AI Support Systems in the Military

John P. Gulon
gulon008@morris.umn.edu
Division of Science and Mathematics
University of Minnesota, Morris
Morris, Minnesota, USA

Abstract

Artificial intelligence (AI) decision support systems are increasingly integrated into military operations to assist commanders in complex, data rich environments. This paper examines how reinforcement learning agents can align with established military doctrine, operational objectives, and command structures. Based on current research, the study evaluates both the capabilities and limitations of an AI model architecture, emphasizing the importance of doctrinal alignment, human oversight, and accountability. Although AI systems can improve decision speed and situational awareness, their effective deployment depends on preserving meaningful human control in high risk operational contexts.

Keywords: Artificial Intelligence (AI), Reinforcement Learning (RL), Military Decision Support, ReLeGSim, Reward Design, Doctrinal Alignment, Human-in-the-loop Systems

1 Introduction

Artificial intelligence is becoming an increasingly significant component of modern military operations as armed forces confront rapidly evolving information and battlefields. Contemporary command environments require the rapid collection of large volumes of diverse data under the possibility of severe time constraints. These pressures have intensified interest in AI-enabled decision support systems (DSS) designed to augment, rather than replace, human judgment [2, 4].

Modern military AI development spans a wide range of applications, including intelligence analysis, autonomous systems, logistics optimization, and command decision support. As operational complexity increases, traditional human-centric processes within the command structure, particularly in the observe-orient-decide-act (OODA) loop, face growing strain. AI systems are therefore explored as tools to consolidate information, evaluate courses of action, and improve situational awareness. To understand and demonstrate these capabilities Möbius et al. have developed a battalion-level combat simulation environment called ReLeGSim (Reinforcement Learning Ground Simulation), where agents can recommend and evaluate military commands [2, 4, 5].

Multiple AI methodologies have emerged as relevant options as advisory tools. Reinforcement learning AI agents are one of those prominent methods designed to pursue goals and complete tasks on behalf of users. Reinforcement learning provides a framework for optimizing sequential decision-making in dynamic environments. This approach offers substantial potential but also introduces significant risks related to model misalignment, automation bias, over-reliance on AI output, and brittleness in novel scenarios [4, 5].

A central challenge in military AI integration lies in alignment with established doctrine, operational objectives, and ethical constraints. However, research shows that RL agents can encode tactical principles through carefully designed reward structures informed by rules of engagement and commander intent [4].

These developments suggest that the effectiveness of AI-enabled decision support will depend less on the raw capability of the model, meaning the trained AI system's ability to process battlefield data and generate tactical recommendations, and more on how that model is integrated into the larger command process. In this context, "raw capability" refers to the model's technical performance by itself, such as how quickly it can process information or how successfully it performs in simulation. However, military usefulness also depends on how learning objectives are constrained, how outputs are validated, and how meaningful human control is preserved within the command structure.

The analysis first introduces ReLeGSim as the simulation environment used to study reinforcement-learning decision support in battalion-level military scenarios. Then explains the model architecture, including the observation space, feature extraction layers, temporal reasoning components, and action/value outputs. Next, the paper analyzes how reward design and the reinforcement-learning objective translate military priorities into model behavior. Finally, it discusses the system's reported results, limitations, risks of automation bias, and the importance of human oversight when applying AI decision-support systems in military contexts.

2 ReLeGSim

ReLeGSim is a turn-based tactical simulation designed for battalion-level engagements between two hypothetical forces that are referred to as BLUE and RED. The environment supports multiple companies, including combat and reconnaissance units, and represents the battlefield as a grid terrain map, as shown in Figure 1. Terrain type, unit composition, and force placement matter because the simulation is intended to resemble the kind of structured battlefield picture

