Quantum Entanglement Path Selection and Qubit Allocation via Adversarial Group Neural Bandits

Yin Huang, Lei Wang, *Graduate Student Member, IEEE*, and Jie Xu, *Senior Member, IEEE*

***Abstract*— Quantum Data Networks (QDNs) have emerged as a promising framework in the field of information processing and transmission, harnessing the principles of quantum mechanics. QDNs utilize a quantum teleportation technique through long-distance entanglement connections, encoding data information in quantum bits (qubits). Despite being a cornerstone in various quantum applications, quantum entanglement encounters challenges in establishing connections over extended distances due to probabilistic processes influenced by factors like optical fiber losses. The creation of long-distance entanglement connections between quantum computers involves multiple entanglement links and entanglement swapping techniques through successive quantum nodes, including quantum computers and quantum repeaters, necessitating optimal path selection and qubit allocation. Current research predominantly assumes known success rates of entanglement links between neighboring quantum nodes and overlooks potential network attackers. This paper addresses the online challenge of optimal path selection and qubit allocation, aiming to learn the best strategy for achieving the highest success rate of entanglement connections between two chosen quantum computers without prior knowledge of the success rate and in the presence of a QDN attacker. The proposed approach is based on multi-armed bandits, specifically adversarial group neural bandits, which treat each path as a group and view qubit allocation as arm selection. Our contributions encompass formulating an online adversarial optimization problem, introducing the EXPNeuralUCB bandits algorithm with theoretical performance guarantees, and conducting comprehensive simulations to showcase its superiority over established advanced algorithms.**

***Index Terms*— Quantum entanglement, path selection, qubit allocation, multi-armed bandits.**

I. Introduction

IN RECENT times, Quantum Data Networks (QDNs) have emerged as a transformative approach, with the potential to reshape information processing and transmission through the evolution of distributed quantum computing [1]. Classical networks have inherent limitations when it comes to ensuring data security during transmission and handling data-intensive processing tasks. QDNs, built upon the principles of quantum mechanics, have the potential to overcome these limitations by leveraging the unique properties of quantum systems, paving the way for unparalleled levels of security, communication efficiency, and computational power [2]. Previous quantum key distribution technique, that aims to generate quantum bits (qubits) and use them to deliver cryptographic keys, can regenerate and retransmit qubits if available [3], while QDNs encode the data information in the data qubits and utilize quantum teleportation technique to avoid the retransmission issue due to no-cloning theorem [4].

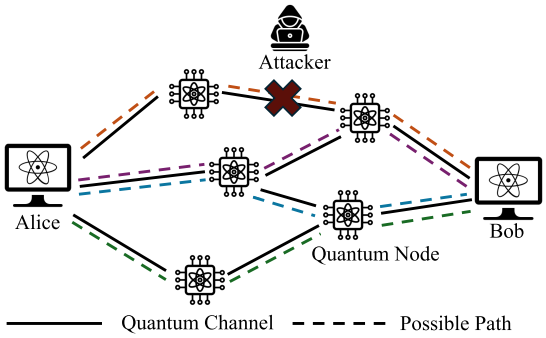


Fig. 1. Quantum data network. There exist four possible paths between Alice and Bob and one attacker aims to disrupt one of them. Note that different possible paths can have different success rates of establishing entanglement connections.

Quantum entanglement stands as a cornerstone in various quantum techniques and applications. Consider two physically adjacent quantum computers, Alice and Bob, where entangled qubit pairs are generated on one end, say Alice. One of these entangled qubits is retained while the other is sent to Bob, via a physical fiber-optic channel, establishing an entanglement link. However, this procedure encounters challenges due to the inherent loss in the optical fiber, resulting in a success rate well below one [5]. Creating an entanglement link is probabilistic and uncertain, relying on the presence of quantum channels between the parties and qubits at both ends [6]. Increasing the number of qubits on both ends and utilizing additional available channels can improve the success rate of the simultaneous attempts made. Nevertheless, quantum channels are limited, and each quantum node has constrained quantum memory for storing qubits, further complicating the process.

As depicted in Figure 1, within QDNs, direct channel connections among quantum computers are often absent. Instead,

Received 3 May 2024; revised 18 September 2024; accepted 31 October 2024; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor D. Elkouss. This work was supported in part by NSF under Grant 2033681, Grant 2006630, Grant 2044991, and Grant 2319780. *(Yin Huang and Lei Wang contributed equally to this work.) (Corresponding author: Jie Xu.)*

The authors are with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32608 USA (e-mail: yin.huang@ufl.edu; leiwang1@ufl.edu; jie.xu@ufl.edu).

Digital Object Identifier 10.1109/TNET.2024.3510550

they are linked through various quantum nodes, known as quantum repeaters. Notably, entanglement links can only be forged between neighboring quantum nodes. To establish a long-range entanglement connection between Alice and Bob, a path must be discovered between them, establishing entanglement links for each successive quantum node along this path and entanglement swapping operations are employed repeatedly at each quantum node along the path [7]. The success rate of establishing such a long-distance entanglement connection is related both to the selected path, such as the length and number of hops of the path and the success rate of each entanglement link on the path and to the allocation of qubits on each quantum node because of the limited quantum memory for storing qubits. This means that more qubits allocated by a quantum node to establish entanglement connections with a predecessor node results in fewer qubits for the entanglement links with the successor.

Entanglement routing is a complex problem that involves not only establishing quantum links but also ensuring the reliable transmission of entanglement across a network. Qubit allocation and path selection has been considered as a pivotal part of entanglement routing in many previous studies such as [8], [9], and [10] since establishing an entanglement link is probabilistic and unstable, relying on the availability of quantum channels between the parties and qubits on both ends, which directly influences the success of entanglement routing. However, these studies operate under the assumption of pre-known probabilities for creating entanglement links [8], [9], [10], [11], [12], [13], [14], [15], [16]. Meanwhile, they tend to overlook the critical aspect of addressing potential network attackers – a well-explored area in conventional network security [17], [18], [19]. Such an attack can be performed on either the data qubit itself or the classical channel that delivers the Bell State Measurement result for quantum teleportation [20]. Note that both categories of attackers can occur in any given time slot and be distinguished from regular entanglement connection failures, by no-cloning theorem or transmission errors in classical channels.

