Proceedings of the 2024 Winter Simulation Conference
H. Lam, E. Azar, D. Batur, S. Gao, W. Xie, S. R. Hunter, and M. D. Rossetti, eds.

CURRICULUM INTERLEAVED ONLINE BEHAVIOR CLONING FOR COMPLEX REINFORCEMENT LEARNING APPLICATIONS

Michael Möbius[1], Kai Fischer[1], Daniel Kallfass[2], Stefan Göricke[3], and Thomas Doll[4]

[1]Operational Analysis and Studies, Airbus Defence and Space, Immenstaad, Germany
[2]Simulations and Studies, Airbus Defence and Space, Immenstaad, Germany
[3]Army Concepts and Capabilities Development Centre, Bundeswehr, Bonn, Germany
[4]Joint Support Service Command, Bundeswehr, Bonn, Germany

ABSTRACT

This paper introduces Curriculum Interleaved Online Behavior Cloning (IOBC) as an approach to train agents for military operations, addressing not only the challenges posed by complex and dynamic combat scenarios but also how military doctrines and strategies are transferred to these agents. It highlights the limitations of traditional reinforcement learning (RL) methods and proposes interleaved online behavior cloning in combination with curriculum learning as a solution to enhance RL agent training. By leveraging rule-based agents for guidance during training, IOBC accelerates learning and improves the RL agent's performance, particularly in early stages of training and complex scenarios. The study conducted experiments using ReLeGSim, a reinforcement learning-focused simulation environment, demonstrating the effectiveness of our method in enhancing agent performance and scalability. Results indicate that IOBC significantly outperforms RL agents without guidance, providing a stable foundation for learning in challenging environments. These findings underscore the potential of IOBC in real-world military applications.

1 INTRODUCTION

The effectiveness of autonomous systems in military scenarios hinges on their ability to quickly adapt and make informed decisions amidst complex and uncertain conditions. Traditional decision-making approaches and reinforcement learning (RL) techniques often struggle with the details of dynamic combat environments. Training RL agents in our environment is time-consuming, taking up to three weeks, and agents tend to opt for straightforward solutions rather than optimal strategies requiring coordination.

Interleaved Online Behavior Cloning (IOBC) offers a solution by providing a structured, iterative learning framework that accelerates the training of autonomous agents. This framework enhances military operational effectiveness by facilitating rapid skill acquisition through expert demonstrations at lower levels, gradually introducing more complex scenarios with reduced support. By incorporating basic military doctrines early on, IOBC prepares RL agents to handle battlefield uncertainties and improve their decision-making processes over time.

IOBC also allows for customized training protocols tailored to specific missions and environments, such as urban navigation or reconnaissance, addressing the unique challenges of each scenario. This adaptability is crucial in modern warfare, where conditions are constantly changing. IOBC integrates rules of engagement, ensuring agents are aware of their actions and encouraging better decision-making when deviations occur.

This paper outlines a reinforcement learning methodology using IOBC and curriculum learning, highlighting its benefits for researchers and practitioners. Key contributions include demonstrating that simple rule-based agents are enough to effectively initiate RL training, showing that IOBC enhances learning and performance. Its simplicity allows IOBC to be integrated into any RL optimization algorithm

as a regularizing loss term, providing a versatile tool for improving autonomous systems in complex environments.

2 RELATED WORK

The integration of domain knowledge into RL agents with imitation learning (IL) approaches such as behavior cloning (BC), especially in complex environments where exploration is challenging, is a powerful technique to transfer domain knowledge to an autonomous agent when one has access to an expert policy to demonstrate desired actions given the current state.

Online Imitation Learning uses expert policies to provide feedback to the learning agent during training. One form of online IL allows the RL agent to interactively query the expert for demonstration at each time step (Ross et al. 2011; Ross et al. 2014; Judah et al. 2014). Specifically, Ross et al. (2011) proposed an iterative IL algorithm called Dataset Aggregation (DAGGER) to reduce the problem of covariance shift during training. Their method iteratively aggregates data from the RL agent's own state distribution, queries the expert policy for action demonstrations at each state, adds the new state-action pairs to the aggregated dataset the RL is trained on.

In their work on learning from imperfect demonstrations, Gao et al. (2019) proposed the Normalized Actor-Critic (NAC) algorithm that uses the same objective to learn initially from noisy demonstrations before transitioning to learning from interactions with the environment. In comparison, IOBC is not restricted by a limited set of human demonstrations and uses expert guidance at every single environment step during the training phase.

Goecks et at. (2020) proposed the Cycle-of-Learning (CoL) framework where they investigated the smooth transition from behavior cloning to reinforcement learning by combining BC and 1-step Q-learning losses with continued re-use of human demonstration data during training to avoid performance degradation during the transition phase.

Context Online Imitation Learning, proposed by Hill et al. (2024) take advantage of existing policies in a strict subset of the state space by adding expert demonstrations to the RL agent's observation. This approach allows the learning agent to take expert demonstrations into account without being directly encouraged to follow the actions of the expert's policy.

Wang et al. (2019) improved reinforcement learning methods by incorporating rule-based trajectory constraints, resulting in better performance of autonomous vehicles. Liu et al. (2023) transferred prior knowledge of an existing rule-based control policy by adding a behavior cloning term to regularize their online policy.

Closest to our work is the work of Zhao et al. (2022) formulates online IL as a sequence of supervised learning problems to encourage the RL agent to perform expert-like actions under its own state distribution. In their work they propose an adaptive weighting scheme of the behavior cloning loss that is based on the agent's performance and overall training stability.

3 METHODS

3.1 The Simulation Environment "ReLeGSim"

ReLeGSim (*Re*inforcement *Le*arning focused *G*eneric AI Training *Sim*ulation, shown in Figure 1) is a rasterized simulation environment for reinforcement learning to develop self-optimizing strategies of the players in the game. Arbitrary players are supposed to reach a defined goal by performing a sequence of actions like moves, requests for support, and interactions with each other. ReLeGSim can be used to model a broad-range of civil and military scenarios such as an ISR mission or a joint battalion ground combat scenario. ReLeGSim allows to define flexibly actors (e.g., single-agent Btl, Multi-Agent UAS) for the game-like environment, to give them corresponding properties and possible actions. For this purpose, the simulation can be extended by appropriate application-specific simulation models (such as a sensor) using the Python programming language.

