# Peering Through the Fog of War

**Song Jun Park, Anne Logie, Manuel Vindiola, Priya Narayanan**
**DEVCOM Army Research Laboratory**
**APG, MD**
**song.j.park.civ@army.mil, anne.c.logie.civ@army.mil,**
**manuel.m.vindiola.civ@army.mil, priya.narayanan.civ@army.mil**

**Deanna Franceschini**
**Cole Engineering Services, Inc.**
**Orlando, FL**
**Deanna.Franceschini@cesicorp.com**

## ABSTRACT

Machine learning based Artificial Intelligence is revolutionizing many areas of modern life, including translating conversations between different languages, identifying people in pictures, and autonomously driving cars. All these advances have created new, more complex, and challenging tasks where even human execution of these events sometimes falters due to lack of specialized training, lack of attention, information overload, or fatigue. Military commanders from the very earliest days have always been faced with the "fog of war", but in today's digital environment, not only can we begin to see through fog, but we can begin to glimpse the foundational components of that "fog". Similarly, the actions of a commander in a battlespace are affected by degree of training, attention to detail, and fatigue. Conducting tactical operations within a multi-domain operation environment exponentially increases the complexity of a leader's command and control with the inclusion of space, cyber-electromagnetic activities, and robotic assets, which are likely to drive the OPTEMPO even higher than in the past. Given the pace of operations, the complexities of the operational environment and the speed at which decisions must be made, sometimes minutes, the commander and their staff are simply overwhelmed in trying to assess in a timely matter what information is needed to allow the commander to make the best decision possible. In collaborative work between the DEVCOM Army Research Laboratory and Cole Engineering Services, Inc., we have been exploring the application of deep reinforcement learning towards the generation of an automated decision-making assistant. There are many practical uses for such a capability, including exposing novel or optimal course of action, identification of timely and unique threat activities that influence friendly plans, and the supervision of autonomous battlefield systems and activities. In this paper we describe the overall approach, the project's status, and provide lessons learned.

## ABOUT THE AUTHORS

**Song Jun Park** is a Computer Engineer at the Advanced Computing Branch, Computational and Information Sciences Directorate, DEVCOM Army Research Laboratory under the Army Futures Command.

**Anne Logie** is a Computer Scientist at the DEVCOM Army Research Laboratory under the Army Futures Command.

**Manuel Vindiola** is a Cognitive Scientist at DEVCOM Army Research Laboratory in the Computational and Information Sciences Directorate. He contributes to computational science, reinforcement learning, and neuromorphic computing research for the Army.

**Priya Narayanan** is a Research Mechanical Engineer at DEVCOM Army Research Laboratory in the Computational and Information Sciences Directorate. She is the lead for AI for C2 of MDO, an ARL Director's Strategic Initiative for developing the next generation AI enabled Command and Control system for the Army.

**Deanna Franceschini** has over 20 years of experience in the Modeling and Simulation (M&S) Industry and holds a master's degree in Computer Science. Ms. Franceschini has been working at Cole Engineering Services, Inc. (CESI) for over 5 years and is currently the CESI Program Manager for developing the Simulation Infrastructure for generating C2 Scenarios for Multi-Domain Operations. She is also the CESI lead for the Cooperative Research and Development Agreement (CRADA) in conjunction with DEVCOM Army Research Laboratory.

# Peering Through the Fog of War

**Song Jun Park, Anne Logie, Manuel Vindiola, Priya Narayanan**
**DEVCOM Army Research Laboratory**
**APG, MD**
**song.j.park.civ@army.mil, anne.c.logie.civ@army.mil, manuel.m.vindiola.civ@army.mil, priya.narayanan.civ@army.mil**

**Deanna Franceschini**
**Cole Engineering Services, Inc.**
**Orlando, FL**
**Deanna.Franceschini@cesicorp.com**

## INTRODUCTION

DoD has been using computer-based modeling and simulation for decades (Hill, 2017) to provide analysis, training, concept development, and acquisition support. Although there are simulations that connect and interact with mission command (MC), there were no simulations that were purpose built to embed in MC systems to provide course of action (COA). To be useful for embedded COA, the simulation environment must run much faster than real-time to provide rapid analysis yet good statistical coverage of a solution space. That is, the simulation must be able to run at least 30 iterations of one blue COA against one Red COA with the varying random number of seeds in less than five minutes. Further, the simulation must pull the plans and orders directly from the MC system – no additional simulation-specific user interface to simplify the use of this tool in the hands of military users.

In 2018, Cole Engineering Services, Inc., (CESI) developed the OpSim simulation to fill that need. OpSim is built as a lightweight force-on-force simulation environment, capable of running much faster than real-time, and operates connected directly to MC. Specifically, OpSim was built to embed within SitaWare (Systematic, n.d.) – the foundational technology of the Army's Command Post Computing Environment (United States Army Acquisition Support Center, n.d.). OpSim is a microservices-based, event-driven simulation that can execute more than 100,000 times faster than real-time. The Tiger Claw reference scenario, developed by the Maneuver Center of Excellence Captain's Career Course, is a five-hour brigade exercise. OpSim can execute 270 iterations of the scenario, against various enemy COA, in just under three seconds. As shown in Figure 1, OpSim presents a decision matrix view, via a SitaWare plugin, to allow warfighters to compare Blue and Red COAs constructed natively in the MC tool. OpSim rapidly plays out those COAs, varying the random seed between replications, to provide a comparison of the results.

