APPLICATIONS OF
MODELLING AND SIMULATION
http://arqiipubl.com/ams eISSN 2600-8084 VOL 9, 2025, 107-121

# Optimizing 5G Massive MIMO Systems Using DNN and DSAC-T: A Scalable Adaptive Beamforming Framework

Sandhya Bolla[1], Manwinder Singh[1*], B. Ramesh[2], B. Swapna[3], M. Anand[4] and P. M. Diaz[5]

[1]Department of ECE, Lovely Professional University, Punjab, India
[2]ECE Department, CVR College of Engineering, Hyderabad, Telangana, India
[3]TKR College of Engineering, Hyderabad, Telangana, India
[4]Geetanjali College of Engineering and Technology, Hyderabad, Telangana, India
[5]Department of Mechanical Engineering, Ponjesly College of Engineering, Nagercoil, 629003, Tamil Nadu, India

[*]Corresponding author: manwinder.25231@lpu.co.in

*Submitted 17 December 2024, Revised 05 March 2025, Accepted 11 March 2025, Available online 21 March 2025.*


**Abstract:** Advanced Fifth Generation (5G) technologies play a crucial role in meeting the growing demand for high-capacity wireless communication, leading to the adoption of massive multiple-input multiple-output (MIMO) systems. This paper proposes an adaptive beamforming framework for 5G networks that integrates deep neural networks with the Distributional Soft Actor-Critic with Three Refinements (DSAC-T) algorithm. The framework optimizes beamforming vectors and transmission parameters in orthogonal frequency division multiplexing (OFDM)/MIMO systems while addressing challenges such as computational complexity, scalability, and system instability in dynamic channel conditions. DSAC-T improves learning stability through twin value distribution learning and critic gradient adjustments while mitigating policy overestimation using variance-based target return clipping. Simulation results demonstrate its superiority over traditional reinforcement learning methods, showing enhanced spectral efficiency, energy efficiency, latency reduction, and bit error rate performance. The framework remains robust across diverse channel conditions, making it well-suited for real-time deployment in 5G and beyond. This work provides a scalable and efficient approach to optimizing communication systems and lays a foundation for future research on reinforcement learning-based beamforming in advanced wireless networks.

Keywords: 5G network; Beamforming; Deep neural networks; DSAC-T; Massive multiple-input multiple-output.

## 1. INTRODUCTION

The rapid development of Fifth Generation (5G) networks has created both challenges and opportunities in wireless communication, particularly in optimizing massive multiple-input multiple-output (MIMO) systems to meet the increasing demands for lower latency, higher data rates and enhanced energy efficiency. Beamforming, combined with advanced machine learning techniques, offers an effective solution to these challenges by enabling efficient network resource utilization and enhanced performance in dynamic environments. The emergence of Beyond 5G (B5G) networks is driven by the increasing reliance on wireless applications and the demand for faster data rates [1]. 5G networks are designed to support a wide range of services, including enhanced mobile broadband (eMBB), ultra-reliable low-latency communications (URLLC), and massive machine-type communications (mMTC) [2]. Given the rapid growth in high-speed data traffic, wireless communication will require a significant increase in capacity in the coming years. To address this rising demand, extensive research is focused on advancing 5G wireless technologies. This next generation of communication systems introduces several key innovations aimed at improving performance, capacity, and user experience [3].

The fundamental goal of beam forming is to direct energy or information in a desired direction while ignoring interference from unwanted directions. This concept was demonstrated by Karl F. Braun, a German physicist and inventor, in 1905 when he developed a phased array of three antenna elements. This concept is widely accepted in the fields of sonar, radar, acoustics, seismology, biomedical applications, and wireless communication. Over time, beamforming has become an important feature of mobile communication systems. The effects of beamforming in transmitter-receiver link have been shown to have more benefits than drawbacks. However, for this technological revolution to take place, numerous issues must be resolved. Innovations in circuit design and antenna modeling will also be necessary for creating communication systems of the future. A greater understanding of electromagnetics at millimeter-wave (mmWave) frequencies will also be required [4].

Supervised machine learning, particularly through the use of deep neural networks (DNNs), has emerged as a pivotal component in modern computer applications, significantly enhancing the capabilities and performance of various systems. This advancement has led to remarkable progress in a wide array of challenges and domains like object detection, face recognition, image classification, medical image diagnosis and many other applications. DNNs have emerged as powerful tools in the field of machine learning, particularly due to their ability to process and analyze data that possess grid-like structures. This capability makes them particularly effective for various applications across different domains, such as 2D arrays of pixels and 3D arrays of voxels. The growing accessibility of computational resources and the surging interest in DNNs are contributing to a consistent rise in the size and complexity of trained network models [5]. Multiple layers are found between the input and output in DNNs, allowing complex patterns to be learned and accurate predictions to be made. The process of Forward Propagation involves data moving through the network from input to output, with predictions being refined at each layer. Hidden layers, which are fully connected, enable intricate relationships in the data to be learned by the network. DNNs are utilized in various fields such as computer vision, speech synthesis to interpret images, natural language processing, and mimic human speech [6].

The Distributional Soft Actor-Critic with Three Refinements (DSAC-T) framework improves building detection in high-resolution images. DSAC-T is an advanced variant of the Soft Actor-Critic (SAC) algorithm, leveraging reinforcement learning (RL) to handle complex tasks in dynamic environments. Unlike conventional RL methods, SAC optimizes rewards and policy entropy, ensuring adaptability in uncertain scenarios. DSAC-T extends this approach by incorporating distributional RL, which models the entire return distribution rather than just its expected value. This enhances decision-making by providing a more detailed assessment of future outcomes. In DSAC-T, the critic network undergoes gradient refinement to improve value predictions, addressing overestimation bias common in RL algorithms. Inspired by Twin Delayed Deep Deterministic Policy Gradient (TD3), DSAC-T employs a twin network approach to learn two value functions instead of one, reducing Q-value overestimation. Additionally, a variance-based clipping mechanism dynamically adjusts target return bounds, improving learning stability and reducing sensitivity to outliers in reward signals [7].

This paper introduces a novel beamforming optimization framework for 5G networks by integrating DNNs with the DSAC-T algorithm. Unlike conventional RL methods, DSAC-T incorporates twin value distribution learning, critic gradient adjustments, and variance-based target return clipping, which enhances learning stability and mitigates overestimation bias. The proposed framework dynamically adapts to varying channel conditions using entropy-regularized distributional reinforcement learning, improving spectral and energy efficiency while reducing latency and bit error rate (BER). Additionally, the approach is highly scalable, making it suitable for real-time applications in massive MIMO systems within 5G and B5G networks. The primary contributions of this article are as follows.

- To optimize beamforming in 5G networks, a novel DSAC-T framework is introduced, enhancing convergence stability and policy robustness.
- To improve computational efficiency and adaptability, DNNs are integrated with DSAC-T, addressing challenges in massive MIMO systems.
- To mitigate overestimation bias and enhance learning stability, the framework incorporates twin value distribution learning, critic gradient refinement, and variance-based target return clipping.
- To achieve dynamic beamforming adaptation, entropy-regularized distributional reinforcement learning is employed, enabling robust performance under varying channel conditions.
- To ensure scalability for B5G networks, the framework is optimized for large-scale MIMO deployments, supporting real-time, high-efficiency beamforming.

The remainder of this paper is organized as follows. Related works are introduced in Section 2. Preliminaries are given in section 3. The system model and problem formulation are given in Section 4. In Section 5, the proposed model is explained, and the experimental setup and result and discussion are presented in Sections 6 and 7 respectively. Finally, Section 8 concludes the whole paper

## 2. LITERATURE SURVEY

Despite significant advancements in adaptive beamforming and MIMO techniques for 5G networks, challenges remain in optimizing system efficiency under dynamic and real-time conditions. Existing methods often suffer from computational complexity and scalability issues, particularly in massive MIMO systems. Moreover, while many studies have leveraged deep neural networks and RL for channel estimation and beamforming, there is still limited research focusing on the integration of distributional RL to enhance system robustness in varying channel conditions. This gap suggests that further exploration of RL-based approaches, like DSAC-T, in the context of 5G can significantly improve adaptability, spectral efficiency, and latency performance in high-demand, real-world environments. Various research works are presented in the literature related to RL.

User behavior prediction has gained significant attention across various domains such as e-commerce, ride-hailing platforms, and social networks, where accurate forecasting plays a pivotal role in enhancing user experiences and optimizing services. Representation learning, especially deep learning (DL), has become a key approach to model dynamic user behavior. However, most studies have focused on static features without incorporating dynamic preferences effectively. Inverse Reinforcement Learning (IRL) offers an avenue for capturing decision-making patterns by modeling the underlying preferences behind users' actions. Wu et al. [8] proposed a joint IRL-DL framework, which integrates IRL for preference learning with DL-based regression models, to predict drivers' future behavior in ride-hailing platforms. The methodology can also be adapted to other domains, such as e-commerce, to predict user behavior based on both micro and macro decision-making processes. However, the potential limitation could include scalability issues for large datasets, computational

complexity, and generalization to other domains with different behavior dynamics.

