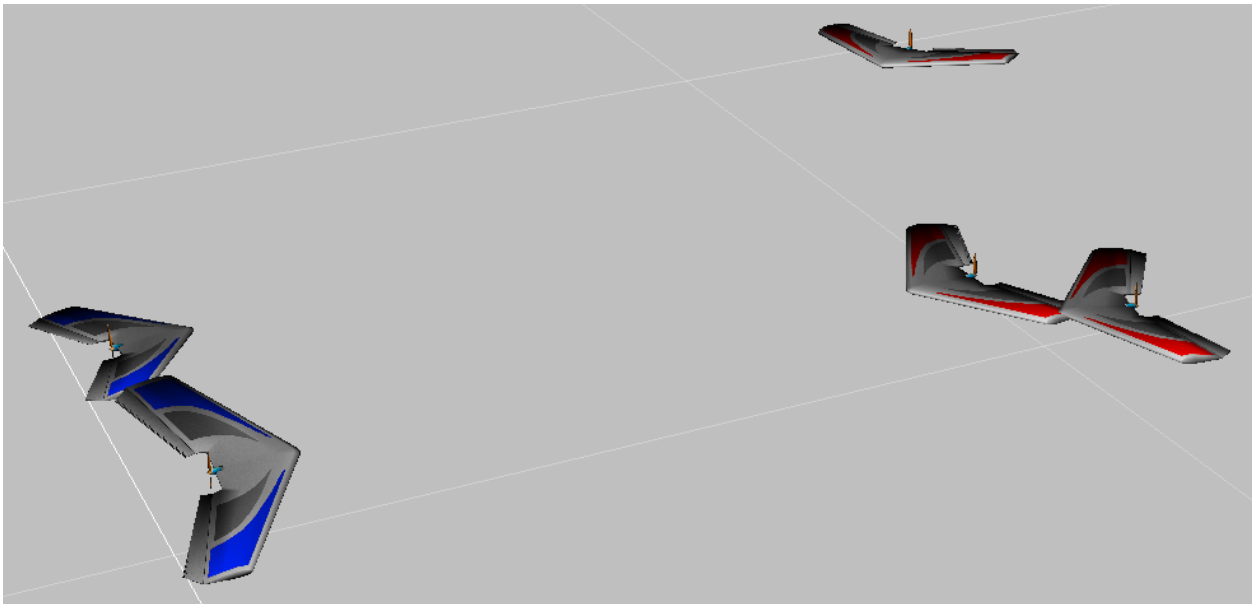


Figure 3.1: A screenshot from an example simulation showing aircraft on two teams (blue, red) engaged in a WVR dogfight.

Similarly, N_{A_f} and N_{B_f} are the final number of aircraft surviving on team A and team B.

$$S(\text{AvB}) = \frac{1}{2}\left(\frac{N_{A_f}}{N_{A_o}}\right) + \frac{1}{2}\left(1 - \frac{N_{B_f}}{N_{B_o}}\right), \tag{3.1}$$

Note that utilizing opponent attrition as the sole metric would score tactics that risk own-team casualties for the sake of opponent annihilation more highly than more conservative tactics that aim to ensure own-team survival. This would penalize coordinating tactics, as agents following coordinating tactics require teammates with whom to coordinate and so aim to preserve teammates when possible. Similarly, a metric that only considers own-team survival would reward evasiveness, not effectiveness in combat scenarios. A combination of both factors, however, is sufficient to score non-coordinating tactics and allows the strength of the simultaneous evasion and firing-opportunity-creation of coordinated tactics to show. The number of aircraft surviving the engagement on each team are recorded as well; this survival data, in combination with the score, provides a more comprehensive picture of how the score of the protagonist team was achieved in the engagement.

Sensing – Each UAV is equipped with a sensor that provides the position of any enemy aircraft within range $r_s = 1\,\mathrm{km}$ of the sensing UAV, as illustrated in Figure 3.2. UAVs can sense the positions and orientations of their team members with no restriction on range.

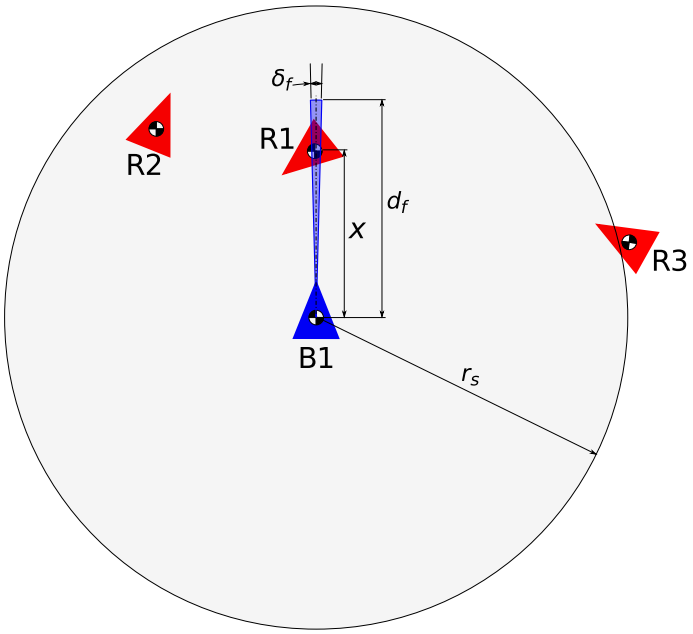


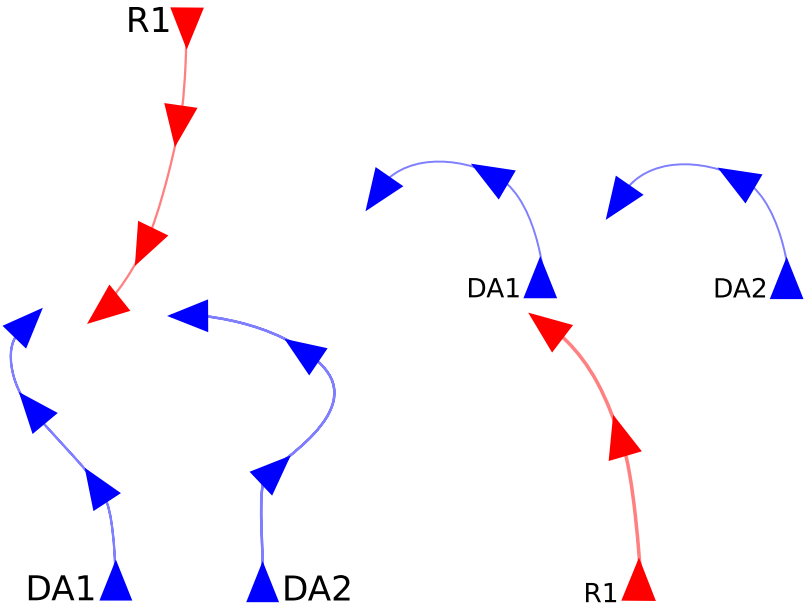
Figure 3.2: The circle denotes UAV B1's sensing range r_s. B1's firing region is indicated by the blue shaded region, which has radius d_f and angle θ_f. B1 can only fire at R1, and can sense R1 and R2, but not R3.

3.2 Experimental Factors

The factors varied in these experiments are the tactical behavior the agents on a team exhibit, the effectiveness of each team's weapons, and the size of the teams.

Agent Behaviors

The UAVs simulated in these experiments follow one of two autonomous behaviors, Greedy Shooter (GS) or Double Attack (DA). These behaviors are modeled after human-pilot-based tactical doctrine [14], and are chosen to emphasize the differences in the efficacy of coordinating and non-coordinating tactics.



(a) Bracket maneuver (b) Sandwich maneuver

Figure 3.3: Maneuvers employed by the DA autonomous behavior. R1 is an enemy aircraft, and DA1 and DA2 are maneuvering according to the DA behavior denoted in the subcaption.

Greedy Shooter (GS) – GS agents aim at the nearest sensable opponent with proportional navigation, updating the opponent it aims at every timestep of simulation [13, 40, 41]. GS agents do not coordinate or communicate with one another in any way.

Double Attack (DA) – Agents employing the DA behavior operate in pairs and depend on coordinated maneuvers to target opponents. This behavior is based on the manned aerial combat tactical doctrine of the same name [14]. The maneuvers DA agents employ, the bracket and the sandwich [14], are illustrated in Figure 3.3 [13].

DA's maneuvers aim to counter individual opponents located in front of or behind a pair of partnered DA agents. The first of these maneuvers, the bracket (shown in Figure 3.3a), is a pincer maneuver that achieves a firing opportunity against an enemy approaching a pair of DA from the front. If an adversary is close behind the DA pair, they instead perform a sandwich (shown in Figure 3.3b), a coordinated turn maneuver. If the enemy is too close behind the DA pair for a sandwich, the DA agents fly directly away from the enemy with speed $v_{max,DA}$. This often results in a stalemate in small engagements, as $v_{max,DA}$ is the same as GS's v_{cruise}.

The pairs DA agents operate in are decided by a centralized pairing function [63]; unpaired Double Attack agents act as GS agents until they can be paired with another unpaired teammate. A pair of DA explicitly coordinate only with each other throughout the simulation, unless one of them is attrited. To encourage DA pairs to spread out and target different opponents from one another, they are equipped with a Boids-inspired pack separation parameter [96], which causes a DA pair to fly away from other DA pairs within $200\,\mathrm{m}$.

To mimic leader-wingman pairs of aircraft flying side-by-side while approaching the locations of enemy agents they intend to intercept, and to ensure that the DA pairs can employ their coordinated maneuvers at the beginning of the engagement, the members of each DA pair are moved to be in echelon formation [14] with each other at the start of each trial. This occurs while all members of both teams are still well beyond each others' sensing ranges, and does not occur at any other time; if DA agents are re-paired later in the engagement, they must achieve echelon formation through their own maneuvering.

Team Size

I conducted experiments for team size configurations of 2-vs.-1, N-vs.-N $\forall$ N $\in \{2, 4, 10\}$, and 2-vs.-M $\forall$ M $\in \{2, 4, 6, 8, 10\}$.

Weapon Model

A UAV fires at the first opponent agent it senses within its firing region (see Figure 3.2 [13]). If the UAV's fire attempt does not violate the frequency restriction of one shot per half-second, whether the shot disables its target depends on p_k, the probability of kill. p_k is a function of β, the firing aircraft's weapon effectiveness; x, the Euclidean distance between the firing aircraft and its target; and d_f, the maximum firing distance:

$$p_k = \begin{cases} \exp\left(\frac{-x}{\beta}\right) & \text{if } x \leq d_f \\ 0 & \text{otherwise} \end{cases}$$

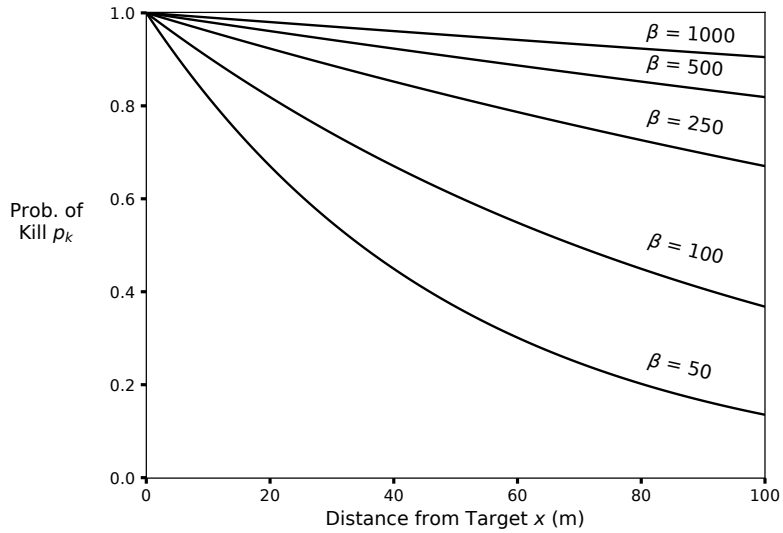


Figure 3.4: The relationship between x, the distance between the firer and firee; the firer's weapon effectiveness, β; and the probability that the firer's shot kills the firee, p_k.

Figure 3.4 shows the relationship between β, x, and the resulting p_k for $x \leq d_f$. The simulations detailed in this chapter and the following chapter ignore the effects of friendly fire, collisions, and the time-of-flight of each shot.

3.3 Procedure

The experiments discussed in this chapter compare the scores of the protagonist team (either DA or GS; team A in Equation (3.1)) against an antagonist team of GS UAVs (team B in Equation (3.1)) in separate engagements of varying sizes and with each team's weapon effectiveness varied. For each team size configuration, team tactic matchup, and weapon effectiveness matchup, I conducted 500 trials. A trial ends either when all members of one team are attrited, or after two hours of simulation time has passed, whichever occurs first. Simulations that reach this time limit are included in the results presented in Section 3.4. All UAVs in these experiments default to flying straight with constant heading when no opponents are within sensing range, and are confined to stay within the bounds of the arena. To ensure that opposing aircraft that are initialized in positions that are

laterally far apart can eventually encounter an opponent, zero-mean, uniformly-distributed random noise in the range $[-0.05\,\mathrm{rad}, 0.05\,\mathrm{rad}]$ is added to each aircraft's initial heading.

3.4 Results

The effects of Double Attack's pairwise coordination are apparent when compared to engagements comprised of two GS teams, though not always to the benefit of the DA team. What DA's coordinated maneuvers lack in aggression and offensive effectiveness, they made up for in terms of resource preservation; as both the bracket and sandwich maneuvers DA agents employ are simultaneously defensive (targeted DA agent evading the enemy) and offensive (un-targeted DA agent targeting the enemy that is chasing after its partner), In most scenarios, DA teams show themselves to be more likely to have more team members survive each engagement than the corresponding GS team. High-density engagements are shown to be a weakness for DA, however; in these dense engagements, groups of GS often surround pairs of DA closely in a way that prevents DA pairs from completing their maneuvers without one or both DA agents being attrited.

3.4.1 2-vs.-1

The 2-vs.-1 engagements are a comparison of DA's coordinated bracketing of a single opponent and GS's naïve approach to the same scenario. DA performs well in these engagements, and strongly outperforms the team of two GS facing a single GS opponent, as seen in Figure 3.5, Figure 3.6, and Figure 3.7. For the 2 GS vs. 1 GS matchup, if the lone GS agent and only one GS on the team of two GS agents survive the initial approach, the remainder of the engagement between the two opponent GS agents generally consists of a tail-chase scenario that ends in a simulation timeout. The DA teams, whose maneuvers incorporate both offensive and evasive components, rarely find themselves in such a 1-vs.-1 tail-chase scenario in these small engagements. As mentioned earlier, DA's bracket is designed for countering isolated opponents and offers a firing opportunity for a DA agent during which the targeted GS agent cannot possibly fire back—the ultimate local force concentration advantage. As can be seen in the decline in scores and blue-team

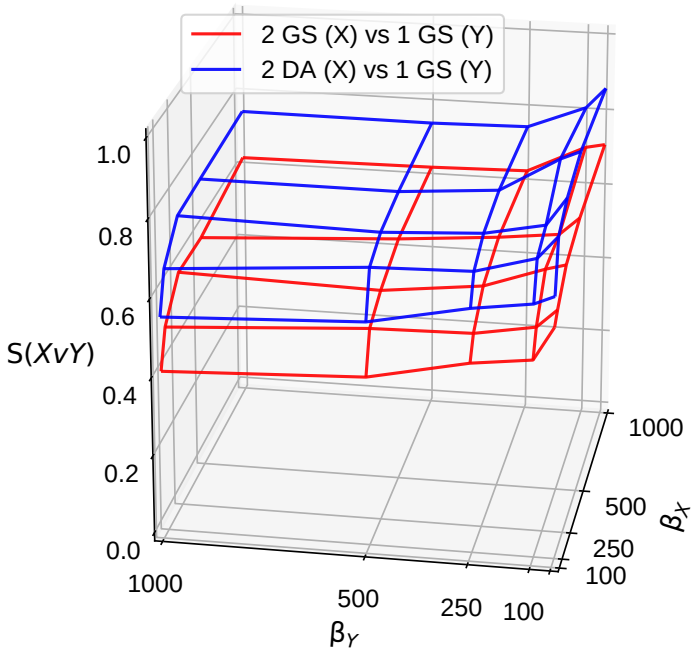


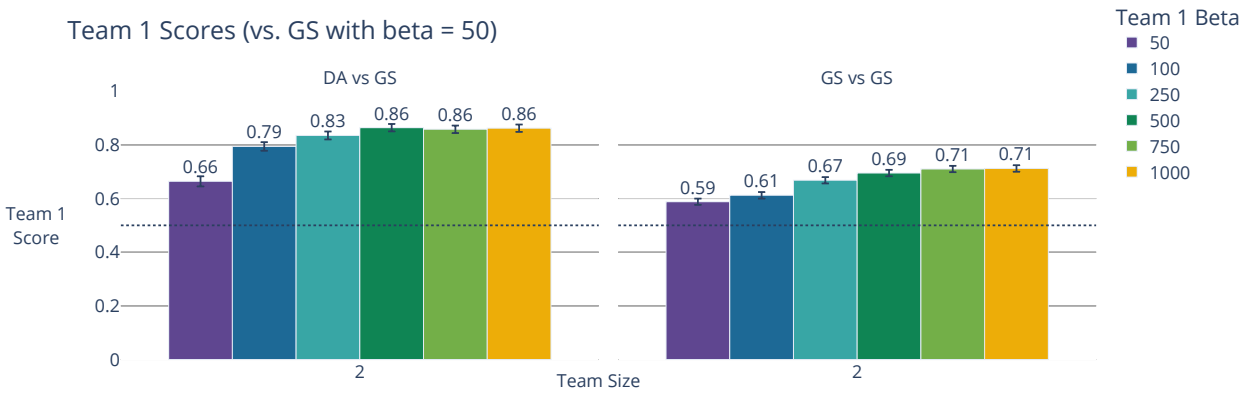
Figure 3.5: Average score plot for two-vs.-one engagements between teams of two DA or GS and one GS.

survival in Figure 3.6 and Figure 3.7, however, the DA team's success declines more sharply than does the team of two GS's as the team of two's weapon effectiveness decreases; this decreased weapon effectiveness causes a firing DA UAV to be less likely to be able to take advantage of the firing opportunity created by it and its partner's maneuvers. Nevertheless, the team of two DA still outscores the team of two GS in all 2-vs.-1 cases.

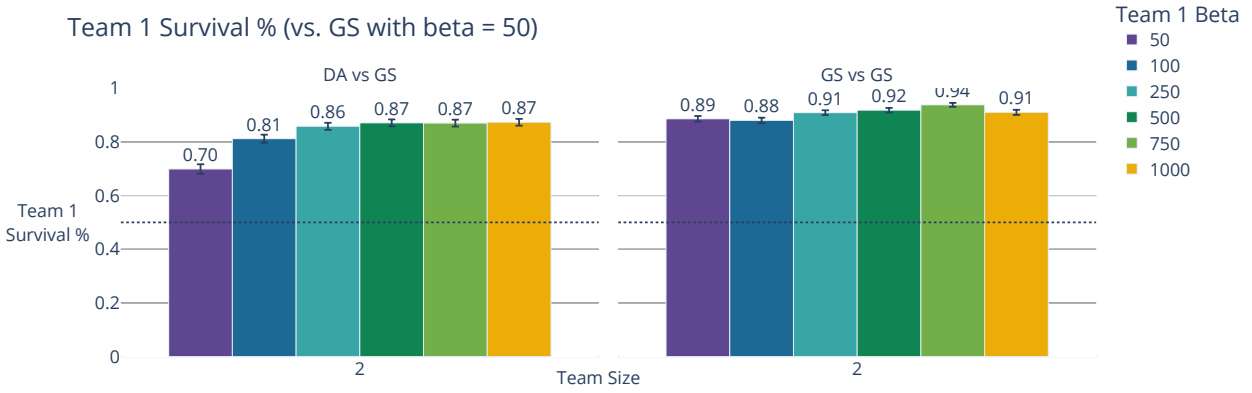
3.4.2 N-vs.-N

The N-vs.-N scenario compares how teams of N DA agents fare against N GS agents to how GS teams of size N counter each other (for N $\in \{2, 4, 10\}$).

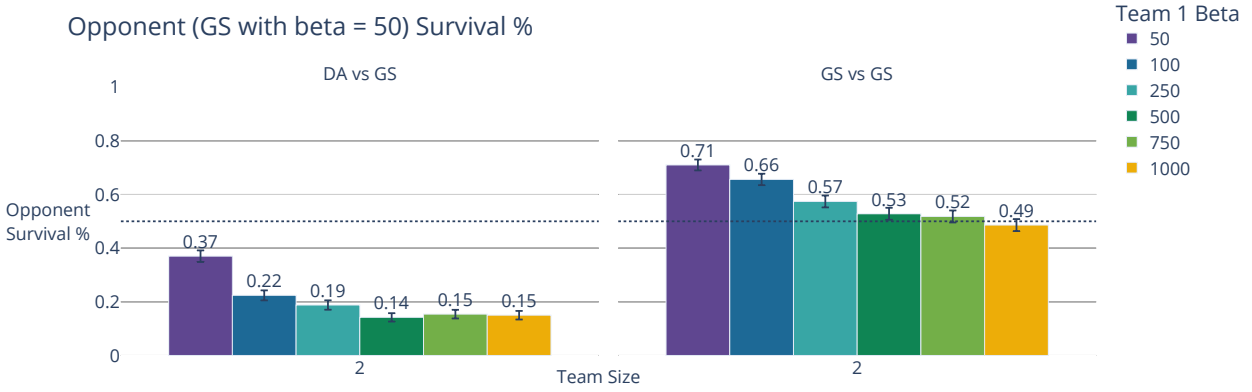
As shown in the 2-vs.-1 cases, DA is adept at countering isolated opponents. DA outscores its corresponding GS team in 2-vs.-2 engagements, both when the opponent's weapon effectiveness is low (Figures 3.11 and 3.12) as well as when the opponent's weapons are high-quality (Figures 3.13 and 3.14). The volatility of DA's scores shown in the 2-vs.-2 and even in the 4-vs.-4 wireframe score plots (Figures 3.8 and 3.9, respectively) is caused by DA's preference for its bracket maneu-



(a) Scores of teams of two DA, two GS agents against a single GS agent with fixed weapon effectiveness of $\beta = 50$.

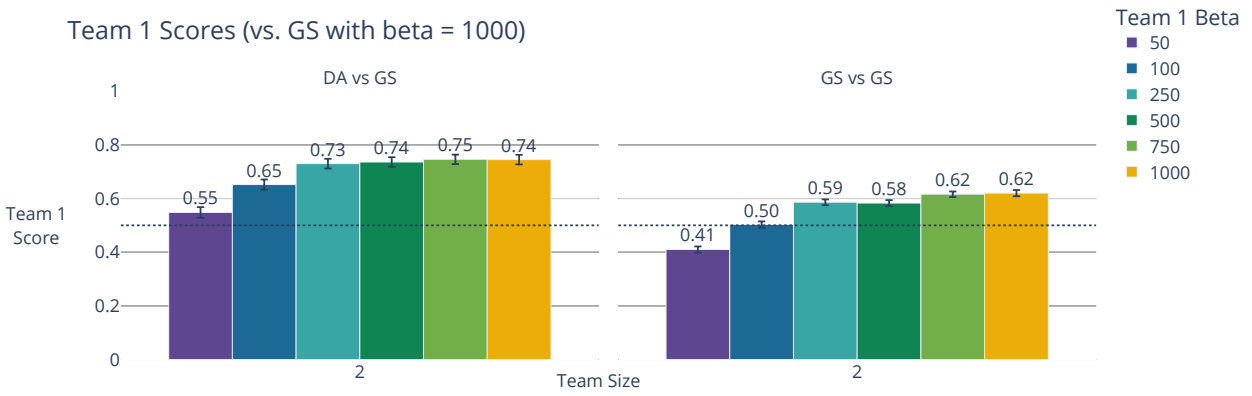


(b) Average survival percentage of teams of two DA, two GS agents against a single GS agent with fixed weapon effectiveness of $\beta = 50$.

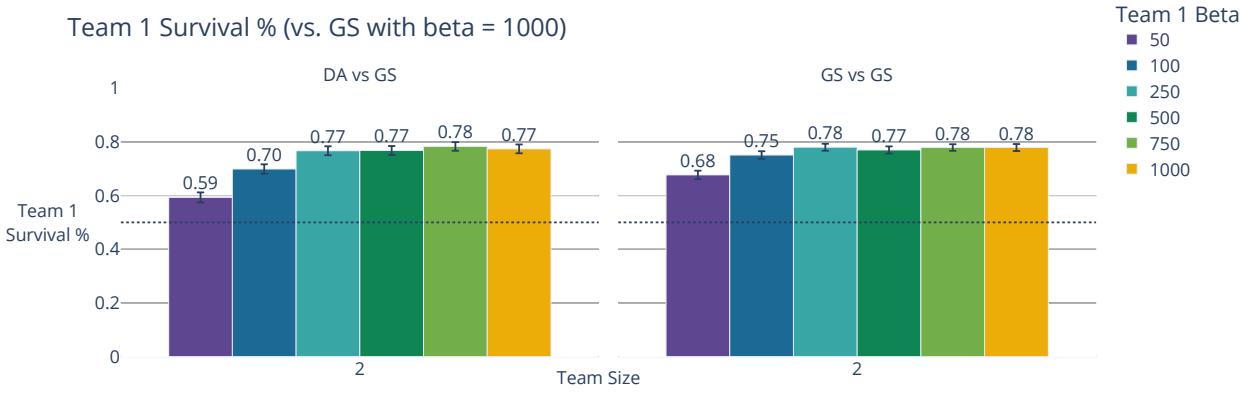


(c) Average survival percentage of single GS agent with fixed weapon effectiveness of $\beta = 50$ in an engagement against either two DA or two GS agents.

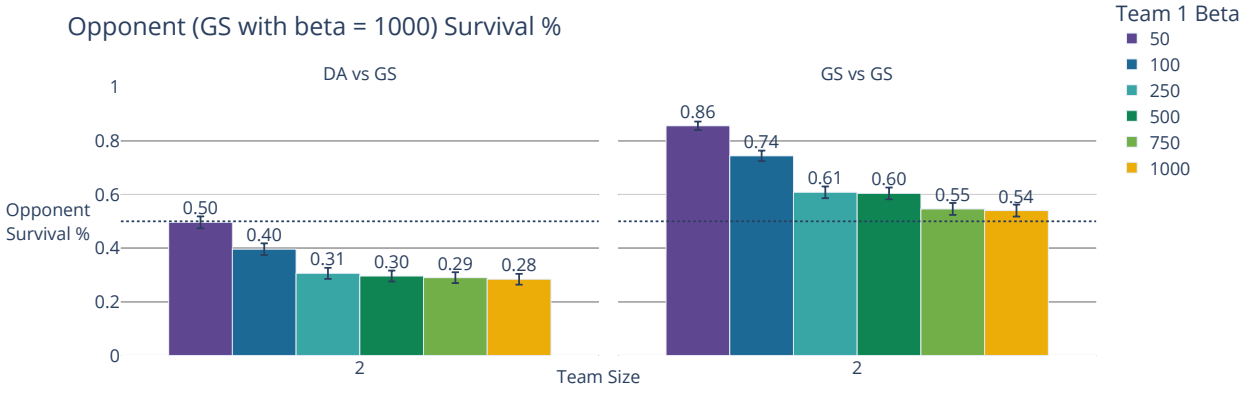
Figure 3.6: Against a single opponent with a poor weapon, the team of two DA outscores the team of two GS for all values of the team of two's β. The team of two GS's own-team survival metrics are stronger across all values of the team of two's β than those of the team of two DA, but the team of two DA's ability to attrit its opponent is unmatched by the team of two GS across all team-of-two β values.



(a) Scores of teams of two DA, two GS agents against a single GS agent with fixed weapon effectiveness of $\beta = 1000$.



(b) Average survival percentage of teams of two DA, two GS agents against a single GS agent with fixed weapon effectiveness of $\beta = 1000$.



(c) Average survival percentage of single GS agent with fixed weapon effectiveness of $\beta = 1000$ in an engagement against either two DA or two GS agents.

Figure 3.7: The team of two DA outscores the team of two GS for every value of the team of two's β when the team of one's $\beta = 1000$. When equipped with a poor weapon, the team of two DA's survival suffers more than that of the team of two GS with the same weapon, highlighting that DA is more dependent upon an effective weapon than GS.

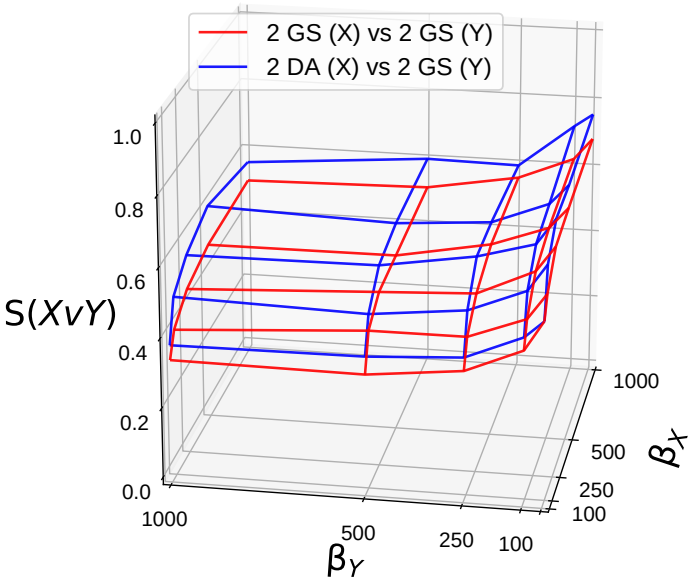


Figure 3.8: Average score plots for 2-vs.-2 engagements.

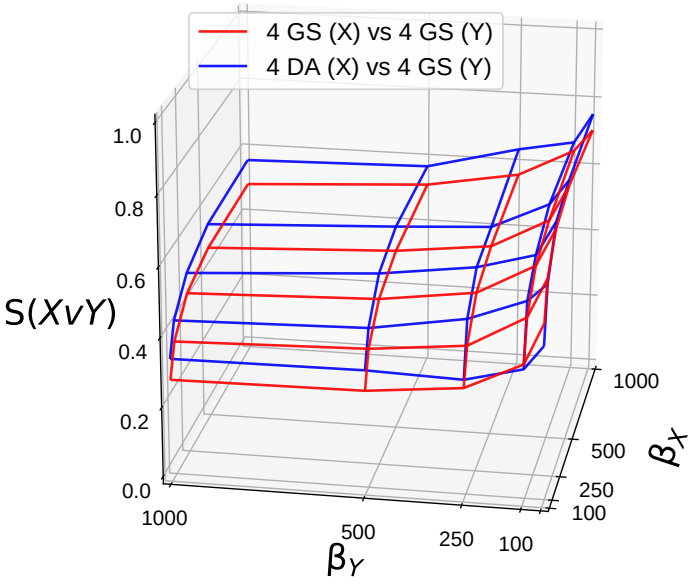


Figure 3.9: Average score plots for 4-vs.-4 engagements.

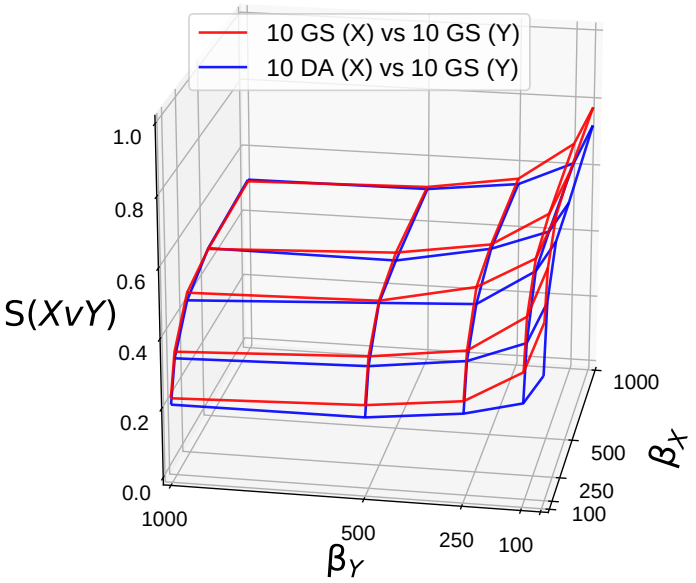
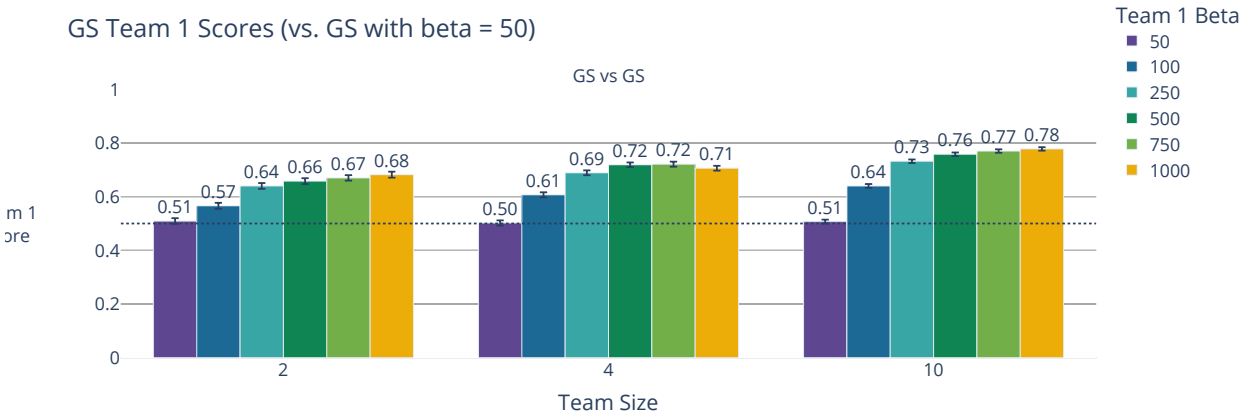


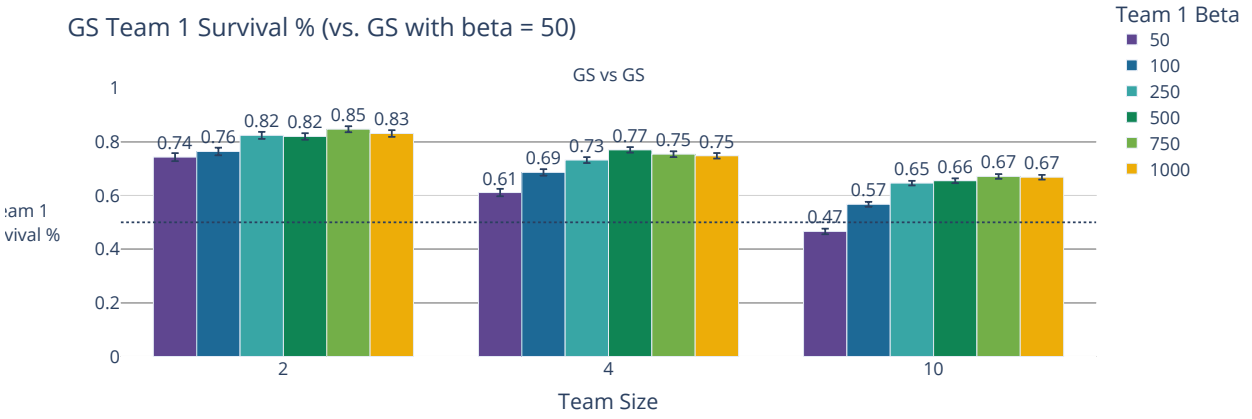
Figure 3.10: Average score plots for 10-vs.-10 engagements.

ver. If the two GS start close to one another, the DA pair tries to bracket one GS, while the other GS can often shoot one DA before the two DA finish their maneuver. The remaining DA must then face two GS in a close-in engagement, acting as a GS agent due to its lack of partner. If the two GS start far apart from one another, however, the 2-vs.-2 DA-vs.-GS scenario is simply two sequential 2 DA vs. 1 GS engagements, the type of scenario DA is designed to counter.