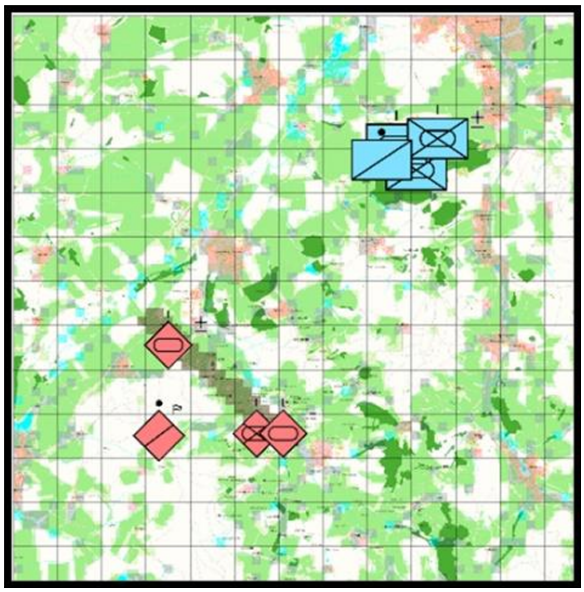

Figure 1. ReLeGSim battlefield representation, adapted from Möbius et al. [4].

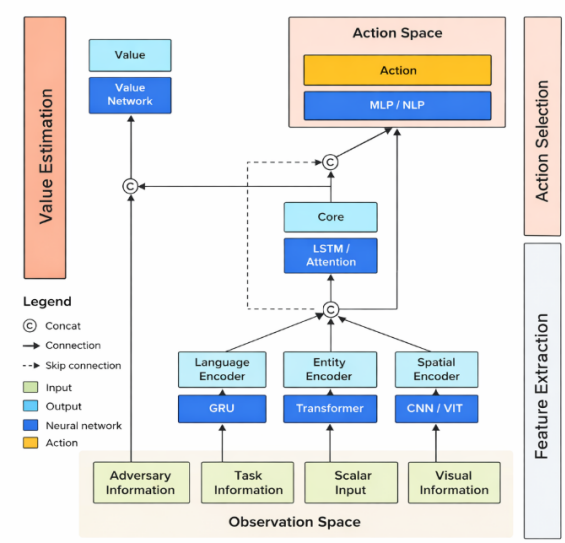

Figure 2. ReLeGSim RL architecture showing multi-modal input processing, feature fusion, LSTM-based decision-making, and action/value outputs. Figure was adapted from Möbius et al. and modified by GPT-5.3 for clarity [4].

that commanders already use in practice [3, 4]. Instead of assuming omniscient knowledge, the system preserves uncertainty through the fog of war, which means the agent must use reconnaissance and partial information rather than perfect visibility [3].

The simulator is valuable not merely because it produces realistic maps. Its importance is methodological. ReLeGSim creates a controlled environment in which an RL agent can repeatedly observe a state, recommend a command, receive a reward, and gradually adjust its policy. That loop makes it possible to test how different objectives alter tactical behavior. It also makes it possible to compare strategies at scale, because the simulation can run headless, faster than real time, and across distributed hardware for large training runs [3].

Equally important, ReLeGSim is designed around the idea that AI should work with a military operator instead of acting independently. In the earlier system description, Möbius et al. argue for a combined human-machine approach in which human knowledge and responsibility remain paired with machine speed [3]. The interface is built so that AI-generated commands can be expressed in natural language, observed by a human, and changed when necessary. That design decision is central to this paper because it directly addresses the problem of automation bias: commanders are less likely to treat the system as an oracle when they can inspect, interpret, and revise its output reducing blind trust [5].

3 Model Architecture: From Battlefield Data to Tactical Output

3.1 Observation Space

Figure 2 provides an overview of the ReLeGSim reinforcement learning architecture, showing how battlefield information moves from the observation space through feature extraction, the core decision network, and finally into action and value outputs. The architecture begins with the observation space, which defines what the AI can know at a given decision step. In the ReLeGSim system, the observation space combines adversary information, task information, scalar data describing friendly forces and support assets, a visual map, the previous action, and an action mask that limits invalid outputs, ensuring only valid actions remain [4]. This matters conceptually because battlefield awareness is never a single data type. The agent must combine enemy and friendly force tables, operational objectives, and geographic context inside one decision process. This is exactly where the AI agent's "battlefield awareness" begins.