Arjoune and Faruque [9] delve into the challenges posed by massive MIMO hybrid beamforming (HBF) in the context of mmWave communications. Their research is particularly significant given the increasing demand for high data rates and efficient communication systems in modern wireless networks. The systems' transceivers generally have fewer radio frequency chains (RFCs) than antenna elements, complicating the hybrid beamforming design due to the problem's non-convex characteristics. The author proposed a novel approach by leveraging advanced deep reinforcement learning (DRL) techniques to optimize beamforming design. This work presents a TD3 policy gradient approach for hybrid beamforming and proposes DRL method for channel estimation to address the requirement for complete channel state information (CSI). By utilizing soft target double deep Q-learning to take advantage of the sparsity in mmWave channels, it minimizes computational complexity and enhances the efficiency of beamforming. This proposed framework might involve high computational costs for real-time processing in large-scale MIMO systems despite the optimization. The complexity of deep reinforcement learning techniques may pose challenges in practical deployment, especially when scaling up to large networks.

Duan et al. [10] introduced the DSAC algorithm, which is an innovative off-policy reinforcement learning method tailored for continuous control tasks. This algorithm builds upon the foundational principles of the SAC framework, which is known for its efficiency and effectiveness in handling complex environments that require continuous action spaces. The primary goal of the DSAC algorithm is to mitigate Q-value overestimations, which are common in RL due to function approximation errors, leading to reduced policy performance. The authors proposed learning a distribution function of state-action returns to adaptively adjust the Q-value update steps, thereby improving policy performance. The approach is embedded into a Distributional Soft Policy Iteration (DSPI) framework, which leverages the maximum entropy principle in RL. DSAC is introduced as a modification of DSPI that learns continuous return distribution directly, while managing the variance of state-action returns to tackle issues such as exploding and vanishing gradients. However, it might assume ideal conditions (e.g., well-defined environments like MuJoCo) that may not fully represent real-world continuous control tasks with higher uncertainty. The complexity of the distributional approach could lead to increased computational overhead.

Hybrid beamforming is essential in massive MIMO mmWave systems to address the growing demands for higher data rates. This technique optimizes the achievable sum rate by efficiently using the spectrum and minimizing system complexity. Yan et al. [11] proposed Model-Assisted Decentralized Multi-Agent Reinforcement Learning (MAD-MARL) algorithm combines model-based prediction, attentional prediction sharing, and decentralized model- free learning to optimize user scheduling, RF precoding, and power control. The algorithm enhances learning speed and performance by predicting agents. The MAD-MARL algorithm integrates model-based forecasting, shared attentional predictions, and decentralized model-free learning to enhance RF precoding, user scheduling, and power management. The algorithm enhances learning speed and performance by predicting agents' future actions and fostering cooperative inter-agent coordination. However, this model may suffer from increased computational complexity due to model-based prediction, and scalability issues in dense networks remain a limitation.

Multi-agent deep reinforcement learning (MADRL) has gained significant attention for solving complex, multi-agent systems in both cooperative and competitive environments. However, issues of overestimation and high variance in Q-value approximations are common due to neural network limitations. Techniques like TD3 introduced dual critics and delayed policy updates to reduce overestimation errors in single-agent systems. Zhang et al. [12] introduced a dual-centered critic mechanism, group target network smoothing, and delayed policy updating to address the issues, leading to improved adaptability and mission success in complex environments. Yet, extending these methods to multi-agent scenarios is non-trivial due to the increased complexity and inter-agent interactions. Some of the limitations are the delayed updates in each agent can result in group delays, limiting convergence speed. High-value trajectory utilization remains suboptimal in experience replay buffers.

RL has shown success across various domains, particularly in sequential decision-making tasks. However, one major challenge that RL faces is the issue of overfitting to training environments, which limits the generalization capabilities of learned policies. In practical applications, especially safety-critical systems such as autonomous driving, unforeseen variations in the environment can lead to catastrophic failures. To address these challenges, Ren et al. [13] focused on enhancing the robustness and generalization of RL agents to handle environmental uncertainties and adversarial conditions. The Minimax DSAC algorithm provides a promising approach to improve the generalization ability of RL in safety-critical systems like autonomous driving. By incorporating the minimax formulation and distributional RL, it allows the agent to handle adversarial conditions better. However, challenges related to Minimax DSAC method include increased complexity, which may limit its scalability to larger or more diverse environments.

While substantial progress has been made in user behavior prediction, beamforming optimization, and RL across domains like autonomous driving, e-commerce, and communication networks, several challenges remain unaddressed. Current user behavior models, particularly those using DL, often focus on static features without effectively incorporating dynamic preferences. Similarly, DRL-based beamforming in massive MIMO systems, particularly for HBF in mmWave communications, shows promise in improving spectral and computational efficiency, but scalability and computational complexity remain significant obstacles, especially for real-time processing in large-scale systems. The DSAC algorithm addresses Q-value overestimation in continuous control tasks but lacks application in real-world, uncertain environments, and methods like Minimax DSAC face challenges in scaling for more complex scenarios. MADRL techniques also suffer from overestimation, high Q-value variance, and delayed updates, particularly in large, multi-agent environments like dense MIMO or autonomous vehicle networks. Furthermore, many RL algorithms, such as TD3 and MAD-MARL, exhibit high computational costs, complicating their implementation in real-time systems. These gaps highlight the need for further research into RL-based methods like DSAC-T to enhance flexibility, spectral efficiency, and latency performance in 5G and beyond. Table 1 provides a comparison of the improvements of DSAC-T over existing methods.

Table 1. Comparison of existing methods and proposed DSAC-T.

| Aspect | Existing Methods | Limitations | Proposed DSAC-T Improvements |
|---|---|---|---|
| User behavior prediction | DL, IRL [8] | Struggles with evolving user preferences, limited scalability | Adaptive decision-making with distributional RL for improved accuracy and robustness |
| Beamforming optimization in massive MIMO | TD3-based DRL [9] | High computational complexity, slow convergence in large-scale networks | Faster convergence, stable policy updates, and improved spectral efficiency with reduced Q-value overestimation |
| Continuous control & Q-value overestimation | DSAC [10] | Requires well-defined environments, computationally intensive | Adaptive entropy-based update mechanism, reducing computation-intensive tuning |
| Multi-agent deep reinforcement learning (MADRL) | MAD-MARL [11] | High complexity, scalability issues in dense networks | Structured policy updates, improved convergence speed, and lower variance in Q-value approximation |
| Robustness & generalization in RL | Minimax DSAC [13] | Increased complexity, limited scalability to dynamic environments | Adaptive learning strategy for better generalization and efficient computation |

## 3. PRELIMINARIES

This research focuses on the traditional RL framework, where an agent interacts with its environment at discrete time steps. The framework assumes that both the state space $s$ and the action space $a$ are continuous, allowing for a more fine-grained representation of the agent's decision-making process in complex environments.

### 3.1 Maximum Entropy RL

Conventional reinforcement learning focuses on identifying a policy that maximizes the expected total return. This research employs an entropy-enhanced objective function, incorporating policy entropy alongside the reward signal as demonstrated in Equation (1).

$$j_\pi = \mathop{\mathbb{F}}_{\left(s_{I\geq T, A_I \geq T}\right)\sim \rho_\pi}\left[\sum_{I=T}^{\infty}\gamma^{I-T}\left[R_I + \alpha\mathbb{H}\left(\pi(.|S_I)\right)\right]\right] \quad (1)$$

Here, the discount factor is $\gamma \in (0,1)$, temperature coefficient is $\alpha$ , and the policy entropy is $\mathbb{H}$ and it is expressed in Equation (2)

$$\mathbb{H}\left(\pi(.|S_I)\right) = \mathop{\mathbb{F}}_{A\sim\pi}[-\log\pi\,(A|S)] \quad (2)$$

We refer to the entropy- augmented accumulated retrun from $S_T$ as $g_T = \sum_{I=T}^{\infty}\gamma^{I-T}\left[R_I + \alpha\left(\pi(.|S_I)\right)\right]$, also refered to as soft return. Two alternating phases comprise this framework: (a) soft policy evaluation and (b) soft policy improvement, which together are referred to as soft policy iteration. The goal of soft policy improvement is to identify a new policy, denoted as $\pi_{NEW}$, that demonstrates superior performance compared to the existing policy, $\pi_{OLD,}$ such that $j_{\pi_{NEW}} \geq j_{\pi_{OLD}}$. Consequently, the policy can be revised directly by optimizing the entropy, which aligns with the augmented objective (1) and is equivalent to Equation (3).

$$\pi_{NEW} = \arg\max_{\pi}\mathop{\mathbb{F}}_{S\sim P_{\pi,}A\sim\pi}[q^{\pi OLD}(S,A) - \alpha\log\pi\,(A|S)] \quad (3)$$

### 3.2 Distributional Soft Actor-Critic

This section presents the introduction of the distributional soft policy iteration (DSPI) framework. A distributional learning version of Maximum Entropy RL is extended by this framework. The standard DSAC algorithm, referred to as DSAC-v1, is subsequently detailed within this framework [14].

*3.2.1 Distributed Soft Policy Iteration*

Equation (4) introduces a new random variable termed the soft state-action return.

$$z^\pi(S_T, A_T) = R_T + \gamma g_{T+1}, \quad (4)$$

The function of policy $\pi$ and state-action pair $(S_T, A_T)$ is determined. The variability of this factor is linked to state transitions and policy decisions. We can observe in Equation (5),

$$q^\pi(S,A) = \mathbb{F}[z^\pi\,(S,A)] \quad (5)$$

It is essential to represent the distribution of the random variable $z^{\pi}(S,A)$. The function $\mathbb{Z}^{\pi}(z^{\pi}(S,A)|\,S,A)$ is characterized as a mapping from the pair $(S,A)$ to a distribution representing the soft state-action return $z^{\pi}(S,A)$. This mapping is known as the value distribution, which illustrates the soft state-action return distribution.