Advantages of simulation technologies are reducing time and cost, while aiding to increase the efficacy of solutions. Force on force simulations approximate battlefield situations by providing models of the actors, their activities, and resultant consequences in that battlespace. A simulator can be augmented with reinforcement learning (RL), which is a paradigm in machine learning that learns optimal behavior interactively from experience (Sutton, 2018). The combination of simulation and RL leads to a model-free RL approach that learns to make sequential decisions in a stochastic environment.

Advancements in computing and deep learning have propelled the field of modern artificial intelligence (AI) and machine learning. This compute-intensive, data-driven approach is adopted in developing an RL-based AI associate for command and control (C2). We envision an AI associate to assist commanders and staff during C2 planning phase by providing alternative strategies and thus expanding commander's decision space. Emergent strategies produced by deep RL training are derived from millions of simulated data, which are free of human biases stemming from personalities or limited past experiences. RL offers to explore and exploit computational power in the C2 domain.

This paper describes the RL interface for OpSim simulator, design lessons for developing an AI associate, and the learning progression of RL training. In addition to enabling AI capability to support decision-making, integrating RL to a simulator has the advantage of rigorous model testing. By leveraging the strengths of exploration and exploitation of RL, a simulation model can be stress tested for validation. As for the development of deep RL-based AI associate, this design process revealed lessons for producing reasonable military strategies. For a deeper analysis of emergent behaviors from the developed AI associate, we present strategies that emerged during the AI training run, which started from a blank slate without any hand-crafted previous knowledge.

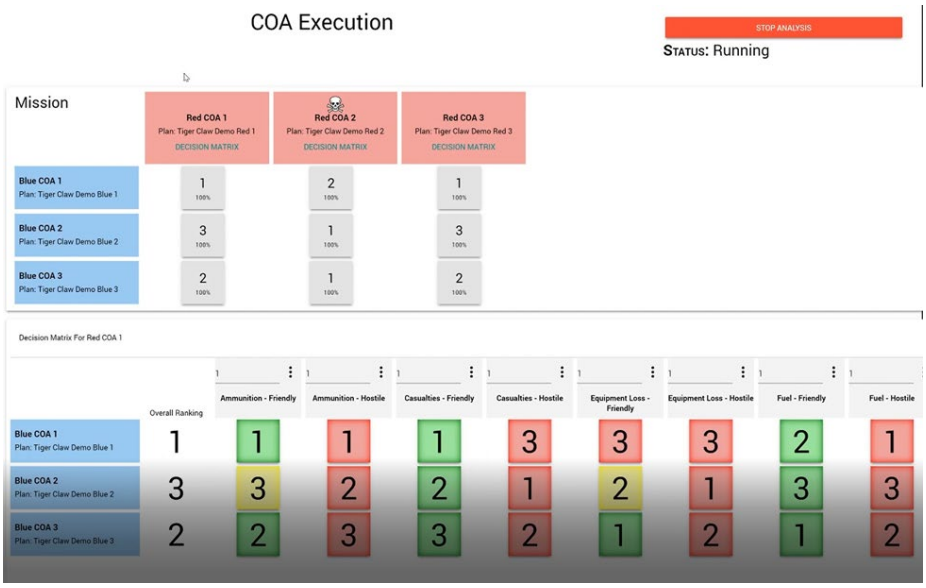



**Figure 1. OpSim Decision Matrix View Plugin to SitaWare Allows Warfighters to Compare Blue and Red COAs**

## ENVISIONED AI CAPABILITIES

Conducting tactical operations within a multi-domain operation (TRADOC, 2018) environment exponentially increases the complexity of C2 with the inclusion of space, cyber-electromagnetic activities, and robotic assets. This increase in complexity will likely drive OPTEMPO even higher than in the past. The increased pace of operations, the complexities of the operational environment and the speed at which decisions must be made, sometimes minutes, the commander and their staff will be overwhelmed in trying to assess, in a timely manner, what information is needed to allow the commander to make the best decision possible. Data-driven AI approaches can search through the possible decision space to assist in the decision making process by providing alternative strategies to the commander. Advantages of an AI associate for C2 include: not being susceptible to training bias, experience bias, information overload, or fatigue.

In addition to a dependable and unbiased AI solution, deep RL can discover novel and insightful strategies via compute-intensive exploration. For instance, an AI associate can produce unconventional and unexplored strategies to the commander and staff, which can elevate a planner's understanding of a scenario as similar to move 37 in AlphaGo (DeepMind, n.d.). An AI associate arms the commander with tactics and strategies that are proven successful in simulated exploration. An AI associate serves to enrich the COA development process by leveraging RL's computational strengths.

Moreover, emergent behaviors produced by RL expose unexpected strengths and weaknesses within dynamic, nonlinear, and complex military operations. The RL discovered knowledge of strong and weak characteristics in a scenario can be leveraged to create windows of superiority. A challenge to address with windows of superiority is automated identification or detection. Future concepts for enabling windows of superiority with RL are described in the focused excursion document (Taliaferro et al., 2020).

