Toward Autonomous Power Control in Semi-Grant-Free NOMA Systems: A Power Pool-Based Approach

Muhammad Fayaz, *Member, IEEE,* Wenqiang Yi, *Member, IEEE,* Yuanwei Liu, *Fellow, IEEE,* Subramaniam Thayaparan, *Member, IEEE,* and Arumugam Nallanathan, *Fellow, IEEE*

***Abstract*—In this paper, we design a resource block (RB) oriented power pool (PP) for semi-grant-free non-orthogonal multiple access (SGF-NOMA) in the presence of residual errors resulting from imperfect successive interference cancellation (SIC). In the proposed method, the BS allocates one orthogonal RB to each grant-based (GB) user, and determines the acceptable received power from grant-free (GF) users and calculates a threshold against this RB for broadcasting. Each GF user as an agent, tries to find the optimal transmit power and RB without affecting the quality-of-service (QoS) and ongoing transmission of the GB user. To this end, we formulate the transmit power and RB allocation problem as a stochastic Markov game to design the desired PPs and maximize the long-term system throughput. The problem is then solved using multi-agent (MA) deep reinforcement learning algorithms, such as double deep Q networks (DDQN) and Dueling DDQN due to their enhanced capabilities in value estimation and policy learning, with the latter performing optimally in environments characterized by extensive states and action spaces. The agents (GF users) undertake actions, specifically adjusting power levels and selecting RBs, in pursuit of maximizing cumulative rewards (throughput). Simulation results indicate computational scalability and minimal signaling overhead of the proposed algorithm with notable gains in system throughput compared to existing SGF-NOMA systems. We examine the effect of SIC error levels on sum rate and user transmit power, revealing a decrease in sum rate and an increase in user transmit power as QoS requirements and error variance escalate. We demonstrate that PPs can benefit new (untrained) users joining the network and outperform conventional SGF-NOMA without PPs in spectral efficiency.**

***Index Terms*—Distributed power control, Internet of things, multi-agent reinforcement learning, non-orthogonal multiple access, semi-grant-free transmission**

I. Introduction

NON-orthogonal multiple access (NOMA) is a promising multiple access paradigm for one of the most important use cases in the fifth-generation and beyond cellular network, namely massive machine type communication (mMTC) [1]. Providing massive connectivity to satisfy the explosive increase in the number of mobile devices is the main challenge for mMTC. To this end, power-domain NOMA has become a suitable solution as it allows multiple users or devices to share limited spectrum resources rather than solely occupying them [2]. In particular, NOMA multiplexes different users in the same time/frequency resource block (RB) using superposition coding at the transmitter side and the successive interference cancellation (SIC) method at receivers [3]. To enable the accomplishment of mMTC, as well as to ensure quality of service (QoS) with low-latency communication and small signaling overhead, NOMA with two types of access methods, i.e., grant-free (GF) and grant-based (GB) access, has been proposed [4]. In GB transmission, a user processes handshakes before actual data transmission, leading to a signal overhead and high access latency. In addition, GB transmission is not suitable for some Internet of Things (IoT) scenarios where IoT applications require a low data rate but massive connectivity [5]. In GF transmission, the user transmits data directly, without any handshakes or schedule requests [6]. Therefore, GF transmission provides massive connectivity for short-packet IoT applications. However, GF transmission leads to frequent collisions because of the absence of base station (BS) involvement in the scheduling of orthogonal RBs [7] [8]. Recently, a hybrid version of GF and GB NOMA, known as semi-grant-free (SGF) NOMA, has been considered for uplink transmission owing to its potential to enhance connectivity and reduce access latency by allowing GF and GB users to share the same RB [9]. It is worth noting that SGF-NOMA also guarantees the QoS of GB users because it only allocates redundant resources of GB users to GF users. However, this uplink transmission depends heavily on the power control (PC) method, especially in the presence of residual errors owing to imperfect SIC.

A. Related SGF-NOMA Works

Recently, pure GF and GB NOMA transmission schemes have been extensively studied from various perspectives. By leveraging the distance-dependent path loss characteristic, a location-based power pool (PP) [4] is developed for pure GF IoT networks using a cooperative multi-agent Double Deep Q Network (MA-DDQN) algorithm. Each GF user is able to randomly choose one transmit power from the received PP for uploading messages. As a result of this efficient design, the PP reduces signal overhead.

To ensure the QoS of GB users and limit the number of GF users, two techniques are proposed in [9]. In the proposed

M. Fayaz is with the Department of Computer Science and IT, University of Malakand, Pakistan (email: m.fayaz@uom.edu.pk).

Y. Liu, and A. Nallanathan are with Queen Mary University of London, London, UK (email:{yuanwei.liu, a.nallanathan}@qmul.ac.uk).

W. Yi is with the School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, U.K. (email: wy23627@essex.ac.uk).

S. Thayaparan is with Department of Electronic and Telecommunication Engineering, University of Moratuwa, Sri Lanka (email: thayaparan@uom.lk).

scheme, contention control mechanisms have been developed to effectively regulate the number of GF users in order to suppress interference to GB users. In the first technique, the BS decodes the GB user's signal in the initial stage of SIC, i.e., only GF users with weak channels are permitted to share resources with GB users. For the second scheme, the BS decodes the GF user's signals first, i.e., users with strong channel gain can share resources with GB users. Therefore, the first scheme is ideal for cell-edge users who are GF users. However, the second scheme is more suitable for scenarios where GF users are close to the BS. In [10], the QoS of GB users was ensured by utilizing the flexibility to select the NOMA decoding order. This new scheme combines the two schemes discussed in [9] with additional benefits. In comparison to the previous two schemes, the approach in [10] effectively mitigates error floors in outage probability and significantly increases transmission robustness without requiring precise PC between users. The authors in [11] investigate SGF-NOMA in two different scenarios. In the first scenario, GF users are considered as cell-center users, and GB is considered as cell-edge users. In the second scenario, GF users are located at the edge of the cell, while GB users are distributed near the BS. To determine whether GF users can share the channels occupied by GB users, the authors proposed a dynamic protocol to establish a dynamic channel quality threshold in order to reduce unexpected interference to GB users. The proposed dynamic protocol demonstrates superior results for both scenarios compared to open-loop protocols. In the study by [12], the received power of GB users is utilized as a dynamic quality threshold, and closed-form expressions for GF and GB users are derived. The maximum rate of GF users and the ergodic rate without an error floor for GB users have been observed. An adaptive power allocation strategy for GB users is implemented in [13] to address performance degradation issues caused by GF users. A scheduling scheme based on maximum data rate is proposed in [14]. GF users that produce more data rates are scheduled to be paired with GB users.

B. Motivation and Contributions

Although NOMA-assisted SGF transmissions have been studied in the aforementioned work, the following critical issues still remain unresolved:

- **Imperfect SIC:** NOMA performance critically depends on the SIC process, especially when considering uplink transmission. Therefore, it is necessary to consider the effect of imperfect SIC on SGF-NOMA schemes.
- **Distributed Power Control:** Channel conditions and interference levels are subject to rapid changes in dynamic wireless environments. To ensure dependable communication and adapt to fluctuating network conditions, users should consistently monitor their environment and adjust their transmit power accordingly. Therefore, IoT users must be able to autonomously acquire appropriate power levels based on their local information (e.g., channel conditions), as it is impractical to expect them to interact frequently with the BS, given their limited resources [15], [16].
- **Designing Scalable ML Based Algorithms:** Reinforcement learning (RL) has the potential of taking decisions and performing learning simultaneously [17], [18]. However, training RL models requires extensive computational resources and may take a long time. It can limit the practicality and scalability of RL applications, particularly in a new resource-constrained environment with a large number of agents and time-sensitive or real-time settings.

As discussed in [4], PP is an efficient design to reduce signal overhead. However, location-dependent PP is not suitable for users with heterogeneous priorities as in SGF-NOMA. More specifically, regardless of the user's location, the GB user has the highest priority. Therefore, its transmit power cannot be sacrificed in order to increase GF access. To this end, the design needs to shift from being location-oriented to RB-oriented. Therefore, this paper focuses on designing RB-oriented received PP for SGF-NOMA. The main contributions are outlined as follows:

1) MA-DRL Framework for SGF-NOMA: We address the throughput optimization problem by formulating it as a Multi-Agent Markov Decision Process (MA-MDP) and employing the MA-Dueling DDQN algorithm for solutions. This method optimizes system performance and facilitates the formation of a RB-oriented power control policy. The defined action space comprises the joint selection of RB and received power level, promoting optimal outcomes. Agent coordination is enhanced through state-informed interactions from the BS, obviating the need for explicit message passing. A systematic approach based on user dynamics is presented to derive a feasible reward function that ensures relevance and applicability. The integration of Dueling architecture with double DQN improves generalization in the learning process and helps to reduce overestimation biases associated with Q-values.

2) Autonomous and Distributed PC via Power Pool: We have designed a RB-oriented PP for each RB, which allows for a distributed open-loop PC (DPC) strategy during the execution phase. To achieve optimal received power levels for each PP, GF users in the network act independently as agents. They learn and implement a policy that guides them to adjust their transmit power adaptively to ensure that interference remains below the QoS threshold of GB users. Additionally, the proposed algorithm is executed without any message exchange or online coordination among the users (agents). The PP design brings significant benefits, especially for new users who join the network without prior training. Simulation results have shown a significant increase in spectral efficiency, with a 20.19% improvement compared to the conventional method.

3) Performance Analysis: We demonstrate that our proposed algorithm is scalable to large-scale IoT networks with minimal signal overhead. Moreover, to reduce training time, we eliminate the received power levels (invalid actions) that cannot be opted for uplink transmission due to users' transmit power constraints. The numerical results show that agents received more rewards when using the network-centered reward function compared to the self-centered and cluster-centered approaches. We demonstrate that MA-Dueling DDQN performs equivalently to MA-DDQN in networks with fewer

TABLE I: Table of Notations

Symbol	Definition	Symbol	Definition
$\mathbf{U}$	The set GF IoT users	$\mathbf{V}$	The set of GB IoT users
R	Cell radius	N_G	Number of (GF, GB) active users
B	Total bandwidth	M	Number of orthogonal sub-channels
$r_{m,j}$	Distance between the BS and GF IoT user j	$r_{m,i}$	Distance between the BS and GB user i
B_s	Sub-channel bandwidth	n_0	Additive white Gaussian noise
P_i^{GB}	Transmit power of i-th GB user	h_i^{GB}	Channel gain of i-th GB user
P_j^{GF}	Transmit power of j-th GF user	h_j^{GF}	Channel gain of j-th GF user
$\gamma_{m,i}^{\mathrm{GB}}$	SINR of i -th GB user on sub-channel m	$\gamma_{m,j}^{\mathrm{GF}}$	SINR of j -th GF user on sub-channel m
$R_{m,i}^{\mathrm{GB}}$	i-th GB user data rate on sub-channel m	$R_{m,j}^{\mathrm{GF}}$	j-th GF user data rate on sub-channel m
τ	Target data rate threshold for GB users	$\bar{\tau}$	Target data rate threshold for GF users
$\mathbf{Q}_m$	Set of GF users share sub-channel m with GB users	ϕ_m	Interference threshold of GB user on sub-channel m
$\mathbf{N}$	The set of agents	$k_{m,j}$	j-th GF user select sub-channel m
C	Network throughput	L_s	Maximum GF users on a sub-channel
$\mathbf{P}_t$	Matrix for received power levels	P_{NP}	Number of available power levels
$\mathbf{K}_t$	Matrix for sub-channel selection	T_s	Total duration of long-term communication

states and actions but outperforms in environments with more extensive state and action spaces. Furthermore, our proposed algorithm outperforms existing SGF-NOMA systems and pure GF-NOMA IoT networks in terms of throughput. Finally, we investigate the impact of varying SIC error levels on the sum rate and average transmit power of the users.

