Reconfigurable Intelligent Surface for High-Speed Railway MIMO Communication System: A Deep Reinforcement Learning Approach

Yannick Abel Talla Nana, Gang Liu, *Member, IEEE*, Bole Wilfried Tienin, *Student Member, IEEE*, and Zheng Ma, *Member, IEEE*

***Abstract*—This paper addresses the significant challenge of enhancing signal quality and maximizing system spectral efficiency (SE) in high-speed railway (HSR) tunnel lines, a critical issue in HSR communication systems. Multiple reconfigurable intelligent surfaces (RIS) are strategically placed on opposite sides of the tunnel to relay signals to the onboard mobile relay (MR) of the high-speed train (HST) as it traverses the tunnel. To achieve SE maximization, we formulate a joint beamforming at the base station (BS) and RISs phase shift optimization problem, which is inherently non-convex. This means the problem is complex and challenging to solve using traditional mathematical optimization techniques. To address this, we propose two novel deep reinforcement learning (DRL) methods: Proportional Batch Prioritization Replay Deep Deterministic Policy Gradient (PBPR-DDPG) and Rank-Base Batch Prioritization Replay Deep Deterministic Policy Gradient (RBPR-DDPG). These methods were chosen for their efficiency in managing complex, multi-variable optimization problems typical in communication systems. DRL, a type of machine learning, is used here to iteratively improve decision-making, while the deep deterministic policy gradient technique specifically helps in continuous action spaces, as is common in signal optimization. Our solutions are compared with an adapted version of the benchmark Uniform Random Batch Replay Deep Deterministic Policy Gradient (URBR-DDPG). Simulation results demonstrate that the placement of multiple RISs along the HSR-tunnel line leads to significant improvements in signal quality within the tunnel and maximizes the SE at the HST mobile relay (HST-MR), thus contributing a substantial advancement in HSR communication technologies.**

***Index Terms*—Beamforming, high-speed railway (HSR), reconfigurable intelligent surface, proportional batch prioritization replay deep deterministic policy gradient (PBPR-DDPG), rank-base batch prioritization replay deep deterministic policy gradient (RBPR-DDPG)).**

I. Introduction

TODAY high-speed railway transportation service has become one of the most important and fast-growing domains of mass transportation due to its fast speed, ability to carry several hundreds of passengers, and high safety standards. The demand for HSR transportation comes with high requirements, especially in the domain of wireless communications. There is a demand for high quality of experience (QoE), uninterrupted connectivity for on-board services such as Train Control and Management System (TCMS), CCTV streaming, seamless handover, edge computing, Wi-Fi, and the smart interconnection [1] of devices such as device-to-device (D2D) communication [2]. The requirements of 5G wireless communication for HSR [3], such as high data rate, low latency, seamless handover, and so on, need to be taken into consideration while designing a 5G HSR communication system. Moreover, high-speed trains (HST) produced nowadays are becoming faster with HST being able to attain record speeds of up to 500 km/h. This calls for the development of advanced technologies to handle high data transfer speeds of up to 150 Mbps as per the 5G communication standard [4]. These HSTs' high velocity comes with several challenges associated with high mobility communication. These include complex channel modeling due to fast-changing channel state information (CSI), Doppler frequency shift, frequent handover, short coherent time, persistent signal loss and lower system spectral efficiency in general [5].

As we look back in the past decades, millimeter wave (mmWave) has emerged as a game changer in addressing some of these challenges and fulfilling the stringent requirements of HSR communication since it comes with the benefits of offering multi-gigabyte-per-seconds of data speed and high signal directivity. However, mmWave also comes with challenges for its applications in such a high mobility communication environment such as fast fading, persistent shortening of system coherent time, change in signal delay time, frequent handover [6], signal blockage at the level of the HST cabin body [7], natural BS signal blockage in HSR tunnel scenario, leading to unstable and less reliable communication link [8].

To overcome the challenges associated with high-speed railway (HSR) mmWave communication, reconfigurable intelligent surfaces (RISs) have emerged as a promising solution due to their ability to passively reflect incident signals without requiring active radio frequency chains, while operating in full-duplex mode without introducing self-interference or thermal noise [9]. The deployment of multiple RISs along railway tunnels offers significant advantages over conventional approaches, including extended coverage in curved tunnel sec-

Part of this paper has been presented in IEEE VTC 2024 Fall. The work of Gang Liu was supported in part by the National Natural Science Foundation of China (NSFC) Project under grant 61971359, and in part by the Sichuan Science and Technology Program under Grant 2023ZHCGO010, Grant 2023YFH0012, and Grant 2023YFG0312. The work of Zheng Ma was supported by NSFC under Project 62271419 and Project U2268201. (Corresponding author: Gang Liu.)

Yannick Abel Talla Nana, Gang Liu, and Zheng Ma, are with the Provincial Key Laboratory of Information Coding and Transmission, Southwest Jiaotong University, Chengdu 611756, China (e-mail: ytallanana@my.swjtu.edu.cn; gangliu@swjtu.edu.cn; zma@home.swjtu.edu.cn).

Bole Wilfried Tienin, School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: tieninwilfried@std.uestc.edu.cn).

tions where direct line-of-sight is physically impossible. They also enhance communication reliability through path diversity when signal paths experience degradation, as demonstrated by [10] where RIS selection strategies provide flexibility without sacrificing performance benefits. Additionally, they reduce handover frequency through stable link maintenance in high-mobility scenarios. They also improve spectral efficiency despite the increased delay spread (from 0.7 μs with 2 RISs to 2.5 μs with 16 RISs) observed in [11] for Doppler effect mitigation in high-speed communications. From an implementation perspective, multiple RIS deployments present compelling economic benefits compared to traditional signal repeaters or distributed antenna systems, as RISs are passive, energy-efficient devices requiring minimal maintenance. This makes them particularly suited for challenging high-mobility environments. The cost-effectiveness and simple integration capabilities of RIS technology enable more extensive deployments in existing railway infrastructure with minimal modifications, providing a practical and efficient solution for high-quality wireless communication in HSR tunnel scenarios.

Also, finding the optimum beamforming at BS and phase shift matrix of the RISs for a mmWave base communication to optimize transmit-receiver signal is hard due to its multivariable nature, non-convexity, and complex mathematical model. This makes it difficult for their optimization problem to be solved with existing mathematical benchmark algorithms such as fractional programming (FP) [12] and weighted minimum mean-square error (WMMSE) [13] algorithm.

Uniform Random Batch Replay Deep Deterministic Policy Gradient (URBR-DDPG) algorithm, a special class of DRL has witnessed a tremendous interest in the wireless communication community for solving very complex problems. Recently, the authors in [14] introduced a long short-term memory DDPG (LSTM-DDPG) algorithm and URBR-DDPG as a benchmark to jointly optimize the continuous beamforming vector at the BS and the continuous phase shift matrix at the RIS in a single RIS-assist HSR communication scenario. Despite URBR-DDPG algorithm success, it still exhibits several limitations. It gives equal sampling probability to all the experiences, failing to prioritize critical experiences during the off-line training process, focusing on less relevant informative experiences leading to sub optimal policy learning. The algorithm's uniform sampling could cause an imbalance in experience replay by under representing significant but rare events, leading to high variance in updates and destabilizing the learning process. URBR-DDPG also neglects the importance of temporal difference errors, essential for policy improvement. Additionally, its inefficiency grows with large replay buffers due to high computational demands. In non-stationary environments, the inability to distinguish between current and outdated experiences further slows convergence and reduces learning efficacy. To address these issues, Proportional Prioritization Experience Replay deep Q-network (PPER-DQN) was introduced [15]. PPER-DQN was used to play the Atari game, allocating higher sampling probabilities to experiences with greater Temporal Difference (TD) errors δ. TD-errors δ indicates how surprising or unexpected a given experience transition is, i.e., how far the current experience value is from its next estimated experience [15], [16]. Despite its advancements, PPER-DQN's biased sampling towards high TD-error experiences can neglect valuable learning opportunities from less surprising transitions. This bias was mitigated with the development of Rank-based Prioritization Experience Replay (RPER-DQN) in [15], which ranks experiences by their TD errors, allowing for more balanced learning from diverse experiences. Both PPER-DQN and RPER-DQN have achieved state-of-the-art performance in complex scenarios like Atari games.

Inspired by all the above research and the state-of-the-art performance of both PPER-DQN and RPER-DQN in playing Atari games, we have proposed two novel prioritization experience replay (PER) DDPG algorithms that combines the power of PER during the off-line sampling replay process and that of DDPG network in handling continuous action space called PBPR-DDPG and RBPR-DDPG to find the solution to the formulated RIS-aided HSR tunnel communication system SE maximization problem. To the best of our knowledge, this is the first work that combines PER and DDPG to jointly optimize the beamforming at the BS and the phase shift of multiple RISs to maximize the spectral efficiency of a RIS-aided HSR tunnel communication system. The main contributions of this paper are as follows:

- We introduce a mmWave multiple RISs-aided MIMO HSR tunnel communication system as a solution to signal blockage that occurs when the HST passes through a tunnel. This is done by considering the time-varying channel of the HSR-scenario using mmWave as the transmission signal frequency to improve the spectral efficiency and reduce the persistent signal loss problem at the HST-MR when moving at high-speed.
- An optimization problem for downlink mmWave multiple RISs-aided MIMO HSR tunnel communication system is developed to jointly optimize the beamforming at the BS and the phase shift matrices of the RISs to maximize the system spectral efficiency. The generated SE maximization problem is then optimized using the modified benchmark URBR-DDPG algorithm.
- Taking into consideration the large amount of continuous data generated during the off-line training process, and the non-convex nature of the optimization problem, we then proposed two novel algorithms PBPR-DDPG and RBPR-DDPG to solve the formulated SE maximization problem.
- Numerous simulation results show that the introduction of multiple RISs in a tunnel HSR communication system leads to the reduction in signal blockage and an increase in the overall spectral efficiency at the HST-MR. Also, our proposed PBPR-DDPG and RBPR-DDPG solutions converge and effectively assist in finding the optimum beamforming at the BS and phase shift matrices for the RISs.

The rest of the paper is organized as follows: A brief review of the related works is presented in Section II. The multiple RIS-aided mmWave HSR tunnel communication system model (which will be called RIS-aided HSR for the rest of the paper)

and problem formulation are presented in Section III. In Section IV, the URBR-DDPG, PBPR-DDPG, and RBPR-DDPG frameworks, their algorithms and their complexity analysis are presented. The simulation results are presented in Section V to verify the performance of the proposed algorithms and the conclusion is presented in Section VI.

Notations: Bold capital letters and lowercase letters represent matrices and column vectors respectively. The notations $(.)^T$, $(.)^H$, and $(.)^{-1}$ represent the transpose, the conjugate transpose, and the inverse respectively. $|.|$ denotes the magnitude of a scalar. $\mathbb{E}$, $\mathbb{C}$, and $\mathbb{R}$ denote the statistical expectation, the set of complex and real numbers respectively. $\Re\{.\}$ and $\Im\{.\}$ represent the real and imaginary parts respectively.

II. Related work

A. RIS-aided mmWave HSR communication

Recent research has explored various RIS applications in HSR communications. The authors in [17] investigated a RIS-assisted HST communication system, deriving closed-form expressions for coverage probability and optimizing RIS placement and phase design to maximize travel distance under coverage constraints. The authors in [18] introduced RIS-aided integrated sensing and communication for mmWave HSR to optimize system rate through joint beamforming and discrete phase shift optimization. The authors in [19] placed RIS on HST windows to reduce penetration loss and provide quasi-static channels between receivers and RIS. Research in [20] developed channel models incorporating Doppler effects and mmWave properties, then designed spectral efficiency optimization for both active and passive beamforming using Riemann gradient descent. Authors in [21] proposed using RIS to dynamically modify signal propagation profiles to counter time-varying channel effects and reduce outage probability, while [22] demonstrated how strategic RIS placement along HSR lines suppresses interference and jamming.

However, existing approaches predominantly employ single-RIS configurations that do not address the severe signal blockage in tunnel environments. Prior works utilizing traditional optimization methods struggle with the non-convex joint beamforming and phase shift optimization in dynamic HSR scenarios. Our approach deploys multiple RISs along tunnel walls, creating reliable signal paths throughout the tunnel while addressing its unique propagation characteristics. We formulate a comprehensive joint optimization problem for BS beamforming and multiple RIS phase shift matrices to maximize spectral efficiency.

B. Multiple RISs-Aided Communication Systems

Multiple RISs deployment has attracted significant research attention for overcoming severe propagation challenges. Research in [23] demonstrated rate enhancements with double-RIS systems compared to single-RIS deployments when direct links are blocked. Studies in [24] established that networks selecting RISs with highest instantaneous SNR achieve diversity order of LN (where L is the number of RISs and N represents reflecting elements per RIS). The research work in [25] showed improved performance with increasing RIS elements in domino-pattern systems for severe obscuration scenarios under Nakagami-m fading channels. The authors in [26] developed frameworks for joint estimation of timing offsets and channel parameters in distributed multiple-RIS communications. The research work in [27] investigated the use of heterogeneous graph neural networks for cooperative beamforming in multi-RIS-aided mmWave MIMO systems, achieving better performance than traditional neural networks with 30% QoS improvement. Prior work in [28] presented simulation frameworks for RIS-based mmWave vehicular communications, proposing algorithms for multi-user service delivery. Research in [29] introduced distributed sum-rate maximization approaches for multi-RIS-empowered multiple access, while [30] established that RISs deployed closer to the base station outperform random deployments in mmWave MIMO systems, though hardware impairments ultimately limit performance.