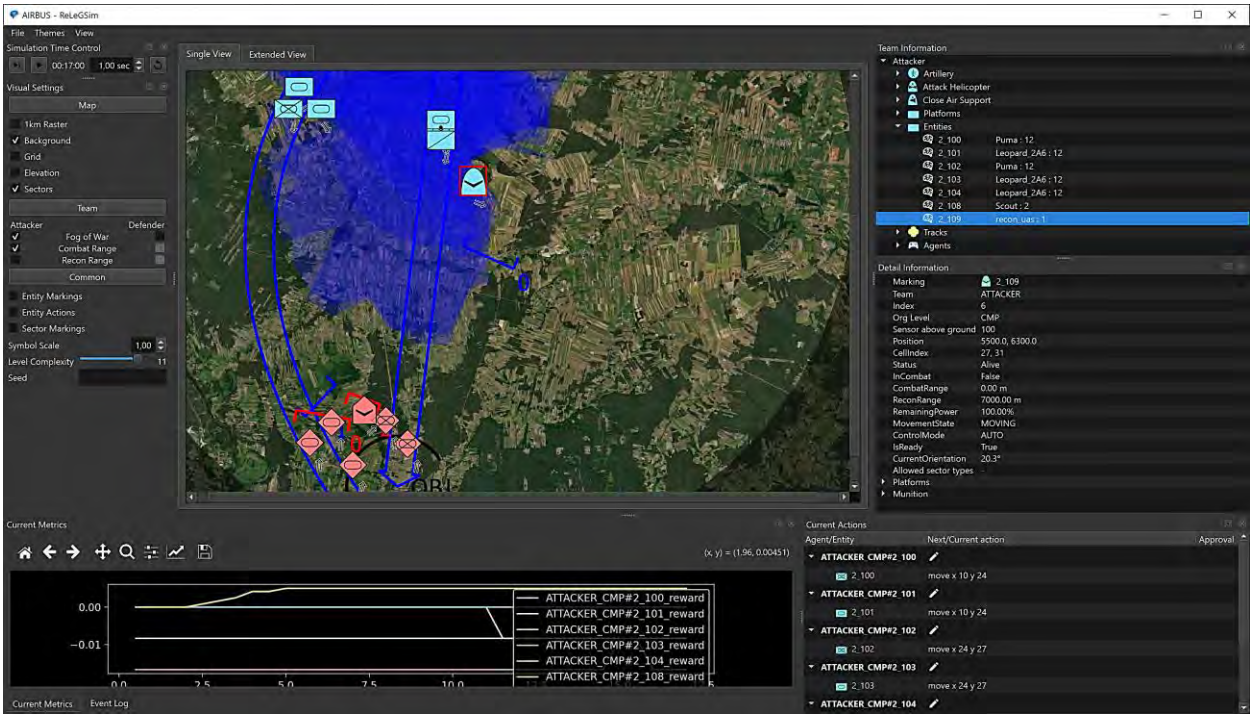

Figure 1: ReLeGSim Expert-UI.

The simulation was last shown at the NATO MSG-207 symposium (Möbius et al. 2023) and uses the "Gymnasium"-API for the connection to reinforcement learning applications. The architecture supports single-agent, multi-agent and hierarchical reinforcement learning.

In ReLeGSim, learning agents encounter a series of challenging problems. ReLeGSim represents a dynamic multi-agent environment where agents must consider the interactions of numerous entities. Due to a partially observed map, actors face the problem of imperfect information. Actors perceive the state space exclusively through raw image and vector features which adds to the difficulty of learning and navigating the environment. The selection and control of many different units comes with a large action space. Furthermore, the environment requires actors to plan over long-term time horizons with hundreds of steps with the difficulty of delayed credit assignment.

In more detail, one actor takes on the role of the attacker, aiming to capture a specific target area from the defender, who must hold it throughout the episode. Both players have access to various groups of soldiers with unique capabilities, consisting of companies, platoons and individual units. To succeed, players must understand their opponent's perspective, know their unit's abilities, and effectively navigate the terrain. The visualization of the state-value estimated by the AI-Model allows the user to understand the AI's behavior better. For example, a low value will explain defensive actions to avoid huge negative reward.

As part of the MLOps-Architecture (Figure 2), the scenario generator allows for the automatic creation of 3D terrain, based on real-world data, such as vector, elevation, and satellite information. This information is embedded into a rasterized map with specific field types (e.g., water, forests, hills, agriculture or roads) and is then used for AI training within ReLeGSim. The simulation framework aims to provide a platform for training different AI models through reinforcement learning, but it also supports human vs. AI gameplay. Therefore, it is possible to benchmark, test, evaluate, and analyze the capabilities of the trained agents. The toolchain also includes automated testing for trained AI agents with various metrics and complex analysis based on customer needs.