Adversary information includes the location, composition, and operational state of opposing units. Task information specifies what the force is currently supposed to do, such as capture an area, attack quickly, preserve friendly forces, or stay concealed. Scalar inputs encode structured numerical descriptions of the battlefield, including force status and support resources. Visual information adds terrain and spatial layout, allowing the model to reason about roads, forests, open ground, and the relative placement of companies [3, 4].

The observation space therefore mirrors real command practice: maps and tables are both necessary, and neither one is sufficient by itself.

3.2 Feature Extraction Layer

Because these inputs have different structures, with features such as unit coordinates on a grid, numerical force-status values, visual terrain information, and dynamic objectives given by the user, the model does not process every input in the same way. Instead, it uses separate encoders. An encoder is a part of a neural network that converts one type of input into a numerical representation that the rest of the model can use. As Figure 2 shows, the model separates language, entity, and spatial information into different encoder pathways before combining them later in the network.

The architecture uses specialized modules for each modality. Natural-language task input is processed through a language encoder. In this context, a language encoder converts command-like task information into a numerical vector. Möbius et al. use a gated recurrent unit, or GRU, which is a neural-network component designed to process sequences such as words or command phrases [4]. Structured entity tables are processed by an entity encoder using attention, a method that helps the model weigh which units, support assets, or relationships are most relevant at a given decision step. The visual map is processed by a spatial encoder, such as a convolutional neural network (CNN) or a vision transformer (ViT), both of which are commonly used for image-like data because they can detect spatial patterns such as terrain, unit locations, and clusters on a map [4]. This division of labor is one of the strongest aspects of the architecture, the ability to be multi-modal allows for more closer to real interpretation of the real world.

The language pathway is especially important because it provides a technical route for command intent to enter the model and output priorities expressed in a more human-readable, language-like format. This is achieved when the GRU encoder converts task information into a vector that becomes part of the downstream state representation. The entity encoder, by contrast, is designed to model relationships among units and support assets, which is why attention-based processing is appropriate: unit interactions are relational rather than purely sequential. In other words, the importance of one unit depends on its relationship to other units, such as distance, support range, enemy contact, or shared objectives, rather than only its order in a sequence [3, 4].

3.3 Fusion, Temporal Reasoning, and Action/Value Outputs

After each modality has been encoded, the feature streams are concatenated, meaning they are joined into one unified internal state, and then passed to the core network. In Figure 2, that core uses an LSTM, attention, and skip connections [3, 4]. An LSTM, or long short-term memory network, is a neural-network component designed to retain information from earlier steps. This is useful because military decisions are not isolated snapshots. A support request issued earlier may affect the battlefield minutes later, and a reconnaissance event from previous steps may still shape the current recommendation.

The temporal, or time-based, part of the model therefore gives it a limited form of memory. This means the model can use information from earlier simulation steps when choosing a current action. The attention component helps emphasize the most relevant parts of the current situation, while skip connections help preserve earlier information as it moves through the network [3, 4].

From the core network, the model branches into two outputs. The first is an action head that generates command sequences. The second is a value-estimation pathway that estimates the long-term desirability of the current state [4]. This separation is analytically useful. Action selection answers the immediate question, “What should be done now?” The value estimation answers a different one, “How promising is this situation over time?” Together, the two outputs make the model more than just a command generator. They make it a sequential decision system that both proposes actions and evaluates strategic promise.

3.4 Natural-Language Action Space and Human Control

A notable design choice in the ReLeGSim line of research is the natural-language action space. Instead of using an opaque index tied to arbitrary actions, the model generates command sequences from a constrained vocabulary that includes action verbs, coordinates, unit references, and support assets [3, 4]. These commands are parsed into executable actions inside the simulator. The restricted vocabulary reduces complexity while keeping the output interpretable. Just as important, action masking is used to remove semantically or operationally invalid choices, including commands that conflict with sentence structure or unavailable resources [4].

This design directly supports human oversight. In the 2023 decision-support article, the authors describe a human-on-the-loop interface in which the operator can observe and change any command given by the AI [3]. In the 2024 article, they extend the idea by embedding objectives and doctrines through natural language and reward shaping [4]. Together, these choices help counteract automation bias in two ways. First, the commands are legible enough for a human to inspect. Second, the operator is not forced to accept the output as final. The architecture is therefore significant not only because it is multi-modal, but because it remains open to intervention at the point where intervention matters most: between recommendation and execution.