While these works establish the theoretical foundations of multiple RIS deployments, they primarily focus on generic wireless scenarios rather than the unique challenges of HSR tunnel environments. Unlike previous research, our work specifically addresses signal blockage in HSR tunnels through strategic RIS placement along tunnel walls. The existing literature lacks comprehensive solutions for the combined challenges of high mobility, mmWave propagation, and tunnel signal blockage.

C. DRL based wireless communication

Deep reinforcement learning has been effective in solving complex, non-convex optimization problems in wireless communications. Research in [31] utilized DDPG to optimize beamforming and RIS phase shifts in multi-user MISO systems, achieving significant sum-rate improvements. In [32] the authors employed distributed DRL to coordinate beamforming in dynamic MISO systems for rate maximization. Authors in [33] applied DRL for interference suppression in HSR communication systems, while research in [34] introduced multi-agent DRL for power allocation in mmWave HSR systems to maximize sum rate. In [35] the authors applied DDPG to jointly optimize beamforming and RIS phase shifts for mmWave V2I communications. Recent studies have begun exploring prioritized experience replay in wireless contexts. In [36] the authors introduced state-aware Prioritization Experience Replay (PER) for handover decisions in 5G ultra-dense networks. Furthermore, in [37] the authors used PER to enhance learning efficiency for secure communications, and the authors in [38] introduced freshness discount factors to PER for improved learning rates.

Despite these advancements, existing DRL approaches for RIS optimization predominantly employ uniform random sampling during replay processes, which fails to prioritize learning from more informative experiences and results in slower convergence and suboptimal solutions. The current literature lacks effective mechanisms to handle the vast amount of continuous data generated during RIS optimization, particularly in high-mobility scenarios like HSR communications where channel conditions change rapidly. Our work addresses this critical gap by introducing two novel prioritization experience replay algorithms PBPR-DDPG and RBPR-DDPG that significantly enhance the learning efficiency and solution quality for the

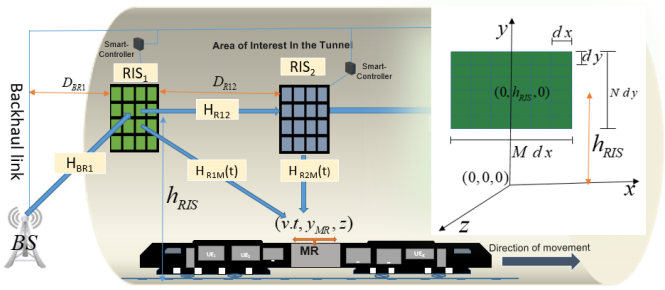

Fig. 1. RIS-aided HSR communication scenario.

complex joint optimization of beamforming and multiple RIS phase shifts. To the best of our knowledge, this is the first work that combines prioritized experience replay techniques with DDPG specifically for RIS-aided HSR tunnel communication optimization.

III. System Model

We introduce an RIS-aided HSR downlink MIMO tunnel communication system featuring a local base station with $\mathrm{N_{BS}}$ isotropic antennas and a high-speed train mobile relay (HST-MR) equipped with $\mathrm{N_{MR}}$ antennas placed on top of the train cabin at a height of y_{MR} from the ground to reduce penetration loss. The BS and HST-MR antennas are assumed to be uniform linear array (ULA), with HST moving at velocity of v m/s in a positive x-direction of the Cartesian plane. Multiple RISs are strategically placed at regular intervals along the railway line, ensuring consistent signal relay from one RIS to the next, and enhanced communication quality throughout the extended tunnel length. In this paper, we have only considered two RISs in the area named "area of interest in the tunnel" as shown in Fig. 1 for simplicity[1]. We have to precise here that the choice to not include the direct signal between the BS and the HST-MR is on the assumption that, mmWave high directivity will be difficult to have an appreciable BS-HST-MR channel when the HST is in the tunnel. $\mathrm{RIS_1}$ is placed close to the entrance of tunnel at a distance of D_{BR1} from the BS ensuring a line-of-sight (LOS) between $\mathrm{RIS_1}$ and BS. $\mathrm{RIS_2}$ is then placed on the side opposite to $\mathrm{RIS_1}$ inside the tunnel at a LOS distance of D_{R12} ensuring a consistent signal relay from $\mathrm{RIS_1}$ to the $\mathrm{RIS_2}$ as the HST travels through the tunnel as shown in Fig. 1. $\mathrm{RIS_1}$ and $\mathrm{RIS_2}$, comprise of $\mathrm{N_1M_1}$ and $\mathrm{N_2M_2}$ reflecting elements, respectively, arranged in uniform planar arrays (UPA) as depicted in Fig. 1. The RISs are positioned in the three-dimensional (3D) Cartesian coordinate system with their geometric centers at $(0, h_{\mathrm{RIS}}, 0)$, where h_{RIS} represents the height of the RISs from the ground inside the tunnel. The RISs' origin will depend on whether $\mathrm{N_1M_1}$ is even or odd for $\mathrm{RIS_1}$, similarly for $\mathrm{RIS_2}$. If $\mathrm{N_1M_1}$ are even, then the center of $\mathrm{RIS_1}$ will be between the two center elements. If $\mathrm{N_1M_1}$ are odd, then the center of $\mathrm{RIS_1}$ will be the middle element at the origin. A similar definition applies to $\mathrm{RIS_2}$. The coordinates of the RISs elements are determined using equations (1) and (2) in [39].

[1]Future research will focus on determining the optimal number and layout of RISs along the entire tunnel length, considering the trade-offs between deployment cost, coverage area, and signal quality enhancement. This will include comprehensive studies on RIS density requirements for various tunnel geometries and HST operating conditions.

A. mmWave channel model for RIS-aided HSR

The channel gains in this research are modeled as mmWave channel following the Saleh-Valenzuela channel model in [40]. The signal from BS takes two paths as it enters the tunnel. The first path of the signal is BS $\rightarrow$ $\mathrm{RIS_1}$ $\rightarrow$ MR, representing the signal path from the BS to $\mathrm{RIS_1}$ and from $\mathrm{RIS_1}$ to HST-MR. The second path is BS $\rightarrow$ $\mathrm{RIS_1}$ $\rightarrow$ $\mathrm{RIS_2}$ $\rightarrow$ MR, representing the signal from BS to $\mathrm{RIS_1}$, from $\mathrm{RIS_1}$ to $\mathrm{RIS_2}$ and from $\mathrm{RIS_2}$ to HST-MR. Considering orthogonal frequency-division multiple access (OFDMA) transmission, with $x(t) \in \mathbb{C}^{S\times 1}$ representing the vector of data symbols being transmitted at time slot t from the BS. Each element of $x(t)$ corresponds to a different subcarrier in the OFDMA transmission, and S is the number of subcarriers (assuming one stream per subcarrier). The received signal $y(t) \in \mathbb{C}^{\mathrm{N_{MR}}\times 1}$ at the HST-MR is given by

$$y(t) = \Big[\Big(\mathbf{H_{R1M}}(\mathbf{t})\mathbf{\Theta_1}\mathbf{H_{BR1}} + \mathbf{H_{R2M}}(\mathbf{t})\mathbf{\Theta_2}\mathbf{H_{R12}}\mathbf{\Theta_1}\mathbf{H_{BR1}}\Big)\mathbf{W_{BS}}x(t)\Big] + \mathbf{n(t)}, \tag{1}$$

where $\mathbf{W_{BS}} \in \mathbb{C}^{(\mathrm{N_{BS}}\times S)}$ is the transmit beamforming matrix at the BS, $\mathbf{\Theta_1}$ and $\mathbf{\Theta_2}$ represent the phase shift matrices at $\mathrm{RIS_1}$ and $\mathrm{RIS_2}$, given by

$$\mathbf{\Theta}_1 = \mathrm{diag}\left(e^{j\phi_{1,1}}, e^{j\phi_{1,2}}, \ldots, e^{j\phi_{1,\mathrm{N_1M_1}}}\right) \in \mathbb{C}^{\mathrm{N_1M_1}\times \mathrm{N_1M_1}},$$
$$\mathbf{\Theta}_2 = \mathrm{diag}\left(e^{j\phi_{2,1}}, e^{j\phi_{2,2}}, \ldots, e^{j\phi_{2,\mathrm{N_2M_2}}}\right) \in \mathbb{C}^{\mathrm{N_2M_2}\times \mathrm{N_2M_2}}, \tag{2}$$

where diag(.) denotes the diagonal matrix, $\phi_{1,\mathrm{N_1M_1}} \in [0, 2\pi)$ and $\phi_{2,\mathrm{N_2M_2}} \in [0, 2\pi)$ respectively represent phase shifts for the $\mathrm{N_1M_1}$-th element and $\mathrm{N_2M_2}$-th element of $\mathrm{RIS_1}$ and $\mathrm{RIS_2}$. $\mathbf{n(t)} \sim \mathcal{CN}(0, \sigma^2\mathrm{I_{MR}}) \in \mathbb{C}^{\mathrm{N_{MR}}\times 1}$ is the addictive white Gaussian noise (AWGN) vector on the signal received at HST-MR, with zero mean and variance σ^2. The channel matrix gain from BS to $\mathrm{RIS_1}$, $\mathrm{RIS_1}$ to $\mathrm{RIS_2}$ and $\mathrm{RIS_1}$ to MR are respectively given in (3), (4) and (5) below;

$$\mathbf{H_{BR1}} = \xi_{\mathrm{BR1}} \sum_{l=1}^{L_{\mathrm{BR1}}} h_{\mathrm{BR1},l}\mathbf{a}_P(\theta_{\mathrm{R1,AOA}}, \vartheta_{\mathrm{R1,AOA}})^H \times \mathbf{a}_L(\theta_{\mathrm{BS,AOD}}) \in \mathbb{C}^{(\mathrm{N_1M_1})\times \mathrm{N_{BS}}}, \tag{3}$$

$$\mathbf{H_{R12}} = \xi_{\mathrm{R12}} \sum_{l=1}^{L_{\mathrm{R12}}} h_{\mathrm{R12},l}\mathbf{a}_P(\theta_{\mathrm{R2,AOA}}, \vartheta_{\mathrm{R2,AOA}})^H \times \mathbf{a}_P(\theta_{\mathrm{R1,AOD}}, \vartheta_{\mathrm{R1,AOD}}) \in \mathbb{C}^{(\mathrm{N_2M_2})\times(\mathrm{N_1M_1})}, \tag{4}$$

$$\mathbf{H_{R1M}}(t) = \xi_{\mathrm{R1M}} \sum_{l=1}^{L_{\mathrm{R1M}}} h_{\mathrm{R1M},l}\mathbf{a}_L(\theta_{\mathrm{MR,AOA}})^H \times \mathbf{a}_P(\theta_{\mathrm{R1,AOD}}, \vartheta_{\mathrm{R1,AOD}}) \times e^{j2\pi f_{\mathrm{RM}}T_s} \in \mathbb{C}^{\mathrm{N_{MR}}\times(\mathrm{N_1M_1})}, \tag{5}$$

where $\mathbf{a}_L(\theta_{\text{BS,AOD}})$ and $\mathbf{a}_L(\theta_{\text{MR,AOA}})$ are the steering vectors of the BS and MR antenna respectively, and are given by equations (6) and (7) below

$$\mathbf{a}_L(\theta_{\text{BS,AOD}}) = \frac{1}{\sqrt{\text{N}_{\text{BS}}}}\Big[1, e^{j\frac{2\pi}{\lambda_c}d_s\sin(\theta_{\text{BS,AOD}})}, \ldots, e^{j(\text{N}_{\text{BS}}-1)\frac{2\pi}{\lambda_c}d_s\sin(\theta_{\text{BS,AOD}})}\Big]^T, \tag{6}$$

$$\mathbf{a}_L(\theta_{\text{MR,AOA}}) = \frac{1}{\sqrt{\text{N}_{\text{MR}}}}\Big[1, e^{j\frac{2\pi}{\lambda_c}d_s\sin(\theta_{\text{MR,AOA}})}, \ldots, e^{j(\text{N}_{\text{MR}}-1)\frac{2\pi}{\lambda_c}d_s\sin(\theta_{\text{MR,AOA}})}\Big]^T, \tag{7}$$

where $d_s = \frac{\lambda_c}{2}$ represents the separation between antenna elements. We used the symbols θ and ϑ to represent the azimuth and elevation angles respectively. The notations AOD and AOA denote angle of departure and angle of arrival respectively, and these notations will be used as such throughout this paper. The channel matrix gain $\mathbf{H_{R2M}}(t) \in \mathbb{C}^{\text{N}_{\text{MR}}\times(\text{N}_2\text{M}_2)}$ from RIS_2 to MR is obtained as in (5).

