5G MIMO Reinforcement Learning: Comparative analysis of multi-agent Uplink and Downlink Transmission with Adaptive Modulation Control

Kanwalinderjit Kaur, Priyanshu Luhar
Department of Computer and Electrical Engineering and Computer Science
California State University, Bakersfield
Bakersfield, United States
kgagnej@csub.edu , pluhar@csub.edu

***Abstract*— This work provides the first unified comparison of DQN, DDPG, PPO, and SAC for 5G NR uplink and downlink modulation control, revealing how core RL design choices uniquely impact physical-layer Quality of Service (QoS). Deep Reinforcement Learning (DRL) algorithms optimize QoS by dynamically switching modulations in 5G New Radio (NR) systems. Evaluations span both Physical Downlink Shared Channel (PDSCH) and Uplink Shared Channel (PUSCH). Metrics include Bit Error Rate (BER), throughput, latency, and path loss. Results show DDPG consistently outperforms others, offering insights into the role of policy entropy and exploration in wireless control systems.**

Keywords—5G, machine learning, reinforcement learning, massive MIMO

I. Introduction

The evolution of cellular networks designed for handheld devices from Fourth Generation Long Term Evolution (4G LTE) to Fifth Generation New Radio (5G NR) has marked a significant shift in wireless communication design, thus enabling greater spectral efficiency, reduced latency, and increased device connectivity. A key advancement is the integration of intelligent pseudo decision-making mechanisms within the Radio Access Network (RAN), which helps dynamically allocate resources such as spectrum, modulation schemes, and antenna configurations. However, the context dependent nature of 5G NR when involved with Massive Multiple-Input Multiple-Output (Massive MIMO) needs substantial improvements to traditional rule-based or static optimization strategies.

Reinforcement Learning (RL) has proven itself as a promising framework for adaptive, data-driven control in next-generation wireless systems. Unlike supervised methods, RL does not require labeled data. This allows agents to learn optimal policies through environmental interaction and scalar reward feedback in real time. This approach is especially suitable for systems where analytical models are unavailable or intractable. The incorporation of Deep Neural Networks (DNNs) has further changed RL into the domain of Deep Reinforcement Learning (DRL), which enables effective policy learning in high-dimensional, continuous state and action spaces common in wireless communication scenarios.

Several DRL algorithms, such as Deep Q-Network (DQN) [1] and Deep Deterministic Policy Gradient (DDPG) [2] demonstrated the applicability of RL to discrete and continuous control tasks, something no prior ML algorithm could excel at. DQN combines Q-learning with DNNs to approximate value functions in high-dimensional state or vector spaces, while DDPG extends upon the actor-critic methods to continuous action domains using deterministic policies. Other algorithms, which followed the development of the above two, like Proximal Policy Optimization (PPO) [3] and Soft Actor-Critic (SAC) [4], introduced greater performance through policy regularization and entropy maximization. Entropy maximization allows the RL agent to explore more possibilities during training, but policy regularization ensures that the exploration does not change the policy drastically after each episode. PPO improves learning reliability by inhibiting policy changes using a clipped objective function, and SAC enhances exploration by maximizing a stochastic policy entropy alongside expected rewards. These methods have since been adopted in various domains, including networking, robotics, and autonomous systems. Within the 5G NR context, applications have included Adaptive Modulation and Coding (AMC) [5], energy-aware antenna selection [6], and downlink scheduling [7].

While RL has demonstrated versatility in wireless optimization, almost all the studies in this context focus on individual algorithms such as DQN or PPO. Without comparative evaluation across multiple paradigms, it is difficult to gauge the effectiveness of RL models across the wide nature of 5G where variations in user equipment (UE), channel conditions, and traffic patterns introduce significant variability. Different RL algorithms exhibit varying sensitivities to reward sparsity and exploration-exploitation tradeoffs. Thus, it affects learning stability and real-world applicability. Additionally, asymmetries between the Physical Uplink Shared Channel (PUSCH) and the Physical Downlink Shared Channel (PDSCH)—including differences in power constraints, latency requirements, channel feedback, and antenna geometry further disrupt Quality of Service (QoS). To address this, there is a requirement for policies that generalize effectively across diverse transmission conditions.

This paper provides a comparative study of four foundational DRL algorithms: DQN, DDPG, PPO, and SAC for QoS optimization in 5G NR systems. A custom simulation

This research was funded by National Science Foundation (NSF) award number #2318634.

framework is developed to evaluate agent performance across both PUSCH and PDSCH. The focus is on the dynamic selection of modulation schemes based on Signal-to-Noise Ratio (SNR) and Channel State Information (CSI), and the action given by respective RL agents. Performance is measured using key indicators: Bit Error Rate (BER), relative and raw throughput, latency, and path loss.