II. System Model

We consider SGF transmission in IoT networks, as shown in Fig. 1, where a single BS is located at the geographic center with radius R. We assume that two types of users (GB and GF users) equipped with a single antenna are randomly distributed and transmit uplink data to the BS. The set of users $\mathbf{U} = \{1, 2, ...N_{\mathrm{GF}}\}$ represents GF users, whereas GB users are denoted by $\mathbf{V} = \{1, 2, ...N_{\mathrm{GB}}\}$. The locations of GF and GB users are modelled as two homogeneous Poisson point processes with densities λ_{GF} and λ_{GB}. Therefore, the number of GB and GF users follows a Poisson distribution. At one time slot t, the probability of the number of active users N_G (where, $G \in \{\mathrm{GB}, \mathrm{GF}\}$) equalling to $N_t \geq 0$ is given by

$$\Pr\{N_G = N_t\} = \frac{\lambda_G^{N_t} \exp(-N_t)}{N_t!}. \tag{1}$$

The probability density function of a random user with distance r_G is given by $f_r(r_G) = \frac{2r_G}{R^2}$. We define the channel gain and transmit power of i-th GB users as $h_i^{\mathrm{GB}} = |h_i|^2 (r_{i,\ \mathrm{GB}})^{-\alpha}$ and P_i^{GB} respectively. Similarly, the channel gain of j-th GF user with transmit power P_j^{GF} is given as $h_j^{\mathrm{GF}} = |h_j|^2 (r_{j,\ \mathrm{GF}})^{-\alpha}$. The h_i, h_j, r_i, r_j, and α are the small-scale Rayleigh fading of user $i \in \mathbf{V}$ and user $j \in \mathbf{U}$, communication distances of user i and user j, and path loss exponent, respectively. Table I summarises all of the notations and their definitions for clarity.

A. SGF-NOMA Transmission

Note that a large portion of IoT users in mMTC does not require ultra-high data rates [5]. The conventional GB transmission is based on prior handshakes with the BS, which provides limited connectivity and more capacity for most IoT applications than required. This extra capacity can be utilized to enhance the connectivity via GF transmission that forms SGF-NOMA transmission. More specifically, in the SGF-NOMA scheme, GB and GF users share the same or a part of the same RB for uplink transmission. Assuming the total number of orthogonal RBs (sub-channels) is M, the combined information received at the BS on sub-channel m in a time slot t_s is

$$y_m(t_s) = \sum_{i=1}^{N_{\mathrm{GB},m}} \sqrt{P_{m,i}^{\mathrm{GB}}(t_s)} h_{m,i}^{\mathrm{GB}}(t_s) x_{m,i}(t_s) + \sum_{j=1}^{N_{\mathrm{GF},m}} \sqrt{P_{m,j}^{\mathrm{GF}}(t_s)} h_{m,j}^{\mathrm{GF}}(t_s) x_{m,j}(t_s) + n_0(t_s), \tag{2}$$

where $N_{\mathrm{GB},m}$ and $N_{\mathrm{GF},m}$ are the numbers of GB and GF users in the m-th sub-channel, respectively. In the m-th sub-channel, the $x_{m,i}$ is the transmitted signal from the i-th GB user, and the $x_{m,j}$ is that from the j-th GF user. The n_0 is the additive white Gaussian noise for each sub-channel with zero mean and variance σ^2.

B. Signal Model

We assume that the GB users have the highest priority (e.g., a sensor for healthcare monitoring) and provide the strongest received power at the receiver. Thus, the BS always decode the GB user in the first stage of SIC to avoid long latency. After that, the BS turns to decode the GF users according to the received power strength order [2]. To simplify the analysis, this work considers a typical scenario that each RB has one GB user[1]. Moreover, we allow a random number of GF users to be clustered with a GB user in a given RB. Therefore, the received power strength order, i.e., the decoding order, at the BS can be expressed as

$$P_{m,1}^{\mathrm{GB}} h_{m,1}^{\mathrm{GB}}(t_s) \geq P_{m,1}^{\mathrm{GF}} h_{m,1}^{\mathrm{GF}}(t_s) \geq P_{m,2}^{\mathrm{GF}} h_{m,2}^{\mathrm{GF}}(t_s) \cdots\cdots \geq P_{m,N_{\mathrm{GF},m}}^{\mathrm{GF}} h_{m,N_{\mathrm{GF}}}^{\mathrm{GF}}(t_s). \tag{3}$$

[1]We allocate a dedicated RB to a GB user. We can multiplex more than one GB user in each NOMA cluster; however, if we group more than one GB user into one NOMA cluster, we are required to satisfy the QoS of multiple GB users.

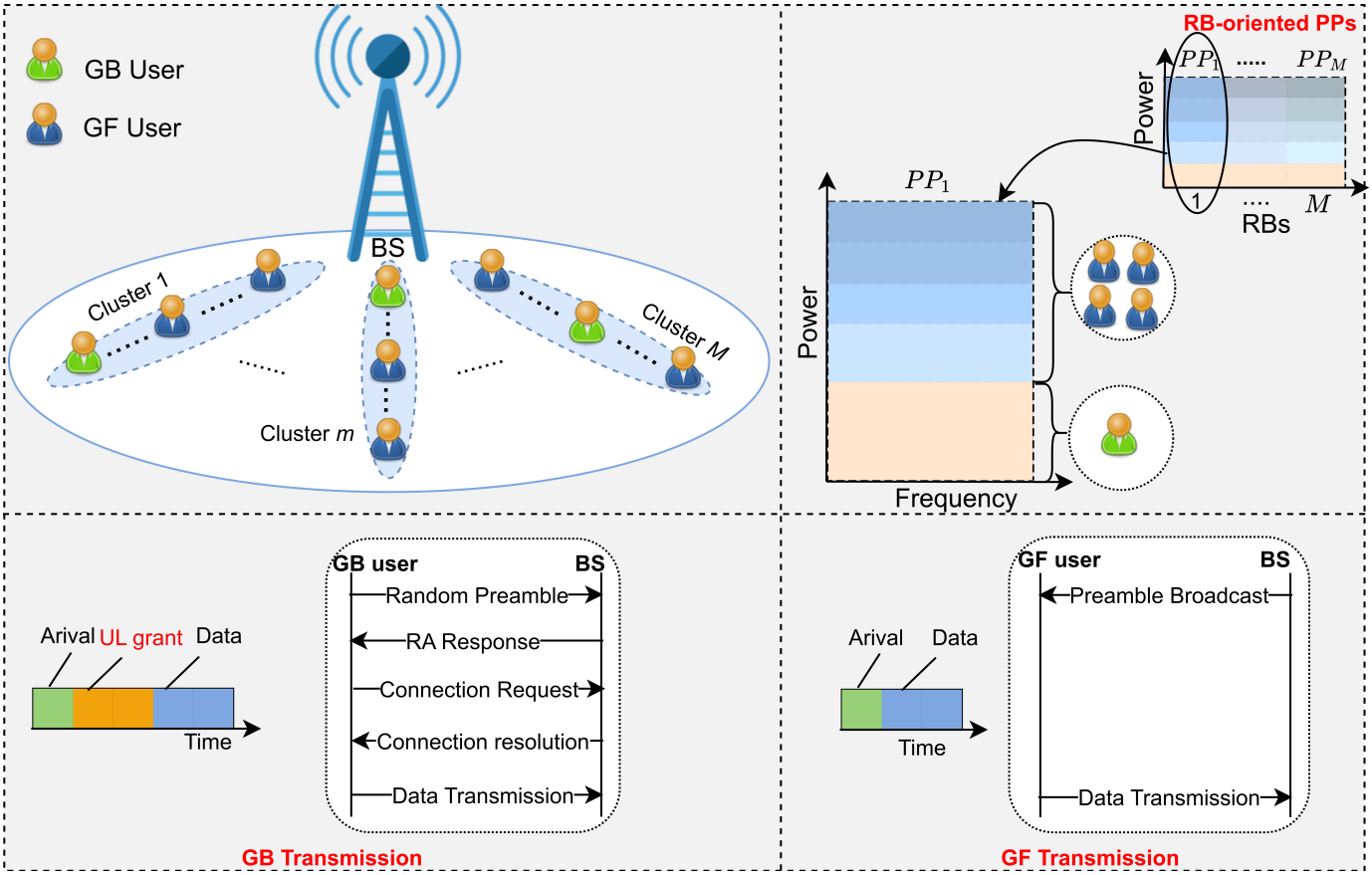


Fig. 1: An illustrative structure of the cluster-based SGF-NOMA IoT network: Top-left sub-figure shows the distribution of GF and GB users in different NOMA clusters. Top-right sub-figure represents the power level of GF and GB users on the same RB from PP_1. The bottom-left sub-figure shows the GB transmission, and the bottom-right sub-figure shows the GF procedure.

This work considers imperfect SIC. In real scenarios, perfect SIC is unfeasible due to error propagation. As a result of imperfect SIC, signals from $j-1$ interferers cannot be decoded perfectly in a NOMA cluster, leaving residual noise for user j. Let the original signal from j-th user on sub-channel m is $y_{m,j}(t_s) = \sqrt{P^{\mathrm{GF}}_{m,j}(t_s)} h^{\mathrm{GF}}_{m,j}(t_s) x_{m,j}(t_s)$ and the estimated signal at the BS can be represented as $\hat{y}_{m,j}$. After SIC, the residual interference $I^{\mathrm{SIC}}_{m,j}$ to j-th user on sub-channel m is

$$\begin{aligned} I^{\mathrm{SIC}}_{m,j} &= \sum_{\hat{j}=1}^{j-1,m} |y_{m,\hat{j}}(t_s) - \hat{y}_{m,\hat{j}}(t_s)|^2 \\ &= \sum_{\hat{j}=1}^{j-1,m} P^{\mathrm{GF}}_{m,\hat{j}}(t_s) |h^{\mathrm{GF}}_{m,\hat{j}}(t_s)|^2 |x_{m,\hat{j}}(t_s) - \hat{x}_{m,\hat{j}}(t_s)|^2. \end{aligned} \tag{4}$$

After cancelling the $\hat{j}$-th user on sub-channel m, the fractional error is a random variable and approximated by a Gaussian distribution with variance σ^2_ε like [19] and can be given as $\varepsilon_{m,\hat{j}} = \mathbb{E}\{|x_{m,\hat{j}}(t_s) - \hat{x}_{m,\hat{j}}(t_s)|^2\}$. The interference due to residual error that j-th user may have on sub-channel m can be expressed as

$$I^{\mathrm{SIC}}_{m,j} = \sum_{\hat{j}=1}^{j-1,m} P^{\mathrm{GF}}_{m,\hat{j}}(t_s) |h^{\mathrm{GF}}_{m,\hat{j}}(t_s)|^2 \varepsilon^2_{m,\hat{j}}. \tag{5}$$

The signal-to-interference-plus-noise ratio (SINR) for the i-th GB user on sub-channel m in time slot t_s is given by

$$\gamma^{\mathrm{GB}}_{m,i}(t_s) = \frac{P^{\mathrm{GB}}_{m,i} h^{\mathrm{GB}}_{m,i}(t_s)}{\sum_{j=1}^{N_{\mathrm{GF},m}} P^{\mathrm{GF}}_{m,j} h^{\mathrm{GF}}_{m,j}(t_s) + \sigma^2}. \tag{6}$$

The SINR of the j-th GF user can be expressed as

$$\gamma^{\mathrm{GF}}_{m,j}(t_s) = \frac{P^{\mathrm{GF}}_{m,j} h^{\mathrm{GF}}_{m,j}(t_s)}{\sum_{j'=j+1}^{N_{\mathrm{GF},m}} P^{\mathrm{GF}}_{m,j'} h^{\mathrm{GF}}_{m,j'}(t_s) + I^{\mathrm{SIC}}_{m,j} + \sigma^2}. \tag{7}$$

To guarantee the SIC process and maintain the QoS of GB and GF users, the following constraints are applied:

$$R^{\mathrm{GB}}_{m,i}(t_s) = B_s \log_2(1 + \gamma^{\mathrm{GB}}_{m,i}(t_s)) \geq \tau, \tag{8a}$$

$$R^{\mathrm{GF}}_{m,j}(t_s) = B_s \log_2(1 + \gamma^{\mathrm{GF}}_{m,j}(t_s)) \geq \bar{\tau}, \tag{8b}$$

where $R^{\mathrm{GB}}_{m,i}(t_s)$ and $R^{\mathrm{GF}}_{m,j}(t_s)$ is the data rate of GB users i and GF user j in time slot t_s, respectively. Furthermore, τ is the required target data rate to ensure the QoS of GB users, and $\bar{\tau}$ is the target threshold for GF users. The B_s is the bandwidth of each sub-channel obtained from $B_s = B/M$, where B is the total bandwidth. Although (8b) is not necessary for GF transmission, it is important for the PP design since this constraint is able to limit the number of potential GF users for each RB, which enhances the connectivity of GF users.