For UPA elements, assume that there are M_x elements on the x-axis and N_y elements on the y-axis of each RIS. We defined as in [40] the steering vector of RIS_1 $\mathbf{a}_P(\theta_{\text{R1,AOA}}, \vartheta_{\text{R1,AOA}})$ using equations (8) and (9) below

$$\mathbf{a}_P(\theta_{\text{R1,AOA}}, \vartheta_{\text{R1,AOA}}) = \mathbf{a}_{\text{M}_x}(p) \otimes \mathbf{a}_{\text{N}_y}(q) \in \mathbb{C}^{\text{M}_x\text{N}_y\times 1}, \tag{8}$$

where $\otimes$ is the Kronecker product, $p \triangleq \frac{2\pi d_e\cos(\vartheta_{\text{R1,AOA}})}{\lambda_c}$, $q \triangleq \frac{2\pi d_e\sin(\vartheta_{\text{R1,AOA}})\cos(\theta_{\text{R1,AOA}})}{\lambda_c}$, with $d_e = \frac{\lambda_c}{2}$ representing the RIS elements spacing and λ_c representing the carrier signal wavelength. $\mathbf{a}_{\text{M}_x}(p)$ and $\mathbf{a}_{\text{N}_y}(q)$ will be given as below

$$\mathbf{a}_{\text{M}_x}(p) \triangleq \frac{1}{\sqrt{\text{M}_x}}[1,\ e^{jp}, \ldots,\ e^{j(\text{M}_x-1)p}]^T,$$
$$\mathbf{a}_{\text{N}_y}(q) \triangleq \frac{1}{\sqrt{\text{N}_y}}[1,\ e^{jq}, \ldots,\ e^{j(\text{N}_y-1)q}]^T, \tag{9}$$

where $\mathbf{a}_{\text{M}_x}(p) \in \mathbb{C}^{\text{M}_x\times 1}$ is a vector that represents the phase shift pattern for elements along the x-axis. The i-th element of $\mathbf{a}_{\text{M}_x}(p)$ is given by $e^{j(i-1)p}$, where i ranges from 1 to M_x. $\frac{1}{\sqrt{\text{M}_x}}$ is the normalization factor that ensures the power remains constant regardless of the RIS elements. A similar definition applies to $\mathbf{a}_{\text{N}_y}(q)$. The UPA steering vectors $\mathbf{a}_P(\theta_{\text{R1,AOD}}, \vartheta_{\text{R1,AOD}})$, $\mathbf{a}_P(\theta_{\text{R2,AOA}}, \vartheta_{\text{R2,AOA}})$, and $\mathbf{a}_P(\theta_{\text{R2,AOD}}, \vartheta_{\text{R2,AOD}})$ are obtained similarly as in (8) and (9).

ξ_{BR1}, ξ_{R12}, ξ_{R1M} and ξ_{R2M} denote the normalized factor of their respective channel gain. $\xi_{\text{BR1}} = \sqrt{\frac{\text{N}_{\text{BS}}\times\text{N}_1\text{M}_1}{L_{\text{BR1}}}}$ and ξ_{R1M}, ξ_{R12} and ξ_{R2M} can be obtained in a similar manner. L_{BR1}, L_{R12}, L_{R1M} and L_{R2M} respectively represents the number of signal paths from BS to RIS_1, from RIS_1 to RIS_2, from RIS_1 to HST-MR and from RIS_2 to HST-MR . $h_{\text{BR1},l}$, $h_{\text{R12},l}$, $h_{\text{R1M},l}$ and $h_{\text{R2M},l}$ represent the time-varying gains of the channels which are the realizations of the large-scale fading for the l^{th} signal path for their respective channel matrix gain. In this paper we will use the time-varying channel gain which follows a complex Gaussian distribution similar to that used in [14], $\mathbb{CN}(0, 10^{-0.1PL(dB)})$, with $PL(dB)$ representing the path loss component of the channel given by $PL_0 + 10\alpha\log(D) + PL_s$, where D represents the distance between two components (signal source and destination). PL_0 is the path loss at a reference distance of 1 m, α represents the path loss factor, with $PL_s \sim \mathbb{CN}(0, \sigma_s^2)$ representing the shadow fading which follows a complex Gaussian distribution with zero mean and variance σ_s^2. T_s is the system sampling period. f_{RM} is the Doppler shift observed on the channel gain due to the train motion and is given by the equation below

$$f_{RM} = \frac{v}{\lambda_c}\cos(\theta_{MR,AOA}), \tag{10}$$

The effect of HST train mobility in computing the channel matrix gain $\mathbf{H_{R1M}}(\mathbf{t})$ and $\mathbf{H_{R2M}}(\mathbf{t})$ must be taken into consideration in an RIS-aided HSR system. This is because as the train moves with a high velocity v, it causes a variation of the delay spread of the signal, transmission delay, and the system processing delay. So considering the outdated CSI during the training process can lead to performance loss [14]. More so, the moment the phase shift matrices used to generate a CSI at time t is returned to the agent in this case the Basestation by a backhaul network, these CSI are already outdated. To obtain the real CSI from the outdated CSI, we used the relation in [33] given below

$$\mathbf{H_{R1M}}(t+\tau_s) = \rho_d\tilde{\mathbf{H}}_{\mathbf{R1M}}(t) + \sqrt{(1-\rho_d^2)}\Delta\mathbf{H_{R1M}}(t+\tau_s), \tag{11}$$

where ρ_d is the temporal correlation coefficient according to Jakes's channel model used to link the real CSI to the outdated CSI [41] and is given by the equation below

$$\rho_d = J_0(2\pi f_{max}\tau_s), \tag{12}$$

where J_0 represents the zeroth-order Bessel function of the first kind. $f_{max} = \frac{v}{c}f_c$, represents the system maximum Doppler shift for a HST traveling at a velocity of v m/s. f_c represents the system carrier frequency and c represents speed of light. $\Delta\mathbf{H_{R1M}}(t+\tau_s)$ is the residual error associated to channel matrix gain $\mathbf{H_{R1M}}(t)$ and $\mathbf{H_{R1M}}(t+\tau_s)$, and is independently identically distributed. τ_s represents the delay between the outdated CSI and the system real CSI. The real CSI of $\mathbf{H_{R2M}}(t+\tau_s)$ is obtained in similar manner using (11) and (12).

B. Problem Formulation

Since both the BS and HST-MR forms a MIMO communication system, all the CSI, beamforming, and RISs phase shift matrices are assumed to be available at the BS. The real CSI are used in this work to compute the system spectral efficiency (SE). The RIS-aided HSR SE is computed as in [42] as shown below from the $\mathbf{H_{comb}}(\mathbf{t}+\tau_\mathbf{s})$

$$\mathbf{H_{comb}}(\mathbf{t}+\tau_\mathbf{s}) = [\mathbf{H_{R1M}}(\mathbf{t}+\tau_\mathbf{s})\mathbf{\Theta_1}\mathbf{H_{BR1}} + \mathbf{H_{R2M}}(\mathbf{t}+\tau_\mathbf{s})\mathbf{\Theta_2}\mathbf{H_{R12}}\mathbf{\Theta_1}\mathbf{H_{BR1}}]\,\mathbf{W_{BS}}, \tag{13}$$

$$SE(t) = \log_2\left[1 + \text{tr}\left(\left(\frac{P_{BS}}{\sigma^2 N_b}\right)(\mathbf{H_{comb}}(\mathbf{t}+\tau_\mathbf{s}))^H\mathbf{H_{comb}}(\mathbf{t}+\tau_\mathbf{s})\right)\right] \tag{14}$$

where $P_{BS} = \text{tr}\{\mathbf{W_{BS}}\mathbf{W_{BS}^H}\}$ represents the BS transmit power, σ^2 represents the AWGN variance, N_b represents the

rank of the combined channel gain matrix $\mathbf{H_{comb}(t+\tau_s)}$. The system SE maximization problem is formulated as

$$\begin{aligned} \max_{\mathbf{W_{BS}},\mathbf{\Theta_1},\mathbf{\Theta_2}} \quad & \frac{1}{T}\sum_{t=1}^{T} \mathrm{SE}(t) \\ \text{s.t.} \quad & C_1 : \mathrm{tr}\{\mathbf{W_{BS}}\mathbf{W_{BS}^H}\} \leq P_{\mathrm{BSmax}} \\ & C_2 : |\mathbf{\Theta}_{\mathrm{N}_r\mathrm{M}_r}| = 1, \quad \forall r \in \{1,2\}, \end{aligned} \tag{15}$$

where P_{BSmax} is the maximum transmit power. $\mathbf{\Theta}_{\mathrm{N}_r\mathrm{M}_r}$ represents the $\mathrm{N}_r\mathrm{M}_r$-th diagonal elements of either $\mathbf{\Theta_1}$ or $\mathbf{\Theta_2}$. T refers to the total number of times the $\mathbf{W_{BS}}$, $\mathbf{\Theta_1}$ and $\mathbf{\Theta_2}$ are varied when HST-MR travel through the tunnel for the simulation reference distance (reference distance = 300 m for this research work). The constraint C_1 defines the BS maximum transmit power level and the constraint C_2 defines the limits of RISs elements[2].

IV. URBR-DDPG, PBPR-DDPG AND RBPR-DDPG FRAMEWORK, ALGORITHM AND COMPLEXITY ANALYSIS

A. DDPG Overview

In this section, a brief overview of the main components of DDPG learning related to our research work is presented. This will be done by a description of the state space, action space, rewards, and experience.

State Space: The state $s^t \in \mathcal{S}$ represents the set of observations that the agent collects from the RIS-aided HSR environment at every time step t. The transmit power P_{BSmax} at the BS and received power P_{MR} at the HST-MR forms the first states of our RIS-aided HSR and are real value numbers. This first states collection allows the DDPG agent to understand the power dynamics and adjust its strategy accordingly to optimize the system's performance. The dynamic change of position $|d_M(t)| = (v.t, y_{MR}, z_{MR})$ of the HST-MR for every time step t is also considered as a state of the environment and will form the second state. This coordinate of the position of the HST-MR $d_M(t)$ impacts the wireless channel properties due to Doppler shifts and varying path losses. The channel matrices $\mathbf{H_{BR1}}$, $\mathbf{H_{R12}}$, $\mathbf{H_{R1M}}(t)$ and $\mathbf{H_{R2M}}(t)$, $(t = (t+\tau_s))$, characterize the communication links between the BS, RISs, and HST-MR, which are essential for the agent to evaluate the current state of the wireless channel. The different channel gain matrices are represented as complex variables, but since the neural network does not accept complex value numbers, the channel gains are separated into their real and imaginary part before being input to the neural network. $\mathbf{H_{BR1}}$ as $\Re\{\mathbf{H_{BR1}}\}$ and $\Im\{\mathbf{H_{BR1}}\}$, $\mathbf{H_{R12}}$ as $\Re\{\mathbf{H_{R12}}\}$ and $\Im\{\mathbf{H_{R12}}\}$, $\mathbf{H_{R1M}}(t)$ as $\Re\{\mathbf{H_{R1M}(t)}\}$ and $\Im\{\mathbf{H_{R1M}(t)}\}$ and $\mathbf{H_{R2M}}(t)$ as $\Re\{\mathbf{H_{R2M}(t)}\}$ and $\Im\{\mathbf{H_{R2M}(t)}\}$. The actions at time step $(t-1)$ are included as states to the total state space.

Action Space: The action represents a set of continuous beamforming at the BS and continuous phase shift matrices at RIS$_1$ and RIS$_2$. The set of actions $a^t \in \mathcal{A} \in \{\mathbf{W_{BS}}, \mathbf{\Theta_1}, \mathbf{\Theta_2}\}$ which when applied to the RIS-aided HSR environment changes their state from s^t to s^{t+1}. These actions are complex and needs to be separated into their real and imaginary part before being used as inputs to the neural network. The beamforming at the BS is represented as $\Re\{\mathbf{W_{BS}}\}$ and $\Im\{\mathbf{W_{BS}}\}$, phase shift matrix of RIS$_1$ is represented as $\Re\{\mathbf{\Theta_1}\}$ and $\Im\{\mathbf{\Theta_1}\}$ and phase shift matrix of RIS$_2$ is represented as $\Re\{\mathbf{\Theta_2}\}$ and $\Im\{\mathbf{\Theta_2}\}$. We have to precise here that our aim is to find the optimum set of actions that will maximize our system spectral efficiency.

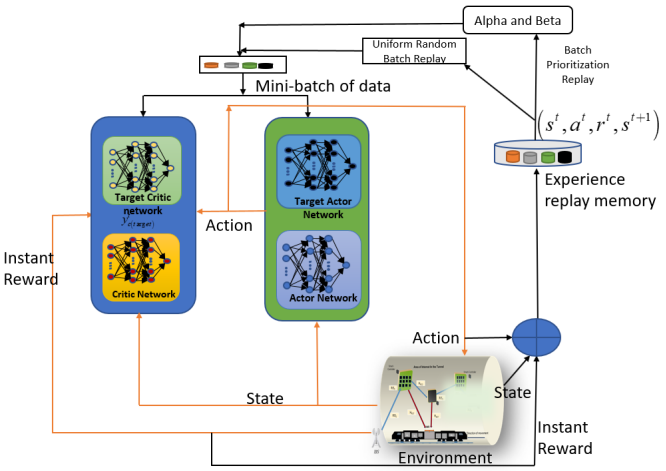

Fig. 2. DDPG Operation flow steps for RIS-aided HSR.

