DEEP REINFORCEMENT LEARNING-BASED RESOURCE ALLOCATION IN MASSIVE MIMO-NOMA SYSTEMS

Pham Hoai An[1,2], Nguyen Dung[1,2], Nguyen Thi Xuan Uyen[1,2],
Nguyen Thai Cong Nghia[1,2], Ngo Minh Nghia[1,2*]

[1]VNUHCM - University of Science, Ho Chi Minh City, Vietnam
[2]Vietnam National University, Ho Chi Minh City, Vietnam

ABSTRACT

Massive MIMO systems with preconfigured spatial beams efficiently serve near-field (NF) users, while far-field (FF) users can be multiplexed on the same beams using non-orthogonal multiple access (NOMA). To realistically capture propagation, the spherical wave model (SWM) is employed for NF channels and the plane wave model (PWM) for FF channels, reflecting the distinct near- and far-field regions. While conventional optimization approaches such as successive convex approximation (SCA) and branch-and-bound (BB) suffer from local optimality or prohibitive complexity, recent advances in deep learning have enabled scalable and adaptive solutions for wireless resource allocation. On this basis, a resource allocation strategy is developed using the Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm, where the base station acts as an agent that dynamically adjusts power and allocation coefficients to maximize the sum throughput of FF users. Simulation results show that the proposed DRL-based method can approach, and in some cases match, deterministic SCA at high SNR, while consistently outperforming randomly initialized SCA in medium-to-high SNR regimes. Compared to optimization-based baselines, the TD3 approach eliminates iterative problem reformulation, reduces computational complexity, and provides stronger adaptability to dynamic channels and user mobility.

KEYWORDS

Deep Reinforcement Learning, Massive MIMO, NOMA, Resource Allocation, TD3

1. INTRODUCTION

A recent advancement in non-orthogonal multiple access (NOMA) is its integration as a complementary technique into space-division multiple access (SDMA) networks, inherited from large-scale multiple-input multiple-output (MIMO) systems [1]. In this context, spatial beams are pre-configured to serve existing users and can further be reused to support additional users [2]. This reuse enhances connectivity and throughput with improved spectral efficiency and reduced computational complexity. For conventional SDMA networks operating in the far-field (FF) region—where the distance between transmitter and receiver exceeds the Rayleigh distance [3]—the application of NOMA is natural. In FF communications, beamforming is typically realized through steering vectors, forming cone-shaped beams [4]. Each beam covers a wide spatial region; thus, multiple users within the same coverage cone can share a beam via NOMA, making this approach both feasible and effective.

In contrast, near-field (NF) communications have gained increasing attention due to the use of higher carrier frequencies and large antenna arrays, which significantly extend the Rayleigh distance [5], [6]. Unlike FF systems, NF propagation requires the spherical-wave model (SWM) to capture distance-dependent phase variations. This motivates the application of beam-focusing techniques, where beams are concentrated not only in angular directions but also on specific

DOI: 10.5121/ijcnc.2025.17601

spatial locations [7]. From this perspective, a key challenge in hybrid NF-FF scenarios is to determine how pre-configured NF beams can be effectively reused to also accommodate FF users. Addressing this challenge serves as the primary motivation of this work.

Recent works have applied optimization-based approaches such as successive convex approximation (SCA) and branch-and-bound (BB) to MIMO-NOMA systems, but these methods suffer from high complexity, sensitivity to initialization, and poor adaptability under dynamic channel conditions. In contrast, deep reinforcement learning (DRL) offers a data-driven alternative. However, existing DRL studies mainly target far-field or discrete-action models, which are inadequate for continuous hybrid NF–FF beamforming and power control. To overcome these limitations, this work introduces a TD3-based framework that enables the base station to learn continuous allocation policies directly from channel feedback without explicit modelling. The main contributions of this paper are summarized as follows:

1) A novel NF-FF Massive MIMO-NOMA architecture is introduced, exploiting pre-configured NF beams to serve additional FF users.
2) A non-convex resource-allocation problem is formulated, aiming to maximize FF throughput while satisfying NF quality-of-service (QoS) constraints.
3) A DRL-based solution using the TD3 algorithm is proposed, enabling the BS to learn continuous allocation policies from system feedback without explicit channel models.
4) Extensive simulations under realistic NF-FF channel models validate the effectiveness of the proposed framework and highlight its advantages compared with conventional optimization methods.

The remainder of this paper is organized as follows. Section II reviews related work on Massive MIMO, NOMA, and reinforcement learning in wireless systems. Section III presents the system model and problem formulation. Section IV describes the proposed DRL-based resource-allocation framework. Section V provides numerical and simulation results. Section VI concludes the paper and discusses future research directions.

2. RELATED WORK

Recent studies have explored the use of machine learning (ML) and deep learning (DL) to address the high computational complexity of resource and power allocation in massive MIMO systems [8]. For example, [9] demonstrated that simple neural networks can approximate optimal power control with reduced complexity. Similarly, [10] highlighted both the effectiveness and vulnerability of DL-based power allocation, while [11] showed how DL-based optimization can enhance energy efficiency. These works confirm the potential of ML/DL in large-scale resource allocation but also reveal limitations in robustness and adaptability. Building on this trend, reinforcement learning (RL) has been increasingly adopted to tackle the dynamic nature of massive MIMO systems. [12] applied RL for adaptive scheduling, [13] used a DQN framework for beam and user grouping, and [14] employed actor-critic methods for balancing spectral and energy efficiency. These studies demonstrate the flexibility of RL beyond supervised DL approaches, particularly in handling time-varying channels and multi-objective trade-offs.

More specifically, recent work has applied RL directly to resource allocation in massive MIMO. [15] integrated DQN for joint user clustering, power allocation, and beamforming, while [16] introduced an actor-critic framework with pointer networks to reduce the action space complexity in scheduling and power allocation. Both approaches highlight the potential of RL to approximate near-optimal performance with significantly lower complexity. Despite these contributions, prior studies largely focus on either scheduling or power allocation in far-field conditions, often with simplified assumptions. Works such as [15] remain constrained by discretized action spaces, whereas [16] does not consider hybrid near-field and far-field propagation. To address these gaps, our work develops a Twin Delayed Deep Deterministic Policy Gradient (TD3)-based framework

for joint beam and power allocation in hybrid NF-FF massive MIMO-NOMA systems. By reusing pre-configured near-field beams for far-field users, the proposed approach ensures both scalability and efficiency, outperforming existing methods in dynamic scenarios. Recent studies such as [14] and [15] further improved DRL-based MIMO optimization but still focused on far-field scenarios. Unlike these works, our TD3-based framework jointly handles beam and power allocation under hybrid NF–FF propagation, offering stable learning with continuous action control.

3. SYSTEM MODEL

3.1. Massive MIMO

Considering the pure spectral efficiency according to Shannon's theory - which forms the basis for (3.1) and (3.2) - it can be shown that, in conventional Multiuser MIMO systems, choosing the number of transmit antennas M approximately equal to the number of users K is optimal: increasing M beyond that yields only logarithmic throughput growth, while the computational complexity increases linearly with M. Massive MIMO marks a significant departure from traditional Multiuser MIMO systems. Instead of attempting to operate close to the Shannon limit, it deliberately operates at a larger distance from that limit, but paradoxically achieves superior performance compared with any conventional Multiuser MIMO system.

Uplink capacity:

$$C^{\text{ul}} = \log_2 \left| \mathbf{I}_M + \rho_{\text{ul}} \mathbf{H}\mathbf{H}^{\text{H}} \right|. \tag{1}$$

Downlink capacity:

$$C^{\text{dl}} = \max_{\substack{\nu_k \geq 0 \\ \sum_{k=1}^{K} \nu_k \leq 1}} \log_2 \left| \mathbf{I}_M + \rho_{\text{dl}} \mathbf{H}\mathbf{D}_{\nu}\mathbf{H}^{\text{H}} \right|. \tag{2}$$

There are three fundamental differences between Massive MIMO and conventional Multiuser MIMO systems.

1) Only the base station (BS) learns the channel matrix $\mathbf{H}$.
2) The number of antennas at the BS M is typically much larger than the number of users K, although this is not strictly required.
3) Simple linear signal processing techniques are used for both uplink and downlink.

These properties enable Massive MIMO to scale effectively with the number of BS antennas M.