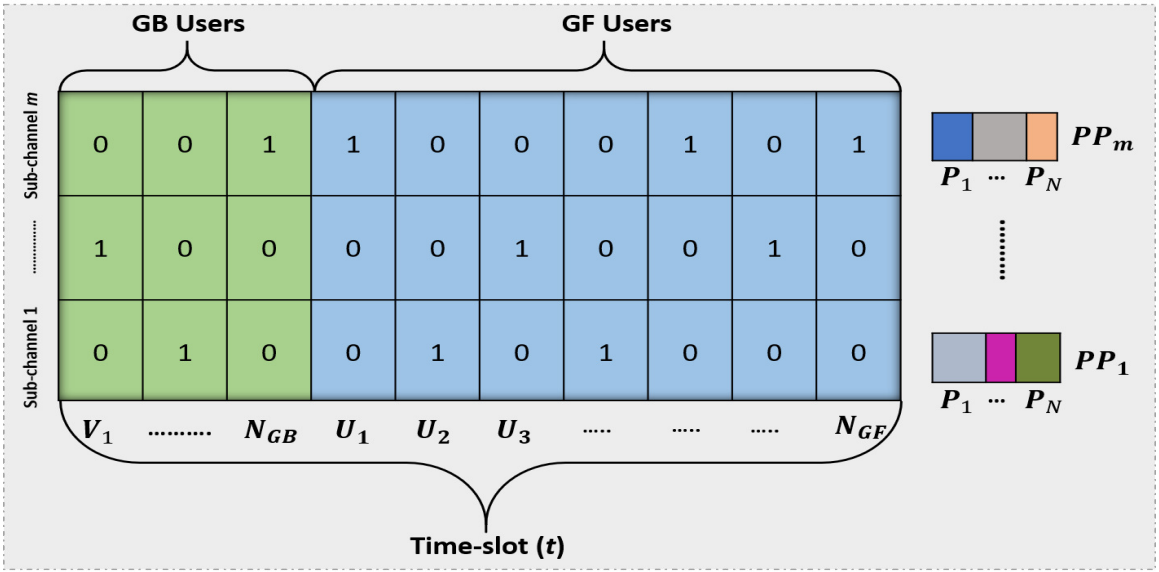


Fig. 2: An illustrative structure of GF and GB users sharing the same RB and PP against each RB.

C. RB-Oriented Power Pool

In conventional SGF-NOMA, users transmit with fixed power. However, determining the optimal power for each user requires a closed-loop PC, which can be costly for IoT applications. This paper demonstrates the capability of MA-DRL techniques to achieve distributed open-loop PC by generating a PP for each RB, i.e., $\{\mathbf{PP}_1, \mathbf{PP}_2, \ldots, \mathbf{PP}_M\}$, where $\mathbf{PP}_m \subset \mathbf{P}_t = \{P_1, P_2, \cdots, P_{NP}\}$ as shown in Fig. 2. In each sub-channel, there is a GB user with varying levels of QoS requirements. Since GB users have diverse QoS requirements, the redundant resources allocated to GF users are unique in each sub-channel. Therefore, these various redundant resources are utilized in designing PPs. As a result, the PPs are specific to each GB user's QoS. We assume that the BS broadcasts these $\mathbf{PPs}$ to GF users in the network. A GF user selects a power level randomly from the $\mathbf{PP}$ associated with the chosen RB. After selecting a power level, each GF user adjusts its transmit power to the specified level for uplink transmission. For example, if a GF user j wants to transmit on sub-channel 1, it selects one received power level from $\mathbf{PP}_1$ for uplink transmission. Selecting a received power level from $\mathbf{PP}$ corresponds to each RB restrict the interference to a tolerable threshold ϕ that ensures the QoS of the GB users.

Remark 1. *In GF transmission, active users should randomly select a RB and received power without any grant, making MUD complex, and BS needs to accurately estimate the varying number of users transmitting over a specific RB [20]. In order to keep MUD simple using a SIC receiver, the BS can use the PP to estimate the number of users in each RB. More specifically, in a PP, if a received power level is idle [21] (i.e., no packet is transmitted on this power level), the BS can estimate the number of users from the remaining power levels used for transmission.*

Remark 2. *These PPs enable GF users to transmit at the optimal power levels without requiring training. More specifically, new users joining the network receive optimal power levels (PPs) from the BS, and they can select transmission power without prior training. Selecting the power level from the PP prevents training complexity and reduces energy consumption. Furthermore, the specified power levels in the PPs give GF users with the flexibility to select their transmit power, accommodating various QoS needs. GF users can select a power level that meets their specific QoS needs while ensuring that interference remains below the QoS threshold for GB users.*

III. RB-oriented PP Generation and Problem Formulation

The objective is to regulate the received power at the base station by controlling the transmit power of GF users so that each GB user achieves the desired QoS. We assume that a single GB user i is connected to the BS through sub-channel m. Let $\mathbf{Q}_q^m$ represent the number of GF users who share the same sub-channel m with GB user and can be expressed as $\mathbf{Q}_q^m \subset \mathbf{U} = \{q : 0 \leq q \leq N_{\mathrm{GF}}\}$. To design the PPs, and restrict interference for maintaining uplink QoS of GB users, the BS has the following steps, shown in Fig. 3.

- The BS obtains the complete CSI and transmit power of the GB user.
- Leveraging the above information, the BS determines the acceptable received power from GF users and calculates interference threshold ϕ_m a GB user can tolerate and allocate it to the channel m where the user can attain the same performance as in OMA [9].
- Following this, the BS formulates a global power pool $\mathbf{P}_t = \{P_1, P_2, \cdots, P_{NP}\}$ and broadcasts it to GF users, along with the interference threshold.

After receiving the global power pool, GF users choose a RB and a power level from the broadcasted pool. The pivotal element is user training. As users undergo this training, they learn to determine the optimal power levels for their uplink

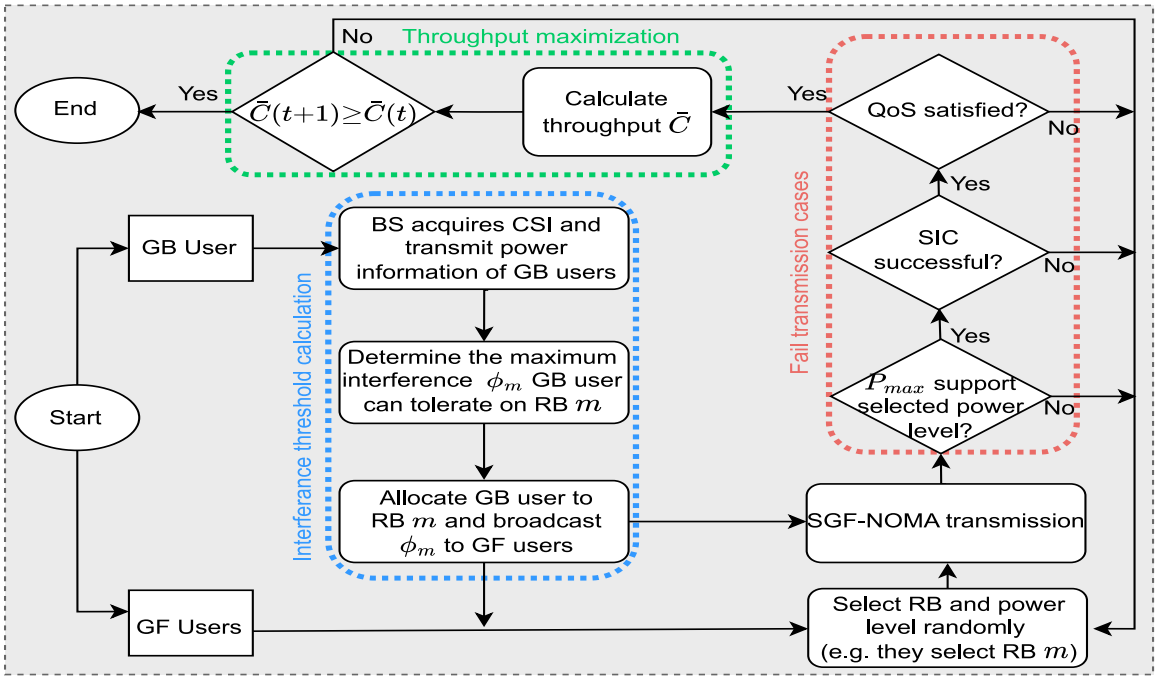


Fig. 3: An illustrative structure of SGF transmission and throughput maximization process.

transmissions, resulting in the formulation of a distinct PP for each RB. We assume that each GF user is allowed to select at most one sub-channel occupied by a GB user in a time slot t_s, as shown in Fig. 2. For this constraint, we define a sub-channel selection variable k, as follows:

$$k_{m,j}(t_s) = \begin{cases} 1, & \text{user } j \in \mathbf{U} \text{ select sub-channel } m \\ & \text{which is occupied by GB user } i \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

We aim to jointly optimize the power allocation $\mathbf{P}_t = \{P_1, P_2, \cdots, P_{NP}\}$ and sub-channel assignment $\mathbf{K}_t = \{k_{1,1}, \cdots, k_{m,j}, \cdots, k_{M,N_{\mathrm{GF}}}\}$ to determine a set of power levels for a given PP to ensure that each GB user can meet its target QoS requirements and maximize the system throughput.

The cumulative capacity can be given as

$$\bar{C}(P, k_{m,j}) = B_s \sum_{t_s=1}^{T_s} \sum_{m=1}^{M} \left[\sum_{i=1}^{N_{\mathrm{GB},m}} \log_2 \left(1 + \frac{P_{m,i}^{\mathrm{GB}} h_{m,i}^{\mathrm{GB}}(t_s)}{\sum_{j=1}^{N_{\mathrm{GF},m}} P_{m,j}^{\mathrm{GF}} h_{m,j}^{\mathrm{GF}}(t_s) + \sigma^2} \right) \right.$$
$$\left. + \sum_{j=1}^{N_{\mathrm{GF},m}} k_{m,j} \log_2 \left(1 + \frac{P_{m,j}^{\mathrm{GF}} h_{m,j}^{\mathrm{GF}}(t_s)}{\sum_{j'=j+1}^{N_{\mathrm{GF},m}} P_{m,j'}^{\mathrm{GF}} h_{m,j'}^{\mathrm{GF}}(t_s) + I_{m,j}^{\mathrm{SIC}} + \sigma^2} \right) \right] \tag{10}$$

Based on (10), the optimization problem can be formulated as

$$\underset{\mathbf{P}_t, \mathbf{K}_t}{\text{maximize}} \quad \bar{C}(P, k_{m,j}) \tag{11}$$
$$\text{s.t.} \quad (3) \tag{11a}$$
$$\sum_{j=1}^{N_{\mathrm{GF},m}} P_{m,j \in \mathbf{U}}(t_s) \leq P_{\max}, \ \forall m, \forall t_s, \tag{11b}$$
$$\sum_{m=1}^{M} k_{i,j \in \mathbf{U}}(t_s) \leq 1, \ \forall j, \forall t_s, \tag{11c}$$
$$N_{G,m}(t_s) \geq 2, \ \forall m, \forall t_s, \tag{11d}$$
$$\sum_{m=1}^{M} R_{m,i}^{\mathrm{GB}}(t_s) \geq \tau, \ \forall i, \forall t_s, \tag{11e}$$
$$\sum_{m=1}^{M} R_{m,j}^{\mathrm{GF}}(t_s) \geq \bar{\tau}, \ \forall j, \forall t_s, \tag{11f}$$
$$\sum_{m=1}^{M} N_{\mathrm{GF},m}(t_s) \leq L_s, \ \forall t_s, \tag{11g}$$

where (11a) is the SIC decoding order and GB user is decoding in the first stage of SIC. The maximum transmit power limit of a user j is given in (11b). Constraint (11c) restricts the IoT users to select at most one sub-channel in a time slot t_s, (11d) represents the minimum number of IoT users to form a NOMA cluster. (11e) is the required data rate of GB users to ensure QoS, and (11f) represents the minimum required data rate threshold for GF users. (11g) shows the maximum number of GF users on each sub-channel.