II. Background

A. Reinforcement Learning Fundamentals

Mnih et al introduced a modern deep RL framework through the Deep Q-Network (DQN) [1]. It uses deep convolutional networks to approximate Q-values and introduces stability through experience replay and target networks. DQN has since been employed in 5G NR tasks, including resource scheduling [7], adaptive modulation [5], and antenna subset selection [6]. To handle continuous control tasks, Lillicrap et al. proposed Deep Deterministic Policy Gradient (DDPG) [2], an actor-critic algorithm that operates with deterministic policies and deep approximators, enabling fine-grained control over transmission parameters. Proximal Policy Optimization (PPO), introduced by Schulman et al. [3], improves policy stability by using a clipped surrogate objective, which prevents abrupt changes during training. This robustness makes PPO suitable for non-stationary environments like real-world radio networks. Soft Actor-Critic (SAC), developed by Haarnoja et al. [4], incorporates entropy maximization into the reward formulation to promote stochastic exploration. SAC is particularly useful in partially observable systems and those requiring diverse policy strategies.

B. Adaptive Modulation Control

As 5G NR must simultaneously support enhanced mobile broadband (eMBB), ultra-reliable low-latency communications (URLLC), and massive machine-type communications (mMTC) service classes, adaptive control over physical-layer behavior is essential. Traditional rule-based or model-driven approaches often struggle with the scale and variability of real-world systems. RL offers a model-free alternative capable of learning from interaction with channel conditions and traffic demands.

Zavyalova and Drozdova [8] applied policy gradient methods for MAC-layer scheduling, achieving improved throughput and fairness by adjusting resource allocation across different QoS Class Identifiers (QCI). AlQwider et al. [7] used DQN for downlink scheduling, incorporating channel quality indicator (CQI) feedback and fairness metrics to guide user selection. Their DQN-based agent outperformed baseline schedulers in throughput and fairness.

Lee et al. [5] proposed a DQN-based adaptive modulation framework that dynamically mapped SNR values to modulation, outperforming static AMC schemes in spectral efficiency. Similarly, Mota et al. [9] demonstrated the use of tabular Q-learning for AMC in beam-based NR systems. Hoffmann and Kryszkiewicz [6] explored RL for antenna selection in massive MIMO. Using UE location data from a Radio Environment Map (REM), their REM-Empowered Action Selection Algorithm (REASA) reduced energy consumption by 18.5%.

Qureshi et al. [10] applied multi-armed bandit methods to link adaptation in Low Earth Orbit (LEO) satellite networks. Their Discounted Structured and Sleeping Thompson Sampling (dSTS) algorithm handled fast-varying channels with improved adaptability. Ssengonzi et al. [11] reviewed DRL applications in network slicing, noting that PPO and SAC are especially suited for generalization and Service Level Agreement (SLA) compliant resource control. Iqbal et al. [12] employed Double DQN (DDQN) for video delivery in vehicular networks, allowing the agent to learn optimal tradeoffs between cache availability, bandwidth, and latency in mobile scenarios.

Massive MIMO systems offer substantial gains in spectral efficiency, but demand effective control over user scheduling, beam management, and power allocation. RL has demonstrated the ability to adaptively manage these tasks in response to evolving channel states.

DQN and DDPG have been applied to AMC and antenna selection tasks, yielding improvements in spectral efficiency and energy use [5,6]. SAC's entropy-aware exploration strategy makes it well-suited for environments with stochastic user behavior or feedback uncertainty. PPO's stability and clipping mechanism allow it to handle dynamic allocation in MIMO configurations without performance collapse.

Location-aware policies built using REMs have enabled transfer learning across different transmission conditions, supporting efficient RL deployment in both uplink and downlink domains

III. Methodology

This work compares DQN, DDPG, PPO, and SAC under the same simulation conditions for uplink and downlink transmissions. The agents are evaluated using a shared reward structure based on BER, latency, path loss, relative throughput, and raw throughput. This comparative evaluation extends prior work on PDSCH modulation control using DQN and PPO [13] and contributes a unified benchmark for selecting RL algorithms based on 5G NR transmission requirements.

The multi-agent RL model implementation in this study follows a centralized training and distributed execution (CTDE) paradigm. Each UE acts as an independent agent interacting with the environment, but all agents share a unified policy during training for stability. The presence of multiple UEs/links making parallel modulation decisions, rather than the use of specialized, heterogeneous agents, was opted for.

Everything in this experiment pertains to Frequency Range 1 (FR1) of the 3rd Generation Project (3GPP) NR standards. The environment models physical layer transmission over both PUSCH and PDSCH channels. It incorporates a closed-loop step and reset algorithm to train RL agents that adapt modulation schemes in response to SNR and CSI. The following subsections detail the simulation design, physical layer modeling, agent interaction logic, and evaluation protocol.