Figure 1 illustrates the basic configuration of a Massive MIMO system. Each BS is equipped with a large number of antennas M and serves a cell with many user equipments (UEs) K. UEs are usually equipped with a single antenna. Different BSs serve different cells, and except for mechanisms such as power control or pilot allocation, Massive MIMO systems do not employ cooperation between BSs. In both uplink and downlink, all UEs simultaneously use the entire time-frequency resources. On the uplink, the BS must recover the separate signals transmitted from UEs. On the downlink, the BS must ensure that each UE only receives its own intended signal. Multiplexing and demultiplexing at the BS are realized by exploiting the large number of antennas and the availability of channel state information (CSI).

Figure 2 illustrates how precoding operates under different propagation conditions. In line-of-sight (LoS) environments, the base station (BS) forms a dedicated narrow beam for each user, with beamwidth decreasing as the number of antennas increases. Under local scattering, the received signal results from the superposition of many reflected and scattered paths, which can combine constructively at the user location when the carrier is properly chosen. Larger antenna arrays thus enable more precise energy focusing.

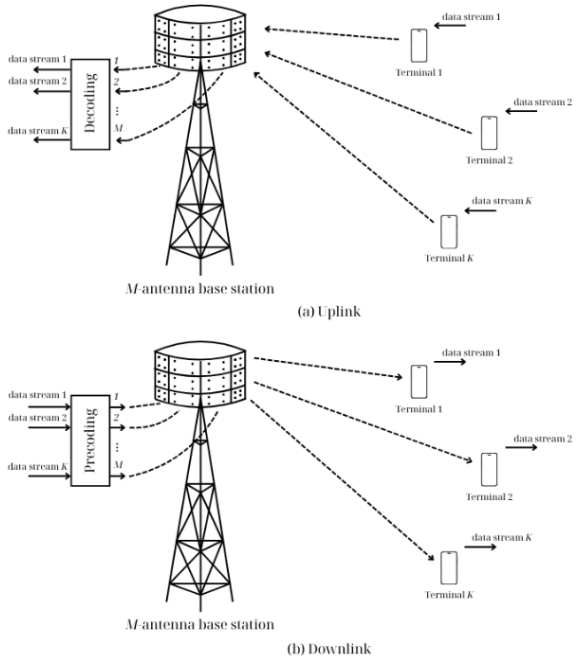

Figure 1. Massive MIMO systems

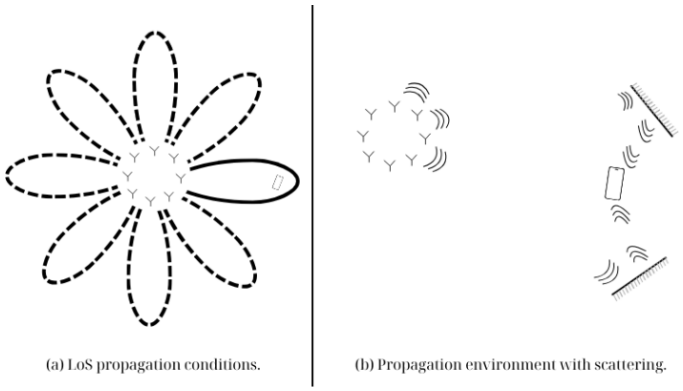

Figure 2. Effect of Precoding in Various Wireless Propagation Environments

Accurate channel state information (CSI) is crucial for such focusing. In time-division duplexing (TDD) systems, the BS estimates CSI from uplink pilots and exploits channel reciprocity, requiring only hardware calibration rather than absolute phase alignment. Increasing the number of antennas always reduces transmit power, allows simultaneous service to more users, and enhances spectral efficiency. Furthermore, Massive MIMO benefits from channel hardening, where small-scale fading effects vanish as M grows, making the effective channel approximate an AWGN channel. This property simplifies signal processing and allows standard modulation and coding schemes to be applied effectively.

3.2. Massive MIMO-NOMA with Coexisting Near-Field and Far-Field Communications

Consider a traditional SDMA downlink network where a base station (BS) employs a uniform linear array (ULA) with N antenna elements to serve M near-field (NF) users, each equipped with a single antenna, under the condition $M \leq N$. In this study, it is assumed that M spatial beamforming vectors, denoted as $\mathbf{p}_m$, are pre-configured to individually serve the NF users. The objective is to further support K far-field (FF) users by reusing these pre-configured NF beams.

Let the two-dimensional coordinates of the m-th NF user, the k-th FF user, the array center, and the n-th antenna element be denoted as $\boldsymbol{\psi}_m^{\mathrm{NF}}$, $\boldsymbol{\psi}_k^{\mathrm{FF}}$, $\boldsymbol{\psi}_0$, and $\boldsymbol{\psi}_n$, respectively. According to the NF condition, $|\boldsymbol{\psi}_m^{\mathrm{NF}} - \boldsymbol{\psi}_0| < d_R(N)$, whereas for FF communication, $|\boldsymbol{\psi}_k^{\mathrm{FF}} - \boldsymbol{\psi}_0| > d_R(N)$, with the Rayleigh distance defined as $d_R(N) = \dfrac{2(N-1)d^2}{\lambda}$, where λ is the wavelength and d is the spacing between adjacent antenna elements in the ULA, consistent with [6]. It should be noted that other antenna configurations, such as uniform planar arrays (UPA) or uniform circular arrays (UCA), can also be applied to support NF communication [4].

3.3. Near-Field and Far-Field Channel Models

The received signal at the m-th near-field (NF) user can be expressed as

$$y_m = \mathbf{h}_m^H \mathbf{x} + n_m, \tag{3}$$

where $\mathbf{x}$ denotes the transmit signal vector from the base station (BS), n_m is the additive Gaussian noise with power σ^2, and $\mathbf{h}_m$ is the channel vector under the spherical-wave assumption:

$$\mathbf{h}_m = \alpha_m \left[e^{-j\frac{2\pi}{\lambda}|\psi_m^{\mathrm{NF}} - \psi_1|} \quad \cdots \quad e^{-j\frac{2\pi}{\lambda}|\psi_m^{\mathrm{NF}} - \psi_N|} \right]^T, \tag{4}$$

with the path-loss coefficient

$$\alpha_m = \frac{c}{4\pi f_c |\psi_m^{\mathrm{NF}} - \psi_0|}, \tag{5}$$

where c is the speed of light, f_c is the carrier frequency, ψ_m^{NF} denotes the position of the m-th NF user, ψ_n is the position of the n-th antenna element, and ψ_0 is the array center. It is assumed that a line-of-sight (LoS) link is always available for NF users, since they are located within the Rayleigh distance of the BS.

The received signal at the k-th far-field (FF) user is given by

$$z_k = \mathbf{g}_k^H \mathbf{x} + w_k, \tag{6}$$

where w_k denotes the additive Gaussian noise (with the same power as n_m). The far-field channel vector is modeled using the conventional beamforming representation as

$$\mathbf{g}_k = \alpha_k e^{-j\frac{2\pi}{\lambda}|\psi_k^{\mathrm{FF}} - \psi_1|} \left[1 \quad e^{-j\frac{2\pi d}{\lambda}\sin\theta_k} \quad \cdots \quad e^{-j\frac{2\pi d}{\lambda}(N-1)\sin\theta_k} \right]^T, \tag{7}$$

where θ_k is the angle of departure (AoD) and α_k is the free-space path-loss coefficient. It is worth noting that the FF channel model is mainly characterized by the angular parameter θ_k, whereas the NF channel model strongly depends on the exact user location.

3.4. Near-Field Beamforming and NOMA Data Rate

The system employs full-digital beamforming based on zero-forcing (ZF) principle, defined as

$$\mathbf{P} \triangleq [\mathbf{p}_1, \ldots, \mathbf{p}_M] = \mathbf{H}(\mathbf{H}^H \mathbf{H})^{-1} \mathbf{Q}, \tag{8}$$

where $\mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_M]$ and $\mathbf{Q}$ is a diagonal $M \times M$ normalized matrix ensuring $\| \mathbf{H}^H \mathbf{p}_m \| = 1$.