## CONNECTING AI AGENTS TO THE SIMULATOR

Deep RL is an area of machine learning where an RL agent learns from experience through an iterative learning process. In each iterative loop, the agent takes an action in its environment and then collects observations about the

new state of the environment and receives a reward. The agent uses an RL algorithm to coordinate this loop and to train a policy for selecting actions that guide the agent closer to its goal. Deep RL takes advantage of recent advancements in deep neural networks to enable function approximation in RL techniques (Mnih et al., 2015; Silver et al., 2016).

The OpenAI Gym interface formalizes exchange of observations, actions, and rewards between simulations and AI agents. Actions are specific instructions that can be provided to simulated actors. Observations are any exposed simulation state, such as the health status of actors, supply status, relative locations of friendly forces, and the locations and types of perceived enemies. Rewards are provided based on how closely an action moves an agent towards its goal.

In support of the deep RL effort, DEVCOM ARL and CESI extended OpSim with the capability to configure two types of commanders. One type of commander follows courses of action created by an army subject matter expert. The other type of commander follows a deep RL policy. For the RL-controlled commander, we incorporated RL into OpSim using OpenAI Gym, which is a standard RL interface for combining and comparing RL algorithms and environments (Brockman et al., 2016). We defined an action space, observation space, and a reward for training the deep RL policy within the OpSim Gym. The actions space consisted of 14 actions describing movement and firing actions. The observation space consisted of 481 observations, 37 per entity, of the current state of the environment. The reward included a multi-objective goal t: protecting ones self, destroy the enemy, and reach COA-defined strategic goal locations for each phase of the battle. The OpSim Gym environment implementation enables observation spaces and rewards to be easily adapted to experimental requirements. A deeper explanation of the chosen observation space and reward is included in the discussion below. For this work we utilized the Advantage Actor Critic (A2C) deep RL algorithm (Mnih et al., 2016) with long short-term memory neural network (Hochreiter, 1997) architecture.

This work is the extension of our previous works (Park et al., 2022; Narayanan et al., 2022; Goecks et al., 2021) where we first presented RL-supported OpSim framework and initial results. In this work we describe refined observation space and reward resulting in new strategies.

**AI Design Lessons for Producing Reasonable Strategies**

This section describes the development and enhancements to a deep RL design that produced reasonable strategies in a wargame modeling and simulation scenario. Experiments indicate that reward structure for the learning problem had a large impact on the resulting behaviors of the AI agent. For example, a reward function that rewards for neutralizing your enemy and penalizes for your own losses, can lead to a behavior that avoids engagements depending on a risk and benefit analysis. The reward function was designed with broad applicability in mind, rather than being specialized for a specific use case. For a broad application, the reward function was structured around equipment losses and COA goal locations. The reward for the equipment is formulated such that a positive reward is received for neutralizing your opponent and a negative penalty is incurred when your own units are destroyed. In addition to equipment, there is a penalty for being away from your objective goal locations. The motivation for providing goal locations to the RL training process was to accelerate the speed of learning by incorporating the operation order's intent of securing objective regions. As for entities without a specified goal location, a penalty is received when movement occurs. Reward values for each component were scaled to be in the [-2, 2] range.

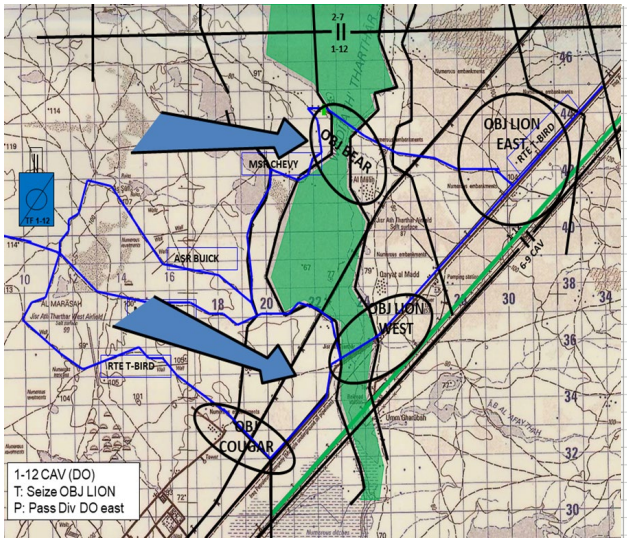
In addition to a reward function, an RL agent's observation space plays a significant role in learning effective policies. To minimize the input size to a neural network, a feature vector was constructed as observations instead of using images as an input. To form an observation space that is receptive to learning, we converted the earth-center earth-fixed (ECEF) coordinate system into a less complex, east north up (ENU) coordinates based on local tangent plane. The ENU coordinate system maps location information into a 2-dimensional space, which simplifies entities' spatial data when compared to ECEF. The scale of each component in the observation space can vary greatly, such as the real valued sensor range [0-30,000 meters] and the categorically encoded damage state [FullyCapable, Damaged, Destroyed]. To mitigate artificial importance weights to the sensor range values, each observation was normalized to achieve a uniform range within an agent's observation space. The OPSIM environment is partially observable, each agent only sees a limited area of the entire battle field which is governed by the on board sensors. The number and types of observed opposition units can vary greatly within and between units depending on their location on the simulated battle field and phase of operations. In order to provide a consistently sized observations space pertaining

to an opposition's composition, we assign OPFOR's equipment into 10 categories and record the total number of each category perceived by our sensors. This approach keeps the length of an observation space fixed even as opposition's force composition changes in size.