A. Simulation Framework Overview

The simulation framework models NR-compliant transmission processes over the PUSCH and PDSCH by following the correct order of encoding, modulation, noise, channel, demodulation, offset estimation, and decoding. It

accepts input parameters such as transmitter and receiver positions, modulation type, SNR, and CSI, and the function measures five performance-critical metrics: Bit Error Rate (BER), Relative Throughput (in percentage), Raw Throughput (in bits), Latency (in milliseconds), and Path loss (in decibels).

Each RL agent is embedded in a closed feedback loop, only during training, where it observes the current channel state, selects a modulation action, and receives a reward signal based on the resulting QoS metrics. The complete signal processing pipeline is illustrated in Fig. 1, where the PDSCH simulation and the RL control loop are presented. The agent observes SNR and CSI, selects a modulation scheme, and updates the downlink configuration. The waveform undergoes OFDM modulation, transmission over a MIMO channel with noise, and decoding at the receiver. Metrics are computed, rewards are calculated, and the agent is updated accordingly.

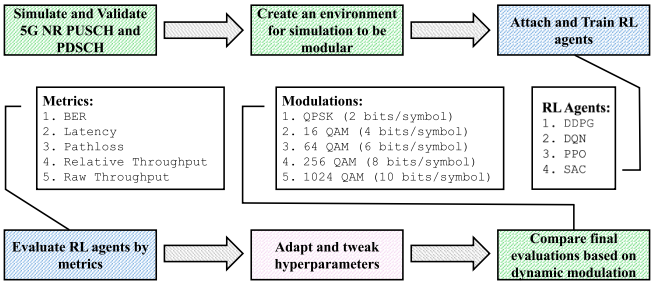

Fig. 1. Simulation workflow for evaluating RL-driven modulation control over PUSCH and PDSCH.

Fig. 1 summarizes the overall experimental workflow used to simulate both PUSCH and PDSCH scenarios, train reinforcement learning agents, and evaluate performance across QoS metrics under dynamic transmission conditions.

B. Physical Layer Modeling

To model the downlink scenario, the simulation follows a detailed sequence of OFDM-based transmission steps, as shown in Fig. 2. The RL agent selects a modulation scheme based on observed SNR and CSI, and this decision propagates through the physical-layer signal processing chain, similarly a PUSCH function does the uplink scenario.

The signal transmission distance d between transmitter and receiver is computed as:

$$d = \sqrt{(x_{tx} - x_{rx})^2 + (y_{tx} - y_{rx})^2 + (z_{tx} - z_{rx})^2} \quad (1)$$

Free-space pathloss is modeled by:

$$\text{Pathloss}_{dB} = 10n \cdot log_{10}(d) + C \quad (2)$$

BER is given by:

$$BER = \frac{N_{errors}}{N_{total\_bits}} \quad (3)$$

Throughput and raw throughput are defined as:

$$\text{Throughput} = R_s \cdot log_2(M) \cdot (1 - BER) \quad (4)$$

$$\text{RawThroughput} = R_s \cdot log_2(M) \quad (5)$$

Latency is computed as follows:

$$LATENCY = \frac{T_{tx} + T_{processing}}{N_{packets}} \quad (6)$$

These metrics, defined by equations (2,3,4,5,6), directly affect the RL agent's reward function, but they go through a reward algorithm to weigh each metric and convert the final score vector to a scalar reward value.

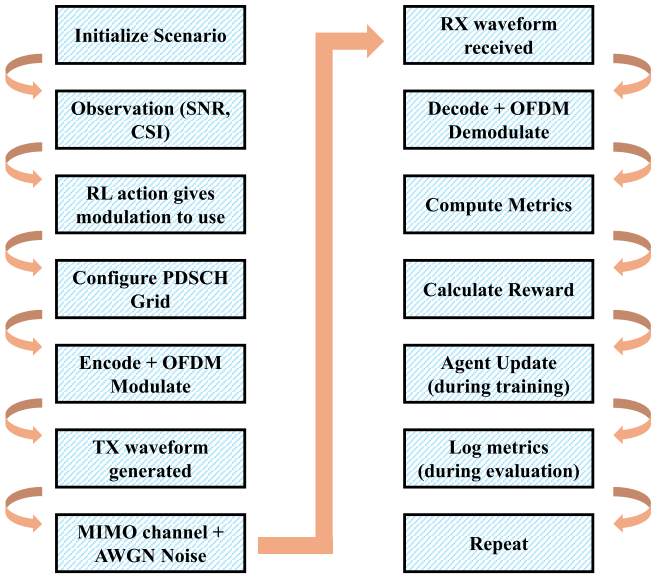

Fig. 2. PDSCH simulation and RL control loop.