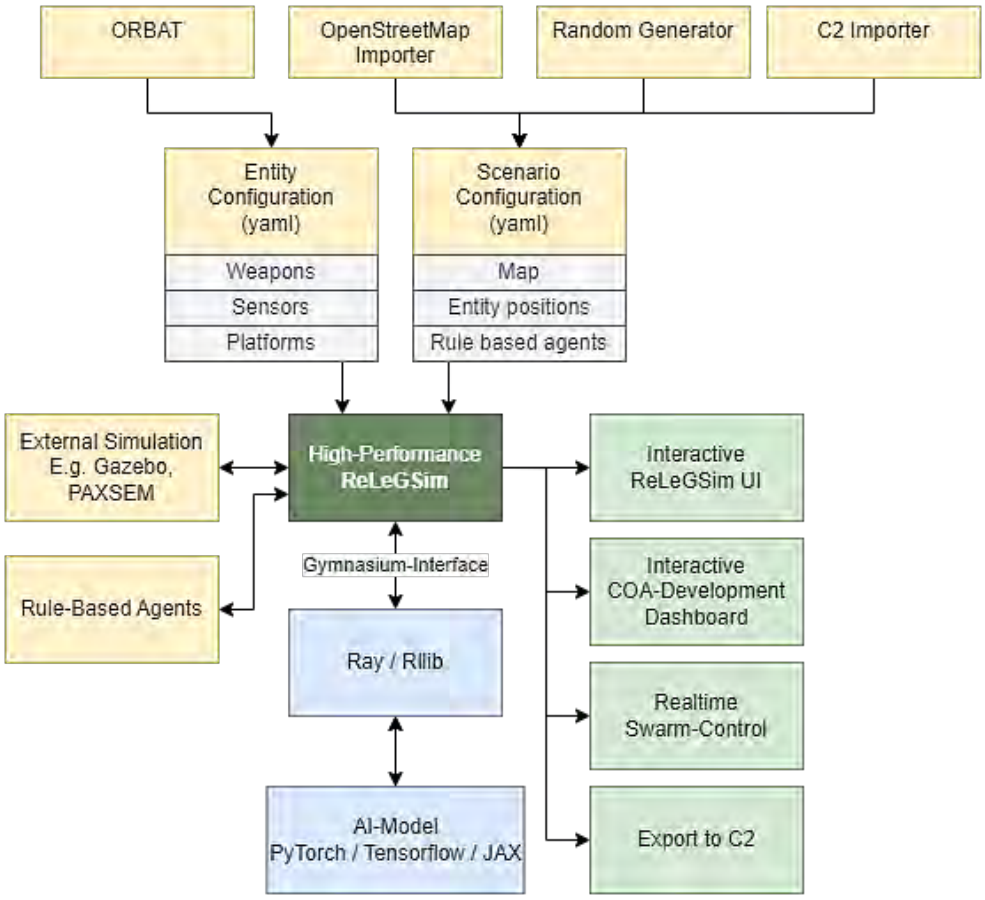

Figure 2: MLOps-Toolchain for ReLeGSim.

3.2 ReLeGSim AI Architecture

The authors of ReLeGSim (Doll et al. 2021) drew inspiration from DeepMind's AlphaStar (Vinyals et al. 2019), a leading model in complex RL problems, to develop an innovative architecture (Figure 3). Influenced by military tactics, the design utilizes scalar data (e.g., troop numbers and ammunition) and visual maps for scenario observation. To understand the terrain, the AI receives a visual map with a lot of terrain information and entity encodings. A spatial encoder with convolutional layers was developed to incorporate this rich data into the AI.

The model architecture reduced to the minimum via an autoencoder setup, reducing parameters and producing a pre-trained model. A language input can take an objective or task into consideration. In a hierarchical setup, the given task is defined by a superior agent. Encoded values from visual, task, and scalar data are input to a core network, an LSTM component, to handle long-term planning.

The action head was initially implemented as a multi-discrete action space based on the AlphaStar implementation. Due to the exploding action space, the action head was replaced by a language model based on the latest research to predict action commands in natural language, as it will be described in the next section.

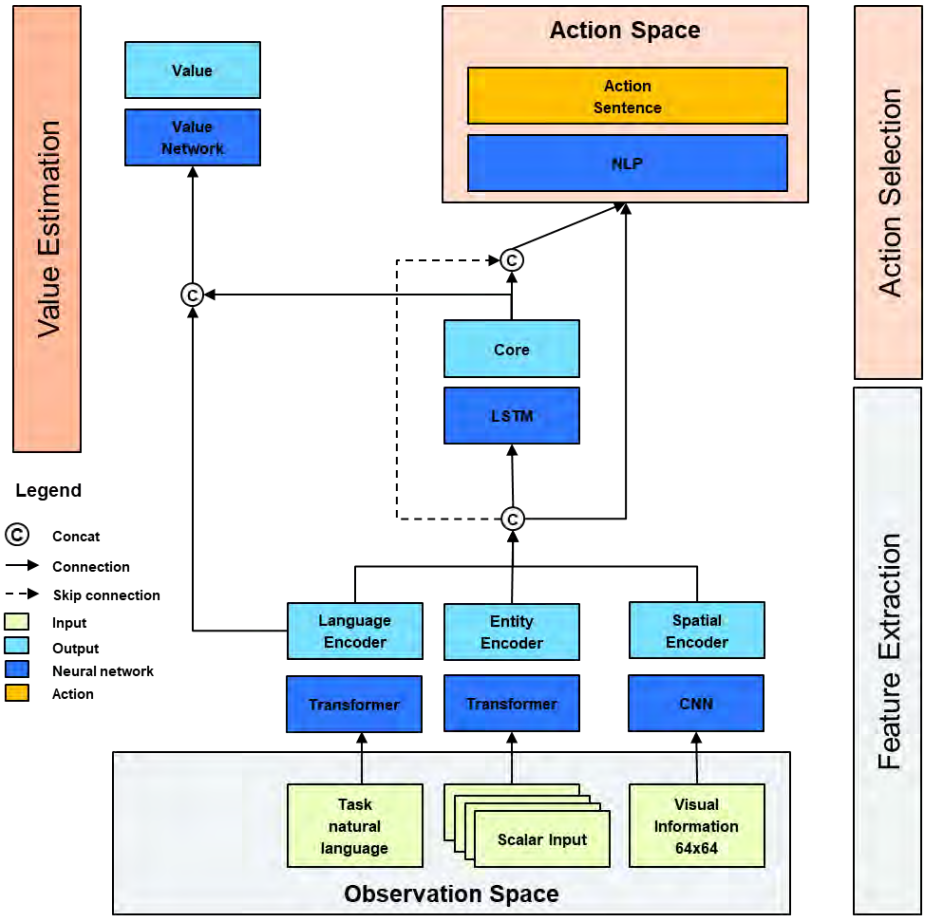

Figure 3: AI Architecture used in ReLeGSim.