Reward: The reward r^t at time step t of our RIS-aided HSR represents the immediate return obtained by the agent after applying action a^t to the environment and causing it to change its current state s^t to s^{t+1}. The reward function associated to our spectral efficiency equation is as shown in below

$$r^t = \frac{1}{T}\sum_{t=1}^{T} \mathrm{SE}(t). \tag{16}$$

Experience: Experience $e^t \in \{s^t, a^t, r^t, s^{t+1}\}$ represents the set RIS-aided HSR environment parameters that are obtained every time an action a^t is applied to the environment causing it to change from its current state s^t to a future state s^{t+1} generating a reward r^t to help the agent in its learning process.

B. DDPG Description

The DDPG network in this study, depicted in Fig. 2, is an enhanced version of the model in [31]. It consists of an actor-network, which takes in as inputs the environmental state s^t and outputs actions to maximize rewards, and a critic-network that evaluates these actions' quality. Both networks have one input layer, two hidden layers, and one output layer. The actor's input layer matches the state space $\mathcal{S}$ dimensions, and its output layer aligns with the action space $\mathcal{A}$ dimensions. The critic's input layer combines both action and state dimensions $(\mathcal{A}+\mathcal{S})$, and its output layer has a single neuron indicating the Q-value for a state-action pair.

[2]Future research will extend this spectral efficiency optimization framework to incorporate service-specific quality requirements. This will involve developing adaptive resource allocation strategies that dynamically balance spectral efficiency with the varying latency, bandwidth, and reliability demands of different service types (voice, video, data) in high-speed scenarios, potentially using multi-objective optimization techniques to ensure stable connectivity for critical HSR applications while maximizing overall system performance.

In both the actor and critic networks of this DDPG model, the Rectified Linear Unit (ReLU) is used as the activation function for the two hidden layers, introducing non-linearity and enabling the learning of complex policies. The actor network's output layer uses a $tanh$ function to handle both positive and negative action values, while the critic network's scalar output does not require an activation function. Both networks utilize the Adam optimizer for adaptive optimization, with specific decaying rates μ_c for the critic and μ_a for the actor. Weights and biases are initialized using the Xavier random initializer to prevent vanishing and exploding gradients, ensuring balanced neuron output variance across layers for more stable and efficient training. This initialization is particularly effective with the $tanh$ function, enhancing network performance and minimizing the risk of suboptimal learning outcomes.

In the DDPG model, the actor-network's hidden layers are batch normalized to stabilize the continuous actions before input to the critic-network, while the critic-network only batch normalizes its first hidden layer to stabilize learning and reduce computational complexity. State whitening is employed to ensure all state variables equally contribute to learning, preventing domination by features with larger numerical values. Additionally, deep copies of both networks, known as target actor and target critic networks, are created at the start of training. These target networks, update their parameters more slowly, and serve to mitigate instability caused by the continual parameter updates in the actor and critic networks, thus maintaining a stable learning environment.

C. URBR-DDPG Training Process

In the uniform random batch replay training depicted in Fig. 2, the current state s^t is input into the actor network to generate an action a^t with added Ornstein-Uhlenbeck noise to improve the agent exploration. Ornstein-Uhlenbeck noise is used here due to its ability to generate smoother, more natural sequences of actions compared to random noise. The critic network evaluates the the actor network action and the environment state, outputting a Q-value for the expected cumulative future rewards, and is updated using the Bellman equation in [31]. The target actor network predicts the next action a^{t+1} from the subsequent state s^{t+1}, believed to be optimal under the current policy. This action and state are then evaluated by the target critic network to estimate the future reward Q-value. The generated experience $e^t \in \{s^t, a^t, r^t, s^{t+1}\}$ during this training process will then be used during the off-line learning process to enhance the decision-making of both networks, aiming to maximize the agent's reward.

Experiences of batch size M_b are then uniformly randomly sampled from uniform replay buffer memory χ_U. The target Q-value for the j^{th} experience in the sampled batch is then computed via the equation below

$$y_{j,c'} = r_j + \gamma Q_{c'}\left(s_j^{t+1}, \pi_{a'}(s_j|\theta_{a'}^{\pi})\right), \tag{17}$$

where r_j is the immediate reward of the j^{th} experience, s_j^{t+1} is the next state of the j^{th} experience, $\pi_{a'}(s_j|\theta_{a'}^{\pi})$ is the target actor's network policy, $Q_{c'}\left(s_j^{t+1}, \pi_{a'}(s_j|\theta_{a'}^{\pi})\right)$ is the target critic-network estimated Q-value of the state-action pair and $\theta_{a'}^{\pi}$ is the target actor network parameter. γ refers to the discount factor and has a value range $[0, 1)$. γ is used to determine the importance given to future reward over immediate rewards during the training process. Considering the main aim of the agent is to minimize the critic loss function which is the difference between the target Q-value estimated by the target critic-network and the predicted Q-value by the critic-network state-action pair. The critic loss is given by the equation below

$$L(\theta_c^Q) = \frac{1}{M_b}\sum_{j=1}^{M_b}\left(y_{j,c'} - Q_c\left(s_j^t, a_j^t|\theta_c^Q\right)\right)^2, \tag{18}$$

where $Q_c(s_j, a_j|\theta_c^Q)$ is the predicted Q-value of the critic-network from current environment state s_j^t and the current action a_j^t from the actor network. θ_c^Q is the critic-network parameter. $y_{j,c'}$ is the target Q-value of the j^{th} experience obtained in (17) above.

The actor network loss $L(\theta_a^{\pi})$ and the policy gradient $(\nabla_{\theta_a^{\pi}} J)$ of actor network are then computed with respect to the equations below

$$L(\theta_a^{\pi}) = -\frac{1}{M_b}\sum_{j=1}^{M_b} Q_c\left(s_j^t, \pi_a\left(s_j^t|\theta_a^{\pi}\right)|\theta_c^Q\right), \tag{19}$$

$$\begin{aligned}\nabla_{\theta_a^{\pi}} J = \frac{1}{M_b}\Bigg[\sum_{j=1}^{M_b}\nabla_a Q_c\left(s, a|\theta_c^Q\right)\Big|_{s=s_j^t, a=\pi_a(s_j^t)} \\ \times \nabla_{\theta_a^{\pi}}\pi_a\left(s|\theta_a^{\pi}\right)\Big|_{s=s_j^t}\Bigg].\end{aligned} \tag{20}$$

The critic and the actor-network parameters can now be updated respectively according to the equations below

$$\theta_c^{Q(t+1)} = \theta_c^{Q(t)} + \lambda_c^Q \nabla_{\theta_c^Q} L(\theta_c^Q), \tag{21}$$

$$\theta_a^{\pi(t+1)} = \theta_a^{\pi(t)} - \lambda_a^{\pi} \nabla_{\theta_a^{\pi}} J, \tag{22}$$

where λ_c^Q and λ_a^{π} are respectively the learning rates of the critic and actor networks. The target critic and target actor network parameters are respectively soft updated during the learning process according to the equations below:

$$\theta_{c'}^{Q(t+1)} = \tau_{c'}^Q \theta_{c'}^{Q(t)} + (1 - \tau_{c'}^Q)\theta_{c'}^{Q(t)}, \tag{23}$$

$$\theta_{a'}^{\pi(t+1)} = \tau_{a'}^{\pi} \theta_{a'}^{\pi(t)} + (1 - \tau_{a'}^{\pi})\theta_{a'}^{\pi(t)}, \tag{24}$$

where $\tau_{c'}^Q$ and $\tau_{a'}^{\pi}$ are respectively learning rates of the target critic and target actor network parameters.

D. PBPR-DDPG Training Process

The URBR-DDPG approach, while employing a non-discriminatory sampling technique from the experience replay buffer χ_U during the off-line learning process, faces several efficiency issues. This method's equal probability assignment to all experiences fails to prioritize learning from more informative instances, resulting in a slower learning rate and inefficiencies in sample replay utilization. Although the method reduces correlations inherent in sequentially generated on-line training experiences through randomization, it doesn't entirely decouple them, which can result in suboptimal learning outcomes. Furthermore, the algorithm's inability to

frequently revisit rare yet crucial experiences due to their low sampling probability further impedes the learning and convergence processes.

In regards to all the above shortcomings witnessed while using URBR-DDPG during the replay process, we have introduce an innovative sampling method, PBPR-DDPG which makes use of batch prioritization replay during its off-line learning process. It is during this sampling process that experiences of batch size M_b are sampled from the prioritization replay buffer memory χ_P to train the networks. For the agent to know which experience should be replayed more frequently, a concept known as Temporal Difference Error (TD-error) δ was introduced in [15]. δ measures the magnitude of the difference between transitions during the learning process. δ is defined as the difference between the predicted Q-value of a state-action pair and the actual reward received plus the predicted Q-value of the next state-action pair. It is used to determine how surprising or unexpected a generated experience is during the training process.The TD-error of the j^{th} experience at any time step t is computed from the equation below

$$\delta_j = r_j + \gamma Q_{c'}(s_j^{t+1}, \pi_{a'}(s_j^{t+1})) - Q_c(s_j^t, \pi_a(s_j^t)), \tag{25}$$

where $Q_c(s_j^t, \pi_a(s_j^t))$ is the predicted Q-value generated by the critic-network from the current state-action pair and $Q_{c'}(s_j^{t+1}, \pi_{a'}(s_j^{t+1}))$ is the future Q-value generated by the target critic-network from the next state-action pair. δ is used to assign priority to each transition after its computation. In the case of PBPR-DDPG, the probability p_j of sampling the j^{th} transition is directly proportional to its δ_j value. $p_j = |\delta_j| + \iota$, ι is a small constant added to the p_j to prevent it from turning to zero when the value of δ_j between two transition is zero. Each time a transition is added to χ_p memory, the priority of each transition has to be updated. In order to maintain the total probability of sampling any j^{th} experience from χ_p at time t to always be 1, the normalized sampling probability $P(j)$ of j^{th} experience out of the total experience j_{tot} present in the χ_p replay buffer is computed using the equation below

$$P(j) = \frac{p_j^\alpha}{\sum_{j_{tot}} P_{j_{tot}}^\alpha}, \tag{26}$$

where $\sum_{j_{tot}} P_{j_{tot}}^\alpha$ represents the sum of all the experience priorities present in χ_p to the power α. α is a stochastic hyperparameter which range between $(0, 1)$, and used to determine the degree of prioritization used during the off-line replay process. It controls how much priority is given to δ during the off-line learning process. So, a higher value of α increases the magnitude of prioritization during the sampling process. When $\alpha = 1$, sampling becomes fully prioritized, and reducing α toward zero pushes the sampling process toward uniform random sampling. Given that in PBPR-DDPG, the proportion of prioritization is directly proportional to δ, i.e, samples with higher priorities or δ values will be sampled more frequently, and in the worst case scenario, samples with lowest δ may end up never being sampled. Hence, introducing a biasing problem which may lead the agent to get stuck at a local maximum. PBPR-DDPG utilizes a hyperparameter β to adjust the importance sampling weight w_j, which compensates for biases in prioritized sampling by balancing each j^{th} experience's impact during the sampling process. This method maintains balanced training updates, preventing the overrepresentation of frequently sampled experiences in the learning process. w_j and its normalized form, w_{jN}, ensure stable training by preventing excessive weight escalation and its equations as outlined in [15] are given below

$$w_j = \left(\frac{1}{\mathfrak{M}_\mathfrak{R}} \times \frac{1}{P(j)}\right)^\beta, w_{jN} = \left(\frac{w_j}{max\ w_j}\right), \tag{27}$$

where $\mathfrak{M}_\mathfrak{R}$ is the total number of experience present in the χ_p memory during the importance sampling weight computation. $max\ w_j$ is the maximum computed weight obtained by a j^{th} experience out of the $\mathfrak{M}_\mathfrak{R}$ experiences present in the χ_p memory. β value ranges between $(0, 1)$, when $\beta = 0$, there is no prioritization, and the sampling process becomes uniformly random. When the value of $\beta = 1$, sampling is fully prioritized, given that the value of $\alpha = 1$. So, varying the value of α and β can reduce the biasing introduced while using prioritization in PBPR-DDPG.

E. RBPR-DDPG Training Process

One of the shortcomings of both URBR-DDPG and PRBR-DDPG is the problem of revisiting rare experiences. In the case of URBR-DDPG, since each experience has equal probability of being sampled, experience that occurs rarely are usually sampled less, which lead to a reduction in their contributions to the learning process. For the case of PBPR-DDPG, since the probability of sampling an experience during the off-line learning process is directly proportional to its priority, the experience with lower priority may never get sampled. Hence, eliminating their contribution to the learning process. This may lead the agent to settle for a local maximum solution for our optimization problem. Rank-base batch prioritization replay was introduced to solve the above-mentioned problems. Unlike PBPR-DDPG which assigns the probability of the j^{th} experience being sample to be directly proportional to its TD-error value, RBPR-DDPG sorts all the experiences in the χ_p memory during each sampling instants and assigns an index rank to each experience based on their $|\delta|$ value. The experience with the highest $|\delta|$ is assigned a $rank = 1$ value, the experience with the second highest $|\delta|$ value is assigned a $rank = 2$. This is done for all the experiences in the χ_p memory. Also, in our algorithm implementation, to reduce the computation power needed to resort all the experience in χ_p memory, each time a new experience is generated, it is automatically given a $rank = 1$. Note that the priorities here are only updated during the sampling replay period. The normalized sampling probability $P(j)$ of the j^{th} experience in RBPR-DDPG is given in (26) above.