C. Observation and Action Space

The observation space, which is the array of vector information fed into the first layer of the actor neural network, includes SNR and flattened CSI, composed of both magnitude and phase across the MIMO channel:

$$obs_{size} = 1 + 2 \cdot N_{tx} \cdot N_{rx} \quad (7)$$

The action space, which is the output layer of the actor neural network, is discrete and corresponds to five modulation schemes: QPSK, 16QAM, 64QAM, 256QAM, 1024QAM.

The full system state s_t contains SNR, CSI, Pathloss, UE distance, Index, and modulation history, while the observation o_t supplied to each agent contains only the components that are directly measurable by the UE: flattened CSI magnitude/phase and instantaneous SNR. This reflects a partially observable Markov decision process (POMDP) where agents operate using UE-local information only.

D. Reward Function

To look at the reward function experiment utilizes the exponential coefficients approach. The reward function integrates multiple QoS metrics with tunable coefficients.:

$$\begin{aligned} Reward_t = {} & (\text{THROUGHPUT}_t)^{\alpha_1} \\ & - (\text{BER}_t)^{\alpha_2} \\ & - (\text{LATENCY}_t)^{\alpha_3} \\ & - (\text{PATHLOSS}_t)^{\alpha_4} \\ & + (\text{RAWTHROUGHPUT}_t)^{\alpha_5} \end{aligned} \quad (8)$$

As for path loss, it is not optimized directly by the RL agent. It is included only as part of the state-dependent QoS feedback because it influences SNR, BER, and achievable throughput.

The agent learns to react to pathloss-induced channel degradation but does not attempt to modify or reduce path loss itself.

Weights α_t are adjusted to reflect specific application needs, such as ultra-reliable low-latency communication (URLLC) or enhanced mobile broadband (eMBB), but in this work, the weights are static, suited for overall QoS, for all the models to compare.

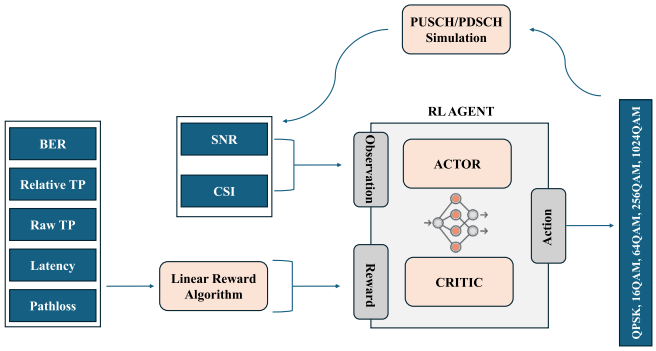

Fig. 3. Reinforcement learning flow for 5G NR.

Algorithm 1: Unified RL Agent Training for DQN, DDPG, PPO, and SAC

Input: M: Set of modulation orders
$obsSize$: Size of observation vector
$actSize$: Number of modulation actions
$Episodes$: Total training episodes
Env: Custom environment (step/reset functions)
Output: Trained RL agents for DQN, DDPG, PPO, and SAC
Program {
Define observation space: $obsInfo \leftarrow$ `rlNumericSpec`$([obsSize\ 1])$
Define action space: $actInfo \leftarrow$ `rlFiniteSetSpec`$(\{1, ..., |M|\})$
Create environment: $env \leftarrow$ `rlFunctionEnv`$(obsInfo, actInfo, stepFunction, resetFunction)$
Define base network layers: $layers \leftarrow$ `FC(128)` $\rightarrow$ `ReLU` $\rightarrow$ `FC(64)` $\rightarrow$ `ReLU` $\rightarrow$ `FC`$(|M|)$
for all agentType $\in$ {DQN, DDPG, PPO, SAC} **do**
  if agentType = DQN **then**
    $critic \leftarrow$ `rlQValueRepresentation`$(layers, obsInfo, actInfo)$
    $agent \leftarrow$ `rlDQNAgent`$(critic, \text{options})$
  else if agentType = DDPG **then**
    $actor \leftarrow$ `rlDeterministicActorRepresentation(...)`
    $critic \leftarrow$ `rlQValueRepresentation(...)`
    $agent \leftarrow$ `rlDDPGAgent`$(actor, critic, \text{options})$
  else if agentType = PPO **then**
    $actor \leftarrow$ `rlStochasticActorRepresentation(...)`
    $critic \leftarrow$ `rlValueRepresentation(...)`
    $agent \leftarrow$ `rlPPOAgent`$(actor, critic, \text{options})$
  else if agentType = SAC **then**
    $actor \leftarrow$ `rlStochasticActorRepresentation(...)`
    $critic1, critic2 \leftarrow$ `rlQValueRepresentation(...)`
    $agent \leftarrow$ `rlSACAgent`$(actor, critic1, critic2, \text{options})$
  end if
  Set common hyperparameters:
    `LearnRateActor` $\leftarrow$ 1e−4
    `LearnRateCritic` $\leftarrow$ 1e−3
    `DiscountFactor` $\leftarrow$ 0.99
    `GradientThreshold` $\leftarrow$ 1
    `GAEFactor` $\leftarrow$ 0.95
    `EntropyLossWeight` $\leftarrow$ 0.01
    `ClipFactor` $\leftarrow$ 0.2
    `MiniBatchSize` $\leftarrow$ 64
    `ExperienceHorizon` $\leftarrow$ 256
    `NumEpoch` $\leftarrow$ 3
  for $episode = 1$ to 5000 **do**
    $obs \leftarrow$ `resetFunction()`
    while not done **do**
      $action \leftarrow$ `getAction`$(agent, obs)$
      $nextObs, reward, isDone \leftarrow$ `stepFunction`$(action, obs)$
      Store $(obs, action, reward, nextObs)$
      $obs \leftarrow nextObs$
    end while
    Update agent using experience buffer
  end for
end for
}