In this paper, we focus on addressing the challenge of online optimal path selection and qubit allocation between two quantum nodes in the presence of a potential QDN attacker. We introduce a novel multi-armed bandits approach grounded, called adversarial group neural bandits. By developing an online adversarial optimization approach using multi-armed bandits, we provide a robust solution that optimizes both path selection and qubit allocation in real-time, without prior knowledge of success rates and under the presence of potential attackers. This enhances the overall efficiency and reliability of quantum entanglement routing, especially in dynamic and adversarial environments. Therefore, our approach contributes to advancing the broader field of entanglement routing by addressing a key challenge that has been largely overlooked in prior studies. Our key contributions encompass the following:

- We present an online optimization problem that concurrently addresses path selection and qubit allocation between two quantum nodes. This scenario accounts for an attacker disrupting data qubit transmission and involves no prior knowledge of the success rate for creating each entanglement link. The goal is to maximize the long-term success rate for establishing long-distance entanglement connections between the chosen quantum nodes.
- To tackle the intricate nonlinear optimization objective, we frame the problem as an adversarial group neural bandits problem and propose a bandits algorithm, called EXPNeuralUCB, which treats possible paths as groups and performs qubit allocation as arm selection in each group. Moreover, by choosing suitable algorithm parameters, we theoretically prove that the algorithm has a regret upper bound of $O(T^{3/4}\sqrt{\log T})$, offering performance guarantees for our algorithm.
- We conduct comprehensive simulations to showcase the superiority of our proposed EXPNeuralUCB over other established advanced bandit algorithms.

II. Related Work

Entanglement Routing. Early research delved into specific network topologies like sphere, ring, diamond star, and chain configurations [21], [22], [23], [24]. Recent studies have shifted focus towards a general QDN topology [8], aiming to maximize expected network throughput while incorporating theoretical analyses for performance guarantees. An extension of this research [11] addresses the challenge of efficiently utilizing network resources to support multiple SD (source-destination) pairs concurrently. Strategies have been explored to augment failure tolerance; some approaches leverage redundant entanglement links in routing to bolster the robustness of QDN [12], [13], [14]. Another avenue of exploration involves a qubit allocation algorithm in QDNs, which combines simulated-annealing and local search techniques [15]. Additionally, an online entanglement routing scenario has been proposed [9], which involves processing requests upon arrival. As an extension, the entanglement routing paradigm in QDNs has evolved from the time slot mode to an asynchronous scheme [10], allowing more proactive utilization of idle quantum resources. Furthermore, opportunistic techniques have been introduced to QDNs [16], enabling the opportunistic establishment of quantum links and augmenting routing flexibility. One work [25] considers the qubit allocation and entanglement routing problem from a user-centric view with a long-term limited qubit budget. However, none of these approaches address the lack of prior knowledge regarding the success rate of each entanglement link or consider adversarial scenarios.

Multi-armed Bandits. Bandit problems are typically categorized as either stochastic bandits or adversarial bandits based on how rewards are generated [26]. The classical UCB algorithm [27] and EXP3 algorithm [28] have been developed for optimal regret bounds in stochastic and adversarial bandits, respectively. However, existing approaches are limited to either the stochastic or adversarial regime, posing challenges for problems with coupled stochastic and adversarial rewards. Recent efforts address the intersection of stochastic and adversarial bandits. In one line of work [29], [30], attempts are made to design a single algorithm applicable to both regimes without prior knowledge. Nevertheless, these works

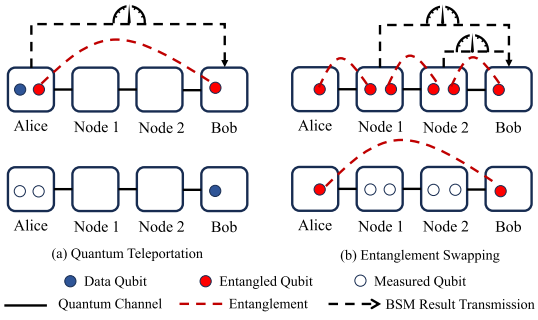


Fig. 2. Quantum teleportation and entanglement swapping.

consider rewards from a single distribution. In another line of research [31], [32], the focus is on stochastic bandits with adversarial corruption of reward observations, though the actual received reward is unaffected. A recent study [33] considers joint rewards influenced by both stochastic distribution and adversarial behavior. However, the stochastic part assumes linearity with the feature vector. The traditional approach to contextual bandits with general nonlinearity is by nonparametric models via Reproducing Kernel Hilbert Space (RKHS) [34], [35], [36], i.e, Kernel bandits [37]. Thanks to the development of NTK theory [38], [39] first utilizes the NTK-based approximation on neural networks and presents a provably efficient MLP-based contextual bandit algorithm, i.e., NeuralUCB. Many follow-up works [40], [41] were inspired by NeuralUCB to model the non-linear reward function under the framework of NeuralUCB. However, these works do not consider the existence of an attacker.

III. Background

1) Quantum Teleportation: Qubits, short for quantum bits, stand as the foundational unit of quantum information. Unlike classical bits restricted to 0 or 1 states, a qubit can simultaneously exist in a coherent superposition of these states. In a two-qubit system, a pair of entangled qubits is known as Bell pairs. One of the significant applications of this entanglement is quantum teleportation, depicted in Figure 2. When Alice and Bob share this pair of entangled qubits (referred to as entangled qubits), Alice can teleport the state of another qubit carrying data information (referred to as data qubits) to Bob. This complex procedure involves Alice conducting a Bell State Measurement between her data qubit and the shared entangled qubit, transmitting the measurement result to Bob through a classical channel. Upon receiving this information, Bob applies unitary operations to his entangled qubit. Consequently, Bob's entangled qubit replicates the state of the original data qubit, while Alice's data qubit collapses, disrupting the entanglement between the two entangled qubits.

2) Entanglement Swapping: In order to perform quantum teleportation between remote quantum nodes, establishing long-distance entanglement is the initial requirement. A crucial technique for achieving this is entanglement swapping, depicted in Figure 2. Here is an illustrative scenario: Alice shares an entangled qubit pair with a third party, Carol, while Bob also holds a qubit pair with Carol. Through a swapping operation on her qubits, Carol can teleport the state of her qubits, initially entangled with Alice, to Bob. As a result, Alice and Bob effectively possess an entangled qubit pair, despite lacking a direct connection. This facilitates the establishment of long-distance entanglement between distant parties. We assume a successful swapping operation due to recent advancements significantly enhancing its success rate to approximately 1, as also presumed in recent state-of-the-art studies [10]. Even when considering the probability of swapping, incorporating a product term akin to the probability of entanglement links will not impact the efficacy of our algorithm.

