This is the author's accepted version of the article. The final version published by IEEE is A. Duttagupta, M. Jabbari, C. Fiandrino, M. Fiore, and J. Widmer, "Interactive Explanation and Steering of DRL Agents for Massive MIMO Scheduling with SymbXRL," IEEE INFOCOM 2025 Demo Session - IEEE Conference on Computer Communications, doi: TBD.

Interactive Explanation and Steering of DRL Agents for Massive MIMO Scheduling with SymbXRL

Abhishek Duttagupta*†◊, MohammadErfan Jabbari*◊, Claudio Fiandrino*, Marco Fiore* and Joerg Widmer*
*IMDEA Networks Institute, Spain, †Universidad Carlos III de Madrid, Spain
Email: {name.surname}@imdea.org

***Abstract*—Future 6th-generation (6G) mobile networks will increasingly rely on Deep Reinforcement Learning (DRL) for real-time decision optimization. However, DRL's opaque nature hinders its adoption, as operators need to understand and control these complex systems, necessitating explainability tools to reveal the model's reasoning. This paper demonstrates SymbXRL, an EXplainable Reinforcement Learning (XRL) framework that translates DRL's internal logic into human-interpretable symbolic representations and enables intent-based action steering. We introduce a novel interactive dashboard that enhances transparency and control by providing a real-time view of the DRL agent's operation. Our demonstration showcases how SymbXRL i) generates human-readable explanations using symbolic Artificial Intelligence (AI) and knowledge graphs, (ii) enables operator-defined, intent-based action steering for performance improvement, and (iii) provides real-time visualization of agent behavior and network metrics. We demonstrate SymbXRL using a DRL agent that schedules users in a Massive MIMO scenario, leveraging real-world channel measurements from a 64-antenna testbed to maximize spectral efficiency while maintaining fairness.**

I. Introduction

Deep Reinforcement Learning (DRL) holds great promise for managing complex network tasks in future 6G networks [1]. However, its opaque nature reduces trust and may prevent full-scale deployment. SymbXRL [2] addresses this through symbolic AI: it translates DRL states and actions into human-interpretable logic expressions and knowledge graphs, enabling explainability and operator-driven action steering (Fig. 1).

In this Demo paper, We demonstrate SymbXRL, using an interactive dashboard (Fig. 1, ❶) to provide explanations for the decisions of a DRL agent in a Massive MIMO scheduling Environment (Fig. 1, ❷). This scenario, vital for 6G, uses a real-world dataset[3] derived from a 64-antenna software-defined massive MIMO Base Station (BS) serving seven clients. The environment includes both Line of Sight (LoS) and Non-Line of Sight (NLoS) conditions. Users are dynamically grouped based on channel correlation to optimize beamforming. A Soft Actor-Critic (SAC) DRL agent (Fig. 1, ❸) makes scheduling decisions to maximize spectral efficiency while maintaining user fairness. The agent bases its actions on processed Key Performance Indicators (KPIs), including Maximum Spectral Efficiency of User (MSE) and Data Transmitted of User (DTU), as well as user grouping information, thereby scheduling users based on their channel conditions and performance history.

Through the interactive dashboard (Fig. 1, ❶), our demo illustrates how SymbXRL effectively translates the agent's DRL decision-making process into intuitive explanations. This is achieved by visualizing the process through dynamic knowledge graphs and showing the impact of intent-based action steering on the agent's overall performance.

◊These authors contributed equally to this work.

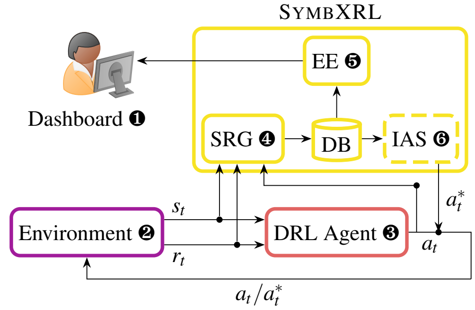

Fig. 1. SymbXRL's architecture showing interaction between its core components (SRG, EE, IAS) and the DRL agent.