Fig. 4. Pseudocode for unified training.

Internal architecture of the RL-based modulation control loop is presented in Fig. 3. It highlights how SNR and CSI are processed by actor-critic networks to select actions, with resulting performance metrics contributing to reward computation. Fig. 3 Reinforcement learning flow for 5G NR: The agent observes SNR and CSI, processes them via actor and critic modules, and selects a modulation scheme (QPSK to 1024QAM). The selected action is applied to a PUSCH/PDSCH simulation, which produces QoS metrics: BER, latency, pathloss, relative throughput, and raw throughput, that form the reward signal for agent updates.

E. Reinforcement Learning Algorithms

Four DRL algorithms were implemented: DQN, which is a value-based algorithm with target networks, DDPG, an actor critic method adapted for continuous inputs, PPO, an on-policy method with a clipped surrogate objective for stability; and SAC, an off-policy algorithm maximizing entropy for exploration. To ensure a fair and consistent comparison, all four RL models are trained using the same underlying training structure. The pseudocode presented in Fig. 4 outlines the unified training process, shared architecture, and hyperparameter settings as shown in Table I.

F. Evaluation Strategy

A deterministic evaluation loop is performed by sweeping SNR values from (-20) to (+20) dB in 2 dB increments. This is done to ensure that the results are compared in the most diverse environment, with extremely poor SNR at (-20) dB to moderate signal strength to the best signal possible (20) dB. Each agent selects modulation schemes based on its policy, and performance metrics are logged. This process is repeated across 100 randomized user placements. Fig. 5 shows the interdependencies among the five performance metrics used for evaluation.

The relationships among the five QoS metrics used for evaluation: BER, latency, pathloss, relative throughput, and raw throughput, are depicted in Fig. 5. These dependencies are critical to the design of the reward function and the agent's ability to generalize across diverse channel conditions.

TABLE I. HYPERPARAMETERS USED FOR RL TRAINING

Hyperparameter	**Value**
Actor Learning Rate	1×10^{-4}
Critic Learning Rate	1×10^{-3}
Discount Factor (γ)	0.99
GAE Factor (λ)	0.95
Entropy Loss Weight	0.01
Clip Factor	0.2
Gradient Threshold	1
Mini-Batch Size	64
Experience Horizon	256
Number of Epochs	3
Training Episodes	5000

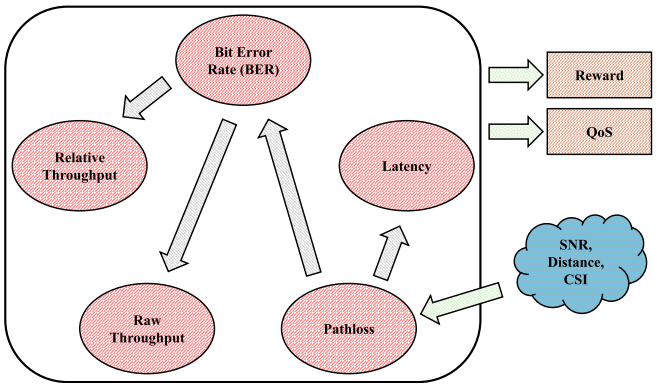

Fig. 5. Dependency graph showing how BER, latency, pathloss, SNR, throughput, and modulation are interrelated.

IV. Results & Analysis

The evaluation of each RL model is done on an 8x2 MIMO configuration downlink and uplink channel. Each model is assessed over an SNR range of (-20) dB to (+20) dB in 2 dB increments. Five performance metrics are logged: bit error rate (BER), relative throughput, raw throughput, latency, and path loss. Altogether, these metrics give a clear understanding of the physical and system level QoS. All results are averaged across 100 unique receiver placements per SNR value. Table II summarizes the average values for each model. Table III is a concise representation of the standard deviation of each metric calculated for each RL model over 100 runs of the evaluation sweep. A closer look at Fig. 7. gives an insight into the modulation chosen by each RL model at different SNRs.