4 Reward Design, Learning Objective, and Reported Results

4.1 Reward as Doctrine Encoding

The reward function is the clearest place where military priorities become mathematical priorities. This key equation is a weighted multi-objective reward:

$$R(s_t, a_t) = w_1 r_{\text{mission}} + w_2 r_{\text{enemy}} + w_3 r_{\text{friendly}} \quad (1)$$

This expression is intentionally simple, but its meaning is central. The reward at time t depends on s_t (current state in timestep) and the chosen a_t (current action in timestep), while the weights determine which aspects of the battlefield performance matter the most: a larger r_{mission} (mission term) pushes the policy toward objective completion, a larger r_{enemy} (enemy term) favors destruction or neutralization of opposing forces, and a larger r_{friendly} (friendly term) rewards force preservation or loss avoidance. In other words, the reward function is not merely a scoring device. It is a formal statement of what the system is taught to value. If the reward is too narrow, the agent can learn brittle or tactically unrealistic behavior. Möbius et al. explicitly warn that an oversimplified reward can lead units to prioritize enemy destruction over avoiding casualties or completing broader objectives [4].

That warning is why the reward design should be treated as doctrine encoding rather than as a secondary tuning step. Military decisions are rarely single-objective. Real commanders balance speed, survivability, mission accomplishment, support timing, and constraint satisfaction. A multi-objective reward structure approximates that reality better than a pure win-or-lose signal. The 2024 ReLeGSim article also notes that changing the reward according to the active objective can improve exploration, especially when combined with curriculum learning and human-readable priorities [4]. In this context, curriculum learning means training the agent through staged or progressively structured tasks rather than expecting it to learn the full problem all at once.

4.2 Learning Objective

The second vital equation is the overall reinforcement-learning objective. In reinforcement learning, a policy is the agent's learned decision-making strategy. It describes how the agent chooses an action based on the current state of the environment. In simplified form, the policy is trained to maximize discounted cumulative reward:

$$J(\pi) = \mathbb{E}_\pi \left[\sum_t \gamma^t R(s_t, a_t) \right] \quad (2)$$

In this equation, $J(\pi)$ represents the overall objective, or score, of the policy π. Here, π is not a single parameter; it represents the agent's general learned decision-making strategy. The expected value $\mathbb{E}_\pi$ means that the objective is averaged over the results produced when the agent follows that policy, since the same policy may lead to different results in different simulation runs. The summation $\sum_t$ means that the model adds rewards in many time steps rather than judging only one decision. The discount factor γ^t controls how much future rewards matter compared to immediate rewards. Finally, $R(s_t, a_t)$ is the reward received when the agent takes action a_t in the state s_t.

This equation shows why the reward function matters so much. The policy does not simply maximize immediate rewards; it maximizes expected long-term return. In tactical terms, that means the model can learn to sacrifice short-term convenience for a better operational position later. What is placed in R is, by definition, what the policy will attempt to optimize through $J(\pi)$.

4.3 What the Results Show

In Figure 3 Möbius et al. compare training with one priority against training with three priorities. The multi-priority configuration learns more slowly in early training but later reaches stronger performance, with the authors reporting a win rate increasing to roughly 77% in the examined scenario. Though this win rate is not treated as the only measure of military usefulness. Rather, it is used as the performance metric to show how different reward structures affected training within the simulation [4].

Adding constraints consisting of rules of engagement, commander intent, and a military doctrine with guidelines and principles to follow in an operation does not make optimization easier; it makes the problem harder at the beginning because the policy must satisfy more than one goal. However, the later improvement suggests that broader constraint sets can produce more robust strategies than narrow optimization.

This finding supports the central argument of the paper. Better-aligned behavior is not produced by removing constraints in the name of efficiency. It is produced by training the system to handle competing priorities in a more realistic way. In the discussion of the Figure 3, Möbius et al. attribute the subsequent improvement to greater exploration and a broader repertoire of skills [4]. Simple rewards can lead to narrow strategies, while multi-objective rewards encourage more realistic tactics. That interpretation fits the architecture well, because the whole system is designed to combine a rich state representation with a reward structure that reflects more than one battlefield value.