3.3 AI Decision Space

The issue of complex decision-making capabilities of AI comes with huge action spaces in RL, which presents a significant challenge as RL applications become more complex and realistic. Small, fixed-action spaces have limitations in terms of expressiveness, exploration, and efficiency. Researchers are continually developing new techniques and algorithms to mitigate the impact of exploding action spaces, such as function approximation, discretization, and hierarchical RL. These approaches enable RL agents to tackle increasingly complex tasks and navigate the challenges of large action spaces more effectively. As RL continues to advance, addressing the issue of exploding action spaces will remain a critical research area to enable the successful application of RL in real-world scenarios. The approach of employing natural language to establish communication with AI, as showed in Möbius et al. (2022), coupled with the developments in the formulation of doctrines using natural language (Möbius et al. 2023), set a precedent for the realization of a versatile AI capability within a multifaceted operational environment. ReLeGSim incorporates a natural language interface between the AI and the agents in the simulation with a complex parsing and execution of given commands. These commands can be on different hierarchical levels and control various agents.

Initial trials showed that a large space of unused vocabulary is disadvantageous and results in slow training. Therefore, we use a small, but effective vocabulary which is inspired by the Battle-Management-Language (BML). The full vocabulary for an environment with a resolution of 64x64 pixels contains the following tokens:

`0, … ,63, x, y, left, right, attack, move, observe, scan, hide, recharge, drop, own_entity, enemy_entity, artillery, attack_helicopter, close_air_support, mines, <colon>`

The used tokens can be masked depending on the current simulation state. For example, the token `enemy_entity` is masked if no enemy is visible to the agent. The token `<colon>` splits the resulting output text sequence into multiple actions, additionally the token can be used to pad the sentence. The reduction of tokens and the optimization were done manually and corresponded directly to the execution of the resulting behavior in the simulation. To tokenize the actions, we use one-hot encoding, as this allows us to use masked stochastic sampling over the given actions and can be easily integrated into any given RL framework through a multi-discrete representation.

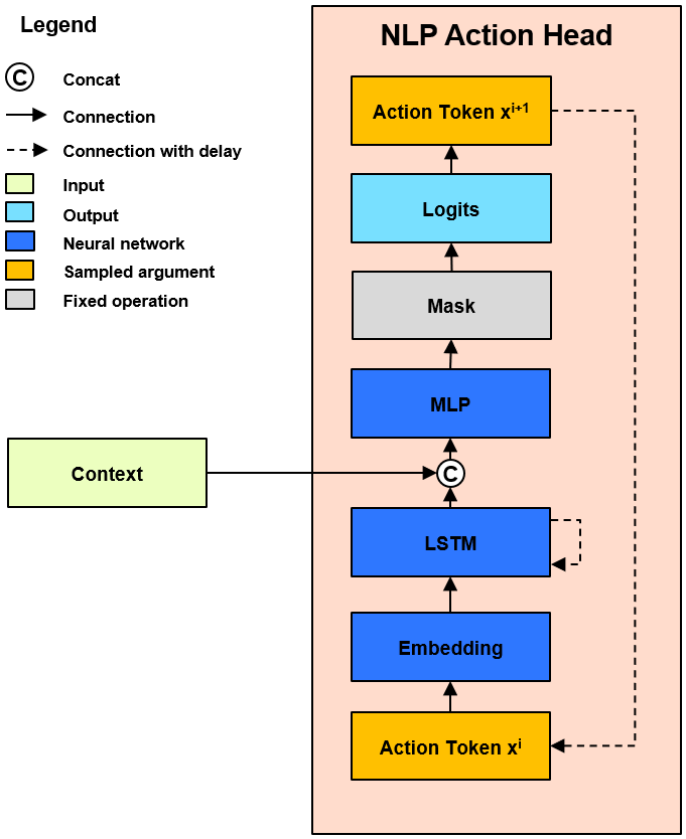

Figure 4: Architecture of the autoregressive NLP action head.

3.4 Curriculum-Based Scenario Creation

In our simulation framework we follow a curriculum learning approach (Narvekar et al. 2020), where we employ incremental levels of difficulty across various scenarios to effectively train, validate, and operate our agents. The training scenarios are designed to incrementally increase in complexity, providing a gradual learning curve for agents. Validation scenarios serve as test levels to ensure the competency of trained agents. Operations scenarios, on the other hand, present highly complex situations involving joint operations and numerous units, which are too challenging for rule-based agents to handle efficiently. Other than human operators only reinforcement learning agents demonstrate successful performance in these scenarios. Moreover, the model architecture and size play a crucial role in shaping the capabilities of our agents. Illustrating the practical application of our framework, we showcase IOBC for single-agent battalion

commander scenarios focusing on strategic decision-making of a single commanding officer. This use-case offers a wide range of scenarios with unique challenges and objectives, thereby providing a comprehensive platform for testing and refining agent capabilities.

3.5 Interleaved Online Behavior Cloning

Instead of learning only from environmental feedback (rewards), the RL agent is guided by the actions of one or more rule-based agents that are active in a subset of levels. Our rule-based agents follow military tactics and doctrines to perform actions at each time step during the episode rollout. Rule-based agents are only used for scenarios up to a certain level of complexity where it can be ensured, that the scripted behavior represents a good expert strategy with a high probability of success (e.g., attacking from opposite site in a 2 vs. 1 scenario).

At each time step, the rule-based agent provides an action during the episode rollout considering the current state of the environment. The presence of these state-action tuples allows us to formulate the learning process as a supervised learning problem enabling the RL agent to learn useful actions during the initial training period. Besides a strong learning signal during the early training stage, IOBC comes with further benefits: As IOBC is an *online* method, it allows us to train agents with memory components (state of LSTM) for effectively modeling long-term dependencies right from the start of training. Furthermore, IOBC allows to train the actor and the critic at the same time and overcomes the problem of untrained critics after actor pre-training that often results in forgetting good initial policies.

We implement IOBC with a rule-based agent that runs during the reinforcement learning training as a background process and is synchronized with the learning agent. The rule-based agent's generated actions are added to the training batch and used later for the loss computation. As the actions of our rule-based agents are cheap to compute, the impact on the total training time is negligible.