Without parallel and distributed computing, learning a reasonable policy from a million battle experiences even with OpSim's 100,000-fold real-time acceleration would require months of RL training time. This time frame is obviously not practical for performing rapid design experiments on reward function, observation space, and hyperparameters. Apart from small toy problems, deep RL research must leverage high performance computing (HPC) systems for computational acceleration. For this project, we used ARL DoD Supercomputing Resource Center SCOUT system. Training is performed an IBM PowerPC system with 160 logical cores. To utilize parallel computing hardware resources, we use a heavily customized version of the scalable computing software framework RLlib (Liang et al., 2018) built on top of Ray (Moritz et al., 2018). RLlib and Ray are projects that originated from the University of California, Berkeley and are currently developed by the Anyscale startup company. Our current RL training setup assigns 100 parallel workers on the HPC system and executes training for two weeks to collect around one million battle experiences.

**Operation Scenario**

To study and develop an AI associate for C2, a brigade-level COA within a division operations order served as an RL development scenario. The scenario takes place in an area of operations provided in Figure 2, where a dry river bed is denoted in green. The goal of the BLUFOR is to conduct an offensive operation to seize objective locations marked on the map and neutralize the enemy such that a following Division force can continue operations to the East. The OPFOR defends its territories near the dry river bed in an effort to prevent BLUFOR's advancement. The BLUFOR is a cavalry task force and the OPFOR is a mechanized infantry battalion with attached artillery battery. The conventional 3:1 rule of combat argues that "numbers approaching three to one are required to turn the scale decisively" (Hart, 1960, p.49). With this doctrine in mind, the BLUFOR's force structure is greater in strength, as illustrated by the equipment numbers in Figure 2. Both BLUFOR and OPFOR forces have a total of 13 entities and their entity types are shown according to NATO joint military symbology (Department of the Army, 2020).




| | Equip 0 | Equip 1 | Equip 2 | Equip 3 | TOTAL |
|---|---|---|---|---|---|
| 1 1 A 1-3 ADA | 2 | | | | 2 |
| 1 A 1-4 AV | 4 | | | | 4 |
| 1 SCT 1-12 CAV | 3 | 5 | | | 8 |
| 2 A 1-4 AV | 4 | | | | 4 |
| 2 SCT 1-12 CAV | 3 | 5 | | | 8 |
| 3 A 1-12 CAV | 4 | | | | 4 |
| A 1-12 CAV | 24 | 2 | 4 | | 30 |
| B 1-12 CAV | 28 | 2 | 4 | | 34 |
| C 1-12 CAV | 2 | 28 | 4 | | 34 |
| D 6-9 CAV | 28 | 2 | 4 | | 34 |
| FIRES 1-12 CAV | 6 | | | | 6 |
| MTR 1-12 CAV | 3 | 3 | 12 | 10 | 28 |
| UAS | 4 | | | | 4 |
| | BLUFOR EQUIPMENT TOTAL | | | | 200 |
| 1 A 3-801 | 8 | | | | 8 |
| 1 A BN 1 | 3 | | | | 3 |
| 1 AA 801 | 4 | | | | 4 |
| 1 B BN 1 | 6 | | | | 6 |
| 1 CAS 801 | 2 | | | | 2 |
| 1 RECON 801 | 2 | 3 | 1 | | 6 |
| A 4-801 | 6 | 3 | | | 9 |
| A BN 1 | 18 | | | | 18 |
| B BN 1 | 18 | | | | 18 |
| BN HQ | 2 | 2 | | | 4 |
| Bilasuvar Freedom Brigade | | | | | 0 |
| D BN 1 | 6 | 1 | | | 7 |
| RECON BN 1 | 2 | 1 | | | 3 |
| | OPFOR EQUIPMENT TOTAL | | | | 88 |

**Figure 2. Scenario Details for Area of Operations and Force Structures**

**RESULTS**

In our RL implementation for C2, both BLUFOR and OPFOR sides can be assigned as RL agents. When both BLUFOR and OPFOR are learning AI agents, we observed a policy adaptation effect where each side's policy evolves in response to opposition's strategy and complicates analysis. As a starting point, to simplify analysis, results are

collected for the setup of BLUFOR AI associate playing against human generated OPFOR plan. Work is currently in the process of training and analyzing results from an OPFOR AI associate. These results will lead to additional insights into mitigation strategies for BLUFOR plans from the OPFOR's perspective. Setting both sides to be RL agents can result in discovering avariety of strategies, in which both sides can learn and evolve in natural curriculum as presented in the multi-agent hide-and-seek research (Baker, 2019).

**Emerged Strategic Behaviors From the AI**

Our approach adopts an end-to-end Deep RL machine learning technique that learns completely from scratch without any previous knowledge of weapons, terrain, or governing rules. By interacting with a simulator and collecting experiences, an RL agent learns to act optimally, by which the RL algorithm is maximizing the expected sum of rewards. We report on the policy development and how it evolved during the training cycle. Figure 3 shows the selected points of interest at various reward levels in the mean reward curve, chosen to be at zero, 9 million, 126 million, and 285 million simulated steps.