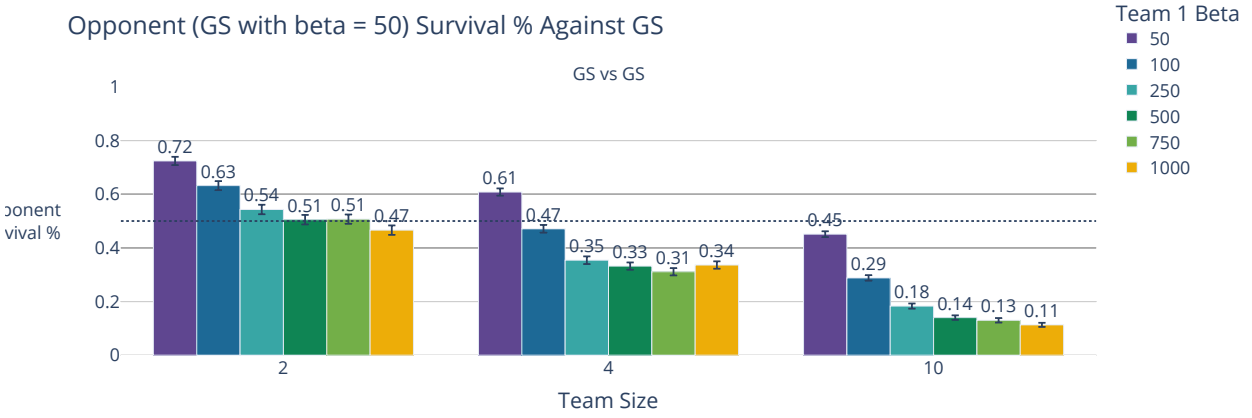
In situations with high opponent density, such as in the 10-vs.-10 engagements shown in Figure 3.10 and shown for specific antagonist team weapon effectivenesses in Figures 3.11 to 3.14, DA is at a disadvantage when compared to GS. A pair of DA require at least six turn radii of empty, non-hostile space between themselves and their target opponent in order to perform a bracket, and their sandwich maneuver is similarly constrained. For larger N in the N-vs.-N experiments, engagements are more likely to develop into a dense melee in the center of the arena after the initial approach of the two teams. A DA pair performing a bracket or sandwich in this dense area will likely lose at least one member to enemy fire before the maneuver can have any effect on the opponent, particularly in engagements in which the opponent team's weapon effectiveness is high. This limitation hampers DA's performance for larger N. Additionally, Figures 3.11 and 3.12 show that, for N $= 10$ and opponent $\beta = 50$, the β-advantaged team of GS in the GS-vs.-GS engagements



(a) Scores of teams of N GS agents against a team of N GS agent with fixed weapon effectiveness of $\beta = 50$.

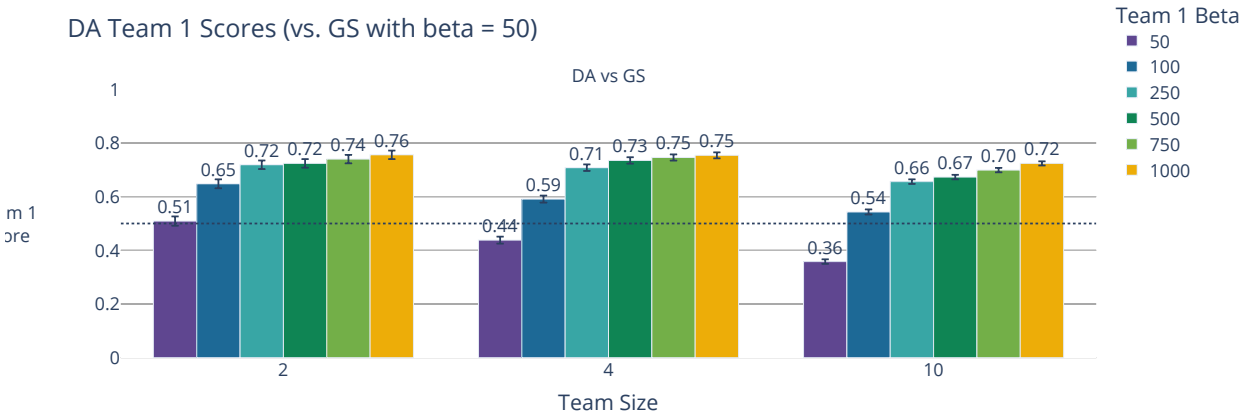


(b) Average survival percentage of teams of N GS agents against a single GS agent with fixed weapon effectiveness of $\beta = 50$.

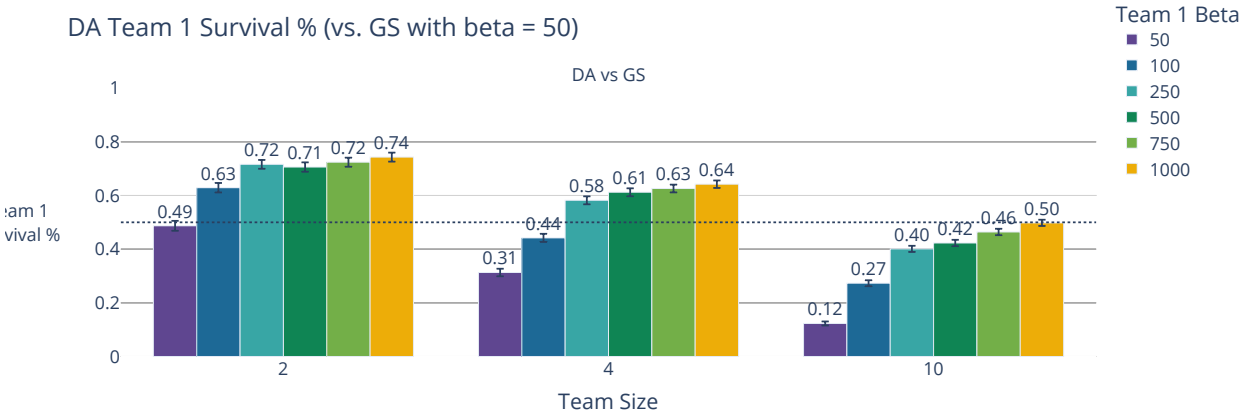


(c) Average survival percentage of N GS agents with fixed weapon effectiveness of $\beta = 50$ in an engagement against N GS agents.

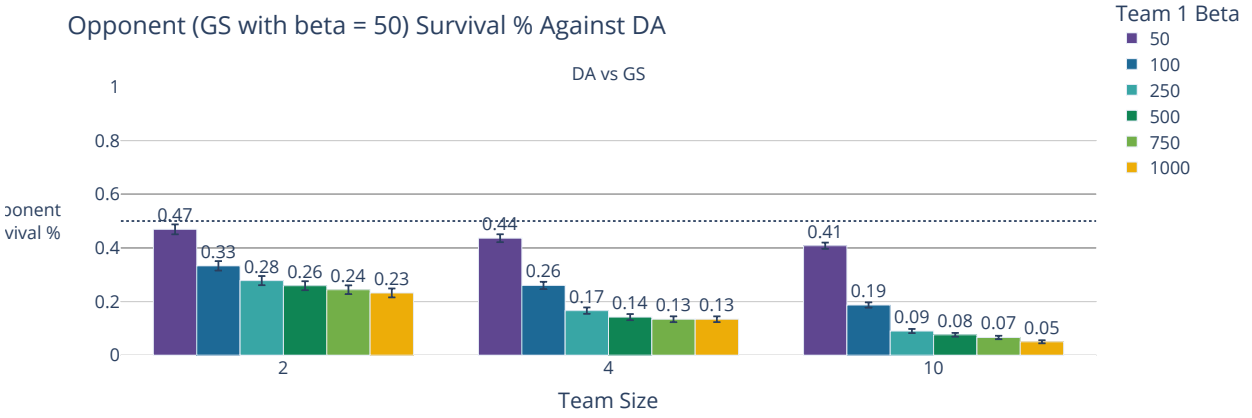
Figure 3.11: Average score, survival percentage, and opponent survival percentage for N-vs.-N engagements with a GS protagonist team and with the opponent team having $\beta = 50$.



(a) Scores of teams of N DA agents against a team of N GS agent with fixed weapon effectiveness of $\beta = 50$.

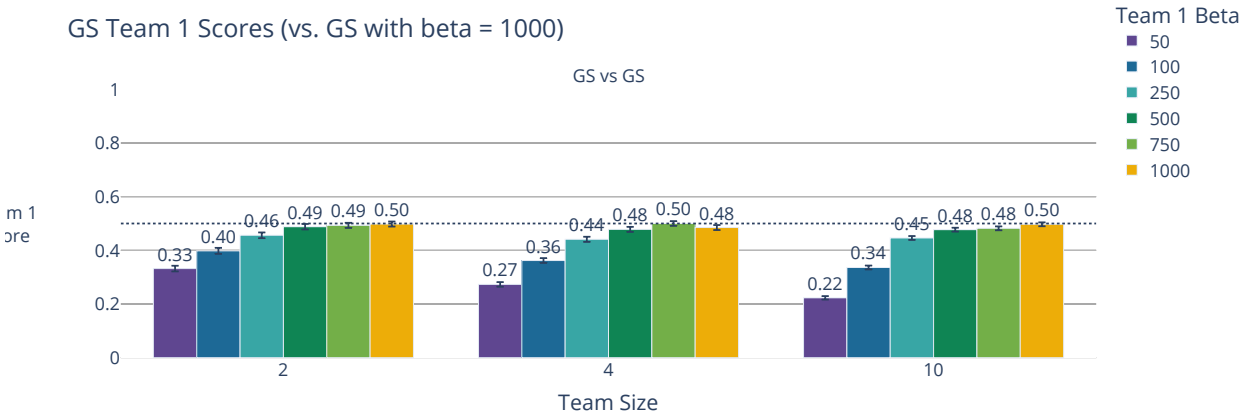


(b) Average survival percentage of teams of N DA agents against a single GS agent with fixed weapon effectiveness of $\beta = 50$.

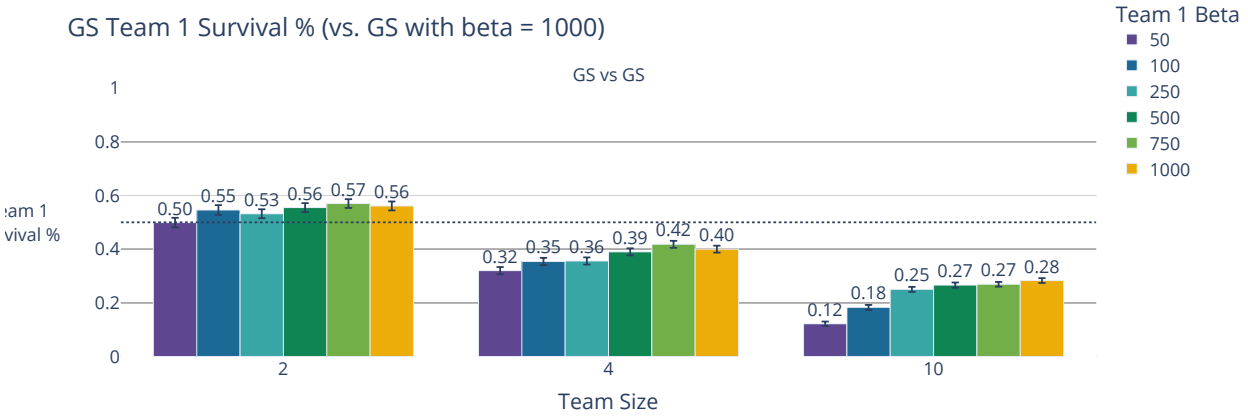


(c) Average survival percentage of N GS agents with fixed weapon effectiveness of $\beta = 50$ in an engagement against N DA agents.

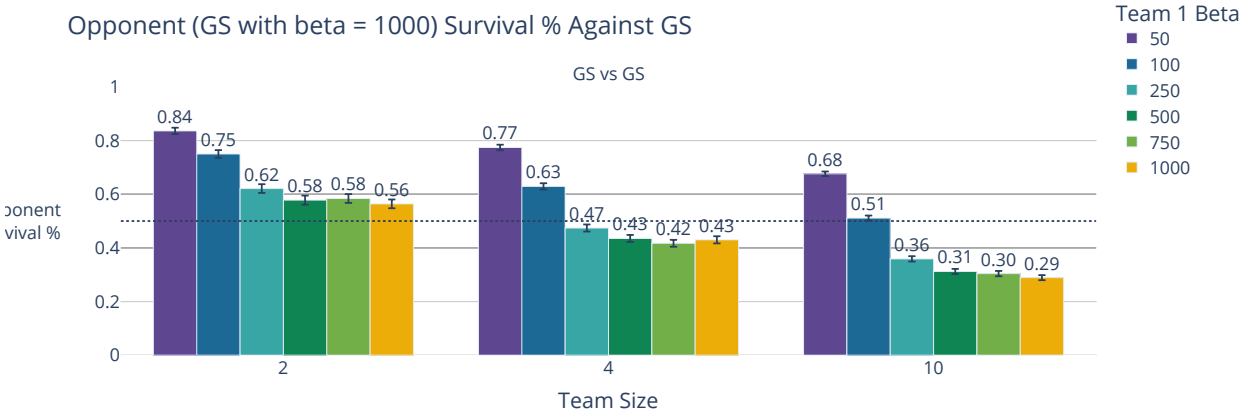
Figure 3.12: Average score, survival percentage, and opponent survival percentage for N-vs.-N engagements with a DA protagonist team and with the opponent team having $\beta = 50$.



(a) Scores of teams of N GS agents against a team of N GS agent with fixed weapon effectiveness of $\beta = 1000$.

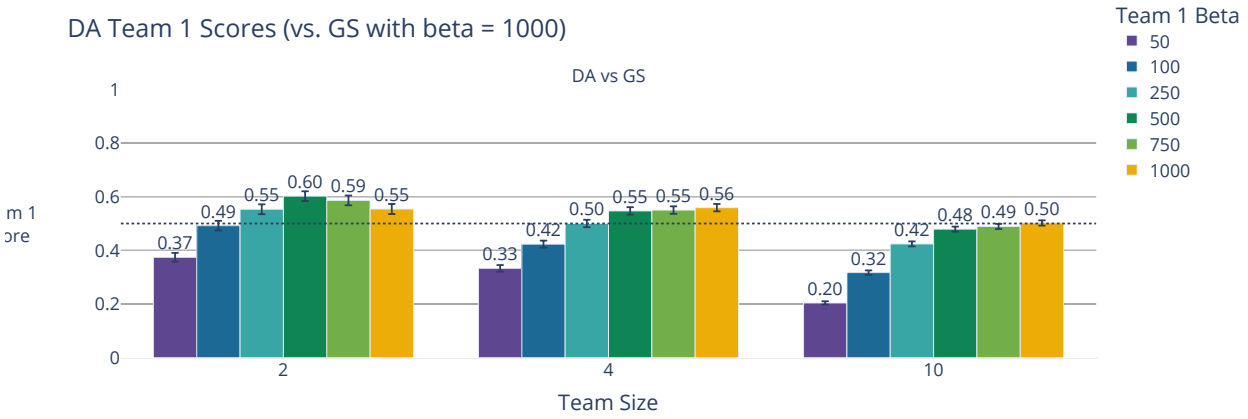


(b) Average survival percentage of teams of N GS agents against a single GS agent with fixed weapon effectiveness of $\beta = 1000$.

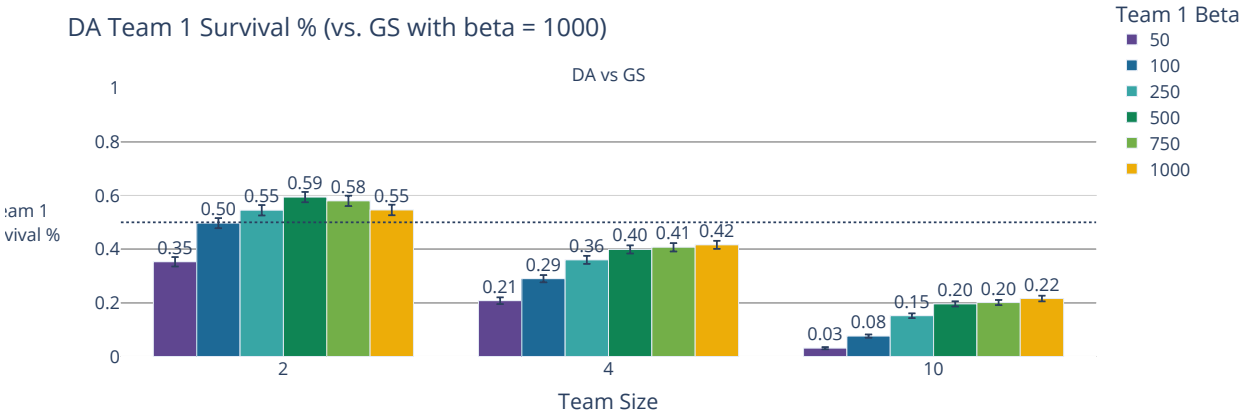


(c) Average survival percentage of N GS agents with fixed weapon effectiveness of $\beta = 1000$ in an engagement against N GS agents.

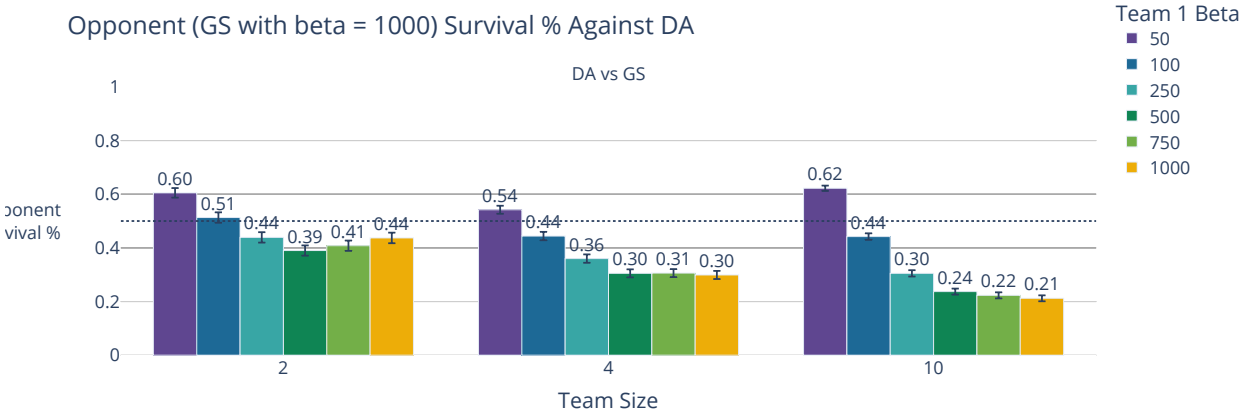
Figure 3.13: Average score, survival percentage, and opponent survival percentage for N-vs.-N engagements with a GS protagonist team and with the opponent team having $\beta = 1000$.



(a) Scores of teams of N DA agents against a team of N GS agent with fixed weapon effectiveness of $\beta = 1000$.



(b) Average survival percentage of teams of N DA agents against a single GS agent with fixed weapon effectiveness of $\beta = 1000$.



(c) Average survival percentage of N DA agents with fixed weapon effectiveness of $\beta = 1000$ in an engagement against N GS agents.

Figure 3.14: Average score, survival percentage, and opponent survival percentage for N-vs.-N engagements with a DA protagonist team and with the opponent team having $\beta = 1000$.

score more highly than do the team of ten DA with the same weapon advantage did in the DA-vs.-GS engagements. The scores of ten-DA and ten-GS teams are more similar across their weapon effectiveness cases when they face opponents with weapon effectiveness $\beta = 1000$ than when they were facing opponents with $\beta = 50$. Nonetheless, the difference in scores between DA and GS teams against opponent GS teams with $\beta = 50$ in 10-vs.-10 engagements highlights how DA's tactics are not designed for dense engagements; for such scenarios, a GS team gets more value out of a weapon advantage than a DA team.

Figures 3.11 and 3.12 offers interesting insights into the role of weapon advantage in air-to-air engagements. As shown in Figure 3.11a, the average scores of the GS teams of N generally increase with increasing N when the GS team being scored has a weapon effectiveness advantage over the opposing GS team with $\beta = 50$. In contrast, the scores of the DA teams in the same scenarios—increasing engagement size, against an opposing team with poor weapons—show a decrease in score as N increases, as shown in Figure 3.12a. The decrease in score of the DA teams as their weapon quality drops is more severe than that of their GS counterparts for all team sizes tested, especially when the DA team's weapon is very poor ($\beta = 50$ or $\beta = 100$); this indicates that DA's sensitivity to dense engagement scenarios generally outweighs its β advantage, particularly in comparison with a GS team with the same weapon advantage against their opponents. In the survival percentage plots (Figures 3.11b, 3.11c, 3.12b and 3.12c), note that the teams of DA sustain more losses—but also attrit more of their opponents—than do the corresponding GS team. In contrast with the GS team's increasing score with increasing N, the survival percentage of the scored team of N GS decreases with increasing N; their increase in scores with increasing N is driven by the increase in the number of opponents they were able to attrit as N increases. The team of N DA, however, sees decreasing survival percentage of its own team with increasing N.

3.4.3 2-vs.-M

The results for the 2-vs.-M engagements, shown for specific team-of-M weapon effectivenesses in Figures 3.19 to 3.22, and for all M $\in \{4, 6, 8, 10\}$ in Figures 3.15 to 3.18 indicate that, as M

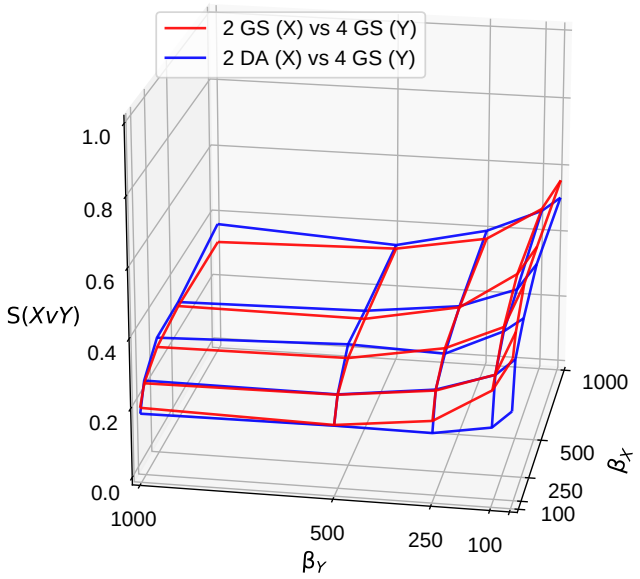


Figure 3.15: Average scores in 2-vs.-4 engagements.

increases, both β and agent behavior become far less influential in the outcome of the engagement, a finding similar to results from [97].

Behavior still plays a large role in engagement outcome, however, particularly when the team of M has low β—in that case, the team of two GS performs better against this weapon-disadvantaged team of M than does the team of two DA $\forall$ M, highlighting DA's dependency on an effective weapon. Two GS combating four GS have a stronger advantage over their opponent than two DA combating four GS when the four GS have weak weapons, but the two DA combating four GS opponents are less severely impacted by an increase in their opponent's β.

While one must expect the teams of two to be at a disadvantage for M > 2, it is interesting to compare the teams of two against one another in how slowly their scores degrade as M increases. Figure 3.23 shows that, with all teams having $\beta_x = \beta_y = \beta = 100$, when M $= \{2, 4\}$, the team of two DA outperforms the team of two GS; only at M $= 6$ does the team of two GS outperform them, and only by a small margin. Trends with both the teams of two and of M having the same β indicate that, for $\beta > 100$, the team of two DA outperforms the team of two GS for all values of M, though still approaches the same asymptote that the scores of all cases are seen approaching in Figure 3.23. This performance of the two DA is likely due to the high β allowing DA's maneuvers to

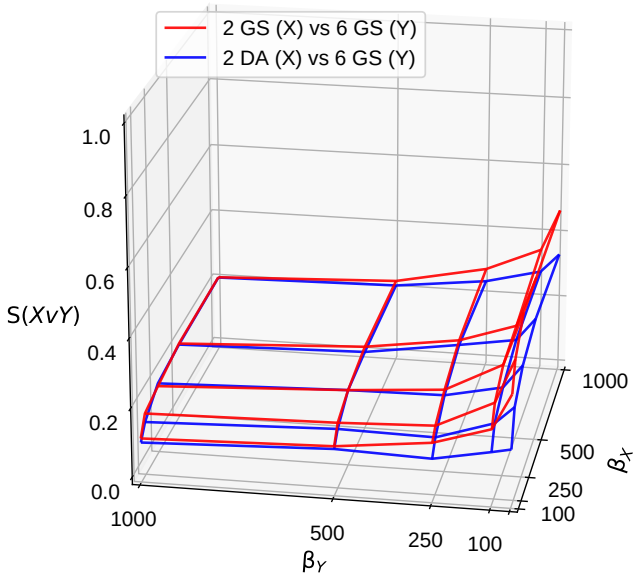


Figure 3.16: Average scores in 2-vs.-6 engagements.

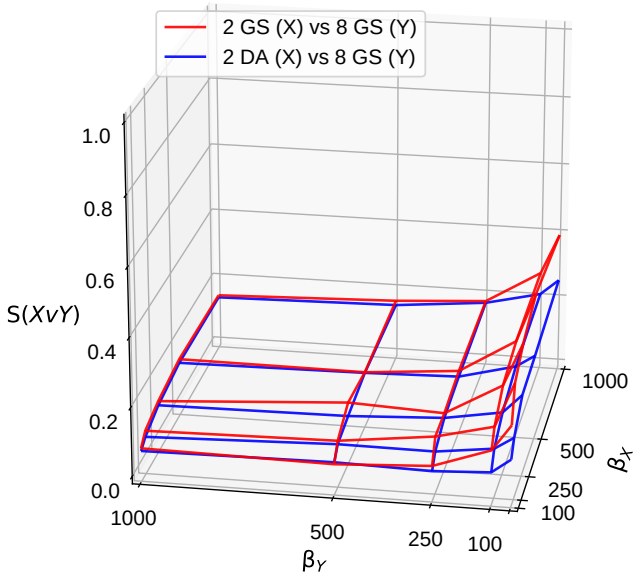


Figure 3.17: Average scores in 2-vs.-8 engagements.

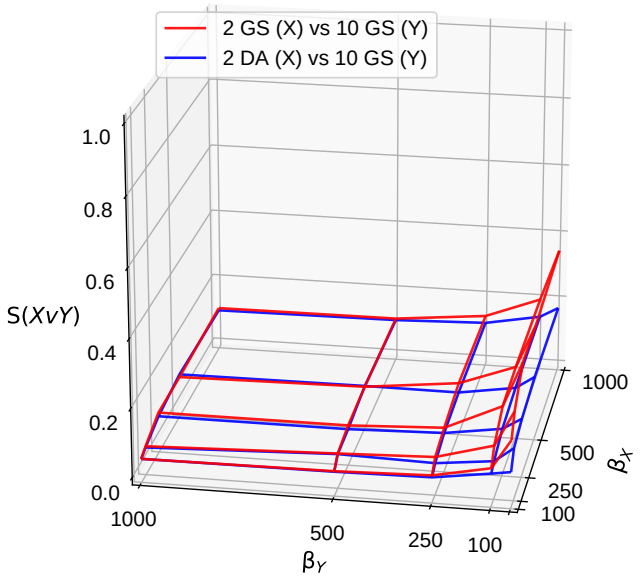


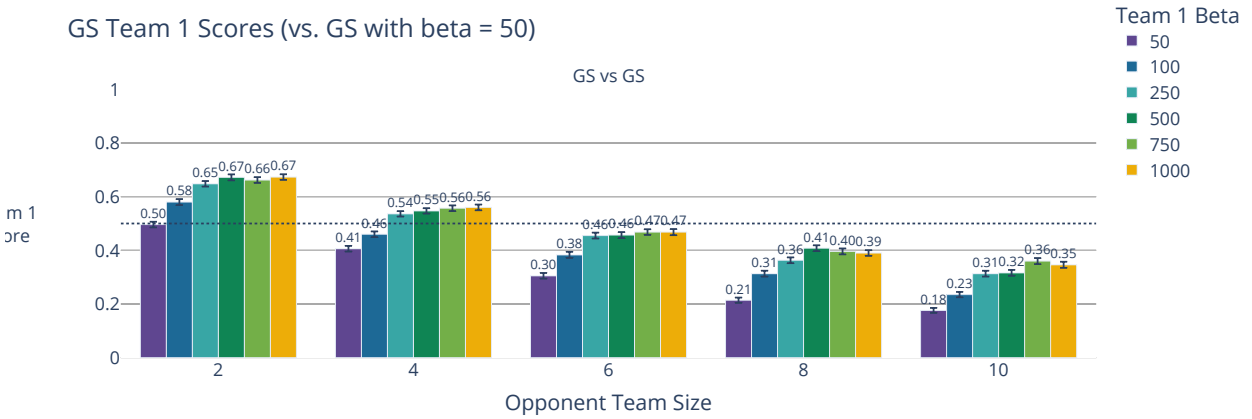
Figure 3.18: Average scores in 2-vs.-10 engagements.

be especially effective.

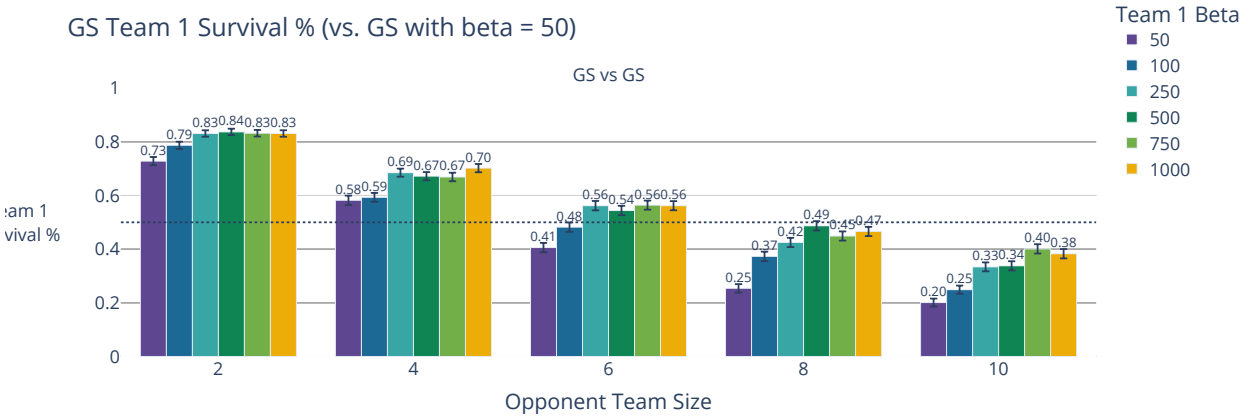
3.5 Discussion

One hypothesis for these experiments is that low weapon effectiveness affects DA agents more than GS agents, support for which is especially apparent in the differences between the DA and GS team-1 scores shown in Figures 3.11 and 3.12, respectively.

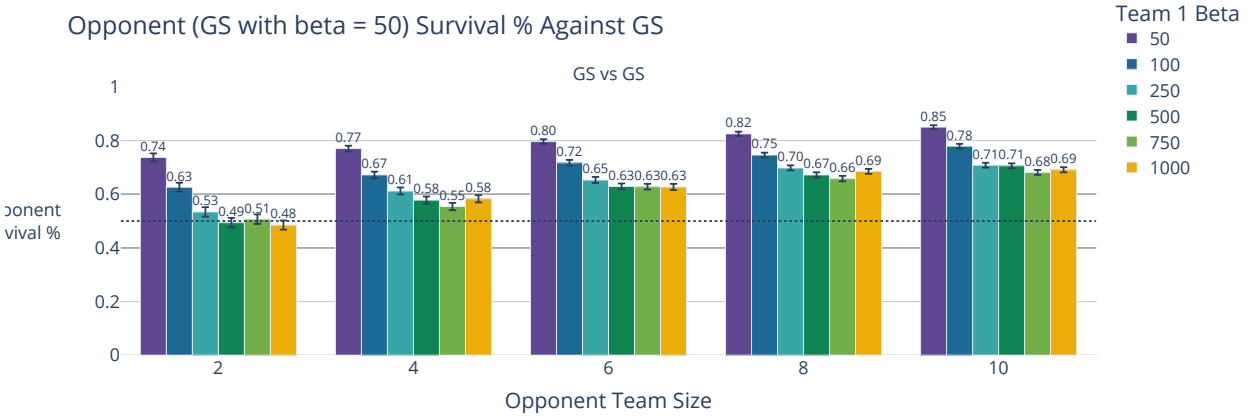
The second hypothesis is that explicitly-coordinated maneuvers are more effective in UAV aerial combat than non-explicitly-coordinated maneuvers, with the null hypothesis being that explicitly-coordinated maneuvers are equally effective or less effective than their non-explicitly-coordinating counterparts. While the coordinated maneuvers of DA are far more effective than their GS-team counterparts in small, 2-vs.-1 engagements and in 2-vs.-2 engagements, as the engagement arena becomes more dense, the DA teams demonstrate less effectiveness than their GS counterparts against opponents equipped with low-quality weapons. The DA teams in the N-vs.-N for $\mathrm{N} = 2, 4$ engagements outperform their GS counterparts for opponents with poor ($\beta = 50$) and high-quality ($\beta = 1000$) weapons, but fall short of their GS counterparts when the opponents' weapons are poor, and only meet the performance of the GS counterparts against opponents with



(a) Average scores of teams of two GS agents against a team of M GS agent with fixed weapon effectiveness of $\beta = 50$.

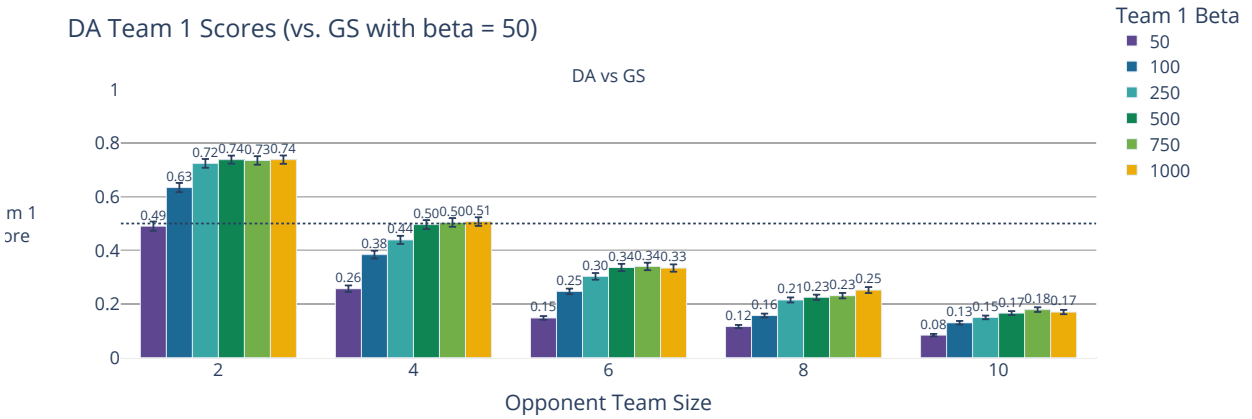


(b) Average survival percentage of teams of two GS agents against a team of M GS agent with fixed weapon effectiveness of $\beta = 50$.