3) Quantum Data Network: QDN is composed of quantum nodes and quantum channels, forming the nodes and edges of the network. Nodes connected via quantum channels are assumed to also be connected through classical channels. All quantum nodes can perform entanglement swapping and establish links with other nodes, but they are limited by their quantum memory capacity. Moreover, quantum channels are prone to losses, with the success rate of a single entanglement attempt dropping as low as 2.18×10^{-4} [5]. The decoherence time for entanglement is approximately 1.46 seconds, as noted in [42], while each entanglement attempt takes about 165μs [5]. Consequently, only a limited number of thousands of entanglement attempts can be made on a single quantum link within a given time slot.

IV. System Model

A. Network Architecture

We examine a QDN modeled as an undirected graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$, where $\mathcal{V}$ denotes the set of quantum nodes and $\mathcal{E}$ represents the set of edges. Each edge $e = (u, v) \in \mathcal{E}$ indicates the presence of quantum channels between the nodes v and u. Every quantum node $v \in \mathcal{V}$ is equipped with a limited quantity of Q_v qubits, and W_e denotes the number of available quantum channels on edge e. Meanwhile, available qubits and channels may vary over time due to changes in the network circumstance, and we denote them as Q_v^t and W_e^t respectively if needed.

The establishment of a quantum link on an edge $e = (v, u)$ requires one qubit each from nodes v and u along with a quantum channel on e. However, successful entanglement establishment is not assured for every quantum channel. Let $\tilde{p}_e$ represent the unknown success probability of establishing a single entanglement link between nodes v and u during a single attempt. This probability is contingent on both the physical properties of the channel material and the distance between the two nodes. Typically, $\tilde{p}_e$ is low and can vary across different edges. In this study, we assume that the entanglement probability for each edge is unknown.

To increase the probability of successful entanglement, nodes v and u can employ multiple quantum channels and make numerous attempts on each channel within a given time slot. Assuming the outcomes of these attempts are independent, the success probability on a single channel after K attempts is given by $p_e = 1 - (1 - \tilde{p}_e)^K$. Then the overall success probability using q_e qubits at two ends of edge e, u

and v respectively, is given by: $P_e(q_e) = 1 - (1 - p_e)^{q_e}$. It is important to note that we have the constraint $q_e \leq \min(Q_v, Q_u, W_e)$.

B. Problem Formulation

Our primary focus revolves around tackling the challenge of entanglement path selection and qubit allocation. This involves establishing entanglement connections and maximizing the cumulative success probability over a specified duration of time slots T between two chosen quantum nodes. Importantly, this is achieved without prior knowledge of the success rate of entanglement links between any two neighboring nodes.

Consider a set of potential paths between the source node and destination node, denoted as $\mathcal{R}$, and $|\mathcal{R}| = R$. These potential paths do not change over time since they are pre-determined based on the maximum qubit capacity and are independent of the traffic pattern. A path $r \in \mathcal{R}$ is defined as a subset of graph edges $\mathcal{E}$ that form a connected path between them. Given a specific path selection $r \in \mathcal{R}$ and qubit allocation $\mathcal{N}(r) = \{q_e(r), \forall e \in r\}$ for a path r, the entanglement success rate can be calculated as the product of the success probabilities of the individual edges on that path as follows:

$$h_r(\mathcal{N}(r)) = \prod_{e \in r} P_e(q_e(r)) \tag{1}$$

Here, $P_e(q_e(r))$ denotes the success probability of edge e when $q_e(r)$ qubits are allocated at each node on both ends of edge e. The success rate of entanglement for each path is treated as an unknown function that requires learning. While our formulation represents this function as (1), there might be additional factors we have not taken into account. This complexity and formulation challenge motivate us to employ a neural network for learning this function.

Moreover, we consider the potential presence of an adversary in the system, which aims to disrupt the quantum teleportation process and will select paths to attack during each time slot. This adversary may execute attacks either directly on the data qubits or on the classical channels transmitting the measurement result for quantum teleportation. The former attack is detectable due to the no-cloning theorem, while the latter can be identified through transmission errors in the classical channel. Both types of attacks fail the quantum teleportation operation and can be distinguished from failing to establish an entanglement connection in the normal case. Specifically, in each time slot t, we choose a path r^t and perform qubit allocation $\mathcal{N}(r^t)$ to establish entanglement connections; the adversary simultaneously chooses a binary attack vector $a^t = (a^t(r), \forall r \in \mathcal{R})$, where $a^t(r) = 0$ if the adversary performs an attack on the path r and $a^t(r) = 1$ otherwise. At the end of time slot t, we can only observe the success or failure of establishing at least one entanglement connection, which can be denoted as a random variable Y^t. Note that this entanglement connection is typically verified by performing a Bell state measurement [20] on the entangled qubits, which indirectly confirms the entangled state without directly measuring the qubits themselves. Y^t conforms to the Bernoulli distribution, which takes the value 1 with probability s^t and 0 with probability $1 - s^t$, where s^t is the unknown success rate depending on the selected path and the allocated qubits strategy and can be formulated as:

$$s^t(r^t, \mathcal{N}(r^t)) = a^t(r^t) \prod_{e \in r^t} P_e(q_e(r^t)). \tag{2}$$

Note that when an attacked path is chosen, we can still differentiate the attack from a normal connection failure, as previously discussed, even though the latent success rate is 0, just like in the failure scenario. Clearly, we have $\mathbb{E}[Y^t] = s^t(r^t, \mathcal{N}(r^t))$.

Objective: The quantum user's objective is to maximize the success rate of entanglement connections between the source node and destination node over T time slots ($t = 0, 1, \ldots, T-1$) by selecting an optimal path and allocating qubits along the path in each time slot:

$$\max \sum_{t=1}^{T} s^t(r^t, \mathcal{N}(r^t)). \tag{3}$$

V. Algorithm Design

In this section, we first frame the problem of quantum entanglement path selection and qubit allocation in the presence of attackers as an adversarial group neural bandits problem. Subsequently, we introduce an algorithm named EXPNeuralUCB designed to address this specific problem. In Section V-D, we provide proof demonstrating that EXPNeuralUCB can attain a sublinear regret bound.

A. Adversarial Group Neural Bandits

We frame the selection of entanglement paths and qubit allocation as a sequential decision-making problem over T rounds. We define R groups of arms, denoted as $\mathcal{R} = \{1, 2, \ldots, R\}$, each corresponding to a set of potential paths. Within each group $r \in \mathcal{R}$, the arms represent various qubit allocation strategies, which differ across groups. Each group r has arms of dimension D_r, corresponding to the number of links along path r. We denote these dimensions collectively by $\mathcal{D} = \{D_1, D_2, \ldots, D_R\}$.