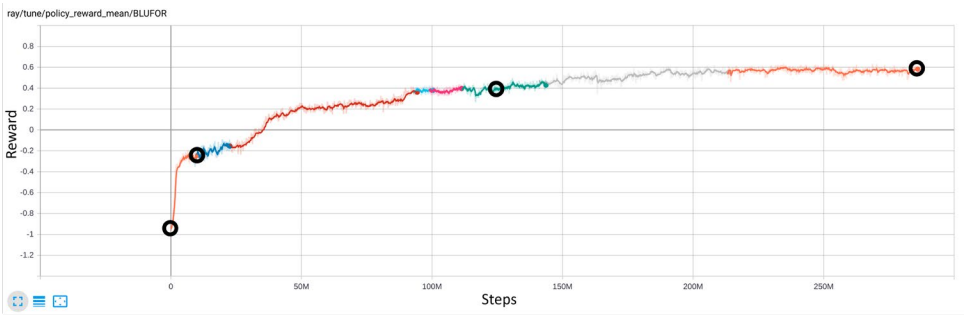



**Figure 3. Training Curve Plotting Mean Reward**

We generated videos of learned AI policy at each point circled on the training curve in Figure 3 to observe how the behavior of the units changed over time. The units shown are icons defined in the standard military symbology. The following discussion provides screen shots taken from videos of emergent strategies. In the beginning, the RL agent's policy, represented by a deep neural network, is randomly initialized, which results in issuing random actions. Hence, random movement behaviors branching out from the clustered initial location are observed for the BLUFOR at this stage, as shown in Figure 4. To prevent units from randomly firing its weapons, actions deemed illegal are masked to prevent selection. For example, the fire weapon action is masked when the detected target acquisition level of an opponent unit is not satisfied.

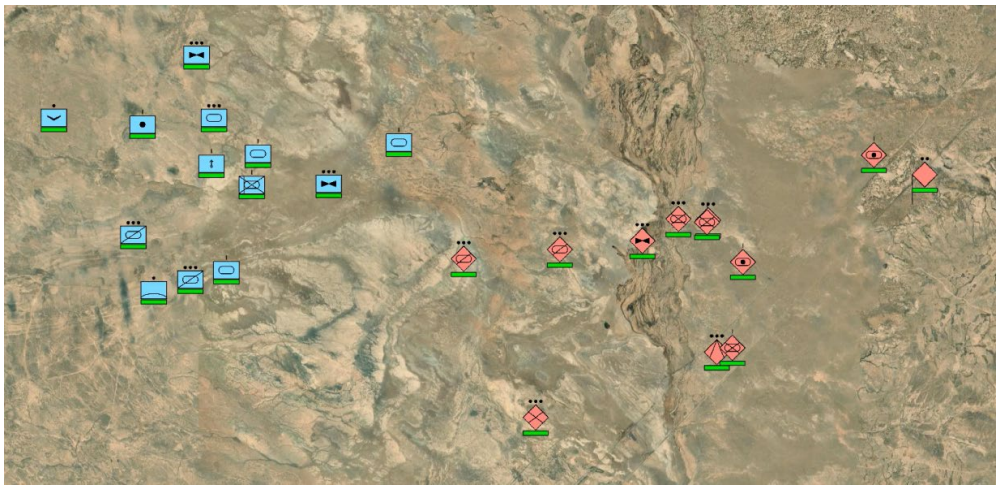


**Figure 4. Resulting Random Movements of End-to-End RL Learning From Scratch**

After 9 million training steps, the video of a learned policy indicates that the RL agent has learned to attack from the air with Rotary Wing Aircraft, as captured in Figure 5. Through simulated experience, the RL method seems to have discovered the lethal effects of an Apache platoon. At this point in RL training, BLUFOR entities fail to reach their goal locations. However, by Apache units neutralizing the enemy, it clears a path for vulnerable ground forces to move toward their strategic goal locations without the risk of being destroyed.