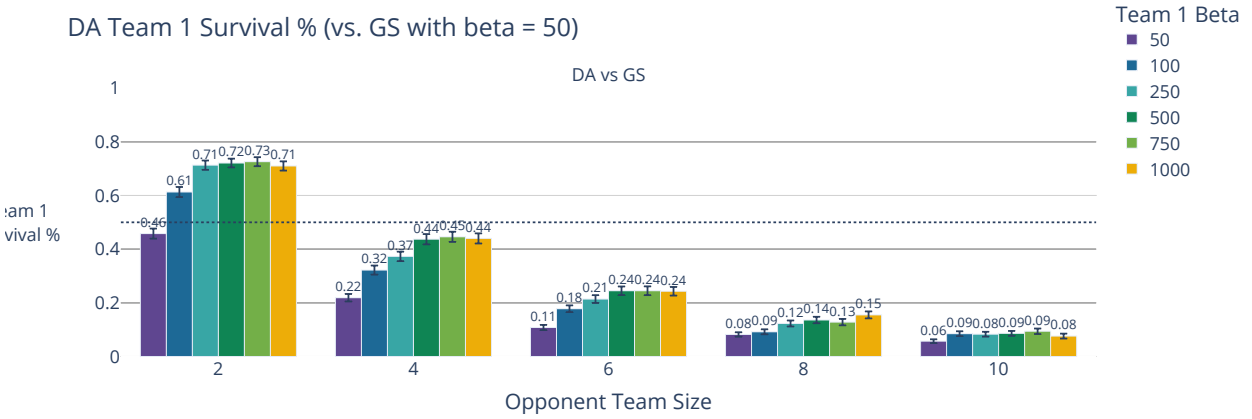


(c) Average opponent survival percentage for teams of two GS agents against a team of M GS agent with fixed weapon effectiveness of $\beta = 50$.

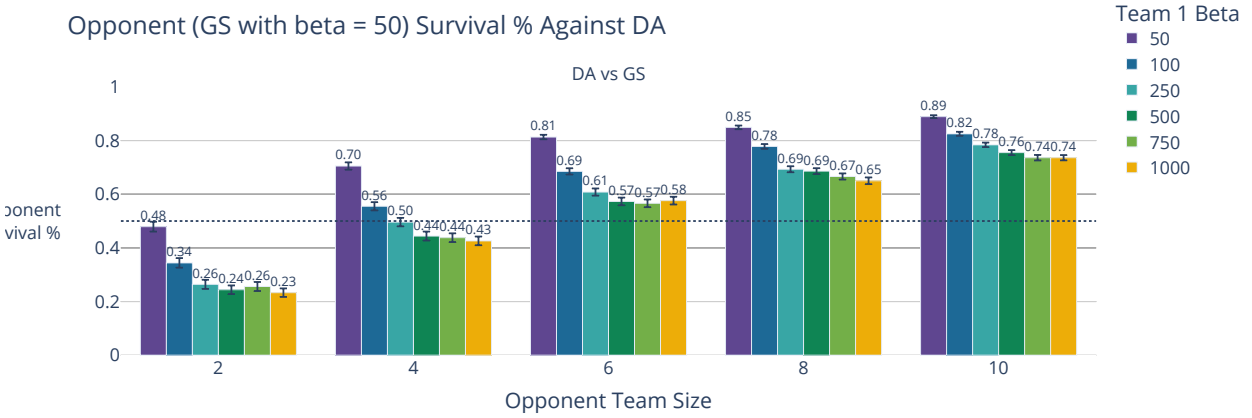
Figure 3.19: Average score, survival percentage, and opponent survival percentage for 2-vs.-M engagements with a GS protagonist team and with the opponent team having $\beta = 50$.



(a) Average scores of teams of two DA agents against a team of M GS agent with fixed weapon effectiveness of $\beta = 50$.

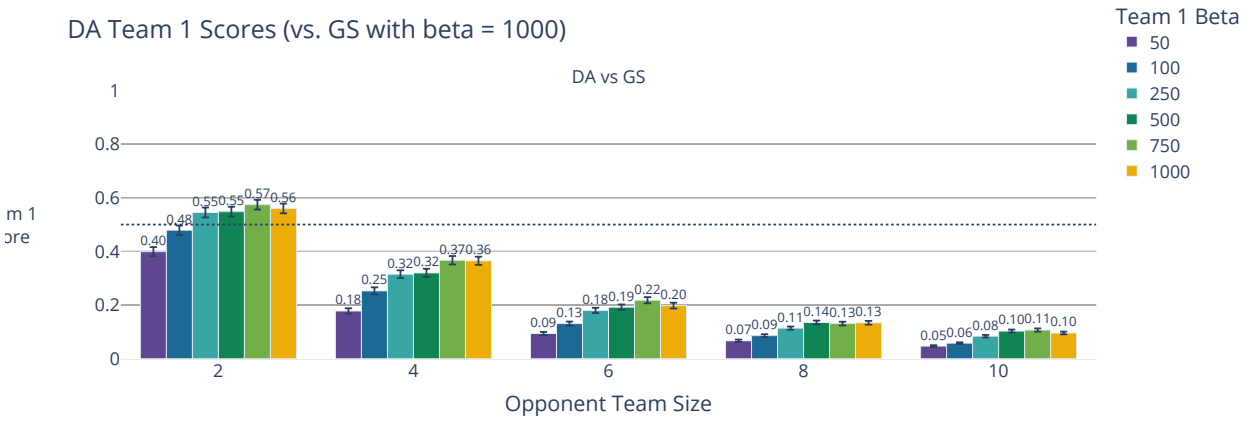


(b) Average survival percentage of teams of two DA agents against a team of M GS agent with fixed weapon effectiveness of $\beta = 50$.

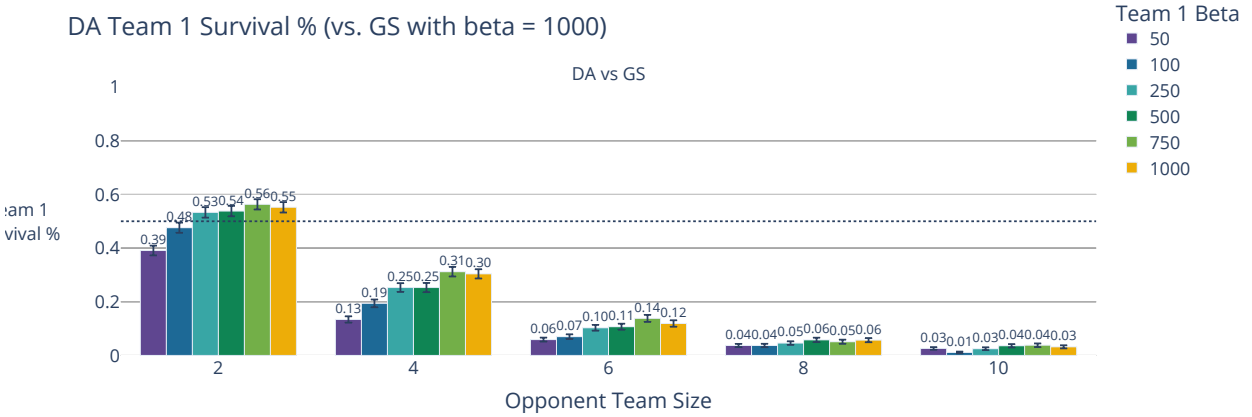


(c) Average opponent survival percentage for teams of two DA agents against a team of M GS agent with fixed weapon effectiveness of $\beta = 50$.

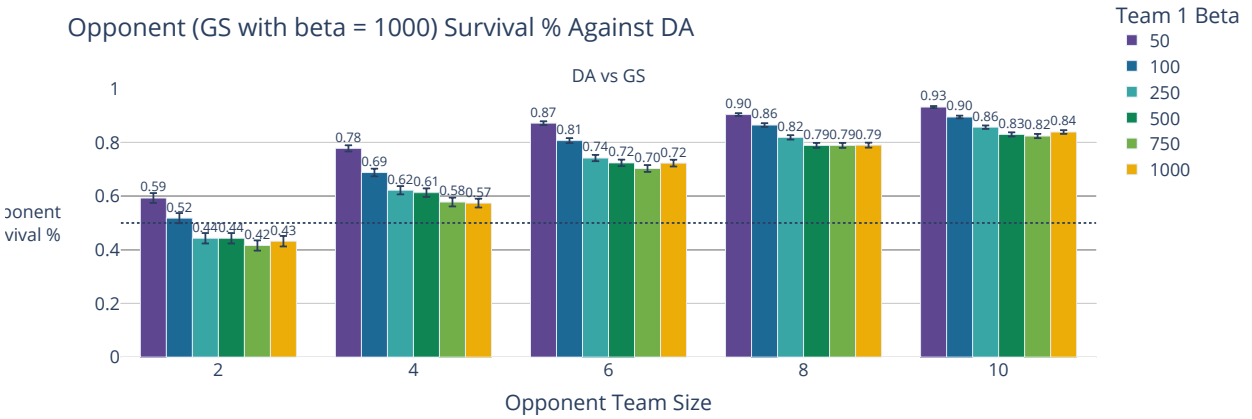
Figure 3.20: Average score, survival percentage, and opponent survival percentage for 2-vs.-M engagements with a DA protagonist team and with the opponent team having $\beta = 50$.



(a) Average scores of teams of two GS agents against a team of M DA agent with fixed weapon effectiveness of $\beta = 1000$.

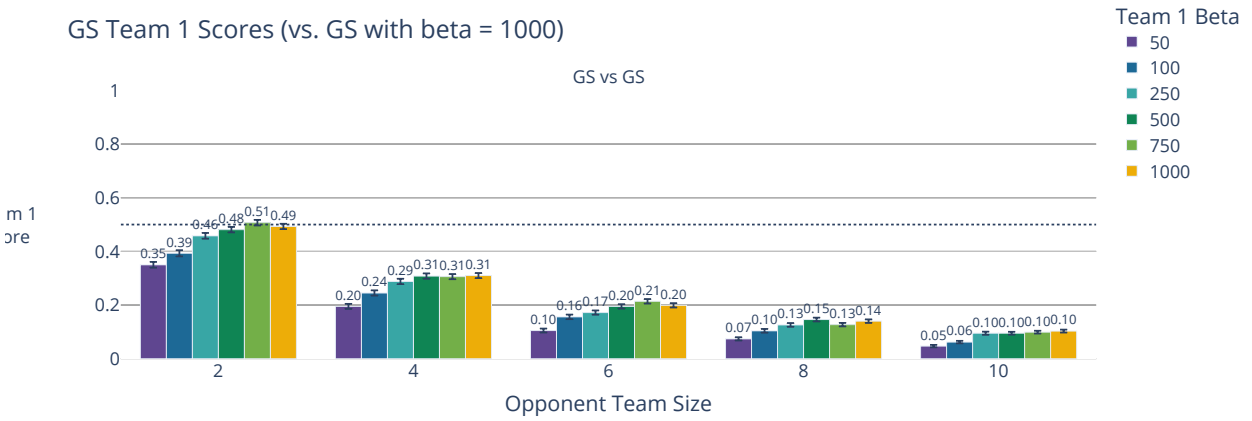


(b) Average survival percentage of teams of two GS agents against a team of M DA agent with fixed weapon effectiveness of $\beta = 1000$.

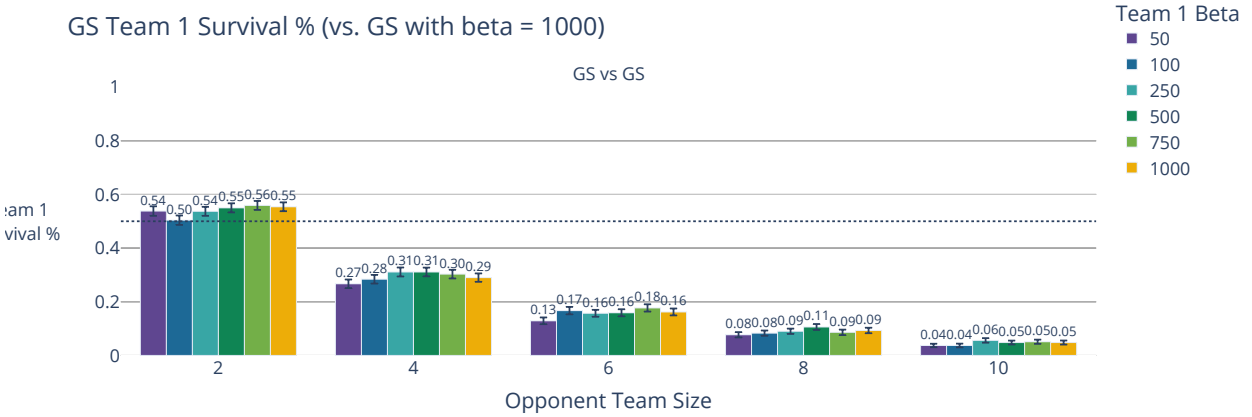


(c) Average opponent survival percentage for teams of two GS agents against a team of M DA agent with fixed weapon effectiveness of $\beta = 1000$.

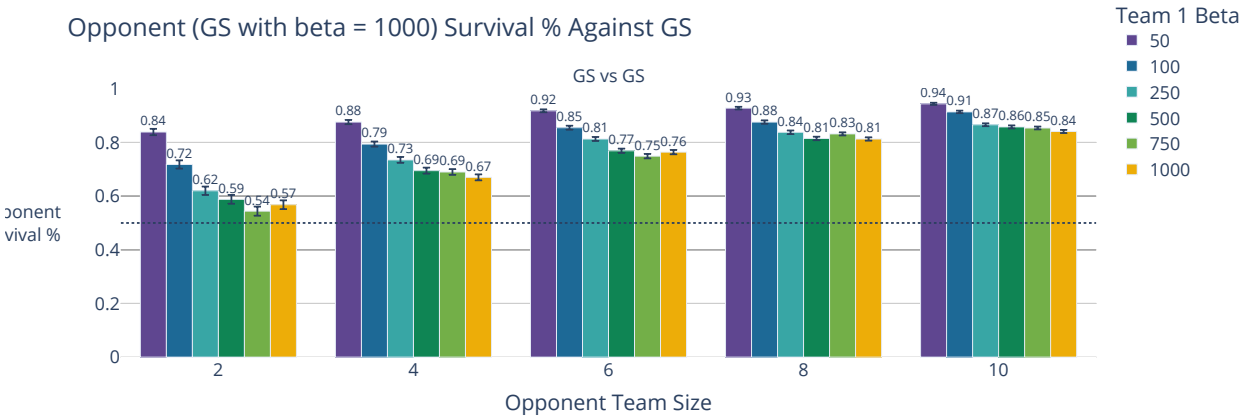
Figure 3.21: Average score, survival percentage, and opponent survival percentage for 2-vs.-M engagements with a DA protagonist team and with the opponent team having $\beta = 1000$.



(a) Average scores of teams of two GS agents against a team of M GS agent with fixed weapon effectiveness of $\beta = 1000$.



(b) Average survival percentage of teams of two GS agents against a team of M GS agent with fixed weapon effectiveness of $\beta = 1000$.



(c) Average opponent survival percentage for teams of two GS agents against a team of M GS agent with fixed weapon effectiveness of $\beta = 1000$.

Figure 3.22: Average score, survival percentage, and opponent survival percentage for 2-vs.-M engagements with a GS protagonist team and with the opponent team having $\beta = 1000$.

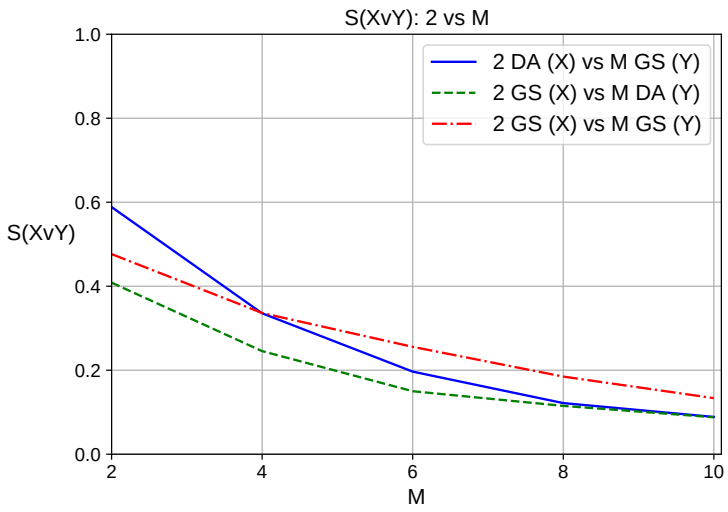


Figure 3.23: Average scores for two-vs.-M cases, with all $\beta = 100$.

high-quality weapons. Qualitative observation of several engagements shows that it is not simply the number of UAVs in the arena that causes DA to lose effectiveness. After the initial approach phase between a DA and a GS team in a large engagement, the engagement becomes a dense melee of tightly-turning UAVs, with DA agents spending more time attempting to achieve echelon form with their partners, finding their new partners as existing partners are attrited, and trying to set up their coordinated maneuvers than they did actually maneuvering to aim their fire. When a pair of DA agents begin a coordinated maneuver in such dense conditions, one or both members of the DA pair are often attrited by nearby GS agents before the maneuver is completed; thus, in dense conditions, DA's dependence upon maneuvers for aiming is to their disadvantage. The GS agents, in contrast, are unyielding in their targeting and do not give the DA agents the space or time they require to maneuver, leading me to fail to reject the null hypothesis that explicitly-coordinating maneuvers are more effective in larger engagements than non-explicitly-coordinating maneuvers.

When considering protagonist team performance in larger engagements with the antagonist team having $\beta = 50$ (Figures 3.11, 3.12, 3.19 and 3.20), note that there is a prioritization aspect to consider with respect to both own-team weapon quality and team size in a hypothetical situation in which one is selecting whether to equip one's UAVs with DA or GS. In Figures 3.11, 3.12, 3.19 and 3.20, the average survival percentages of the teams of N DA (Figure 3.12b) and the teams of

two DA (Figure 3.19b) are lower than those of the corresponding GS teams, and these DA survival percentages are more affected by the DA teams' weapon quality than are the DA teams' scores. Against opponents with high-quality weapons, in the N-vs.-N cases (see Figures 3.13b and 3.14b) and in the 2-vs.-M cases (see Figures 3.21b and 3.22b), the DA teams survive roughly as well as the GS teams for opponent team sizes of N = 2 and N = 4 when the protagonist teams are equipped with high-quality weapons. The team of two DA in Figure 3.21b is unable to maintain this survival percentage parity with the teams of two GS in Figures 3.21b and 3.22b when the opposing team size is six or greater, a conclusion also supported by the N = 10 survival percentages in the N-vs.-N cases seen in Figures 3.13b and 3.14b. As DA's behavior is constructed around employing coordinated maneuvers to aim at opponents precisely, its disadvantage in dense scenarios—particularly against tactics that are well-equipped to exploit force concentration in dense scenarios—is not unexpected. DA's maneuvers require one partner to evade the opponent aircraft to entice the opponent into a predictable position where the DA-evader's pursuer partner could achieve a brief moment of favorable force concentration—one where the pursuer can fire at the opponent and the opponent cannot fire at either the pursuer or evader. If the pursuer is unable to attrit the opponent in that brief firing opportunity window, one or both of the DA agents—usually the evader—are at risk of being fired upon by their escaped opponent. Despite the risks of employing these coordinating maneuvers that require non-hostile space and maneuvering time to target and fire at opponents, note that, in Figures 3.11c, 3.12c, 3.13c and 3.14c, the DA teams' ability to attrit opponents outstripped that of the corresponding GS teams with corresponding weapon effectiveness in all of the N-vs.-N engagements. In the 2-vs.-M engagements shown in Figures 3.19c and 3.20c, however, observe that the team of two DA outperforms the team of two GS for opponent team sizes between 2 and 8; the team of two DA are unable to attrit as many opponents as the team of two GS is in the 2-vs.-10 engagements for any team-of-two's weapon quality tested. The team of two DA's difficulty for opponent team sizes of 10 supports the claim that the DA tactical behavior's performance suffers in especially dense engagement conditions, whereas the GS behavior is much more capable in such dense scenarios. The GS agents simply do not need to coordinate aiming opportunities when they

find themselves in dense conditions, making them more resilient than DA agents in these scenarios. The opponent survival metrics for the 2-vs.-M experiments shown in Figures 3.21c and 3.22c in which opponents had weapon effectiveness of $\beta = 1000$, however, show the teams of two DA and two GS were approximately equally unable to attrit many opponent agents when facing teams of size M. As shown in Figures 3.21b and 3.22b, the team of two GS has better survival percentages against opponent teams of M $\beta = 1000$ GS agents than do their DA counterparts, suggesting that the DA pair's risky aiming maneuvers are especially problematic for them in these 2-vs.-M engagements. In such engagements, DA pairs simply do not have the non-hostile maneuvering space they need to aim at opponents, and so are more likely to be attrited than their GS counterparts, especially when their opponent team has high-quality weapons. The opponent's high weapon quality makes them better able to take advantage of brief snapshot opportunities against the maneuvering DA pair, especially if the maneuvering DA pair is not aiming at the opponent who takes advantage of one of these snapshot opportunities.

3.6 Limitations

DA's decline in performance in dense engagement scenarios highlights a limitation of this work alluded to in Section 3.2: neither the DA nor the GS tactical behavior can command increases in speed beyond v_{cruise}, and only DA agents can slow down (briefly, only to satisfy timing needs in specific maneuvers or maneuver setups) below a speed of v_{cruise}. In this chapter and Chapter 4, I choose to focus on how these tactics perform under this limitation, but in Chapter 6 discuss possible future work in which GS and DA are capable of modifying their speed more freely. Based on early versions of the DA tactical behavior, I postulate that, were both tactics to be equipped with more freedom in their velocity choices, such experiments would likely see far more effectiveness from the DA agents, particularly in their ability to isolate opponents to set up their aiming maneuvers. In the early development stages of DA, when the DA agents were tested with $v_{max,DA} > v_{cruise}$, DA was even more effective at leveraging its coordinated maneuvers than it is shown to be in this chapter. A DA pair that recognizes that it is in a situation in which the opponents near the

pair are too close to execute an effective bracket or sandwich enters a "flee" internal state, during which the pair turn away from the nearest opponents and fly at their maximum speed. DA agents with $v_{max,DA} > v_{cruise}$ will continue this (literal and figurative) flight until they are several more turn radii away from an opponent than they would need to bracket that opponent. They then turn around, line up in echelon formation, and bracket the opponent, who, if GS, likely either attempted to follow the DA pair or shifted its aim to another agent, both conditions under which set the DA pair up well for a bracket. In an effort to keep the GS tactical behavior simple, I did not add any capability for GS to adjust its speed, and so, to make the experiments in this chapter fair between the two teams, DA's $v_{max,DA}$ was set to be the same as v_{cruise}. Despite this limitation, DA is nonetheless quite effective in not only small engagements, but also as engagement size increases.

3.7 Contributions

In this chapter, I investigate two tactical behaviors for swarm-vs.-swarm aerial engagements between fixed-wing UAVs, and identify under what conditions each performs best. Both GS and DA are very effective tactical behaviors, but the coordinated maneuvers of DA allow DA teams to perform most strongly when the DA team has highly-effective weapons and is able to employ those weapons against isolated opponents—scenarios in which DA pairs have the space and time to maneuver without interruption. The decentralized GS behavior is more effective than DA in denser situations and is more resilient to being equipped with a poor-quality weapon, as its simple logic concentrates each team member's force upon the nearest threat in ways that allow GS agents to fire at their targets repeatedly without dependence upon any teammates' exact actions. Given these insights, in Chapter 4, I introduce and demonstrate a deep reinforcement learning scheme that trains agents to select when to select DA or GS or when to maneuver differently than either DA or GS would dictate. This approach, which fosters well-timed coordination between specific team members by taking paired agent state representations as inputs to the neural network which defines the protagonist team members' policy, performs more strongly according to the metric introduced in this chapter than either DA or GS alone.

CHAPTER 4
LEARNING TO LEVERAGE TACTICS

4.1 Motivation

In aerial combat scenarios such as those presented in Chapter 3, it is evident that there are differences between the ideal scenarios in which one would employ DA, GS, or perhaps another tactic altogether; thus, in this chapter, I present a novel algorithm that leverages deep reinforcement learning from pairs of agent states to decide which tactic each agent should utilize, with which teammate an agent should partner if DA is selected, and what an agent should do if neither DA nor GS are tactically favorable. This approach is trained and tested in the same simulation environment as is employed in the experiments for Chapter 3, albeit in a smaller ($1\,\mathrm{km}$-by-$1\,\mathrm{km}$) arena, with all agents having high weapon effectiveness $\beta = 1000$ (good-quality weaponry), and without allowing DA agents to start in echelon formation with their initial partner. I present this trained tactical behavior for usage in similar situations to those in which one might employ the far-off defense approaches given in the previous chapter.

4.2 Background

As discussed in Section 3.7, the UAVs on the protagonist team would benefit from being able to evaluate their own and their teammates' situational context in order to select when to perform GS, when (and with whom) to perform DA, and when to do something else, all for the team's overall tactical advantage. To this end, I turn to a classic RL algorithm, REINFORCE [98, 99], to equip agents to learn in what situations these action choices are appropriate. The algorithm itself is well-known and has been used to train neural networks in many applications; my primary theoretical contributions in this chapter are how the inputs to the agents' policy network are structured and how the outputs of the network are employed in action selection and agent partner selection.

4.2.1 Problem Formulation

Suppose the 2D aerial combat problem is an MDP [99], a five-tuple, as given in Equation (4.1).

$$\langle \mathbf{S}, \mathbf{A}, \mathbf{R}, T(s, a, s'), \gamma \rangle, \tag{4.1}$$

$\boldsymbol{s} = \bigtimes_{i=1}^{m} s_i$ is the joint state for the team of m homogeneous learner agents, each of which has individual state s_i. $\boldsymbol{S}$ is the set of all joint states. Likewise, $\boldsymbol{a} = \bigtimes_{i=1}^{m} a_i$ is the joint action for the m agents on the team, with a_i being agent i's action and $\boldsymbol{A}$ being the set of all joint actions. $\boldsymbol{R} = r_i(\boldsymbol{s}, \boldsymbol{a})$ is the reward given to all agents on the team for having joint state $\boldsymbol{s}$ and taking joint action $\boldsymbol{a}$, and γ is the discount factor.

4.2.2 REINFORCE

In the REINFORCE [98, 99] policy gradient RL algorithm, the goal is to learn a policy, in this case $\pi\left(a_i|\boldsymbol{s}, s_i, \boldsymbol{\theta}\right)$, where $\boldsymbol{\theta}$ is a parameter vector by which the policy is differentiable. In the formulation introduced above, as the scenarios begin with a number of agents and agents are attrited as time progresses, episodic REINFORCE [98, 99], where agents operate in a short beginning-to-end scenario, learn from the resulting trajectories, and then begin anew in another short scenario, but operating under the policy updated by the most recent training epoch.

The aim of a policy gradient method is to to maximize a performance measure, $J(\theta)$, via gradient ascent, as in Equation (4.2).

$$\boldsymbol{\theta}_{k+1} \leftarrow \boldsymbol{\theta}_k + \alpha \nabla \hat{J}\left(\boldsymbol{\theta}_k\right) \tag{4.2}$$

In this chapter, k is the training epoch, α is the learning rate, and $\hat{J}(\boldsymbol{\theta}_k)$ is the estimation of the gradient of the performance measure with respect to $\boldsymbol{\theta}_k$ [99]. In episodic REINFORCE, the gradient estimate $\nabla J(\theta) \sim \mathbb{E}[\nabla \hat{J}(\theta)]$ is obtained by recording the joint states encountered, joint actions executed, and rewards received in an episode, then performing the gradient ascent step given in Equation (4.3) and updating the parameter vector $\boldsymbol{\theta}$ as in Equation (4.4).

$$\nabla J\left(\boldsymbol{\theta}_k\right) \propto \mathbb{E}_{\pi, i \in m}\left[\sum_{a_i} G_t \frac{\nabla \pi\left(a_{it} \mid \left(s_{it}, \boldsymbol{s}_t\right), \boldsymbol{\theta}_k\right)}{\pi\left(a_{it} \mid \left(s_{it}, \boldsymbol{s}_t\right), \boldsymbol{\theta}_k\right)}\right] \tag{4.3}$$

$$\boldsymbol{\theta}_{k+1} \leftarrow \boldsymbol{\theta}_k + \alpha \mathbb{E}_{i \in m} G_t \frac{\nabla \pi\left(a_{it} \mid \left(s_{it}, \boldsymbol{s}_t\right) \boldsymbol{\theta}_k\right)}{\pi\left(a_{it} \mid \left(s_{it}, \boldsymbol{s}_t\right), \boldsymbol{\theta}_k\right)} \tag{4.4}$$

Throughout episode k, the training framework stores the joint states, joint actions, and rewards agents encounter. Upon conclusion of epoch k's episode, the framework then updates the gradient estimate of the performance measure from the joint states, joint actions, and rewards stored in episode k and updates the parameter vector accordingly, which updates the policy. Then, the framework starts episode $k+1$ for epoch $k+1$'s training with the agents operating under the updated policy, $\pi(a_i|s_i, \boldsymbol{\theta}_{k+1})$.

4.2.3 Entropy

In training off-policy RL methods such as Deep Q-Networks (DQN), ϵ-greedy action selection [99, 100] is frequently employed to encourage agents to explore a variety of action choices in a given state to prevent the learned policy that will be employed at test time from converging to a local optimum instead of the global optimum. Policy gradient methods such as REINFORCE, however, are classed as *on-policy* methods, meaning that the policy being utilized during training is exactly that which will ultimately be employed at test time; thus, training with an alternative policy (such as that implicitly constructed when leveraging ϵ-greedy action selection method during training episodes) is not permissible [99]. An on-policy-friendly method for encouraging exploration of actions is adding a term, given in Equation (4.9), to the performance measure objective function that accounts for entropy in agent action choices.

$$H\left(s_{it}\right) = -\sum_{a_t \in A_t} \pi\left(a_{it} \mid s_{it}, \boldsymbol{s}_t\right) \log \pi\left(a_{it} \mid s_{it}, \boldsymbol{s}_t\right) \tag{4.5}$$

While an entropy term encourages agents to explore different actions during training than it might visit under a deterministic policy-in-training, as REINFORCE is an on-policy method, the agent's

policy at test time will be somewhat stochastic. Some stochasticity in a policy is not always detrimental [99], however; in terms of tactical decisions, an agent with no stochasticity in its actions is likely to see its opponents exploit its determinism!

In the performance measure gradient estimate utilized for training in this chapter, the entropy term in Equation (4.9) is multiplied by a small coefficient hyperparameter, λ, to ensure that the entropy term does not overpower the other terms of the performance measure gradient estimate. This hyperparameter has a minimum value $\lambda = 0.01$ and is increased by a factor of 0.005 (up to a maximum of $\lambda = 0.02$) after every epoch in which one action is selected less than 10% of all of the action selections made on the learner team in that epoch. These values for the entropy coefficient provided sufficient encouragement during training for agents to explore their action choices.

4.3 Procedure

As mentioned in Section 3.7, both training and testing are conducted in the same simulation environment and with the same type of fixed-wing aircraft as were used in Chapter 3, albeit in a smaller engagement arena (in this chapter, $1\,\mathrm{km}$-by-$1\,\mathrm{km}$, while in Chapter 3, the arena was $10\,\mathrm{km}$-by-$10\,\mathrm{km}$) and with minor motion model and controller parameter changes to fine-tune the performance of both baseline tactics. In this section, I present the procedure with which the agents are trained, define the agents' state space, detail the procedure agents employ to select actions based on the outputs of the policy network, define the policy network's architecture, and detail the measures with which the training framework ensures consistent agent initialization across cases during training and testing.

4.3.1 Training Details

The training framework employs REINFORCE to train a neural network that takes as input pairs of agent states and outputs which action the first of the two agents whose states were passed in should take with respect to the context of the state pair, as well as the mean of the yaw rate the agent should use if it chooses to execute neither DA nor GS. The network is trained for 1000 epochs,

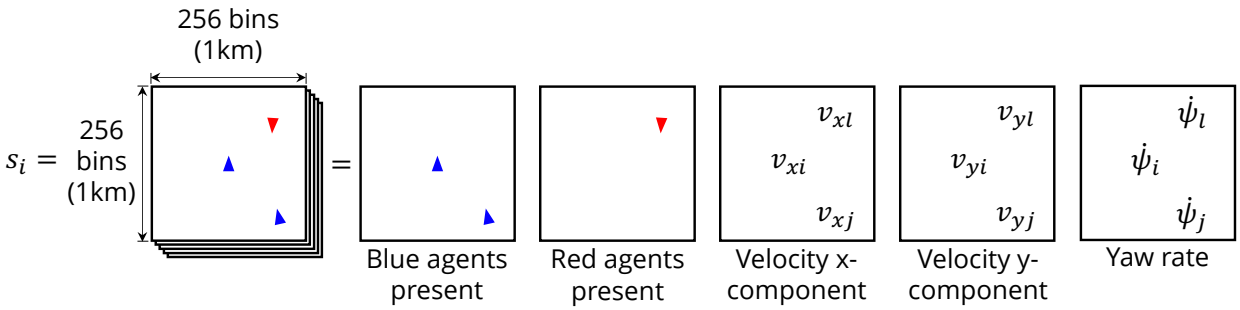


Figure 4.1: The channels in the state representation of agent i.

Table 4.1: Channels of an Agent's State

Channel	**Quantity**
0	Positions of self and teammates*
1	Positions of members of opponent team*
2	Sum of x-components of velocity of agents in each bin*
3	Sum of y-components of velocity of agents in each bin*
4	Sum of yaw rates of agents in each bin

**Relative to observing agent's body-fixed reference frame*

with each epoch being trained on the most recent engagement's joint-state-joint-action pairs and the reward received by all teammates at the end of the engagement. The teams train against teams of either two or four agents that employ either DA or GS for the entirety of training. A training episode may last up to 500 timesteps; with each timestep being $0.1\,\mathrm{s}$ long, a training episode may last $50\,\mathrm{s}$ at most.

4.3.2 State Representation

The state representation of agent i is defined as a multi-channeled ego-centric discretization of the $1{,}000\,\mathrm{m}$ square around agent i. The channels in this discretization are illustrated in Figure 4.1 and defined in Table 4.1. If multiple agents are present in the same bin of the discretization when the simulation assembles an agent's state, the co-binned agents' relevant values are summed[1].

[1] `https://github.com/numpy/numpy/blob/v1.22.0/numpy/lib/twodim_base.py`; see lines 689-693

These discretized state representations are generated for each alive agent on the protagonist team. As stated in Section 4.2.1, the joint state of the team of learner agents is $\boldsymbol{s} = \bigtimes_{i=1}^{m} s_i$. I discuss in Section 4.3.4 how each learner agent leverages the joint state of its team along with its own state to make action decisions.

4.3.3 Policy Network Architecture

The agents trained and tested in the work described in this chapter, Paired Situational-Context Evaluator (PSCE) agents, operate under the policy of a neural network, the architecture of which is depicted in Figure 4.2. The policy network architecture is depicted in Figure 4.2; this network begins with five convolutional layers with ReLU [101] activation, then splits into two branches. The first branch, the output of which the action-selection algorithm utilizes for selection of an agent's discrete action, consists of of two fully-connected layers with ReLU activation and ends with a softmax. The second branch, which provides a continuous output leveraged in selecting the agent's yaw rate if the agent selects Maneuver (MN), is made up of three fully-connected layers, the first two of which utilize ReLU activation and the last of which employs linear activation, resulting in a scalar output that can be positive or negative. The policy network weights are initialized via Xavier Uniform initialization with a gain of 1.0[2]. At each timestep, a PSCE agent may choose to execute a single timestep of either DA (with a specific partner, explained below), GS, or MN. If MN is selected, the agent's yaw rate setpoint is assigned to be the yaw rate drawn from a distribution defined by the output of the continuous branch of the policy network. The output of these two branches is utilized in the action selection procedure detailed in Section 4.3.4.