Where $p_j = \frac{1}{(rank_j)^\alpha}$ is the probability of sampling the j^{th} experience and $rank_j$ is the rank of j^{th} experience for the case of RBPR-DDPG. The importance sampling weight w_j and the normalized importance sampling weight w_{jN} are computed using (27). In RBPR-DDPG, the transitions with the highest rank values are sampled most and those with lowest rank values are sampled less, but all the experiences are sample during the off-line learning process. Hence, RBPR-DDPG applies prioritization in its sampling process and also

gives the opportunity to all the experiences to sampled. This approach leads to an agent having a stable learning rate and a faster convergence rate with less biasing witnessed in URBR-DDPG and PRBR-DDPG above. The algorithm for the implementation of PRBR-DDPG and RRBR-DDPG is given in algorithm 1.

Algorithm 1 PBPR-DDPG/RBPR-DDPG Algorithm

Initialize: Initialize the priority replay buffer memory χ_P;
Create: Actor-network $\pi_a(.)$ and critic-network $Q_c(.)$, initialize their biases and network weights θ_a^π and θ_c^Q using Xavier uniform initializer;
Create: Target actor $\pi_{a'}(.)$ and target critic $Q_{c'}(.)$ and deep copy $\theta_{a'}^\pi \leftarrow \theta_a^\pi$ and $\theta_{c'}^Q \leftarrow \theta_c^Q$;
for episode = 1 to eps_{max} **do**
  Initialize the RIS-aided HSR communication network and observe the initial state s^t;
  for steps t = 1 to T **do**
    Select an action a^t according to the policy $\pi_a(s^t|\theta_a^\pi)$ and add the exploration noise;
    Execute the action a^t, observe the reward r^t and the new state s^{t+1};
    Add the experience tuple $e^t = (s^t, a^t, r^t, s^{t+1})$ in the priority replay buffer memory χ_P;
    Compute δ_j for j^{th} transition using (25) and update the experiences priorities in χ_P;
    Assign priority value to each experiences in χ_P according to their δ_j value for the case of PBPR-DDPG;
    Assign a rank to each experiences in χ_P according their δ_j value for the case of RBPR-DDPG;
    Assign the highest $rank$ to any newly generated experience in χ_P for the case of RBPR-DDPG;
    Sample e^t of batch size M_b , and compute their normalized sampling probability $P(j)$ of the j^{th} using (26) ;
    Compute the normalized importance sampling weight w_{jN} for the j^{th} transition using (27) ;
    Compute the target values $y_{j,c'}$ for j^{th} transition using (17);
    Compute the critic-network loss of the j^{th} transition $L(\theta_c^Q) = \frac{1}{M_b} \cdot w_{jN} \sum_j \left(y_{j,c'} - Q_c\left(s_j^t, a_j^t | \theta_c^Q \right) \right)^2$
    Update critic-network parameter θ_c^Q using (21) by gradient descent;
    Compute the actor network loss of the j^{th} transition $L(\theta_a^\pi) = -\frac{1}{M_b} \cdot w_{jN} \sum_j Q_c\left(s_j^t, \pi_a\left(s_j^t | \theta_a^\pi \right) | \theta_c^Q \right)$
    Compute the actor network policy gradient $(\nabla_{\theta_a^\pi} J)$ using (20);
    Update actor network parameter θ_a^π using (22);
    Soft update the critic and actor target networks parameters $\theta_{c'}^Q$ and $\theta_{a'}^\pi$ respectively using (23) and (24);
    Update the experience priorities in χ_p according to the latest computed δ;
  end for
end for

F. Algorithms computational complexity

The computational complexity analysis of the three algorithms (URBR-DDPG, PBPR-DDPG, and RBPR-DDPG) reveals significant differences in their efficiency. Let $|\mathcal{N}|$ denote the sampling mini batch size, $|\mathcal{M}|$ denote the replay buffer size, $|\mathcal{S}|$ denote the state dimension, $|\mathcal{A}|$ denote the action dimension, $(\text{eps}_{\max})$ denote the number of episodes and, $(\mathcal{T})$ denote the time steps per episode. The complexity of the three algorithms can be divided in to 3 subclasses, the offline training complexity (experience replay, actor-critic updates, target actor-critic update and priority handling), the online inference complexity (real-time decision-making) and memory complexity (Storage requirements). During the offline training phase, the computational burden mainly stems from the architecture and operations of actor-critic and target actor-critic neural networks. In each training iteration, the actor network is responsible for determining the optimal actions, such as beamforming vectors and phase shifts from the input states, while the critic network assesses the effectiveness of these actions by computing the state-action value function (Q-value) from state-action inputs. The offline training complexity calculation will be performed as in [43] and that associated with a forward pass through the layers of the actor network can be represented as:

$$O\left(|\mathcal{S}| L_1 + \sum_{p=1}^{P_a} L_p L_{p+1} \right), \tag{28}$$

where L_1 denotes the number of neurons in the first hidden layer of the network, L_p and L_{p+1} denote the number of neurons in the p-th hidden layer and the number of neurons on the subsequent $(p+1)$ hidden layer of the network respectively. P_a denotes the number hidden layers in the actor network. Likewise, the complexity of the critic network follows a similar structure to that of the actor network but includes additional computations for evaluating actions. The complexity of the critic network is expressed as:

$$O\left((|\mathcal{S}| + |\mathcal{A}|) L_1 + \sum_{p=1}^{P_c} L_p L_{p+1} \right), \tag{29}$$

where P_c denotes the number hidden layers in the critic network. Both networks undergo updates using mini-batches, and the total training complexity is influenced by factors such as the number of episodes $(eps_{\max})$, the time steps per episode, the mini-batch size. The overall training complexity (time complexity) for the URBR-DDPG algorithm can be formulated as:

$$O\Bigg(\text{eps}_{\max} \mathcal{T} |\mathcal{N}| \Bigg(|\mathcal{S}|.|\mathcal{A}| + 2\Bigg(|\mathcal{S}| L_1 + \sum_{p=1}^{P_a} L_p L_{p+1} \Bigg) + 2\Bigg((|\mathcal{S}| + |\mathcal{A}|) L_1 + \sum_{p=1}^{P_c} L_p L_{p+1} \Bigg) \Bigg) \Bigg) \tag{30}$$

The overall training complexity (time complexity) for the PBPR-DDPG algorithm can be formulated as:

$$O\Bigg(\text{eps}_{\max}\mathcal{T}|\mathcal{N}|\Bigg(|\mathcal{S}|.|\mathcal{A}|+2\Bigg(|\mathcal{S}|L_1+\sum_{p=1}^{P_a}L_pL_{p+1}\Bigg)+ 2\Bigg((|\mathcal{S}|+|\mathcal{A}|)L_1+\sum_{p=1}^{P_c}L_pL_{p+1}\Bigg)+|\mathcal{N}|\log|\mathcal{M}|\Bigg)\Bigg) \tag{31}$$

where $|\mathcal{N}|\log|\mathcal{M}|$ accounts for sum-tree prioritized experience replay sampling complexity. The overall training complexity (time complexity) for the RBPR-DDPG algorithm can be formulated as:

$$\begin{aligned}O\Bigg(&\text{eps}_{\max}\mathcal{T}|\mathcal{N}|\Bigg(|\mathcal{S}|.|\mathcal{A}|+2\Bigg(|\mathcal{S}|L_1+\sum_{p=1}^{P_a}L_pL_{p+1}\Bigg)\\ &+2\Bigg((|\mathcal{S}|+|\mathcal{A}|)L_1+\sum_{p=1}^{P_c}L_pL_{p+1}\Bigg)\Bigg)\\ &+\text{eps}_{\max}\mathcal{T}|\mathcal{N}|\Bigg(|\mathcal{N}|\log|\mathcal{M}|+|\mathcal{M}|\log|\mathcal{M}|\Bigg)\Bigg)\end{aligned} \tag{32}$$

where: $|\mathcal{M}|\log|\mathcal{M}|$ denotes the sorting complexity for ranking the experiences in the priority replay buffer. The online inference complexity is similar for all the three algorithms and consists of the actor network making real-time decision on the beamforming and the phaseshifts matrices configuration for all the RISs present. This involves a forward pass through the actor network, with complexity as:

$$O\left(L_0^aL_1+\sum_{p=1}^{P_L}L_p^aL_{p+1}^a\right), \tag{33}$$

where L_0^a and L_p^a respectively denote the input and the hidden layer neurons of the actor network, and P_L represents the total number of layers in the actor network. As observed from (33), the computational complexity during this inference phase is lower compared to the training stage and remains similar across all three algorithms, making them suitable for practical implementation in rapidly changing HSR communication systems. Additionally, since training is performed at the BS, which has substantial computational resources, the proposed algorithms benefit from enhanced practical viability. The memory complexity (space complexity) requirements remain consistent at $\mathcal{O}(\mathcal{M}\times(|\mathcal{S}|+|\mathcal{A}|))$ for all three algorithms. The space complexity with respect to the networks size for all three algorithms is given as:

$$\mathcal{O}\Bigg(\mathcal{M}\times(|\mathcal{S}|+|\mathcal{A}|)+2\sum_{p=1}^{P_a}L_pL_{p+1}+2\sum_{p=1}^{P_c}L_pL_{p+1}\Bigg), \tag{34}$$

This analysis demonstrates a clear trade-off between computational complexity and sampling sophistication, with more advanced sampling methods incurring higher computational costs while requiring fewer episodes to achieve convergence.

In Table I, the performance of the URBR-DDPG, PBPR-DDPG, and RBPR-DDPG algorithms are compared for Graphics Processing Unit (GPU) using Computing Unified Device Architecture (CUDA) technology and Central Processing Unit (CPU) using Open Multi-Processing (OpenMP) platforms in [44] for parallel implementation. The GPU platform achieves substantially lower average step duration, episode duration, and total training time as shown in Table I. This is because NVIDIA GPU acceleration with CUDA technology enables massive parallelization of neural network computations. GPUs contain thousands of cores optimized for matrix operations, which are the fundamental calculations in deep reinforcement learning algorithms. This parallel architecture allows simultaneous processing of multiple network layers, batch samples, and gradient calculations that would otherwise be processed sequentially on CPUs. In contrast, the CPU implementation using OpenMP with parallel implementation exhibits higher average step duration, episode duration, and total training time with a varying memory range shown in Table I, reflecting dynamic memory assignment as the algorithm processes different computational loads on a multi-core system. The discrepancy between average step time and episode duration stems from computational overhead not captured in step measurements. While step time reflects neural network passes, episode duration includes experience replay management, target network updates, and environment resets. PBPR-DDPG and RBPR-DDPG introduce substantial prioritization overhead at episode boundaries rather than during individual steps. These results, obtained from extensive simulations of 100 episodes per algorithm with 50 steps per episode, underscore the robustness of the performance evaluation and highlight the advantages of GPU acceleration in real-time applications.

V. Simulation results and analysis

In this section, we evaluated the performance of URBR-DDPG, PBPR-DDPG, and RBPR-DDPG algorithms for solving the formulated optimization problem. The machine used for this simulation is a windows Server, Intel(R) Xeon(R) Gold 6226R CPU, clock speed of 2.90GHz, base frequency of 2893MHz with 16 cores and 32 threads with GPU NVIDIA A40. The evaluation was done for the performance of the different solutions at different sampling replay batch sizes of $16, 32, 64, 128$, in terms of instantaneous spectral efficiency, average spectral efficiency, training stability and actor network loss. We then compare the performance of all three algorithms using the same sampling replay batch size for which the best spectral efficiency results was obtained. We then verify the correctness of our proposed solutions by observing the effect of changing the velocity of the HST-MR and the number of reflecting elements of the RISs on the overall system spectral efficiency. The execution of each of the algorithm was done using the same RIS-aided HSR communication system simulation parameters given in Table II and those given in this work.

The actor, critic, and their target networks consist of two hidden layers. Bitwise decomposition of the state dimension was done to obtain the number of neurons used in their

TABLE I
PERFORMANCE COMPARISON OF URBR-DDPG, PBPR-DDPG, AND RBPR-DDPG ALGORITHMS ON DIFFERENT HARDWARE PLATFORMS

Performance Metric	**GPU Implementation**			**CPU Implementation (OpenMP)**		
	URBR-DDPG	**PBPR-DDPG**	**RBPR-DDPG**	**URBR-DDPG**	**PBPR-DDPG**	**RBPR-DDPG**
Average Step Duration (s)	0.004412	0.004898	0.005058	0.006044	0.006862	0.007154
Episode Duration (s)	1.0334	1.1609	1.1878	4.2503	5.5483	5.8750
Total Training Time (s)	108.88	123.00	124.37	485.79	595.76	601.71
Memory Usage (MB)	47.79	53.53	53.53	374.30-507.57	380.00-516.45	381.04-517.84

Hardware Specifications:
GPU Platform: Windows Server Intel Xeon Gold 6226R CPU (2.90GHz, 16 cores, 32 threads) with GPU NVIDIA A40 acceleration with CUDA support. CPU Platform: Intel Core i7-7700 CPU (3.60GHz, 4 cores, 8 logical processors) with OpenMP parallelization. Test Configuration: 100 episodes, 50 steps per episode, RIS elements = 16, BS antennas = 8, MR antennas = 8, batch size = 128.