3.6 Online Behavior Cloning Loss

In IOBC a rule-based agent's guiding policy π_{RB} produces state-action pairs (s_t, a_t) at each time step during the episode rollout, providing the learning agent with a set of demonstrations of the form $\big(s, \pi_{RB}(s)\big)$, where $\pi_{RB}(s)$ is the action a_{RB} the guide policy takes in state s. The RL-agent's goal is to learn a policy π_{RL} that imitates π_{RB}. We transfer the guiding policy to the RL agent by incorporating a regularization term into the Asynchronous Proximal Policy Optimization (APPO) objective function $J(\theta)$ as follows:

$$J(\theta) = J_{APPO}(\theta) + \beta \cdot J_{IOBC}(\theta),$$

where $J_{APPO}(\theta)$ represents the original objective function, $J_{IOBC}(\theta)$ is the IOBC regularization term, and β the weighting parameter for the IOBC objective. The regularization objective, defined as a log-likelihood objective, can be expressed as follows:

$$J_{OBC}(\theta) = \mathbb{E}_{s\sim\rho_\pi, a\sim\pi_\theta}\big[\log\big(\pi_{RL}(a|s)\big)\big],$$

where ρ represents the distribution of states experienced by policy π_{RL}.

3.7 Online Behavior Cloning Coefficient

Imitating the actions of a rule-based agents provides a good source of learning signal at the early stages of training. However, the limited capabilities of hand-crafted rule-based agents provide a good learning signal only during the early stages of a curriculum learning setup with increasingly complex and more difficult levels. In rare and challenging scenarios our guiding policies quickly reach their performance limits and reinforcement learning has to take over. For this reason, a transition period from rule-based guidance to pure reinforcement learning is implemented, continually reducing the influence of the rule-based guidance during training and eventually disappearing completely. This also mitigates the risk of overfitting to the

behavior of rule-based agents, limiting the exploration capabilities of the RL agent. We use a linear decay schedule over N episode steps where we reduce the loss coefficient β to zero over N episode steps.

3.8 Experimental Setup

We train our agents following an incremental learning strategy where they are confronted with randomly generated ground combat scenarios at a battalion level. Higher levels come with more complex scenarios and are sampled more often during training. To reach the next level, agents must achieve a win rate of 70% of the current highest level. Initially we expose the agent to the first 3 out of 12 levels. In level 1 and 2, we use a rule-based agent to guide combat units to move to a predefined position on the map. Level 3 adds complexity by introducing enemy units. For this level we use a rule-based agent that performs a flanking attack to eliminate enemy units. This training procedure steers the RL agent's actions during the early stages of training and interleaves levels without guidance as the agent progresses to higher levels.

3.9 Experiments

In our experiments we compare baseline RL agents without rule-based guidance to training runs that use IOBC. For the training we use RLLib's (Liang et al. 2018) implementation of the APPO algorithm (Schulman et al. 2017) with custom extensions for masked autoregressive action generation (Figure 4) and interleaved online behavior cloning. Unless stated otherwise, we use $\beta = 2$ with 40 M decay steps during which the coefficient is reduced to zero.

4 RESULTS

4.1 Optimizing Loss Coefficient Dynamics in Online Behavior Cloning

The loss coefficient β, controlling the strength of guidance during training, has a significant impact on the final performance of our agents. Figure 5 shows the agent's performance index (aggregated win rates over all levels) for $\beta = 2$ with 100 M (red) and 300 M (green) decay steps. Simple rule-based agents can only provide high-signal actions during the early stages of training. Guidance of rule-based agents with inadequate actions can interfere with exploration and inhibits the RL agent from learning.

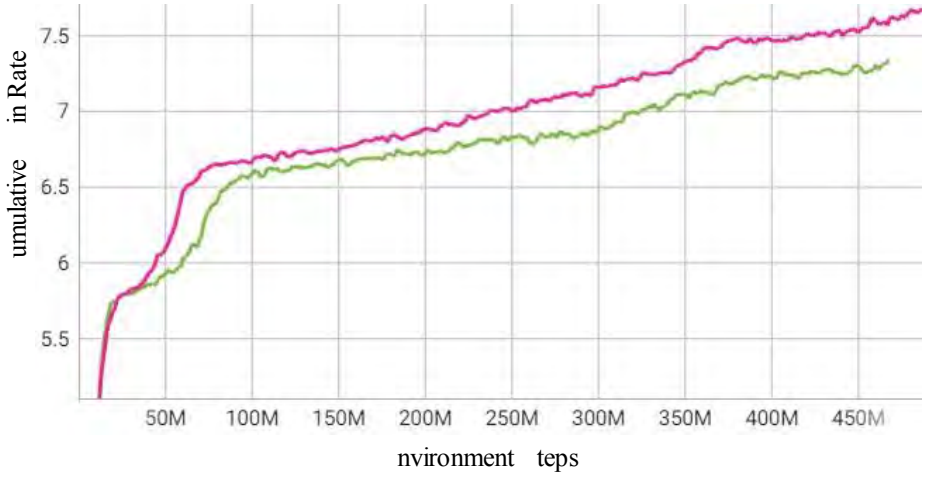

Figure 5: Impact of guidance during training on the RL agent. Longer guidance can lead to worse results. Cumulative win rate as a function of environment steps shown for $\beta = 2$ and 100 M (red) and 300 M (green) decay steps.

The following results show that the initial value of β has a significant impact on the learning process of IOBC agents. We run two experiments with $\beta = 1$ and $\beta = 4$ and 20 M decay steps. Figure 6 shows that

while high values boost the agent's performance in the early stages of training (blue), lower values let the agent reach higher win rates later in training as they leave the agent more room for exploration (black).