[2] `https://pytorch.org/docs/stable/nn.init.html#torch.nn.init.xavier_uniform_`

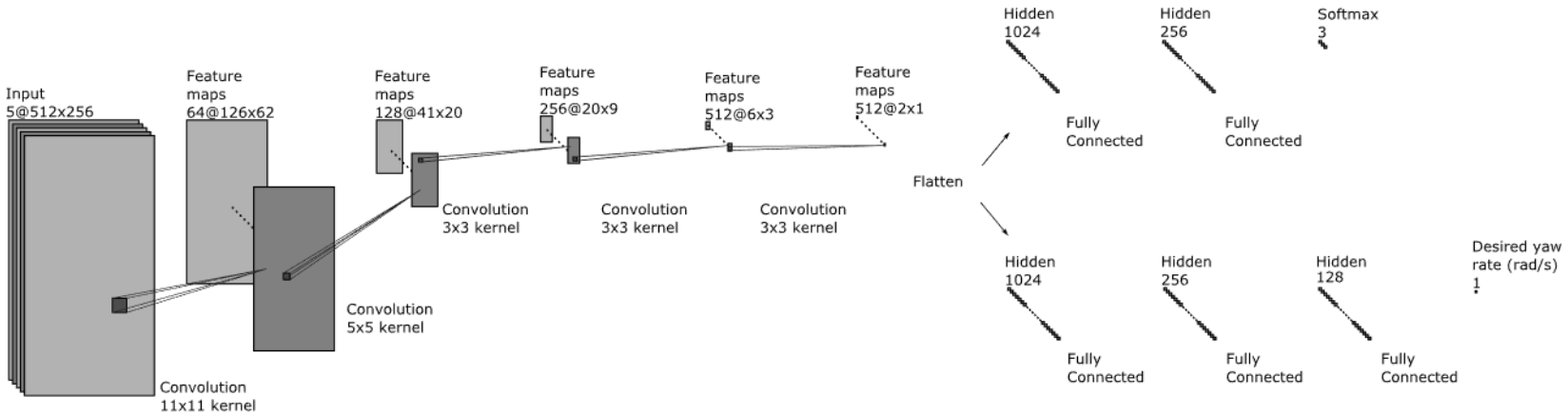
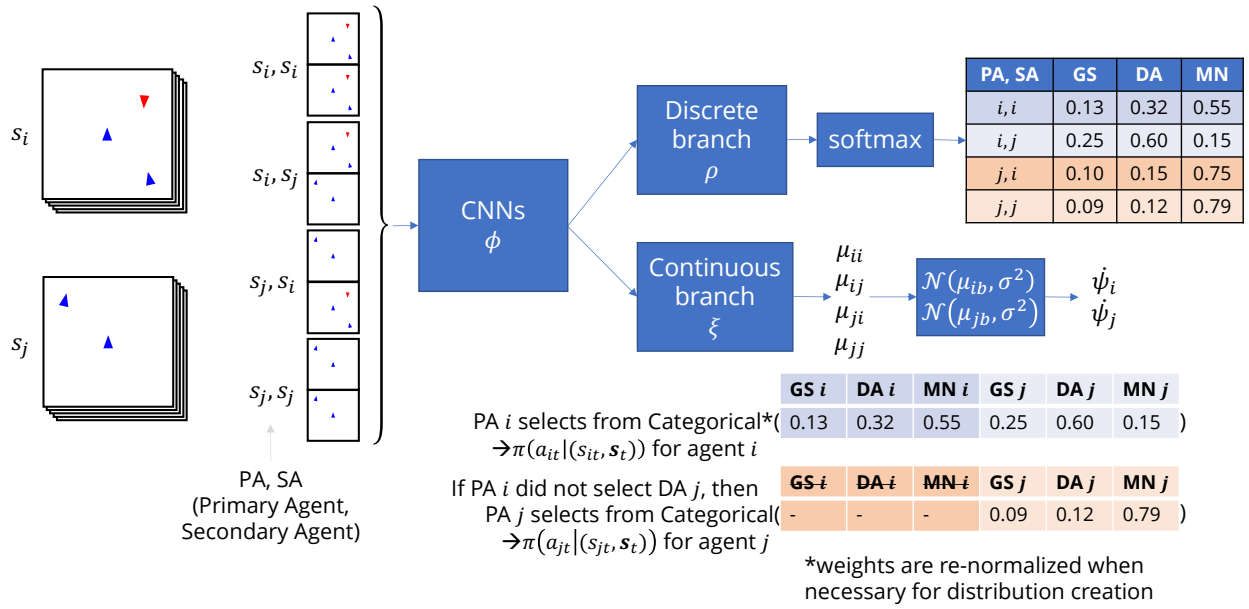


Figure 4.2: The agents' actions are selected based on the outputs of this neural network's evaluation of pairs of the state of the agent whose action is being decided and states of its possible partners.



PA, SA	GS	DA	MN
i,i	0.13	0.32	0.55
i,j	0.25	0.60	0.15
j,i	0.10	0.15	0.75
j,j	0.09	0.12	0.79

GS i	DA i	MN i	GS j	DA j	MN j
0.13	0.32	0.55	0.25	0.60	0.15

~~GS i~~	~~DA i~~	~~MN i~~	GS j	DA j	MN j
-	-	-	0.09	0.12	0.79

Figure 4.3: Illustration demonstrating PSCE agent action selection for a team containing two agents, i and j.

4.3.4 Action Selection

The action selection procedure for agent i begins with all possible pairings of agent states between agent i's state and all alive teammate agents with un-selected actions; Figure 4.3 shows this for a team containing two agents, agents i and j.

Consider a team with two agents, i and j, whose actions are not yet assigned for the current timestep. Action selection for agents i and j starts by with the computation of the joint state—the Cartesian product of the states of all of the agents on the team. The Cartesian product provides all possible pairings of teammates' state representations. These state representation pairings are passed through the policy network to generate the distributions from which each agent will select its action at this timestep. Note that own-state pairings are included in the inputs to the network; this allows agents whose teammates have already all selected actions—or all been attritted—to continue to leverage the policy network for action selection. The agent whose state appears first in a two-agent-state pairing is the primary agent of the pairing. The primary agent of a pairing selects its action based on all outputs from the policy network in which it is the primary agent.

Agents whose states are paired with that of the primary agent to select the primary agent's action are referred to as secondary agents. Each timestep, all agents may be primary agents if they are not selected to perform DA with another teammate before their turn for action selection arrives.

The outputs of the softmax branch are collected and organized according to the state pairing that generated them, as shown in Figure 4.3. The agents, in order of agent ID number[3], select their action—GS, MN, or DA (with the secondary agent in the corresponding state pairing as the DA partner)—from a categorical distribution defined by the outputs from the softmax branch (re-normalized so as to create a valid probability distribution). If an agent's action has already been selected, whether by it being primary agent already for the current timestep or by being selected to be the DA partner of another teammate, that agent does not undergo the primary-agent action selection procedure, nor are softmax outputs generated from it being a secondary agent included in other primary agents' action selection distributions; this is illustrated in Figure 4.3 by the crossed-out column headings for agent j. The softmax output generated for the selected action from the state pairing that resulted in that action being selected is $\pi(a_{it} \mid s_{it}, \boldsymbol{s}_t)$ for that agent at timestep t. If an agent selects MN from the categorical distribution over its action choices, it sets its yaw rate setpoint for the next timestep to $\dot{\psi}_i$, where $\dot{\psi}_i$ is sampled from $\mathcal{N}(\mu_{ib}, \sigma^2)$. For this yaw rate distribution, $\sigma = 0.1\,\mathrm{rad/s}$ and μ_{ib} is the output yaw rate mean in $\mathrm{rad/s}$ from the continuous branch of the policy network, with b corresponding to the ID of the agent whose state, when paired with that of agent i, generated the outputs from which the MN action was sampled. Equation (4.6), a component of the performance measure used to train the policy network, comes from the sampled yaw rate and the Probability Density Function (PDF) of the normal distribution from which the yaw rate is sampled.

$$\pi\left(\dot{\Psi}_{it} = \dot{\psi}_{it} \mid s_{it}, \boldsymbol{s}_t, a_{it} = a_{it}^{MN}\right) \tag{4.6}$$

[3]Any defined agent ordering is sufficient.

4.3.5 Performance Measure Gradient Estimate Components

The estimate of the gradient of the performance measure with which the neural network defined in Figure 4.2 is trained is given in Equation (4.7).

$$\nabla \hat{J}(\rho, \phi, \xi) = \frac{1}{m} \sum_{i \in m} \frac{1}{T} \sum_{t=t_0}^{T} \Bigg[G(t) \Bigg[\nabla_{\rho,\phi} \log \left(\pi \left(a_{it} \mid s_{it}, \boldsymbol{s}_t \right) \right) + M_{it} \nabla_{\xi,\phi} \log \left(\pi \left(\dot{\Psi}_{it} = \dot{\psi}_{it} \mid s_{it}, \boldsymbol{s}_t, a_{it} = a_{it}^{MN} \right) \right) \Bigg] + \lambda \nabla_{\phi,\rho} H \left(s_{it} \right) \Bigg] \tag{4.7}$$

In Equation (4.7), Equation (4.8) is the return, and Equation (4.9) is the entropy of agent i's action at timestep t. $\lambda = 0.1$ is a hyperparameter.

$$G(t) = \sum_{t'=t}^{\infty} \left(\gamma^{t'-t} \boldsymbol{R}_{t'} \right) \tag{4.8}$$

$$H \left(s_{it} \right) = - \sum_{a_t \in A_t} \pi \left(a_{it} \mid s_{it}, s_t \right) \log \pi \left(a_{it} \mid s_{it}, s_t \right) \tag{4.9}$$

The reward signal the agents receive follows the scheme given in Equation (4.10), and is provided to all agents on the learner team in the final timestep of the simulation. In Equation (4.10), T is the final timestep, $N_{R,T}$ is the final number of agents on the antagonist team alive in the final timestep, and $N_{B,T}$ is the final number of blue-team agents—the learner agents—alive in the final timestep.

$$r_{i,T} = \begin{cases} -N_{R,T} & \text{if any red-team alive} \\ +N_{B,T} & \text{otherwise} \end{cases} \tag{4.10}$$

This reward signal is biased towards rewarding the agents for focusing more on opponent-team attrition than on own-team survival. While the primary metric I leverage to examine the results of training, the score introduced in Equation (3.1), purposefully considers own-team survival to be of equal importance as opponent-team attrition, when agents were trained with reward signals with such equal weighting, their scores and their survival percentages indicated that the agents

were primarily learning to run away from the opponents, rather than learn to counter them. The reward structure, Equation (4.10), results in agents that show a willingness to risk some own-team attrition in order to achieve opponent-team attrition, but with sufficient self-preservation so as to still achieve good own-team survival overall.

4.3.6 Agent Initial Position Management

In an effort to enforce consistency between the trained teams' experiences, the simulator is seeded with the next integer from a sequence of pre-defined seeds at the beginning of each training engagement such that the initial positions of the agents in training episode k are consistent between PSCE (4-vs.-4 trained vs. GS) and PSCE (4-vs.-4 trained vs. DA) and between PSCE (2-vs.-2 trained vs. GS) and PSCE (2-vs.-2 trained vs. DA) for all k in the number of training epochs. This seed sequencing guarantees that the 4-vs.-4 training engagements have two agents on each team in the same positions that they are in in the 2-vs.-2 training engagements, so any difference in what occurs in training episode k of a 4-vs.-4 engagement versus what happens in the kth training engagement of the 2-vs.-2 training procedure must be due to either differences in what the policies-in-training have learned in their respective $k-1$ training epochs, the difference in the number of agents per team, or the team against which the learner agents are training, not due to some agents starting in different locations between episode k's 2-vs.-2 engagement initialization and episode k's 4-vs.-4 engagement initialization. To ensure that neither team begins with a significant positional advantage, the agents of one team are initialized at the same x-coordinate locations and mirrored y-coordinate locations as the other team's positions. Note that the seed sequencing and initial position mirroring are also implemented in the simulation seeds employed in seeding the test set experiments. The set of test set experiment seeds and training set seeds do not intersect.

To equip the learner agents to start on either side of the engagement arena, every two training engagements, the learner team's starting positions are swapped with the starting positions of the opponent team. At test time, for each matchup of a specific trained team to a specific opponent in a specific engagement size, the test set includes both the scenario with the trained team starting on

the left half of the arena as well as the scenario with the trained team starting on the right half of the arena. The results for these flipped cases are averaged together in the plots shown in Section 4.5.

4.4 Evaluation Metrics

The metrics I present for the training process include the reward received by the agents-in-training, their score (as computed by Equation (3.1)), and the percent of the learner team that survives each training episode. I evaluate success of trained teams in a test environment primarily via the trained team's score, as computed by Equation (3.1). As in Chapter 3, I also examine the percent of each team that survives each engagement to obtain a more comprehensive picture of how the protagonist teams achieve their scores.

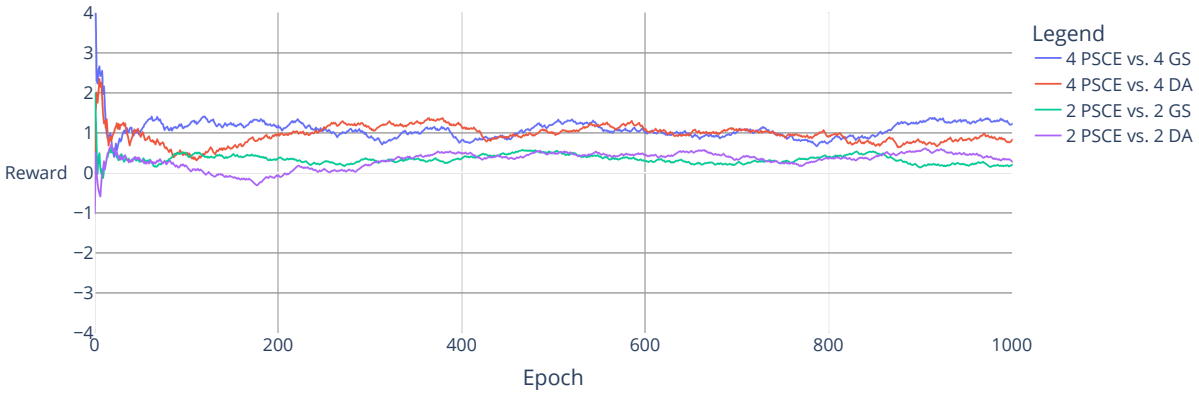
4.5 Results

4.5.1 Training

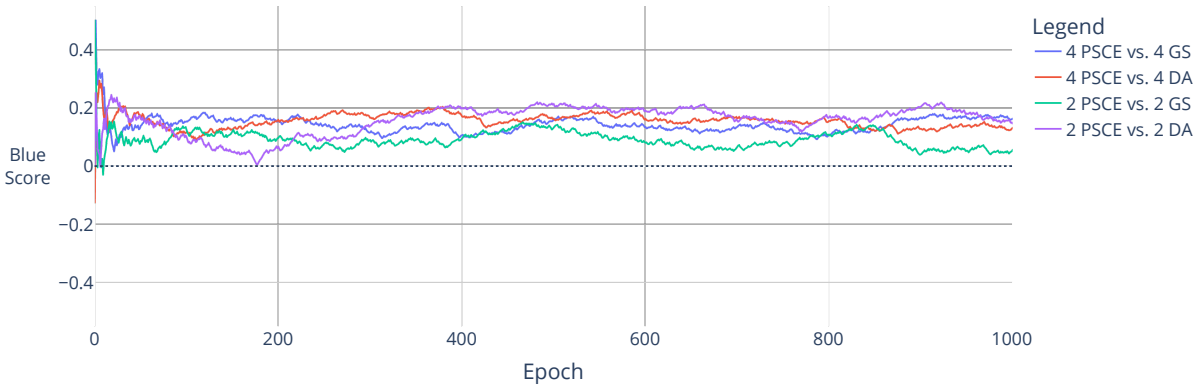
Agents trained using the training scheme presented in this chapter do not show a clear increase in reward during training, as shown in Figure 4.4a, nor do they show a clear increase in score or survival percentage, as shown in Figure 4.4b or Figure 4.4c, respectively. The testing results (Section 4.5.2), however, show that the learner agents gain a noticeable benefit from training and score higher than an untrained team in most scenarios.

4.5.2 Testing

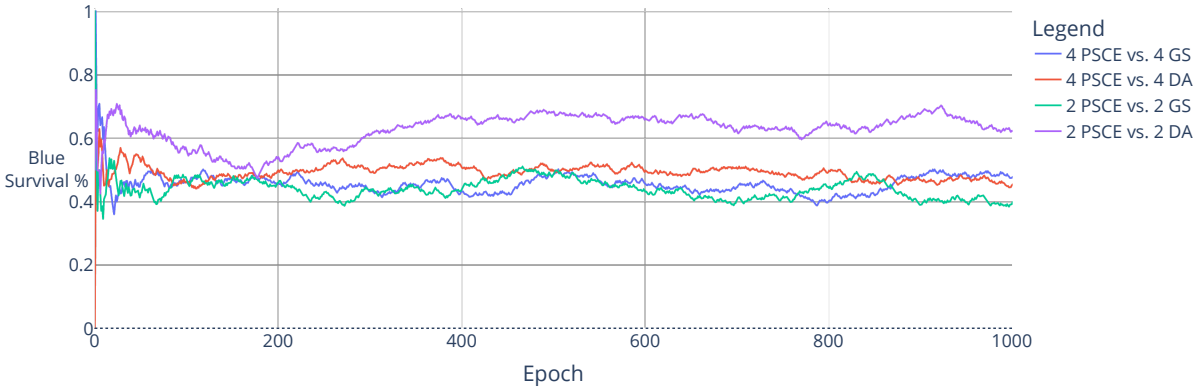
While the lack of decisive reward increase during the training of the PSCE agents does not suggest that the trained agents learned much about good tactic selection, the results from the testing engagements show that the agents learned how to select tactics and partners better than an untrained team does, and that the policy learned is effective in engagements of sizes other than those in which the agents were trained.



(a) Smoothed reward received by PSCE agents during training



(b) Smoothed scores of PSCE agents during training



(c) Smoothed survival percentage of teams of PSCE agents during training

Figure 4.4: These plots show the smoothed rewards, scores, and survival percentages for each of the PSCE agent teams during training. The networks employed during the testing process are trained for 1000 epochs. Smoothing was performed using a procedure adapted from TensorBoard's plot smoothing function [102] with a smoothing weight of 0.99.

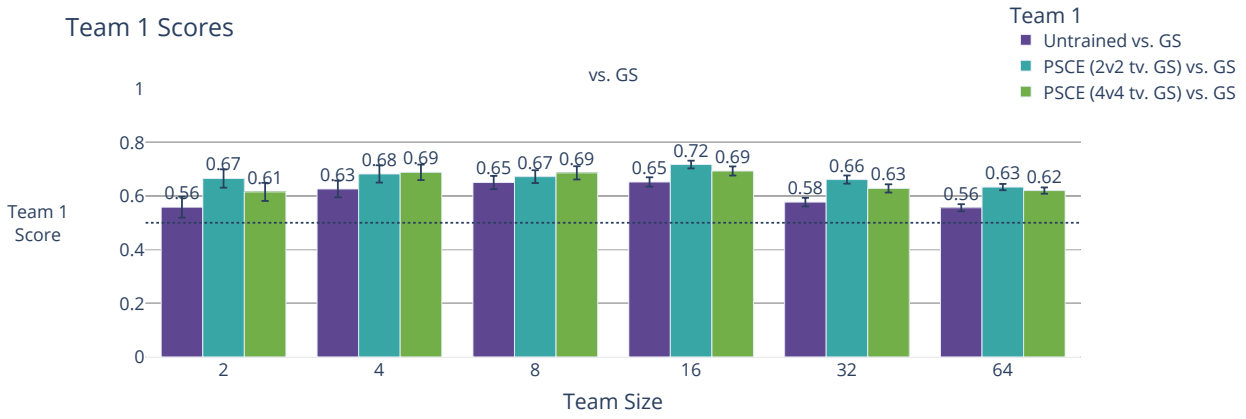
Performance Against Training Opponent

Teams of GS agents, which were difficult to outscore in the densest of the experiments discussed in Chapter 3, are outscored by both the untrained and trained teams, indicating that even the simple strategy of randomly switching between tactics has some merit. From Figure 4.5a, one can see that the untrained PSCE agents are outperformed with respect to the protagonist team's score by PSCE agents trained in both 4-vs.-4 and 2-vs.-2 against GS. The largest difference in performance between untrained and trained agents here is in own-team survival for 16-vs.-16 engagements; the PSCE agents trained in 2-vs.-2 vs. GS have an average survival percentage of 45%, which is a large increase over the untrained agents' survival percentage of 36%. Both trained teams are very capable of attriting large portions of the GS opponent teams, with generally increasing opponent attrition as training team size increases.

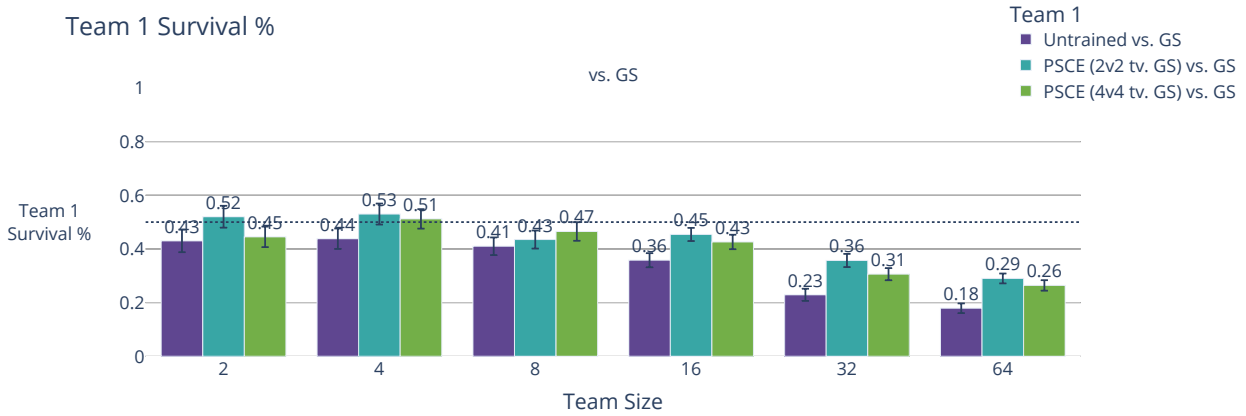
The results for PSCE teams trained in either 4-vs.-4 or 2-vs.-2 engagements against DA in engagements against DA, shown in Figure 4.6, show a general trend of overall lower scores and survival percentages than seen in Figure 4.5. Note that the PSCE teams trained in 2-vs.-2 against DA outscore the PSCE agents trained in 4-vs.-4 engagements against DA in both the 2-vs.-2 and 4-vs.-4 engagements by attriting more opponents. It is feasible that, as the PSCE (2v2 tv. DA) agents were trained in less-dense engagements, they were able to be more effective in these less-dense engagements than the PSCE (4v4 tv. DA) teams.

Performance Scaling With Team Size

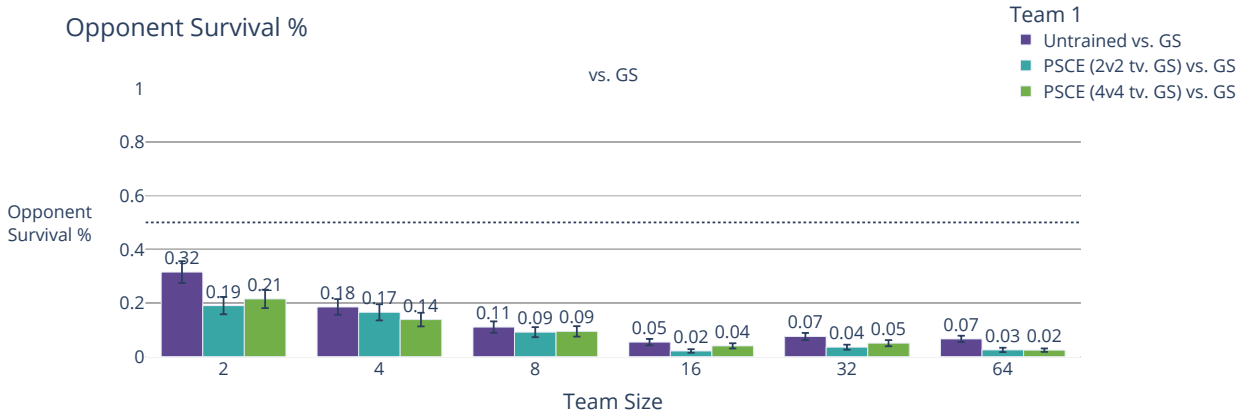
In the results discussed so far, it is apparent that the trained PSCE teams learned general good tactics that allow them to outscore, out-survive, and out-attrit untrained agents as well as their engagement opponents. Recall from Chapter 2 that one of my aims in this document is to examine how the trained agents we examine in this chapter are effective in scaling from small engagements to large engagements. In the smaller, less-dense engagements, as noted in Appendix A, agents trained in 2-vs.-2 engagements against either DA or GS outperformed the teams that trained in 4-vs.-4 engagements in both 2-vs.-2 and 4-vs.-4 engagements against either DA or GS, though their



(a) Average score of untrained, trained teams in engagements against teams of GS in N-vs.-N engagements $\forall$ N $\in$ 2, 4, 8, 16, 32, 64. Trained teams were trained against GS in either 2-vs.-2 or 4-vs.-4 engagements.

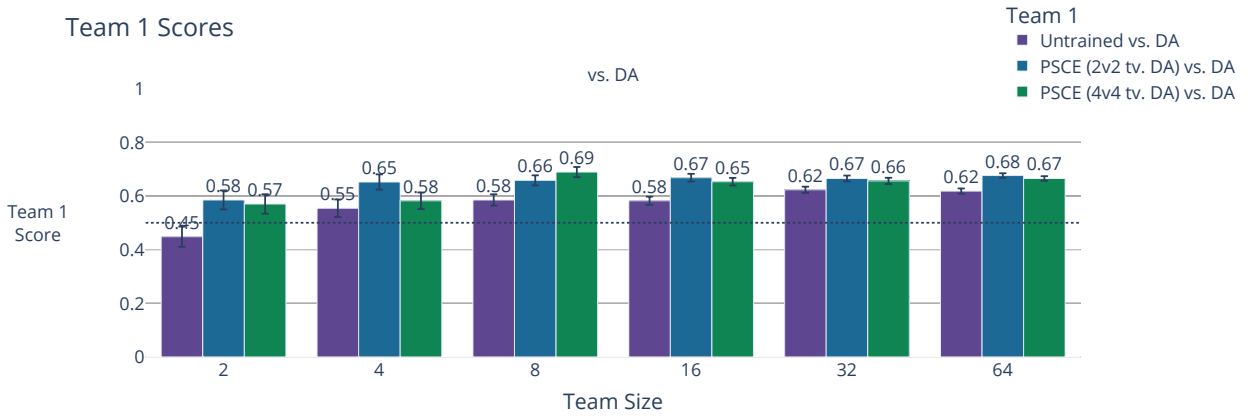


(b) Average percentage of untrained, trained teams that survive in engagements against teams of GS in N-vs.-N engagements $\forall$ N $\in$ 2, 4, 8, 16, 32, 64. Trained teams were trained against GS in either 2-vs.-2 or 4-vs.-4 engagements..

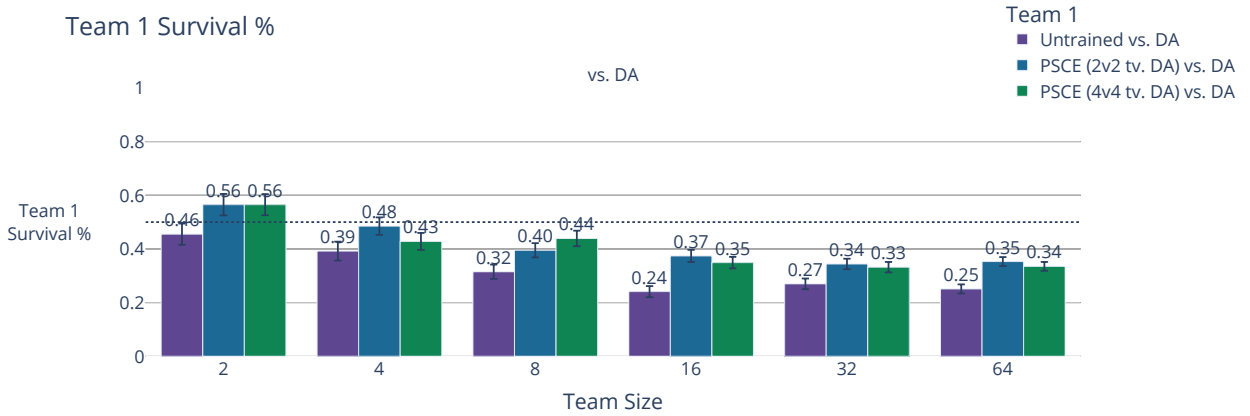


(c) Average percentage of GS opponents surviving against untrained, trained PSCE teams in N-vs.-N engagements $\forall$ N $\in$ 2, 4, 8, 16, 32, 64. Trained teams were trained against GS in either 2-vs.-2 or 4-vs.-4 engagements..

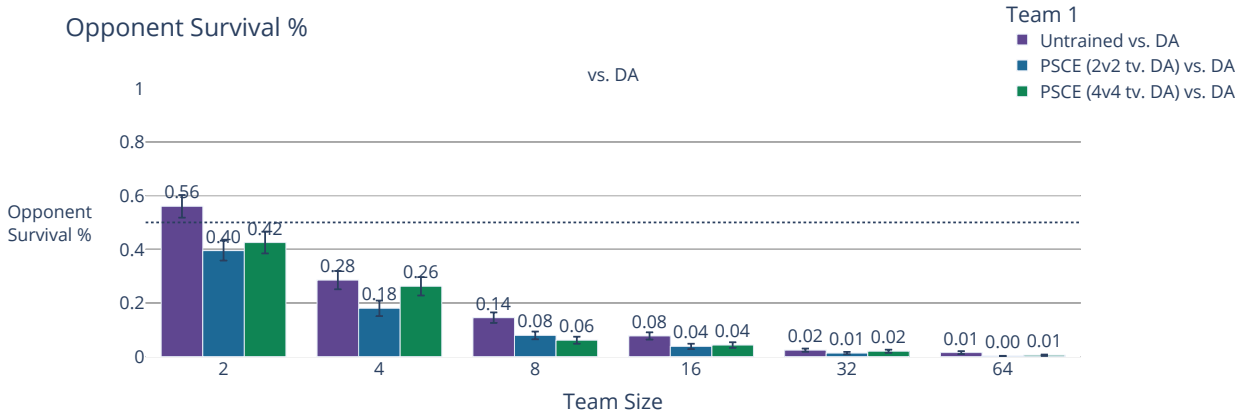
Figure 4.5: These plots show the scores, survival percentages, and opponent survival percentages of PSCE agents, with the training (or lack thereof) denoted in the legend, in engagements against GS teams. The untrained agents score more poorly than the trained agents in a number of scenarios, especially with increasing engagement size, in contrast to how DA's performance declined in more dense engagements in Chapter 3.



(a) Average score of untrained, trained teams in engagements against teams of DA in N-vs.-N engagements $\forall$ N $\in$ 2, 4, 8, 16, 32, 64. Trained teams were trained against DA in either 2-vs.-2 or 4-vs.-4 engagements.



(b) Average percentage of untrained, trained teams that survive in engagements against teams of DA in N-vs.-N engagements $\forall$ N $\in$ 2, 4, 8, 16, 32, 64. Trained teams were trained against DA.



(c) Average percentage of DA opponents surviving against untrained, trained PSCE teams in N-vs.-N engagements $\forall$ N $\in$ 2, 4, 8, 16, 32, 64. Trained teams were trained against DA.

Figure 4.6: These plots show the scores, survival percentages, and opponent survival percentages of PSCE agents, with the training (or lack thereof) denoted in the legend, in engagements against DA teams. The untrained agents score more poorly than the trained agents in most cases, but the scores and protagonist team survival metrics are overall lower than are those for the PSCE agents trained against GS in engagements against GS shown in Figure 4.5.

performance boost in the 4-vs.-4 engagements over the 4-vs.-4-trained teams was not as dramatic as it was in the 2-vs.-2 engagements. In general, however, as engagement size increases, *all* trained teams see a general trend of increasing score, albeit with less-dramatic increases as team size reaches the much larger 32-vs.-32 and 64-vs.-64 cases. In the engagements against DA, however, the protagonist team's survival (in Appendix A, Figure A.2b) decreases as the engagement size increases. As a DA agent without a partner acts as a GS agent until it can find a partner, however, it is possible that the agents on the DA teams in the especially dense engagements, while suffering attrition, were forced to act as GS agents often enough to accidentally operate in a manner similar to PSCE, frequently switching between tactics. (PSCE agents—especially trained PSCE agents—are still generally somewhat more effective than DA, however, as they have learned *when* these tactic switches are appropriate.) In comparing the opponent survival plots between the engagements between trained teams and DA or GS (in Appendix A, Figures A.1c and A.2c, respectively), it is clear that, as engagement size increases, the DA teams are more effective at surviving the onslaught of the trained teams than are the GS teams. With DA's tendency to default to GS when unpartnered, and considering the much smaller arena used in these experiments, it is possible that what makes DA more difficult to attrit and outscore in the experiments in this chapter is also what I believe is helping the PSCE agents perform well—alternating between GS and the initial phase of the DA state machine. In the DA maneuver selection state machine, before a DA pair ever evaluates whether bracketing or sandwiching an opponent is feasible, the agents prioritize achieving echelon form (lining up wings abreast within their partnership 1-2 turn radii apart), even temporarily slowing down slightly if necessary to assist in lining up. The brief attempts to achieve echelon form with their temporary partner, combined with short stretches of aiming at the nearest opponent with GS, provide the benefit of GS's exploitation of force concentration with DA's more position-focused maneuvering to make these DA teams caught in very dense scenarios difficult to best. Trained PSCE agents benefit from this tactic switching even more than the DA agents do with their tactic switching out of necessity. Tactic switching for PSCE agents is more informed than for either DA or untrained agents, all due to the evaluation of the policy network guiding the selection of when

to choose GS and when to select DA. Their switching policy makes trained PSCE agents *more* effective at achieving tactically-advantageous force concentration against not only the most imminently-threatening opponents, but also against the adversaries providing backup to the foremost enemy agents before these additional enemies can inflict unnecessary losses upon the protagonist team's agents. As seen in Figures 4.5 and 4.6, and in Appendix A's Figures A.1c and A.2c, the trained PSCE teams are effective, and more effective than untrained teams. Thus, I postulate that the training scheme detailed in this chapter is effective in training agents to switch between GS and DA's echelon-finding sequence in such a way that concentrates the PSCE team's force more effectively than a GS team or DA team can. I discuss this hypothesis further in Section 4.6.

4.6 Discussion

The PSCE agents are trained by means of REINFORCE, a policy-gradient reinforcement learning algorithm [98, 99]. This work does not seek to compare deep RL methods against one another in the context of aerial combat; rather, I aim to show that, by structuring the inputs to the network to evaluate action choices in terms of agent pairings, and utilizing the output from several evaluations of agent pairings to select the action for one agent in the context of its teammates' states, PSCE agents are able to learn to perform well when tested against teams of DA and GS.

Interestingly, the trained PSCE agents performed well against not only opponents operating under the tactical behavior against which the PSCE agents trained, but also against opponents against which they did not train. This suggests that the PSCE agents did not only learn how to select appropriate tactics and partnerships against the particular opponent battle style they trained against, but also that they learned how to make good tactical choices in general.