IV. MA-DRL Framework for SGF-NOMA Systems

DRL method aims to find good quality policies for decision-making problems and is able to evaluate the best utility among available actions with no prior information about the system model. DRL algorithms were originally proposed for a single agent interacting with a fully observable Markovian environment with guaranteed convergence to an optimal solution. Recently, MA-DRL has been widely used in more complex environments and shows stronger performance than single-agent DRL algorithms [22].

A. Modelling the Formulated Problem as a Stochastic Markov Game

Stochastic games model the dynamic interactions of players (agents), where the environment changes in response to the players' behavior. Stochastic games progress in stages, during which each player selects the actions available to them in the current state. The chosen action has two effects: 1) it produces a stage reward, and 2) it determines the probability of the next state. Consequently, players (agents) receive a reward or penalty in the current state and strive to attain high rewards in the next state. A Markov Game is an abstraction of the MDP [23]; MDPs are commonly used in modern RL problems to model the interaction between the environment and the agent. An MDP is a tuple of $(\mathbf{N}, \mathbf{S}, \mathbf{A}, R(\cdot), \mathcal{P}(\cdot))$, where $\mathbf{N}$ represents the number of agents, $\mathbf{S}$ represents the set of states in the environment, $\mathbf{A}$ represents the set of actions that can be performed by an agent, $R(\cdot)$ represents the immediate reward signal an agent receives from the environment for a given state-action pair, and $\mathcal{P}(\cdot)$ shows the transition probabilities between states. Agents act in the environment according to a specific policy $\pi(\cdot)$, which determines the probabilities that guide the agent's decision to take an action based on the current state of the environment. In RL algorithms, the agent's objective is to maximize its long-term rewards by iteratively adjusting its policy based on the rewards it receives from the environment after taking action. Briefly, these functions can be expressed mathematically as

$$\begin{aligned}\mathcal{P}(s,a,s') &= \mathbb{P}\left[S_{t+1}=s'|S_t=s, A_t=a\right],\\ R(s,a) &= \mathbb{E}\left[R_{t+1}|S_t=s, A_t=a\right],\\ \pi(s,a) &= \mathbb{P}\left[A_t=a|S_t=s\right],\end{aligned}$$

where S_t represents the state of an agent at a learning step t of an episode. The A_t is the action the agent takes at that step. The R_{t+1} is the reward received by the agent corresponding to the state-action pair. Detailed definitions are given below.

- **Set of Agents $\mathbf{N}$** : We define the GF IoT user as an agent who interacts with the wireless communication environment. Multiple IoT users collaboratively explore the environment and gain experiences for their policy π design. IoT users adjust their policies based on the insights gained from environmental observations. All agents adjust their actions (sub-channel and power level selection) to obtain an optimal policy π^* by maximizing the reward [24]. At each time-step (TS) t, each agent j receive a state$s_j(t)$, and performs an action $a_j(t)$ that forms a joint action $a(t) = (a_1(t), a_2(t), \cdots, a_j(t), \cdots, a_N(t))$.
- **State space $\mathbf{S}$** : All agents collectively explore the wireless environment by observing various states within the environment. More specifically, we represent the data rate of GF users as the current state $s_j(t) \in S_j$ in learning step t as follows

$$S_j = \left\{R^{\mathrm{GF}}_{1,1}(t), R^{\mathrm{GF}}_{2,1}(t), \cdots R^{\mathrm{GF}}_{m,j}(t), \cdots, R^{\mathrm{GF}}_{M,N_{\mathrm{GF}}}(t)\right\}, \tag{12}$$

  where $R^{GF}_{m,j}$ is the data rate of GF user j on sub-channel m and depends on previous time slot $(t-1)$.
- **Action Space $\mathbf{A}$** : We define the action of a GF user j as a selection of power level and sub-channel. The action space of agent j can be expressed as

$$A_j(t) = \{1, 2, \cdots pm, \cdots, P_{NP}M\}. \tag{13}$$

  We use a set of discrete power levels $\mathbf{P}_t = \{P_1, P_2, \cdots, P_{NP}\}$. Agent j is only allowed to select one power level in time slot t to update its transmit power strategy. All agents have same action space $[A_1 = A_2 = \cdots = A_j = \cdots = A_N]$ for $\forall j \in \mathbf{N}$. Action space dimension is $M \times P_{NP}$, where M is the number of sub-channels and P_{NP} is the number of available discrete power levels.
- **Reward Engineering** Re : The reward function evaluates the actions of an agent as either positive or negative. The design of the reward function is not a trivial task. As it directly impacts the optimization function and accelerates the learning process [25]. Based on the optimization problem under consideration, we present a systematic approach for developing a viable reward function by incorporating user dynamics (user behavior).

1) Self-Centred: Agents exhibiting this type of behavior are short-sighted and act in a completely selfish manner. The agent only considers its own interests and chooses actions that may harm other users' rewards in order to enhance its own reward. For example, if an agent j (IoT user) receives its data rate as a reward, it will select a high power (action) to maximize the reward. However, selecting a high power level creates intense intra-RB interference, which degrades the rewards of other users. Thus, agents following this approach receive varying rewards based on their individual behavior, leading to a greedy strategy. The reward function for such an approach can be defined as

$$r_j(t) = \begin{cases} R^{\mathrm{GF}}_{m,j}(t), & \text{if } R^{\mathrm{GF}}_{m,j}(t+1) \geq R^{\mathrm{GF}}_{m,j}(t) \\ & \text{and ensure constraints given} \\ & \text{in (11a)-(11g),} \\ 0, & \text{otherwise.} \end{cases} \tag{14}$$

2) Cluster-Centred: Agents exhibiting cluster-centered behavior have a wider field of vision and make decisions based on the rewards associated with their clusters (RBs). According to the problem formulation, a user needs to balance its power within its RB. Thus, users should select the power level that minimizes intra-RB interference and maximizes cluster throughput. Agents who receive cluster throughput as a reward only prioritize their selected RB and focus on maximizing cluster throughput. For example, a user creating high interference in one RB might

be a low-interference user in another RB.

$$r_j(t) = \begin{cases} C, & \text{if } C(t+1) \geq C(t) \text{ and ensure} \\ & \text{constraints given in (11a)-(11g),} \\ 0, & \text{otherwise,} \end{cases} \tag{15}$$

where C is the cluster throughput given by $C = \sum_{j=1}^{N_{\mathrm{GF},m}} R_{m,j}^{\mathrm{GF}}(t)$.

3) Network-Centred: Finally, agents with network-centered behavior receive a global reward (such as network throughput in our case) and provide support beyond their individual interests. Therefore, users select the power levels and sub-channels that maximize network throughput (reward). Based on the problem formulation, this type of reward informs the users that we desire to optimize (maximize $\bar{C}$) the network throughput. Therefore, all agents (IoT users) coordinate with each other to adjust their actions, as the IoT user selects a NOMA cluster and power level that either increases or decreases intra-RB interference. Markov games, in which all agents receive the same reward, are known as Team Markov Games [26]. The reward function of each agent $j \in \mathbf{N}$ at learning step t is given by

$$r_j(t) = \begin{cases} \bar{C}, & \text{if } \bar{C}(t+1) \geq \bar{C}(t) \text{ and ensure} \\ & \text{constraints given in (11a)-(11g),} \\ 0, & \text{otherwise.} \end{cases} \tag{16}$$

We have included (11a) in reward function to ensure the given decoding order.

- **Transition Probability** $\mathcal{P}$: The state transition probability function is the probability of transitioning to next state $s(t+1)$ after taking a joint action $a(t)$ in current state $s(t)$.

Remark 3. *After decoding all users' information, the BS learns the data rates of GF users. As a result, the state $s(t)$ and the reward $r(t)$ are available at the BS in each time slot. Therefore, providing data rates of users to the agents as a state can prevent direct collaboration among the agents, thereby reducing signaling overhead and energy consumption.*

We define Q function $Q_j^{\pi}(s_j(t), a_j(t))$ associated with policy π as the expected cumulative discounted reward for each agent j after taking action a_j in state s_j i.e.,

$$Q_j^{\pi}(s_j, a_j) = \mathbb{E}^{\pi}\left[Re(t) \middle| s_j(t) = s, a_j(t) = a\right], \tag{17}$$

where Re is the long-term accumulated and discounted reward and calculated as

$$Re = \sum_{k=0}^{K} \beta^k r^{(t+k+1)}, \qquad 0 < \beta \leq 1, \tag{18}$$

where β, k and K represent the discount factor, epoch and maximum epoch, respectively. The policy π map the state $s(t)$ to the corresponding Q-value under action $a(t)$. All agents aim to maximize the expected reward, which leads them to derive an optimal policy π^*. Once the optimal Q function $Q^*(s,a)$ is obtained, each agent determines an optimal policy π^* such that $Q^*(s,a) \geq Q(s,a)\ \forall s \in S$ and $a \in A$. In a stochastic TMG, the joint optimal policy is known as Nash equilibrium (NE) and can be described as $\pi^* = (\pi_1^*, \pi_2^*, \cdots, \pi_N^*)$. The NE is a combination of policies from all agents, with each policy representing the best retaliation to other policies. Therefore, in a NE, each agent's action is the best response to the other agent's action choice.

The classic Q learning algorithm [23] maintains a Q-table to record Q-values for each state-action pair. However, in the IoT scenario, the size of the Q-table increases with the expanding state-action spaces (i.e., an increase in IoT users), making Q-learning costly in terms of memory and computation. Therefore, the Deep Q learning algorithm [27] is proposed to overcome the aforementioned problem by integrating Q learning with a Deep Neural Network (DNN) with weights denoted as θ for Q function approximation, represented as $Q(s,a;\theta)$. In MA-DRL, each agent consists of a primary (online) network, a target network (both networks with the same architecture), and a replay memory. During the training phase of DNN, the learnable parameters (weights and biases) are updated based on the system's transition history, which consists of a tuple $(s_j(t), a_j(t), r(t), s_j(t+1))$. This process enhances the accuracy of the Q-function approximation. In a learning step t, each agent j inputs the current state s_j to the DQN and receives Q-values corresponding to all actions as output. The agent selects the action with the highest Q-value and obtains an experience in the form of a tuple $(s_j(t), a_j(t), r(t), s_j(t+1))$, which is then stored in the replay memory. To update the weights θ of the target Q-network, a mini-batch of data is randomly sampled from the replay memory. The target value generated by the target Q-network from a randomly sampled tuple is

$$y_j(t) = r(t) + \beta \underset{a_j(t+1) \in A_j}{\operatorname{argmax}} Q(s_j(t+1), a_j(t+1); \theta). \tag{19}$$

In DQN, agents use the same Q-values for both action selection and action evaluation. This leads to a Q-value overestimation problem and causes the algorithm to converge with a non-optimal solution, as the max operator uses the same value for both purposes. To address this issue and enhance the learning efficiency of agents, the following versions of DQN are proposed.