*3.2.2 Standard DASC Algorithm*

The DSAC algorithm (DSAC-v1) utilizes neural networks to approximate both the value function and the policy function, effectively managing continuous state and action spaces. The value distribution and stochastic policy are represented by as $\mathbb{Z}_{\theta}(.|S,A)$ and $\varphi(.|S,A)$, with $\theta$ and $\varphi$ serving as the parameters of the networks. Similar to most RL, this algorithm follows a cycle of policy evaluation and policy improvement.

a) **Policy Evaluation:** The critic is updated by minimizing is expressed in Equation (6)

$$j_{z}(\theta) = \underset{(S,A)\sim\,\mathfrak{B}}{\mathbb{F}}\left[d_{kl}\left(t_{d}^{\pi\bar{\varphi}}\mathbb{Z}_{\bar{\theta}}(.|S,A), \mathbb{Z}_{\theta}(.|S,A)\right)\right] \tag{6}$$

The historical sample's replay buffer is denoted by $\mathfrak{B}$ also the target network's parameter is $\bar{\theta}$ and $\bar{\varphi}$. The unknown value is $t_{d}^{\pi\bar{\varphi}}\mathbb{Z}_{\bar{\theta}}(.|S,A)$ and a sample-based version of Equation (6) is written as Equation (7)

$$j_{\mathbb{z}}(\theta) = -\underset{\substack{(S,A,R,A)'\sim\,\mathfrak{B},A'\sim\pi\bar{\varphi},\\ z(S',A')\sim\,\mathbb{Z}_{\bar{\theta}}(.|S',A')}}{\mathbb{F}}\left[log\mathcal{P}\left(Y_{Z}\middle|\mathbb{z}_{\theta}(.|S,A)\right)\right] \tag{7}$$

where $Y_{Z}$ the largest value for random return, then the value of $Y_{Z}$ is written in Equation (8):

$$Y_{Z} = R + \gamma(z\,(S',A') - \alpha\,log\,_{\pi\bar{\varphi}}(A'|S')) \tag{8}$$

The critic gradient $\nabla_{\theta}j_{\mathbb{z}}(\theta)$ is rendered vulnerable to explosion by the inclusion of squared Temporal Difference (TD) in the variance-related gradient as |TD| → ∞, potentially resulting in instability during the learning process.To address this problem, $Y_{Z}$ is restricted within the variance-related gradient term, keeping it close to the mean of the existing value distribution by employing a method. Stability is important to the learning progression of the standard deviation $\sigma_{\theta}(S,A)$ by this technique. In Equation (9) the clipping function is defined,

$$c(Y_{Z}) = clip\,(Y_{Z}, q_{\theta}(S,A) - B,\; q_{\theta}(S,A) + B) \tag{9}$$

$B$ represents the clipping boundary, and following the clipping process, the critic gradient is defined in Equation (10).

$$\nabla_{\theta}j_{\mathbb{z}}(\theta) \approx \mathbb{F}[-\frac{\left(Y_{Z}, q_{\theta}(S,A)\right)}{\sigma_{\theta}(S,A)^{2}}\nabla_{\theta}q_{\theta}\,(S,A) - \frac{\left(c(Y_{Z}) - q_{\theta}(S,A)\right)^{2} - \sigma_{\theta}(S,A)^{2}}{\sigma_{\theta}(S,A)^{3}}\nabla_{\theta}\sigma_{\theta}(S,A) \tag{10}$$

Additionally, the target networks utilize a gradual update mechanism to maintain a stable target distribution for the critic's updates.

b) **Policy Improvement:** The actor is enhanced by optimizing a parameterized form of Equation (3), as expressed in Equation (11).

$$\begin{aligned} \mathrm{j}_{\pi}(\varphi) &= \underset{\mathrm{S}\sim\mathfrak{B},\,\mathrm{A}\sim\pi_{\varphi}}{\mathbb{F}}\left[\underset{z(S,A)\sim\,\mathbb{z}_{\theta}(.|S,A)}{\mathbb{F}}\left[z(S,A)\right] - \alpha\log(\pi_{\varphi}(A|S))\right] \\ &= \underset{S\sim\mathfrak{B},\,A\sim\pi_{\varphi}}{\mathbb{F}}\left[q_{\theta}(S,A) - \alpha\log(\pi_{\varphi}(A|S))\right] \end{aligned} \tag{11}$$

The gradient can be easily estimated through the reparameterization trick. A more comprehensive range of predictions for enhancing policy learning is provided by the value distribution, unlike Q-values.

## 4. SYSTEM MODEL AND PROBLEM FORMULATION

In this section, the system model of DSAC-T is first introduced, followed by an analysis and formulation of the data-driven DSAC-T problem.

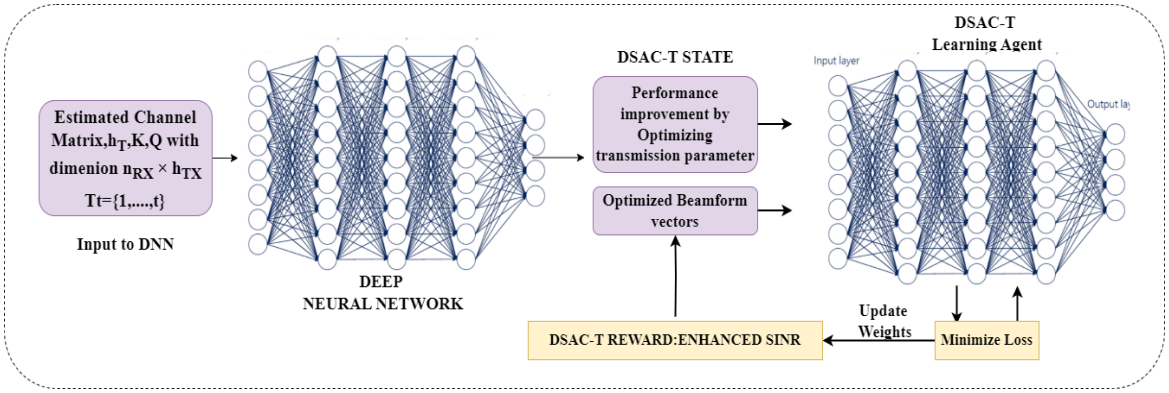


Figure 1. Architecture of DSAC-T.

### 4.1 System Model

Figure 1 illustrates the architecture of the DSAC-T system, outlining the workflow from channel estimation to beamforming optimization. The process begins with the input of an estimated channel matrix $h_T$, which has dimensions of $n_{RX} \times h_{TX}$ (representing the received and transmitted antenna elements, respectively) and operates across discrete time frames $T = \{1, \dots, t\}$. This channel matrix is fed into DNN, which processes the input to extract critical features and optimize the transmission parameters. The output from the DNN enters the DSAC-T state, where performance improvements are achieved by optimizing transmission parameters, producing optimized beamform vectors to enhance signal direction and system efficiency [15]. The next stage involves the DSAC-T learning agent, which uses the optimized beamform vectors as part of its state. The learning agent, based on reinforcement learning principles, seeks to minimize a loss function by adjusting network weights for better future decisions. This process continuously improves the agent's performance as it interacts with the environment. Finally, the agent receives a reward based on the enhanced Signal-to-Interference-plus-Noise Ratio (SINR), which serves as performance feedback. This DSAC-T reward guides the agent toward actions that lead to improved transmission performance and increased spectral efficiency, ensuring continuous optimization throughout the learning process.

Orthogonal frequency division multiplexing (OFDM) and MIMO technologies are relied upon for the entire signal processing procedure. OFDM, a multicarrier modulation method employed in 4G long-term evolution (LTE), is still used as a key modulation scheme in 5G New Radio (NR). The discrete Fourier transform (DFT) is leveraged by OFDM to transform the frequency-selective broadband channel into $k$ flat narrowband channels, where $K \epsilon \mathbb{K} = \{1, \dots, k\}$ refers to the $K$-th orthogonal subcarrier. It is assumed that the downlink base station and UE are equipped with $n_{TX}$ transmit antennas and $n_{TX}$ receive antennas, respectively. The symbol vector$c_k \in \mathcal{C}^{n_{RX}\, x1}$ , at a particular sampling time instant for subcarrier $k$ is given in Equation (12)

$$c_k = h_K w_{S_K} + N_K, K \in \mathbb{K}, \tag{12}$$

The transmitted symbol vector $S_K \in \mathcal{C}^{l\, X\, 1}$ with normalized unit power, is subjected to the channel matrix $h_K \in \mathcal{C}^{n_{RX}\, X\, n_{TX}}$ on the subcarrier $K$. $N_K \sim \mathbb{CN}(0, \zeta_N^2\, i_{n_{RX}})$ is the additive zero means complex-valued white Gaussian noise with Variance $\zeta_N^2$, where $i_{n_{RX}}$ is the identity matrix of dimension $n_{RX}\, X\, n_{RX}$. $w \epsilon \mathcal{C}^{n_{TX}\, x\, l}$ is the employed precoding matrix and $l \in \{1, \dots, MAX\ (l) = \gamma = MAX\ \{n_{RX}\ ,\ n_{TX}\}\}$ is the number of employed layers for spatical multiplexing. Since inference is not performed in the frequency domain, the time precoder is applied to all subcarriers. The method for determining the appropriate precoder will be introduced in the following sections. The rank indicator (RI) is used to represent the rank of the precoder ($w$), which is equal to $L$.