TABLE II
SIMULATION HYPER-PARAMETERS

Symbol	Representation	Simulation value
f_c	Carrier frequency	28 GHz
γ	Discount factor	0.95
λ_a^{π}	Learning rate of actor-network	0.001
λ_c^{Q}	Learning rate of critic-network	0.001
μ_a	Decay rate of actor optimizer	0.00001
μ_c	Decay rate of critic optimizer	0.00001
$\tau_{a'}^{\pi}$	Learning rate of target actor	0.001
$\tau_{c'}^{Q}$	Learning rate of target critic	0.001
χ_U	Uniform replay memory size	100000
χ_p	Priority replay memory size	100000
eps_{max}	Number of training Episode	15000
T_{steps}	Training steps per episode	50
h_{RIS}	Height of the RIS from ground	5 m
v	HST-MR velocity	350 km/h
T_s	system sampling period	10 ms
τ_s	Outdated CSI delay	10 ms

hidden layer. This is essentially done by breaking down the dimensions of the state into sums of powers of two. This helps to establish a direct correlation between the state space complexity and the neural network's structure. This technique not only ensures that the size of the network is appropriately matched to the intricacies of the state space but also aids in stabilizing the learning process. Given the vast number of experiences generated during DDPG's off-line training, having a network architecture that effectively mirrors the complexity of the state space can lead to more efficient learning and reduced risks of overfitting or underfitting. Ornstein-Uhlenbeck noise is added to each action after being generated from the actor-network so as to improve the agent exploration ability. State whitening and batch normalization were used to stabilize the training.

RIS-aided HSR communication system parameters, the length of the tunnel is taken as 300 m (considered as the reference distance), BS $-$ RIS$_1$ separation is taken as D_{BR1} = 100 m and RIS$_1$ $-$ RIS$_2$ separation is taken as D_{B12} = 100 m . The width of the tunnel is taken as 6.5 m. The HST-MR height above the ground is taken as y_{MR} = 4.5 m and is assumed to be moving in linear path with a constant velocity of v = 350 km/h for the indicated reference distance. The maximum BS transmitting power is taken as P_{BSmax} = 30 dBm, number of antennas at BS is N$_{\text{BS}}$ = 8, number of antennas on the HST-MR is N$_{\text{MR}}$ = 8. The number of elements at RIS$_1$ is taken as N$_1$M$_1$ = 16 and RIS$_2$ elements is taken as N$_2$M$_2$ = 16. The path loss factor used in this simulation is similar to that used in [14] with $\alpha_{\text{BS}-\text{RIS}_1} = \alpha_{\text{RIS}_1-\text{RIS}_2}$ = 3 and $\alpha_{\text{RIS}_1-\text{MR}} = \alpha_{\text{RIS}_2-\text{MR}}$ = 2.8, the variance of shadow fading σ_s^2 = 4 dB. The AWGN variance is taken as σ^2 = -114 dBm.

A. *Case1:* $\alpha = 1$ *and annealed* β

Throughout the 15000 episodes, we maintained $\alpha = 1$ while β was gradually reduced from $\beta_{start} = 1.0$ to $\beta_{end} = 0.1$. As shown in Fig. 3, for PBPR-DDPG, during the initial 1000 episodes, the spectral efficiency (SE) was highest for batch 16, likely due to the smaller batch size and the maximal values of α and β, enhancing SE early on. However, between episodes 1000 and 4000, SE became unstable for batch sizes 16 and 32 as β was annealed, diminishing prioritization's impact. Larger batch sizes, such as 64 and 128, showed better and more stable performance in this range because sampling from a larger pool allowed for the inclusion of rare and more informative experiences, promoting stable convergence as seen in Fig. 5. Conversely, RBPR-DDPG quickly reached a stable SE of about 12.5 bps/Hz across all batch sizes by episode 500, as illustrated in Fig. 4. The initial high prioritization coupled with effective experience replay allowed the agent to rapidly assimilate environmental variances and achieve optimal performance. This is evident from Fig. 5, where RBPR-DDPG's smaller batches matched the performance of larger PBPR-DDPG batches.

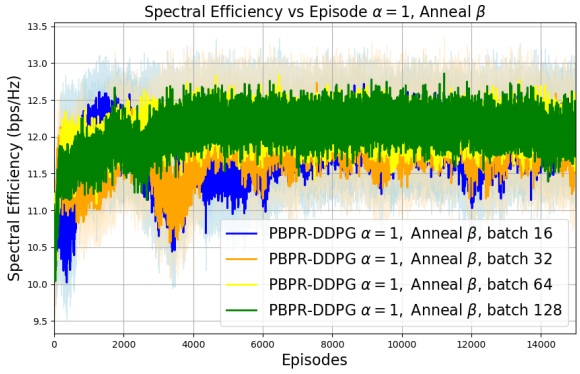

Fig. 3. PBPR-DDPG Spectral Efficiency $\alpha = 1$, annealed β from $\beta_{start} = 1.0$ to $\beta_{end} = 0.1$ for different sampling batch size.

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVT.2025.3582264

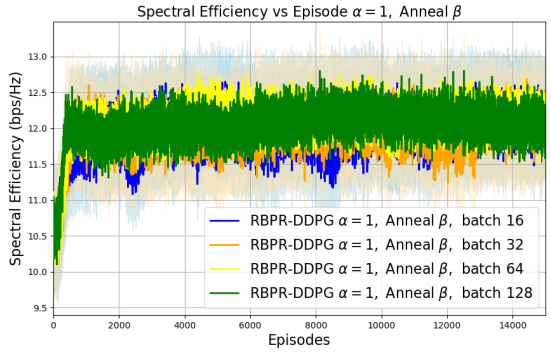

Fig. 4. RBPR-DDPG Spectral Efficiency $\alpha = 1$, annealed β from $\beta_{start} = 1.0$ to $\beta_{end} = 0.1$ for different sampling batch size.

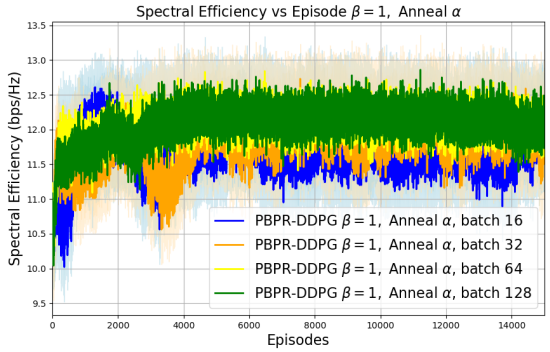

Fig. 6. PBPR-DDPG Spectral Efficiency $\beta = 1$, annealed α from $\alpha_{start} = 1.0$ to $\alpha_{end} = 0.1$ for different sampling batch size.

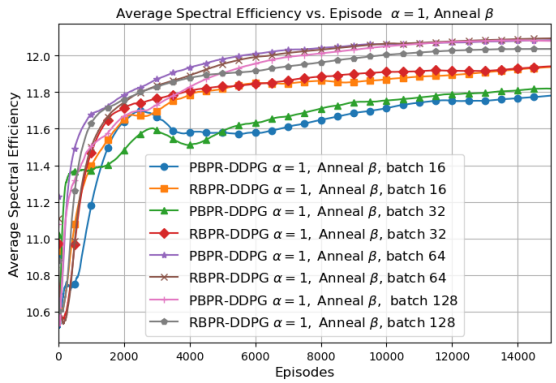

Fig. 5. Combined PBPR-DDPG and RBPR-DDPG Average Spectral Efficiency $\alpha = 1$, annealed β from $\beta_{start} = 1.0$ to $\beta_{end} = 0.1$ for different sampling batch size.

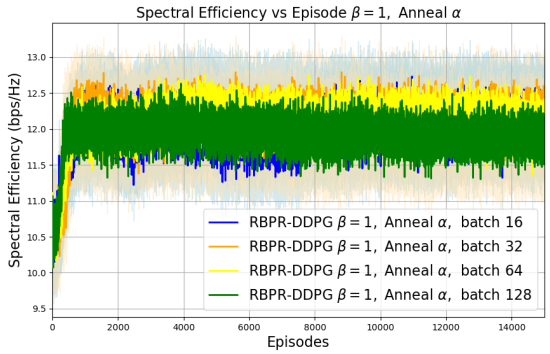

Fig. 7. RBPR-DDPG Spectral Efficiency $\beta = 1$, annealed α from $\alpha_{start} = 1.0$ to $\alpha_{end} = 0.1$ for different sampling batch size.

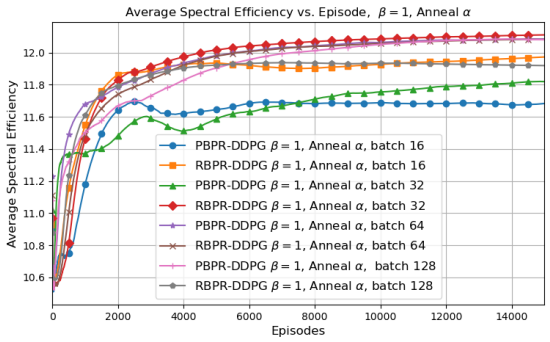

Fig. 8. Combined PBPR-DDPG and RBPR-DDPG Average Spectral Efficiency $\beta = 1$, annealed α from $\alpha_{start} = 1.0$ to $\alpha_{end} = 0.1$ for different sampling batch size.

B. Case2: $\beta = 1$ and annealed α

For the 15000 episodes, β was maintained at 1 while α values were annealed from $\alpha_{start} = 1.0$ to $\alpha_{end} = 0.1$. Fig. 6 illustrates that the SE performance of the PBPR-DDPG algorithm was akin to that of Case1, with batch 16's quickly reaching an average SE of 11.6 bps/Hz as depicted in Fig. 8 and maintaining this value throughout the simulation. The consistent prioritization from a constant β and the diminishing effect from annealed α allowed batch 16 to learn from a broader range of experiences, yielding a more stable learning process than in Case1. SE performance for RBPR-DDPG, as shown in Fig. 7, indicates all batch sizes reached their optimal SE post-500 episodes, similar to Case1. However, batch 128's performance declined toward the end of the simulation, as seen in Fig. 8. This drop is attributed to the constant β and decreasing prioritization over time, leading to a surplus of replayed, less informative experiences towards the simulation's end, resulting in a reduction of the SE.

C. Case3: Annealed α and ramp-up β

Throughout 15000 episodes, α was reduced from $\alpha_{start} = 1.0$ to $\alpha_{end} = 0.1$, while β was increased from $\beta_{start} = 0.1$ to $\beta_{end} = 1.0$. Notably from Fig. 9, the PBPR-DDPG algorithm achieved a SE over 13 bps/Hz, surpassing the results of Case 1 and Case2. Batches 64 and 128 reached an SE of 13 bps/Hz by 2000 episodes, while smaller batches 16 and 32 became unstable afterward. Initially, a low β induced uniform random sampling, but as β rose, smaller batches' SE dropped, contrary to earlier Cases where they thrived on high prioritization (Fig. 11). In contrast, RBPR-DDPG's SE remained stable across all batches except for batch 128, which destabilized after 6000 episodes, as shown in Fig. 10. At this point, $\alpha \sim 0.6$ and $\beta \sim 0.4$ led to a mix of uniform random sampling and prioritization. Nonetheless, batch 128 maintained an SE of

13 bps/Hz due to its ranking system, which included replaying informative experiences as shown in Fig. 11.

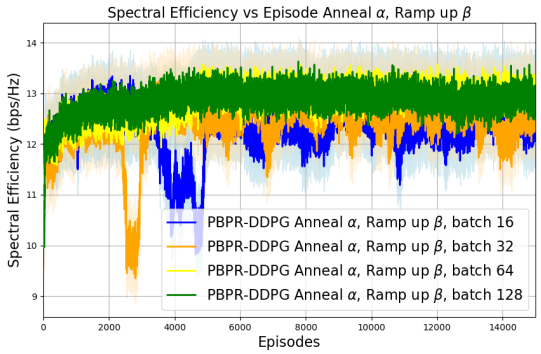

Fig. 9. PBPR-DDPG Spectral Efficiency annealed α from $\alpha_{start} = 1.0$ to $\alpha_{end} = 0.1$ and ramp up β from $\beta_{start} = 0.1$ to $\beta_{end} = 1.0$ for different sampling batch size.

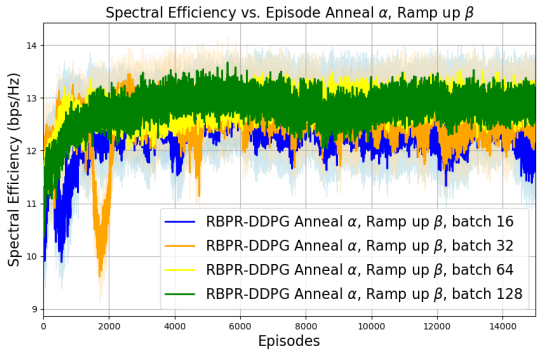

Fig. 10. RBPR-DDPG Average Spectral Efficiency annealed α from $\alpha_{start} = 1.0$ to $\alpha_{end} = 0.1$ and ramp up β from $\beta_{start} = 0.1$ to $\beta_{end} = 1.0$ for different sampling batch size.