B. Double Deep Q-Network Algorithm

The DDQN [28] prevents the aforementioned problem by decoupling the max operation in the target network for action selection and action evaluation. More specifically, we use two neural networks (NNs) DQN_1 and DQN_2, where DQN_1 is used to select actions and DQN_2 is used to evaluate the corresponding Q-value of those actions. For DDQN, the target

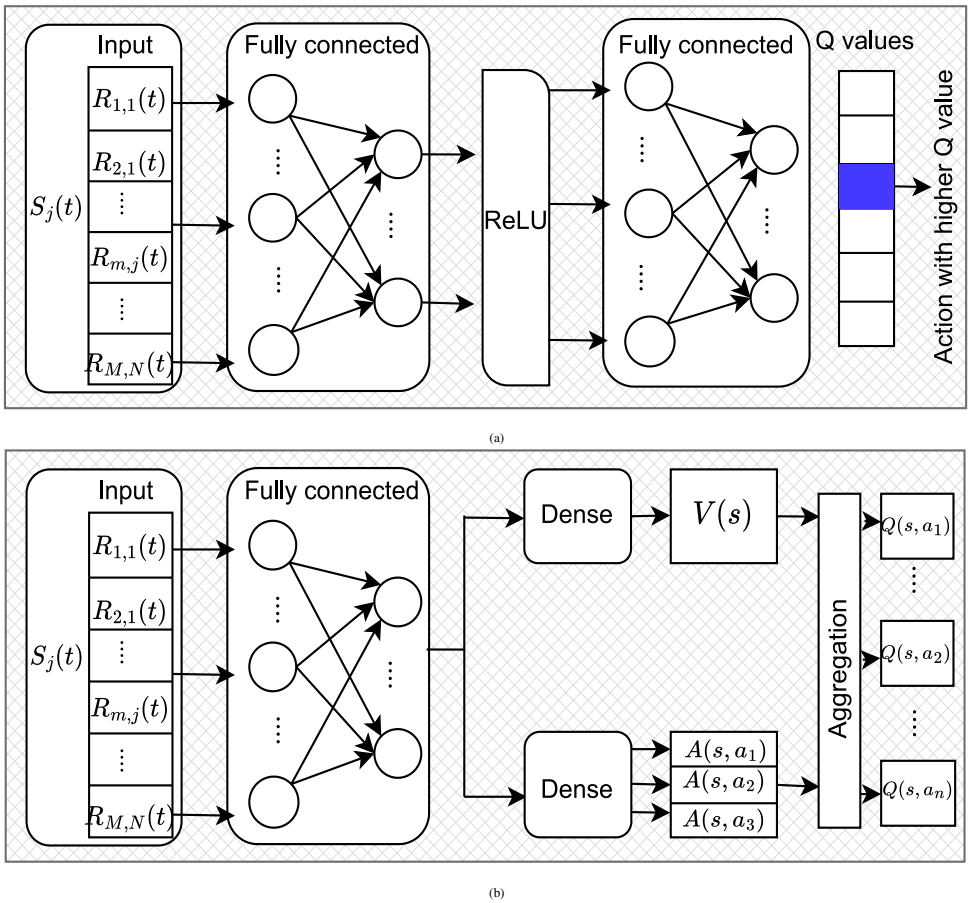


Fig. 4: DNN structures of the proposed algorithms: (a) DNN structure for DDQN algorithm, which consist of input layer followed by hidden layers and an output layer. (b) DNN structure for Dueling DDQN algorithm.

of DQN can be rewritten as:

$$y_j(t) = r(t) + \beta Q\Bigg(s_j(t+1), \underset{a_j(t+1)\in A_j}{\operatorname{argmax}} Q\Big(s_j(t+1), a_j(t+1); \theta\Big); \theta\Bigg). \quad (20)$$

Using a variant of stochastic gradient descent (SGD), the primary Q-network can be trained by minimizing the loss function.

$$Loss(\theta) = (y_j(t) - Q_j(t)(s_j(t), a_j(t); \theta))^2. \quad (21)$$

The proposed MA-DDQN algorithm for power and sub-channel selection with proposed DNN architecture depicted in Fig. 4(a) is given in **Algorithm 1**.

C. Dueling Double Deep Q Network Algorithm

The conventional DQN algorithm calculates the value of each action in a given state. However, different policies may lead to the same value function in certain states. This phenomenon may impede the learning process of identifying the best action in a given state. Dueling architectures have the advantage of efficiently generalizing state values across similar states. As the value approximations are decoupled from the action advantages, the value function can be shared and reused across different actions, reducing redundancy. As a result, the generalization capabilities are improved, and computational resources are used more efficiently. The dueling DQN [29] is an enhanced version of DQN. In this model, the Q-network consists of two streams (sequences) Q-function (i.e., the state-action value function is decomposed), namely the state value function $V_\pi(s)$ and the advantage function $A_\pi(s,a)$. This decomposition of the Q-function accelerates convergence and improve the efficiency. The value function $V_\pi(s)$ represents the expected return from a particular state under policy π, while the advantage function $A_\pi(s,a)$ quantifies the relative importance of taking a specific action compared to other actions in a given state. The output of the dueling network is obtained by combining these two streams to create an aggregate module and a single output Q-function.

$$Q_\pi(s,a;\theta,\theta^V,\theta^A) = V_\pi(s;\theta,\theta^V) + A_\pi(s,a;\theta,\theta^A), \quad (22)$$

where, θ, θ^V, and θ^A represent the parameters of the common network, the value stream parameters, and the advantage stream parameters, respectively. Practically, the agent cannot distinguish between $V_\pi(s)$ and $A_\pi(s,a)$. Since the agent may not be able to obtain a unique solution for $Q_\pi(s,a;\theta,\theta^V,\theta^A)$, it may be unidentifiable and result in poor performance. To solve this problem, the Q-values for each action a in state s

Algorithm 1 Proposed MA-DDQN based SGF-NOMA Algorithm

Initialize primary network with random weights θ
initialize target Q-network with same weights as primary network
Initialize replay memory with size Z, and other training parameters β, ϵ
for episode = 1 to M **do**
  reset initial state of the environment
  for time-step = 1 to N **do**
    Input state $s(t)$
    Take joint action $a(t)$ following $\epsilon-greedy$ policy, receive next state $s(t+1)$ and reward $r(t)$
    Store $s(t), a(t), r(t), s(t+1)$ in replay memory Z
    Sample mini-batches from memory Z
    minimize the loss between the primary network and target network using SGD:

$$\Big[r(t)+\beta Q(s_j(t+1), \underset{a_j(t+1)\in A_j}{\operatorname{argmax}} Q(s_j(t+1), a_j(t+1))) - Q_j(t)(s_j(t), a_j(t); \theta)\Big]^2$$

    if episode% ==U_{steps} **then**
      copy primary network weights to target Q-network weights
    end if
  end for
end for

are generated by the aggregation layer as follows:

$$Q_\pi(s,a;\theta,\theta^V,\theta^A) = V_\pi(s;\theta,\theta^V) + A_\pi(s,a;\theta,\theta^A) - \frac{1}{|\mathcal{A}|}\sum_{a(t+1)} A_\pi(s(t),a(t+1);\theta,\theta^A). \quad (23)$$

The operation of (23) ensures that the primary function of each action in this state remains unchanged and reduces the range of Q-values and excess degrees of freedom, thereby enhancing stability. In particular, it reduces variance in learned action values and enables policy updates to be more stable and reliable. As a result, learning and decision-making processes are smoother and more consistent. The proposed MA-Dueling DDQN algorithm is presented in **Algorithm 2**, and the DNN architecture used is depicted in Fig. 4(b).

D. Proposed MA-SGF-NOMA Algorithms

In our proposed MA-DRL algorithms, each GF user acts as an agent and runs an independent DQN. All agents collectively explore the wireless environment and learn an optimal policy to find a NE. For the exploration and exploitation trade-off, we use the ϵ-greedy method. To fully explore the environment and find the action with the best reward, the agent considers taking a random action with a probability $\epsilon \in [0,1]$. To improve performance, the GF user chooses the best action linked to the highest Q-value in a given state with a probability $1-\epsilon$. In a single learning step t, each GF user j uploads the current state $s_j(t)$ to its primary Q-network and retrieves all the Q-values associated with all actions. The agent then decides its action according to the ϵ-greedy method and takes the joint action $a(t)$. The environment transitions to a new state $s(t+1)$ with probability $\mathcal{P}$, and all agents (GF users) receive the same reward, which is the system throughput. In each time step t, agents create a new experience by interacting with the wireless environment and store it in memory Z as a tuple $(s(t), a(t), r(t), s(t+1))$. To calculate the Q-value of the target network, we randomly sample mini-batches of stored transitions from the replay memory[2]. In each training iteration, to improve the policy π, the primary Q-network is trained by minimizing the error between the actual value and the predicted value using the SGD method with (21). After a set number of training iterations, the primary network weights are copied to the target Q-network. At the end of the training process, each agent j finds an optimal policy π_j^*, which contributes to the formation of the global (joint) optimal policy π^*.

Algorithm 2 Proposed MA-Dueling DDQN based SGF-NOMA Algorithm

Repeating lines 1-10 in **Algorithm 1**
Calculate two streams of the evaluated deep network $V_\pi(s;\theta,\theta^V)$ and $A_\pi(s,a;\theta,\theta^A)$, and combine them using (22)
minimize the loss between the primary network and target network using SGD:

$$\Big[r(t)+\beta Q(s_j(t+1), \underset{a_j(t+1)\in A_j}{\operatorname{argmax}} Q(s_j(t+1), a_j(t+1))) - Q_j(t)(s_j(t), a_j(t); \theta)\Big]^2$$

if episode% == U_{steps} **then**
  copy primary network weights to target Q-network weights
end if

E. Analysis of the Proposed Algorithm

1) Computational Complexity: Floating Point Operations (FLOPs) are used to measure the computational complexity of our algorithm for a single prediction (predicting the power and sub-channel selection policies) or operation. The computational complexity of our model for a single prediction, considering a DNN with L layers and each layer l having g_l nodes, and X as the size of the input layer, is given by:

$$\mathcal{O}\left(2Xg_1 + \sum_{l=1}^{L-1} 2g_l g_{l+1} + \sum_{l=1}^{L} g_l\right).$$

The computational complexity in terms of FLOPs for the whole learning process can be expressed as:

$$\mathcal{O}\left(N_t \cdot M \cdot N \cdot \left(2Xg_1 + \sum_{l=1}^{L-1} 2g_l g_{l+1} + \sum_{l=1}^{L} g_l\right)\right).$$

In the above expression, N_t represents the total number of agents, M denotes the number of episodes, and N signifies the learning steps involved.

2) Signalling Overhead: The overhead is determined by the number of information bits required to provide feedback on sub-channel indicators, channel status data, and a specific user's transmission power over a sub-channel [32]. Moreover, in ML-based approaches, the

[2]The dueling DDQN, an additional step is required: calculating $V_\pi(s;\theta,\theta^V)$ and $A_\pi(s,a;\theta,\theta^A)$ and combining them using (22).