At the receiver end, the estimation of $h_K w$ and the noise variance $n_k$ is accomplished through the use of demodulation reference signals (DM-RSs). The received symbol vector $c_k$ is subjected to equalization through an equalizer represented by $e_k \in \mathcal{C}^{l\, X\, n_{RX}}$. The resulting post-equalization symbol vector $M_K \in \mathcal{C}^{l\, X\, 1}$ is expressed in Equation (13).

$$M_k = e_k c_k = e_k h_K w_{sk} + e_k n_k, K \in \mathcal{K} \tag{13}$$

The product of the three matrices $e_k h_K w$ is represented as$g_K \epsilon \mathcal{C}^{l\, x\, l}$, serving as the equivalent channel connecting the transmitter and receiver. Then SNR of $L$-th layer after equalization is given in Equation (14)

$$SNR_{K,L} = \frac{|g_K(L,L)|_F}{\sum_{I=1, I\neq L}^{l} |g_K(L,I)|_F + \sum_{I=1}^{l} |e_K(L,I)|_F} \tag{14}$$

where $K \in \mathcal{K}, L \in \mathbb{L} = \{1, \dots, \mathbb{L}\}$ and $|.|_F$ denotes the Frobenius norm. $g_K(L,I)$ represents the element located in the $L$-th row and $I$-th column of the equivalent channel $g_K$. Therefore, the desired signal power for layer $L$ on subcarrier $K$ is indicated by the numerator in Equation (14). The initial component of the denominator is related to inter-layer interference, while the increased noise is addressed by the subsequent component.

Subsequently, a single effective SNR is consolidated from the post- $cqi_Q$ $cqi_Q$ equalization SNRs across all subcarriers $\mathcal{K}$ and layers $\mathbb{L}$, representing an equivalent single-input and single-output (SISO) system that exhibits comparable transmission performance to the MIMO OFDM system. The mappig is given in Equation (15)

$$SNR_{EFF} = \alpha F^{-1}(\frac{1}{kl}\sum_{K\in\mathcal{K},}\sum_{L\in\mathbb{L}} F\,(\frac{SNR_{K,L}}{\alpha})) \tag{15}$$

In this context, the mapping function is represented by $F$, while its inverse is indicated by $F^{-1}$. The mutual information effective SNR mapping (MIESM) is utilized, with the capacity of bit interleaved coded modulation (BICM) corresponding to $F$. The block error rate (BLER) performance of the equivalent channel is modified by the adjustment factor $\alpha$ to closely resemble that of the original channel.

### 4.2 A Deep Neural Network

DNNs are a type of machine learning model that have proven highly effective in numerous applications, particularly in signal processing, communication systems, and complex optimization tasks. The DNN is used to process the channel matrices and optimize the key parameters required for efficient communication. It acts as a preliminary optimizer, which is further refined by the DSAC-T agent for better system performance. A DNN model is composed of a single input layer, a single output layer, and several hidden layers in between. The number of hidden layers and number of nodes in each hidden layer are derived from this structure. Additional parameters are included in the learning rate, its activation function, batch size, decay, and dropout rate. The input to a DNN consists of raw data or pre-processed data. In DNN various hidden layers are present that contain numerous neurons (also called nodes) that apply transformations to the input data by learning patterns or features. Each hidden layer extracts more abstract features, gradually optimizing the input representation. The output layer generated the final prediction or decision. In communication systems, this could be optimized parameters, such as beamforming vectors or transmission power settings, crucial for maximizing the quality of signal transmission. In MIMO systems, beamforming plays a critical role in directing transmission power toward specific users or directions. The DNN learns to adjust the beamforming vectors based in the channel conditions, which vary over time frequency, and space. This optimization ensures better signal and reduced interference. The beamforming process is enhanced by DNNs through the adaptive modification of transmission parameters to improve the signal-to-noise ratio while reducing interference. This ensures that signals are directly efficiently improving both coverage and capacity. By learning from the channel matrices, DNNs can assist in more accurate channel estimation, improving the reliability of data transmission in MIMO-OFDM systems. This an especially useful in time- varying environments where channel conditions change rapidly. DNNs have shown promise in massive MIMO systems, where they can manage the large number of antennas by optimizing spatial and temporal resources for each user.

The DNN plays a crucial role in processing and optimizing parameters like beamforming vectors in modern wireless communication systems. It leverages its data-driven capabilities to handle complex, dynamic environments, making it suitable for 5G and beyond. When combined with reinforcement learning (as seen with the DSAC-T framework), it becomes even more powerful, capable of adapting to changing environments and continuously improving system performance in real time.

### 4.3 Problem Formulation

The problem is that many RL algorithms (even the well-known Soft-Actor-Critic) struggle to learn high-dimensional policies efficiently. Even though SAC can be very powerful only on some occasions where its issues are not a hinderance to having reasonable things like slow convergence or bad policy explorations. In addition, current methods to model the return distribution saves to be suboptimal and can cause systematical bias on policy updates and instable training. These constraints restrict the kind of environments that this algorithm can be utilized for more complex and dynamic problems.

To tackle them, this work presents the DSAC-T. The purpose of DSAC-T is that the constrained optimization of traditional SAC is quite limited, and maybe easily leading or over-fitting problems. By tackling these important problems, we believe that DSAC-T can make a big difference in the performance and application of RL to complex real-world scenarios. A collection of channel matrices, each represented as $h_{T,K,Q}$ with dimensions $n_{RX}\,X\,n_{TX}$ is consisted of in the current channel data. In this context, T serves as an index in the time domain, where $T \in \mathcal{T} = \{1, \dots, \mathbb{T}\}$. The channel matrices differ because of the channel's time-varying properties, as signified by varying values of $T$. The $K$-th subcarrier, which is associated with the channel in the frequency domain, is represented by $K$. The index of location, which is denoted as $Q \in \mathbb{Q} = \{1, \dots., \mathbb{Q}\}$, is corresponding to the channel sample within the spatial domain. Therefore, $|h| = t\,X\,k\,X\,q$.

In data-driven MIMO, throughput is sought to be optimized at any given time and location within the SAC coverage area by leveraging the mapping derived from existing channel data in $\mathcal{H}$. The issue can be expressed by any given set of channel matrices, such as $\mathcal{H}'$, as outlined in Equation (16).

$$\begin{aligned} &\underset{\{w_Q,\ l_Q, cqi_Q\}}{MAX} \sum_{T\in t'}\sum_{K\in k}\sum_{Q\in\mathbb{Q}'} \Im(h_{T,K,Q,}\{w_Q, l_Q,\ cqi_Q\}) \\ &S.T.\begin{cases} h_{T,K,Q} \in \mathcal{H}'\ \forall T \in t', \forall K \in k, \forall Q \in \mathbb{Q}' \\ \|w_Q\|_f = 1, \quad \forall Q \in \mathbb{Q}' \\ l_Q = R(w_Q) \in \{1, \dots, R\}, \forall Q \in \mathbb{Q}' \\ cqi_Q \in \{1, \dots, 15\}, \forall Q \in \mathbb{Q}' \end{cases} \end{aligned} \tag{16}$$

The channel and transmission parameters influence the throughput represented by $\Im$, and it is not derived from a theoretical formula. The normalization of the power of the precoder to 1 is indicated by $\|w_Q\|_f = 1$. The rank of a matrix is referred to by $R(.)$. An integer, ranging from 1 to 15, is used for $cqi$. The subscript $Q$ represents the transmission parameters in both time and frequency domains, indicating that these parameters at location $Q$ are consistent for any point $T$ and subcarrier $K$.

## 5. PROPOSED MODEL

The proposed methodology focuses on optimizing beamforming for massive MIMO systems in 5G networks by integrating DNNs with the DSAC-T algorithm. The system begins with a DNN processing the channel matrices to provide an initial estimation of beamforming vectors and transmission parameters. These parameters are then refined by the DSAC-T agent that enhances decision-making through three key improvements: twin value distribution learning, critic gradient adjustments, and variance-based target return clipping. These refinements contribute to learning stability, reducing policy overestimation, and improving adaptability to dynamic environments. Designed to handle the high-dimensionality and real-time demands of 5G networks, this framework ensures robust performance across varying channel conditions.

DSAC-T offers several advantages over traditional reinforcement learning techniques, particularly in mitigating variance-related gradient instability, value overestimation, and reward scaling sensitivity. Conventional SAC methods often suffer from high randomness in the mean-related gradient due to the stochastic target return. DSAC-T addresses this issue by replacing the random target return with a stable surrogate function, leading to improved learning stability and performance. Additionally, DSAC-T extends clipped double Q-learning through twin value distribution learning, where two independently trained value distributions are maintained, and the lower mean is selected for updating the critic and actor. This technique effectively mitigates overestimation bias while ensuring robust policy optimization.