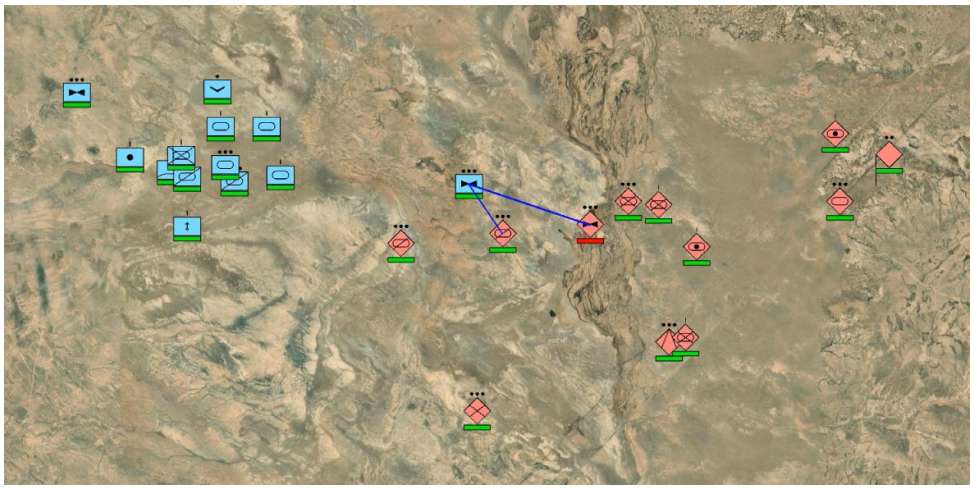


**Figure 5. Learned Effectiveness of Apache Platoon After 30,000 Battle Simulations (9 Million Steps)**

The next point of interest during the RL training was after the second increase in mean reward at 126 million steps. The video of the policy at this moment of training demonstrated that the RL agent has learned to call for fire support. Figure 6 shows a snapshot of the BLUFOR field artillery's firing line of attack. Field artillery's fire support is called by the combat armor company, which has learned to maneuver toward OPFOR's headquarter unit. After 420,000 battle simulations, the RL agent employs the most lethal entities, combat aviation platoon and combat armor company with field artillery support, to engage with the enemy and move toward entities' assigned goal locations.

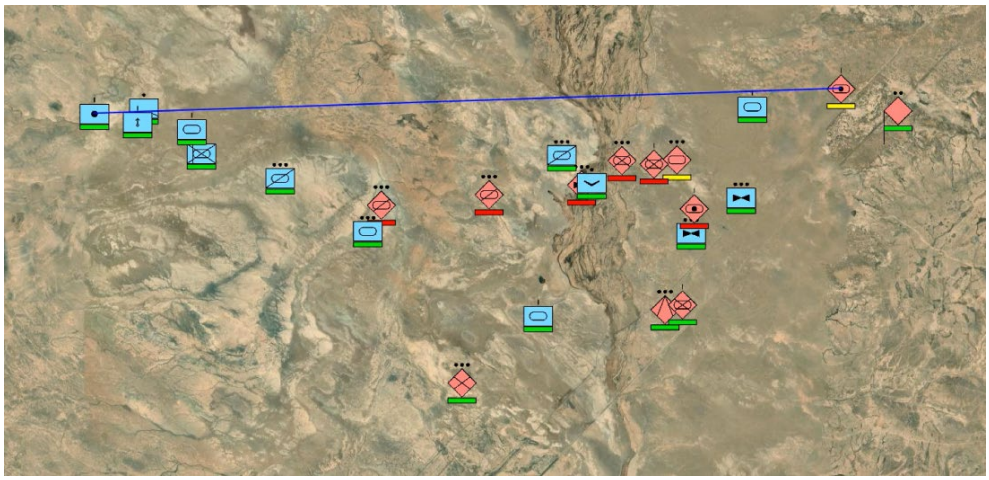


**Figure 6. Learned to Utilize Field Artillery After 420,000 Battle Simulations (126 Million Steps)**

After approximately a million battle simulations, one notable alternative strategy that emerged was the northern route taken by a combat armor company to avoid detection and to target the opponent's field artillery company that poses a substantial threat. Compared to the human generated plan, attack route and timing for engagement at the objective Lion East (objective locations are shown in Figure 2) are different. Employing this strategy, Figure 7 shows that OPFOR is neutralized and BLUFOR units reach their objective locations while maintaining combat power. The RL agent successfully completes the mission.

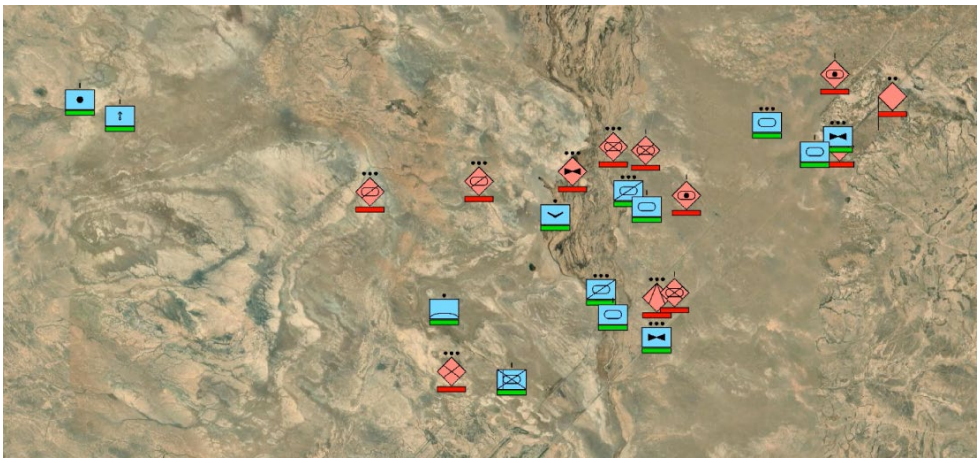


**Figure 7. Mission Objectives Achieved After 950,000 Battle Simulations (285 Million Steps)**

As mentioned earlier, the starting point for the deep RL training was a random policy. An alternative approach would be to initialize a policy to mimic a human expert, which reduces the initial exploratory search space. Instead of a random policy at start, a policy neural network can be initialized with a policy derived from imitation learning. Pre-training a policy network with imitation learning can substantially reduce the training time by starting from a reasonable policy, but this can constrain the discovery of unexpected novel solutions due to biasing the network toward a human solution in the beginning. A tradeoff exists for determining how much human guidance to provide the initial policy for RL training. Seeking to discover alternative solutions, we opted for an unbiased, random initialization.