In preparing for these experiments, I expected PSCE agents who selected DA to switch actions or partners before they could complete an entire DA maneuver with a selected partner, which, from qualitative observation of a number of testing engagements, was borne out in testing. With the PSCE agents re-evaluating their action choices every timestep, selecting the DA action option with the same partner for the few dozen timesteps necessary to complete an entire bracket or

sandwich maneuver is highly unlikely. Thus, the results discussed in this document do not evaluate how adept PSCE is at selecting when performing a bracket or a sandwich is most appropriate. Instead, as postulated in Section 4.5, it is the switching between DA—and the brief attempt of the newly-minted DA pair to achieve echelon formation—and GS that lines the PSCE agents (and DA agents in especially small, dense engagements) up in a way that grants them favorable force concentration over a broad section of the opposing force. This advantageous force concentration is especially evident in Figure 4.7, where the blue team (PSCE (4-vs.-4 trained vs. GS), on the left) is countering a team of GS (red team, on the right). The GS agents are maneuvering via proportional navigation to aim at the nearest PSCE agent, and their aim angles are shown with the red lines that intersect the red aircrafts' centroids. The PSCE agents, whose aim angles are show with blue lines, are aimed much more broadly as a group, but with the group's overall aim centered roughly on the closest opponent threat. As the simulation shown in Figure 4.7 continues, the PSCE agents lose some of their frontmost team members, but in the process, attrit not only the foremost GS threats, but also many of its teammates farther back in the GS swarm, preventing those farther-away opponents from becoming more imminent threats. While very different in scenario and engagement style, this spread of force concentration against not only the most imminent threats, but also against other opponent-team members is similar in prioritization to the higher-opponent-threat maximizing guard schedules that I introduce in Section 5.4.

4.7 Contributions

In this chapter, I demonstrated a deep-RL training scheme that utilizes cleverly-arranged inputs to a neural network and unique processing of the outputs generated from multiple passes through the network for each agent at each timestep to learn when to select which hand-crafted tactic and when to maneuver arbitrarily instead. The agents trained with this scheme outperform both of the hand-crafted baseline-tactic opponents as well as untrained agents, demonstrating that the agents' training is effective. These trained PSCE agents learn to switch between the tactics available to them based on their and their teammates' local situational contexts in a way that achieves favorable

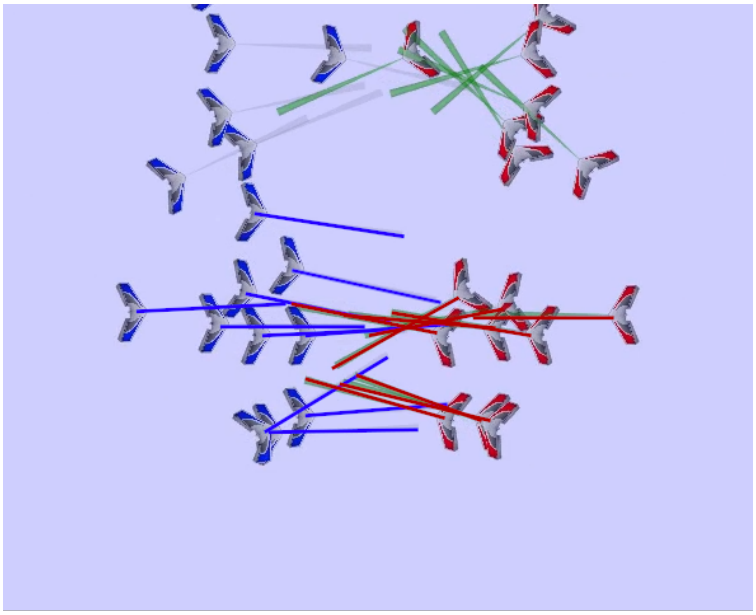


Figure 4.7: An annotated still from a test set simulation (blue team won). Note that, in the annotated cluster of agents, the blue team agents (PSCE (4-vs.-4 trained vs. GS), on the left) spread their aim across all of the approaching red-team members, while the opponents, a team of 16 GS (on the right, red), are aiming at the foremost PSCE agents. The trained PSCE agents are more adept at spreading their force concentration among the opponent agents than GS.

force concentration against the opposing team as well as directs their fire across the opponent team more effectively than either of the hand-crafted tactics could alone. These trained agents are well-suited for intercepting a hostile swarm of enemy fixed-wing UAVs to attrit as many adversaries as possible before they reach a high-strategic-value location. In the next chapter, I consider the close-in defense scenario for this hypothetical high-strategic-value location by a heterogeneous team of multirotors. The experiments in Chapter 5 are inspired by a biological site-defense scenario, and emphasize further how agent spread and establishing local Lanchester's-Square-Law advantage are especially important in swarm-vs.-swarm engagements.

CHAPTER 5
BIO-INSPIRED COORDINATION

This chapter demonstrates prioritization of force concentration advantage by a heterogeneous team against an invading heterogeneous team via the allocation of multirotor guards to guard roles in a biologically-inspired close-in site defense scenario. The simulations described in this chapter emphasize the importance of team spread across the opposing force in swarm-vs.-swarm engagements, as well as demonstrate some key considerations for allocation and prioritization of force concentration in terms of agent types and the guard agents' fuel.

Computer scientists and computational biologists frequently draw inspiration for robot and algorithm design from modeling or emulating animal swarms, their abilities, and their behaviors. Biological swarms provide a wealth of knowledge and inspiration for the modeling and development of man-made swarms, from flocking behaviors and motion seen in shoaling fish [103–106] and swarming locusts [107, 108]; to foraging [109–114], nest-site-location-seeking [115–122], or fighting ants [123–129]; to bee nest defense [130–138]. The experiments detailed in this chapter are focused around a site defense task, which is similar in some ways to the RoboFlag Drill project defense problem for mobile robots [139, 140], where a team of defending robots attempts to deter attacker robots from entering a specific region of a horizontal playing field. In the RoboFlag scenario, all defending robots could counter all individual attackers equally effectively. In contrast, the work in this chapter covers the more complex problem of two types of attackers, each type of which can only be accurately detected by one of two types of sensor, with defender robots equipped with one enemy-detecting sensor each. By not equipping an entire defending swarm of agents with the sensors necessary to detect both types of attackers, the heterogeneous swarm of defenders is less costly to build than a homogeneous swarm of the same size with each agent having both sensors. The results of these experiments empirically find the best allocation of these sensors

and of the fuel that powers the robots employing the sensors given the composition of the incoming adversary force. Furthermore, this chapter explores the force concentration implications of this bio-inspired guard scheme. While some literature exists that examines battery- or fuel-aware constraints for multi-agent applications [141, 142], to my knowledge, I and the other authors of the paper published on the work in this chapter [15] are the first to explore resource prioritization in the defense of a fuel source with agents powered by (and thereby depleting) that same fuel source.

The close-in site defense scenario modeled in this chapter [15] is inspired by the allocation of guard bees to hovering-guard and standing-guard guarding roles in colonies of *Tetragonisca angustula* (*T. angustula*) in their efforts to defend their nests and the valuable resources within from invading bees of both the same species (conspecific) and other species (heterospecific) attackers [134–138]. Standing guards stand on the nest entrance to prevent conspecific non-nestmates from entering the nest, and hovering guards hover around the nest's entrance to identify and defend against heterospecific attackers. In the experiments detailed in this chapter, I show that the structure of the guard force deployed by *T. angustula* for nest defense focuses on the creation of force concentration advantage against the attackers that have the greatest potential to harm the guards' colony, while still allocating enough of the guard force to defense against lower-threat opponents to make the most efficient use of the fuel expended in guarding the fuel reservoir.

5.1 Problem Formulation

Consider the problem of defending a secured location, or High-Value Target (HVT), from a heterogeneous swarm of adversarial multirotor UAVs with a heterogeneous swarm of multirotor UAVs, with the guard swarm following the two-tiered guarding structure *T. angustula* employ in defending their nests from heterospecific (other-species) and conspecific (same-species) invaders. I frame this problem of defending an HVT as a value maximization problem. The HVT is a fuel reservoir containing a specific amount of fuel, which the guard force aims to maximize at the end of the simulation. The attackers approach the HVT one-by-one and attempt to steal fuel from it, while the guards act to intercept, engage, and attrit the attackers before the attackers reach the HVT and steal

fuel. Guard initialization also deducts fuel from the HVT, however, so finding guard schedules that maximize the fuel remaining in the HVT must take not only the composition of the attacking force into account, but also the number of guards. The hypothesis explored in these experiments is that the guarding structure employed by *T. angustula* causes the most damaging enemies to be spread out across the guards and through time, giving the defending swarm a Lanchester's-Square-Law-type advantage against the most-penalizing attackers and providing the defenders with the time they need to successfully counter their opponents before additional opponents can arrive to overwhelm the guards. The simulations discussed here seek to validate this claim.

5.2 Experimental Environment

Our goal in these experiments is to investigate how the value-maximizing allocation of guards (i.e., both in amount and type) is affected by the potential cost of an invasion (i.e., how many adversaries of each type are present and how much value each can individually deduct from the HVT). In these simulations, the HVT is initialized with a specific amount of energy, $V = V_0$, and specific events (guard initialization, guard refueling, attackers entering the HVT) deduct from the HVT energy reservoir; the defending swarm wishes to maximize the energy in the HVT, V, at the end of the simulation. The initial HVT energy reservoir value, guard initialization costs, guard energy burn rates, and attacker breakthrough penalties are given in Table 5.1

Figure 5.1 is a screenshot captured from one of these simulations and shows hovering and standing guard UAVs, conspecific and heterospecific attacker UAVs, and the HVT. The regions in which the hovering and standing guards and the attackers can be initialized are depicted in Figure 5.2.

To protect the HVT energy reservoir, each simulation is initialized with up to ten of each type of guard in increments of two. Up to ten of each type of attacker may approach the HVT during the simulation, with one randomly-selected attacker beginning its approach every $60\,\mathrm{s}$ of simulation time. Guards are generated at random positions within specific bounds to mimic their biological counterparts; hovering guards start on either side of the area in front of the HVT entrance and

Table 5.1: Value-Related Parameters

	Parameter	Value
HVT	Initial value V	$100{,}000\,\mu\text{cal}$
Hovering	Caloric burn rate	$1.250\,\mu\text{cal/s}$
	Average guard time	$3{,}420\,\text{s}$ ($57\,\text{min}$)
	Initialization cost	$4{,}275\,\mu\text{cal}$
Standing	Caloric burn rate	$0.250\,\mu\text{cal/s}$
	Average guard time	$4{,}440\,\text{s}$ ($74\,\text{min}$)
	Initialization cost	$1{,}110\,\mu\text{cal}$
Heterospecific	Breakthrough penalty	$5{,}000$, $10{,}000$, $15{,}000\,\mu\text{cal}$
Conspecific	Breakthrough penalty	$4{,}275\,\mu\text{cal}$

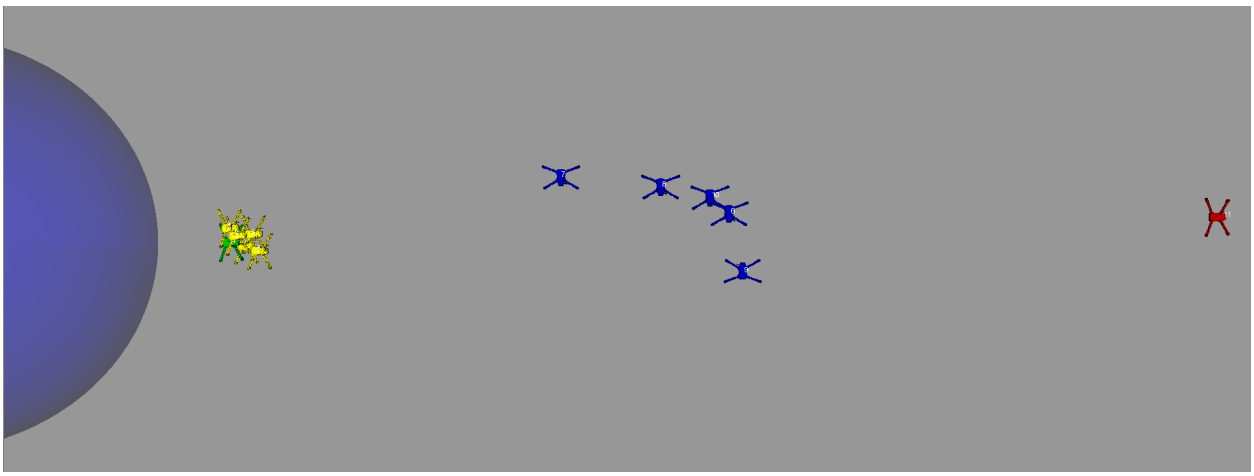


Figure 5.1: Screenshot of an example simulation run being executed in SCRIMMAGE [95]. Blue UAVs are hovering guards, yellow UAVs are standing guards, the green UAV is a conspecific attacker being investigated by one of the standing guards, and the red UAV is a heterospecific attacker.

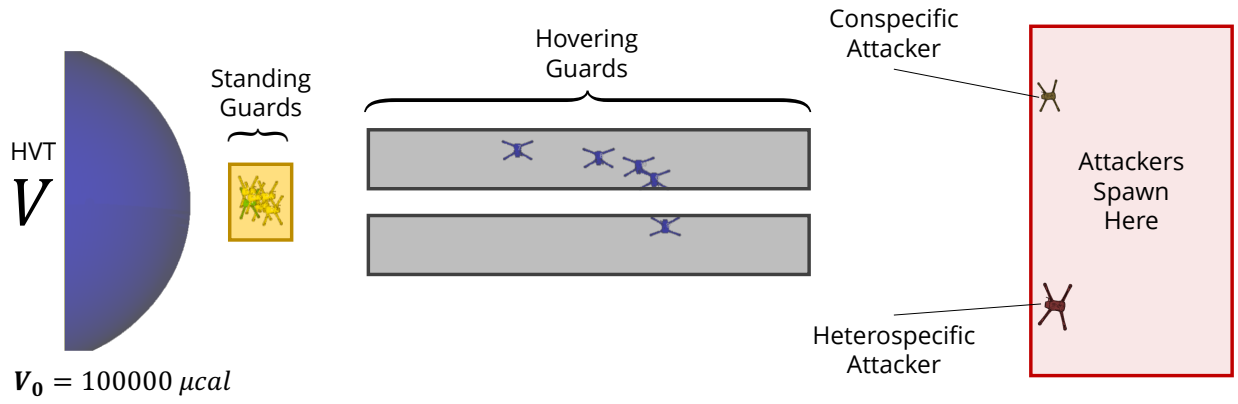


Figure 5.2: Diagram showing the initial caloric content of the HVT and the general regions in which the two types of guards and the attackers can spawn in these experiments.

facing the path to the entrance, and standing guards are initialized very close to and encircling the entrance. All attackers are initialized far from the HVT, well beyond the guards' sensing ranges. Approaching attackers follow waypoints to the HVT entrance that ensure that both types of guards have a chance to sense them. If an attacker reaches the HVT, it deducts a breakthrough penalty (see Table 5.1) from V.

A guard's initialization cost, given in Table 5.1, is also its initial and maximum energy level; when each guard is created, that guard type's initialization cost is deducted from the HVT value, and the same amount of energy is allocated to the guard agent as its initial fuel store. Guards' energy depletes at the role-specific caloric burn rates given in Table 5.1. A guard whose energy level reaches zero is removed from simulation. The attackers' and guards' actions are independent of the HVT value. I executed Monte-Carlo simulations in this framework, varying attacker breakthrough penalties, attacker type distributions, and guard role distributions. The results given in Section 5.4 show the average over 100 simulations for each combination of these factors.

5.3 Agent Behavior

During each timestep of the simulation, an unengaged guard checks its energy level, and with a probability inversely proportionate to its remaining energy level, returns to the HVT to refuel. If it does not choose to refuel, the guard queries its sensor for any potential attacker detections. If

the guard identifies any adversaries, it selects one at random to engage. Once a UAV has survived any encounter with a standing guard, it is close enough to the HVT entrance to enter unhindered; thus, a guard engages its target if the target (1) is not already engaged, and (2) has not yet been in an engagement with a standing guard. Upon engaging the adversary, both the guard and adversary survive with independent probabilities of 0.5. The probability of an ongoing engagement ending is 0.01 at each timestep. While engaged, the guard is fully occupied, leaving the colony more vulnerable to attack. If a guard survives an engagement, it returns to its initial position and continues to watch for attackers.

To represent the primary foci of their biological counterparts, hovering guard agents specialize in identifying only heterospecific adversaries, and standing guard agents focus only on identifying conspecific attackers. The probabilities of each type of guard correctly identifying a member of each enemy type as an adversary are defined in Table 5.2.

Table 5.2: Threat Discernment Probabilities

Guard Type	**Invader Type**	$p(X = a\|a)$
Standing	Heterospecific	0.01
	Conspecific	0.99
Hovering	Heterospecific	0.99
	Conspecific	0.01

Note here the importance of hovering guards being capable of re-engaging adversaries that survived prior engagements with hovering guards; though the engagements themselves are one-on-one duels, the ability of hovering guards to quickly re-engage an attacker that survived a previous engagement with other hovering guards for as long as the attacker is alive and in the hovering guards' sensing range gives the hovering guards a force concentration advantage— a local Lanchester's Square Law (see Section 2.2) within the sensing ranges of the hovering guards—over the individual approaching heterospecific attackers. Under Lanchester's Square Law, the hovering guards' force strength against the heterospecific attackers against which they defend is the difference between the square of the number of hovering guards that can attempt to engage an attacker

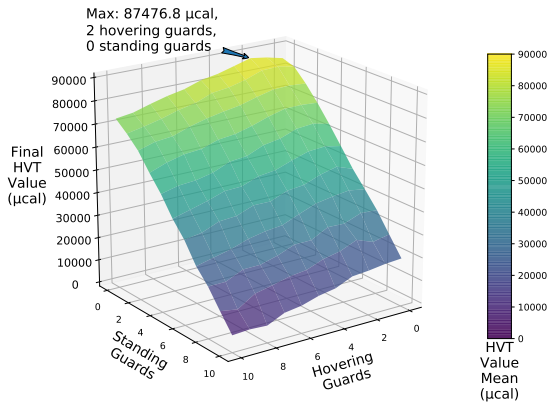


Figure 5.3: Results (averaged over 100 simulations per point) of the final HVT value for the scenario of two low-penalty heterospecific attackers and two conspecific attackers. Note that the maximum, found at two standing guards and zero hovering guards, is almost the same as the HVT value for no guards at all. Guards are not worth the cost to initialize and fuel in this very-low-threat scenario.

minus the square of the number of attackers that can engage the local hovering guards. Hovering guards are often capable of dispatching one heterospecific attacker before another heterospecific attacker arrives at the HVT approach path's start waypoint, as the attackers approach the HVT one by one. With these enemies approaching the HVT as singletons, even a small force of hovering guards has a strong local Lanchester's-Square-Law force strength advantage against most—if not all—heterospecific attackers they encounter. Standing guards, however, are incapable of exploiting Lanchester's Square Law, as only one standing guard may ever engage a specific attacker in a duel-like engagement, and thus must be considered in the context of Lanchester's Linear Law.

5.4 Experimental Findings

A significant factor that affects the maximizing guarding schedule for each scenario is the maximum threat penalty cost. Generally, scenarios with few attackers (up to approximately four total attackers, with low- or mid-penalty heterospecific attackers) required little to no guard force to maximize the value remaining at the end of the simulation. The metabolic cost of maintaining guards to defend the HVT was not worth the low penalty incurred by the losses from attacking UAVs. Figure 5.3 is an example of such a scenario.

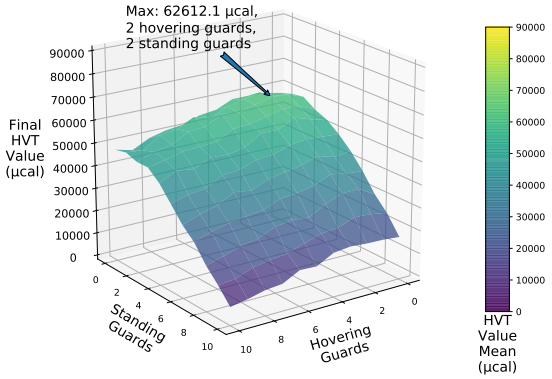


Figure 5.4: Results (averaged over 100 simulations per point) of the final HVT value for the scenario of three high-penalty heterospecific attackers and four conspecific attackers. The maximum in this case is at two hovering guards and two standing guards, but the slope of the mesh is much steeper in the direction of increasing standing guards than in the direction of increasing hovering guards. In such a high-overall-threat scenario, with the majority of the threat concentrated in only three heterospecific attackers, it is of more worth to initialize and maintain a strong hovering guard force and prevent taking the penalty from even a single heterospecific attacker reaching the HVT than it is to initialize and fuel more than two standing guards.

Figure 5.4 shows the results for the moderate-threat case of three high-cost, heterospecific attackers and three conspecific attackers. The results suggest that the best guard schedule for this specific case is two hovering guards and two standing guards, but note that the decline in HVT value as the number of hovering guards increases is much shallower than the HVT value decline as standing guard count increases. This trend indicates that, although hovering guards are more expensive to initialize and maintain, the large cost of even a single heterospecific attacker reaching the HVT in this scenario makes the hovering guards worth investing in to preserve the HVT's value.

Figure 5.5 shows a high-threat case with eight moderate-penalty, heterospecific attackers and four conspecific attackers, for which results indicate that the best guard schedule is to deploy two hovering guards and four standing guards. Due to the high potential threat cost and the threat cost being spread across many attackers, the maximum is at four standing guards and two hovering guards. With the high number of heterogeneous threats, the standing guards' ability to prevent further loss benefits the protagonists. Note especially that, in Figure 5.5, there are maxima between

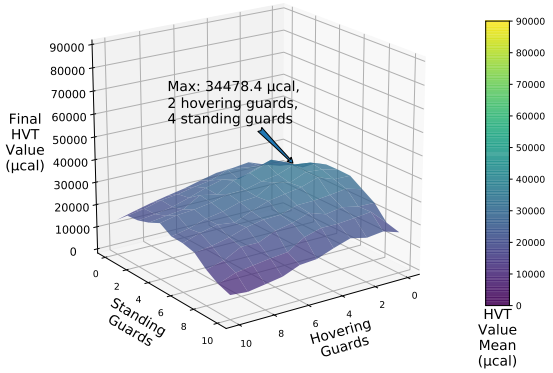


Figure 5.5: Results (averaged over 100 simulations per point) of the final HVT value for the scenario of eight mid-penalty heterospecific attackers and four conspecific attackers. The maximum here is at two hovering guards and four standing guards, but the HVT takes heavy losses under all guard schedules. In this high-threat scenario, the best guard schedule contains far fewer hovering guards than there are heterospecific attackers,but shows maxima between two and six standing guards. The cost of initializing and fueling many hovering guard UAVs is more detrimental to the HVT value than simply allowing some heterospecific attackers to enter the HVT, while deploying standing guards is worth their (smaller) cost.

two and six standing guards, in stark contrast to the steep decline of final V seen with increasing standing guards in the case of Figure 5.4. In such a high-threat case, but with the threat spread across a large number of adversaries, the standing guards' ability to counter conspecific attackers makes a sizable difference. Interestingly, this proportion of hovering-to-standing guards shown in Figure 5.5 is similar to the guard proportions observed by Baudier et al. [138] and Grüter et al. [136] at actual *T. angustula* nests.

5.5 Discussion

As a guard is unable to do anything else when it is engaged with an attacker, maintaining a guard force with multiple of each type of guard is important when an attacker reaching the HVT could cause devastating losses. It is to the defenders' benefit that attackers approach the HVT as singletons; it spreads out the threat of the attackers over time, increasing the chance that an un-engaged guard of the necessary type will be available to engage an incoming attacker, even if some guards are already engaged. Figure 5.5 and Figure 5.4 support this conclusion, as their maxima lie at

guarding structures comprised of multiple hovering guards and multiple standing guards. Grüter et al. [136] found that standing guards are a larger proportion of the guarding force than hovering guards during the majority of a day (similar to the maximizing guard proportions in Figure 5.5), and that a half-standing-half-hovering guard force (such as the maximizing guard schedule in Figure 5.4) was more typical during a nest's busiest times of the day; these observations were corroborated by Baudier et al. [138]. The results from the presented approach are consistent with this observation for some combinations of guard allocation and threat forces, which suggests that the guarding structure *T. angustula* nests use may be influenced by energy efficiency. To the knowledge of my coauthors on this project [15, 138] and me, we are the first to demonstrate the feasibility of this potential motivation behind *T. angustula*'s guard structure in simulation [15].

Section 5.3 clarifies that the combat situation of the hovering guards and the attackers they counter fall under Lanchester's Square Law, and that the standing guards' engagement of the conspecific attackers is instead characterized by Lanchester's Linear Law. With respect to the hovering guards, as the attackers' approach to the HVT is spread out over time, a guard force containing any hovering guards has the advantage of potentially being able to engage a single heterospecific attacker multiple times before the next attacker arrives. As only one standing guard may ever engage a specific attacker, however, the standing guard force's losses are, on average, one-to-one. Due to the lower cost of standing guards, more standing guards may be deployed quite cheaply to defend against conspecific attackers, but the lower-penalty threat of the conspecific adversaries precludes the necessity of an impenetrably-large standing guard force,unless, as in Figure 5.5, the overall threat is large and spread across enough attackers to make guarding the HVT from even low-penalty attackers worth more investment into standing guards. The higher-threat adversaries are countered by the more-expensive agents that are capable of exploiting their local Lanchester's Square Law advantage, justifying the extra expense of initializing and fueling these costly hovering guards.

5.6 Contributions

The main hypothesis in this chapter is that the guarding structure employed by nests of *T. angustula* slows the advance of the most damaging attackers towards the HVT, thereby spreading these attackers out across the hovering guards and providing the guards that can counter these attackers with a better chance at having a local Lanchester's Square Law advantage in engaging them. The two-tiered guarding structure of *T. angustula* balances the expense of hovering guards with them being able to re-engage previously-engaged attackers (and so being able to take advantage of the implications of Lanchester's Square Law). Both the results of the experiments described in this chapter as well as the behavior of the bees that inspired these experiments support this hypothesis, and emphasize the importance of spreading opponents out over one's own team so as to gain a force concentration advantage over them in a swarm-vs.-swarm scenario. Standing guards are nonetheless advantageous in preserving HVT value despite their Lanchester's-Linear-Law combat style and the low penalty of the attackers against which the standing guards defend, as evidenced by the maxima of Figures 5.4 and 5.5. Nonetheless, for both standing and hovering guards, most maxima in the results schedule fewer guards than there are total attackers, implying that, as the members of the guarding force are powered by the same commodity they are defending, it is usually advantageous to try to prevent some theft from the HVT by both types of attackers, but also to to expect and accept that some losses will occur as well rather than spend large quantities of fuel deploying a guard force large enough to constantly prevent any attackers from reaching the HVT at all.

In this chapter, the deployment of the most costly guards provided a strong force-concentration advantage against the most threatening opponents by spreading them across the high-cost guards, but cheaper guards operating under Lanchester's Linear Law were still deployed as well to shore up the overall defense structure against less-threatening adversaries. This raises a point of emphasis with respect to own-team spread, opponent spread, and force concentration: exploiting such a Lanchester's Square Law advantage against a group of enemies is most effective when force is

not only concentrated at the highest-priority threat (the foremost GS agents in Figure 4.7, and the heterospecific attackers in this chapter), but is also distributed in such a way that lower-damage or lower-priority threats are attrited before they exact losses on the protagonist team that cause it to be less effective against the enemy force as a whole (protagonist team members being attrited, or, specific to the current chapter, losing fuel to continue refueling hovering and standing guards). As demonstrated throughout this dissertation, it is likely that, in engagements against aggressive teams of agents, some protagonist-team losses are unavoidable in larger engagements if the blue team wishes to attrit opponent team members. The conclusions and tenets of force concentration, own-team spread, and opponent-team spread I have elucidated in this dissertation are beneficial for ameliorating these necessary losses to the protagonist team in engagements between either fixed-wing UAVs or multirotors.

CHAPTER 6
PRACTICALITIES: LIMITATIONS AND FUTURE WORK

In this dissertation, I have presented schemes for engagement of opponent teams of fixed-wing UAVs as well as a bio-inspired approach for close-in site defense. The experiments demonstrating these approaches were conducted in simulation, however, and are thus constrained in their applicability due to assumptions made during simulation construction. To deploy such approaches on actual aircraft, many of these assumptions will need to be dropped and limitations will need to be overcome. In this section, I cover the major limitations and postulate how they might be overcome or sidestepped, as well as discuss how to approach dropping some unrealistic assumptions.

6.1 Fixed-Wing Aircraft Approaches: Addressing Limitations and Future Directions

The fixed-wing aircraft simulated in the experiments in Chapter 3 and Chapter 4 employ simplified aircraft damage and weapon models, are constrained to the horizontal plane, are equipped with perfect, non-noisy sensors, and can communicate infallibly. Furthermore, the PSCE agents detailed in Chapter 4 train against simulated opponents, when in reality, no antagonist team would willingly provide their agents' behavior model to the protagonist team for them to train against, and would likely try to frustrate the protagonist team's training efforts or enter engagements with different behaviors or capabilities in an attempt to render the PSCE agents' training useless. Here, I present potential future work that aims to explore the performance of PSCE agents in environments in which these assumptions are dropped or made more realistic.

6.1.1 Addressing Aircraft Damage Model and Weapon Model Limitations

The agents in the experiments detailed in Chapter 3 [13] and Chapter 4 all ignore friendly fire, collisions between aircraft, and make the assumption that a single successful shot from one aircraft

to its target destroys the target. Further simplifications include the details of the weapon model itself. In preparing to test PSCE agents in live-flight experiments with actual weaponry, one would ideally select the weapon the real-world aircraft will be employing in live-flight experiments, then model that weapon in simulation for further testing. As the types of weapon employable in live-flight conditions are strongly dependent upon the exact type of airframe being utilized, I cover only the generalities of the damage and weapon model simplifications below that would need to be addressed in the process of preparing PSCE agents for live-flight live-fire experiments.

In modeling damage to aircraft in the experiments detailed in this dissertation, all weapon models (cannon model for the fixed-wing experiments; duel engagement model for Chapter 5's bio-inspired approach to close-in site defense) assume that a single shot that hits a target destroys the target, that friendly fire has no effect, and that all agent attrition is the result of opposing team fire—aircraft can pass through one another without worry of collision or damage. With actual aircraft, the lethality of a successful fired shot depends not only on the type of weapon that fired the shot, but also on what part of the target it strikes. A shot that clips an agent's wing may decrease that agent's agility somewhat, but is less likely to disable the aircraft than a shot that strikes a more crucial part of the airframe, such as a propeller motor or compute board. In the fixed-wing simulations detailed in this dissertation, an aircraft is attrited if it is hit by a single shot, and that shot may hit anywhere within a $2\,\mathrm{m}$ radius of the target aircraft's centroid. For training and testing agents in more realistic conditions in which not all damage is lethal and some shots are more detrimental than others, partial damage to an aircraft could be estimated assigning so-called Hit Points (HPs) to aircraft upon their initialization, and shots that hit closer to an aircraft's centroid—where the most vulnerable components would likely be located—would detract more HP than shots that hit the aircraft further from its centroid. Including friendly fire as another way for an agent to lose HP would further add to the realism of this potential future work. Potential future work also includes training PSCE agents in simulated environments in which collisions can occur and can damage or destroy aircraft, with the number of HP deducted from aircraft involved in a collision is proportional to the relative velocity of the aircraft in the collision. I predict that agents trained

with the conditions of HP-based damage, allowing multiple hits per agent, and with collisions and friendly fire inflicting damage will learn to maneuver more conservatively and precisely, and will also find ways to switch tactics or manipulate their maneuver actions in ways that can take advantage of these altered damage model components, e.g. learning to maneuver in ways that cause opponents to collide with one another. The baseline tactics, especially GS, would require some modification in order to stay competitive in collision-capable engagements, as GS agents tend to cluster together (and collide) as they approach an opponent. For such potential future work, I propose equipping GS and DA with collision avoidance behavior (e.g. [143]), and re-training PSCE agents with these modified GS and DA behaviors as selectable actions.

Which agent a firing agent hits with its fire is, in these simulations, determined by the firing agent; the firing agent attempts to fire at an agent with a specific ID number. With this aiming mechanism, only the agent with the ID matching that that the firing agent is aiming at will be affected by the fire if the shot is successful, even if other agents are within the firing agent's Weapon Engagement Zone (WEZ). For example, if agent 5 is in agent 1's WEZ, agent 1 can specifically fire at agent 5. If agent 1 fires at agent 5 while agent 4 is directly between agent 1 and 5 (and is therefore also in agent 1's WEZ), agent 4 is completely unaffected by agent 1's fire. Additionally, if teammate agent 2 is in agent 1's WEZ when agent 1 fires, agent 2 remains completely unaffected, as friendly fire has no effect in the simulations detailed in this dissertation. For more accuracy in weapon-employment aspects of the engagement types explored in this dissertation, potential future work includes dropping the assumptions of friendly fire being ineffective and agents being able to specify at which agent within their WEZ their fire is aimed. More accurate modeling of where on an aircraft a successful shot hits and what impact that damage would have on the aircraft—rather than assuming that a single shot that hits an aircraft takes it out of the engagement completely—would also increase the accuracy of simulated experiments.

Under the weapon model employed in Chapter 3 [13] and Chapter 4, probability of kill decays as a function of the distance between the firing aircraft and its target, and goes to $p_k = 0.0$ upon the distance between firing aircraft and its target being greater than a maximum firing distance.

Probability of kill is a common simplification of whether a fired weapon destroys its target [64, 65, 68–71]. The assumption that p_k suddenly drops to 0.0 at a fixed maximum WEZ distance is a simplification that would be dropped if testing for deployment in a realistic situation. Furthermore, due to the relatively close range in which the experiments in Chapter 3 [13]— and even moreso in Chapter 4—are conducted, the weapon model in those experiments assumes that a fired shot that hits its target will do so instantly. Depending on the weapon modality selected, the time-of-flight of a fired shot may differ based on various factors, but is highly unlikely to be instantaneous; future work includes testing with the time-of-flight of fired shots dependent upon the distance between firing aircraft and target. Gravity and air resistance also affect fired rounds, in contrast to the weapon model of the fixed-wing experiments, in which each fired round is assumed to follow a straight-line path. Realistic weapon model testing should take these weapon-specific factors into account.

6.1.2 Three Dimensional Environment

Aircraft that operate in three dimensions—as non-simulated aircraft do—must take care to maneuver in ways within the physical capabilities of their aircraft. Sequencing maneuvers in a way that allows the aircraft to remain stably aloft is much more complex in three dimensions than in two; nonetheless, for practical usage in actual aircraft, PSCE agents must be capable of exploiting within-team coordination in 3D and recognizing situations in which various tactical behaviors are appropriate. No adversary with a swarm that can operate in 3D would willingly limit their own UAVs to the 2D plane just so PSCE agents could counter them. Furthermore, the depth of tactical intricacy of 3D dogfighting is far deeper than that of 2D dogfighting—practical PSCE agents should be capable of exploiting a potential energy advantage or opportunities in which the protagonist team could confuse opponent UAVs, e.g. by having some protagonist aircraft engaging enemy aircraft within easy sight of the enemy aircraft while other protagonist-team teammates attack from high in the sky with the sun behind them to "hide" their approach from the antagonist team.

The chief concern in equipping PSCE agents for operation in three dimensions is avoiding

states that can lead to crashes or other causes of aircraft inoperability. For these safety concerns, I propose that each aircraft perform checks for safe flight each timestep before it begins its action selection procedure. Any aircraft that determines in a given timestep that it is in or approaching a state that requires emergency recovery bypasses that timestep's action selection process, informs its teammates that it is unavailable for partnership in the current timestep, and instead selects an appropriate pre-scripted emergency recovery behavior. As these aircraft are unmanned, however, it is possible that sacrificing an airframe to perform a risky maneuver to confuse adversaries or to act as a decoy may be advantageous, so further experimentation, testing, and development would benefit the selection of these safe-flight-condition thresholds and the conditions under which tactical needs could override them.