The NOMA principle is applied so that each spatial beam not only serves its designated NF user but also shares resources with one or more FF users. Accordingly, the transmit signal at the BS is

$$\mathbf{x} = \sum_{m=1}^{M} \mathbf{p}_m \left(\sqrt{P_m} s_m^{\mathrm{NF}} + \sum_{k=1}^{K} f_{m,k} s_k^{\mathrm{FF}} \right), \tag{9}$$

where P_m denotes the transmit power for the m-th NF user, $f_{m,k}$ is the allocation coefficient for the k-th FF user on beam m, and s_m^{NF}, . s_k^{FF} . are the transmitted symbols of NF and FF users, respectively.

The received signal at the m-th NF user is

$$y_m = \mathbf{h}_m^H \mathbf{p}_m \left(\sqrt{P_m} s_m^{\mathrm{NF}} + \sum_{k=1}^{K} f_{m,k} s_k^{\mathrm{FF}} \right) + n_m. \tag{10}$$

In this framework, NF users employ successive interference cancellation (SIC) to first remove the signals of FF users before decoding their own data. In contrast, FF users directly decode their intended signals without applying SIC.

The achievable rate of the k-th FF user on beam m is

$$R_k^{\mathrm{FF}} = \log_2 \left(1 + \frac{| \mathbf{g}_k^H f_{m,k} |^2}{\gamma_k} \right), \tag{11}$$

where γ_k represents the interference-plus-noise term.

The achievable rate of the m-th NF user after successful SIC is

$$R_m^{\mathrm{NF}} = \log_2 \left(1 + \frac{P_m | \mathbf{h}_m^H \mathbf{p}_m |^2}{\sigma^2} \right). \tag{12}$$

The resource-allocation problem is formulated to maximize the sum rate of FF users while guaranteeing the QoS of NF users, under per-beam power and allocation constraints. Specifically, the optimization problem can be expressed as

$$\max_{\{P_m, f_{m,k}\}} \quad \sum_{k=1}^{K} \left(R_k^{\mathrm{FF},p} + R_k^{\mathrm{FF},s} \right) \tag{P1a}$$

$$\text{s.t. } R_m^{\mathrm{NF}} \geq R, \quad \forall m \in \mathcal{S}_k, \quad k = 1, \ldots, K \tag{P1b}$$

$$P_m + \sum_{k=1}^{K} \left| f_{m,k} \right|^2 \leq P, \qquad 1 \leq m \leq M \tag{P1c}$$

$$P_m \geq 0, \qquad 1 \leq m \leq M \tag{P1d}$$

Here, P_m denotes the transmit power allocated to the m-th NF beam, and $f_{m,k}$ is the allocation coefficient of FF user k on beam m. $R_k^{\mathrm{FF},p}$ and $R_k^{\mathrm{FF},s}$ represent the achievable rates of FF user k on its primary and secondary beams, respectively, while R_m^{NF} denotes the achievable rate of NF user m after performing SIC. The parameter R is the minimum QoS threshold required for NF users, P is the per-beam power limit, and $\mathcal{S}_k$ is the set of NF users that share beams with FF user k. The indicator function $1_{x \neq 0}$ is defined as $1_{x \neq 0} = 1$ if $x \neq 0$ and $1_{x \neq 0} = 0$ otherwise.

It is worth noting that solving (P1) requires the BS to have knowledge of the CSI of all users. This can be obtained by letting each user estimate its channel from pilot signals transmitted by the BS and feed the CSI back to the BS via a reliable feedback link.

The objective function in (P1a) implies that the k -th FF user tends to select beams for which both $|\mathbf{h}_m|$ and $|g_{m,k}|$ are large. To determine the set $\mathcal{S}_k$ and remove the indicator function in (P1a), an approximate scheduling algorithm can be applied. Each FF user is sequentially assigned to candidate beams by selecting D_x beams according to the criterion

$$\arg\max_m \min\left\{\frac{h_m}{\max\{h_1,\ldots,h_M\}}, \frac{g_{m,k}}{\max\{g_{1,k},\ldots,g_{M,k}\}}\right\}, \tag{13}$$

where the channel gains h_m and $g_{m,k}$ are normalized to the same scale. Consequently, $f_{m,k}=0$ if $m \notin \mathcal{S}_k$, and only the coefficients $f_{m,k}$ with $m \in \mathcal{S}_k$ need to be optimized.

4. Conventional Optimization Methods for the Resource-Allocation Problem

To address the problems of power allocation and beamforming design in MIMO-NOMA systems, classical optimization methods such as Successive Convex Approximation (SCA) and Branch-and-Bound (BB) [15] are employed. These two analytical techniques are used to obtain global and approximate optimal solutions, serving as benchmarks for performance comparison with modern approaches such as deep reinforcement learning.

4.1. Rationale for Choosing SCA and BB as Baselines

The optimization problem in (P1) is formulated as:

$$\max_{\{P_m, f_{m,k}\}} \sum_{k=1}^{K}\left(R_k^{FF,p} + R_k^{FF,s}\right), \tag{14}$$

with QoS and power constraints, this problem is non-convex since the objective function involves $log_2(1 + SINR)$ and the constraints are nonlinear due to interference. Obtaining the exact global solution becomes intractable when M and K are large.

- SCA: A widely use approximation technique for solving nonlinear problems by sequentially approximating them as convex problems and solving them with convex optimization methods.
- BB: A technique for finding the global optimal solution in small-scale (or special) cases, commonly used to evaluate the performance upper bound.

4.2. Successive Convex Approximation (SCA) Method

Principle of SCA: SCA is based on replacing nonlinear functions with convex approximations (typically linear or tightly convex functions). At each iteration:

1. Take a reference point from the current solution.
2. Linearize the nonlinear functions or non-convex constraints.
3. Solve the approximated convex problem.
4. Update the solution and repeat until convergence.

Applied to the MIMO-NOMA problem: the data rate of far-field (FF) user *k* is written as:

$$R_k^{FF} = \log\left(1+\frac{|\tilde{\mathbf{g}}_k^H \mathbf{f}_k|^2}{\gamma_k}\right), \tag{15}$$

where γ_k contains the noise and interference components:

$$\gamma_k = \sigma^2 + \sum_{m=1}^{M} P_m g_{m,k} + \sum_{i\neq k}\left|\tilde{\mathbf{g}}_k^H \mathbf{f}_i\right|^2. \tag{16}$$

The constraints (such as power) and the objective function are both non-convex. SCA addresses this by:

- Introduce an auxiliary variable x_k to replace $\log_2(1+x_k)$.
- The SINR constraint can then be rewritten as:

$$\left|\tilde{\mathbf{g}}_k^H \mathbf{f}_k\right|^2 \geq x_k \cdot \gamma_k. \tag{17}$$

- This is a nonlinear form due to the product $x_k \cdot \gamma_k$. SCA linearizes it around the current values $\left(x_k^{(t)}, \mathbf{f}^{(t)}\right)$:

$$x_k \cdot \gamma_k \approx x_k^{(t)} \gamma_k + \gamma_k^{(t)} \left(x_k - x_k^{(t)}\right). \tag{18}$$

Characteristics:

- SCA converges to a stationary point but does not guarantee global optimality.
- The computational complexity depends on the number of convex problems to be solved, each problem has a complexity of $O\left((M+K)^3\right)$ when using an interior-point solver.

Algorithm 1: Successive Convex Approximation - SCA

Input: A feasible initialization $(P_m^{(0)}, f_{m,k}^{(0)})$, and the convergence threshold , ϵ_{SCA}.

Output: A near-optimal solution $(P_m^*, f_{m,k}^*)$.

Initialize the iteration counter $t \leftarrow 0$

Step 1: Compute the values at the current point $(P_m^{(t)}, f_{m,k}^{(t)})$

Step 2: Approximate the non-convex constraints and the objective function.

- Data rate of the FF user:

$$R_k^{\text{FF}} = \log_2\left(1 + \frac{|\tilde{\mathbf{g}}_k^H \mathbf{f}_k|^2}{\gamma_k}\right),\ \gamma_k = \sigma^2 + \sum_{m=1}^{M} P_m g_{m,k} + \sum_{i,k} \left|\tilde{\mathbf{g}}_k^H \mathbf{f}_i\right|^2.$$

- Linearize the product $x_k \cdot \gamma_k$ using a Taylor expansion at the point $\left(x_k^{(t)}, \gamma_k^{(t)}\right)$:

$$x_k \cdot \gamma_k \approx x_k^{(t)} \gamma_k + \gamma_k^{(t)} \left(x_k - x_k^{(t)}\right)$$

Step 3: Formulate the approximated convex problem:

$$\max\nolimits_{\{P_m, f_{m,k}, x_k\}} \sum_{k=1}^{K} \log_2(1+x_k)$$

$$\text{s.t. } |\tilde{\mathbf{g}}_k^H \mathbf{f}_k|^2 \geq x_k \cdot \gamma_k \text{ (approx.)},$$

$$P_m + \sum_{k=1}^{K} |f_{m,k}|^2 \leq P, \quad P_m \geq 0.$$

Step 4: Solve the convex problem using a solver (e.g., CVX, MOSEK) to obtain the solution $(P_m^{(t+1)}, f_{m,k}^{(t+1)})$.