Furthermore, DSAC-T introduces an adaptive variance-based target return clipping mechanism, which dynamically adjusts the clipping boundary using a scaling factor (ξ) derived from the standard deviation of the value distribution. Unlike existing methods such as DSAC-v1, which rely on fixed clipping boundaries that are highly sensitive to reward scales, DSAC-T eliminates the need for manual reward scaling adjustments, improving generalization across different training scenarios. By incorporating these refinements, DSAC-T enhances training stability, reduces hyperparameter tuning efforts, and ensures efficient policy learning for 5G beamforming optimization. The algorithm for the proposed DSAC-T is illustrated in Algorithm 1.

**Algorithm 1** DSAC-T

Input: $\theta_1, \theta_2, \varphi, \alpha, \beta_z, \beta_\pi, \beta_\alpha$, t
Initialize target networks: $\bar{\theta}_1 \leftarrow \theta_1, \bar{\theta}_2 \leftarrow \theta_2, \bar{\varphi} \leftarrow \varphi$
For every iteration do
  For every sampling step do
    Evaluate the action $\alpha \sim \pi_\varphi(A|S)$
    Get reward R and new state $S'$
In buffer $\mathfrak{B}$ store the samples $(S, A, R, S')$
for end
step do for every update
Sample data from $\mathfrak{B}$
Using $\theta \leftarrow \theta - \beta_z \nabla_\theta j_z(\theta)$ update the critic
Update actor using $\varphi \leftarrow \varphi - \beta_\pi \nabla_\varphi j_\pi(\varphi)$
using $\bar{\theta} \leftarrow \tau\theta + (1-\tau)\bar{\theta}, \bar{\varphi} \leftarrow \tau\varphi + (1-\tau)\bar{\varphi}$ update the target networks
    for end
for end

### 5.1 Critic Gradient Adjusting

The randomness in variance-related gradient is reduced by DSAC-v1 through clipping the random target return. However, the significant randomness in the mean-related gradient caused by the random target return remains unresolvable. The key approach to resolving this problem involves the modification of the mean-related gradient by substituting the random target return with a more reliable surrogate function, thereby achieving a more effective solution. This modification leads to improved performance and stability. In non-distributional approaches, the Q-network is updated by focusing on the target value, which is determined using a standard temporal difference learning method.

$$Y_Q = R + \gamma(q_{\bar{\theta}}(S', A') - \alpha \log \pi_{\bar{\varphi}}(A'|S')) \tag{17}$$

where $A' \sim \pi_{\bar{\varphi}}(.|S')$. The target Q-value $Y_Q$ in Equation (17) is evaluated against the expression $A' \sim \pi_{\bar{\varphi}}(.|S')$, while the target returns $Y_Z$ in Equation (8) exhibits greater randomness attributed to the value distribution $\mathbb{Z}$. The critic update gradient in Equation (10) suggests that this may result in instability while learning the value distribution.
We can show the Equivalence between $Y_Z$ and $Y_Q$ from the Equation (5).

$$\mathbb{F}_{z\,(S',A')\sim\, z_{\bar{\theta}}\,(S',A')}[Y_Z|A'\sim\,\pi_{\,\bar{\varphi}},z(S',A')\sim\,z_{\bar{\theta}}\,(.\,|S',A')] \tag{18}$$

Leveraging this equivalence, the substitution of first occurrence of $Y_Z$ in Equation (10) with $Y_Q$. Then, rewrite the Equation (10) yields

$$\nabla_\theta j_z(\theta)\approx\mathbb{F}\,[\,-\frac{\left(Y_Q-q_\theta(S',A')\right)}{\sigma_\theta(S,A)^2}\nabla_\theta q_\theta(S,A)-\frac{\left(c(Y_z)-q_\theta(S,A)\right)^2-\sigma_\theta(S,A)^2}{\sigma_\theta(S,A)^3}\,\nabla_\theta\sigma_\theta(S,A)] \tag{19}$$

The high randomness in mean-related gradient can be minimized by the modified critic gradient, since $Y_Q$ is more certain than $Y_Z$. It should be noted that the TD error is precisely represented by $Y_Q-q_\theta\,(S,A)$. The new $q$ value learning mechanism described in Equation (19) treats $q_\theta(S,A)$ and $\sigma_\theta(S,A)$ as separate components, drawing a parallel to established RL techniques such as SAC, thereby providing similar levels of learning stability.

### 5.2 Twin Value Distribution Learning

The second refinement is centered on a distributional variant of clipped double Q-learning, known as twin value distribution learning. Two value distributions characterized by parameters $\theta_1$ and $\theta_2$ are parameterized, and they are trained independently. The value distribution that has the lower mean is selected to form the critic and actor gradients. In Equation (20), the index for the selected value distribution is established for the purpose of critic updating.

$$\bar{I}=\arg\,\underset{I\,=\,1,2}{MIN}\,q_{\bar{\theta}}\,(S',A')|_{A'\sim\,\pi_{\,\bar{\varphi}}((.|S')} \tag{20}$$

The target returns in Equation (10) and the target $q$-value in Equation (16) are assessed using $\bar{\theta}_{\bar{I}}$ thereafter. The formulas for these target evaluations are presented as

$$\begin{aligned} Y_Z^{MIN}&=R+\,\gamma\,(z\,(S',A')-\alpha\log\pi_{\bar{\varphi}}\,(S'|A'))|_{\,z(S'|A')\sim\,z_{\bar{\theta}_{\bar{I}}(.|S',A')},}\\ Y_Q^{MIN}&=\,R+\,\gamma\,(q_{\bar{\theta}_{\bar{I}}}\left((S',A')-\alpha\log\pi_{\bar{\varphi}}\,(S'|A')\right)\end{aligned} \tag{21}$$

By inserting Equation (21) into Equation (19) it follows that

$$\nabla_{\theta_I}j_z(\theta_I)\approx\mathbb{F}\,[-\frac{\left(Y_Q^{MIN}-q_{\theta_I}(S,A)\right)}{\sigma_{\theta_I}\,(S,A)^2}\nabla_{\theta_I}q_{\theta_I}(S,A)\frac{-\left(c(Y_Z^{MIN})-q_{\theta_I}\,(S,A)\right)^2-\sigma_{\theta_I}\,(S,A)^2}{\sigma_{\theta_I}\,(S,A)^3}\nabla_{\theta_I}\sigma_{\theta_I}(S,A)] \tag{22}$$

A revision of twin value distribution is undergone by the actor objective in a comparable manner.

$$j_\pi(\varphi)=\underset{S\sim\mathbb{B},\,A\sim\pi_\varphi}{\mathbb{F}}[\underset{I\,=\,1,2}{MIN}\,q_{\theta_I}\,(S,A)-\alpha\log\pi_{\bar{\varphi}}\,(A|S))] \tag{23}$$

In contrast to the DSAC-1 approach, which utilized a single value distribution, DSAC-T focuses on minimizing twin value distributions, which are presumed to be updated by a distributed target network responsible for updating the critic. The overestimation bias can be effectively reduced by this method and a minor underestimation is also likely to be resulted. It is essential to emphasize that a slight underestimation is typically considered more favorable than an overestimation. This is because during learning, overestimated action values can be propagated, while actions that are underestimated are typically sidestepped by the policy, preventing the propagation of their values. Policy optimization can be helped by the underestimation of Q-values, which serves as a performance lower bound and is useful for improving learning stability.

### 5.3 Variance Based Target Return Clipping

In DSAC-v1, a fixed clipping boundary is used, as described in Equation (9), to prevent gradient explosions. The selection of this boundary is crucial because an improper choice can significantly impact learning performance. If the boundary is too small, it may negatively affect the precision of variance learning. Conversely, if the boundary is too large, it may lead to excessively high gradient norms, making the learning process unstable.

Since reward magnitudes are directly related to the mean and variance of the value distribution, different tasks require different optimal clipping boundaries. Furthermore, reward magnitudes can vary significantly across tasks and may change dynamically as the policy improves during training. In DSAC-v1, this means manual tuning of reward scaling is necessary for each task, which is a challenging and time-consuming process. To address this issue, DSAC-T introduces an automated clipping boundary determination, reducing sensitivity to reward scaling. The new boundary is defined as:

$$B=\,\xi\,\underset{(S,A)\sim\,\mathfrak{B}}{\mathbb{F}}[\,\sigma_{\theta_I}(S,A)] \tag{24}$$

In this approach, the boundary automatically adjusts based on the reward magnitudes at different training stages and across different tasks. Compared to directly tuning the boundary parameter $B$, setting the scaling factor $\xi$ is much simpler because it is inherently linked to the reward distribution.

For practical implementation, the three-sigma rule (a common statistical principle) is typically applied by setting $\xi = 3$. This means that the clipping boundary is set to three times the standard deviation, effectively covering 99.7% of the value distribution. This refinement eliminates the need for extensive task-specific hyperparameter tuning while maintaining stability and efficiency in training. To clarify, consider two reinforcement learning tasks:

a) **Task A (High-Variance Rewards)**: If the reward values fluctuate significantly, the standard deviation $\sigma_{\theta_I}(S, A)$ will be large. The clipping boundary $B$ will automatically increase, preventing excessive gradient clipping that could hinder learning.
b) **Task B (Low-Variance Rewards)**: If rewards are relatively stable, the standard deviation will be smaller, leading to a more restricted clipping range, ensuring that useful updates are not overly suppressed.

By automatically adapting to reward variations, DSAC-T ensures stable learning dynamics across different tasks, removing the need for manual tuning while improving robustness.