In each round t, for each group r, there are $|\mathcal{X}_r^t|$ available arms, where $\mathcal{X}_r^t \subseteq \mathbb{R}^{D_r}$ consists of D_r-dimensional vectors. Each vector $x \in \mathcal{X}_r^t$ represents a feasible qubit allocation strategy along path r, adhering to node qubit capacity constraints Q_v^t for all nodes v, and link qubit channel capacity constraints W_e^t for all links e. Each group r is associated with a function h_r, defined on the domain $\mathcal{X}_r \subseteq \mathbb{R}^{D_r}$, representing the entanglement success rate on path r in a stochastic environment, as described by equation (1). We obtain the available arm set by exhaustively exploring all possible combinations of qubit allocations along the nodes of the path, which guarantees that all feasible qubit allocation strategies within the capacity constraints are thoroughly evaluated. This function does not account for potential adversarial actions and remains unknown to the learner. The reason why we do not use $h_r(x)$ to represent equation (2) directly is that NeuralUCB is under the stochastic assumption [39]. The reward for choosing an arm x in group r is given by $h_r(x)$, where h_r is constrained such that $0 \leq h_r(x) \leq 1$ for any x in any group r.

Optimal Benchmark. Given that the adversary can adopt any attack strategy against the groups, we focus on the optimal benchmark among the group-static strategies in hindsight. A group-static strategy involves selecting a fixed group throughout the entire duration of T rounds, while allowing the chosen arm to vary. For a given group r, the optimal arm from the set $\mathcal{X}r^t$ that maximizes the expected reward can be computed as $\xi_r^t \triangleq \arg\max x \in \mathcal{X}_r^t h_r(x)$, independent of the adversary's attacks. In the special case where the set of available arms $\mathcal{X}_r^t$ remains constant across all rounds, the optimal arm ξ_r^t for group r is also fixed, allowing the time index to be omitted.

With the optimal arms in each group in each round understood, the optimal group given an attacking sequence $a^1, \ldots, a^T$ is thus the one that maximizes the total reward, which corresponds to maximizing times of successful entanglement connection, or equivalently, maximizing the accumulative success rate of entanglement connections defined in (3),

$$\gamma(a^1, \ldots, a^T) = \arg\max_{r \in \mathcal{R}} \sum_{t=1}^{T} s^t(r, \xi_r^t, a^t). \tag{4}$$

For notation simplicity, we write $\gamma(a^1, \ldots, a^T) = \gamma$ by dropping the attack sequences but the readers should be cautious that the optimal group γ depends on the attacks (and T).

Regret. To optimize the objective function specified in (3), we defined the regret of the learner as the difference between the total reward achieved by the optimal benchmark and the total reward attained by the learner's algorithm.

$$\text{REGRET}(T) = \sum_{t=1}^{T} s^t(\gamma, \xi_\gamma^t) - \mathbb{E}[\sum_{t=1}^{T} s^t(r^t, x^t)], \tag{5}$$

where this expectation is over the possible internal randomization of the algorithm. The aim is to design a bandits algorithm that exhibits sublinear regret, indicating that the round-average regret diminishes to 0 as T approaches infinity.

The bandit problem under consideration exhibits a semi-stochastic and semi-adversarial nature. On one hand, the task of learning the optimal arm within a group represents a stochastic non-linear bandit problem. On the other hand, learning the optimal group is framed as an adversarial bandit problem. Consequently, the problem at hand involves uncertainties arising from both the stochastic characteristics of the environment and the adversarial behavior of the opponent simultaneously. The learner's received reward is, therefore, a combined outcome influenced by both factors.

B. Function Approximation via Neural Network

We employ the NeuralUCB framework [39] to model the non-linear reward function. NeuralUCB utilizes neural networks for estimation in the following manner:

Neural Network Architecture: Let f_r be overparameterized multi-layer perceptions (MLPs)[1] with depth $L \geq 2$ and width m for each hidden layer to represent the unknown function h_r:

$$f_r(x; \theta_r) = \sqrt{m} Z_r^L \sigma(Z_r^{L-1} \sigma(\ldots \sigma(Z_r^1 x))), \tag{6}$$

where θ_r are stacked by $Z_r^l, \forall l \in [L], \forall r \in \mathcal{R}$; Given $x \in \mathbb{R}^{D_r}$, $Z_r^1 \in \mathbb{R}^{m \times D_r}$,$Z_r^l \in \mathbb{R}^{m \times m}$, $2 \leq l \leq L-1$,$Z_r^L \in \mathbb{R}^{m \times 1}$, $\sigma(x) = \max\{x, 0\}$ is the rectified linear unit (ReLU) activation function. We denote the gradient of the neural network function by $g_r(x; \theta_r) = \nabla_{\theta_r} f(x; \theta_r)$.

For every $r \in \mathcal{R}$, the parameters θ_r follow similar initialization and update procedures as follows.

Neural Network Initialization: We initialize θ_r with $\theta_r^0 = [\text{vec}(Z_r^1)^\top, \ldots, \text{vec}(Z_r^L)^\top] \in \mathbb{R}^k$ with $k = m + mD_r + m^2(L-1)$, where for each $1 \leq l \leq L-1$, $Z_r^l = (Z, 0; 0, Z)$, each entry of Z is generated independently from $\mathcal{I}(0, 4/m)$; $Z_r^L = (z^\top, -z^\top)$, each entry of z is generated independently from Gaussian $\mathcal{I}(0, 2/m)$.

Neural Network Update: In round t, θ_r is updated to θ_r^t, i.e., the optimal solution to the loss function for neural network training. The loss function is defined as

$$\mathcal{L}(\theta) = \sum_{b=1}^{t} (f_r(\{x_r^b\}; \theta) - h_r(x_r^b))^2/2 + m\lambda \|\theta - \theta_r^0\|_2^2/2, \tag{7}$$

where λ is the regularization parameter. Set $\theta^{(0)} = \theta_r^0$ and we adopt gradient-based methods to optimize the loss function for J steps with step size ζ:

$$\theta^{(j+1)} = \theta^{(j)} - \zeta \nabla \mathcal{L}(\theta^{(j)}), \tag{8}$$

where $j \in \{0, \ldots, J-1\}$, then the estimation for θ_r at round t is set $\theta_r^t = \theta^{(J)}$.