Step 5: Check the convergence condition:

$$\frac{\|\mathbf{f}^{(t+1)} - \mathbf{f}^{(t)}\|}{\|\mathbf{f}^{(t)}\|} < \epsilon_{\text{SCA}}.$$

If the condition is not satisfied, increase $t \leftarrow t+1$ and repeat.

until convergence;

4.3. Branch-and-Bound (BB) Method

BB is employed to obtain the global optimal solution for two special cases:

1. Case $K = 1$: only one FF user.
2. Case $D_x = 1$: each FF user is assigned to only one beam.

Principle of BB:

- Define the initial solution space.
- Partition this space into subregions (branching).
- For each subregion, compute:

$$UB(D) = \text{upper bound}, \quad LB(D) = \text{lower bound}$$

- If $UB(D) < LB_{\text{best}}$, discard the subregion.
- Stop when the gap $UB - LB < \epsilon$.

Application for $K = 1$: The problem is reduced to:

$$\max y \ \ s.t. \left(\sum_{m=1}^{M} \sqrt{g_m z_m} \right)^2 \geq \eta_0 y, \quad 0 \leq z_m \leq P - P_m^*, \tag{19}$$

this form can be solved optimally by partitioning and searching for boundary solutions.

Application for $D_x = 1$: Similarly, BB enumerates the beam configuration for each FF user and computes the upper and lower bounds by assuming the minimum and maximum interference, respectively.

Algorithm 2: Branch-and-Bound (BB) Algorithm for Resource Allocation

Input: Initial solution space $\mathcal{B}_0$, tolerance ϵ.

Output: Global optimal solution and the objective value.

Initialization: $\mathcal{S}_0 = \{\mathcal{B}_0\}, i \leftarrow 0$ Compute the upper and lower bounds for $\mathcal{B}_0$:

$$\beta_0^u = \phi^{up}(\mathcal{B}_0), \quad \beta_0^l = \phi^{lb}(\mathcal{B}_0)$$

Set the gap $\delta \leftarrow \beta_0^u - \beta_0^l$

while $\delta \geq \epsilon$ **do**

$i \leftarrow i + 1$ Select a region $\mathcal{B} \in \mathcal{S}_{i-1}$ according to the criterion:

$$\min \phi^{lb}(\mathcal{B})$$

Partition $\mathcal{B}$ along its longest edge into two subregions: $\mathcal{B}_1$ và $\mathcal{B}_2$. Update the region set:

$$\mathcal{S}_i = \{\mathcal{B}_1, \mathcal{B}_2\} \cup \left(\mathcal{S}_{i-1} \setminus \{\mathcal{B}\} \right)$$

Recalculate the upper and lower bounds for all regions in $\mathcal{S}_i$:

$$\beta_i^u = \max_{\mathcal{B} \in \mathcal{S}_i} \phi^{up}(\mathcal{B}), \quad \beta_i^l = \max_{\mathcal{B} \in \mathcal{S}_i} \phi^{lb}(\mathcal{B})$$

Update the gap $\delta \leftarrow \beta_0^u - \beta_0^l$ Discard the regions whose upper bound is smaller than β_0^l.

Characteristics:

- BB guarantees finding the global optimal solution.
- The complexity increases exponentially with the number of variables, making it practical only for small *K* or special cases.

5. REINFORCEMENT LEARNING-BASED RESOURCE-ALLOCATION OPTIMIZATION

5.1. Formulation of the Resource-Allocation Problem as an MDP and Selection of the TD3 Algorithm

5.1.1. Formulation of the Resource-Allocation Problem as an MDP

To address the above Resource Allocation problem, we formulate it as a Markov Decision Process (MDP) and apply Twin Delay Deep Deterministic Policy Gradient (TD3) for this problem.
The Resource Allocation problem in MIMO-NOMA systems modelled as MDP which allow take advantage of Deep Reinforcement Learning (DRL) to solve complex optimization problems, without closed-form target function.
In this MIMO-NOMA system, the MDP is defined as a tuple with (S, A, P, r), where:

- S (State): The set of state, in this problem every state $s \in S$ include channel state information between BS and UE (include NF and FF), system configuration, Quality of Service (QoS) indicator, etc.

$$s_t = \left[h_1, \ldots, h_m, g_{1,1}, g_{1,2}, \ldots, g_{m,k} \right], \tag{20}$$

where:

h_m : channel gain between BS and NF user m after applying ZF beamforming

$g_{m,k}$: channel gain between beam m and FF user k .

Values are normalized to the domain [0,1]

- A (Action): Action set, for each action $a \in A$ is a resource allocation vector (i.e. power for per beam or per UE), in continuous space $\mathbb{R}^d$.

$$a_t = \left[\hat{P}_1, \ldots, \hat{P}_M, \hat{f}_{1,1}, \ldots, \hat{f}_{m,k} \right], \tag{21}$$

where:

$\hat{P}_m$: values generated by agent, mapping to power P_m .

$\hat{f}_{m,k}$: values mapping to $\left| f_{m,k} \right|^2$

The action generated by the Actor is subsequently normalized to satisfy:

$$P_m + \sum_{k=1}^{K} \left| f_{m,k} \right|^2 \le P, \forall m \tag{22}$$

- P (The state transition probability): The state transition probability, $P(s' \mid s, a)$, describe probability of transition from s to s' when taking the action a . In non-deterministic environment (due to fading, noise, etc.), this function may not be explicitly defined, but it can be sampled through simulations.
- r (Reward function): The reward function $r(s, a)$ evaluates the quality of acting a in state s . It can be designed based on the total throughput, the number of UEs satisfying QoS requirements, or a weighted function combining multiple system performance metrics.

$$r(s_t, a_t) = \alpha \sum_{k=1}^{k} R_k^{FF}(t) - \beta \sum_{m=1}^{M} \max\left(0, R_{min} - R_m^{NF}(t)\right) \tag{23}$$

The first term maximizes the sum throughput of far-field users, while the second softly penalizes near-field users that violate their QoS thresholds. The penalty term increases

linearly with the magnitude of violation, acting as a differentiable soft barrier that stabilizes training. The weighting factors α and β are empirically tuned to maintain a balance between overall throughput maximization and QoS satisfaction, ensuring that neither objective dominates during policy learning.

- Data rate for FF user k :

$$R_k^{FF} = \log_2\left(1 + \frac{\left|\sum_{m=1}^{M} f_{m,k} g_{m,k}\right|^2}{\sigma^2 + \sum_{m=1}^{M} P_m g_{m,k} + \sum_{i \neq k}\left|\sum_{m=1}^{M} f_{m,i} g_{m,k}\right|^2}\right), \tag{24}$$

- Data rate for NF user m :

$$R_m^{NF} = \log_2\left(1 + \frac{P_m h_m}{\sigma^2}\right), \tag{25}$$

where:

σ^2 : noise power at each user,

P_m : power allocated to NF user m on beam m ,

$g_{m,k}$: channel gain from beam m to FF user k ,

$f_{m,k}$: beamforming coefficient corresponding to FF user k on beam m .

The optimization problem is to find an optimal policy π^* that maps states to actions such that the expected cumulative reward is maximized:

$$\pi^* = \arg\max_{\pi} \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t r\left(s_t, a_t\right)\right], \tag{26}$$

where $\gamma \in (0,1)$ is discount factor, s_0 is the initial state, and $a_t \sim \pi\left(s_t\right)$.

5.1.2. Selection of the TD3 Algorithm

The above problem is modelled as an MDP with both state and action spaces being continuous, non-convex, and even non-linear, under complex system dynamics (due to multipath, fading, and channel variations).