5 Discussion: Automation Bias, Safety, Current and Future Relevance, and Limitations

The technical strengths of ReLeGSim do not eliminate the central danger of military AI decision support: automation bias may be present where humans may trust the system

Figure 3. Training comparison for one-priority versus three-priority learning, adapted from Möbius et al. [4].

too much, or the system may optimize the wrong objective. Probasco et al. warn that AI-enabled decision support should not be treated as an oracle because battlefield prediction is inherently limited and military environments are shaped by incomplete, changing, and noisy information [4, 5]. Their safety argument complements the ReLeGSim research. The solution is not to reject decision support entirely, but to preserve meaningful human control, bind the system to decision-support roles, and make parts of the basis for recommendations more inspectable. In this framework, with reference to Figure 2, inspectability does not mean that a human can fully read the internal reasoning of the neural network. The internal weights of the model remain difficult to interpret directly. The more inspectable parts are the human-readable command output, the active objective, the reward weights, and the action constraints. These features allow an operator to see what kind of priority the system is optimizing, even if the full internal reasoning of the model remains shadowed.

In this respect, the ReLeGSim design already contains several features that help overcome automation bias. The command interface is readable rather than opaque; the operator can validate, evaluate, or change the output; the objective can be modified during a scenario; and invalid commands are reduced through action masking [3, 4]. None of these features guarantees safety, but together they shift the system away from full autonomy and toward assisted decision-making. That is a meaningful distinction. A support system that exposes command intent and remains interpretable is safer than one that optimizes silently in the background.

Recent military developments make these design choices timely. In 2025 the U.S. Army announced an Army Enterprise Large Language Model Workspace intended to bring secure generative-AI capabilities into Army operations [6]. The Defense Department has also established an AI Rapid Capabilities Cell to accelerate the adoption of advanced AI tools throughout the department [1]. In April 2026, the Army announced the Army Data Operations Center to improve the way war fighters use data for “decision dominance” [7]. These developments do not prove that tactical reinforcement-learning advisors are ready for battlefield use, but they do show that the institutional push toward AI-enabled support is accelerating. As military organizations build more AI infrastructure, the architectural questions raised by ReLeGSim become more important: what objectives are encoded, what data are used, how are outputs constrained, and where exactly does the human remain in control?

For that reason, the most important lesson of the ReLeGSim case study is methodological rather than promotional. The system should be understood as an example of how to responsibly structure AI decision support: make the observation space explicit, use specialized encoders for different battlefield modalities, tie tactical behavior to a transparent reward structure, and keep the interface interpretable enough for humans to contest. A model architecture alone does not solve ethics or policy. But a poorly designed architecture can make those problems worse, whereas a carefully designed one can at least make alignment visible and debatable.

Despite the strengths of the ReLeGSim framework, its reported success should be interpreted as evidence of promise inside a simulation environment rather than proof of immediate battlefield readiness. A simulator can represent terrain, force composition, fog of war, and doctrinal objectives, but it still simplifies the real conditions of war. Actual military operations involve deception, communications failures, political constraints, civilian presence, and human behavior that can be inconsistent or irrational. Those factors are difficult to model fully, which means a policy that performs well in simulation may still fail when transferred to messier real-world settings.

A second limitation involves validation. Reinforcement learning systems can appear effective because they optimize the metrics that designers provide, yet those metrics may not capture every feature of sound tactical judgment. In this sense, the same reward structure that enables doctrinal alignment can also become a source of hidden weakness if it overvalues one objective or fails to anticipate rare edge cases. Before systems like ReLeGSim could be trusted more broadly, they would need extensive testing across diverse scenarios, adversarial conditions, and unexpected failures. Commanders would also need clear evidence not only that the system performs well on average, but that its recommendations remain interpretable, contestable, and interruptible when conditions change suddenly. Being able to interrupt the system allows the human operator to pause, override, reject, or modify the system’s recommendation before it becomes an executed command.