Emergency recovery is not the only way in which expanding to three dimensions might be seen as problematic, however; as explored in the work of Ure and Inalhan [144–146], fixed-wing UAVs performing aerobatic maneuvers cannot always directly transition from one complex 3D maneuver to another without intervening control actions to preserve stable flight. For PSCE agents, this maneuver sequencing issue is handled by the tactical behavior action options from which agents select each timestep; each tactical behavior is treated as a self-contained black-box policy for agent behavior. So long as each tactical behavior has appropriate contingencies for how an aircraft should behave in arbitrary states, PSCE agents should be able to learn to switch between tactical behavior actions meaningfully.

An additional concern to address in adapting PSCE agents for three dimensions is their state representation. The PSCE state representation, described in Section 4.3.2, is a discretization of a bird's-eye view of the observing agent and the agents surrounding it, with multiple 2D channels containing various quantities of the agents represented therein. In expanding the training and testing of PSCE agents to 3D, this state representation necessarily grows larger, and each of what are channels in the 2D-flight-version representation becomes a discretized prism in the 3D representation. This expansion in dimension naturally necessitates further hyperparameter tuning—how far above and below the ego agent to extend the 3D state representation, the size of the bins of the

height aspect of the discretization, whether the bin size for the horizontal plane discretization needs to change to provide additional precision to handle teammates and opponents with more agility, and so on. With such a large state representation for each agent, the computing resources—especially GPU memory—for training 3D PSCE agents increases dramatically.

6.1.3 Limitations on Specific Tactical Behaviors

As mentioned in Section 3.2 [13], DA primarily flies at its cruise velocity—only allowed to briefly slow down to achieve echelon formation with its partner or as a part of a precise part of a coordinated maneuver—and GS only flies at its cruise velocity. The implementation of DA employed in these experiments only utilizes tactics that take place within the horizontal plane. Naturally, upon extending the problem into three dimensions, aerial combat becomes much more complex; many more factors influence the outcome in a 3D engagement than in 2D, even just between two aircraft. To take full advantage of the 3D environment, DA would require expansion to utilize 3D pairwise-coordinated maneuvers, and would also need further refining for DA agents to make tactical decisions regarding their own potential energy and velocity, and to do so in the context of their current partner's and opponent's altitude and velocity. GS is equipped to fly in three dimensions, but in its current implementation, it does not alter its velocity; to take full advantage of three dimensions and the ability to exploit a potential energy advantage over its opponents, GS would need to be altered to speed up and slow down as appropriate to pursue its chosen target. These enhancements to GS or DA would provide PSCE with more flexibility in moving to the 3D space. Alternatively, PSCE agents could be modified to leverage tactical doctrine implementations besides GS or DA; tactical doctrines that incorporate maneuvers that take advantage of potential and kinetic energy advantages over opponents and are a more natural fit for a three-dimensional environment. Even PSCE's maneuver action option would need to be altered; instead of the continuous branch of the policy network outputting the mean of a yaw rate setpoint distribution, it would output means for distributions that would be used to select the roll rate and pitch rate setpoints of the aircraft.

An additional area of future work with respect to the specific behaviors employed by the PSCE

agents introduced in this dissertation is when each behavior is selected by PSCE agents. In future experiments between PSCE agents and various opponents, recording exactly when in each engagement each agent selects which behavior, and comparing these behavior selection choices by trained teams to behavior selection choices made in similar scenarios by untrained PSCE agents has the potential to be a fascinating realm of insight into exactly what the PSCE agents learn during training and what in their decisionmaking process allows them to outperform their untrained counterparts. Such data would also provide further insight into how the behaviors selected change when other factors, such as limits on communication, sensing, or which agents' state representations a PSCE agent may leverage during action selection. Limits on sensing, communication, and partnership are of especial interest for practical application of PSCE-trained agents, even in the absence of this additional behavior-selection-timing data.

6.1.4 Sensing and Communications

Another area of unrealistic aspects of the fixed-wing aircraft experiments detailed in this document is perfect, noiseless sensing and communications. The sensing of other aircraft is simulated as a simplified generic "aircraft sensor," and is given a wide range ($1\,\mathrm{km}$) in which it provides non-noisy exact positions of aircraft. In reality, this aircraft sensor functionality would likely be replaced by a combination of sensors and processing functionality. If communications remain reliable and feasible, agents on a team could obtain their own positions in the world via GPS, estimate the state information of nearby adversaries (discussed later in this section), and communicate this information to their teammates; With some modern fighter aircraft protocols, such as those with which the F-35 Lightning II is equipped [147], it would be simple to share state representations or other data about one's surroundings with one's teammates for evaluation. If such a comprehensive communication protocol is not available (such as may be the case on small, simple Commercial Off-the-Shelf (COTS) or Government Off-the-Shelf (GOTS) UAVs), communicating entire state representations between all teammates may be infeasible, and agents may need to limit themselves to sending shorter messages about what they sense so their teammates can reconstruct their state

representations locally to run the PSCE action selection procedure. As each agent operates based on its own and its team members' ego-centric state representations, however, PSCE reduces or eliminates the need for agents to come to a consensus on an inertial-frame world view of all agents in the engagement for the purposes of decisionmaking [148]; if each agent communicates its ego-centric state representation to its teammates, all agents can employ PSCE to make action decisions. If communication between teammates is limited to within some range around each agent, though, the problem arises that agents on a team must not only make decisions on what action choices are tactically advantageous, but also whether those action choices will disrupt the connectivity of the team's communication network [149, 150]. Possible future work includes further investigation into the incorporation of mutual information into the reward structure or state representation so that agents can include it in their decisionmaking process, or modifying the training framework to include mutual information in other ways [93, 151]. Mutual information may also be a fruitful metric with which to evaluate the performance and behaviors of trained agents at test-time. Traditional radio/wireless communications, however, may be superfluous and even detrimental to the protagonist team—leveraging radio communications in the presence of adversary team members is risky, even when the adversaries cannot understand some or all of the communicated data. An enemy force intercepting an encrypted message transmitted from one protagonist-team aircraft to another may not be able to decrypt the message, but can still uncover a large amount of information from the message itself and the manner in which it was intercepted. A single message transmitted between two members of the protagonist team that is intercepted by the enemy force reveals to the enemy force that:

- Some contingent of the protagonist team is nearby,
- This protagonist team force contains more than one agent (hence their communication), and
- Some information about each communicating agent's proximity and bearing relative to the location at which the enemy intercepts the communication.

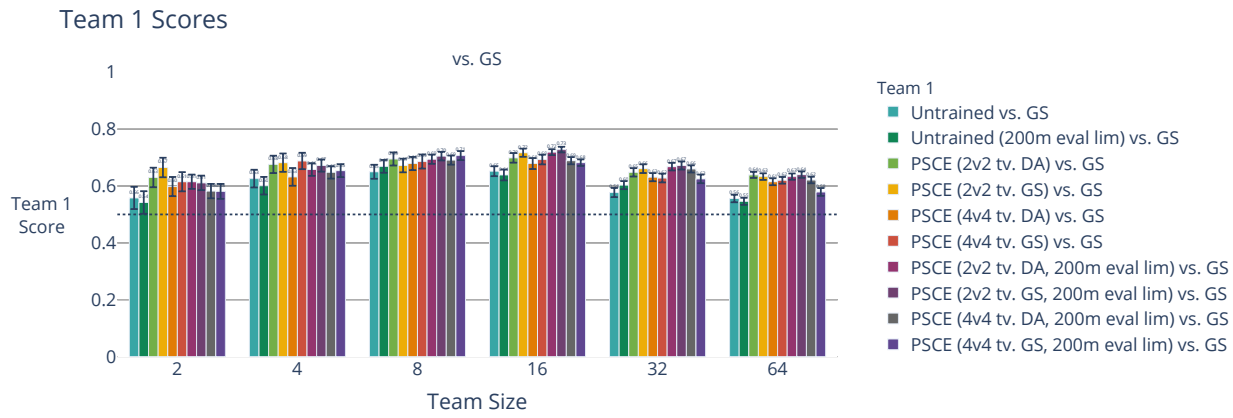
Thus, for the most practical application of PSCE agents in live-flight and especially live-fire sce-

narios, and especially when the protagonist team would gain tactical advantage from the element of surprise, the protagonist team agents must each be capable of sensing with enough range and accuracy to estimate each other's state representations, and perform any crucially-necessary communication with radio silence.

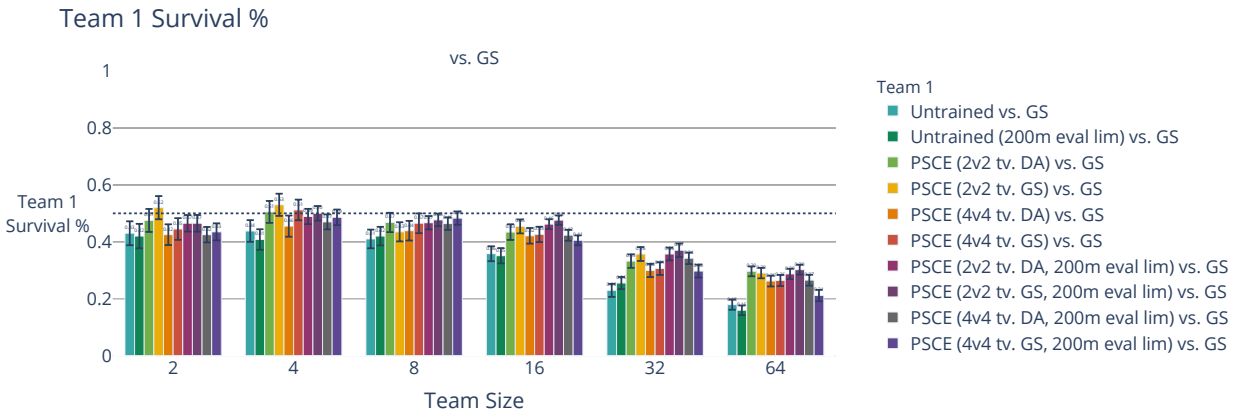
As in all but friendly competitions [2, 40, 152, 153] with opponents, one cannot expect the opponents to divulge their own positions to the protagonist team, the protagonist team needs to be capable of sensing adversary agents and estimating their state information in order to construct their own *and* their teammates' state representations for decisionmaking. For the sake of simplicity, in this section, assume that the protagonist team UAVs are equipped with a number of cameras and the processing capability to identify and track hostile and friendly agents all around the agent, but exactly what sensing equipment might be employed by the aircraft is dependent on what the aircraft is capable of carrying and powering, what the onboard compute resources are capable of processing quickly enough for decisionmaking, and the ongoing evolution of sensing and vision capability. Regardless of the exact sensor modality, all sensors are noisy to some degree, and no sensor in the real world is omniscient. The aircraft on the protagonist team need to be capable of handling noisy sensor data, as well as making decisions based on sensor data from within a limited range of the sensor's location. A number of methods for estimating the state of objects based on noisy observations of those objects exist [154, 155]; the implementation details of such filtering algorithms within the context of making the application of PSCE practical are important future work, but are outside the scope of this dissertation. Assume that, for the aircraft an agent can observe with its sensors, the agent can construct a reasonably-accurate estimate of the state representation detailed in Section 4.3.2. Even with the protagonist team agents able to independently and autonomously estimate state representations of the senseable agents around themselves, and even if the protagonist team chooses to communicate their state representations with the teammates they can sense, the question remains: do PSCE agents whose sensing range—and therefore state representation size—is limited have sufficient information to make good tactical decisions? To begin answering this question, I train modified PSCE agents; these agents are the same as the

agents for whom results are presented in Chapter 4, Section 4.5, except their state representations are only $200\,\mathrm{m}$-by$200\,\mathrm{m}$, and each PSCE agent may only evaluate state representation pairings with teammates who fall within the evaluating agent's state representation region. For simplicity of implementation, the DA and GS target selection procedures are allowed to retain their original sensing capabilities; thus, PSCE agents that select DA or GS may target opponent agents within $1{,}000\,\mathrm{m}$ of themselves. As discussed in Section 4.6, however, with PSCE agents re-evaluating their actions every timestep, it is highly unlikely that a pair of DA-acting PSCE agents aiming a bracket at an opponent $1{,}000\,\mathrm{m}$ away will continually select DA with each other for long enough to see that bracket maneuver through to the attrition of that same opponent. A DA pair under PSCE will likely only stay partnered long enough to achieve echelon formation with one another; PSCE agents are more likely to fire at opponents opportunistically or while executing GS, negating much influence of DA's target selection logic. Due to PSCE's frequent action switching, then, the partnership evaluation limit distance is the most influential in a PSCE team's effectiveness. As shown in Figure 6.1 and Figure 6.2, in engagements against DA and GS, PSCE agents trained with this evaluation limit performed similarly to teams trained with the larger state representation size and no limits to potential partner evaluation. This trend does appear to start to break down, however, for PSCE (4v4 tv. GS, 200m eval. lim.) when team size reaches 16 and the trained PSCE team scores only slightly higher than the untrained agent team. Note that engagements against teams that the PSCE agents did not train against are included in the plots shown in Figures 6.1 and 6.2, which are also discussed (without the sensing- and evaluation-limited cases) in Appendix A.

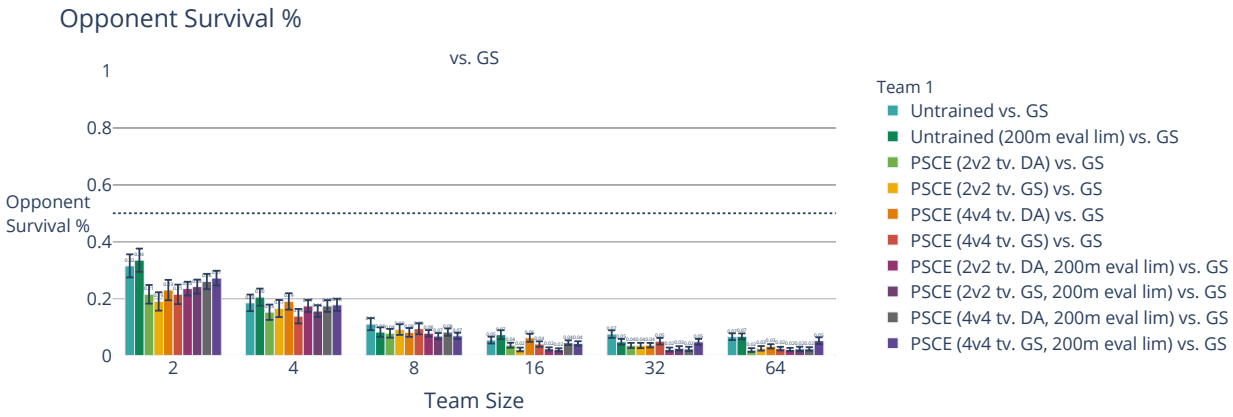
In the results shown for sensing- and partner-evaluation-limited agents shown in Figures 6.1 and 6.2, it is apparent that the PSCE teams are relatively unaffected in their performance against DA (Figure 6.2), but against GS (Figure 6.1), not only does protagonist team survival (and, to a lesser extent, score) decline as team size increases, but PSCE (4v4 tv. GS, 200m eval lim.), while still outscoring its GS opponent, shows a much stronger pattern of decline than does its non-evaluation-limited non-limited-sensing PSCE counterpart. Especially against non-coordinating teams such as GS, further testing and development would be beneficial to mitigate the potential problems



(a) Average score of untrained, trained teams with and without evaluation limits and restricted state representation sizes against teams of GS in N-vs.-N engagements $\forall$ N $\in 2, 4, 8, 16, 32, 64$.

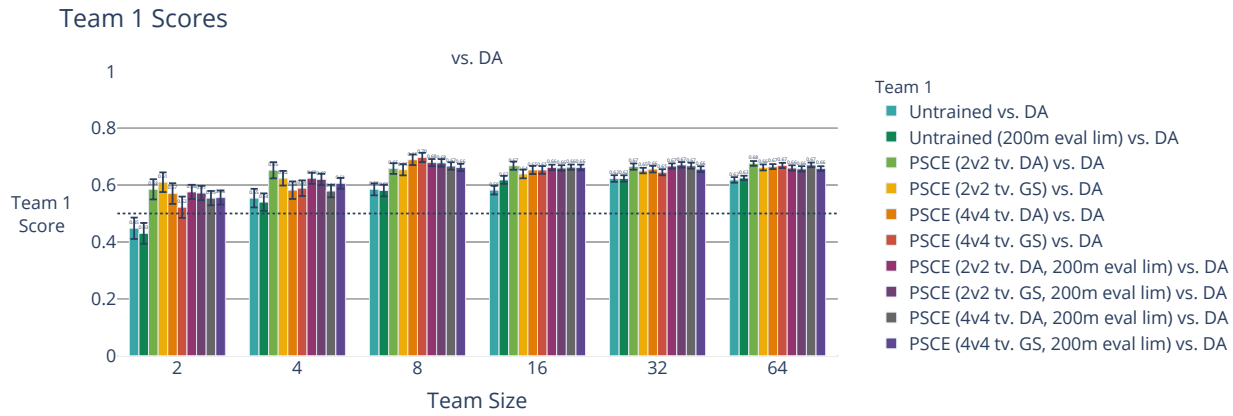


(b) Average percentage of untrained, trained teams with and without evaluation limits and restricted state representation sizes that survive in engagements against teams of GS in N-vs.-N engagements $\forall$ N $\in 2, 4, 8, 16, 32, 64$.

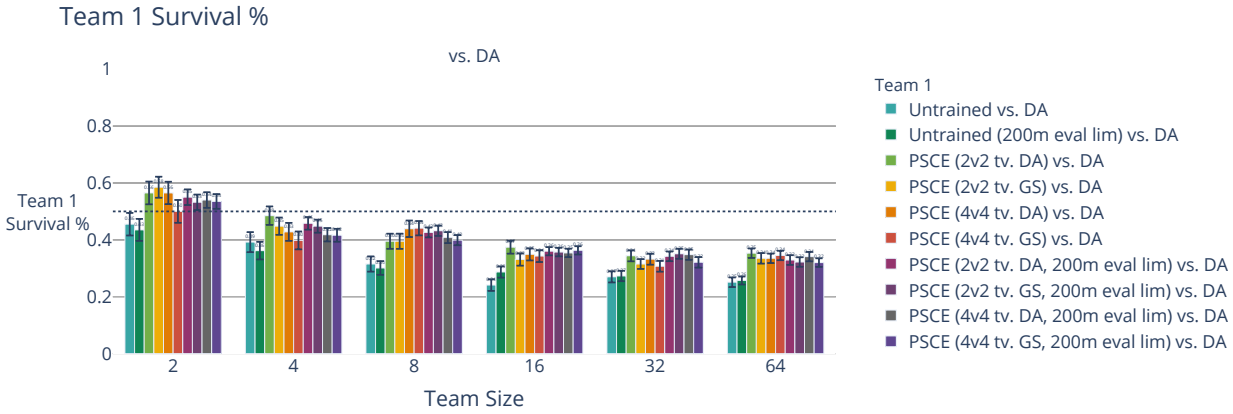


(c) Average Percentage of GS teams that survive in engagements against teams of untrained or trained PSCE agents with and without evaluation limits and restricted state representation sizes in N-vs.-N engagements $\forall$ N $\in 2, 4, 8, 16, 32, 64$.

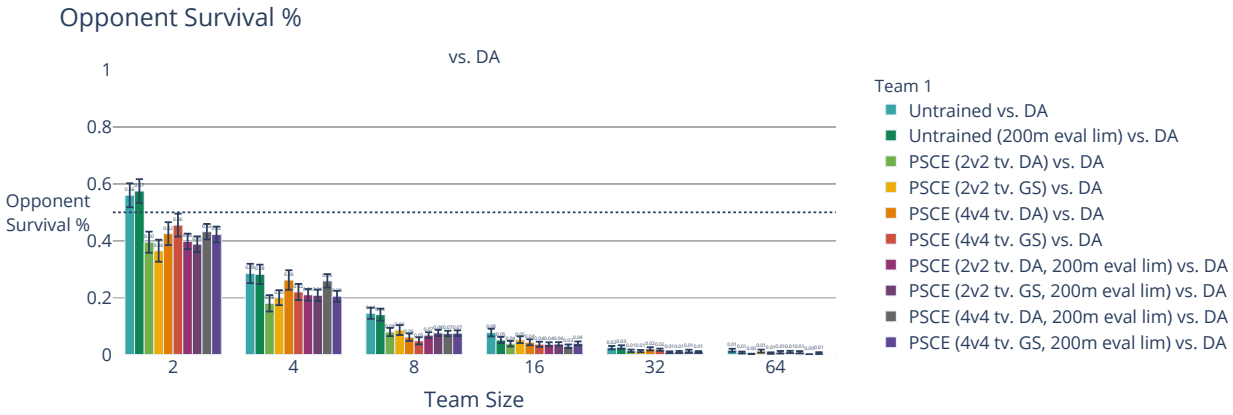
Figure 6.1: These plots show the scores of PSCE agents, with the training and evaluation limit (or lack thereof) denoted in the legend, in engagements against GS teams.



(a) Average score of untrained, trained teams with and without evaluation limits and restricted state representation sizes against teams of DA in N-vs.-N engagements $\forall$ N $\in 2, 4, 8, 16, 32, 64$.



(b) Average percentage of untrained, trained teams with and without evaluation limits and restricted state representation sizes that survive in engagements against teams of DA in N-vs.-N engagements $\forall$ N $\in 2, 4, 8, 16, 32, 64$.



(c) Average Percentage of DA teams that survive in engagements against teams of untrained or trained PSCE agents with and without evaluation limits and restricted state representation sizes in N-vs.-N engagements $\forall$ N $\in 2, 4, 8, 16, 32, 64$.

Figure 6.2: These plots show the scores of PSCE agents, with the training and evaluation limit (or lack thereof) denoted in the legend, in engagements against DA teams.

associated with less-than-perfect sensing and limited sensing ranges.

Further experimentation with sensing-limited agents constructing state representations from noisy data for both an individual agent and its senseable teammates will be required to determine how PSCE agents could best operate in communications-limited or even communications-denied environments with imperfect sensing. In steps towards practical application of PSCE agents in communications-limited environments, I postulate that paired PSCE agents executing DA could potentially run an estimation of their teammates' decision processes locally to attempt to determine what stage of DA their partner is in, but this approach is complicated by each agent not being able to sense everything that its partner can, and thus potentially not reaching the same evaluation results for its partner as its partner does. As an alternative, as sensing ranges of the agents are limited in these cases, it is reasonable, albeit potentially somewhat error-prone, to instead leverage visual signals to communicate any necessary small messages for coordination in a radio-silent manner—wing wags, flashing LEDs in pre-arranged patterns, etc.

6.1.5 Mitigating Adversary Deception

These measures for handling practical limitations such as restricted communications and noisy, limited-range sensing show that there is a path forward towards deploying PSCE agents in live flight, all of these mitigating measures assume that the opponents the PSCE agents are training against or facing at test time are truly capable of inflicting damage and are operating under a static policy. Problematic assumptions about the behavior of the opponent teams yet remain, however; thus, in additional steps towards practical deployment of PSCE agents against adversarial swarms, I present potential approaches to ameliorate the potential issues of adversary teams behaving in ways that confuse the training of PSCE agents, as well as heterogeneous enemy teams and adversaries that can deploy decoy agents.

Pathological Training Opponents

The training of the PSCE agents in Chapter 4 occurred purely in simulation, and against opponent teams that operate under a static, non-learning hand-crafted policy. Thus, the question naturally arises: how would PSCE agents fare if trained against a learning opponent team? Furthermore, could a learning opponent team change its behavior from epoch to epoch in such a way that the PSCE agents' policies would serve them poorly in test engagements? And, finally, if the opponent team is switching its behavior from epoch to epoch, is there a point at which the protagonist team does not gain anything from further training against this unpredictable opponent team?

As mentioned earlier in this chapter, real-world adversaries would not be willing to provide information on their UAVs to the protagonist team for the sake of the protagonist team's advantage. Given that PSCE agents train in simulation, while I showed that the policies learned by the trained PSCE agents learned strong tactics that generalize well to at least one other enemy team tactical behavior, it would perhaps benefit the protagonist team to train during live engagements to gain a stronger understanding of how to counter a specific adversary team. An enemy, however, would not sit idly by and allow its agents to keep a static policy against which the protagonist team could train and learn to best. If the adversary agents are also learning while in engagements against the protagonist team, is there a means by which the protagonist team can tell when it has learned "enough"? Is there a point in training in which it no longer benefits the protagonist team to continue training against the adversary team? These questions are difficult to answer; as Smith et al. discuss in some of their Learning Classifier System (LCS) work, meaningfully measuring the effectiveness of one learning team when comparing it to another learning team is especially difficult [46]. Naturally, results from tests between agents employing various versions of the protagonist team's policy and a baseline-tactic team will give some quantification of the protagonist learner team's improvement, but whether that improvement in performance against a baseline team truly translates to improvement against an adversary team of unknown policy is more difficult to state conclusively. Comparing two learning teams' performance invites comparison to two-player zero-sum games, and indeed, game theory is an apt framing of some learning frameworks, such as

Generative Adversarial Networks (GANs) [156]; the two neural networks, the Generator and the Discriminator, each select "plays" (network weights) each "turn" (training epoch) in an effort to, for the Generator, prevent the discriminator from telling a true output from a false one, and for the Discriminator, to correctly identify whether the Generator's output does or does not belong in some category. Against a pathological learning adversary team—one that aims to behave in ways that frustrate and confuse the protagonist team's training—I believe that there may indeed be a point at which the protagonist team may be unable to learn any new *useful* data, particularly if the adversary team changes its behavior from engagement to engagement in ways that force the PSCE-agent policies to operate in local maxima. In a step towards circumventing this issue, I propose that deployed PSCE agents maintain a known-good backup of their network weights that the agents can revert their policy networks to if the protagonist team finds themselves averaging fewer opponent attritions or more losses than they have in previous engagements, suggesting an altered opponent policy, a confused update to their own policy since the most recent engagement, or both. Alternatively, it may be beneficial to alter the structure of the training framework to include a critic that, each epoch, trains from data selected randomly from a replay buffer containing a variety of past experiences' state representations, actions, and rewards; this approach is commonly leveraged to ameliorate the effects of *catastrophic forgetting*—when a neural network "forgets" what it learned from past experiences when it trains on newer data [157]—and would assist the agents in continuing to learn tactic selection in a way that generalizes well against a variety of opponents. With these measures, while PSCE agents training against pathological opponents may not always be able to improve their policy by training against the opponents, PSCE agents should at least retain a good policy that works well and generalizes to handle a variety of swarm-vs.-swarm situations regardless of how the opponent team attempts to confuse their training.

Decoys and Heterogeneous Adversary Teams

In the fixed-wing experiments detailed in this dissertation, the adversary team is assumed to all be using the same policy for the entire engagement. In adjusting PSCE agents for live-flight de-

ployment, however, one must consider the preparedness of trained PSCE agents for countering teams of agents with heterogeneous behavior, as well as teams that deploy decoy agents to distract the protagonist team's members. As shown in Appendix A and Figures 6.1 and 6.2, PSCE agents trained against either GS or DA learn generally good tactics that perform well against teams for which the PSCE agents were not trained. While that does not automatically mean that the trained PSCE agents will definitely perform well against a mixed-behavior team, it is a good indicator that the trained PSCE agents are likely to be well-equipped against unknown teams, especially if the (reasonable) assumption is made that the opponent team's objective is to attrit members of the protagonist team. Currently, PSCE agents have no way of discerning different types of opponent team agents from one another, e.g. decoys vs. opponent agents that can truly attack. Thus, trained PSCE agents would be vulnerable to the deception of an adversary team that deploys decoy agents. Assuming that decoy opponent team aircraft would be launched from an existing opponent team UAV, and would follow a much simpler flight pattern than the non-decoy adversaries, however, I hypothesize that it would be feasible to pre-emptively train PSCE agents against simulated adversary teams that deploy decoys that fly with simple non-aggressive flight patterns and expect somewhat better results than could be expected than if a team of GS or DA were pitted against the decoy-deploying team. To improve PSCE-like agents training to perform truly well against decoy-deploying teams, however, I postulate that the agents will need to train in the context of more than just the current timestep's state representation pairs. To this end, I propose incorporating more complex architecture elements into the training of PSCE-like agents, such as training with a critic network, and perhaps equipping the critic network with Long Short-Term Memory (LSTM) cells [90] to allow the critic to learn the relationships of the behavior of active adversaries and their deployed decoys over multiple timesteps and train the agents accordingly.

6.1.6 Further PSCE Training Improvements

Additional enhancements that could be incorporated into the process of training PSCE agents include testing alternate network architectures and training algorithms, hyperparameter tuning, test-

ing alternative reward structures, including additional information in the state representation, and recording additional metrics for more detailed evaluation of performance.

In Chapter 4, I test only one policy network architecture, and only train that policy network with REINFORCE. Potential future work includes comparing agents trained with REINFORCE to agents trained with other deep multi-agent RL algorithms—potentially algorithms that provide security against catastrophic forgetting—and further training and testing with additional policy network architectures. Still, the results obtained with my REINFORCE-trained architecture in Section 4.5 show that the fundamentals of the training framework and input/output structuring are effective.

Further improvements to PSCE agent performance may be possible through additional efforts into hyperparameter tuning, though such tuning adjustments may also be specific to particular deployment scenarios or aircraft platforms—hyperparameter tuning is, debatably, more art than exact science. Key hyperparameters to tune further include the minimum and maximum values of the entropy coefficient of the performance measure gradient estimate function, as well as the threshold value for the condition of increasing the entropy coefficient and the factor by which it is increased. This would adjust how much the agents-in-training value the entropy of their action choices with respect to the other terms of the performance measure gradient estimate.

Additional enhancements to the training procedure include testing alternate reward structures and additional refinements to the state representation. As mentioned in Section 4.3.5, the reward currently in use for training PSCE agents is biased towards encouraging the protagonist team to attrit opponents, and only rewards the protagonist team with positive rewards if all opponents are attrited. I explored training with less biased reward structures, but found that those alternate reward structures (especially in the various structures I tested in which own-team and opponent-team attrition were rewarded with equal positive and negative reward) resulted in the trained agents focusing on learning to evade the antagonist team agents rather than attrit the antagonists. For the sake of simplicity, I kept the same reward structure throughout the training of the agents tested in Chapter 4, but promising potential future work could include a training curriculum approach in

which agents are trained with the reward structure described in Section 4.3.5 for a few hundred episodes and then trained with a less-biased reward structure for another few hundred episodes, or alternates reward structures every episode. I predict that this would train the agents to be somewhat more evasive and conservative, but depending on how the trained agents would be deployed, more-evasive agents may be beneficial (e.g. when conducting missions that are primarily intended for reconnaissance to learn about opponents over opponent territory rather than to intercept and attrit opponents). In addition to the mutual information reward scheme concept presented earlier, incorporating a small reward for agents that choose to coordinate with a partner, and particularly partnered agents who select the same partner agents multiple timesteps in a row, is a potential area of future work experimentation. With the current PSCE neural network architecture, the PSCE agent does not have any way of letting its partner in the previous timestep influence its current timestep decision. To equip agents with the information they need to learn whether coordinating with the same partner for multiple timesteps in a row earns additional reward for the team, I propose adding a masking layer to the state representation to mark the ego agent and, if the ego agent selected DA in the previous timestep, to mark its partner in the previous timestep.

6.2 Multirotor Approach: Addressing Limitations and Future Directions

In Chapter 5, I introduce a bio-inspired defense scheme that one could imagine being utilized as the last line of defense of a secure location. Based on the success of PSCE agents in learning to switch between DA and GS behaviors, I propose potential future work in applying PSCE's paired state representation decisionmaking in application to selecting guard roles in the bio-inspired defense framework of Chapter 5. The experiments presented in Chapter 5 assume that the number of each type of adversary attempting to breach the HVT is static for a given engagement scenario, but in real-world application, that assumption may not always hold— enemies may send reinforcements. Enabling agents to learn to switch between performing hovering guard and standing guard roles, and to learn when to signal for reinforcements from the HVT, would allow for much greater flexibility in this defense scheme. With the defending agents staying in the general vicinity of

the location they are protecting, and all staying relatively close to one another, the defenders could rely on their own individual sensing to construct the state representations of themselves and of their teammates and could use visual signals to perform any necessary communication (e.g. to request reinforcements from the "nest" when guards are attrited). Thus, this close-in defense scheme is suitable for comms-denied scenarios, so long as the UAVs have sufficient sensor coverage around themselves. As the guards for the bio-inspired scheme must regulate their fuel level, if a PSCE-like scheme were to be used to enhance the bio-inspired defense scheme, an agent's fuel level could be incorporated as a part of its state representation to assist in making decisions about what role to take, when to refuel, and when to signal for reinforcements. The PSCE paired state representation evaluation formulation for action selection is especially relevant when including an agent's fuel level in its state representation—guards could leverage this knowledge of their own and their fellow guards' fuel levels to decide who should refuel when so as to maintain sufficient guard coverage during each guard's refueling, though in a comms-denied environment, an agent would need to either incorporate only an inexact estimate of its potential partner's fuel level based on what could be inferred from the teammate's flight time and standing time, caloric burn rate, and perhaps some inexact fine-tuning of this estimate from visual signals from the teammate in question with regard to its fuel level. With that difficulty, rather than including an agent's exact (or inexact) fuel level in its state representation, I propose, for deployment in a comms-denied environment, that the state representation of an agent instead be augmented by three values: the most recent time the agent refueled, how many seconds of hovering guard duty the agent has performed since their most recent refueling, and how many seconds of standing guard duty the agent has performed since their most recent refueling. This arrangement of including hovering and standing times and refuel timestamps in each agent's state representation makes the assumptions that an agent that refuels will always collect its maximum amount of fuel from the HVT (and that some entity aside from the guards guarantees that the HVT contains adequate fuel to refuel the deployed guards), and, for the purposes of arranging coordinated agent role allocation and refueling trips, the fuel consumption rates are consistent across all guard agents, with allowance for the different fuel consumption rates

of hovering guard duty and standing guard duty. As each agent is also evaluating its own state representation in the process of evaluating those of its teammates, however, I propose that each agent have emergency override behaviors that are dependent upon a that fellow guards could not visually discern—true fuel level. If an agent's internal sensing of its fuel level is inconsistent with how long the agent is able to perform either hovering or standing guard duties, the agent would visually signal that it was unavailable for partnership to its teammates, then execute its emergency-go-home behavior to return to its deployers for repair. If the assumption that the entities deploying the UAVs maintain sufficient fuel in the HVT is dropped and an agent recognizes that it was not able to fully refuel from the HVT (which would indicate that the HVT itself needed refueling), the agent would, again, visually signal to its teammates that it was unavailable for partnership evaluation, then return to safety inside the HVT, where it presumably could communicate the emergency lack of fuel to the entities deploying the UAVs.

6.3 Conclusions

Overall, while there yet remains a large number of improvements that could be made to prepare PSCE agents to be trainable and testable in live flight, there are paths forward to address each of these limitations in future work. The enhancements detailed in this chapter pave the way forward to equipping future PSCE agents to be evaluated and eventually deployed in practical application.

In this dissertation, I have introduced defense schemes for intercepting and attriting far-off swarms of adversary fixed-wing UAVs with a swarm of fixed-wing UAVs, as well as for the fuel-efficient close-in defense of a secure location with a heterogeneous team of multirotors. These schemes have only been evaluated in simulation, but have shown themselves to be effective within that framework. While the practical limitations separating these defense schemes from deployment in live-flight and even live-fire environments are not trivial, I believe that the future work to make the defensive arrangements presented in this dissertation to be achievable, and well worth the effort.

CHAPTER 7
CONCLUSIONS

7.1 Contributions

In this dissertation, I explore tactics and coordination strategies for engagements between fixed-wing UAVs and how best to leverage existing coordinating tactics in aerial combat scenarios.