Among the deep reinforcement learning policy-based methods, TD3 (Twin Delayed Deep Deterministic Policy Gradient) stands out because:

- Support for Continuous Action Space: TD3 constructs a policy function (Actor) $a = \mu\left(s|\theta^{\mu}\right)$ and two value functions (Critics) $Q_1\left(s,a|\theta^{Q_1}\right), Q_2\left(s,a|\theta^{Q_2}\right)$ using deep neural networks, allowing optimization of power allocation across beams/UEs.
- Reduction of Value Overestimation Bias in Q : By employing two parallel Critics and updating the target with the minimum value:

$$y = r + \gamma \min_{i=1,2} Q_i\left(s', \mu\left(s'\right)\right), \tag{27}$$

this approach avoids overestimation and stabilizes the learning process.
- Policy Smoothing (Target Policy Smoothing): TD3 introduces small Gaussian noise into target actions when updating Critics, preventing the policy from overfitting sharp, undesired action values.
- Delayed Policy Update: The Actor is updated only after several Critic updates, ensuring that better stability in environments with high noise.

5.2. Proposed TD3 Algorithm

We use the Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm to solve the formulated MDP. For continuous control issues like beamforming and power allocation in MIMO-NOMA systems, TD3 is especially well-suited. Through three main approaches, the algorithm enhances learning performance and stability: (i) target policy smoothing to lessen sensitivity to strong Q-value peaks; (ii) double Critics to decrease overestimation bias; and (iii) delayed policy updates to stabilize Actor learning. Algorithm 3 provides a summary of the comprehensive training process.

Algorithm 3: TD3-Based Resource Allocation in MIMO-NOMA

Input: Number of training steps T , discount factor γ, smoothing coefficient τ , minibatch size B .

Output: Optimized Actor policy π_θ .

Initialization: Initialize Actor network π_θ , two Critic networks Q_{ϕ_1} , Q_{ϕ_2} and their target networks.

Initialize an empty Replay Buffer D .

for $t \leftarrow 1$ **to** T **do**

Observe state s_t .

Generate action $a_t = \pi_\theta(s_t) + N(Gaussian\,Noise)$

Execute action, obtain reward r_t, and next state s_{t+1}

Store (s_t, a_t, r_t, s_{t+1}) into buffer D

if $|D| \geq B$ then

- Sample a minibatch from D
- Generate target action:

$$\tilde{a} = \pi_{\theta'}(s') + \epsilon, \epsilon \sim clip\left(N(0,\sigma), -c, c\right)$$

- Compute target value:

$$y = r + \gamma \min_{i=1,2} Q_{\phi_i'}(s', \tilde{a})$$

- **Update Critic:**

$$L = \left(Q_{\phi_i}(s,a) - y\right)^2$$

if $t \bmod d = 0$ **then**

- **Update Actor**

$$\nabla_\theta J = \mathbb{E}\left[\nabla_a Q_{\phi_1}(s,a)\big|_{a=\pi_\theta(s)} \nabla_\theta \pi_\theta(s)\right]$$

- **Update target networks:**

$$\theta' \leftarrow \tau\theta + (1-r)\theta'$$

$$\phi_i' \leftarrow \tau\phi_i + (1-\tau)\phi_i'$$

6. NUMERICAL AND SIMULATION RESULTS

6.1. Simulation Setup

In this section, we present the simulation setup and discuss the obtained results. The simulation framework was implemented in Python 3.10 using TensorFlow 2.x and NumPy. Each experiment was executed on an Intel Core i7-10850H with GPU Quadro P620. The TD3 agent was trained using a replay buffer with random sampling and mini-batch gradient updates. The system-level

parameters of the considered Massive MIMO-NOMA system, including carrier frequency, antenna configuration, number of users, power levels, and noise assumptions, are summarized in Table 1. Furthermore, the training hyperparameters and neural network architecture of the proposed TD3-based algorithm are reported in Table 2. These configurations ensure reproducibility of all results and provide a fair basis for performance comparison with conventional optimization methods.

Table 1. System parameters of the Massive MIMO-NOMA simulation.

Parameter	**Value**	**Note**
Carrier frequency f_c	28 GHz	mmWave
Number of BS antennas N	64	ULA, $d = \lambda/2$
Number of NF users M	36	Each NF user ⇔ one ZF beam
Number of FF users K	1 or 2	Two evaluation scenarios
Maximum transmit power $P_{\max}$	{0, 10, 20, 30} dBm	Four SNR levels
Thermal noise σ^2	10^{-11} W	≈ -80 dBm
NF QoS requirement $R_{\min}$	0.1 bps/Hz	$\varepsilon = 2^{0.1} - 1$
Rayleigh distance d_R	$2D^2/\lambda$	With $D = (N-1)d$

Table 2. Training hyperparameters and TD3 architecture.

Parameter / Architecture	**Value**	**Description**
Actor architecture	$(d_s) \to 512 \to 512 \to d_a$	ReLU - ReLU - Tanh
Critic architecture	$(d_s + d_a) \to 512 \to 512 \to 1$	Two parallel Critics (ReLU)
Optimizer	Adam (lr = 3×10^{-4})	For both Actor and Critic
Discount factor K	0.99	
Polyak factor τ	0.005	Target network update
Policy noise	$\mathcal{N}(0, 0.2)$, clipped to ± 0.5	Target policy smoothing
Actor update delay	Every 2 Critic steps	Update Actor and target every 2 steps
Training steps T	10,000	Total training steps
Minibatch size B	128	Training minibatch size
Replay buffer	2×10^5 samples	FIFO replay memory
Initial exploration noise	0.10	Gaussian exploration
Max_action	1.0	Action normalization (Tanh)

The above parameters ensure reproducibility of all results. In particular, the 512-512 Actor/Critic network architecture is selected based on empirical tuning, enabling TD3 to converge quickly without significantly increasing the online inference complexity.

6.2. Analysis of the DRL Training Process

To examine the convergence and stability of the TD3 algorithm, we monitor (i) the average reward and (ii) the Actor/Critic loss functions over 10,000 training steps.

Scenario $K = 1$, $D_x = 1$

Figure 3 illustrates the convergence of the reward curve of TD3 with a smoothing window of $w = 300$. This smoothed curve highlights the underlying trend of performance improvement, filtering out short-term fluctuations, thus offering a more reliable view of the convergence behavior.

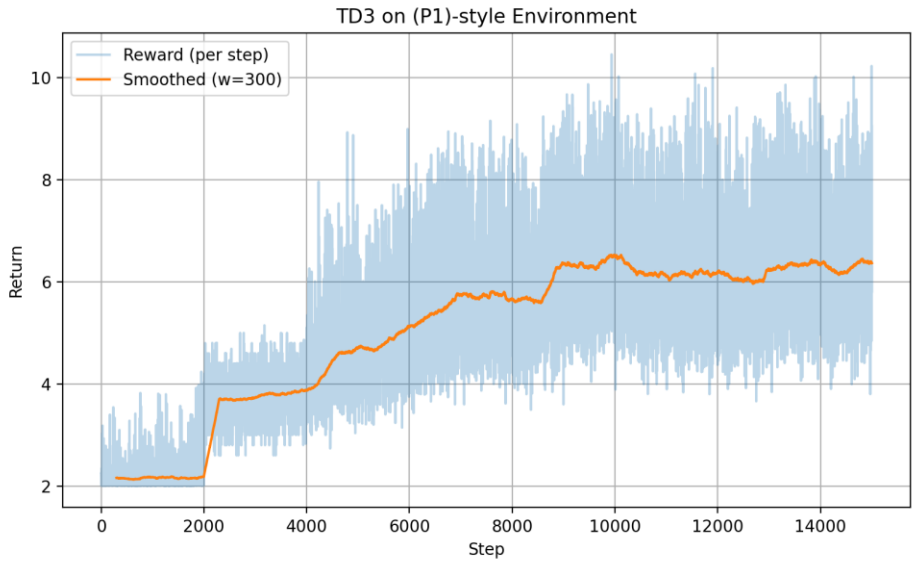

Figure 3. Convergence process of the TD3 reward ($K = 1$, $D_x = 1$) with a smoothing window of $w = 300$.