**Analysis of AI Generated Plan**

Figure 8 illustrates the RL generated plan that prioritizes three enemy occupied regions with avenue of approach depicted as colored arrows. These three regions contain units that pose a greater threat and corresponds to key objective locations specified in the operations order. Compared to the human generated plan, the AI associate decides to take the northern route, displayed as orange arrow in Figure 8, to bypass detection and neutralize the opposition's artillery as early as possible. The AI associate leads the attack with lethal units in the brigade, which are combat aviation and combat armor entities with field artillery support. For statistic analysis, survivability and engagements are computed for 10 simulated runs in Figure 9. The AI associate decides to engage heavily at the start of operation leading with Apache units. As for the remaining forces at the end of five-hour operation, the majority of BLUFOR are fully capable with unit losses occurring less than 20% for the scouts and combat armor platoon. OPFOR on the other hand, suffers significant losses where all of its forces are destroyed by the approaching BLUFOR.

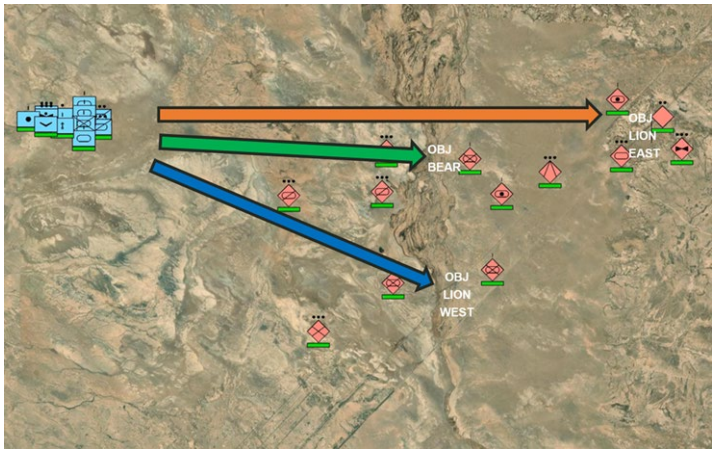



**Figure 8. BLUFOR AI Associate's Strategy**

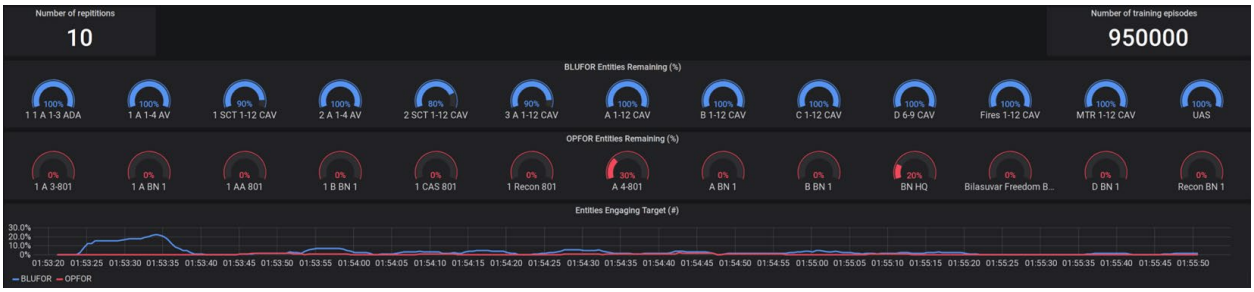



**Figure 9. Statistics on Survivability and Engagements for 10 Evaluation Runs**

## DISCUSSION

There are three insights that can be drawn from the commander's AI associate. First, RL generated behavior demonstrates that a viable alternative BLUFOR strategy could be incorporated into current human generated plan. RL has discovered an unbiased data-driven plan for commander's consideration that deviates from the original plan. Second, the OPFOR's COA can be refined and strengthened. After learning BLUFOR AI's strategy for this scenario, OPFOR's plan can be enhanced by accounting for this knowledge. And lastly, simulation models can be updated for improved realistic behaviors. RL's exploitation has assisted in locating unexpected behaviors in the simulator. Overall, the combination of RL and OpSim has a potential to augment multiple aspects of C2 planning and analysis.

RL was successfully applied to a complex, real-world battle simulator and produced a reasonable strategy. However, there are limitations as we envision operationalizing AI associate for C2. There will be variability, noise, and knowledge gaps in real-world military operations. For example, the assumptions about enemy force strength will not be perfect, terrain mobility can change due to weather, and information regarding the status of your unit will be at the mercy of communication reliability. These factors suggest that an AI agent's observations in the real-world will deviate from observations experienced at training time. Thus, in order to transition AI associate for C2, the learned RL policy needs to be robust to variations in operational details. For an AI associate to be effective, its policy should generalize to variations within a scenario. Furthermore, the single-agent RL makes a strong assumption that an RL agent has access to a complete knowledge of every unit in a C2 plan during an operation. In an actual operation, central command's view of a battle space will be incomplete and limited by communication and sensing technologies. The multi-agent RL framework relaxes the assumption of having a complete knowledge of all your units by decomposing a single BLUFOR AI associate into multiple RL agents. We are currently conducting research on generalization and multi-agent RL in C2 setting to mitigate these limitations.