C. EXPNeuralUCB

EXPNeuralUCB combines strengths from both EXP3 and NeuralUCB. On one hand, it preserves an unbiased cumulative historical reward estimate, denoted as S_r^t, for each group, enhancing the process of group selection. On the other hand, EXPNeuralUCB maintains a parameter estimate θ_r^t for each group. These estimates are incrementally updated across rounds using equations (7) and (8), facilitating arm selection within a group. The algorithm's pseudo-code is outlined in Algorithm 1, and we explain the algorithm's procedure below.

Group Selection: In each round t, EXPNeuralUCB calculates an unbiased estimate of the cumulative historical reward S_r^{t-1} up to round $t-1$ for each group r. This estimation is based on past group and arm selections, as well as reward realizations, following (9). Here, $\mathbf{1}\{\cdot\}$ denotes the indicator function, and P^b represents the group sampling distribution calculated and utilized in round b. it is worth noting that we slightly abuse notation by using $P_e(q_e(r))$ to express the success probability of edge e when $q_e(r)$ qubits are allocated to e in the past. When computing S_r^{t-1}, the actual reward (i.e., whether the entanglement was successfully established) received in round b, denoted as Y^b, is added to the cumulative reward of group r only if the selected group in round b is r. This addition is followed by division by the selection probability. Using the updated S_r^{t-1}, a new group sampling distribution

[1]For the simplicity of notation, we assume the width of each layer is m and that MLPs for each group share the same set of hyperparameters, e.g., m, L, λ, ζ, J in Algorithm 9. In practice, their hyperparameters may vary from each other.

Algorithm 1 EXPNeuralUCB

Input: Time horizon T, regularization parameter λ, $\theta_r^0 \sim \text{init}(\cdot)$, NeuralUCB exploration parameter ν, confidence parameter δ, step size ζ, number of gradient descent steps J, network width m, network depth L, learning rate $\eta > 0$, EXP3 exploration rate $\beta \in (0, 1)$
Initialization: $V_r^0 = \lambda I_{D_r}, \forall r \in \mathcal{R}$.
for $t = 1, \ldots, T$ **do**
  Compute estimated cumulative reward for each r

$$S_r^{t-1} = \sum_{b=1}^{t-1} \frac{\mathbf{1}\{r^b = r\}}{P^b(r)} Y^b \tag{9}$$

  Compute the sampling distribution for each r

$$P^t(r) = (1-\beta)\frac{\exp(\eta S_r^{t-1})}{\sum_{r'=1}^{R}\exp(\eta S_{r'}^{t-1})} + \frac{\beta}{R} \tag{10}$$

  Sample group $r^t \sim P^t$
  $\forall x \in \mathcal{X}_{r^t}^t$, compute $U_{r^t}^t(x)$ within r^t:

$$U_{r^t}^t(x) = f(x; \theta_{r^t}^{t-1}) + \alpha^t \|g_{r^t}(x; \theta_{r^t}^{t-1})/\sqrt{m}\|_{(V_{r^t}^{t-1})^{-1}} \tag{11}$$

  Select the best estimated arm within r^t: $x^t = \arg\max_{x \in \mathcal{X}_{r^t}^t} U_{r^t}^t(x)$
  Play group/arm (r^t, x^t)
  Observe reward r^t
  if $a^t(r^t) = 0$ **then**
    Update $V_{r^t}^t = V_{r^t}^{t-1} + g_r(x^t; \theta_{r^t}^{t-1}) g_r(x^t; \theta_{r^t}^{t-1})^\top / m$
    Update $\theta_{r^t}^t$ by training MLPs f_{r^t} using collected feedbacks as in (7) and (8).
  else
    $V_{r^t}^t = V_{r^t}^{t-1}, \hat{\theta}_{r^t}^t = \theta_{r^t}^{t-1}$
  end if
  For all $r \neq r^t$, $V_r^t = V_r^{t-1}$, $\theta_r^t = \theta_r^{t-1}$
end for

can be computed through (10). This distribution is a weighted sum of two distributions with weights $1-\beta$ and β. The first distribution selects group r proportionally to $\exp(\eta S_r^{t-1})$, favoring groups with higher cumulative reward estimates and emphasizing exploitation. The second distribution is simply a uniform distribution, promoting the exploration of all groups with equal probability. The weights $1-\beta$ and β adjust the trade-off between exploitation and exploration at the group level. With the group sampling distribution P^t, a group r^t is then sampled and subsequently chosen in the current round.

Arm Selection: Subsequently, EXPNeuralUCB determines the best-estimated arm within the selected group r^t based on (11). This computation utilizes the estimated group parameter $\theta_{r^t}^{t-1}$ and the auxiliary variable $V_{r^t}^{t-1}$. In (11), the first term represents the reward estimate of an arm x in the group r^t, while the second term signifies the confidence associated with this estimate. The parameter α^t plays a crucial role in adjusting the balance between the exploitation and exploration of arms within each group as further illustrated in (14).

Variable Update: Then EXPNeuralUCB updates the various variables depending on the present attacker strategy. Specifically, if the received reward $a^t(r^t) = 0$, which means the adversary attacks the selected group in round t, then all parameters are unchanged. Otherwise, for the selected group r^t, the auxiliary variable $V_{r^t}^t$ is first updated. Then update the group parameter $\theta_{r^t}^t$ according to (7) and (8) to (approximately) minimize $\mathcal{L}(\theta)$ using gradient descent. For the unselected groups, their auxiliary variables and the parameter estimates also remain unchanged.

D. Regret Analysis

In EXPNeuralUCB, combating the stochastic non-linearity uncertainty within a group is intertwined with combating the adversarial uncertainty across groups. Thus, the regret analysis of EXPNeuralUCB must consider the regrets due to these two aspects simultaneously. In this subsection, we show that through a careful selection of the algorithm parameters, EXPNeuralUCB achieves a sublinear regret bound.

We start with several lemmas on estimating the parameters of the groups.

Lemma 1 (Lemma 5.1 in [39]): For a sufficiently large network width m, $\forall r \in \mathcal{R}, \forall x \in \mathcal{X}_r^t$, there exists a θ_r^* at round t such that with probability at least $1-\delta$, we have

$$h_r(x) = \langle g_r(x, \theta_r^0), \theta_r^* - \theta_r^0 \rangle$$
$$\sqrt{m}\|\theta_r^* - \theta_r^0\|_2 \leq \sqrt{2h_r^\top H_r^{-1} h_r},$$

where H_r is the neural tangent kernel (NTK) matrix for h_r defined in [39].