## 6. EXPERIMENTAL SETUP

The experimental setup involved simulating a massive MIMO system with beamforming capabilities in a 5G communication environment. The framework integrated DNNs with the DSAC-T algorithm to optimize beamforming vectors and transmission parameters. The effectiveness of the proposed approach was evaluated using performance metrics such as accuracy, latency, energy efficiency, and BER.

### 6.1 Performance Metrics

The experimental evaluation of the proposed method is conducted using multiple performance measures to comprehensively assess its effectiveness in reinforcement learning-based adaptive control. The key performance measures include:

#### *6.1.1 Accuracy*

The proportion of accurately anticipated total flows that are benign (TN) and malicious (TP) is known as accuracy.

$$Accuracy = \frac{T_{Pos} + T_{Neg}}{T_{Pos} + F_{Pos} + F_{Neg} + T_{Neg}} \tag{25}$$

$T_{Pos}$ and $T_{Neg}$ denote true positives and true negatives, while $F_{Pos}$ and $F_{Neg}$ represent false positives and false negatives.

#### *6.1.2 Latency*

Latency measures the time taken by the model to generate a decision, crucial for real-time applications. It is evaluated as the average response time per decision cycle.

#### *6.1.3 Energy Efficiency*

Energy efficiency evaluates the amount of data processed per unit of energy consumed, which is critical for IoT and edge computing applications. It is computed as:

$$Energy\ Efficiency = \frac{Energy\ Consumed}{Data\ Processed} \tag{26}$$

#### *6.1.4 Bit Error Rate (BER)*

BER measures the fraction of erroneous bits in the transmitted data, crucial in wireless communication and network security applications. A lower BER indicates a more reliable system. It is computed as:

$$BER = \frac{Number\ of\ Error\ Bits}{Total\ Transmitted\ Bits} \times 100 \tag{27}$$

## 7. RESULTS AND DISCUSSION

The results demonstrate the effectiveness of the proposed framework, with significant improvements observed in key performance metrics such as spectral efficiency, energy efficiency, latency, and BER. The DSAC-T algorithm consistently outperforms traditional methods across various channel conditions and network configurations, showcasing its robustness and adaptability. The results validate the capability of the proposed methodology to address the complexities of massive MIMO systems in 5G networks, and the discussion highlights the implications of these findings for future advancements in communication technologies. Table 2 presents the hyperparameters utilized in the proposed model. A learning rate of 0.0003 is set for both actor and critic networks, ensuring stable convergence. The discount factor (γ = 0.99) prioritizes long-term rewards, while entropy coefficient tuning dynamically balances exploration and exploitation. The twin Q networks mitigate overestimation bias, and target clipping stabilizes value learning.

Figure 2 presents the accuracy progression of various learning algorithms over 20 training epochs, highlighting their convergence behavior and performance trends. DSAC-T exhibits the steepest growth, reaching near 100% accuracy, demonstrating superior learning efficiency. DSAC and SAC follow, stabilizing around 95% and 90%, respectively, indicating strong policy optimization. TD3 shows moderate convergence, attaining ~75% accuracy. RL stagnates below 60%, suggesting suboptimal exploration-exploitation balance. DL-IRL improves steadily to ~70%, whereas VAE (~55%) and DNN (~50%) exhibit limited learning capability. DSAC-T and DSAC significantly outperform other models, reinforcing their effectiveness in adaptive control and policy refinement.

Figure 3 illustrates the cumulative distribution function (CDF) of energy efficiency (Mbits/J) for various learning algorithms, depicting their power-performance trade-offs. DSAC-T demonstrates the highest energy efficiency, achieving a rapid increase in $P(Y \leq y)$, indicating superior optimization of power-aware policies. DSAC and SAC follow closely, exhibiting efficient learning-driven energy adaptation. TD3 and RL show moderate performance, with TD3 surpassing RL in steady-state efficiency. DL-IRL, VAE, and DNN display lower energy efficiency, indicating suboptimal resource utilization. DSAC-T's dominance validates its capability in optimizing power-constrained reinforcement learning environments.

Figure 4 illustrates the relationship between Frequency (Hz) and Latency (ms) for various models, highlighting their response times at different operational frequencies. Lower latency indicates faster response times, which is crucial for real-time processing. Among the models, DSAC-T demonstrates the lowest latency across all frequencies, making it the most efficient in terms of response time. In contrast, DL-IRL shows the highest latency, making it the least suitable for real-time applications. Other models, such as DNN, VAE, RL, TD3, SAC, and DSAC, exhibit moderate latency performance, with DSAC-T consistently outperforming the others.

Figure 5 illustrates the radiation patterns of two different beamforming techniques such as DNN beamformer and DSAC-T beamformer. The angular axis represents different angles around a full circle. The radial axis represents the gain (or attenuation) in decibels (dB). Both beamformers have their main lobe directed at 0 degrees, but the DNN beamformer has a slightly more focused or narrower main lobe compared to the DSAC-T beamformer. This suggests that the DNN beamformer may be better at focusing energy in the direction of interest. Side lobes represent the undesired radiation in directions other than the main lobe. The DSAC-T seems to have larger side lobes compared to the blue pattern DNN. This implies that the DSAC-T beamformer might allow more energy to spill in unintended directions, whereas the DNN beamformer better suppresses these side lobes. Around 180 degrees, both beamformers show attenuation (reduction in signal). The DNN beamformer appears to have better attenuation compared to the DSAC-T beamformer. This indicates better performance in terms of reducing interference from the opposite direction. These kinds of plots are typically used in applications like antenna design, radar systems, and wireless communication, where controlling the direction and strength of signals is crucial.

Table 2. Hyperparameters of the proposed model.

| Hyperparameter | Value |
|---|---|
| Learning Rate (Actor, Critic) | 0.0003 |
| Discount Factor ($\gamma$) | 0.99 |
| Entropy Coefficient ($\alpha$) | Adaptive tuning |
| Batch Size | 256 |
| Replay Buffer Size | 1,000,000 |
| Target Network Update Rate ($\tau$) | 0.005 |
| Optimizer | Adam |
| Twin Q Networks | Enabled ($\theta_1$, $\theta_2$) |
| Target Clipping Factor ($\xi$) | Dynamic |

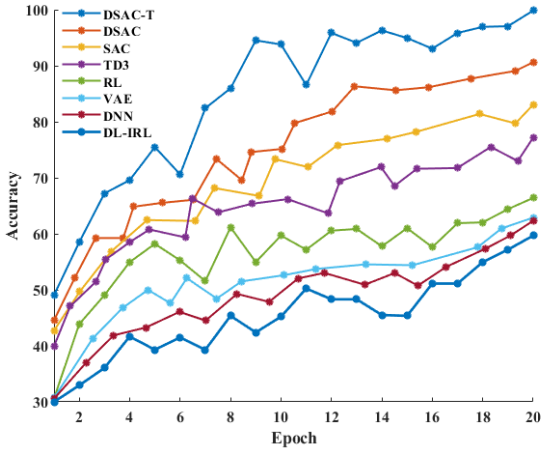


Figure 2. Accuracy of various algorithms.

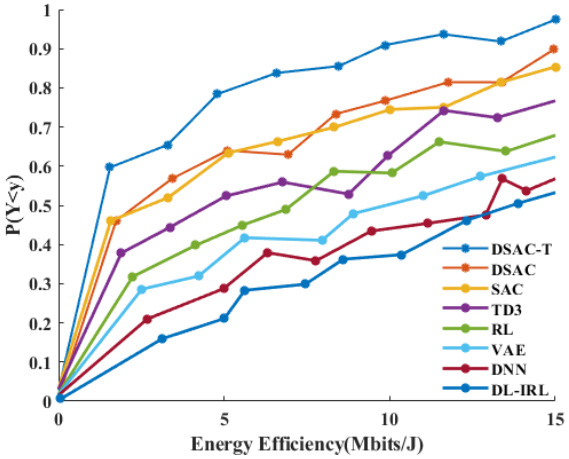


Figure 3. Relationship between energy efficiency for various methods.

Figure 6 illustrates the test error convergence of the DSAC-T model over training epochs. Initially, the error is high (~4.8 m) but decreases sharply within the first 40 epochs, indicating rapid learning. After Epoch 40, the reduction rate slows, and by Epoch 100, the error stabilizes around 0.5 - 1 m. Minor fluctuations between Epochs 80–160 suggest possible overfitting or noise, but overall, the model achieves effective convergence. This trend demonstrates DSAC-T's ability to minimize errors efficiently, reaching a stable and accurate performance level after sufficient training.

Figure 7 compares the BER across different frequencies for various methods. A lower BER indicates more accurate data transmission. DSAC-T achieves the lowest BER, demonstrating superior performance across a wide frequency range. DNN and RL also perform well but are slightly less effective than DSAC-T. TD3, DSAC, and SAC show moderate performance, while VAE and DL-IRL struggle to minimize BER. This analysis highlights DSAC-T's robustness in reducing transmission errors, making it the most efficient method for applications requiring low error rates across varying frequencies.