After the initial warm-up phase (~2000 steps), the smoothed reward exhibits a clear upward trend, indicating progressive improvement of the agent's learning process over time. Several noticeable step-like increases can be observed, each reflecting moments when the agent discovers more effective policies through exploration and exploitation. Although the per-step rewards show considerable fluctuations, the smoothed curve consistently demonstrates a positive trajectory, providing reliable evidence of stable policy enhancement.

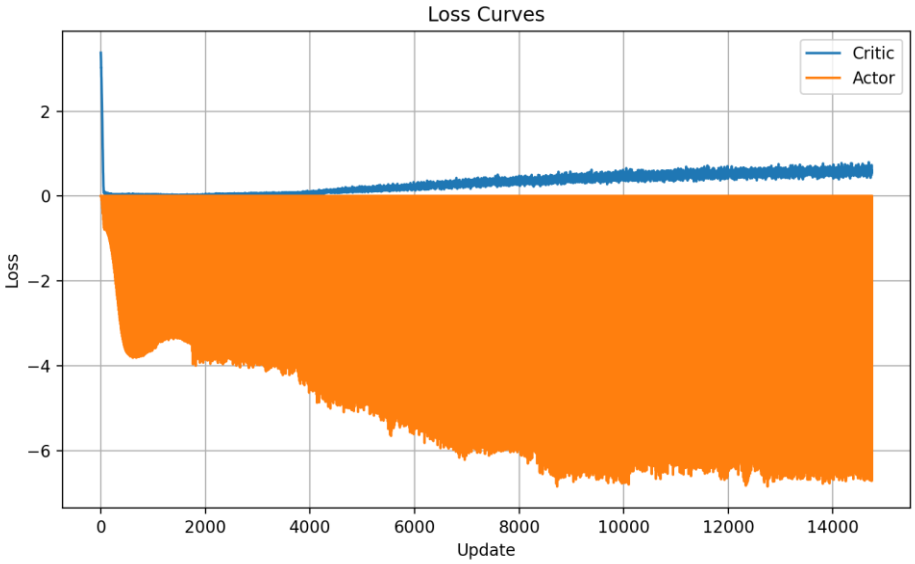

Figure 4. Actor and Critic loss curves of TD3 ($K = 1$, $D_x = 1$)

Figure 4 illustrates the training dynamics of the Actor and Critic networks under the TD3 algorithm for the case of $w=300$, $D_x=1$. At the beginning, the Critic loss shows a sharp peak before rapidly decreasing, followed by a gradual rise to around 0.5 toward the end of training. This behavior indicates that the Critic learns quickly but begins to experience noise or mild overfitting as training progresses. In contrast, the Actor loss steadily decreases and remains relatively stable at a low level, reflecting that the Actor gradually acquires a more effective policy. The smoothness of the Actor loss curve, without significant oscillations, further confirms the stability of the policy updates throughout the training process.

The relationship between reward and loss is evident as the reward sharply increases from ~2.2 to above 5.5 once the initial spike in the Critic loss disappears and both losses stabilize, and although the Critic loss does not continue to decrease further, the reward remains high and stable because the Actor directly optimizes the expected $Q(s,\pi(s))$ rather than depending on the Critic's mean-squared error converging to zero. This interplay explains both the step-like jumps in reward and the final plateau observed in figure.

Scenario $K=2$, $D_x=1$

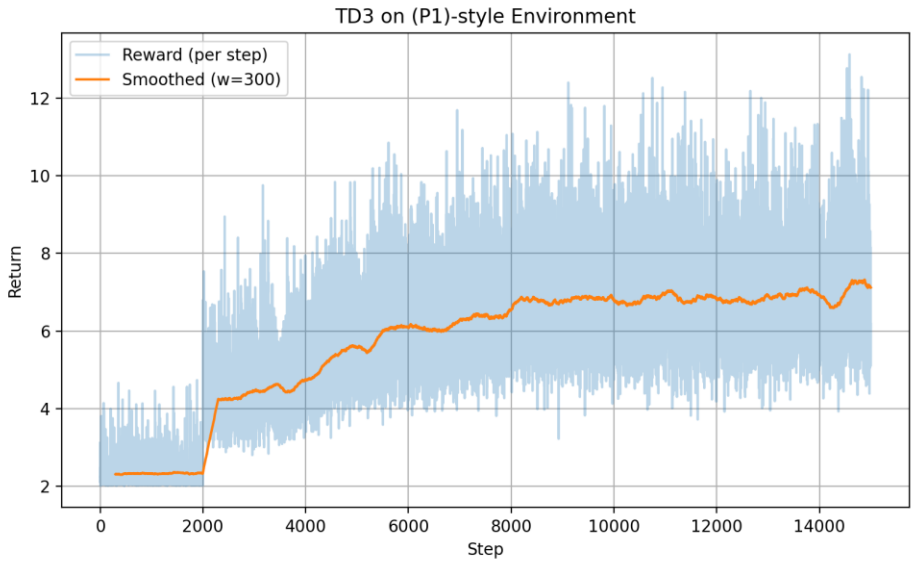

Figure 5. Convergence process of the TD3 reward ($K=2$, $D_x=1$) with a smoothing window of $w=300$.

In Figure 5, after the warm-up phase (around step 2000), the smoothed reward increases steadily, reflecting the progressive improvement of the agent's learning process. Several noticeable step-like jumps can be observed, each corresponding to the discovery of more effective policies through the exploration-exploitation trade-off. Following this stage, the reward gradually converges to a relatively high and stable plateau toward the end of training. Although the per-step rewards exhibit considerable fluctuations, the smoothed curve consistently demonstrates a clear upward trend, providing reliable evidence of convergence.

Figure 6 illustrates training losses of the Actor and Critic networks for the case $K=2$, $D_x=1$. The Critic loss shows a very high initial spike (peak ≈ 11) but quickly drops to below 0.1, then gradually increases with small fluctuations and stabilizes within a narrow range of about 0.5 - 2.3 toward the end of training. This indicates that the Critic converges rapidly but may experience

mild noise or overfitting during extended training. Meanwhile, the Actor loss decreases steadily from around −4 to approximately −7.5 to −7.8. Because the optimization objective is $\mathcal{L}_{\text{actor}} = -\mathbb{E}[Q(s,\pi(s))]$, this downward trend (more negative values) reflects that the estimated $Q(s,\pi(s))$ increases, demonstrating continuous policy improvement. The smooth trajectory of the Actor loss, without strong oscillations, further indicates stable and reliable policy updates throughout the training process.

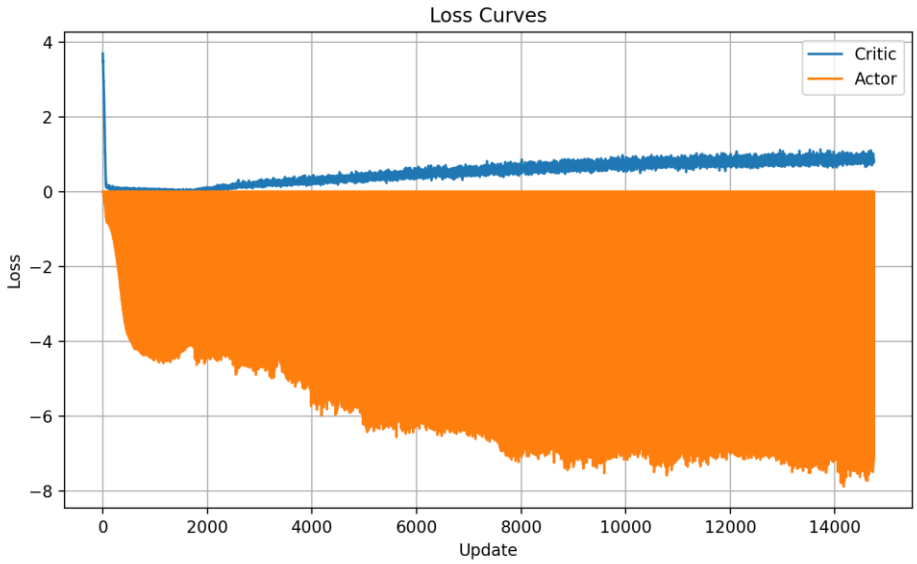

Figure 6. Actor and Critic loss curves of TD3 ($K = 2$, $D_x = 1$)

Once the initial spike of the Critic loss vanishes and both critics stabilize at low levels (after a few tens of steps), the reward exhibits a sharp step-like increase. This indicates that a stable Q-value estimation enables more effective policy updates for the Actor network. In the later stages, even though the Critic loss shows a slight upward drift - mainly due to changing temporal-difference targets as the policy evolves - the Actor loss continues to decrease into deeper negative values. Consequently, the reward keeps increasing and then maintains a high level. This behavior is consistent with the TD3 mechanisms, where twin critics and delayed policy updates mitigate noise and value overestimation, thereby ensuring stable policy improvement.