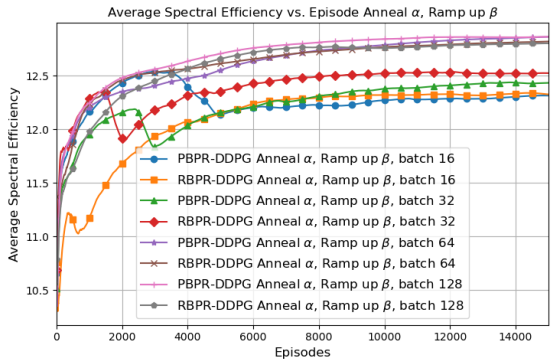

Fig. 11. Combined PBPR-DDPG and RBPR-DDPG Average Spectral Efficiency annealed α from $\alpha_{start} = 1.0$ to $\alpha_{end} = 0.1$ and ramp up β from $\beta_{start} = 0.1$ to $\beta_{end} = 1.0$ for different sampling batch size.

D. Case4: Annealed β and ramp-up α

During 15000 episodes, β is decreased from $\beta_{start} = 1.0$ to $\beta_{end} = 0.1$, while α increased from $\alpha_{start} = 0.1$ to $\alpha_{end} = 1.0$. PBPR-DDPG's SE, as shown in Fig.12, peaked at 13 bps/Hz for all batches post-2000 episodes, similar to Case3, except for batch 16, which destabilized past 3000 episodes, obtaining the lowest results of all as shown in Fig. 14. This instability is due to reduction in prioritization combined with an increased sampling bias from the rising α and a smaller sampling batch size, limiting learning from fewer experiences. Contrarily, RBPR-DDPG, as shown in Fig. 13, achieved optimal SE performance, with batch 128 hitting 13 bps/Hz within 200 episodes and sustaining it. Fig. 14 highlights that all RBPR-DDPG batches surpassed an average SE of 12.5 bps/Hz, marking the highest achievement among the proposed solutions across all batch sizes.

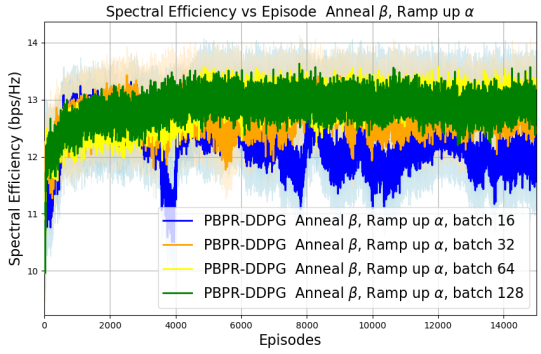

Fig. 12. PBPR-DDPG Spectral Efficiency annealed β from $\beta_{start} = 1.0$ to $\beta_{end} = 0.1$ and ramp up α from $\alpha_{start} = 0.1$ to $\alpha_{end} = 1.0$ for different sampling batch size.

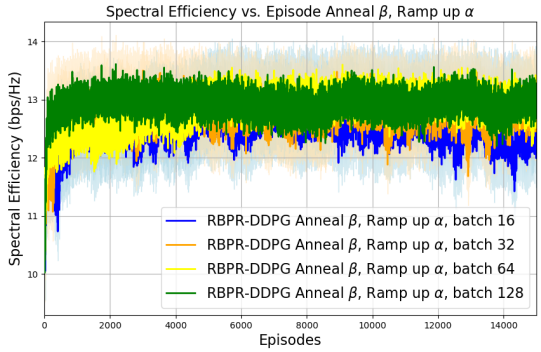

Fig. 13. RBPR-DDPG Spectral Efficiency annealed β from $\beta_{start} = 1.0$ to $\beta_{end} = 0.1$ and ramp up α from $\alpha_{start} = 0.1$ to $\alpha_{end} = 1.0$ for different sampling batch size.

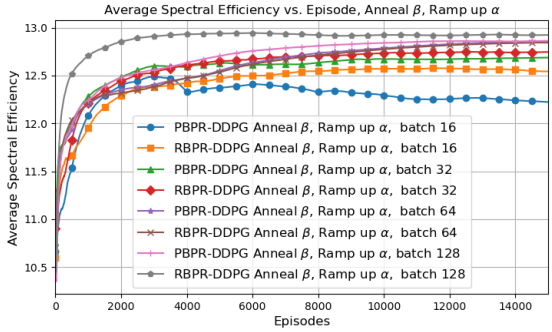

Fig. 14. Combined PBPR-DDPG and RBPR-DDPG Spectral Efficiency annealed β from $\beta_{start} = 1.0$ to $\beta_{end} = 0.1$ and ramp up α from $\alpha_{start} = 0.1$ to $\alpha_{end} = 1.0$ for different sampling batch size.

E. Case5: PBPR-DDPG, RBPR-DDPG and URBR-DDPG Performance

In the interest of a fair comparison and with the goal of maximizing SE at the HST-MR in the tunnel, we selected

Case4's strategy of annealing β and ramping up α for a batch size of 128 for both PBPR-DDPG and RBPR-DDPG. This choice was made as it yielded the best results across the four cases. As depicted in Fig. 15, RBPR-DDPG outperformed the others, showcasing superior convergence rate, training stability, and SE maximization. In contrast, URBR-DDPG had the lowest performance, converging to around 12 bps/Hz, which, despite being the least effective, was an improvement from its starting SE of approximately 10.5 bps/Hz as seen in Fig. 15. This comparison underscores the efficacy of the proposed algorithms relative to the benchmark URBR-DDPG, affirming the validity of our optimization approach.

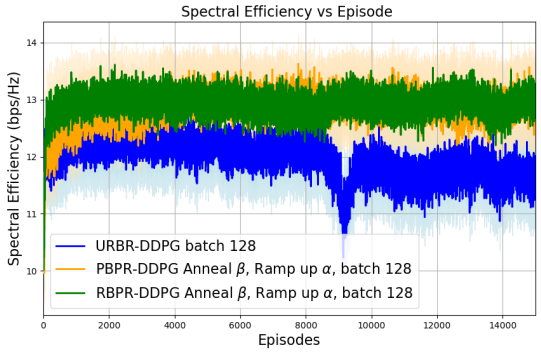

Fig. 15. Compare RBPR-DDPG, PBPR-DDPG, URBR-DDPG Spectral Efficiency.

The actor network in DDPG determines an optimal policy mapping states to actions that maximize the action-value function. Its loss function is defined as the negative expected Q-value, thus enabling the use of gradient descent methods to perform effective gradient ascent on the Q-value. This formulation allows the deterministic policy to be updated in the direction that increases expected future returns as estimated by the critic network. Our proposed algorithms achieved substantial actor loss values around -13, with RBPR-DDPG recording the lowest (most negative) value, indicating better policy improvement. URBR-DDPG's inferior SE performance correlates with its actor loss behavior in Fig. 16, where it unexpectedly increased toward simulation end rather than continuing to decrease, resulting in lower SE convergence.

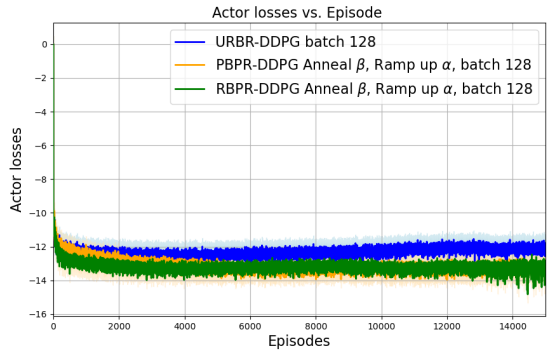

Fig. 16. Compare RBPR-DDPG, PBPR-DDPG, URBR-DDPG Actor loss.

Fig. 17, illustrates how HST-MR velocity affects spectral efficiency across all algorithms, demonstrating the relationship with Doppler shift ($f_{RM} = \frac{v}{\lambda_c}\cos(\theta_{\text{MR,AOA}})$). At 50 km/h, RBPR-DDPG achieves optimal performance 14 bps/Hz due to high temporal correlation $\rho_d \approx 1$ from (12) ensuring accurate channel estimation. At 350 km/h, RBPR-DDPG maintains superior performance 12.8 bps/Hz compared to PBPR-DDPG 12.2 bps/Hz and URBR-DDPG 11.5 bps/Hz, highlighting our prioritized experience replay's effectiveness in adapting to rapidly changing channels. Even at extreme velocity 600 km/h, our algorithms show remarkable resilience with only a 1.5 bps/Hz reduction despite increased Doppler shift ($f_{max} = \frac{v}{c}f_c$). This robustness stems from effective learning of channel temporal patterns and prioritized sampling of challenging scenarios, confirming our approach's viability for next-generation HSR communications at higher speeds.

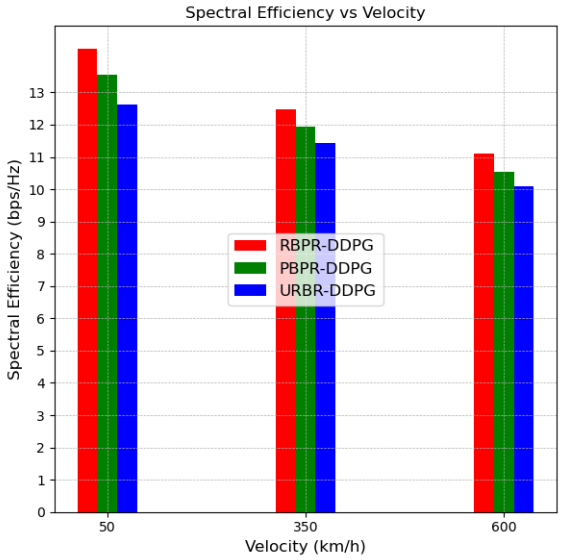

Fig. 17. Compare RBPR-DDPG, PBPR-DDPG, URBR-DDPG Spectral Efficiency for different velocities.

Fig. 18, shows the variation of the system SE with an increase in the number of RIS elements. It can be observed that there is a steady increase in the system SE as the number of RIS elements is increased for all the algorithms, with our two proposed algorithms exhibiting the best performance. This improvement occurs for several fundamental reasons: First, a larger number of reflecting elements enhances the beamforming gain by providing more degrees of freedom for phase shift optimization, allowing for more precise signal focusing toward the HST-MR. Second, additional RIS elements improve the spatial diversity and signal-to-noise ratio by coherently combining multiple signal paths, which is particularly beneficial in the tunnel environment where signal blockage and multipath fading are prevalent. Furthermore, the enhanced performance of our proposed algorithms compared to URBR-DDPG demonstrates their superior ability to learn optimal phase configurations when handling the increased complexity and dimensionality associated with more RIS elements. This result not only validates the effectiveness of our prioritized experience replay mechanisms but also confirms that deploy-

ment of multiple RISs with sufficient reflecting elements can significantly improve communication quality in HSR tunnel scenarios.

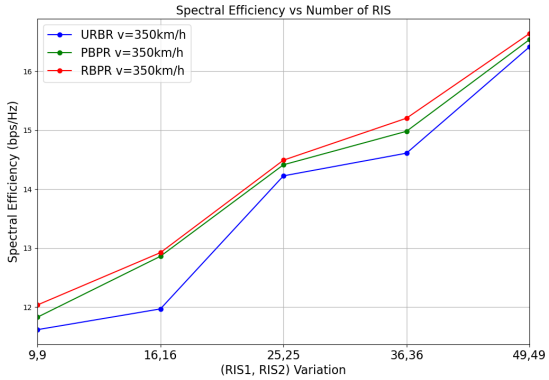

Fig. 18. Compare RBPR-DDPG, PBPR-DDPG, URBR-DDPG Spectral Efficiency for different RIS elements number.

VI. Conclusion

In this paper, the problem of signal quality improvement and spectral efficiency maximization at the level of HST-MR as it passes through a tunnel was presented. A joint beamforming and RISs phase shift matrices optimization problem was formulated to maximize the spectral efficiency for the RIS-aided HSR scenario. Two batch prioritization replay algorithms namely: PBPR-DDPG and RBPR-DDPG were proposed to solve the formulated problem. The simulation results demonstrated the effectiveness of our proposed solution and its advantage over the modified benchmark URBR-DDPG algorithm. As a future work direction, through transfer learning, the real world application of this proposed work will be tested in an existing HSR tunnel network. Also, we plan to work on reducing the sorting and ranking computation complexity in both our proposed PBPR-DDPG and RBPR-DDPG algorithm to fully adapt to the fast changing HSR communication system. Additionally, we will investigate service-aware optimization strategies that balance spectrum efficiency with the varying latency, bandwidth, and reliability requirements of different service types (voice, video, and data) in high-speed scenarios. Furthermore, we aim to address security challenges by developing anti-spoofing mechanisms for RIS-aided HSR communications and ensuring user data privacy when applying machine learning techniques in safety-critical railway environments.

References

[1] J. Xu and B. Ai, "Artificial intelligence empowered power allocation for smart railway," *IEEE Communications Magazine*, vol. 59, no. 2, pp. 28–33, 2021.

[2] D. Jia, F. Hu, Z. Ling, and S. Na, "AoI-aware power control and subcarrier assignment in D2D-aided underlaying cellular networks for high-speed railways," *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[3] J. Zhao, J. Liu, L. Yang, B. Ai, and S. Ni, "Future 5G-oriented system for urban rail transit: Opportunities and challenges," *China Communications*, vol. 18, no. 2, pp. 1–12, 2021.