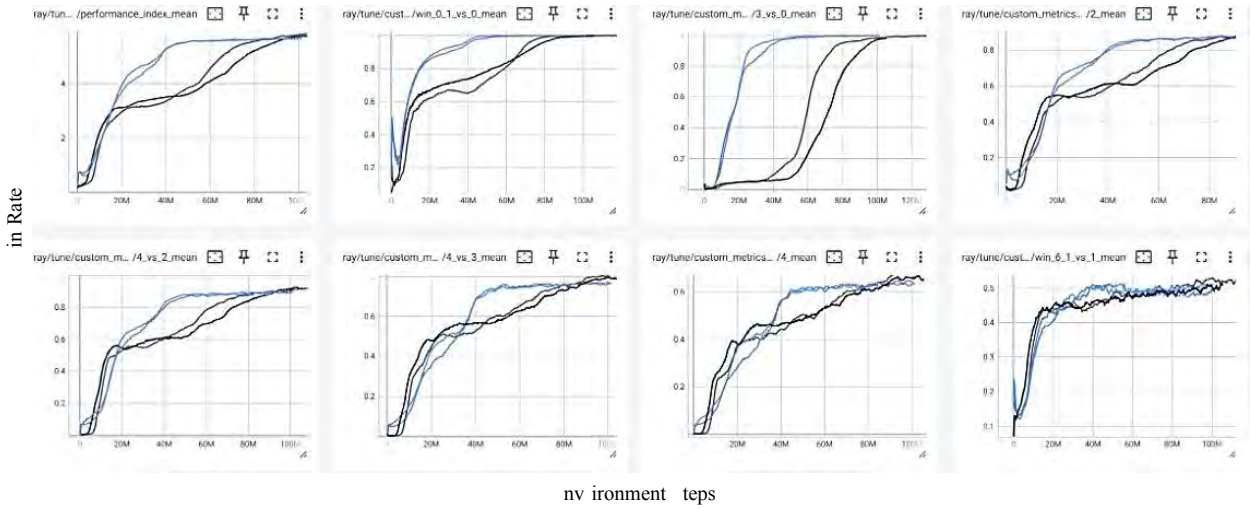

Figure 6: While higher loss coefficients boost performance during the early stages of training (blue), lower values reach higher final win rates (black).

4.2 Single-Agent Battalion Commander Scenarios

In all levels with expert guidance (1, 2, and 3) IOBC significantly outperforms pure RL training showing the strong effect of transferring domain specific knowledge with rule-based regularization on to the RL agent. Figure 7 shows that without guidance, the baseline RL-agent performs significantly worse during the early stages of training and is converging much later.

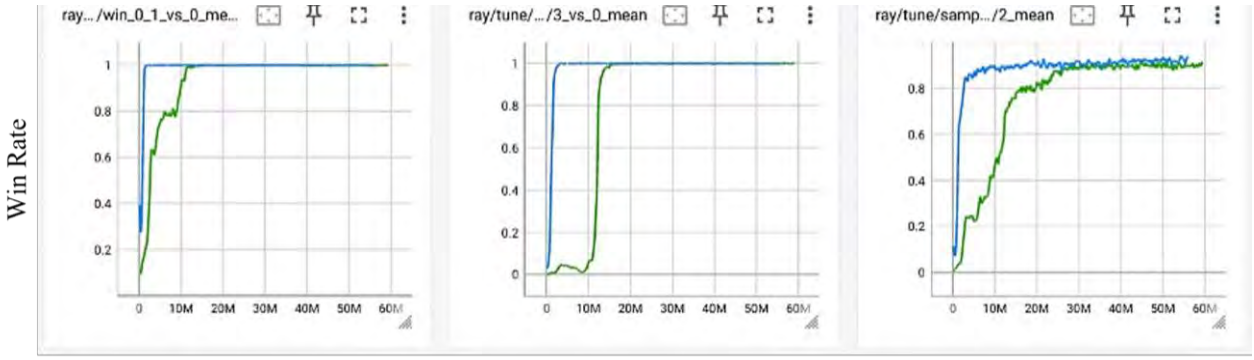

Figure 7: RL-agents reaches near perfect win rates with IOBC (blue) and baseline without guidance (green) for levels 1, 2 and 3 (from left to right). The IOBC agent reaches high win rates in significantly fewer steps.

As the RL-agent is guided in the first 3 out of 12 levels, the rest of the evaluation focuses on the performance in the remaining levels. Win rates for levels without guidance by an expert policy, show that the agent learns good actions faster and that the IOBC agent reaches higher levels significantly earlier. Furthermore, we observe higher win rates of the IOBC agent in higher and more difficult levels. The overall performance on all levels shows a significant advantage of the integration of IOBC into the training process of autonomous agents. Especially during the early stages of training, results obtained with IOBC significantly outperform the baseline RL agent.

Training runs performed with IOBC clearly outperform the baseline RL agent without guidance (Figure 8). Despite the only moderate win rate of only 45% of the rule-based flanking attacker, the provided guidance allows the RL agent to learn actions much faster (< 50% steps) compared to the pure RL baseline.