These limitations point directly toward future research. One important direction is broader scenario design that exposes the agent to more varied terrain, force structures, operational goals, and contested information conditions. Another is improved explainability: if a system can show why a recommendation emerged from a particular state and reward

structure, human operators are better positioned to evaluate it critically rather than defer to it automatically. Future work could also explore more explicit ways of encoding doctrine, rules of engagement, and commander intent so that alignment is not treated as a static constraint but as a dynamic part of military decision support. In that sense, the long-term challenge is not only to make AI advisors more capable, but to make them more testable, interpretable, and accountable.

6 Conclusion

AI support systems in the military should not be evaluated only by whether they produce successful outputs in simulation. They should also be evaluated by how their recommendations are generated, constrained, presented, and controlled. ReLeGSim is useful because it makes several parts of that decision pipeline explicit, even though the internal reasoning of the neural network remains difficult for humans to interpret. The model begins with a multi-modal observation space, transforms battlefield data through specialized encoders, reasons over time with a core network, and produces both action and value outputs. Within that architecture, the reward function and learning objective are especially important because they define what the system is trained to value.

The reported results suggest that richer reward structures can improve later performance even when they slow early training. However, simulation success should not be treated as proof of battlefield readiness. Real military decision-making is multi-objective, uncertain, and shaped by conditions that are difficult to fully model. For that reason, human oversight cannot be treated as an afterthought. Readable commands, operator intervention, explicit objective-setting, and clear limits on system authority are part of the safety structure of the system itself.

As military organizations invest more heavily in AI infrastructure, decision-support systems will likely become more common. The value of this paper is not to claim that AI should replace commanders or that ReLeGSim solves military decision-making. Rather, ReLeGSim should be understood as a step toward showing how command intent, doctrinal priorities, and human review can be built into an AI-supported decision process.

Acknowledgments

I would like to thank my advisor and professor, Nic McPhee along with professor Elena Machkasova, for their help with the development and research of this paper.

References

[1] Chief Digital and Artificial Intelligence Office. 2025. *AI Rapid Capabilities Cell.* https://www.ai.mil/Initiatives/AI-Rapid-Capabilities-Cell/

[2] Shilong Li, Chenyi Zhang, and Zhihan Yang. 2025. Research on the Military Application and Development Suggestions of Artificial Intelligence. In *Proceedings of the 2024 2nd International Conference on Artificial Intelligence, Systems and Network Security (AISNS '24)*. Association for Computing Machinery, New York, NY, USA, 8–12. doi:10.1145/3714334.3714336

[3] Michael Möbius, Daniel Kallfass, Thomas Doll, and Dietmar Kunde. 2023. AI-Based Military Decision Support Using Natural Language. In *Proceedings of the Winter Simulation Conference* (Singapore, Singapore) *(WSC '22)*. IEEE Press, 2082–2093.

[4] Michael Möbius, Daniel Kallfass, Matthias Flock, Thomas Doll, and Dietmar Kunde. 2024. Incorporation of Military Doctrines and Objectives into an AI Agent via Natural Language and Reward in Reinforcement Learning. In *Proceedings of the Winter Simulation Conference* (San Antonio, Texas, USA) *(WSC '23)*. IEEE Press, 2357–2378.

[5] Emelia Probasco, Matthew Burtell, Helen Toner, and Tim G. J. Rudner. 2025. Not Oracles of the Battlefield: Safety Considerations for AI-Based Military Decision Support Systems. In *Proceedings of the 2024 AAAI/ACM Conference on AI, Ethics, and Society* (San Jose, California, USA) *(AIES '24)*. AAAI Press, 1157–1165.

[6] U.S. Army Public Affairs. 2025. *Army launches Army Enterprise LLM Workspace, the revolutionary AI platform that wrote this article.* https://www.army.mil/article/285537/army_launches_army_enterprise_llm_workspace_the_revolutionary_ai_platform_that_wrote_this_article

[7] U.S. Army Public Affairs. 2026. *Army launches data operations center to give warfighters the decisive edge.* https://www.army.mil/article/291655/army_launches_data_operations_center_to_give_warfighters_the_decisive_edge U.S. Army.