TABLE II: Quantitative comparison of the signalling overhead

Reference	Overhead for U GF and V GB users	Power decision	Optimization method
[9]	$(\underbrace{4V}_{\text{power}} + \underbrace{2V}_{\text{CSI}} + \underbrace{2}_{\text{channel quality threshold}}$ or $\underbrace{2U}_{\text{beacon transmission}})$bits	BS	Conventional
[10]	$(\underbrace{4V}_{\text{power}} + \underbrace{2V}_{\text{CSI}} + \underbrace{2}_{\text{data rate threshold}})$ bits	BS	Conventional
[14]	$(\underbrace{2}_{\text{pilot}} + \underbrace{2V}_{\text{SNR}} + \underbrace{2V}_{\text{CSI}} + \underbrace{2}_{\text{target rate}} + \underbrace{2}_{\text{decoding threshold}} + \underbrace{2}_{\text{SNR threshold}})$ bits		Conventional
[30]	$(\underbrace{2}_{\text{pilot}} + \underbrace{4V}_{\text{power}} + \underbrace{2V}_{\text{CSI}} + \underbrace{2}_{\text{interference threshold}})$ bits	BS	Conventional
[31]	$(\underbrace{10U}_{\text{state}} + \underbrace{2}_{\text{reward}})$ bits	User	RL based
Proposed	$(\underbrace{2U}_{\text{state}} + \underbrace{2}_{\text{reward}} + \underbrace{16}_{\text{PPs}})$ bits	User	RL based

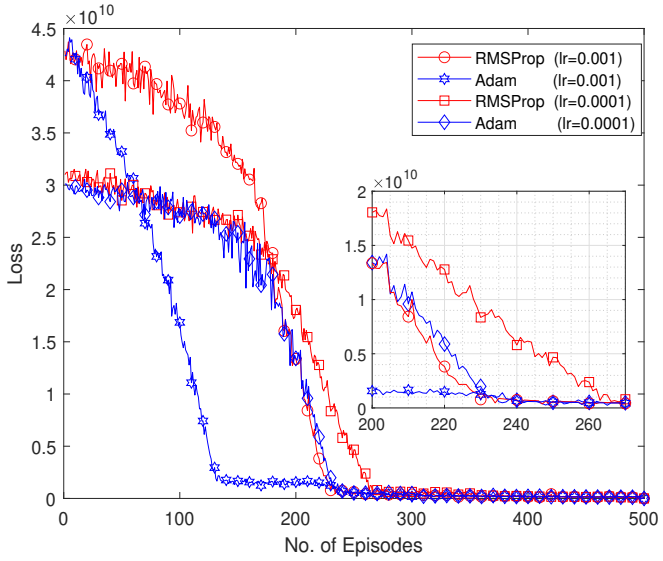
transfer of states and rewards between the agent and the environment also impacts the overhead. Similar to [32], we assume the set $\{16, 4, 4\}$ as the number of information bits to transmit channel status, sub-channel indicators, and the transmission power in the feedback process, and 2 bits for obtaining a single value of a state and reward. Conventional optimization methods typically lead to high overhead because they depend on instantaneous CSI and other threshold information. ML-based approaches, as illustrated in [31], can also result in significant overhead, particularly when the environment states comprise multiple values, such as current channel gain, transmit power, and sub-channel indicator. Therefore, the ML-based approach outlined in [31] also leads to significant overhead because it includes three values in the environment states: current channel gain, transmit power, and sub-channel indicator. The signaling overhead in our proposed Dueling DQN model is significantly influenced by the essential information exchange inherent in the learning process. This includes the data rates of the users within the state, feedback on reward signals, and broadcast of PPs. A detailed quantitative comparison considering these aspects of signaling overhead is presented in Table II, offering a comprehensive understanding of how our model differs from other approaches in terms of the generated overhead.

V. Numericals Results

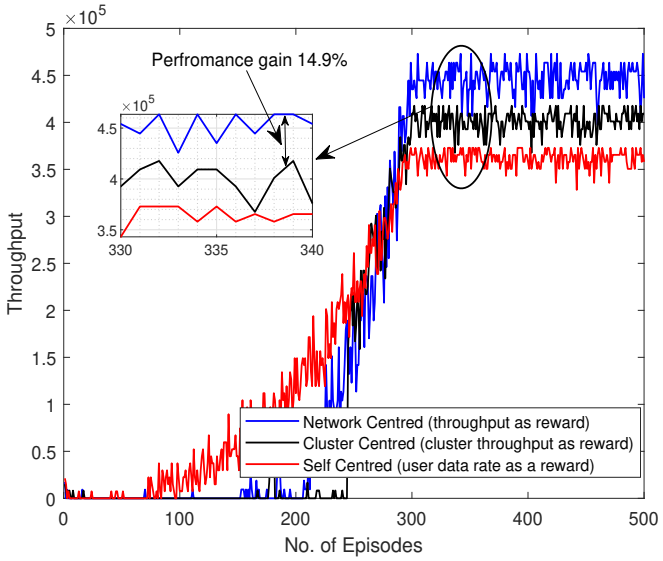
In this section, we evaluate the performance results of our proposed scheme. BS is located at the centre of a circle with a radius of $1000m$. The GF and GB users follows a Poisson distribution across the cell area. We set the path-loss exponent $\alpha = 3.0$, $n_0 = -90$ dBm and the sub-channel bandwidth is 10 KHz [33]. In addition, the GB users transmit data at a fixed power, while GF users select the power from the available power levels $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7,\ 0.8, 0.9\}$ w, and choose among the 3 sub-channels occupied by GB users. To ensure the QoS of GB users, we have set a threshold data rate of $\tau = 4$ bps/Hz. Next, we define the hyperparameters used for the architecture of our DQN algorithm. Our proposed algorithm trains a DQN with three hidden layers, each containing 250, 120, and 60 neurons, denoted as X_1, X_2, and X_3, respectively. We employ this relatively small network architecture for training purposes to ensure that the agent can make decisions (actions) as quickly as possible. We use the Rectified Linear Unit as an activation function to accelerate the learning rate and achieve fast convergence. We set the learning rate to 0.001 and the discount factor β = 0.9 [33]. We set the memory size to $Z = 10000$, with a batch size of 32 and a target network update frequency of 1000. Additionally, the initial value of ϵ is set to 1.0 and gradually decreases to a final value of 0.01 to balance the exploration and exploitation phenomena. The training lasts for 500 episodes, with each episode consisting of 100 time steps.

A. Optimizer and Reward Function Selection

In ML, an optimizer with an appropriate learning rate significantly impacts the model training. An optimizer with a low learning rate progresses slowly, while an optimizer with a high learning rate is susceptible to instability and divergence. Therefore, it is crucial to carefully select an optimizer with the appropriate learning rate, as it can significantly impact the effectiveness of training. Fig. 5(a) displays the average loss value of each agent using Adam and the RMSProp optimizers for two different learning rates. It is evident that the loss value decreases significantly for both optimizers when using a learning rate of 0.001. However, the Adam optimizer performs well and a chieves a minimum loss value within 130 episodes. The loss value of both optimizers decreases slowly with a relatively low learning rate and reaches its minimum in almost 230 episodes for Adam and 270 for RMSProp. It is concluded that the Adam optimizer with a learning rate of 0.001 converges faster than RMSProp with the same learning rate. Fig. 5(b) shows the throughput obtained using various reward functions. The users with a self-centered reward function obtained less throughput compared to those with cluster and network-centered reward functions. In this type of approach, the agents interact with the environment in a greedy manner and converge to a locally optimal solution. In the cluster-centered method, agents aim to maximize their reward within their respective cluster. In the network-centered approach, agents identify actions that contribute to network throughput. This means that agents select power levels that create less interference for other users and choose clusters where they can provide the highest data rates.



(a) Optimizers comparison



(b) Reward function comparison

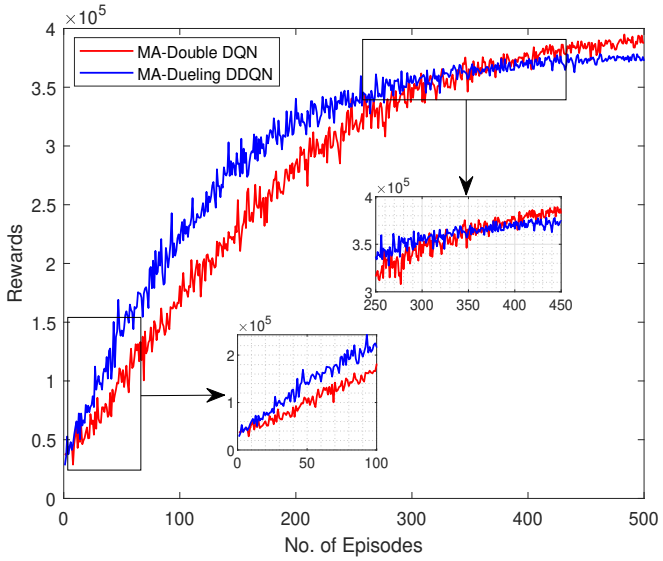
Fig. 5: The loss and reward value: Sub-figure (a) shows the loss value of Adam and RMSProp optimizers with different learning rates. Sub-figure (b) illustrates the throughput obtained using different reward functions.

B. Proposed Algorithm Learning Performance

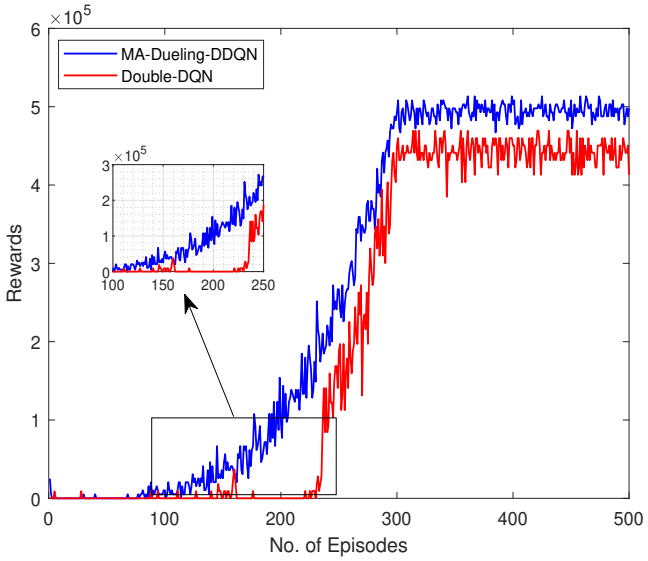
Fig. 6(a) illustrates the comparison of learning efficiency and convergence between MA-DDQN and MA-dueling DDQN-based SGF-NOMA algorithms using a relatively small set of actions and states. Both algorithms perform similarly in terms of learning and reward gain. It can be concluded that MA-DDQN performs better on problems with a limited action space. As a result, MA-DDQN can be applied to problems with a limited set of actions, unlike MA-Dueling DDQN, which necessitates training a distinct neural network to calculate the function estimator. Moreover, identifying the optimal actions and critical states is particularly important in large action and state spaces to enhance the learning process, making MA-Dueling DDQN the most suitable choice. Fig. 6(b) shows the comparison of learning efficiency and convergence between MA-DDQN and MA-dueling DDQN-based SGF-NOMA algorithms across a large number of action and state spaces. The initial performance of both algorithms is the worst due to random action selection during the exploration phase. However, after gaining experience from interacting with the wireless environment, MA-Dueling DDQN demonstrates better learning performance compared to the MA-DDQN algorithm. In the MA-dueling DDQN algorithm, agents learn and refine their policies more rapidly, typically converging after approximately 100 episodes. On the other hand, agents in the MA-DDQN algorithm learn slowly and begin to converge after 220 episodes. This occurs because different policies may result in the same value function in certain states, and this phenomenon hinders the learning process in determining the best action for a specific state. However, the MA-Dueling DDQN generalizes the learning process for all actions and can quickly identify the best actions and important states without having to learn the effects of each action for each state, thus accelerating the learning process for each agent. During the exploitation phase, both algorithms' agents exploit the environment by taking better actions, gradually increasing the reward value, and reaching its maximum in 300 episodes. However, the MA-Dueling DDQN demonstrates rapid learning efficiency and superior performance in terms of reward (system throughput) acquisition.

C. Scalability of the Proposed Algorithm

One of the main challenges with MA-DRL algorithms is scalability. One solution to this problem is to utilize decentralized learning with networked agents, which allows agents to share information about their actions, states, or policies with other agents [34]. However, such communication among agents increases communication overhead and reduces energy efficiency. To minimize communication overhead, our proposed algorithm involves agents indirectly receiving information (data rate) of other users from the BS as a state and updating their policies in a decentralized manner. More specifically, agents (users) are independent learners and cannot communicate with each other directly. This is advantageous for applications with high communication costs or unreliable communications, such as in UAV networks or IoT networks. We illustrated the scalability of our proposed algorithm in Fig. 7(a). It is evident that our proposed algorithm converges in almost the same number of episodes across different numbers of users. With $\mathbf{N} = 9$ agents, the algorithm begins to converge after approximately 255 episodes and reaches its maximum reward value after about 325 episodes. A similar performance can be observed when we increased the density, i.e., the number of agents to $\mathbf{N} = 12$, $\mathbf{N} = 15$, $\mathbf{N} = 24$, and $\mathbf{N} = 30$. Increasing this number further makes it difficult for power-domain NOMA to successfully decode more than 10 users in a NOMA cluster. Furthermore, new agents can be added to the existing trained agents by simply copying the NN parameters of the trained agents. This approach allows for the generalization of the proposed method to diverse scenarios. Therefore, our proposed algorithm is suitable for SGF-NOMA IoT networks with a large number of users.