Lemma 2: For a sufficiently large network width m, if

$$m \geq \text{poly}(T, L, \sup(\mathcal{D}), \lambda^{-1}, \log(1/\delta));$$
$$\lambda \geq \max\{1, (2h^\top H^{-1} h)^{-1}\}; \zeta = O\left((mTL + m\lambda)^{-1}\right), \tag{12}$$

the estimated θ_r^t satisfies the following error bound for all group r and round t with probability at least $1-\delta$,

$$|\theta_r^t - \theta_r^0\|_2 \leq 2\sqrt{t/(m\lambda)}, \|\theta_r^t - \theta_r^*\|_{V_r^t} \leq \alpha^t\sqrt{m},$$
$$\alpha^t = O(\sqrt{\tilde{d}\log(1+t)}),$$

where $\tilde{d}$ is the effective dimension of the NTK matrix.

Lemma 3: For a sufficiently large network width m, with probability of $1-\delta$, the single step regret bound for the group r at t round is bounded by $h_r(\xi_r) - h_r(x^t) \leq 2\alpha^t \|g_r(x; \theta_r^{t-1})/\sqrt{m}\|_{(V_r^{t-1})^{-1}} + 3O(m^{-1/6})$.

Lemma 4: For a sufficiently large network width m, then we have for all r,

$$\sum_{t=1}^{T} \mathbf{1}\{r^t = r\} a^t(r) \alpha^t \|g_r(x_r^t; \theta_r^{t-1}/\sqrt{m})\|_{(V_r^{t-1})^{-1}} \leq O(\sqrt{\tilde{d}\log(1+T)}) O(\sqrt{T\tilde{d}\log(1+T)}), \tag{13}$$

with probability at least $1-\delta$.

Theorem 1: For any $\delta \in (0,1)$, by choosing $\beta = T^{-1/4}\sqrt{\log(T)}$ and $\eta = T^{-1/2}$, such that for any $\delta \in (0,1)$, if m, λ, ζ satisfy the same condition as Lemma 2, then with probability at least $1-\delta$, EXPNeuralUCB yields the following expected regret bound $\text{REGRET}(T) = O(T^{3/4}\sqrt{\log(T)})$.

Remark: In the scenario where the optimal arm/allocation strategy in each group/path is known to the learner (pure-adversarial setting), the EXP3 algorithm yields a regret bound

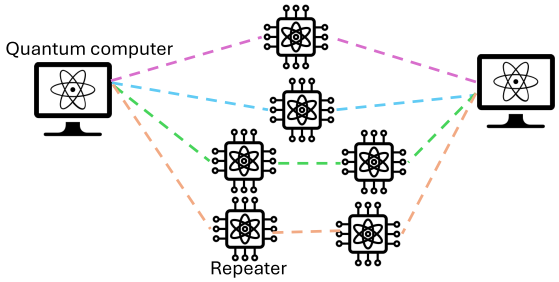


Fig. 3. The topology of the QDN used in the simulation.

of $O(\sqrt{T})$. Conversely, in the pure-stochastic setting, where paths/groups are not attacked by the adversary in all rounds, NeuralUCB, specifically for the group-disjoint parameter case, achieves a regret bound of $\sqrt{T \log T}$. The regret incurred by EXPNeuralUCB is higher than both, attributable to simultaneously addressing adversarial uncertainty and stochastic uncertainty. However, it is noteworthy that neither EXP3 nor NeuralUCB can achieve a sublinear regret in our considered problem. Although EXPUCB [33] effectively addresses the challenge posed by the intersection of stochastic distribution and adversarial behavior, it faces limitations in learning the stochastic component with nonlinear features.

VI. Simulation Results

In this section, we evaluate our proposed algorithm EXPNeuralUCB and compare its performance against several baselines.

A. Simulation Setup

1) Network Setting: We model a QDN with four potential paths ($R = 4$) connecting a source node to a destination node, as illustrated in Fig. 3. The success probabilities for entanglement establishment $\tilde{p}_e$ vary across these paths and are detailed in Table I. To increase the probability of successful entanglement, each link undergoes 4000 entanglement connection attempts ($K = 4000$) in our simulations.

2) Qubit Capacity: We simulate a dynamic network environment within the QDN where the qubit availability at quantum repeaters fluctuates over time. This variation is influenced by the usage patterns of other network users. Specifically, we model two network states, "busy" and "idle". In the "idle" state, the qubit capacity at the quantum repeaters is greater than in the "busy" state. Additionally, the capacity of these repeaters varies across different paths, as detailed in Table I. It is assumed that both the source and destination quantum nodes possess a sufficient number of qubits, where "sufficient" means that their available qubits are equal to the maximum qubit capacity of any quantum repeater in the network. This ensures that the source and destination nodes can handle both "busy" and "idle" network states without being a bottleneck.

B. Adversary's Strategy

We simulate two types of attacking strategies: an oblivious strategy and an adaptive strategy. In the oblivious strategy, the adversary employs a randomized Markov attacking strategy. The transition matrix for this strategy is:

$$\begin{bmatrix} 0.35 & 0.15 & 0.35 & 0.15 \\ 0.3 & 0.2 & 0.3 & 0.2 \\ 0.35 & 0.15 & 0.35 & 0.15 \\ 0.3 & 0.2 & 0.3 & 0.2 \end{bmatrix},$$

where the entry in row i and column j denotes the probability of attacking path j at time slot t given that path i was the target at time slot $t-1$. In the adaptive strategy, the adversary observes which path the learner chose in the previous time slot $(t-1)$ and then targets the same path in the current slot (t). In both strategies, the adversary attacks exactly one path in each time slot.