Figures 3–6 show that the TD3 agent converges smoothly after approximately 10^4 iterations, with the critic losses remaining bounded and the actor policy steadily improving. The training cost is mainly determined by critic network updates but remains manageable due to experience replay and mini-batch learning. In summary, the characteristic learning patterns of the TD3 algorithm. First, the Critic loss shows an initial spike in mean-squared error during the early updates, which quickly diminishes once the replay buffer becomes sufficiently diverse. Second, the Actor loss remains negative and steadily decreases over time, consistent with minimizing $-\mathbb{E}[Q(s,\pi(s))]$. Third, the reward curves follow a step-like trajectory with short plateaus, reflecting the delayed policy updates and the conditioning effect of the twin critics. In both scenarios, the major jumps in reward coincide with periods when the Critic stabilizes, confirming that reliable Q-value estimation is essential for effective policy improvement.

6.3. Sum-Rate Comparison of RL (TD3) and SCA Baselines

To further evaluate the effectiveness of the proposed reinforcement learning approach, this section compares the achievable sum-rate performance of TD3 against the successive convex

approximation (SCA) algorithm under two initialization strategies: deterministic and random. The comparison is conducted across different transmit power levels and in both considered scenarios, providing insights into how the RL-based method performs relative to traditional optimization-based baselines.

Scenario $K=1$, $D_x=1$

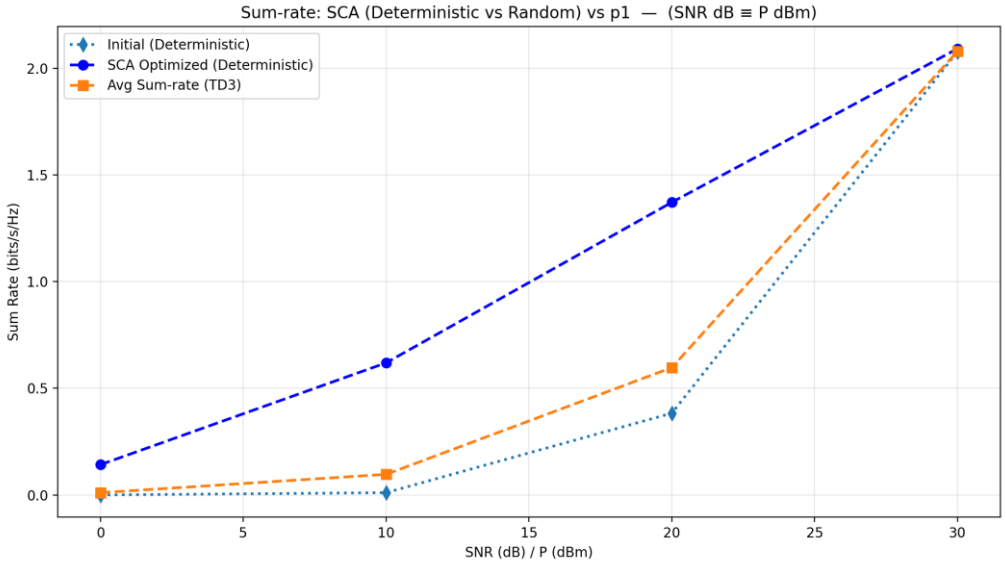

Figure 7. Sum-rate comparison versus transmit power between RL (TD3), SCA with deterministic initialization, and SCA with random initialization for the scenario $K=1$, $D_x=1$

In figure 7, the deterministic SCA consistently achieves the highest sum-rate across all transmit power levels, demonstrating its strong optimization capability when initialized properly. The RL-based method (TD3) attains comparable performance to deterministic SCA at high transmit powers, while significantly outperforming the randomly initialized SCA in the low-to-medium power range. The random initialization case yields the lowest performance, especially at low power, confirming the importance of efficient optimization strategies. Overall, the RL-based approach proves to be a practical and effective solution, as it converges near-optimal performance without requiring favorable initialization. In terms of system performance, the RL-based method lags behind deterministic SCA at low and medium transmit power levels but catches up and reaches near-identical performance at high power (30 dBm), while consistently outperforming randomly initialized SCA at 10–20 dBm. This observation highlights the robustness of the learned policy, which can generalize well across different power regimes. Moreover, the RL framework offers greater scalability for dynamic or large-scale scenarios where iterative optimization methods become computationally expensive.

Scenario $K=2$, $D_x=1$

In Figure 8, the deterministic SCA continues to achieve the highest sum-rate at all transmit power levels, highlighting the strong effectiveness of the algorithm when initialized properly. The RL-based method (TD3) consistently outperforms the randomly initialized SCA across all power levels, with the performance gap becoming more pronounced in the high-power region (20-30 dBm). The random initialization case yields relatively low performance, and the gap with RL-based methods increases further when the number of users rises to $K=2$.

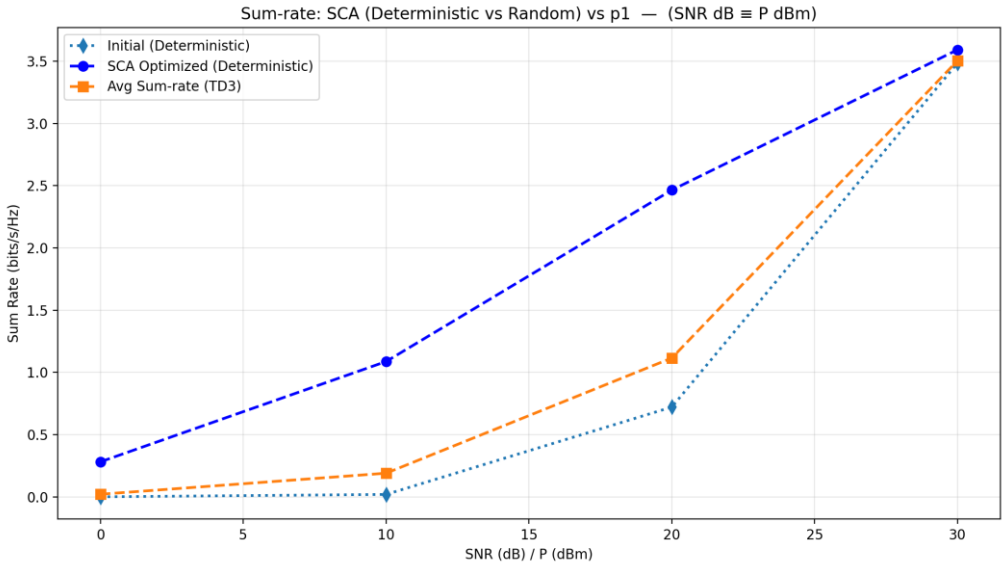

Figure 8. Sum-rate comparison versus transmit power between RL (TD3), SCA with deterministic initialization, and SCA with random initialization for the scenario $K = 2$, $D_x = 1$

In summary, in terms of system performance, the RL-based method remains inferior to deterministic SCA across all transmit power levels but consistently outperforms randomly initialized SCA (except at 0 dBm). This demonstrates that the learned policy is meaningful and improves steadily with increasing transmit power. The sum-rate results further reveal that (i) in the low-to-medium power regime, the RL policy tends to be more rigid due to NF QoS constraints and the associated reward-penalty structure, which leads to lower performance compared with deterministic SCA; and (ii) as the transmit power increases, RL approaches – and in some cases can even match – the deterministic SCA, while clearly outperforming random initialization. This is an encouraging indication that when constraints are relaxed, the actor is able to learn near-optimal allocation structures without requiring the iterative process of SCA. Overall, the RL-based approach proves to be a practical solution, converging reliably even with more users and effectively overcoming the limitations of optimization algorithms under poor initialization, thereby offering a scalable and real-time alternative for next-generation Massive MIMO-NOMA systems. Future extensions may integrate meta-learning or multi-agent coordination to further enhance adaptability and scalability in practical deployments.