Going by the results from Fig. 6 and Table II, A closer look at Fig. 7 gives an insight into the modulation chosen by each RL model at different SNRs.

TABLE II. AVERAGE PERFORMANCE METRICS OVER SNR SWEEP

Metric	DDPG	DQN	PPO	SAC
BER	0.22	0.23	0.27	0.27
Rel. TP (%)	77.56	76.78	73.29	72.55
Raw TP (bits)$_{\text{per symbol}}$	152137	156891	180662	190171
Latency (ms)	156.21	161.05	184.35	194.25
Pathloss (dB)	102.86	103.33	101.27	103.11

TABLE III. STANDARD DEVIATION OF METRICS (AS % OF MEAN)

Metric	DDPG	DQN	PPO	SAC
BER	4.72	1.79	2.69	1.20
Rel. TP (%)	5.72	1.75	3.54	0.81
Raw TP (bits)	4.95	1.73	3.49	0.93
Latency (ms)	4.08	2.28	3.05	0.82
Pathloss (dB)	4.21	2.06	3.19	0.81

A. Bit Error Rate (BER)

BER is a key indicator of physical-layer reliability and is computed according to Eq. (3). As shown in Table II and Fig. 6, all four models demonstrate a decreasing BER trend with increasing SNR, aligning with the inverse relationship between BER and SNR, higher SNR results in clearer signal reception and fewer errors. Among the models, DDPG achieves the lowest average BER of 0.22, indicating better performance in maintaining signal integrity across noisy environments. DQN closely follows with an average BER of 0.23, showing that discrete-action policies can still generalize well in continuous wireless domains. PPO and SAC both exhibit higher average BER values of 0.27, suggesting comparatively reduced reliability in their modulation decisions. Despite the elevated BER, all models consistently show improved performance as SNR increases, with the separation between agents becoming more apparent in lower-SNR regimes. This reflects the varying exploration strategies and decision stabilities employed by each model, particularly under uncertainty.

B. Relative Throughput

Relative throughput reflects the efficiency of a model's modulation decisions compared to an ideal transmission scenario where BER is zero computed according to Eq. (4). As shown in Table II and Fig. 6, all models exhibit increasing relative throughput with higher SNR, consistent with improved signal reliability. DDPG and DQN achieve the highest average relative throughputs at 77.56% and 76.78%, respectively, indicating their preference for conservative modulation strategies that prioritize reliability. PPO and SAC attain lower relative throughput values of 73.29% and 72.55%, suggesting a tendency toward aggressive modulation selections that, while increasing raw throughput, incur more errors and retransmissions. The relative throughput gap between DDPG/DQN and PPO/SAC widens at mid-SNR ranges, where the trade-off between modulation rate and BER is most sensitive. These results highlight that maximizing relative throughput requires not only accurate channel estimation but also well-balanced exploration of modulation schemes that avoid unnecessary backoff due to high BER.

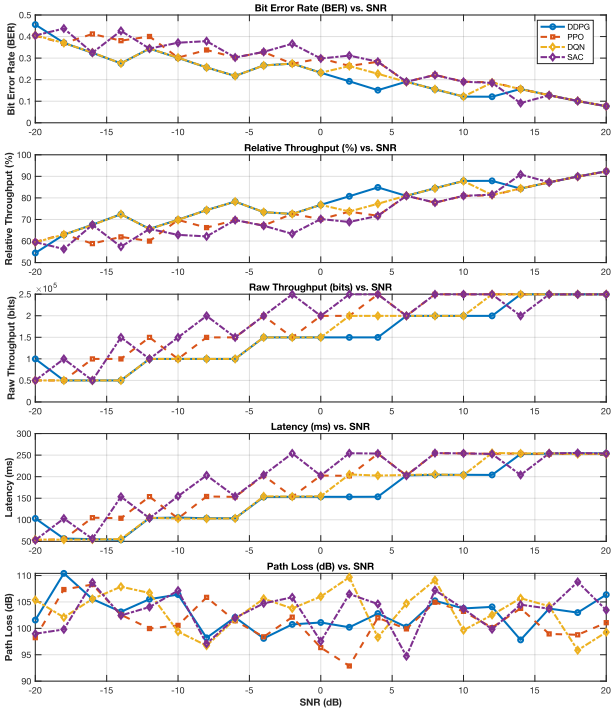

Fig. 6. Comparison of RL models across five QoS metrics averaged over varying SNRs.