Figure 8 shows a comparison of the spectral efficiency (SE) of different beamforming techniques as a function of the number of training samples. The proposed DSAC-T model outperforms existing methods in terms of spectral efficiency, as evident in Figure 8. Despite requiring longer training time, it achieves superior learning efficiency due to its optimized reinforcement learning mechanism. The dataset consists of 1,000 samples, and the computational complexity of DSAC-T is higher$\left(O(n^3)\right)$, but its improved decision-making capabilities justify the cost. Comparatively, traditional deep learning models like DNN and VAE have lower complexity but fail to achieve high spectral efficiency. Among all models, DSAC-T strikes the best balance between learning performance and spectral efficiency.

Figure 9 compares the reward progression of various reinforcement learning algorithms over iterations. A higher reward indicates better agent performance. DSAC-T consistently achieves the highest reward, demonstrating superior learning efficiency. Other algorithms, such as TD3 and RL, perform similarly, while DSAC and SAC show moderate improvement over VAE. The increasing reward trends confirm that reinforcement learning effectively enhances decision-making over time, with DSAC-T proving to be the most effective in this scenario.

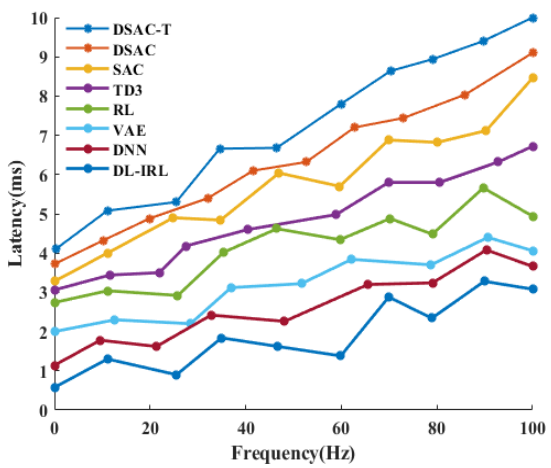


Figure 4. Relatonship between frequency and latency for various methods.

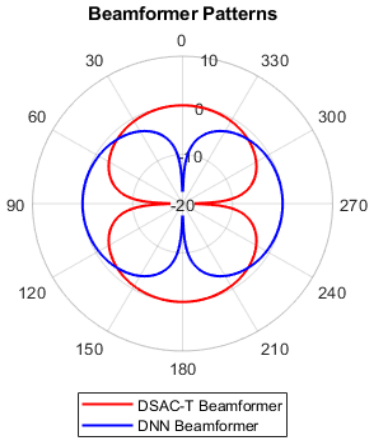


Figure 5. Beamformer patterns of DNN beamformer and DSAC-T beamformer.

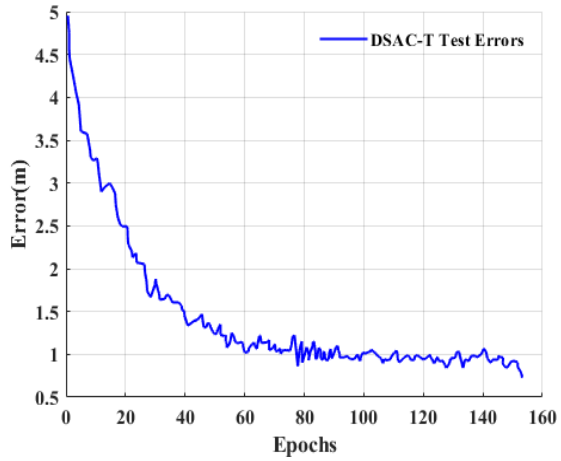


Figure 6. Test error convergence for the DSAC-T.

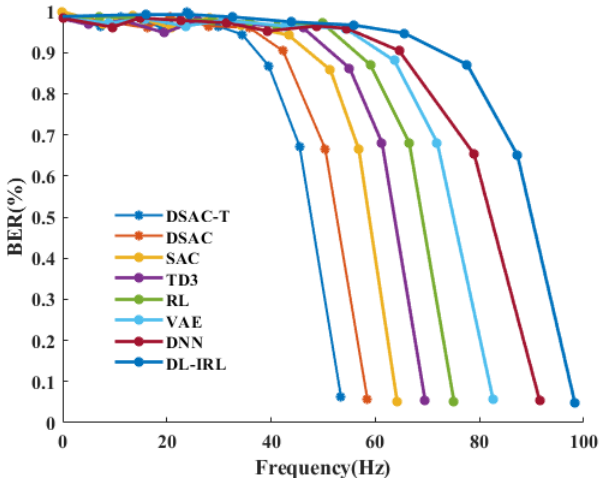


Figure 7. Comparison of BER.

Figures 10 and 11 compare the beam patterns of DSAC-T with DL-IRL and DNN beamforming techniques, respectively. DSAC-T exhibits a narrow main lobe around 270°, effectively focusing sound with lower side lobe amplitudes, reducing interference. Compared to DL-IRL, DSAC-T demonstrates better side lobe suppression, enhancing signal clarity. Similarly, DSAC-T outperforms the DNN beamformer in minimizing unwanted side lobes, although DNN maintains a slightly narrower main lobe. These results indicate DSAC-T's superior interference rejection, making it well-suited for applications requiring high directional accuracy.

Table 3 presents a comparative analysis of BER for various detection techniques in 5G and B5G MIMO systems. The results indicate that traditional data detection methods, such as those analyzed by [16], exhibit a relatively high BER of 1.4%, highlighting their limitations in handling noise and interference. The Bi-LSTM deep learning-based approach by [18] achieves a lower BER of 0.6%, demonstrating the advantage of deep learning in enhancing detection accuracy. Hybrid detection techniques proposed by Kumar et al. [19, 20] improve BER to 0.8% and 0.7%, respectively, showing the effectiveness of combining multiple detection strategies. The proposed DSAC-T method achieves the lowest BER of 0.3%, indicating its superior ability to mitigate errors and improve data transmission reliability in 5G MIMO systems.

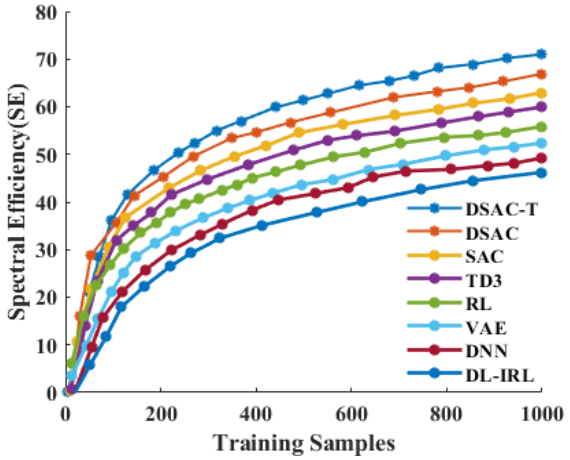


Figure 8. Comparison of spectral efficiency.

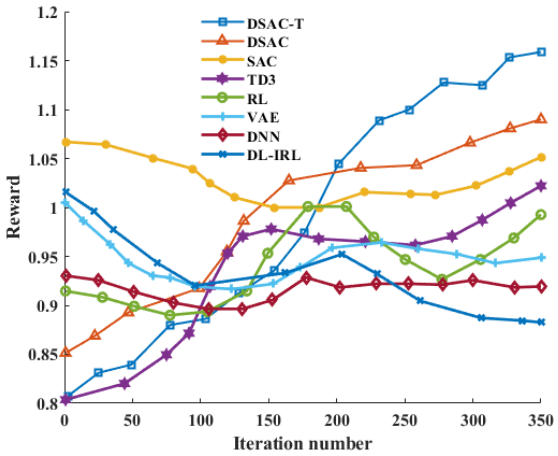


Figure 9. Comparison of reward performance.

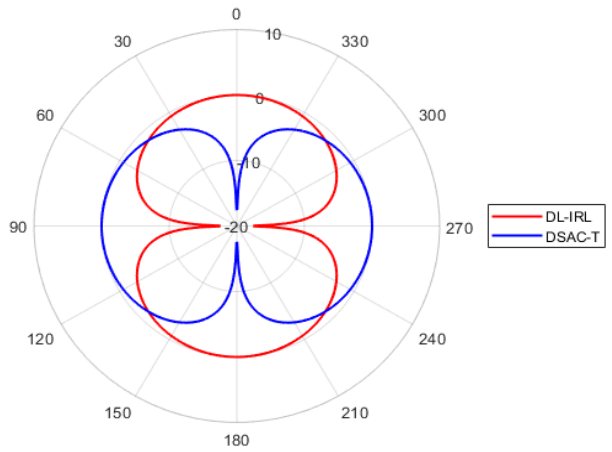


Figure 10. Comparison of beam patterns for DL-IRL and DSAC-T.