II. Demonstration Overview

A. Symbolic Representation

SymbXRL's Symbolic Representation Generator (SRG) (Fig. 1, ❹) converts the numerical states of the environment and the decisions of the agent into First-Order Logic (FOL) terms and stores these symbolic representations in a dedicated database (DB) (Fig. 1) for persistent storage and retrieval. States, including KPIs such as Maximum Spectral Efficiency of User (MSE) and Data Transmitted of User (DTU), are represented as their magnitude (using quartiles Q1-Q4) and trends (using predicates). For instance, `inc(MSEUr, Q4)` indicates a high spectral efficiency of a user increasing to values in the fourth quartile. Scheduling decisions are encoded as `sched(group_number, quartile, percentage)`, where "percentage" is the proportion of scheduled users in the group and the "quartile" is calculated relative to the total number of users.

B. Interactive Experience

SymbXRL's Explanation Engine (EE) (Fig. 1, ❺) retrieves stored symbolic representations from the DB to generate human-readable explanations of agent behavior through Knowledge Graph (KG) and probability distributions. The interactive dashboard (❶) dynamically loads these representations from the DB, providing a real-time view of the DRL agent's operation each timestep. The dashboard comprises the Topology Graph, visualizing BS and users with color-coded group assignments, highlighting scheduled users. The KG visualizes behavioral patterns via action transitions and probability-weighted nodes

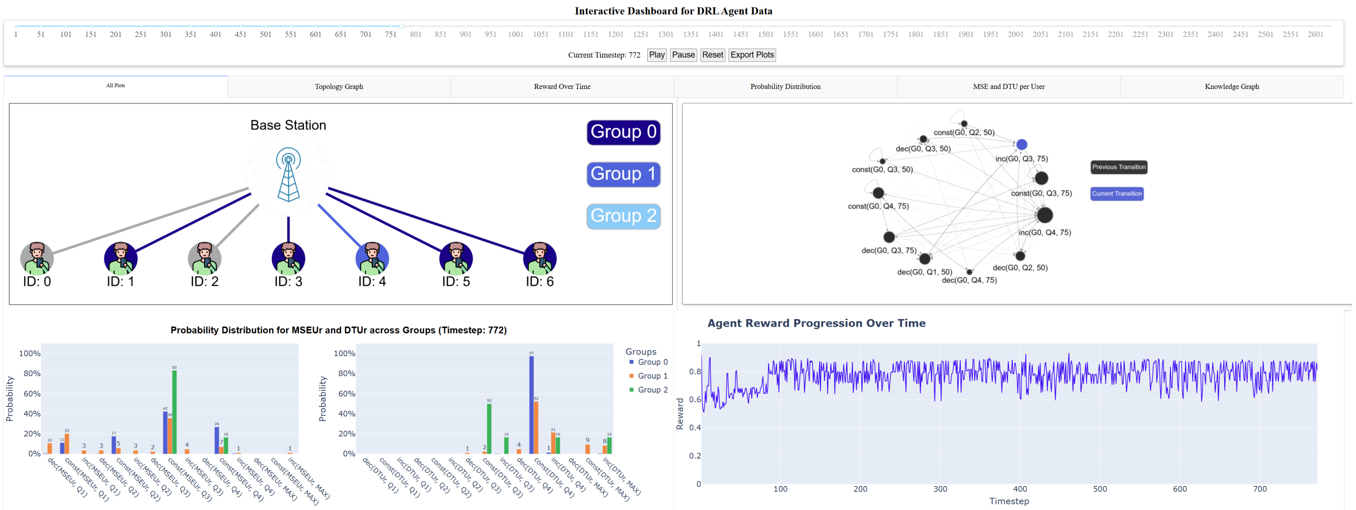

Fig. 2. SymbXRL's Interactive Dashboard Comprehensive Layout

and edges. Probability Distributions show KPIs impacts across user groups, and Performance Metrics tracks agent rewards for spectral efficiency and fairness. Through IAS (❻), we demonstrate high-level rule implementation (e.g., "Avoid scheduling users in Group 1") and reward maximization, with SymbXRL guiding the agent via constraint-based filtering and knowledge graph-based action selection. Notably, SymbXRL achieved a median 11.76% improvement in cumulative reward compared to the baseline DRL agent. This is significantly higher than the 0.07% gain achieved by METIS [4], a state-of-the-art XRL tool using decision trees for DRL agent explanations. Unlike SymbXRL, METIS uses its fitted decision trees for action steering, a method that is not scalable as it requires constant refitting when the environment or policy changes.