My investigation begins in the realm of aerial combat engagements between teams of fixed-wing UAVs (in Chapter 3 [13]), where I introduce two aerial combat tactical behaviors that are inspired by tactics employed by human fighter pilots [14]. My primary contribution in Chapter 3 is showing that the two tactical behaviors are both effective in various sizes of engagements, but are vulnerable when employed in scenarios for which they were not designed. I establish that the pairwise-coordinated maneuvers of DA are better suited for countering isolated opponents and are more dependent on weapon quality, while GS is most effective in dense engagement scenarios and is less affected by low weapon quality.

As I establish in Chapter 3, GS and DA are both effective aerial combat tactical behaviors, but are suited to different scenarios. Thus, in Chapter 4, I employ a deep-RL framework with a novel input structure and novel output processing to equip agents to decide when in an engagement to employ GS, when (and in coordination with which teammate) to leverage DA, and when to maneuver at some specified yaw rate rather than operating under pre-scripted tactics. My primary contribution in Chapter 4 is the demonstration of these trained agents showing improvements in own-team survival and opponent attrition over the individual tactics between which they learn to select. I show that these agents learn through training how and when to switch tactical behaviors to create and exploit advantageous force concentration against their adversaries. The trained agents effectively split duties within their team between countering the nearest, most threatening opponents on the opposing team and spreading their fire across the opposing team's farther-away,

less-threatening opponents so the far-away opponents are more likely to be attrited before they can come closer and become a greater threat.

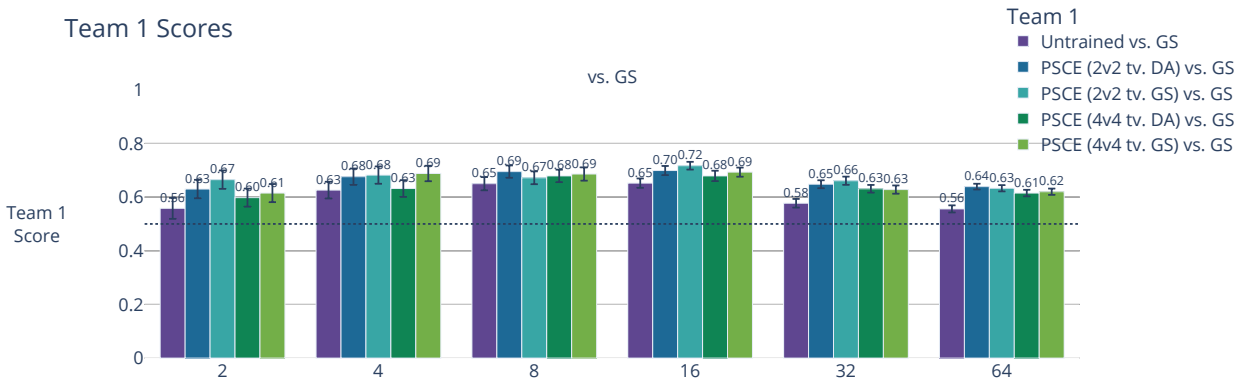
Finally, in Chapter 5 [15], I introduce a bio-inspired close-in defense scenario between heterogeneous teams of multirotors. My primary contribution in Chapter 5 is demonstrating through the experiments detailed therein that the emphasis and conclusions on force concentration and spread across opponents detailed in the experiments of Chapter 3 [13] and Chapter 4 are also important with respect to site-defense with a heterogeneous team of multirotors against a heterogeneous team of attackers attempting to steal the resources the guard force need to continue operating. I show that the maximizing guard schedules in this bio-inspired defense scheme prioritize force concentration advantage against especially-threatening opponents, but also demonstrate that it is still prudent to continue devoting resources to defending against lower-threat opponents. Despite their greater cost to the resource the defending team is attempting to maximize, employing high-cost guards that are specifically tasked with countering high-penalty attackers helps to maximize the resource the defenders are guarding, particularly due to the high-cost guards' ability to re-engage escaped high-penalty attackers before these attackers reach the HVT—a Lanchester's Square Law advantage. This tactical advantage against the most threatening attackers is to the defenders' benefit, but in some scenarios, especially when the opposing team is large and comprised of many of both low-threat and high-threat attackers, the less-expensive guards that defend against lower-penalty attackers are still important to deploy and maintain in order to maximize the guarded fuel resource.

Overall, I have shown the benefits of locally-advantageous force concentration in swarm-vs.-swarm scenarios, both for long-distance intercept and engagement of fixed-wing UAVs as well as for close-in defense of a protected location with a heterogeneous team of multirotors. A number of real-world practical hurdles separate these swarm-vs.-swarm defensive strategies from immediate deployment in live-flight and live-fire scenarios, but my demonstration of these tactics and strategies in action and my elucidation of the paths forward that future work could take shows the viability and utility of such defense schemes for the swarm-vs.-swarm aerial combat scenarios of the future.

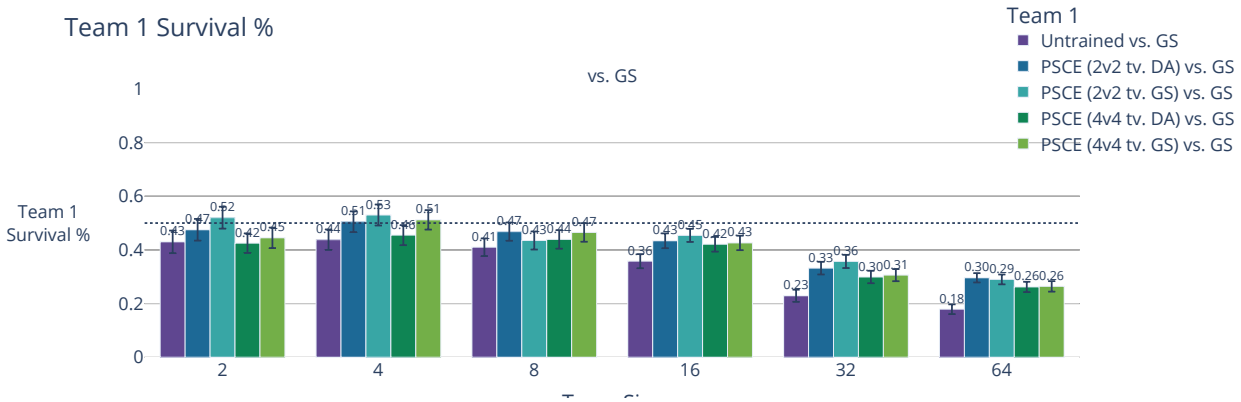
APPENDIX A
PSCE PERFORMANCE AGAINST OPPONENTS AGAINST WHICH PSCE AGENTS DID NOT TRAIN

In Chapter 4, I showed the performance of untrained and trained PSCE agents against teams consisting of agents operating under the hand-crafted tactical behavior against which the trained agents trained. Here, I present those results along with results from tests in which agents that were trained against a team employing one behavior are tested against a team employing a different behavior.

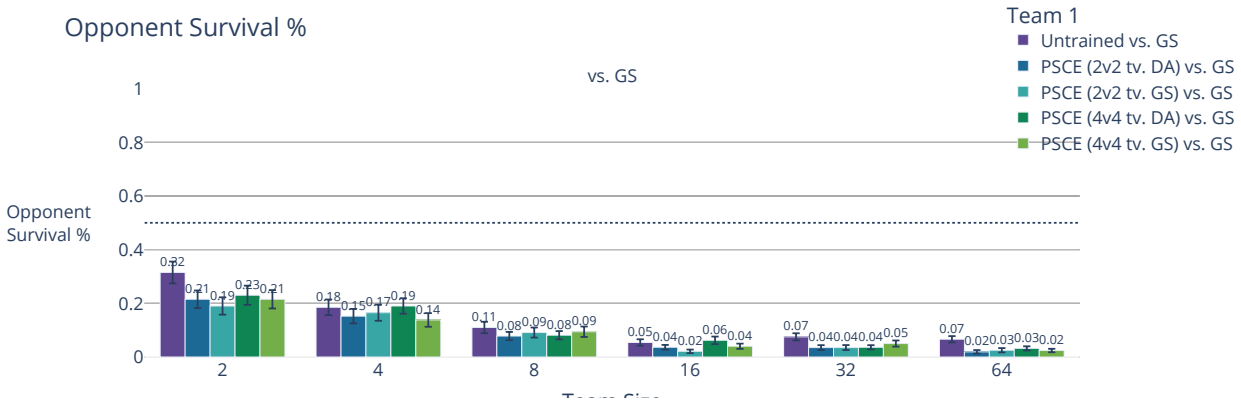
Against GS, the trained PSCE teams—even those that did not train against GS—outscore the untrained teams across the various cases tested, as shown in Figure A.1. This is especially noticeable in the 16-vs.-16 engagements, both for differences in own-team survival and opponent-team attrition. The trained PSCE agents mostly outscored the untrained agents in engagements against DA agents, as shown in Figure A.2, but, as noted in the previous section, the trained teams all find DA more difficult to outscore, survive against, and attrit than GS. From Figures A.1 and A.2, it is apparent that not only are the trained teams somewhat more effective than untrained teams against teams employing the tactic against which the trained teams trained, but they are also well-trained in good tactics, as they can counter opponents that they did not encounter during training more effectively than untrained PSCE agents. Interestingly, in the smaller engagements shown in Figure A.2, the 2-vs.-2-trained PSCE teams—both those trained against GS and against DA—scored more highly than the 4-vs.-4-trained teams, despite smaller engagements being where one would expect DA teams to be more difficult to defeat. I again postulate that these 2-vs.-2-trained teams, in training in the less-dense training engagements, learned good general tactics for such small engagements that proved to be more effective than the tactics of DA.



(a) Average score of untrained, trained teams against teams of GS in N-vs.-N engagements $\forall$ N $\in 2, 4, 8, 16, 32, 64$.

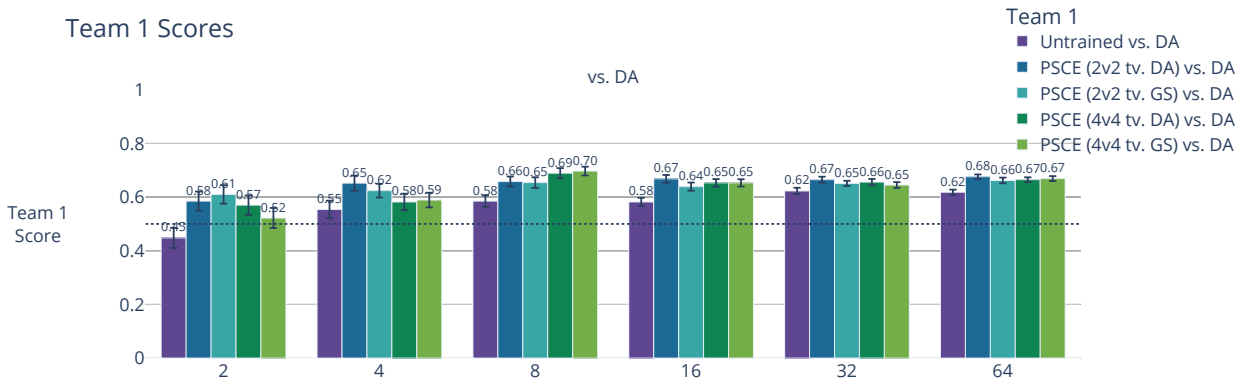


(b) Average percentage of untrained, trained teams that survive in engagements against teams of GS in N-vs.-N engagements $\forall$ N $\in 2, 4, 8, 16, 32, 64$.

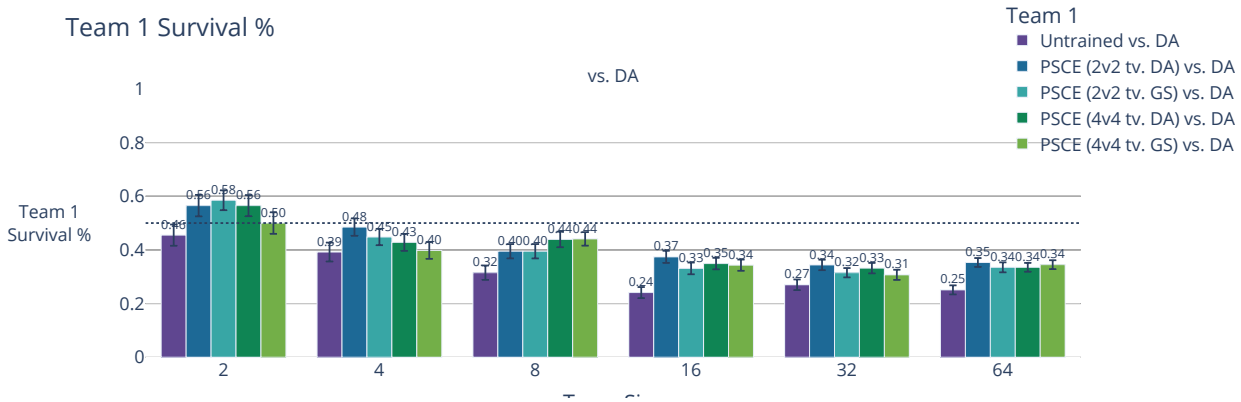


(c) Average Percentage of GS teams that survive in engagements against teams of untrained or trained PSCE agents in N-vs.-N engagements $\forall$ N $\in 2, 4, 8, 16, 32, 64$.

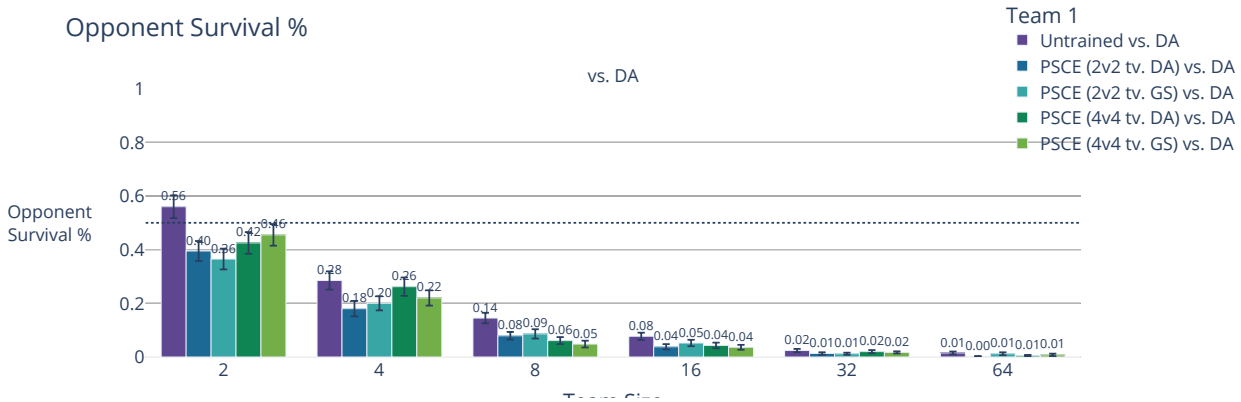
Figure A.1: These plots show the scores of PSCE agents, with the training (or lack thereof) denoted in the legend, in engagements against GS teams.



(a) Average score of untrained, trained teams against teams of DA in N-vs.-N engagements $\forall$ N $\in 2, 4, 8, 16, 32, 64$.



(b) Average percentage of untrained, trained teams that survive in engagements against teams of DA in N-vs.-N engagements $\forall$ N $\in 2, 4, 8, 16, 32, 64$.



(c) Average Percentage of DA teams that survive in engagements against teams of untrained or trained PSCE agents in N-vs.-N engagements $\forall$ N $\in 2, 4, 8, 16, 32, 64$.

Figure A.2: These plots show the scores of PSCE agents, with the training (or lack thereof) denoted in the legend, in engagements against DA teams.

APPENDIX B

ASSORTED SCRIMMAGE MISSION FILES

This appendix contains example SCRIMMAGE mission files utilized in the experiments detailed in Chapter 3 and Chapter 4.

B.1 Sample Mission Files From Baseline Tactic (DA and GS) Experiments

B.1.1 Two DA agents vs. one GS agent

Example script used to generate results shown in Section 3.4.1:

```
<?xml version="1.0"?>
<runscript xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  name="Fixed␣Wing␣Experiment">

  <run
  start="0.0"
  end="7200"
  motion_multiplier="12"
  dt="0.1"
  start_paused="false"
  time_warp="1"/>

  <camera
  mode="free"
  follow_id="4"
  pos="0,1,3000"
  focal_point="0,0,0"
  show_fps="false"/>

  <log_dir>~/.scrimmage/logs/da2vga1</log_dir>
  <output_type>seed,mission,summary</output_type>

  <end_condition>time, one_team</end_condition>

  <network>GlobalNetwork</network>
  <network>LocalNetwork</network>
```

```
<entity_interaction
  fire_rate_max="2"
  fire_rounds_max="-1"
  fire_range_max="100"
  hit_detection="beta_weapon_model"
  >FiringInteraction</entity_interaction>
<entity_interaction type="cuboid"
                    lengths="1000,␣1000,␣1010"
                    center="0,␣0,␣500"
                    rpy="0,␣0,␣0"
                    >Boundary</entity_interaction>
<entity_interaction
  startup_collisions_only="true">SimpleCollision</
     entity_interaction>
<entity_interaction>DoubleAttackPartners</entity_interaction>
<entity_interaction>SeedOutput</entity_interaction>

<metrics>SimpleCollisionMetrics</metrics>
<metrics>FiringMetrics</metrics>
<metrics>SeedInCSV</metrics>

<entity_common name="all">
  <z>1000</z>
  <health>1</health>
  <use_variance_all_ents>true</use_variance_all_ents>
  <variance_x>1000</variance_x>
  <variance_y>2000000</variance_y>
  <altitude>999</altitude>
  <controller>SimpleAircraftControllerPID</controller>
  <motion_model max_roll="45">SimpleAircraft</motion_model>
  <script_name>rascal_piedmont.xml</script_name>
</entity_common>

<entity entity_common="all">
  <team_id>1</team_id>
  <color>0 0 255</color>
  <count>2</count>
  <x>-400</x>
  <y>0</y>
  <autonomy dummy="0"
    sensing_range="1000"
    fire_range_max="100"
    beta="100"
    enable_prop_nav="true"
```

```
      fire_FOV="3"
      vel_cruise="18.5"
      vel_max="18.5"
      enable_boundary_control="true"
      use_beta_weapon_model="true"
      >DoubleAttack_SimpleAircraft_PropNavCounter</autonomy>
    <heading>0</heading>
    <visual_model>zephyr-blue</visual_model>
  </entity>

  <entity entity_common="all">
    <team_id>2</team_id>
    <color>238 0 0</color>
    <count>1</count>
    <x>400</x>
    <y>0</y>
    <autonomy dummy="0"
      sensing_range="1000"
      fire_range_max="100"
      beta="750"
      enable_prop_nav="true"
      fire_FOV="3"
      speed="18.5"
      enable_boundary_control="true"
      use_beta_weapon_model="true"
      pct_greedy="1"
      >GreedyShooter</autonomy>
    <heading>180</heading>
    <visual_model>zephyr-red</visual_model>
  </entity>
</runscript>
```

B.1.2 N DA agents vs. N GS agents

Example script used to generate results shown in Section 3.4.2:

```
<?xml version="1.0"?>
<runscript xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  name="Fixed␣Wing␣Experiment">

  <run dt="0.1"
  enable_gui="true"
  end="7200"
  motion_multiplier="12"
  start="0.0"
  start_paused="true"
  time_warp="0"/>

  <camera mode="free"
  follow_id="4"
  pos="0,1,3000"
  focal_point="0,0,0"
  show_fps="false"/>

  <show_plugins>false</show_plugins>
  <log_dir>~/.scrimmage/logs</log_dir>
  <output_type>seed,mission,summary</output_type>

  <end_condition>time, one_team</end_condition>

  <entity_interaction
    fire_rate_max="2"
    fire_rounds_max="-1"
    fire_range_max="100"
    hit_detection="beta_weapon_model"
    >RLInteraction</entity_interaction>
  <entity_interaction type="cuboid"
                      lengths="10000,␣10000,␣1010"
                      center="0,␣0,␣500"
                      rpy="0,␣0,␣0"
                      >Boundary</entity_interaction>
  <entity_interaction
    startup_collisions_only="true">SimpleCollision</
       entity_interaction>
  <entity_interaction>DoubleAttackPartners</entity_interaction>
  <entity_interaction>SeedOutput</entity_interaction>
```

```
<metrics>SimpleCollisionMetrics</metrics>
<metrics>SimpleCollisionMetrics</metrics>
<metrics>TheRLMetrics</metrics>
<metrics>SeedInCSV</metrics>

<entity_common name="all">
  <z>1000</z>
  <health>1</health>
  <use_variance_all_ents>true</use_variance_all_ents>
  <variance_x>1000</variance_x>
  <variance_y>2000000</variance_y>
  <altitude>999</altitude>
  <controller>SimpleAircraftControllerPID</controller>
  <motion_model max_roll="45">SimpleAircraft</motion_model>
  <script_name>rascal_piedmont.xml</script_name>
</entity_common>

<entity entity_common="all">
  <team_id>1</team_id>
  <color>0 0 255</color>
  <count>10</count>
  <x>-4000</x>
  <y>0</y>
  <autonomy dummy="0"
    sensing_range="1000"
    fire_range_max="100"
    beta="1000"
    enable_prop_nav="true"
    fire_FOV="3"
    vel_cruise="18.5"
    vel_max="18.5"
    enable_boundary_control="true"
    use_beta_weapon_model="true"
    >DoubleAttack_SimpleAircraft_PropNavCounter</autonomy>
  <heading>0</heading>
  <visual_model>zephyr-blue</visual_model>
</entity>

<entity entity_common="all">
  <team_id>2</team_id>
  <color>238 0 0</color>
  <count>10</count>
  <x>4000</x>
  <y>0</y>
  <autonomy dummy="0"
```

```
      sensing_range="1000"
      fire_range_max="100"
      beta="1000"
      enable_prop_nav="true"
      fire_FOV="3"
      speed="18.5"
      enable_boundary_control="true"
      use_beta_weapon_model="true"
      pct_greedy="1"
      >GreedyShooter</autonomy>
    <heading>180</heading>
    <visual_model>zephyr-red</visual_model>
  </entity>
</runscript>
```

B.2 Sample Mission Files From PSCE Experiments

B.2.1 Sample Mission File From Training PSCE Agents Against DA Agents

Note that the mission file included in this section cannot be run with a basic compilation of SCRIMMAGE; SCRIMMAGE must be compiled with Python bindings, and each engagement must be run from within a Python script that manually steps the SCRIMMAGE simulation.

Example script used to generate results shown in Section 4.5.1:

```
<?xml version="1.0"?>
<runscript xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  name="Fixed␣Wing␣Experiment">

  <run
  start="0.0"
  end="1000"
  motion_multiplier="12"
  dt="0.1"
  time_warp="1"
  start_paused="false"/>

  <camera
  mode="free"
  follow_id="4"
  pos="0,1,3000"
  focal_point="0,0,0"
  show_fps="false"/>

  <end_condition>time</end_condition>

  <network>GlobalNetwork</network>
  <network>LocalNetwork</network>

  <output_type>frames,summary,git_commits</output_type>
  <metrics>OpenAIRewards</metrics>
  <metrics>TheRLMetrics</metrics>
  <log_dir>~/.scrimmage/logs</log_dir>

  <entity_interaction
    fire_rate_max="2"
    fire_rounds_max="-1"
    fire_range_max="100"
```

```
  dead_altitude_min="9000"
  max_in_range_alt="1005"
  min_in_range_alt="-5"
  hit_detection="beta_weapon_model"
  >RLInteraction</entity_interaction>
<entity_interaction id="1"
                    team_id="3"
                    type="cuboid"
                    lengths="1000,␣1000,␣1010"
                    center="0,␣0,␣500"
                    rpy="0,␣0,␣0"
                    >Boundary</entity_interaction>
<entity_interaction startup_collisions_only="true"
  >SimpleCollision</entity_interaction>
<entity_interaction>SeedOutput</entity_interaction>

<entity_interaction da_team="2">DARoombaPartnersRL</
   entity_interaction>

<metrics>SimpleCollisionMetrics</metrics>
<metrics>SeedInCSV</metrics>

<entity_common name="all">
  <z>1000</z>
  <health>1</health>
  <use_variance_all_ents>true</use_variance_all_ents>
  <variance_x>1000</variance_x>
  <variance_y>10000</variance_y>
  <altitude>999</altitude>
  <script_name>rascal_piedmont.xml</script_name>
</entity_common>

<entity entity_common="all">
  <x>-200</x>
  <y>0</y>
  <team_id>1</team_id>
  <count>4</count>
  <color>77 77 255</color>
  <heading>0</heading>
  <controller
    heading_pid="0.35,␣0.0001,␣4.5,␣9"
    >SimpleAircraftControllerPID</controller>
  <motion_model max_roll="45">SimpleAircraft</motion_model>
  <autonomy
    reject_death="true"
```

```
    sensing_range="1000"
    fire_range_max="100"
    beta="1000"
    enable_prop_nav="true"
    fire_FOV="3"
    vel_cruise="18.5"
    enable_boundary_control="true"
    use_beta_weapon_model="true"
    radius="2"
    prop_nav_gain="5.0"
    prop_nav_dist_ahead="2.0"
    hdg_perturbation_max="0.05"
    da_vel_min="11"
    da_vel_max="18.5"
    fire_2D_mode="true"
    bank_max="45"
    dead_altitude_min="9000"
    max_dt_out_of_bounds="30"
    max_in_range_alt="1005"
    min_in_range_alt="-5"
    da_offense_start_dist="5.0"
    da_offense_sep_dist="2.10"
    da_offense_sep_time="0.1"
    da_offense_sep_time2="2.5"
    da_offense_engage_time="60"
    da_offense_sep_ang_rad="0.20"
    da_offense_bracket_max_sep_dist="2.0"
    da_max_dist_from_bogey="750"
    da_pack_separation="200"
    da_defense_sep_dist="2"
    da_defensive_engage_time="60"
    da_defensive_engage_dist="100"
    da_defensive_fire_rng_coe="4"
    da_defense_sep_dist_err="20"
    da_flee_angle="0.1"
    da_vel_gain="0.9"
    >PSCE</autonomy>

  <sensor>RLFeaturesSensor</sensor>
  <visual_model visual_scale="0.00658892">zephyr-blue</
     visual_model>
</entity>

<entity entity_common="all">
  <x>200</x>
```

```
<y>0</y>
<team_id>2</team_id>
<count>4</count>
<color>255 77 77</color>
<heading>-180</heading>
<controller
  heading_pid="0.35,␣0.0001,␣4.5,␣9"
  >SimpleAircraftControllerPID</controller>
<motion_model max_roll="45">SimpleAircraft</motion_model>
<autonomy
  reject_death="true"
  sensing_range="1000"
  fire_range_max="100"
  beta="1000"
  enable_prop_nav="true"
  fire_FOV="3"
  vel_cruise="18.5"
  enable_boundary_control="true"
  use_beta_weapon_model="true"
  radius="2"
  prop_nav_gain="5.0"
  prop_nav_dist_ahead="2.0"
  hdg_perturbation_max="0.05"
  vel_min="11"
  vel_max="18.5"
  fire_2D_mode="true"
  bank_max="45"
  dead_altitude_min="9000"
  max_dt_out_of_bounds="30"
  max_in_range_alt="1005"
  min_in_range_alt="-5"
  heading_noise_variance="0.0"
  offense_start_dist="5.0"
  offense_sep_dist="2.10"
  offense_sep_time="0.1"
  offense_sep_time2="2.5"
  offense_engage_time="60"
  offense_sep_dist_err="10"
  offense_sep_ang_rad="0.20"
  offense_bracket_max_sep_dist="2.0"
  max_dist_from_bogey="750"
  pack_separation="200"
  defense_sep_dist="2"
  defensive_engage_time="60"
  defensive_engage_dist="100"
```

```
      defensive_fire_rng_coe="4"
      defense_sep_dist_err="20"
      flee_angle="0.1"
      vel_gain="0.9"
      >DoubleAttack_SimpleAircraft_PropNavCounter</autonomy>
    <visual_model visual_scale="0.00658892">zephyr-red</
       visual_model>
  </entity>

</runscript>
```

B.2.2 Sample Mission File From Experiment Between 16 PSCE (4v4 tv. DA) Trained Agents Against 16 GS Agents

Note that the mission file included in this section cannot be run with a basic compilation of SCRIMMAGE; SCRIMMAGE must be compiled with Python bindings, and each engagement must be run from within a Python script that manually steps the SCRIMMAGE simulation.

Example script used to generate results shown in Section 4.5.2:

```
<?xml version="1.0"?>
<runscript xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  name="Fixed␣Wing␣Experiment">

  <run
  start="0.0"
  end="1000"
  motion_multiplier="12"
  dt="0.1"
  time_warp="1"
  start_paused="false"/>

  <camera
  mode="free"
  follow_id="4"
  pos="0,1,3000"
  focal_point="0,0,0"
  show_fps="false"/>

  <end_condition>time</end_condition>

  <network>GlobalNetwork</network>
  <network>LocalNetwork</network>

  <output_type>frames,summary,git_commits</output_type>
  <metrics>OpenAIRewards</metrics>
  <metrics>TheRLMetrics</metrics>
  <log_dir>~/.scrimmage/logs</log_dir>

  <entity_interaction
    fire_rate_max="2"
    fire_rounds_max="-1"
    fire_range_max="100"
    dead_altitude_min="9000"
```

```
  max_in_range_alt="1005"
  min_in_range_alt="-5"
  hit_detection="beta_weapon_model"
  >RLInteraction</entity_interaction>
<entity_interaction id="1"
                    team_id="3"
                    type="cuboid"
                    lengths="1000, 1000, 1010"
                    center="0, 0, 500"
                    rpy="0, 0, 0"
                    >Boundary</entity_interaction>
<entity_interaction startup_collisions_only="true"
  >SimpleCollision</entity_interaction>
<entity_interaction>SeedOutput</entity_interaction>

<metrics>SimpleCollisionMetrics</metrics>
<metrics>SeedInCSV</metrics>

<entity_common name="all">
  <z>1000</z>
  <health>1</health>
  <use_variance_all_ents>true</use_variance_all_ents>
  <variance_x>1000</variance_x>
  <variance_y>10000</variance_y>
  <altitude>999</altitude>
  <script_name>rascal_piedmont.xml</script_name>
</entity_common>

<entity entity_common="all">
  <x>-200</x>
  <y>0</y>
  <team_id>1</team_id>
  <count>16</count>
  <color>77 77 255</color>
  <heading>0</heading>
  <controller
    heading_pid="0.35, 0.0001, 4.5, 9"
    >SimpleAircraftControllerPID</controller>
  <motion_model max_roll="45">SimpleAircraft</motion_model>
  <autonomy
    flip_pos_on_init="true"
    reject_death="true"
    sensing_range="1000"
    fire_range_max="100"
    beta="1000"
```

```
    enable_prop_nav="true"
    fire_FOV="3"
    vel_cruise="18.5"
    enable_boundary_control="true"
    use_beta_weapon_model="true"
    radius="2"
    prop_nav_gain="5.0"
    prop_nav_dist_ahead="2.0"
    hdg_perturbation_max="0.05"
    da_vel_min="11"
    da_vel_max="18.5"
    fire_2D_mode="true"
    bank_max="45"
    dead_altitude_min="9000"
    max_dt_out_of_bounds="30"
    max_in_range_alt="1005"
    min_in_range_alt="-5"
    da_offense_start_dist="5.0"
    da_offense_sep_dist="2.10"
    da_offense_sep_time="0.1"
    da_offense_sep_time2="2.5"
    da_offense_engage_time="60"
    da_offense_sep_ang_rad="0.20"
    da_offense_bracket_max_sep_dist="2.0"
    da_max_dist_from_bogey="750"
    da_pack_separation="200"
    da_defense_sep_dist="2"
    da_defensive_engage_time="60"
    da_defensive_engage_dist="100"
    da_defensive_fire_rng_coe="4"
    da_defense_sep_dist_err="20"
    da_flee_angle="0.1"
    da_vel_gain="0.9"
    >PSCE</autonomy>

  <sensor>RLFeaturesSensor</sensor>
  <visual_model visual_scale="0.00658892">zephyr-blue</
     visual_model>
</entity>

<entity entity_common="all">
  <x>200</x>
  <y>0</y>
  <team_id>2</team_id>
  <count>16</count>
```

```
    <color>255 77 77</color>
    <heading>-180</heading>
    <controller
      heading_pid="0.35, 0.0001, 4.5, 9"
      >SimpleAircraftControllerPID</controller>
    <motion_model max_roll="45">SimpleAircraft</motion_model>
    <autonomy
      flip_pos_on_init="true"
      reject_death="true"
      sensing_range="1000"
      fire_range_max="100"
      beta="1000"
      enable_prop_nav="true"
      fire_FOV="3"
      speed="18.5"
      enable_boundary_control="true"
      use_beta_weapon_model="true"
      pct_greedy="1"
      prop_nav_gain="5.0"
      prop_nav_dist_ahead="2.0"
      hdg_perturbation_max="0.05"
      bank_max="45"
      fire_2D_mode="true"
      max_dt_out_of_bounds="30"
      dead_altitude_min="9000"
      max_in_range_alt="1005"
      min_in_range_alt="-5"
      heading_noise_variance="0.0"
      >GreedyShooter</autonomy>

    <visual_model visual_scale="0.00658892">zephyr-red</
       visual_model>
  </entity>

</runscript>
```

REFERENCES

[1] H. K. B. G. von Moltke, "Aufsatz vom Jahre 1871 'Über Strategie'," in *Kriegsgeschichtliche Abteilung II*, vol. II, Berlin: Ernst Siegfried Mittler und Sohn, 1900, pp. 291–293.

[2] T. H. Chung, M. R. Clement, M. A. Day, K. D. Jones, D. Davis, and M. Jones, "Live-fly, large-scale field experimentation for large numbers of fixed-wing UAVs," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 1255–1262.

[3] J. Lin and P. Singer. (Jan. 8, 2018). China Is Making 1,000-UAV Drone Swarms Now, Popular Science, [Online]. Available: https://www.popsci.com/china-drone-swarms/ (visited on 04/05/2022).

[4] M. A. Day, "Multi-Agent Task Negotiation Among UAVs to Defend Against Swarm Attacks," Thesis, Naval Postgraduate School, Monterey, CA, Mar. 2012.

[5] Raytheon. (Jun. 17, 2019). What to Do About Drones - The Tech That Will Protect Airports, Stadiums and Military Bases | Raytheon, Raytheon, [Online]. Available: https://www.raytheon.com/news/feature/what-do-about-drones (visited on 10/22/2019).

[6] A. Johnson and A. Blankstein. (May 15, 2019). Drone Pilot Charged With Violating Secure Airspace Over Two NFL Games, NBC News, [Online]. Available: https://www.nbcnews.com/news/crime-courts/drone-pilot-charged-violating-secure-airspace-over-two-nfl-games-n1006241 (visited on 10/22/2019).

[7] M. Laris. (May 11, 2018). Stadium and Team Owners See Drones as Major League Threat, Washington Post, [Online]. Available: https://www.washingtonpost.com/local/trafficandcommuting;nationalinclude;/stadium-and-team-owners-see-drones-as-major-league-threat/2018/05/10/83e0b954-50ad-11e8-84a0-458a1aa9ac0a_story.html (visited on 10/22/2019).

[8] B. Watson. (Oct. 18, 2018). Against the Drones: How to Stop Weaponized Consumer Drones, [Online]. Available: https://www.defenseone.com/feature/against-the-drones/ (visited on 10/22/2019).

[9] J. A. Gross. (Nov. 8, 2021). Iron Dome Shoots Down Hamas Drone Flown Out to Sea, The Times of Israel, [Online]. Available: https://www.timesofisrael.com/iron-dome-shoots-down-hamas-drone-flown-out-to-sea/ (visited on 04/05/2022).