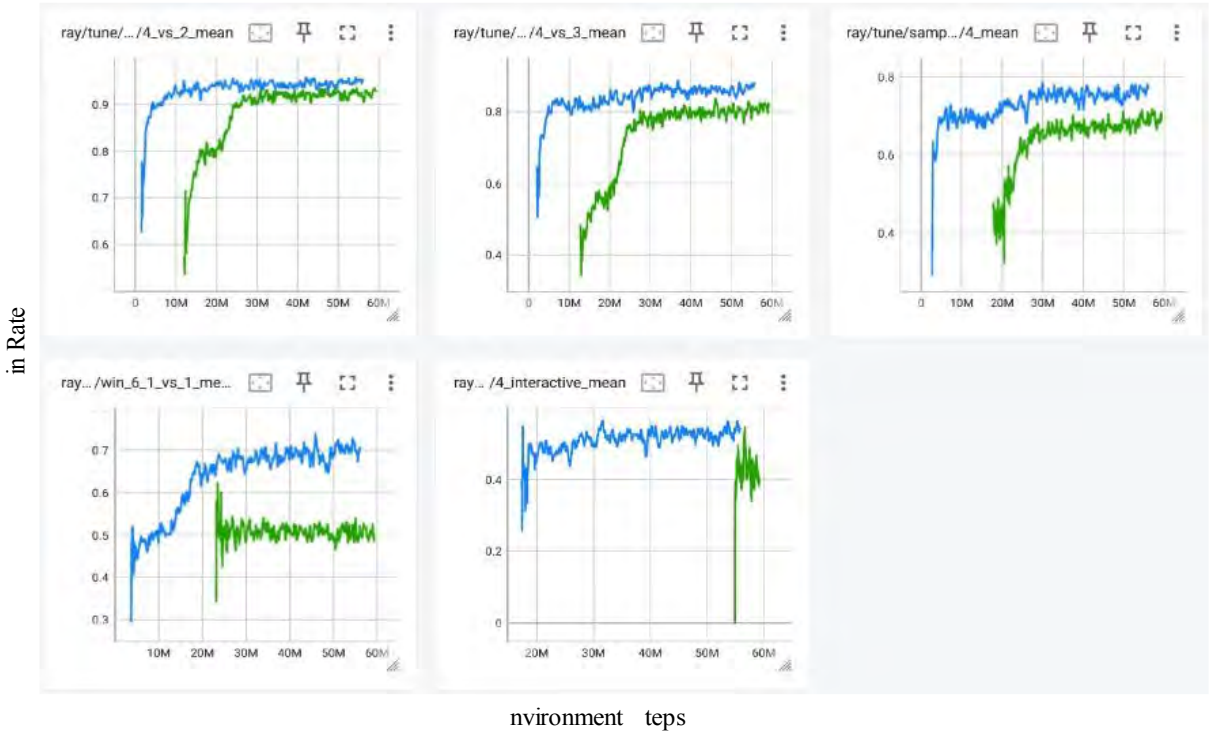

Figure 8: The IOBC agent (blue) outperforms the RL agent (green), reaches higher levels earlier, and achieves higher win rates.

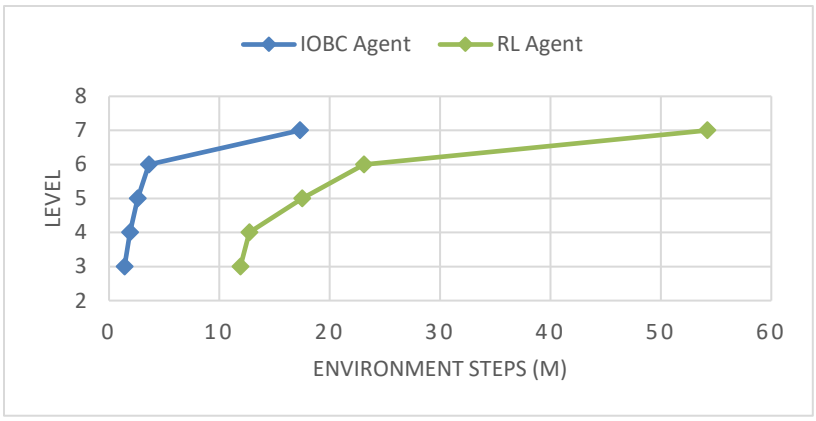

Figure 9: Next level reached as a function of environment steps (in million). The IOBC agent reaches higher and more difficult levels earlier during training.

Figure 9 shows the number of required environment steps to reach the respective level. Despite receiving help in only the first three levels, the IOBC agent receives higher levels earlier and performs significantly better compared to the RL agent. The highest level could be reached with IOBC in ~1/3 of steps. Figure 10 shows win rates for different levels with and without IOBC at the end of training. As to be expected, the win rates decrease for both agents in higher and thus more complex levels. Once again, it can be seen that agents trained with IOBC achieve higher win rates in all levels at the end of training.

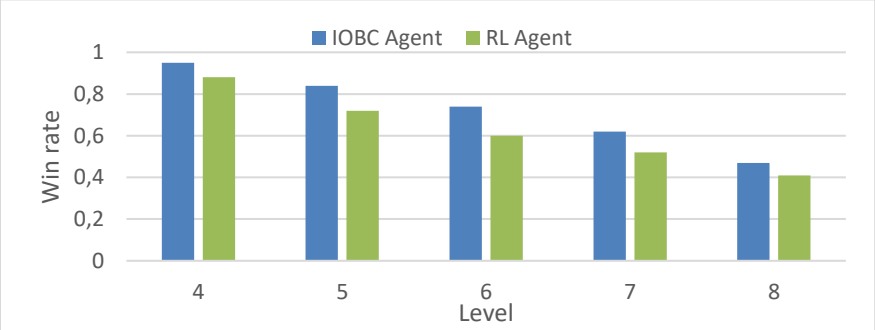

Figure 10: Final win rates for levels without guidance. IOBC agents reach higher win-rates in each level.

5 DISCUSSION AND CONCLUSION

We have shown, that by leveraging expert demonstrations alongside a traditional APPO learning algorithm in a subset of levels, enables agents to be trained more quickly. Our method enables agents to solve complex tasks significantly faster and achieve higher performance scores much earlier in the training process. Our method is simple to implement, can be integrated into any reinforcement learning algorithm, and shows strong results even for simple rule-based agents as guides. The strong initial learning signal makes IOBC particularly compelling for complex policies that involve memory components and autoregressive action generation. These policy elements can be challenging to train effectively when starting from a random policy and dealing with noisy gradients. To avoid poor generalization of the learning agent, the initial choice of the behavior cloning coefficient and reducing the influence of the rule-based guidance are key factors in training a successful agent that can effectively explore and adapt to novel situations.

An interesting direction of future work could be to leverage scenario-wise expert policies. With our method, the collective capabilities of the individual agents could then be distilled into a single agent, resulting in a more comprehensive and capable system. These domain experts could be pre-trained RL agents that were trained on a small subset of scenarios. Developing rule-based agents for complex environments can be highly challenging due to the difficulty in crafting comprehensive and effective rule sets. Future studies might address this by replacing rule-based agents with large language models (LLMs), which could allow distilling the rich knowledge representation of an LLM into a much smaller and more efficient policy for a reinforcement learning agent.

ACKNOWLEDGMENTS

In memoriam of LTC Dr. Dietmar Kunde from the German Army Headquarters, whose enduring mentorship, guidance, and dedication to the practical application of AI continue to inspire and drive our research. In addition, we extend our gratitude to the "KITU" and "ReLeGs" study groups, with special recognition to LTC Matthias Kuc from the German Army Headquarters and Prof. Oliver Rose from the University of the Bundeswehr Munich for their invaluable contributions to our research activities.