7. CONCLUSION AND FUTURE WORK

This work considered a conventional downlink NOMA system where spatial beams are preconfigured for near-field users, and far-field users are superimposed onto these beams through power-domain multiplexing. Under the assumptions of single-antenna users and perfect SIC, the TD3 algorithm was implemented to learn power allocation policies. Simulation results demonstrated that the proposed DRL-based approach can approach, and in some cases even surpass, the performance of SCA at high SNR, while a performance gap remains in the low-SNR regime.

Several research directions remain open. Extending to multi-antenna users with attention-enhanced Actor-Critic architectures can improve large channel matrix processing. Multi-agent DRL with CTDE enables scalability for many users. Online learning under imperfect CSI, aided by meta- or transfer learning, enhances robustness. These directions can further reduce the SNR-range performance gap and advance Massive MIMO-NOMA systems resilient to channel variability, hardware impairments, and diverse service needs.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

ACKNOWLEDGEMENTS

This research is funded by University of Science, VNU-HCM under grant number ĐT-VT 2023-02.

REFERENCES

[1] Mathews, Belcy & Muthu, Tamilarasi. (2024). Adaptive Hybrid Deep Learning Based Effective Channel Estimation in MIMO-Noma for Millimeter-Wave Systems with an Enhanced Optimization Algorithm. International journal of Computer Networks & Communications. 16. 113-131. 10.5121/ijcnc.2024.16507.

[2] Z. Ding, "NOMA Beamforming in SDMA Networks: Riding on Existing Beams or Forming New Ones?," in IEEE Communications Letters, vol. 26, no. 4, pp. 868-871, April 2022, doi: 10.1109/LCOMM.2022.3146583

[3] R. W. Heath, N. González-Prelcic, S. Rangan, W. Roh and A. M. Sayeed, "An Overview of Signal Processing Techniques for Millimeter Wave MIMO Systems," in IEEE Journal of Selected Topics in Signal Processing, vol. 10, no. 3, pp. 436-453, April 2016, doi: 10.1109/JSTSP.2016.2523924.

[4] Y. Zou, W. Rave and G. Fettweis, "Analog beamsteering for flexible hybrid beamforming design in mmwave communications," 2016 European Conference on Networks and Communications (EuCNC), Athens, Greece, 2016, pp. 94-99, doi: 10.1109/EuCNC.2016.7561012.

[5] E. Björnson and L. Sanguinetti, "Power Scaling Laws and Near-Field Behaviors of Massive MIMO and Intelligent Reflecting Surfaces," in IEEE Open Journal of the Communications Society, vol. 1, pp. 1306-1324, 2020, doi: 10.1109/OJCOMS.2020.3020925.

[6] J. Zhu, Z. Wan, L. Dai, M. Debbah and H. V. Poor, "Electromagnetic Information Theory: Fundamentals, Modeling, Applications, and Open Problems," in IEEE Wireless Communications, vol. 31, no. 3, pp. 156-162, June 2024, doi: 10.1109/MWC.019.2200602.

[7] H. Zhang, N. Shlezinger, F. Guidi, D. Dardari, M. F. Imani and Y. C. Eldar, "Beam Focusing for Near-Field Multiuser MIMO Communications," in IEEE Transactions on Wireless Communications, vol. 21, no. 9, pp. 7476-7490, Sept. 2022, doi: 10.1109/TWC.2022.3158894.

[8] Aljumaily, Mustafa & Li, Husheng. (2020). Mobility Speed Effect and Neural Network Optimization for Deep MIMO Beamforming in mmWave Networks. International Journal of Computer Networks and Communications. 12. 1-14. 10.5121/ijcnc.2020.12601.

[9] Z. Zhang, M. Hua, C. Li, Y. Huang and L. Yang, "Beyond Supervised Power Control in Massive MIMO Network: Simple Deep Neural Network Solutions," in IEEE Transactions on Vehicular Technology, vol. 71, no. 4, pp. 3964-3979, April 2022, doi: 10.1109/TVT.2022.3146434.

[10] B. R. Manoj, M. Sadeghi and E. G. Larsson, "Downlink Power Allocation in Massive MIMO via Deep Learning: Adversarial Attacks and Training," in IEEE Transactions on Cognitive Communications and Networking, vol. 8, no. 2, pp. 707-719, June 2022, doi: 10.1109/TCCN.2022.3147203

[11] T. T. Vu, H. Q. Ngo, M. N. Dao, D. T. Ngo, E. G. Larsson and T. Le-Ngoc, "Energy-Efficient Massive MIMO for Federated Learning: Transmission Designs and Resource Allocations," in IEEE Open Journal of the Communications Society, vol. 3, pp. 2329-2346, 2022, doi: 10.1109/OJCOMS.2022.3222749

[12] Q. An, S. Segarra, C. Dick, A. Sabharwal and R. Doost-Mohammady, "A Deep Reinforcement Learning-Based Resource Scheduler for Massive MIMO Networks," in IEEE Transactions on Machine Learning in Communications and Networking, vol. 1, pp. 242-257, 2023, doi: 10.1109/TMLCN.2023.3313988.

[13] I. Ahmed, M. K. Shahid and T. Faisal, "Deep Reinforcement Learning Based Beam Selection for Hybrid Beamforming and User Grouping in Massive MIMO-NOMA System," in IEEE Access, vol. 10, pp. 89519-89533, 2022, doi: 10.1109/ACCESS.2022.3199760.

[14] Y. Oh, A. Ullah and W. Choi, "Multi-Objective Reinforcement Learning for Power Allocation in Massive MIMO Networks: A Solution to Spectral and Energy Trade-Offs," in IEEE Access, vol. 12, pp. 1172-1188, 2024, doi: 10.1109/ACCESS.2023.3347788.

[15] Y. Cao, G. Zhang, G. Li and J. Zhang, "A Deep Q-Network Based-Resource Allocation Scheme for Massive MIMO-NOMA," in IEEE Communications Letters, vol. 25, no. 5, pp. 1544-1548, May 2021, doi: 10.1109/LCOMM.2021.3055348.

[16] L. Chen, F. Sun, K. Li, R. Chen, Y. Yang and J. Wang, "Deep Reinforcement Learning for Resource Allocation in Massive MIMO," 2021 29th European Signal Processing Conference (EUSIPCO), Dublin, Ireland, 2021, pp. 1611-1615, doi: 10.23919/EUSIPCO54536.2021.9616054.

[17] Z. Ding, R. Schober and H. V. Poor, "NOMA-Based Coexistence of Near-Field and Far-Field Massive MIMO Communications," in IEEE Wireless Communications Letters, vol. 12, no. 8, pp. 1429-1433, Aug. 2023, doi: 10.1109/LWC.2023.3277469.

AUTHORS

Pham Hoai An received a B.S. degree in Electronics and Telecommunications, from the University of Science, Vietnam National University, Ho Chi Minh City (VNU-HCM). His research interests focus on new-generation wireless communications, particularly 5G-Advanced and 6G mobile networks.

Nguyen Dung received a B.S. degree with honors in Electronics and Telecommunications Engineering, specializing in Telecommunications and Networks, from the University of Science, Vietnam National University, Ho Chi Minh City (VNU-HCM). He is now an M.S. student at the same university. His research interests include telecommunication networks and the application of artificial intelligence in optimization problems for telecommunications.

Nguyen Thi Xuan Uyen obtained her Master's degree in Electronics Engineering, specializing in Electronics, Telecommunications, and Computer Engineering, in 2023 at the University of Science, Vietnam National University, Ho Chi Minh City (VNU-HCM). She is currently a lecturer at the same university. Her current research interests include wireless communications and telecommunication systems in 5G and 6G networks.

Nguyen Thai Cong Nghia is a lecturer in the Faculty of Electronics - Telecommunications, University of Science, VNU-HCM. He received his M.Sc. degree in Electronic Engineering, speciality of Electronics, Telecommunications, and Computer from University of Science, VNU-HCM. His research interests include wireless communications and deep learning in telecommunications.

Ngo Minh Nghia obtained his Master's degree in Electronics Engineering, specializing in Electronics, Telecommunications, and Computer Engineering, in 2023 at the University of Science, Vietnam National University Ho Chi Minh City (VNU-HCM). He is currently a lecturer at the same university. His current research interests focus on the application of machine learning in telecommunication systems for 5G and 6G networks.