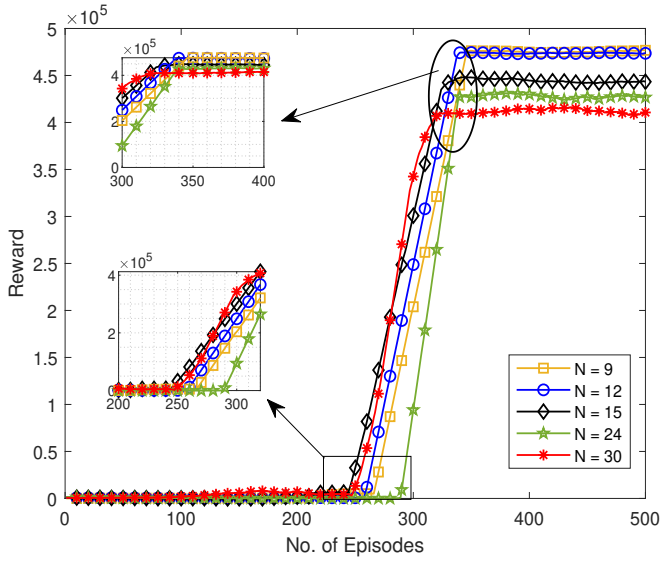


(a) Performance w.r.t. small state and action set

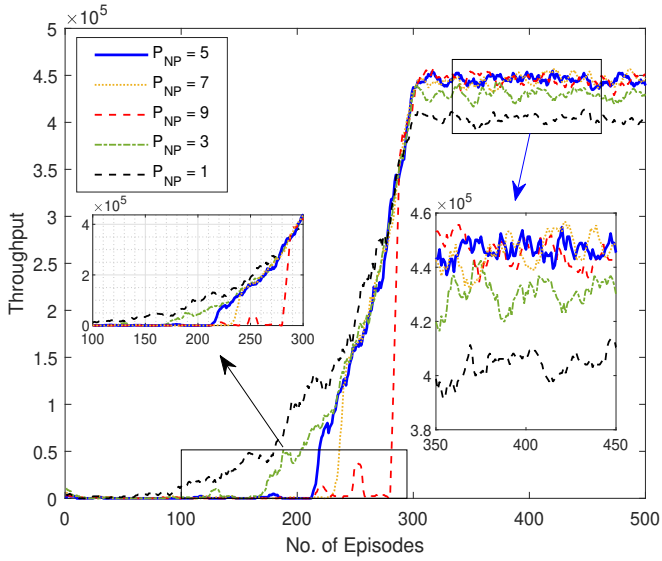


(b) Performance w.r.t. large state and action set

Fig. 6: The convergence comparison: Sub-figure (a) shows the performance against small action and state spaces. Sub-figure (b) represents the convergence comparison against large action and state spaces.



(a) Convergence w.r.t different number of agents



(b) No. of Power Levels vs. throughput

Fig. 7: Scalability and impact of the number of power levels and performance comparison: Sub-figure (a) shows the scalability of our proposed algorithm with increasing number of agents. Sub-figure (b) represents network throughput w.r.t. different number of power levels.

D. Impact of the Number of Power Levels

We discretize the received power into different levels in order to assess the network performance with varying numbers of power levels. Fig. 7(b) illustrates the impact of the number of power levels on network performance in terms of throughput and convergence. A small number of power levels reduces the state and action spaces to $(M \times P_{NP}, (3 \times 1 = 3))$, leading to quick convergence. It can be observed that after approximately 100 episodes, each agent (GF user) discovers an optimal policy and receives a consistent reward in terms of throughput. However, this leads to the lowest network throughput because each GF user has a limited range of power levels to choose from. Network with $P_{NP} = 3$ increases the action space from 3 to 9 (i.e., $3 \times 3 = 9$), which requires more training episodes to explore favorable states and identify an optimal policy. From the figure, it can be seen that after 175 episodes, agents receive a reward in the form of throughput. With higher received power levels, the network throughput increases as each user selects power fairly based on the channel gain, which reduces interference to other users in the cluster. Next, we evaluate network performance w.r.t. $P_{NP} = 5$, which yields the best performance results compared to the other power levels. The throughput increases continuously from 200 episodes, reaches its peak throughput at 300 episodes, and remains stable until the end of the training. When the algorithm has $P_{NP} = 7$ and $P_{NP} = 9$, users spend more time training to explore the environment for optimal actions, while achieving the same throughput as $P_{NP} = 5$. With more power levels, most of the actions (with high power level) become

invalid due to users' transmit power constraints in each sub-channel. Thus, increasing the number of power levels does not always improve system performance. It becomes difficult to determine the best states because the number of actions directly impacts the state space in our proposed method.

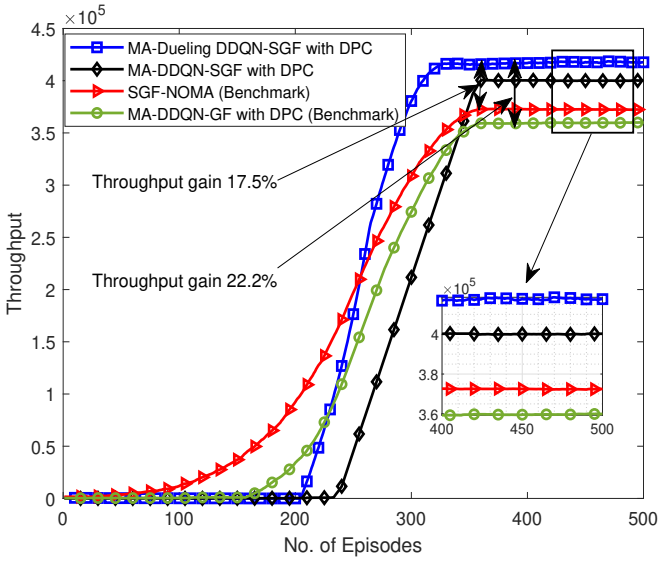
E. Performance Comparison of the Proposed Algorithm

Fig. 8(a) shows the performance improvement achieved by the proposed MA-DDQN-based SGF-NOMA and MA-dueling DDQN-based methods. SGF-NOMA with the DPC mechanism was compared to other methods, including pure GF-NOMA and SGF-NOMA [9]. It is evident that MA-DDQN-based SGF-NOMA and MA-dueling DDQN-based SGF-NOMA with a DPC mechanism outperform benchmarks in terms of throughput. The proposed MA-dueling DDQN-based SGF-NOMA achieved 22.2% and 17.5% higher throughput than pure GF-NOMA and SGF-NOMA, respectively. This is because, in our proposed algorithm, only a subset of GF users are permitted to transmit on a sub-channel exclusively occupied by GB users, resulting in interference within a tolerable threshold for GB users. Unlike the benchmark scheme mentioned in [9], all GF users transmit at a fixed power, regardless of their channel gain. In our proposed algorithm, GF users distribute transmission power based on their channel gain and geographical location. Each GF user acts as an agent to maximize its reward, which is the network throughput. Therefore, GF users select the power level that minimizes interference with other users, forming a NOMA cluster, thereby increasing the system throughput. In the pure GF IoT network, all GF users are allowed to transmit data, leading to strong interference on the sub-channels. This increases the intra-RB interference and results in low network throughput.

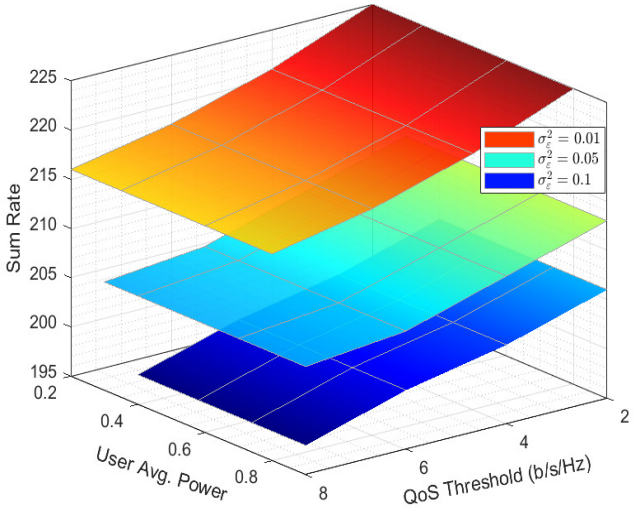
F. SIC Error Impact

SIC performance depends on several factors, including the power level of the interfering signal, complexity of the signal-processing algorithm, and the quality of the receiver hardware. Therefore, power control is a technique used to achieve reliable communication in the presence of imperfect SIC errors. In a system with an imperfect SIC, power control can be implemented by adjusting the transmit power of each user according to the estimated interference level at the receiver. Fig. 8(b) shows the impact of the increasing variance of the Gaussian distribution on the sum rate[3] and the transmit power of users with different QoS requirements. It can be observed that a small variance value and QoS requirements lead to the highest sum rate. It is also important to note that this high sum rate is achieved with minimal average user power consumption. With a slight variation, the distribution will have fewer widely dispersed values and less fluctuation in SIC error, leading to reduced residual interference. As QoS requirements and variance values are further increased, the sum rate decreases, and the average transmission power of the users increases. Due to the increased variation, the residual interference also increases, requiring users to transmit at higher power levels to ensure QoS requirements are met. With high variance and large QoS requirements, the proposed power allocation scheme performs poorly in terms of the sum rate and average user transmission power consumption. However, due to the intelligent power control, the decoding process remains successful even in the presence of imperfect SIC.

[3]For the primary comparisons and evaluations, throughput is the metric we have adhered to. However, as an indicator of the inherent system capability, we have provided additional discussions on the maximum achievable sum rate.



(a) Performance comparison



(b) Impact of error in SIC

Fig. 8: Performance comparison and impact of SIC error level: Sub-figure (a) shows the performance comparison of the proposed MA-Dueling DDQN based SGF-NOMA and MA-DDQN based SGF-NOMA with pure GF-NOMA scheme [4] and SGF-NOMA [9]. Sub-figure (b) shows the impact of the SIC error level on the throughput.

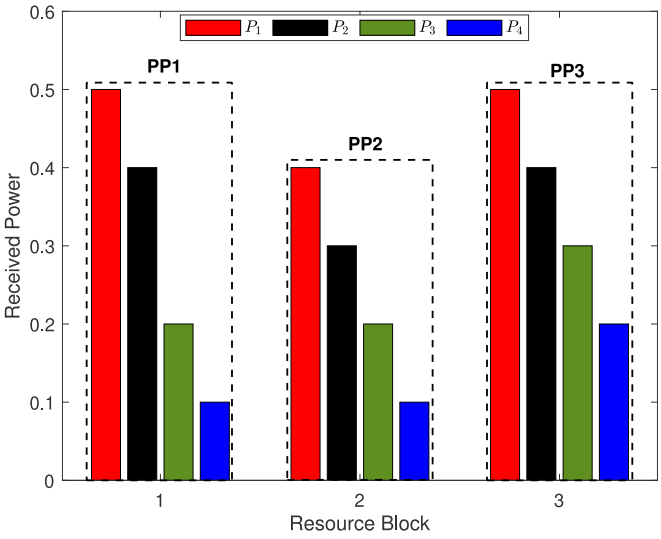
G. The Designed RB Oriented PPs

The designed PPs associated with each sub-channel are shown in Fig. 9(a). We utilized three sub-channels that are pre-occupied by GB users but are available to GF users for

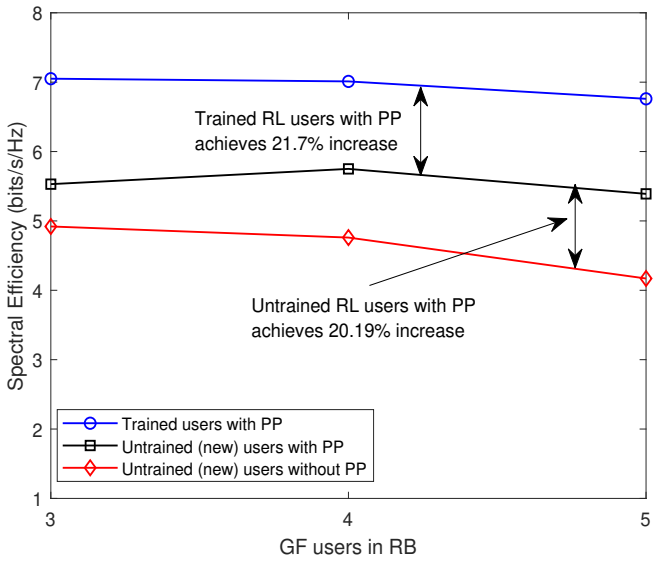
uplink transmission. From the figure, it can be observed that we have identified the optimal received power levels for each PP and mapped them to the corresponding RBs. The BS broadcasts PPs and other information to all GF users in the network. After receiving this information, GF users randomly select the power received from the PP associated with the selected RB. This PP approach enables the DPC, as each GF user chooses an appropriate power level from the available power levels within the PP based on its local information (e.g., channel conditions). Furthermore, the channel conditions and interference levels are susceptible to rapid changes in dynamic wireless environments. However, the DPC enables users to adapt to these changes in real time, ensuring consistent performance. To ensure consistent communication and adapt to changing network conditions, users should regularly monitor their environment and adjust their transmit power as needed. Therefore, the proposed SGF-NOMA provides a distributed open-loop PC with low signaling overhead and low latency.