[4] H. Ghazzai, T. Bouchoucha, A. Alsharoa, E. Yaacoub, M.-S. Alouini, and T. Y. Al-Naffouri, "Transmit power minimization and base station planning for high-speed trains with multiple moving relays in OFDMA systems," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 1, pp. 175–187, 2016.

[5] J. Wu and P. Fan, "A survey on high mobility wireless communications: Challenges, opportunities and solutions," *IEEE Access*, vol. 4, pp. 450–476, 2016.

[6] A. Ghazal, Y. Yuan, C.-X. Wang, Y. Zhang, Q. Yao, H. Zhou, and W. Duan, "A non-stationary imt-advanced MIMO channel model for high-mobility wireless communication systems," *IEEE Transactions on Wireless Communications*, vol. 16, no. 4, pp. 2057–2068, 2016.

[7] B. Ai, A. F. Molisch, M. Rupp, and Z.-D. Zhong, "5G key technologies for smart railways," *Proceedings of the IEEE*, vol. 108, no. 6, pp. 856–893, 2020.

[8] T. S. Rappaport, Y. Xing, O. Kanhere, S. Ju, A. Madanayake, S. Mandal, A. Alkhateeb, and G. C. Trichopoulos, "Wireless communications and applications above 100 GHz: Opportunities and challenges for 6G and beyond," *IEEE access*, vol. 7, pp. 78 729–78 757, 2019.

[9] H. Liu, J. Zhang, Q. Wu, Y. Jin, Y. Chen, and B. Ai, "RIS-aided next-generation high-speed train communications: Challenges, solutions, and future directions," *arXiv preprint arXiv:2103.09484*, 2021.

[10] I. Yildirim, A. Uyrus, and E. Basar, "Modeling and analysis of reconfigurable intelligent surfaces for indoor and outdoor applications in future wireless networks," *IEEE transactions on communications*, vol. 69, no. 2, pp. 1290–1301, 2020.

[11] K. Wang, C.-T. Lam, and B. K. Ng, "Positioning information based high-speed communications with multiple RISs: Doppler mitigation and hardware impairments," *Applied Sciences*, vol. 12, no. 14, p. 7076, 2022.

[12] K. Shen and W. Yu, "Fractional programming for communication systems—part I: Power control and beamforming," *IEEE Transactions on Signal Processing*, vol. 66, no. 10, pp. 2616–2630, 2018.

[13] Q. Shi, M. Razaviyayn, Z.-Q. Luo, and C. He, "An iteratively weighted MMSE approach to distributed sum-utility maximization for a MIMO interfering broadcast channel," *IEEE Transactions on Signal Processing*, vol. 59, no. 9, pp. 4331–4340, 2011.

[14] J. Xu and B. Ai, "When mmWave high-speed railway networks meet reconfigurable intelligent surface: A deep reinforcement learning method," *IEEE Wireless Communications Letters*, vol. 11, no. 3, pp. 533–537, 2021.

[15] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," *arXiv preprint arXiv:1511.05952*, 2015.

[16] D. Andre, N. Friedman, and R. Parr, "Generalized prioritized sweeping," *Advances in Neural Information Processing Systems*, vol. 10, 1997.

[17] C. Liu, R. He, Y. Niu, Z. Han, B. Ai, M. Gao, Z. Ma, G. Wang, and Z. Zhong, "Reconfigurable intelligent surface assisted high-speed train communications: Coverage performance analysis and placement optimization," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 3, pp. 3750–3766, 2023.

[18] P. Li, Y. Niu, H. Wu, Z. Han, G. Sun, N. Wang, Z. Zhong, and B. Ai, "RIS-assisted high-speed railway integrated sensing and communication system," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 12, pp. 15 681–15 692, 2023.

[19] Y. Yuan, R. He, B. Ai, Y. Niu, M. Yang, G. Wang, R. Chen, Y. Li, J. Li, J. Ding *et al.*, "A 3D geometry-based reconfigurable intelligent surfaces-assisted mmwave channel model for high-speed train communications," *IEEE Transactions on Vehicular Technology*, pp. 1–16, 2023.

[20] Y. Gao, Y. Wang, C. Li, J. Xie, and M. Wang, "Research on joint beamforming of high-speed railway millimeter-wave MIMO communication with reconfigurable intelligent surface," *Alexandria Engineering Journal*, vol. 74, pp. 317–326, 2023.

[21] M. Gao, B. Ai, Y. Niu, Z. Han, and Z. Zhong, "IRS-assisted high-speed train communications: Outage probability minimization with statistical CSI," in *ICC 2021-IEEE International Conference on Communications*. IEEE, 2021, pp. 1–6.

[22] Z. Ma, Y. Wu, M. Xiao, G. Liu, and Z. Zhang, "Interference suppression for railway wireless communication systems: A reconfigurable intelligent surface approach," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 11, pp. 11 593–11 603, 2021.

[23] C. You, B. Zheng, and R. Zhang, "Wireless communication via double irs: Channel estimation and passive beamforming designs," *IEEE Wireless Communications Letters*, vol. 10, no. 2, pp. 431–435, 2020.

[24] L. Yang, Y. Yang, D. B. da Costa, and I. Trigui, "Outage probability and capacity scaling law of multiple RIS-aided networks," *IEEE Wireless Communications Letters*, vol. 10, no. 2, pp. 256–260, 2020.

[25] Y. Wang, W. Zhang, Y. Chen, C.-X. Wang, and J. Sun, "Novel multiple RIS-assisted communications for 6G networks," *IEEE Communications Letters*, vol. 26, no. 6, pp. 1413–1417, 2022.

[26] Y. Zhao, W. Xu, X. You, N. Wang, and H. Sun, "Cooperative reflection and synchronization design for distributed multiple-RIS communications," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 5, pp. 980–994, 2022.

[27] M. Liu, C. Huang, M. Di Renzo, M. Debbah, and C. Yuen, "Cooperative beamforming and RISs association for multi-RISs aided multi-users mmwave MIMO systems through graph neural networks," in *ICC 2023-IEEE International Conference on Communications*. IEEE, 2023, pp. 4286–4291.

[28] M. Segata, P. Casari, M. Lestas, A. Papadopoulos, D. Tyrovolas, T. Saeed, G. Karagiannidis, and C. Liaskos, "Cooperis: A framework for the simulation of reconfigurable intelligent surfaces in cooperative driving environments," *Computer Networks*, vol. 248, p. 110443, 2024.

[29] K. D. Katsanos, P. Di Lorenzo, and G. C. Alexandropoulos, "Multi-RIS-empowered multiple access: A distributed sum-rate maximization approach," *IEEE Journal of Selected Topics in Signal Processing*, vol. 18, no. 7, pp. 1324–1338, 2024.

[30] G.-H. Li, D.-W. Yue, and S.-N. Jin, "Performance analysis of multiple RISs aided multi-user mmwave MIMO systems," *Wireless Networks*, vol. 30, no. 3, pp. 1911–1924, 2024.

[31] C. Huang, R. Mo, and C. Yuen, "Reconfigurable intelligent surface assisted multiuser MISO systems exploiting deep reinforcement learning," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 8, pp. 1839–1850, 2020.

[32] J. Ge, Y.-C. Liang, J. Joung, and S. Sun, "Deep reinforcement learning for distributed dynamic MISO downlink-beamforming coordination," *IEEE Transactions on Communications*, vol. 68, no. 10, pp. 6070–6085, 2020.

[33] J. Xu, B. Ai, T. Q. Quek, and Y. Liuc, "Deep reinforcement learning for interference suppression in RIS-aided high-speed railway networks," in *2022 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2022, pp. 337–342.

[34] J. Xu and B. Ai, "Experience-driven power allocation using multi-agent deep reinforcement learning for millimeter-wave high-speed railway systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 6, pp. 5490–5500, 2021.

[35] Y. Lee, J.-H. Lee, and Y.-C. Ko, "Beamforming optimization for IRS-assisted mmwave V2I communication systems via reinforcement learning," *IEEE Access*, vol. 10, pp. 60 521–60 533, 2022.

[36] D.-F. Wu, C. Huang, Y. Yin, S. Huang, Q. Guo, L. Zhang *et al.*, "State aware-based prioritized experience replay for handover decision in 5G ultradense networks," *Wireless Communications and Mobile Computing*, vol. 2022, 2022.

[37] H. Yang, Z. Xiong, J. Zhao, D. Niyato, L. Xiao, and Q. Wu, "Deep reinforcement learning-based intelligent reflecting surface for secure wireless communications," *IEEE Transactions on Wireless Communications*, vol. 20, no. 1, pp. 375–388, 2020.

[38] J. Ma, D. Ning, C. Zhang, and S. Liu, "Fresher experience plays a more important role in prioritized experience replay," *Applied Sciences*, vol. 12, no. 23, p. 12489, 2022.

[39] Y. A. T. Nana, G. Liu, and B. W. Tienin, "Reconfigurable intelligent surface for high-speed railway mmwave communication system: A deep reinforcement learning approach," in *2024 IEEE 100th Vehicular Technology Conference (VTC2024-Fall)*. IEEE, 2024, pp. 1–7.

[40] P. Wang, J. Fang, H. Duan, and H. Li, "Compressed channel estimation for intelligent reflecting surface-assisted millimeter wave systems," *IEEE signal processing letters*, vol. 27, pp. 905–909, 2020.

[41] Y. Chen, Y. Wang, and L. Jiao, "Robust transmission for reconfigurable intelligent surface aided millimeter wave vehicular communications with statistical CSI," *IEEE Transactions on Wireless Communications*, vol. 21, no. 2, pp. 928–944, 2021.

[42] K. Singh, A. F. Makarim, H. Albinsaid, C.-P. Li, and Z. J. Haas, "Passive beamforming design and DNN-based signal detection in RIS-assisted MIMO systems with generalized spatial modulation," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 2, pp. 1879–1892, 2022.

[43] M. M. Kamal, S. Z. U. Abideen, S. S. Shah, N. Sehito, S. Khan, B. S. Virdee, M. Alibakhshikenari, and P. Livreri, "Secure satellite downlink with hybrid RIS and ai-based optimization," *IEEE Access*, vol. 13, pp. 3726–3737, 2024.

[44] M. Rakhimov, S. Javliev, and R. Nasimov, "Parallel approaches in deep learning: Use parallel computing," in *Proceedings of the 7th International Conference on Future Networks and Distributed Systems*, 2023, pp. 192–201.

Yannick Abel Talla Nana (PhD Student) completed his Bachelor's of Technology in Electronics and Communication Engineering at the National Institute of Technology Silchar, India, in 2016. He then received his Master's Degree in Information and Communication Engineering from the University of Electronics Science and Technology of China in 2021. Currently, he is pursuing his Ph.D. in Information and Communication Engineering at Southwest Jiaotong University in China. His current research interests includes High-Speed Rail Wireless Communications, Internet of Things, and the application of Artificial Intelligence in Wireless Communication for mmWave transmission in 5G and beyond.

Gang Liu (Member, IEEE) received the B.S. degree in electronics and information engineering from Sichuan University (SCU) in 2010, and the Ph.D. degree in communication and information systems from the Beijing University of Posts and Telecommunications (BUPT) in 2015. He was also with the University of British Columbia and Carleton University as a Visiting Ph.D. Student from November 2013 to November 2014. He is currently an Associate Professor with the School of Information Science and Technology, Southwest Jiaotong University (SWJTU), Chengdu, China. His current research interests include resource management and protocol optimization in the next generation cellular networks, connected vehicle networks, wireless communications for railway, and massive IoT networks. Dr. Liu has coauthored more than 40 technical papers in international journals and conference proceedings. He won the Excellent Doctoral Dissertation Award of BUPT in 2015, the Best Paper Award in IEEE ICC'2014, and the Second Prize in National Undergraduate Electronic Design Contest of China in 2009. He is currently the Secretary and Treasurer for IEEE ComSoc, Chengdu Chapter. He was a Reviewers/TPC members for numerous journals and conferences, such as IEEE Journal on Selected Areas in Communications, IEEE Transaction on Wireless Communications, IEEE Transaction on Communications, IEEE Transaction on Vehicular Technology, IEEE Transaction on Green Communications and Networking, IEEE Communications Letters, IEEE Access, Digital Signal Processing, Wireless Networks, International Journal of Communication Systems, China Communications, KSII Transactions on Internet and Information Systems, IEEE the International Conference on Communications, IEEE Global Communications Conference, IEEE Wireless Communications and Networking Conference and so on.

Bole Wilfried TIENIN (Student Member, IEEE) completed his Bachelor's degree in Computer Engineering, at Bobo-Dioulasso Polytechnic University in Burkina Faso, in 2013. He received his Master's degree in Computer Engineering at Cukurova University in 2018. He is currently pursuing a Ph.D. degree at the University of Electronic Science and Technology of China (UESTC), majoring in Information and Communication Engineering. His research interests include Deep Learning, Radar detection, and Synthetic Aperture Radar (SAR) images.

Zheng Ma (Member, IEEE) is currently a Professor with the School of Information Science and Technology, Southwest Jiaotong University, China. He has authored or coauthored more than 120 research papers in high quality journals and conferences. He was an area Editor of IEEE COMMUNICATIONS LETTERS, and the Leading Editor of IEEE INTERNET OF THINGS JOURNAL Special Issue "Low-latency communications for IoT". He was the recipient of the Marie Curie Individual Fellowship in 2018. He is also with Division of Information Science and Engineering, KTH, Sweden.