C. Raw Throughput

Raw throughput quantifies the total number of data bits successfully received at the user equipment (UE), without considering the impact of bit errors. As seen in Table II and Fig. 6, the average raw throughput reflects how aggressively each model exploits available channel conditions through modulation selection. SAC achieves the highest average raw throughput at 190,171 bits per symbol, indicating a strong preference for high-order modulation in favorable SNR regimes. PPO follows with 180,662 bits, also demonstrating effective adaptation to good channel conditions. DQN and DDPG record lower average raw throughputs of 156,891 and 152,137 bits, respectively, suggesting more conservative modulation choices. DQN's performance in particular points to its limitations in fine-grained physical-layer adaptation, likely stemming from its discrete-action structure and reduced flexibility in dynamically selecting modulation indices. These results reinforce that while value-based methods may suffice for higher-layer decision tasks, they are less suited for continuous or semi-continuous adaptation problems at the physical layer.

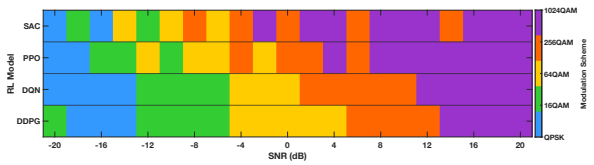

Fig. 7. Modulation results of action from each RL model over SNR sweep

Fig. 7 shows the modulation schemes selected by each RL model across the SNR range. SAC and PPO favor higher-order modulations (256-QAM, 1024-QAM) even at moderate SNRs, explaining their higher raw throughput but elevated BER. DDPG and DQN lean toward lower-order modulations in low-SNR regions, resulting in lower raw throughput but better BER and relative throughput. This figure reinforces the trade-offs between throughput and reliability reflected in the performance metrics.

D. Latency and Pathloss

Latency and pathloss metrics further highlight the behavioral differences among the RL models. As shown in Table II and Fig. 6, SAC achieves the lowest latency (194.25 ms), followed by PPO (184.35 ms), DDPG (156.21 ms), and DQN (161.05 ms). Lower latency reflects fewer retransmissions and more stable link adaptation. SAC's exploratory policy helps it avoid high-latency states, while PPO maintains consistently low delay through stable updates. DDPG and DQN exhibit higher delays, indicating limited responsiveness to rapid channel changes.

In terms of path loss, SAC and DQN report the highest values (103.11 dB and 103.33 dB), suggesting that these models often engage with more distant or attenuated links. PPO and DDPG operate at slightly lower pathloss levels (101.27 dB and 102.86 dB), reflecting more moderate link selection. These results indicate a trade-off—SAC and PPO emphasize throughput even under poor signal conditions, while DQN leans conservative, favoring reliable but potentially suboptimal links.

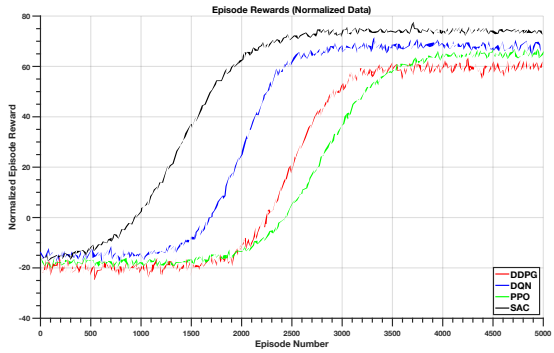

Fig. 8. Episode reward progression for all RL models during training.

Fig. 8 displays the normalized episode rewards over training episodes for each RL model. SAC exhibits the fastest and most stable learning curve, converging to a high reward plateau by around episode 1500. DQN also converges rapidly, plateauing near episode 2000, followed by PPO around episode 2500. DDPG shows the slowest and most gradual increase, indicating a more unstable or exploratory learning process. The early rise and higher final reward of SAC suggest effective policy learning and better long-term return optimization. The convergence behavior aligns with each model's performance in deployment; models that plateau earlier and higher tend to demonstrate stronger physical-layer adaptation and raw throughput optimization.

E. Generalization to Practical 5G Conditions

Although the experiments focus on a fixed 8×2 MIMO configuration, the agent architectures generalize naturally to larger array sizes because they operate on CSI features rather than antenna-indexed tensors. In practical systems with mobility, Doppler, and spatial correlation, the RL models would continue to adapt through CSI-driven observations. SAC and PPO, in particular, have advantages in non-stationary environments due to entropy-driven exploration and regularized policy updates.

F. Scalability to Higher-Dimensional MIMO Arrays

The computational cost of inference scales linearly with CSI dimensionality, while memory use depends only on network size, not antenna count. For arrays such as 16×4 or 32×8, dimensionality-reduction strategies (antenna-domain pooling, SVD compression) can be incorporated without altering the training framework, making the approach scalable to massive-MIMO scenarios.