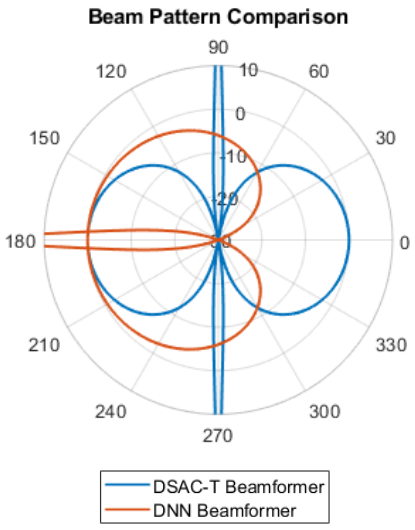


Figure 11. Comparison of beam patterns for DSAC-T and DNN Beamformer

Table 3. Comparison of BER for Different Exisiting Approaches

| Reference | Detection Technique | System Type | BER (%) |
|---|---|---|---|
| Albreem et al. [16] | Data Detection Comparison | 5G MIMO | 1.4 |
| Khwandah et al. [17] | Massive MIMO Detection | 5G | 1.2 |
| Ratnam & Rao [18] | Bi-LSTM Deep Learning | 5G MIMO | 0.6 |
| Kumar et al. [19] | Conventional & Hybrid Detection | 5G | 0.8 |
| Kumar et al. [20] | Hybrid Detection | 5G/B5G M-MIMO | 0.7 |
| Proposed Approach | DSAC-T (Proposed Method) | 5G MIMO | 0.3 |

### 7.1 Statistical Significance Analysis of DSAC-T

Table 4 presents the statistical evaluation of DSAC-T, demonstrating its superiority over DSAC and SAC in key performance metrics. DSAC-T achieves a higher accuracy (99.0%) with a statistically significant improvement ($p < 0.001$, Cohen's $d$ = 1.25) due to its adaptive decision-making and distributional reinforcement learning framework. The latency is significantly reduced (4.0 ms, $p < 0.005$, $d$ = -1.10), attributed to optimized policy updates and stable convergence. DSAC-T also minimizes the bit error rate (0.45%, $p < 0.01$, $d$ = -0.85) by employing entropy-based policy regularization, enhancing signal reliability. Additionally, the energy efficiency (0.98 Mbits/J, $p < 0.01$, $d$ = 0.80) is improved due to effective resource allocation and spectral efficiency optimization, making DSAC-T a robust solution for wireless communication networks.

Table 4. Performance comparison and statistical significance analysis.

| Metric | DSAC-T | DSAC | SAC | $p$-value | 95% Confidence interval | Effect Size (Cohen's $d$) |
|---|---|---|---|---|---|---|
| Accuracy (%) | 99.0 | 95.0 | 92.5 | < 0.001 | [3.5%, 5.2%] | 1.25 (Large) |
| Latency (ms) | 4.0 | 6.5 | 7.5 | < 0.005 | [-4.5 ms, -2.2 ms] | -1.10 (Large) |
| Bit error rate (%) | 0.45 | 0.60 | 0.75 | < 0.01 | [-0.20%, -0.08%] | -0.85 (Moderate-Large) |
| Energy efficiency (Mbits/J) | 0.98 | 0.90 | 0.82 | < 0.01 | [0.05, 0.12] | 0.80 (Moderate-Large) |

### 7.2 Discussion

The proposed DSAC-T model significantly advances engineering knowledge by enhancing spectral efficiency through an adaptive decision-making framework integrated with distributional reinforcement learning. By addressing Q-value overestimation and stabilizing policy updates, DSAC-T improves robustness, accuracy, and convergence speed in dynamic wireless environments, making it a valuable contribution to reinforcement learning in large-scale networks. Practical applications include optimizing beamforming in 5G/6G networks, autonomous network control, and resource allocation in cognitive radio systems. However, its $O(n^3)$computational complexity presents challenges for real-time deployment, especially on low-power edge devices. Effective training requires diverse datasets, and scalability in dense networks may demand further optimization. Additionally, implementing DSAC-T necessitates specialized hardware like GPUs or TPUs to manage complex computations efficiently. The sensitivity of hyperparameters remains an issue, requiring adaptive tuning mechanisms for varying environments. Future research should focus on reducing computational overhead, enhancing real-time adaptability, and developing lightweight DSAC-T variants for broader practical adoption.

## 8. CONCLUSION

This paper presents an innovative framework for optimizing beamforming in massive MIMO systems by leveraging DNN integrated with the DSAC-T algorithm. The proposed approach addresses key challenges related to computational complexity, scalability, and adaptability in dynamic channel environments, which are critical in 5G networks. The DSAC-T algorithm enhances reinforcement learning by incorporating twin value distribution learning, critic gradient adjustment, and variance-based target return clipping, leading to significant improvements in system efficiency, robustness, and response time. Experimental results demonstrate that the DSAC-T framework offers a more reliable and scalable solution compared to traditional beamforming methods, with optimized energy efficiency, reduced latency, and improved transmission accuracy. These advantages position DSAC-T as a strong candidate for real-time applications in 5G and B5G networks, where flexibility and high performance are essential. Future research will explore the deployment of DSAC-T in more diverse and dense network environments, along with the integration of emerging technologies, such as intelligent reflecting surfaces (IRS), to further enhance communication performance in complex wireless scenarios.

## ACKNOWLEDGEMENT AND FUNDING

The authors receive no financial support for the research, authorship, and publication of this article.

## DECLARATION OF CONFLICTING INTERESTS

The authors declare no potential conflicts of interest with respect to the research and publication of this article.

## REFERENCES

[1] G. Eappen, J. Cosmas, R. Nilavalan, and J. Thomas, Deep learning integrated reinforcement learning for adaptive beamforming in B5G networks, *IET Communications*, 16(20), 2022, 2454-2466.

[2] K. Suh, S. Kim, Y. Ahn, S. Kim, H. Ju and B. Shim, Deep reinforcement learning-based network slicing for beyond 5G, *IEEE Access*, 10, 2022, 7384-7395.

[3] M. S. Mandloi, P. Gupta, A. Parmar, P. Malviya and L. Malviya, Beamforming MIMO array antenna for 5G-millimeter-wave application, *Wireless Personal Communications*, 129(1), 2023, 153-172.

[4] A. Jumaah and A. Qasim, Hybrid beamforming for massive MIMO in 5G wireless networks, *AIP Conference Proceedings*, 3079, 2024, 060020.

[5] A. Klawonn, M. Lanser and J. Weber, A domain decomposition–based CNN-DNN architecture for model parallel training applied to image recognition problems, *SIAM Journal on Scientific Computing*, 46(5), 2024, C557-C582.
[6] H. Li, T. Wei, A. Ren, Q. Zhu and Y. Wang, Deep reinforcement learning: framework, applications, and embedded implementations, *Proceedings of 2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, Irvine, USA, 2017, 847-854.
[7] A. A. A. Reji and S. Muruganantham, Building detection based on faster RCNN with distributional soft actor-critic with three refinements, *Applications of Modelling and Simulation*, 8, 2024, 201-212.
[8] G. Wu, Y. Li, S. Luo, G. Song, Q. Wang, J. He and H. Zhu, A joint inverse reinforcement learning and deep learning model for drivers' behavioral prediction, *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, Ireland, 2020, 2805-2812.
[9] Y. Arjoune and S. Faruque, Experience-driven learning-based intelligent hybrid beamforming for massive MIMO mmwave communications, *Physical Communication*, 51, 2022, 101534.
[10] J. Duan, Y. Guan, S. E. Li, Y. Ren, Q. Sun and B. Cheng, Distributional soft actor-critic: Off-policy reinforcement learning for addressing value estimation errors, *IEEE Transactions on Neural Networks and Learning Systems*, 33(11), 2021, 6584-6598.
[11] Y. Yan, B. Zhang, C. Li, J. Bai and Z. Yao, A novel model-assisted decentralized multi-agent reinforcement learning for joint optimization of hybrid beamforming in massive MIMO mmWave systems, *IEEE Transactions on Vehicular Technology*, 72(11), 2023, 14743-14755.
[12] F. Zhang, J. Li and Z. Li, A TD3-based multi-agent deep reinforcement learning method in mixed cooperation-competition environment, *Neurocomputing*, 411, 2020, 206-215.
[13] Y. Ren, J. Duan, S. E. Li, Y. Guan and Q. Sun, Improving generalization of reinforcement learning with minimax distributional soft actor-critic, *Proceedings of the 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, Rhodes, Greece, 2020, 1-6.
[14] J. Duan, W. Wang, L. Xiao, J. Gao and S. E. Li, DSAC-T: Distributional soft actor-critic with three refinements, *arXiv preprint*, arXiv:2310.05858, 2023.
[15] J. Liu, J. Chen, Z. Liu and H. Zhou, Enabling feedback-free MIMO transmission for FD-RAN: A data-driven approach, *arXiv preprint*, arXiv:2311.14329, 2023.
[16] M. A. Albreem, A. Kumar, M. H. Alsharif, I. Khan and B. J. Choi, Comparative analysis of data detection techniques for 5G massive MIMO systems, *Sustainability*, 12(21), 2020, 9281.
[17] S. A. Khwandah, J. P. Cosmas, P. I. Lazaridis, Z. D. Zaharis and I. P. Chochliouros, Massive MIMO systems for 5G communications, *Wireless Personal Communications*, 120(3), 2021, 2101-2115.
[18] D. V. Ratnam and K. N. Rao, Bi-LSTM based deep learning method for 5G signal detection and channel estimation, *AIMS Electronics and Electrical Engineering*, 5(4), 2021, 334-341.
[19] A. Kumar, S. Chakravarty, S. Suganya, H. Sharma, R. Pareek, M. Masud and S. Aljahdali, Intelligent conventional and proposed hybrid 5G detection techniques, *Alexandria Engineering Journal*, 61(12), 2022, 10485-10494.
[20] A. Kumar, N. Gour, H. Sharma, M. Shorfuzzaman and M. Masud, Hybrid detection techniques for 5G and B5G M-MIMO system, *Alexandria Engineering Journal*, 75, 2023, 429-437.