TABLE I
Parameters of the Simulated QDN

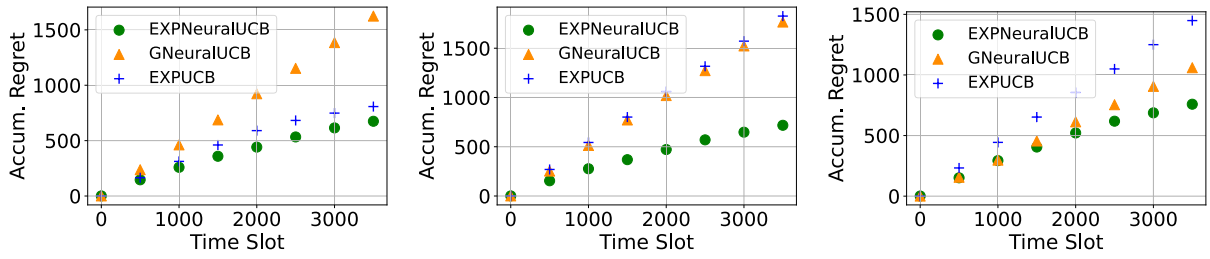
Path	# of Repeaters	Success Rate	Capacity (Busy/Idle)
1	1	$1.5e-4$	8/9
2	1	$1e-4$	10/11
3	2	$2e-4$	5/11, 5/11
4	2	$1.5e-4$	6/12, 6/12

1) Baseline Schemes: We consider the following baselines in addition to the **Oracle** strategy in hindsight defined in equation (4). **GNeuralUCB**: This variant adopts the classical NeuralUCB algorithm for the group setting, disregarding attacks. In each time slot, it selects the group and arm with the highest NeuralUCB of the estimated reward, according to equation (11). The auxiliary variables remain unaltered if the received reward is 0. **EXPUCB**: This algorithm is similar to our proposed approach. It employs the EXP3 algorithm to choose the group based on the historical accumulated reward for each group. However, it leverages the LinUCB algorithm, as proposed by [33], for arm selection.

For EXPNeuralUCB, we use the Adam [43] optimizer for neural network training and set the default algorithm parameters as $\lambda = 1, \delta = 0.1, m = 128, L = 2, J = 8$, $\zeta = 1 \times 10^{-4}, \beta = T^{-1/4}\sqrt{\log T}, \nu = 1$ and $\eta = T^{-1/2}$.

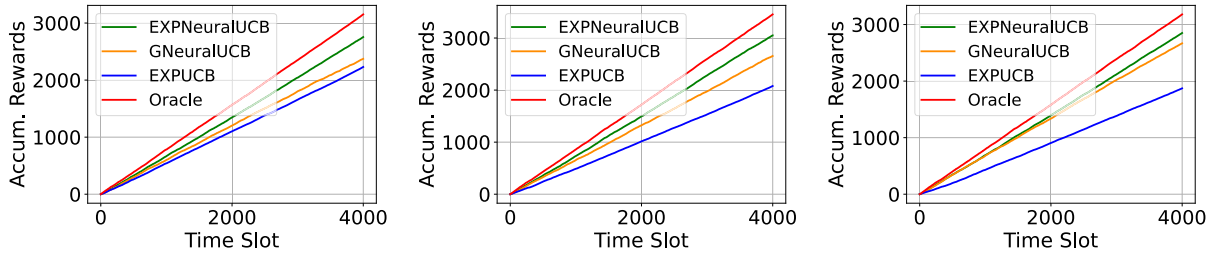
C. Performance Comparison

We start by comparing the performance of EXPNeuralUCB with baseline algorithms in terms of total regret (as depicted in Fig. 4) and total reward (illustrated in Fig. 5), specifically under scenarios involving an oblivious attacker. The simulations cover three distinct network state distributions: ALL-BUSY, where every time slot is "busy"; ALL-IDLE, where every slot is "idle"; and HALF-HALF, where slots are "busy" or "idle" with equal probability of 0.5. It is noteworthy that the Oracle strategy may differ based on the number of slots, hence only selected data points are shown in Fig. 4. The Oracle results presented in Fig. 5 apply specifically to simulations with $T = 4000$ slots. In all scenarios, EXPNeuralUCB outperforms non-Oracle baselines and exhibits sublinear regret throughout the simulation. The relative underperformance of non-Oracle baselines compared to EXPNeuralUCB can be attributed to specific limitations. **EXPUCB** adjusts to the attacking strategy and selects paths less vulnerable to attacks,



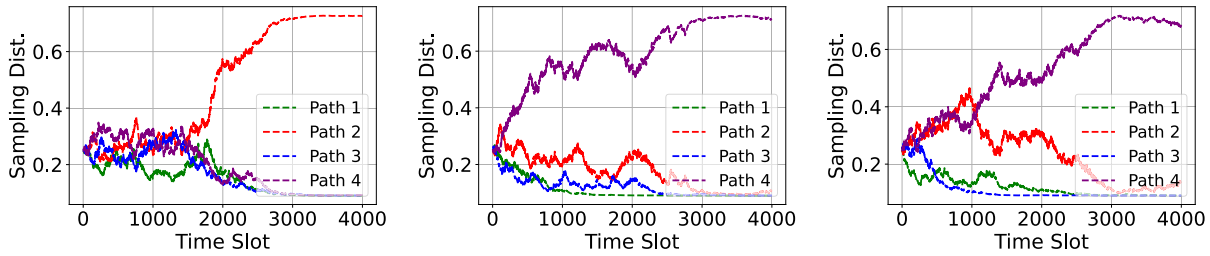
(a) Only busy period scenes (b) Only free period scenes (c) 50% busy period 50% free period

Fig. 4. Regret achieved by EXPNeuralUCB and baselines.



(a) Only busy period scenes (b) Only free period scenes (c) 50% busy period 50% free period

Fig. 5. Total reward achieved by EXPNeuralUCB and baselines.



(a) Only busy period scenes (b) Only free period scenes (c) 50% busy period 50% free period

Fig. 6. Evolution of the sampling distribution for each path.

yet it does not effectively learn the success probabilities of entanglement establishment across different quantum channels, thus failing to optimize qubit allocation. **GNeuralUCB**, on the other hand, overlooks the uncertainties introduced by path-level attacks, potentially choosing paths that suffer frequent attacks, resulting in lower overall rewards compared to EXPNeuralUCB.