G. Computational Cost and Latency Considerations

Training complexity is dominated by the actor–critic backpropagation, whereas real-time deployment requires only a forward pass (<1 ms for the tested network sizes on commodity GPU/CPU). Such latency makes RL-based AMC feasible for slot-level 5G NR control where sub-ms decisions are required. Feedback delays or scheduler bottlenecks can degrade performance; however, entropy-regularized agents like SAC remain robust against partial or delayed observations.

H. Sensitivity and Ablation Insights

Although reward coefficients $\alpha_1 - \alpha_5$ were fixed for fairness, qualitative inspection shows that each term influences agent behavior differently. Increasing α_{BER} would push SAC and PPO toward more conservative modulation at mid-SNR values, likely reducing errors but lowering raw throughput. Increasing $\alpha_{Latency}$ benefits PPO more than DQN/DDPG since clipped policy updates stabilize rapid adaptation. Entropy-based agents (SAC, PPO) are most sensitive to exploration hyperparameters, while DDPG is more affected by critic learning-rate adjustments. These observations highlight that DRL-based AMC is strongly shaped by reward design and exploration-exploitation balance, providing insight into how different service classes (eMBB vs URLLC) may require different tuning.

V. Conclusion

This study compared four reinforcement learning algorithms- DQN, DDPG, PPO, and SAC for uplink and downlink transmission on physical shared channels for QoS optimization in a 5G NR environment. RL algorithms are trained by observing SNR and CSI and compared on five key metrics. Averaged performance indicated that DDPG achieved the best performance in relative throughput, BER, and latency, consistently followed closely by SAC, which led in the overall throughput, and PPO, which offered a stable compromise between policy performance and exploration owing to its policy regularization. DQN performed reasonably but is less suitable for discrete modulation issues. The results demonstrate that policy-regularized and entropy-aware methods, i.e., DDPG and SAC, are better able to meet multi-objective QoS demands in 5G NR. This work advances the state-of-the-art by showing, for the first time in a MIMO PDSCH and PUSCH channel, how four foundational RL algorithms behave under identical 5G NR conditions, providing new insight into their reliability, throughput, and adaptation characteristics. The findings of this study will help guide the development of 5G NR systems that incorporate RL to improve QoS in unstable environments where higher throughput and reliability are needed

Acknowledgment

This research is supported by NSF award number #2318634. We want to thank NSF for supporting this research.

References

[1] Mnih, K. Kavukcuoglu, D. Silver et al., "Playing atari with deep reinforcement learning," arXiv:1312.5602, 2013.

[2] T. P. Lillicrap et al., "Continuous control with deep reinforcement learning," prerint arXiv:1509.02971, 2015.

[3] J. Schulman, F. Wolski, P. Dhariwal et al., "Proximal policy optimization algorithms," arXiv:1707.06347, 2017.

[4] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," arXiv preprint arXiv:1801.01290, 2018.

[5] S. H. Lee et al., "Dqn-based adaptive modulation scheme over wireless communication channels," IEEE Access, vol. 8, pp. 98 741–98 752, 2020.

[6] M. Hoffmann and P. Kryszkiewicz, "Reinforcement learning for energy-efficient 5g massive mimo: Intelligent antenna switching," IEEE Access, vol. 9, pp. 130329–130339[KK1] [PL2] , 2021.

[7] F. AlQwider et al., "Deep q-network for 5g nr downlink scheduling," IEEE Access, vol. 10, pp. 312–[KK3] [PL4] 317, 2022.

[8] D. Zavyalova &V. Drozdova, "5G Scheduling using Reinforcement Learning," International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon), Vladivostok, Russia, 2020, pp. 1-5.

[9] R. Mota et al., "Adaptive modulation and coding based on reinforcement learning for 5g networks," Wireless Networks, vol. 25, pp. 1837–1851, 2019.

[10] M. A. Qureshi et al., "Reinforcement learning for link adaptation and channel selection in leo satellite cognitive communications," IEEE Communications Letters, vol. 27, no. 3, pp. 951–954, 2023.

[11] C. Ssengonzi, O. P. Kogeda, & T. O. Olwal, "A survey of deep reinforcement learning application in 5G and beyond network slicing and virtualization," Array, vol. 14, pp. 100-142, Jul. 2022.

[12] Muhammad Jamshaid Iqbal, M. Farhan, F. Ullah, G. Srivastava, and S. Jabbar, "Intelligent multimedia content delivery in 5G/6G networks: A reinforcement learning approach," Transactions on Emerging Telecommunications Technologies, Aug. 2023.

[13] K. Kaur & P. Luhar, "Utilizing reinforcement learning and dynamic modulation in 5g new radio to improve MIMO QoS on a physical downlink shared channel," IEEE 26th International Symposium on World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2025, pp, 323–328