## CONCLUSION AND FUTURE WORK

Coupling reinforcement learning and a COA simulator illustrates a proof of concept for a data-driven AI associate for C2 planning. Computational exploration and exploitation of RL searches and discovers experience-based optimal strategies that is guided by a reward function. The developed AI associate for C2 produced an alternate AI generated plan for commanders to consider and expand their decision space. We envision a computational RL method producing informative strategies from data.

Our work has made great strides at applying deep RL to simulations in support of creating automated decision aids, but there are many more areas left to explore. Specifically, to date our team has only explored a subset of goals, such as self-protection, and destruction of enemy forces. Future effort should be applied to examining goals that are more subtle, such as maintain supplies, gathering intelligence, etc. Additionally, the training environments to date have been relatively static – no variation in equipment holdings or changes to terrain. Varying those conditions during training potentially leads to more adaptable RL agents.

As the OpSim environment can directly connect to fielded Mission Command, another next step in the maturation of this technology is applying agents directly to live snapshots of Mission Command data. In this way, the DRL agents

could either recommend Blue COAs against current known Red situation or dynamically create Red COA variants based on the known Red situation to inform BLUFOR vulnerabilities.

## ACKNOWLEDGEMENTS

The authors wish to acknowledge the Army Research Laboratory-hosted Department of Defense Supercomputing Resource Center (ARL DSRC) for its support of this work.

## REFERENCES

Baker, B., Kanitscheider, I., Markov, T., Wu, Y., Powell, G., McGrew, B., & Mordatch, I. G. O. R. (2019). Emergent tool use from multi-agent interaction. *Machine Learning, Cornell University*.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., & Zaremba, W. (2016, June 5). *OpenAI Gym*. arXiv preprint arXiv:1606.0154

DeepMind. (n.d.). AlphaGo. https://www.deepmind.com/research/highlighted-research/alphago

*Fm 1-02.2 military symbols*. Department of the Army. (2020). Retrieved May 8, 2021, from https://armypubs.army.mil/epubs/DR_pubs/DR_a/ARN31121-FM_1-02.2-000-WEB-1.pdf

Goecks, V. G., Waytowich, N., Asher, D. E., Park, S. J., Mittrick, M., Richardson, J., Vindiola, M., Logie, A., Dennison, M., Trout, T., Narayanan, P., & Kott, A. (2022). On games and simulators as a platform for development of artificial intelligence for command and control. *The Journal of Defense Modeling and Simulation*.

Hart, B. H. L. (1960) The ratio of troops to space. *RUSI Journal*, 105(618), 201-212.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735-1780.

Liang, E., Liaw, R., Nishihara, R., Moritz, P., Fox, R., Goldberg, K., Gonzalez, J., Jordan, M. I., & Stoica, I. (2018). RLlib: Abstractions for distributed reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning* (Vol. 80, pp. 3053–3062). PMLR.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*, 529–533.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T. P., Harley, T., Silver, D., & Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. In *Proceedings of the 33rd International Conference on Machine Learning* (Vol. 48, pp. 1928–1937). PMLR.

Moritz, P., Nishihara, R., Wang, S., Tumanov, A., Liaw, R., Liang, E., Elibol, M., Yang, Z., Paul, W., Jordan, M. I., & Stoica, I. (2018). Ray: A distributed framework for emerging AI applications. In *Proceedings of the 13th USENIX Symposium on Operating Systems Design and Implementation* (pp. 561–577). USENIX.

Narayanan, P., Hawkins, T., Park, S. J., Cassenti, D. N., Logie, A., Armstrong, S., Vindiola, M., Waytowich, N. R., Pak, M., Arthur, M., & Holland, T. (2022). A conceptual real-world experimental framework for AI-enabled command and control applied to terrain shaping operations as use case. *Journal of DoD Research & Engineering*, *5*(1), 2–13.

Park, S. J., Vindiola, M. M., Logie, A. C., Narayanan, P., & Davies, J. (2022). Deep reinforcement learning to assist command and control. In *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications IV* (Vol. 12113, pp. 430–438). SPIE.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., ... & Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587), 484-489.

Sutton, R.S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. A Bradford Book.

Systematic. (n.d.). SitaWare C4I suite. https://www.systematicinc.com/products/n/sitaware/

Taliaferro, A., Stump, E., Narayanan, P., Kott, A., Foresta, J., Colegrove, & S., Burland, B. (2020). *Discovery enabler concept of operations - artificial intelligence: Reinforcement learning to enable decision overmatch*. U.S. Army Futures Command.

TRADOC United States Army Training and Doctrine Command. (2018). *The U.S. Army in multi-domain operations 2028.* https://adminpubs.tradoc.army.mil/pamphlets/TP525-3-1.pdf

United States Army Acquisition Support Center. (n.d.). Command post computing environment (CPCE). https://asc.army.mil/web/portfolio-item/command-post-computing-environment-cpce/