[10] outreach@darpa.mil. (Aug. 26, 2020). AlphaDogfight Trials Foreshadow Future of Human-Machine Symbiosis, [Online]. Available: https://www.darpa.mil/news-events/2020-08-26 (visited on 05/23/2022).

[11] L. C. R. Hefron. (). Air Combat Evolution (ACE), [Online]. Available: https://www.darpa.mil/program/air-combat-evolution (visited on 05/23/2022).

[12] L. G. Strickland, C. E. Pippin, and M. Gombolay, "Learning to Steer Swarm-vs.-swarm Engagements," in *AIAA Scitech 2021 Forum*, VIRTUAL EVENT: American Institute of Aeronautics and Astronautics, Jan. 11, 2021, ISBN: 978-1-62410-609-5.

[13] L. G. Strickland, E. G. Squires, M. A. Day, and C. E. Pippin, "On Coordination in Multiple Aerial Engagement," in *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, Jun. 2019, pp. 557–562.

[14] R. L. Shaw, *Fighter Combat: Tactics and Maneuvering*. Naval Institute Press, 1985.

[15] L. G. Strickland, K. M. Baudier, K. P. Bowers, T. P. Pavlic, and C. E. Pippin, "Bio-Inspired Role Allocation of Heterogeneous Teams in a Site Defense Task," in *The 14th International Symposium on Distributed Autonomous Robotic Systems*, ser. Springer Proceedings in Advanced Robotics, vol. 9, Springer International Publishing, Oct. 15, 2018.

[16] F. W. Lanchester, *Aircraft in Warfare: The Dawn of the Fourth Arm*. Constable limited, 1916.

[17] J. G. Taylor, *Lanchester-Type Models of Warfare, Volume I*, 2 vols. Calhoun, 1980, vol. 1.

[18] P. M. Morse and G. E. Kimball, *Methods Of Operations Research*. The Technology Press of Massachusetts Institute of Technology, John Wiley and Sons, Inc., and Chapman And Hall Limited, 1951, 177 pp.

[19] G. H. Burgin, L. J. Fogel, and J. P. Phelps, "An Adaptive Maneuvering Logic Computer Program for the Simulation of One-on-One Air-to-Air Combat. Volume 1: General Description," NASA, Contractor Report CR-2582, Sep. 1975.

[20] G. H. Burgin and L. B. Sidor, "Rule-Based Air Combat Simulation," DTIC Document, 1988.

[21] R. M. Jones and J. E. Laird, "Constraints on the Design of a High-Level Model of Cognition," in *Proceedings of the Nineteenth Annual Conference of the Cognitive Science Society*, 1997, pp. 358–363.

[22] R. M. Jones, J. E. Laird, P. E. Nielsen, K. J. Coulter, P. Kenny, and F. V. Koss, "Automated Intelligent Pilots for Combat Flight Simulation," *AI Mag.*, vol. 20, no. 1, pp. 27–41, Mar. 15, 1999.

[23] F. Austin, G. Carbone, M. Falco, H. Hinz, and M. Lewis, "Automated Maneuvering Decisions for Air-to-Air Combat," in *Guidance, Navigation, and Control and Co-located Conferences*, American Institute of Aeronautics and Astronautics, 1987.

[24] ——, "Game Theory for Automated Maneuvering During Air-to-Air Combat," *J. Guid. Control Dyn.*, vol. 13, no. 6, pp. 1143–1149, Nov. 1, 1990.

[25] R. L. Spicer and L. G. Martin, "TACTICS II: Maneuver Logic for Computer Simulation of Dogfight Engagements," The RAND Corporation, Santa Monica CA, USA, Scientific Report R-979-PR, Jul. 1972.

[26] J. Sprinkle, J. M. Eklund, H. J. Kim, and S. Sastry, "Encoding Aerial Pursuit/Evasion Games with Fixed Wing Aircraft into a Nonlinear Model Predictive Tracking Controller," in *2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601)*, vol. 3, IEEE, 2004, pp. 2609–2614, ISBN: 978-0-7803-8682-2.

[27] J. M. Eklund, J. Sprinkle, and S. Sastry, "Implementing and Testing a Nonlinear Model Predictive Tracking Controller for Aerial Pursuit/Evasion Games on a Fixed Wing Aircraft," in *Proceedings of the American Control Conference*, IEEE, 2005, pp. 1509–1514, ISBN: 978-0-7803-9098-0.

[28] K. Virtanen, T. Raivio, and R. P. Hämäläinen, "Modeling Pilot's Sequential Maneuvering Decisions by a Multistage Influence Diagram," *J. Guid. Control Dyn.*, vol. 27, no. 4, pp. 665–677, 2004.

[29] K. Virtanen, J. Karelahti, and T. Raivio, "Modeling Air Combat by a Moving Horizon Influence Diagram Game," *J. Guid. Control Dyn.*, vol. 29, no. 5, pp. 1080–1091, 2006.

[30] D.-I. You and D. H. Shim, "Design of an Aerial Combat Guidance Law Using Virtual Pursuit Point Concept," *Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng.*, vol. 229, no. 5, pp. 792–813, Apr. 2015.

[31] R. Isaacs, *Differential Games*. New York: John Wiley and Sons, Inc., 1965, ISBN: 0-486-40682-2.

[32] J. V. Breakwell and A. W. Merz, "Minimum Required Capture Radius in a Coplanar Model of the Aerial Combat Problem," *AIAA J.*, vol. 15, no. 8, pp. 1089–1094, Aug. 1977.

[33] B. S. A. Järmark, A. W. Merz, and J. V. Breakwell, "The Variable-Speed Tail-Chase Aerial Combat Problem," *J. Guid. Control Dyn.*, vol. 4, no. 3, pp. 323–328, May 1981.

[34] C. Hillberg and B. Järmark, "Pursuit-Evasion Between Two Realistic Aircraft," *J. Guid. Control Dyn.*, vol. 7, no. 6, pp. 690–694, 1984.

[35] A. W. Merz, "To Pursue or to Evade—That is the Question," *J. Guid. Control Dyn.*, vol. 8, no. 2, pp. 161–166, Mar. 1, 1985.

[36] N. Greenwood, "A Differential Game in Three Dimensions: The Aerial Dogfight Scenario," *Dyn. Control*, vol. 2, no. 2, pp. 161–200, 1992.

[37] J. S. McGrew, "Real-time Maneuvering Decisions for Autonomous Air Combat," Massachusetts Institute of Technology, 2008.

[38] J. S. McGrew, J. P. How, B. Williams, and N. Roy, "Air-Combat Strategy Using Approximate Dynamic Programming," *J. Guid. Control Dyn.*, vol. 33, no. 5, pp. 1641–1654, Sep. 2010.

[39] X. Ma, L. Xia, and Q. Zhao, "Air-Combat Strategy Using Deep Q-Learning," in *2018 Chinese Automation Congress (CAC)*, Nov. 2018, pp. 3952–3957.

[40] L. G. Strickland, M. A. Day, K. J. DeMarco, E. G. Squires, and C. E. Pippin, "Responding to Unmanned Aerial Swarm Saturation Attacks with Autonomous Counter-Swarms," in *Ground/Air Multisensor Interoperability, Integration, and Networking for Persistent ISR IX*, vol. 10635, Orlando, Florida: SPIE, May 4, 2018, pp. 10635-10635 –17, ISBN: 978-1-5106-1781-0.

[41] B. Vlahov, E. Squires, L. Strickland, and C. Pippin, "On Developing a UAV Pursuit-Evasion Policy Using Reinforcement Learning," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Orlando, FL, USA, Dec. 18, 2018, pp. 859–864.

[42] R. E. Smith and B. A. Dike, "Learning Novel Fighter Combat Maneuver Rules via Genetic Algorithms," *Int. J. Expert Syst.*, vol. 8, no. 3, pp. 247–276, 1995.

[43] R. E. Smith, B. A. Dike, R. K. Mehra, B. Ravichandran, and A. El-Fallah, "Classifier Systems in Combat: Two-Sided Learning of Maneuvers for Advanced Fighter Aircraft," *Comput. Methods Appl. Mech. Eng.*, vol. 186, no. 2, pp. 421–437, Jun. 2000.

[44] R. E. Smith, B. A. Dike, B. Ravichandran, A. El-Fallah, and R. K. Mehra, "The Fighter Aircraft LCS: A Case of Different LCS Goals and Techniques," in *Learning Classifier Systems*, vol. 1813, Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 283–300, ISBN: 978-3-540-45027-6.

[45] R. E. Smith, B. A. Dike, B. Ravichandran, A. El-Fallah, and R. K. Mehra, "Discovering Novel Fighter Combat Maneuvers: Simulating Test Pilot Creativity," in *Creative Evolutionary Systems*, P. J. Bentley and D. W. Corne, Eds., San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2002, pp. 467–486, ISBN: 1-55860-673-4.

[46] R. E. Smith, A. El-Fallah, B. Ravichandran, R. K. Mehra, and B. A. Dike, "The Fighter Aircraft LCS: A Real-World, Machine Innovation Application," in *Applications of Learning Classifier Systems*, ser. Studies in Fuzziness and Soft Computing, vol. 150, Berlin, Heidelberg: Springer, Berlin, Heidelberg, 2004, pp. 113–142, ISBN: 978-3-540-39925-4.

[47] A. Toubman, J. J. Roessingh, P. Spronck, A. Plaat, and J. van den Herik, "Rapid Adaptation of Air Combat Behaviour," Netherlands Aerospace Centre (NLR) Aerospace Operations, The Netherlands, NLR-TP-2016-425, Nov. 2016.

[48] ——, "Dynamic Scripting with Team Coordination in Air Combat Simulation," in *Modern Advances in Applied Intelligence*, M. Ali, J.-S. Pan, S.-M. Chen, and M.-F. Horng, Eds., ser. Lecture Notes in Computer Science, vol. 8481, Cham: Springer, Cham, Jun. 3, 2014, pp. 440–449, ISBN: 978-3-319-07455-9.

[49] ——, "Centralized Versus Decentralized Team Coordination Using Dynamic Scripting," in *28th European Simulation and Modelling Conference - ESM '2014'*, EUROSIS, Oct. 2014.

[50] ——, "Rewarding Air Combat Behavior in Training Simulations," in *2015 IEEE International Conference on Systems, Man, and Cybernetics*, Oct. 2015, pp. 1397–1402, ISBN: 978-1-4799-8697-2.

[51] ——, "Improving Air-to-Air Combat Behavior Through Transparent Machine Learning," in *I/ITSEC 2014*, Creative Computing, 2014.

[52] ——, "Transfer Learning of Air Combat Behavior," in *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, Dec. 2015, pp. 226–231.

[53] A. Toubman, J. J. Roessingh, J. van Oijen, *et al.*, "Modeling Behavior of Computer Generated Forces with Machine Learning Techniques, the NATO Task Group Approach," in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Oct. 2016, pp. 001 906–001 911.

[54] A. Toubman, J. J. Roessingh, S. Pieter, A. Plaat, and J. van den Herik, "Rapid Adaptation of Air Combat Behaviour," in *22nd European Conference on Artificial Intelligence*, The Hague, Netherlands: Creative Computing, 2016.

[55] A. W. Merz, "The Homicidal Chauffeur - A Differential Game," PhD thesis, Stanford, Mar. 1971.

[56] ——, "The Homicidal Chauffeur," *AIAA J.*, vol. 12, no. 3, pp. 259–260, Mar. 1, 1974.

[57] ——, "The Game of Two Identical Cars," *J. Optim. Theory Appl.*, vol. 9, no. 5, pp. 324–343, May 1, 1972.

[58] J. Shinar and A. Davidovitz, "Pursit-Evasion Game Analysis in a Line of Sight Coordinate System," Feb. 1, 1985.

[59] ——, "A Two-Target Game Analysis in Line-of-Sight Coordinates," *Comput. Math. Appl.*, vol. 13, no. 1-3, pp. 123–140, 1987.

[60] H. J. Kelley and L. Lefton, "Differential Turns," *AIAA J.*, vol. 11, no. 6, pp. 858–861, 1973.

[61] B. A. S. Järmark, "Differential Dynamic Programming Techniques in Differential Games," in *Control and Dynamic Systems*, ser. Advances in Theory and Applications, C. T. Leondes, Ed., vol. 17, Academic Press, 1981, pp. 125–160, ISBN: 978-0-12-012717-7.

[62] F. Imado, "Some Aspects of a Realistic Three-Dimensional Pursuit-Evasion Game," *J. Guid. Control Dyn.*, vol. 16, no. 2, pp. 289–293, Mar. 1993.

[63] H. W. Kuhn, "The Hungarian Method for the Assignment Problem," *Nav. Res. Logist. Q.*, vol. 2, no. 1-2, pp. 83–97, Mar. 1955.

[64] G. G. denBroeder, R. E. Ellison, and L. Emerling, "On Optimum Target Assignments," *Operations Research*, vol. 7, no. 3, pp. 322–326, Jun. 1, 1959.

[65] S. Matlin, "A Review of the Literature on the Missile-Allocation Problem," *Operations Research*, vol. 18, no. 2, pp. 334–373, 1970. JSTOR: 168691.

[66] S. Hunt, Q. Meng, C. Hinde, and T. Huang, "A Consensus-Based Grouping Algorithm for Multi-agent Cooperative Task Allocation with Complex Requirements," *Cogn. Comput.*, vol. 6, no. 3, pp. 338–350, Sep. 2014.

[67] D. Palmer, M. Kirschenbaum, J. Murton, K. Zajac, M. Kovacina, and R. Vaidyanathan, "Decentralized Cooperative Auction for Multiple Agent Task Allocation Using Synchronized Random Number Generators," in *2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 2, IEEE, 2003, pp. 1963–1968, ISBN: 978-0-7803-7860-5.

[68] K. Volle, J. Rogers, and K. Brink, "Decentralized Cooperative Control Methods for the Modified Weapon–Target Assignment Problem," *J. Guid. Control Dyn.*, vol. 39, no. 9, pp. 1934–1948, 2016.

[69] K. Volle, J. D. Rogers, and K. Brink, "Scalable Cooperative Control Algorithms For the Weapon Target Assignment Problem," American Institute of Aeronautics and Astronautics, Jan. 4, 2016, ISBN: 978-1-62410-389-6.

[70] K. Volle and J. D. Rogers, "Simultaneous Arrival Control Algorithms for Weapon Target Assignment," American Institute of Aeronautics and Astronautics, Jan. 9, 2017, ISBN: 978-1-62410-450-3.

[71] K. Volle and J. Rogers, "Weapon–Target Assignment Algorithm for Simultaneous and Sequenced Arrival," *J. Guid. Control Dyn.*, vol. 41, no. 11, pp. 2361–2373, Sep. 2, 2018.

[72] U. Gaertner, "UAV Swarm Tactics: An Agent-based Simulation and Markov Process Analysis," Master's thesis, Naval Postgraduate School, Dept. of Operations Research, Monterey CA, 2013.

[73] N. D. Ernest, "Genetic Fuzzy Trees for Intelligent Control of Unmanned Combat Aerial Vehicles," PhD thesis, University of Cincinnati, Mar. 2015.

[74] N. D. Ernest and K. Cohen, "Fuzzy Logic Based Intelligent Agents for Unmanned Combat Aerial Vehicle Control," *J. Def. Manag.*, vol. 06, no. 139, pp. 2167–0374, 2015.

[75] N. D. Ernest, K. Cohen, E. Kivelevitch, C. Schumacher, and D. Casbeer, "Genetic Fuzzy Trees and their Application Towards Autonomous Training and Control of a Squadron of Unmanned Combat Aerial Vehicles," *Unmanned Syst.*, vol. 03, no. 03, pp. 185–204, Jul. 2015.

[76] N. Ernest, D. Carroll, C. Schumacher, M. Clark, K. Cohen, and G. Lee, "Genetic Fuzzy based Artificial Intelligence for Unmanned Combat Aerial Vehicle Control in Simulated Air Combat Missions," *J. Def. Manag.*, vol. 06, no. 144, pp. 2167–0374, Mar. 22, 2016.

[77] D. McIlroy and C. Heinze, "Air Combat Tactics Implementation in the Smart Whole Air Mission Model (SWARMM)," in *Proceedings of the First International SimTecT Conference*, 1996.

[78] G. Tidhar, M. C. Selvestrel, and C. Heinze, "Modelling Teams and Team Tactics in Whole Air Mission Modelling," in *IEA/AIE*, 1995, pp. 373–381.

[79] G. Tidhar, C. Heinze, and M. Selvestrel, "Flying Together: Modelling Air Mission Teams," *Appl. Intell.*, vol. 8, no. 3, pp. 195–218, 1998.

[80] J. E. Laird, R. M. Jones, O. M. Jones, and P. E. Nielsen, "Coordinated behavior of computer generated forces in TacAir-Soar," in *In Proceedings of the fourth conference on computer generated forces and behavioral representation*, 1994, pp. 325–332.

[81] M. Tambe and P. S. Rosenbloom, "Event Tracking in Complex Multi-Agent Environments," in *In Proceedings of the Fourth Conference on Computer Generated Forces and Behavioral Representation*, 1994, pp. 473–484.

[82] M. Tambe, "Tracking Dynamic Team Activity," in *AAAI/IAAI, Vol. 1*, 1996, pp. 80–87.

[83] M. Tambe, W. L. Johnson, R. M. Jones, *et al.*, "Intelligent Agents for Interactive Simulation Environments," *AI Mag.*, vol. 16, no. 1, p. 15, Mar. 15, 1995.

[84] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, *et al.*, Eds., Curran Associates, Inc., 2017, pp. 6379–6390.

[85] C. HolmesParker, A. Agogino, and K. Tumer, "Exploiting Structure and Utilizing Agent-Centric Rewards to Promote Coordination in Large Multiagent Systems," in *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems*, (St. Paul, MN, USA), ser. AAMAS '13, Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2013, pp. 1181–1182, ISBN: 978-1-4503-1993-5.

[86] X. Chu and H. Ye, "Parameter Sharing Deep Deterministic Policy Gradient for Cooperative Multi-agent Reinforcement Learning," *CoRR*, vol. abs/1710.00336, 2017.

[87] J. N. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, "Counterfactual Multi-Agent Policy Gradients," in *Thirty-Second AAAI Conference on Artificial Intelligence*, Apr. 29, 2018.

[88] T. N. Hoang, Y. Xiao, K. Sivakumar, C. Amato, and J. P. How, "Near-Optimal Adversarial Policy Switching for Decentralized Asynchronous Multi-Agent Systems," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 6373–6380.

[89] S. Omidshafiei, A.-a. Agha-mohammadi, C. Amato, S.-Y. Liu, J. P. How, and J. Vian, "Graph-Based Cross Entropy Method for Solving Multi-Robot Decentralized POMDPs," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 5395–5402.

[90] R. Zhang, Q. Zong, X. Zhang, L. Dou, and B. Tian, "Game of Drones: Multi-UAV Pursuit-Evasion Game With Online Motion Planning by Deep Reinforcement Learning," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–10, 2022.

[91] E. Seraj, Z. Wang, R. Paleja, M. Sklar, A. Patel, and M. Gombolay, "Heterogeneous Graph Attention Networks for Learning Diverse Communication," Oct. 28, 2021.

[92] Y. Niu, R. Paleja, and M. Gombolay, "Multi-Agent Graph-Attention Communication and Teaming," in *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, ser. AAMAS '21, Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, May 3, 2021, pp. 964–973, ISBN: 978-1-4503-8307-3.

[93] S. Konan, E. Seraj, and M. Gombolay, "Iterated Reasoning with Mutual Information in Cooperative and Byzantine Decentralized Teaming," Jan. 20, 2022.

[94] A. Radigales, *Combat Models: Lanchester's Laws*, Aug. 31, 2020.

[95] K. DeMarco, E. Squires, M. Day, and C. Pippin, "Simulating Collaborative Robots in a Massive Multi-Agent Game Environment (SCRIMMAGE)," in *International Symposium on Distributed Autonomous Robotic Systems*, N. Correll, M. Schwager, and M. Otte, Eds., ser. Springer Proceedings in Advanced Robotics, Cham: Springer International Publishing, 2018, pp. 283–297, ISBN: 978-3-030-05816-6.

[96] C. W. Reynolds, "Flocks, Herds and Schools: A Distributed Behavioral Model," *ACM SIGGRAPH Comput. Graph.*, vol. 21, no. 4, pp. 25–34, Aug. 1, 1987.

[97] P. D. Feigin, O. Pinkas, and J. Shinar, "A Simple Markov Model for the Analysis of Multiple Air Combat," *Nav. Res. Logist. Q.*, vol. 31, no. 3, pp. 413–429, Sep. 1984.

[98] R. J. Williams, "Simple Statistical Gradient-Following Algorithms For Connectionist Reinforcement Learning," *Mach Learn*, vol. 8, no. 3-4, pp. 229–256, May 1, 1992.

[99] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge: MIT Press, 2017, 1054 pp., ISBN: 978-0-262-19398-6. arXiv: 1603.02199.

[100] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Prentice Hall, 2010, ISBN: 978-0-13-604259-4.

[101] V. Nair and G. E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ser. ICML'10, Haifa, Israel: Omnipress, Jun. 21, 2010, pp. 807–814, ISBN: 978-1-60558-907-7.

[102] *TensorBoard*, TensorFlow, Jun. 21, 2021.

[103] D. J. T. Sumpter, J. Krause, R. James, I. D. Couzin, and A. J. W. Ward, "Consensus Decision Making by Fish," *Current Biology*, vol. 18, no. 22, pp. 1773–1777, Nov. 25, 2008.

[104] A. J. W. Ward, J. E. Herbert-Read, D. J. T. Sumpter, and J. Krause, "Fast and Accurate Decisions Through Collective Vigilance in Fish Shoals," *PNAS*, vol. 108, no. 6, pp. 2312–2315, Feb. 8, 2011. pmid: 21262802.

[105] J. E. Herbert-Read, A. Perna, R. P. Mann, T. M. Schaerf, D. J. T. Sumpter, and A. J. W. Ward, "Inferring the Rules of Interaction of Shoaling Fish," *PNAS*, vol. 108, no. 46, pp. 18 726–18 731, Nov. 15, 2011. pmid: 22065759.

[106] A. J. W. Ward, D. J. T. Sumpter, I. D. Couzin, P. J. B. Hart, and J. Krause, "Quorum Decision-Making Facilitates Information Transfer in Fish Shoals," *PNAS*, vol. 105, no. 19, pp. 6948–6953, May 13, 2008.

[107] J. Buhl, D. J. T. Sumpter, I. D. Couzin, *et al.*, "From Disorder to Order in Marching Locusts," *Science*, vol. 312, no. 5778, pp. 1402–1406, Jun. 2, 2006. pmid: 16741126.

[108] C. A. Yates, R. Erban, C. Escudero, *et al.*, "Inherent Noise Can Facilitate Coherence in Collective Swarm Motion," *PNAS*, vol. 106, no. 14, pp. 5464–5469, Apr. 7, 2009. pmid: 19336580.

[109] S. Wilson, T. P. Pavlic, G. P. Kumar, A. Buffin, S. C. Pratt, and S. Berman, "Design of Ant-Inspired Stochastic Control Policies for Collective Transport by Robotic Swarms," *Swarm Intell*, vol. 8, no. 4, pp. 303–327, Dec. 1, 2014.

[110] S. Berman, Q. Lindsey, M. S. Sakar, V. Kumar, and S. Pratt, "Study of Group Food Retrieval by Ants as a Model for Multi-Robot Collective Transport Strategies," presented at the International Conference on Robotics Science and Systems, RSS 2010, MIT Press Journals, 2011.

[111] D. J. T. Sumpter and M. Beekman, "From Nonlinearity to Optimality: Pheromone Trail Foraging by Ants," *Animal Behaviour*, vol. 66, no. 2, pp. 273–280, Aug. 1, 2003.

[112] M. Beekman, D. J. T. Sumpter, and F. L. W. Ratnieks, "Phase Transition Between Disordered and Ordered Foraging in Pharaoh's Ants," *PNAS*, vol. 98, no. 17, pp. 9703–9706, Aug. 14, 2001. pmid: 11493681.

[113] D. J. T. Sumpter and S. C. Pratt, "A Modelling Framework for Understanding Social Insect Foraging," *Behav Ecol Sociobiol*, vol. 53, no. 3, pp. 131–144, Feb. 1, 2003.

[114] G. P. Kumar, A. Buffin, T. P. Pavlic, S. C. Pratt, and S. M. Berman, "A Stochastic Hybrid System Model of Collective Transport in the Desert Ant *Aphaenogaster cockerelli*," in *Proceedings of the 16th international conference on Hybrid systems: computation and control - HSCC '13*, Philadelphia, Pennsylvania, USA: ACM Press, 2013, p. 119, ISBN: 978-1-4503-1567-8.

[115] G. T. Cooke, E. G. Squires, L. G. Strickland, *et al.*, "Bio-Inspired Nest-Site Selection for Distributing Robots in Low-Communication Environments," in *Highlights of Practical Appl. Agents, Multi-Agent Sys., and Complex.: The PAAMS Collection*, J. Bajo, J. M. Corchado, E. M. Navarro Martínez, *et al.*, Eds., ser. Communications in Computer and Information Science, Toledo, Spain: Springer, Cham, 2018, pp. 517–524, ISBN: 978-3-319-94779-2.

[116] S. Berman, Á. Halász, V. Kumar, and S. C. Pratt, "Bio-Inspired Group Behaviors for the Deployment of a Swarm of Robots to Multiple Destinations," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, Apr. 2007, pp. 2318–2323.

[117] M. A. Hsieh, Á. Halász, S. Berman, and V. Kumar, "Biologically Inspired Redistribution of a Swarm of Robots Among Multiple Sites," *Swarm Intell*, vol. 2, no. 2-4, pp. 121–141, Dec. 1, 2008.

[118] S. Berman, Á. Halász, V. Kumar, and S. Pratt, "Algorithms for the Analysis and Synthesis of a Bio-Inspired Swarm Robotic System," in *Swarm Robotics*, ser. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, Sep. 30, 2006, pp. 56–70, ISBN: 978-3-540-71541-2.

[119] Á. Halász, M. A. Hsieh, S. Berman, and V. Kumar, "Dynamic Redistribution of a Swarm of Robots Among Multiple Sites," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2007, pp. 2320–2325.

[120] S. C. Pratt, D. J. T. Sumpter, E. B. Mallon, and N. R. Franks, "An Agent-Based Model of Collective Nest Choice by the Ant *Temnothorax albipennis*," *Animal Behaviour*, vol. 70, no. 5, pp. 1023–1036, Nov. 1, 2005.

[121] S. C. Pratt, E. B. Mallon, D. J. Sumpter, and N. R. Franks, "Quorum Sensing, Recruitment, and Collective Decision-Making During Colony Emigration by the Ant *Leptothorax albipennis*," *Behav Ecol Sociobiol*, vol. 52, no. 2, pp. 117–127, Jul. 1, 2002.

[122] D. J. Sumpter and S. C. Pratt, "Quorum Responses and Consensus Decision Making," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 364, no. 1518, pp. 743–753, Mar. 27, 2009.

[123] B. Hölldobler, "Tournaments and Slavery in a Desert Ant," *Science*, vol. 192, no. 4242, pp. 912–914, 1976. JSTOR: 1742162.

[124] I. Scharf, T. Pamminger, and S. Foitzik, "Differential Response of Ant Colonies to Intruders: Attack Strategies Correlate With Potential Threat," *Ethology*, vol. 117, no. 8, pp. 731–739, Aug. 1, 2011.

[125] I. Scharf, O. Ovadia, and S. Foitzik, "The Advantage of Alternative Tactics of Prey and Predators Depends on the Spatial Pattern of Prey and Social Interactions Among Predators," *Popul. Ecol. Tokyo*, vol. 54, no. 1, pp. 187–196, Jan. 2012.

[126] E. I. Cash, "Proximate and Ultimate Mechanisms of Nestmate Recognition in Ants," PhD thesis, Arizona State University, United States – Arizona, 2016, 163 pp.

[127] M. W. Moffett, "Ants and the Art of War," *Sci. Am.*, vol. 305, no. 6, pp. 84–89, Nov. 15, 2011. JSTOR: 26002920.

[128] C. J. Lumsden and B. Hölldobler, "Ritualized Combat and Intercolony Communication in Ants," *J. Theor. Biol.*, vol. 100, no. 1, pp. 81–98, 1983.

[129] C. J. Tanner, "Numerical Assessment Affects Aggression and Competitive Ability: A Team-Fighting Strategy for the Ant *Formica xerophila*," *Proc. R. Soc. B Biol. Sci.*, vol. 273, no. 1602, pp. 2737–2742, Nov. 7, 2006.

[130] T. D. Seeley, R. H. Seeley, and P. Akratanakul, "Colony Defense Strategies of the Honeybees in Thailand," *Ecol. Monogr.*, vol. 52, no. 1, pp. 43–63, Feb. 1982.

[131] G. Kastberger, E. Schmelzer, and I. Kranner, "Social Waves in Giant Honeybees Repel Hornets," *PLOS ONE*, vol. 3, no. 9, H. Tanimoto, Ed., e3141, Sep. 10, 2008.

[132] I. Farkas, D. Helbing, and T. Vicsek, "Social Behaviour: Mexican Waves in an Excitable Medium," *Nature*, vol. 419, no. 6903, pp. 131–132, Sep. 12, 2002.

[133] G. Kastberger, F. Weihmann, M. Zierler, and T. Hötzl, "Giant Honeybees (*Apis dorsata*) Mob Wasps Away from the Nest by Directed Visual Patterns," *Naturwissenschaften*, vol. 101, no. 11, pp. 861–873, Nov. 2014.

[134] D. Wittmann, "Aerial Defense of the Nest by Workers of the Stingless Bee *Trigona (Tetragonisca) angustula* (Latreille)(Hymenoptera: Apidae)," *Behav. Ecol. Sociobiol.*, vol. 16, no. 2, pp. 111–114, 1985.

[135] M. H. Kärcher and F. L. W. Ratnieks, "Standing and Hovering Guards of the Stingless Bee *Tetragonisca angustula* Complement Each Other in Entrance Guarding and Intruder Recognition," *J. Apic. Res.*, vol. 48, no. 3, pp. 209–214, 2009.

[136] C. Grüter, M. H. Kärcher, and F. L. W. Ratnieks, "The Natural History of Nest Defence in a Stingless Bee *Tetragonisca angustula* (Latreille) (Hymenoptera: Apidae), with Two Distinct Types of Entrance Guards," *Neotrop. Entomol.*, vol. 40, no. 1, pp. 55–61, Feb. 2011.

[137] C. M. Jernigan, J. Birgiolas, C. McHugh, D. W. Roubik, W. T. Wcislo, and B. H. Smith, "Colony-Level Non-Associative Plasticity of Alarm Responses in the Stingless Honey Bee *Tetragonisca angustula*," *Behav Ecol Sociobiol*, vol. 72, no. 3, p. 58, Mar. 1, 2018.

[138] K. M. Baudier, M. M. Ostwald, C. Grüter, *et al.*, "Changing of the Guard: Mixed Specialization and Flexibility in Nest Defense (*Tetragonisca angustula*)," *Behavioural Ecology*, vol. 30, no. 4, pp. 1041–1049, Jan. 1, 2019.

[139] M. G. Earl and R. D'Andrea, "A Study in Cooperative Control: The RoboFlag Drill," in *Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301)*, vol. 3, IEEE, May 2002, pp. 1811–1812.

[140] ——, "Modeling and Control of a Multi-Agent System Using Mixed Integer Linear Programming," in *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, vol. 1, IEEE, Dec. 2002, pp. 107–111, ISBN: 978-0-7803-7516-1.

[141] R. D. Arnold, H. Yamaguchi, and T. Tanaka, "Search and Rescue with Autonomous Flying Robots Through Behavior-Based Cooperative Intelligence," *Int J Humanitarian Action*, vol. 3, no. 1, p. 18, Dec. 5, 2018.

[142] G. Diehl and J. A. Adams, "Battery Variability Management for Swarms," in *Distributed Autonomous Robotic Systems*, F. Matsuno, S.-i. Azuma, and M. Yamamoto, Eds., ser. Springer Proceedings in Advanced Robotics, Cham: Springer International Publishing, 2022, pp. 214–226, ISBN: 978-3-030-92790-5.

[143] P. Fiorini and Z. Shiller, "Motion Planning in Dynamic Environments Using Velocity Obstacles," *Int. J. Robot. Res.*, vol. 17, no. 7, pp. 760–772, 1998.

[144] N. K. Ure and G. Inalhan, "Design of higher order sliding mode control laws for a multi modal agile maneuvering UCAV," in *2008 2nd International Symposium on Systems and Control in Aerospace and Astronautics*, IEEE, Dec. 2008, pp. 1–6.

[145] ——, "Design of a multi modal control framework for agile maneuvering UCAV," in *2009 IEEE Aerospace Conference*, IEEE, Mar. 2009, pp. 1–10.

[146] ——, "Autonomous Control of Unmanned Combat Air Vehicles: Design of a Multimodal Control and Flight Planning Framework for Agile Maneuvering," *IEEE Control Syst.*, vol. 32, no. 5, pp. 74–95, Oct. 2012.

[147] S. Taranovich. (Nov. 13, 2015). F-35 Lightning II: Advanced electronics for stealth, sensors, and communications, EDN, [Online]. Available: `https://www.edn.com/f-35-lightning-ii-advanced-electronics-for-stealth-sensors-and-communications/` (visited on 05/28/2022).

[148] D. T. Davis, T. H. Chung, M. R. Clement, and M. A. Day, "Consensus-based data sharing for large-scale aerial swarm coordination in lossy communications environments," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, Oct. 2016, pp. 3801–3808, ISBN: 978-1-5090-3762-9.

[149] O. Dengiz, A. Konak, and A. E. Smith, "Connectivity management in mobile ad hoc networks using particle swarm optimization," *Ad Hoc Networks*, vol. 9, no. 7, pp. 1312–1326, Sep. 1, 2011.

[150] C. Sabo, D. Kingston, and K. Cohen, "A Formulation and Heuristic Approach to Task Allocation and Routing of UAVs under Limited Communication," *Unmanned Syst.*, vol. 02, no. 01, pp. 1–17, Jan. 2014.

[151] A. Klyubin, D. Polani, and C. Nehaniv, "Empowerment: A universal agent-centric measure of control," in *2005 IEEE Congress on Evolutionary Computation*, vol. 1, Sep. 2005, 128–135 Vol.1.

[152] T. H. Chung, K. D. Jones, M. A. Day, M. Jones, and M. Clement, "50 vs. 50 by 2015: Swarm vs. swarm UAV live-fly competition at the Naval Postgraduate School," *AUVSI N. Am.*, 2013.

[153] D. T. Davis, T. H. Chung, M. R. Clement, and M. A. Day, "Multi-swarm Infrastructure for Swarm Versus Swarm Experimentation," in *Distributed Autonomous Robotic Systems: The 13th International Symposium*, ser. Springer Proceedings in Advanced Robotics, R. Groß, A. Kolling, S. Berman, *et al.*, Eds., Cham: Springer International Publishing, 2018, pp. 649–663, ISBN: 978-3-319-73008-0.

[154] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. 1999, ISBN: 0-262-20162-3.

[155] S. Thrun, "Robotic mapping: A survey," in *Exploring Artificial Intelligence in the New Millennium*, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., Jan. 1, 2003, pp. 1–35, ISBN: 978-1-55860-811-5.

[156] T. Hazra and K. Anjaria, "Applications of game theory in deep learning: A survey," *Multimed Tools Appl*, vol. 81, no. 6, pp. 8963–8994, Mar. 1, 2022.

[157] R. M. French, "Catastrophic Forgetting in Connectionist Networks," *Trends in Cognitive Sciences*, vol. 3, no. 4, pp. 128–135, Apr. 1, 1999.