H. PP Advantages for New Users

Training RL models requires extensive computational resources, which may take hours, days, or even weeks to complete the training process. This limitation can hinder the practicality and scalability of RL applications, especially in resource-constrained environments with time-sensitive or real-time requirements. New users joining the network need training to optimize system performance, which is impractical and wasteful of resources. We demonstrate that new users joining the network can benefit from an RB-oriented PP. Fig. 9(b) illustrates a performance comparison in terms of the spectral efficiency of trained users with PP, untrained (new) users with PP, and untrained users without PP (conventional). When trained users selected the action with the highest Q-value, they achieved the highest spectral efficiency. New users joining the network without training receive PPs containing the optimal received power levels from the BS via a broadcast signal. To transmit uplink signals, users randomly select a RB and adjust the transmission power. Compared to conventional systems, RB-oriented PPs achieve higher spectral efficiency in all load scenarios. The low spectral efficiency of untrained users with PP is due to potential power collisions in comparison to trained users. In this scenario, the BS is unable to decode the signals from colliding users. As the number of users in each RB increases, the probability of collisions also increases, leading to a decrease in spectral efficiency. However, conventional systems (without PPs) suffer from collisions, and they may select suboptimal power levels that cannot guarantee GB users and their own QoS requirements.



(a) PP associated to each RB



(b) Performance with and without PP.

Fig. 9: The designed PPs and performance with and without PPs. Sub-figure (a) shows the PP corresponding to RBs. Sub-figure (b) Represents the performance of trained users with PP and untrained (new) users with and without PP.

VI. Conclusions and Future Research Directions

In this paper, we have proposed an MA-DRL-based SGF-NOMA algorithm to generate a PP and map it to each RB, which enables a distributed open-loop PC. In the proposed scheme, a single user is granted access to the sub-channel through a GB protocol, and GF users are admitted to the same sub-channel via the GF protocol. The designed network-centered reward function provides higher throughput compared to self-centered and cluster-centered functions. We have demonstrated that the proposed algorithm is computationally scalable, regardless of the number of users. Numerical results show that the proposed MA-DRL-based SGF-NOMA outperforms the SGF-NOMA system and networks with pure GF protocols, achieving gains in system throughput of 17.5% and 22.2%, respectively. We have demonstrated the impact of the error level in SIC on network performance and the average transmit power of users. Finally, the benefits of PP are presented to new users. Investigating user fairness in terms of energy consumption and received power collisions is a promising direction for future research. Exploring adaptive and power-efficient implementations of multiple antenna systems in the context of SGF-NOMA is another promising future research direction.

REFERENCES

[1] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. K. Soong, and J. C. Zhang, "What will 5G be?" *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1065–1082, June 2014.

[2] Y. Liu, Z. Qin, M. Elkashlan, Z. Ding, A. Nallanathan, and L. Hanzo, "Nonorthogonal multiple access for 5G and beyond," *Proc. IEEE*, vol. 105, no. 12, pp. 2347–2381, Dec. 2017.

[3] D. Wan, M. Wen, F. Ji, H. Yu, and F. Chen, "Non-orthogonal multiple access for cooperative communications: Challenges, opportunities, and trends," *IEEE Wireless Commun.*, vol. 25, no. 2, pp. 109–117, Apr. 2018.

[4] M. Fayaz, W. Yi, Y. Liu, and A. Nallanathan, "Transmit power pool design for grant-free NOMA-IoT networks via deep reinforcement learning," *IEEE Trans. Wireless Commun.*, vol. 20, no. 11, pp. 7626–7641, 2021.

[5] W. Yi, W. Yu, Y. Liu, C. H. Foh, Z. Ding, and A. Nallanathan, "Multiple transmit power levels based NOMA for massive machine-type communications," 2020, arXiv preprint arXiv:2011.12388. [Online]. Available: https://arxiv.org/abs/2011.12388

[6] S. Ali, N. Rajatheva, and W. Saad, "Fast uplink grant for machine type communications: Challenges and opportunities," *IEEE Commun. Mag.*, vol. 57, no. 3, pp. 97–103, Mar. 2019.

[7] R. Huang, V. W. Wong, and R. Schober, "Throughput optimization in grant-free NOMA with deep reinforcement learning," in *IEEE Global Commun. Conf. (GLOBECOM)*, 2019, pp. 1–6.

[8] L. Liu, E. G. Larsson, W. Yu, P. Popovski, C. Stefanovic, and E. de Carvalho, "Sparse signal processing for grant-free massive connectivity: A future paradigm for random access protocols in the internet of things," *IEEE Signal Processing Mag.*, vol. 35, no. 5, pp. 88–99, Sep. 2018.

[9] Z. Ding, R. Schober, P. Fan, and H. V. Poor, "Simple semi-grant-free transmission strategies assisted by non-orthogonal multiple access," *IEEE Trans. Commun.*, vol. 67, no. 6, pp. 4464–4478, June 2019.

[10] Z. Ding, R. Schober, and H. V. Poor, "A new QoS-guarantee strategy for NOMA assisted semi-grant-free transmission," *IEEE Trans. Commun.*, vol. 69, no. 11, pp. 7489–7503, 2021.

[11] C. Zhang, Y. Liu, and Z. Ding, "Semi-grant-free NOMA: A stochastic geometry model," *IEEE Trans. Wireless Commun.*, vol. 21, no. 2, pp. 1197–1213, 2021.

[12] C. Zhang, Y. Liu, W. Yi, Z. Qin, and Z. Ding, "Semi-grant-free NOMA: Ergodic rates analysis with random deployed users," *IEEE Wireless Commun. Lett.*, vol. 10, no. 4, pp. 692–695, Apr. 2021.

[13] Z. Yang, P. Xu, J. Ahmed Hussein, Y. Wu, Z. Ding, and P. Fan, "Adaptive power allocation for uplink non-orthogonal multiple access with semi-grant-free transmission," *IEEE Wireless Commun. Lett.*, vol. 9, no. 10, pp. 1725–1729, Oct. 2020.

[14] H. Lu, X. Xie, Z. Shi, H. Lei, H. Yang, and J. Cai, "Advanced NOMA assisted semi-grant-free transmission schemes for randomly distributed users," *IEEE Trans. Wireless Commun.*, vol. 22, no. 7, pp. 4638–4653, 2023.

[15] T. Park and W. Saad, "Distributed learning for low latency machine type communication in a massive internet of things," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5562–5576, 2019.

[16] S. K. Sharma and X. Wang, "Toward massive machine type communications in ultra-dense cellular IoT networks: Current issues and machine learning-assisted solutions," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 426–471, 2020.

[17] G. Gui, H. Huang, Y. Song, and H. Sari, "Deep learning for an effective nonorthogonal multiple access scheme," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 8440–8450, 2018.

[18] W. Ahsan, W. Yi, Z. Qin, Y. Liu, and A. Nallanathan, "Resource allocation in uplink NOMA-IoT networks: A reinforcement-learning approach," *IEEE Trans. Wireless Commun.*, vol. 20, no. 8, pp. 5083–5098, 2021.

[19] S. Wang, T. Lv, W. Ni, N. C. Beaulieu, and Y. J. Guo, "Joint resource management for MC-NOMA: A deep reinforcement learning approach," *IEEE Trans. Wireless Commun.*, vol. 20, no. 9, pp. 5672–5688, 2021.

[20] M. B. Shahab, S. J. Johnson, M. Shirvanimoghaddam, and M. Dohler, "Enabling transmission status detection in grant-free power domain non-orthogonal multiple access for massive internet of things," *Trans. Emerging Telecommun. Technol.*, p. 4565–4591, 2022.

[21] W. Yu, C. H. Foh, A. U. Quddus, Y. Liu, and R. Tafazolli, "Throughput analysis and user barring design for uplink NOMA-enabled random access," *IEEE Trans. Wireless Commun.*, vol. 20, no. 10, pp. 6298–6314, 2021.

[22] O. Naparstek and K. Cohen, "Deep multi-user reinforcement learning for distributed dynamic spectrum access," *IEEE Trans. Wireless Commun.*, vol. 18, no. 1, pp. 310–323, Jan. 2019.

[23] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[24] L. Liang, H. Ye, and G. Y. Li, "Spectrum sharing in vehicular networks based on multi-agent reinforcement learning," *IEEE J. Sel. Areas Commun.,*, vol. 37, no. 10, pp. 2282–2292, Oct. 2019.

[25] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *Advances in neural information processing systems*, vol. 30, 2017.

[26] A. Nowé, P. Vrancx, and Y.-M. De Hauwere, *Game Theory and Multi-agent Reinforcement Learning*, M. Wiering and M. van Otterlo, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

[27] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[28] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," 2015, arXiv preprint arXiv:1509.06461. [Online]. Available: https://arxiv.org/abs/1509.06461

[29] Z. Wang, T.Schaul, M.Hessel, H.Van Hasselt, M. Lanctot, and N.De Freitas, "Dueling network architectures for deep reinforcement learning," 2016, arXiv preprint arXiv:1511.06581. [Online]. Available: https://arxiv.org/abs/1511.06581

[30] H. Liu, T. A. Tsiftsis, B. Clerckx, K. J. Kim, K. S. Kwak, and H. V. Poor, "Rate splitting multiple access for semi-grant-free transmissions," 2021, arXiv preprint arXiv:2110.02127. [Online]. Available: https://arxiv.org/abs/2110.02127

[31] J. Chen, L. Guo, J. Jia, J. Shang, and X. Wang, "Resource allocation for IRS assisted SGF NOMA transmission: A MADRL approach," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 4, pp. 1302–1316, 2022.

[32] A. Nouruzi, A. Rezaei, A. Khalili, N. Mokari, M. R. Javan, E. A. Jorswieck, and H. Yanikomeroglu, "Toward a smart resource allocation policy via artificial intelligence in 6G networks: Centralized or decentralized?" 2022, arXiv preprint arXiv:2202.09093. [Online]. Available: https://arxiv.org/abs/2202.09093

[33] J. Zhang, X. Tao, H. Wu, N. Zhang, and X. Zhang, "Deep reinforcement learning for throughput improvement of the uplink grant-free NOMA system," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6369–6379, July 2020.

[34] T. Li, K. Zhu, N. C. Luong, D. Niyato, Q. Wu, Y. Zhang, and B. Chen, "Applications of multi-agent reinforcement learning in future internet: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 2, pp. 1240–1279, 2022.