III. SymbXRL in Action

A. Technical Setup

Our demo runs on a laptop (Intel i7-13700H, 32GB RAM, NVIDIA RTX 4050) with Ubuntu 22.04.4 LTS on Windows Subsystem for Linux (WSL). SymbXRL operates either in the same or separate container as the DRL agent, maintaining real-time access to states, rewards, and actions. The interactive dashboard (❶) runs locally via a lightweight web server.

B. Dashboard: Flow and Control

The dashboard provides real-time insights into the DRL agent's operation, offering intuitive explanation and control. It guides users from a high-level network topology view to detailed agent decisions and performance metrics at each timestep. Upon launching, users see a comprehensive system overview: the Topology Graph (top left), Knowledge Graph (top right), Probability Distributions (bottom left), and Agent's Reward Progression (bottom right). A timeline allows timestep selection; users can play, pause, reset the simulation, or export plots. Separate tabs offer detailed views of all plots and Performance Metrics. The IAS is activated via a toggle switch, reflecting real-time performance changes.

C. Insights from SymbXRL

SymbXRL bridges the gap between theoretical DRL capabilities and operational requirements in 6G networks. Our demonstration showcases key insights through visualization: The probability distributions (Fig. 2, bottom left) reveal how the agent achieves fairness by maintaining high Data Transmission (DTU) across all groups (Q4 and MAX) despite varying channel conditions, evidenced by different MSE distributions. The knowledge graph (Fig. 2, top right) reveals the agent's decision patterns for Group 0, highlighting transitions between Q3 and Q4 states that maintain balanced performance while adapting to channel variations. These visualizations enable several key advantages: human-readable explanations, translating complex DRL decisions into intuitive symbolic representations; IAS (❻) allowing operators to inject high-level directives while maintaining performance; and real-time visualization of network metrics, enabling operators to observe how scheduling decisions affect system efficiency. These capabilities show that complex DRL systems can be both interpretable and controllable, paving the way for broader adoption in production networks.

Acknowledgment

This work is partially supported by bRAIN project PID2021-128250NB-I00 funded by MCIN/ AEI /10.13039/501100011033/ and the European Union ERDF "A way of making Europe"; by Spanish Ministry of Economic Affairs and Digital Transformation, European Union NextGeneration-EU/PRTR projects MAP-6G TSI-063000-2021-63, RISC-6G TSI-063000-2021-59 and AEON-ZERO TSI-063000-2021-52; C. Fiandrino is a Ramón y Cajal awardee (RYC2022-036375-I), funded by MCIU/AEI/10.13039/501100011033 and the ESF+.

References

[1] W. Saad, M. Bennis *et al.*, "A vision of 6G wireless systems: Applications, trends, technologies, and open research problems," *IEEE Netw.*, vol. 34, no. 3, pp. 134–142, 2020.

[2] A. Duttagupta, M. Jabbari *et al.*, "SymbXRL: symbolic explainable deep reinforcement learning for mobile networks," in *Proc. of IEEE INFOCOM*, 2025, Available online: https://github.com/RAINet-Lab/symbxrl.

[3] Q. An, S. Segarra *et al.*, "A deep reinforcement learning-based resource

scheduler for massive MIMO networks," *IEEE Trans. Mach. Learn. Commun. Netw.*, vol. 1, 2023.
[4] Z. Meng, M. Wang *et al.*, "Interpreting deep learning-based networking systems," in *Proc. of ACM SIGCOMM*, 2020, pp. 154–171.