REFERENCES

Doll, T., J.- . Brendecke, M. Behm, and D. Kallfass. 2021. "From the Game Map to the Battlefield – Using DeepMind's Advanced AlphaStar Techniques to Support Military Decision-Makers". In *NATO MSG-184: Towards Training and Decision Support for Complex Multi-Domain Operations*, October 21st-22th, Amsterdam, Netherlands.

Gao, Y., H. Xu, J. Lin, F. Yu, S. Levine, and T. Darrell. 2018. "Reinforcement Learning from Imperfect Demonstrations". *arXiv preprint arXiv:1802.05313*.

Goecks, V.G., G.M. Gremillion, V.J. Lawhern, J. Valasek, and N.R. Waytowich. 2019. "Integrating Behavior Cloning and Reinforcement Learning for Improved Performance in Dense and Sparse Reward Environments". *arXiv preprint arXiv:1910.04281*.

Hill, A., M. Groefsema, M. Sabatelli, R. Carloni, and M. Grzegorczyk. 2024. "Contextual Online Imitation Learning (COIL): Using Guide Policies in Reinforcement Learning". In *International Conference on Agents and Artificial Intelligence*, February 24th-26th, Rome, Italy.

Judah, K., A.P. Fern, T.G. Dietterich, and P. Tadepalli, 2014. "Active Limitation Learning: Formal and Practical Reductions to IID Learning". *Journal of Machine Learning Research* 15(1): 3925-3963.

Liang, E., R. Liaw, R. Nishihara, P. Moritz, R. Fox, K. Goldberg et al. 2018. "RLlib: Abstractions for Distributed Reinforcement Learning". In *International Conference on Machine Learning*, July 10th-15th, Stockholm, Sweden, 3053-3062.

Liu, H.Y., B. Balaji, R. Gupta, and D. Hong. 2023. "Rule-Based Policy Regularization for Reinforcement Learning-based Building Control". In *Proceedings of the 14th ACM International Conference on Future Energy Systems*, June 21st-23rd, Orlando, USA, 242-265.

Möbius, M., D. Kallfass, T. Doll, and D. Kunde. 2022. "AI-based Military Decision upport Using Natural Language". In *2022 Winter Simulation Conference (WSC)*, 2082-2093 http://dx.doi.org/10.1109/WSC57314.2022.10015234.

Möbius, M., D. Kallfass, T. Doll, and D. Kunde. 2023. "Natural Language AI for Military Decision Support and Swarm Control for Autonomous UAS Trained in a Combat Simulation". In *NATO MSG-207: Simulation: Going Beyond the Limitations of the Real World*, October 19th-20th, Monterey, CA, USA.

Möbius, M., D. Kallfass, T. Doll, and D. Kunde. 2023. "Incorporation of Military Doctrines and Objectives into an AI Agent Via Natural Language and Reward in Reinforcement Learning". In *2023 Winter Simulation Conference (WSC)*, 2357-2367 http://dx.doi.org/10.1109/WSC60868.2023.10408462.

Narvekar, S., B. Peng, M. Leonetti, J. Sinapov, M.E. Taylor, and P. Stone. 2020. "Curriculum Learning for Reinforcement Learning Domains: A Framework and Survey". *Journal of Machine Learning Research* 21(181): 1-50.

Ross, S. and J.A. Bagnell. 2014. "Reinforcement and Imitation Learning via Interactive No-Regret Learning". *arXiv preprint arXiv:1406.5979*.

Ross, S., G. Gordon, and D. Bagnell. 2011. "A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning". In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, April 11th-13th, Ft. Lauderdale, USA, 627-635.

Schulman, J., F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. 2017. "Proximal Policy Optimization Algorithms". *arXiv preprint arXiv:1707.06347*.

Vinyals, O., I. Babuschkin, W.M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung et al. 2019. "Grandmaster Level in StarCraft II using Multi-Agent Reinforcement Learning". *Nature* 575(7782): 350.

Wang, J., Q. Zhang, D. Zhao, and Y. Chen. 2019. "Lane Change Decision-Making through Deep Reinforcement Learning with Rule-Based Constraints". *International Joint Conference on Neural Networks (IJCNN)*, July 14th -19th, Budapest, Hungary.

Zhao, Y., R. Boney, A. Ilin, J. Kannala, and J. Pajarinen. 2022. "Adaptive Behavior Cloning Regularization for Stable Offline-to-Online Reinforcement Learning". *arXiv preprint arXiv:2210.13846*.

AUTHOR BIOGRAPHIES

MICHAEL MÖBIUS is a system engineer at Airbus Defence and Space in Germany, leading AI and software projects in the "Operational Analysis and Studies" department. His research focuses on large language models (LLMs), stochastic simulation, autonomous systems, reinforcement learning, and operational analysis. His email address is michael.moebius@airbus.com.

KAI FISCHER is a machine learning engineer working in the "Operational Analysis and Studies" department at Airbus Defence and Space in Germany. His primary interests include designing, implementing, and optimizing advanced deep learning models and AI-powered systems. His email address is kai.fischer@airbus.com.

DANIEL KALLFASS is senior expert at Airbus Defence and Space in 3D simulation of System of Systems. With over 18 years of experience in defense and security research, he specializes in simulation-based operational analysis, distributed simulations, and AI, focusing on advanced data farming and deep reinforcement learning. His email address is daniel.kallfass@airbus.com.

MAJ STEFAN GÖRICKE, of the German Army Concepts and Capabilities Development Center, focuses on constructive simulation and AI. He holds dual Master's degrees in c onomics from Helmut chmidt University and in Modeling and imulation from the Naval Postgraduate School in Monterey, California, USA. His email address is stefangoericke@bundeswehr.org.

LTC THOMAS DOLL is a German Army officer currently working for the German Joint Support Service Command, where he is responsible for conducting military analyses and studies. His main areas of interest are modeling and simulation, the development and deployment of unmanned systems, and artificial intelligence. From 1990 to 1994, he studied electrical engineering at the University of the German Armed Forces in Munich. He received his Master of Science degree in 2004 from the Naval Postgraduate School in Monterey, California, USA. His email address is thomasmanfreddoll@bundeswehr.org.