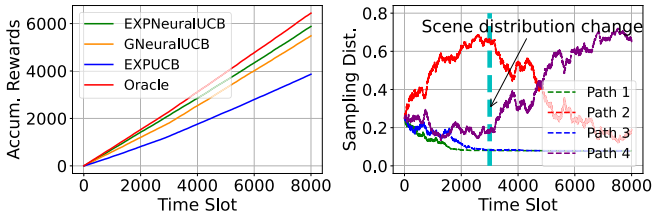
D. Behaviors of EXPNeuralUCB

Let us explore in greater detail the performance of EXPNeuralUCB through an analysis of its path sampling distribution, depicted in Fig. 6. Our simulation is configured to favor paths 1 and 2 in the ALL-BUSY scenario, with a preference for path 1, and paths 3 and 4 in the ALL-IDLE scenario, where path 3 is the optimal choice under no attack conditions. Fig. 6(a) demonstrates that EXPNeuralUCB effectively identifies path 2 as the preferred choice in the ALL-BUSY scenario. Notably, it frequently avoids selecting path 1, even though it is the best option when there are no attacks, because it recognizes path 1's higher susceptibility to attacks. In contrast, Fig. 6(b) shows that in the ALL-IDLE scenario, EXPNeuralUCB primarily selects path 4, acknowledging that path 3—although optimal in an attack-free environment—is more prone to attacks. Furthermore, in the HALF-HALF scenario, where paths 3 and 4 offer higher overall rewards compared to paths 1 and 2 in the absence of attacks, EXPNeuralUCB, as illustrated in Figure Fig. 6(c), accurately identifies path 4 as the best choice when under attack, consistent with the selections made by the Oracle strategy.

We also report the execution time and memory usage of EXPNeuralUCB and the compared algorithms, as shown in Table II. The execution time and memory usage are averaged over 4000 rounds. As depicted in Table II, EXPUCB has the smallest execution time and memory usage among the three algorithms, albeit with the worst performance. While EXPNeuralUCB and GNeuralUCB exhibit similar execution time and memory usage, EXPNeuralUCB outperforms GNeuralUCB in terms of overall performance. Hence, EXPUCB is suitable when minimizing execution time and memory is critical, whereas EXPNeuralUCB is a better choice for higher performance needs.

TABLE II
EXECUTION TIME AND MEMORY USAGE

	EXPNeuralUCB	GNeuralUCB	EXPUCB
Execution Time (s)	0.06275	0.06282	0.00161
Memory Usage (MB)	334.03	333.30	314.30



(a) Accumumative rewards (b) Sampling distribution.

Fig. 7. Performance of EXPNeuralUCB under changing period scenes.

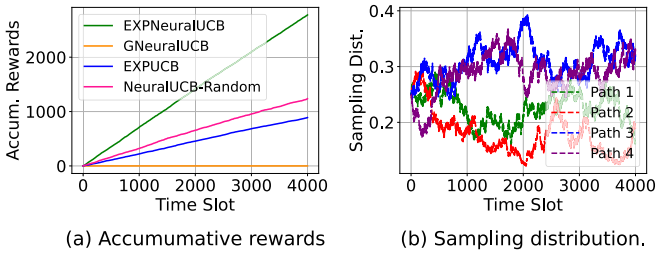
E. Impact of Time-Varying State Distributions

In this set of experiments, we assess the performance of EXPNeuralUCB under conditions where the state distributions change over time. Initially, for the first 3000 slots, the probability of a slot being "busy" is 0.8 and "idle" is 0.2. From slot 3000 onwards, the probabilities invert, with "busy" at 0.2 and "idle" at 0.8, marking a significant shift in the state distribution at time slot 3000. Fig. 7 illustrates the performance of EXPNeuralUCB in this dynamic environment. As depicted in Fig. 7(a), EXPNeuralUCB consistently achieves the highest total reward when compared to non-Oracle baseline strategies. This performance is attributed to its capacity to adapt and make strategic decisions in response to environmental changes. The adaptability of EXPNeuralUCB is further highlighted in Fig. 7(b), where there is a noticeable shift in the path sampling distribution following the change in state distribution. Initially, EXPNeuralUCB predominantly favors path 2; however, following the transition at slot 3000, it shifts to path 4.

F. Impact of Attacker Strategies

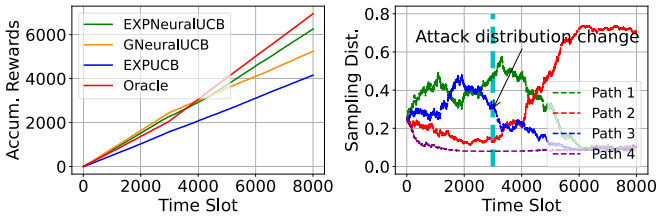
Adaptive Attacking Strategy. In this subsection, we examine the performance of EXPNeuralUCB when facing an adaptive attacking strategy, contrasting with the previously discussed oblivious attacking strategy. Notably, GNeuralUCB tends to perform poorly in this adaptive scenario due to its inability to adjust to ongoing attacks, often becoming stuck on paths that are consistently targeted by the adaptive strategy. To mitigate this, we have enhanced GNeuralUCB to create a new variant, NeuralUCB-Random. This modified version employs a strategy where the learner randomly selects a path each slot and then utilizes NeuralUCB to decide the allocation strategy.

Fig. 8(a) illustrates the total rewards achieved by EXPNeuralUCB and the baseline strategies in the ALL-IDLE scenario, where the discrepancy in performance is more pronounced than in scenarios with an oblivious attacker. This underscores EXPNeuralUCB's superior capability to adapt its path selection and allocation strategies in response to more complex and challenging conditions. Furthermore, Fig. 8(b) sheds light on the path sampling probabilities over time. In the



(a) Accumumative rewards (b) Sampling distribution.

Fig. 8. Performance of EXPNeuralUCB under the adaptive attacking strategy. (Only free period scenes).



(a) Accumumative rewards (b) Sampling distribution.

Fig. 9. Performance of EXPNeuralUCB under the dynamic Markov attacking strategy (Only free period scenes).

ALL-IDLE scenario, paths 3 and 4 are typically favored, and this preference is reflected in our simulation outcomes, where these paths are chosen more frequently. Remarkably, EXPNeuralUCB employs a strategic randomization between these paths, deviating from selecting the optimal path in an attack-free environment (path 3). This approach helps to circumvent persistent attacks linked with the adaptive attacking strategy. Although NeuralUCB-Random also incorporates path randomization, it does not match EXPNeuralUCB in identifying and selecting the most advantageous paths during attack-free periods, resulting in lower overall rewards.

Time-varying Attacking Strategy. We conducted simulations to investigate how EXPNeuralUCB responds to dynamic strategies used by an oblivious attacker who alters the attacking transition matrix at time slot 3000. The matrix changes as follows:

$$\begin{bmatrix} 0.2 & 0.3 & 0.2 & 0.3 \\ 0.15 & 0.35 & 0.15 & 0.35 \\ 0.2 & 0.3 & 0.2 & 0.3 \\ 0.15 & 0.35 & 0.15 & 0.35 \end{bmatrix} \Rightarrow \begin{bmatrix} 0.35 & 0.15 & 0.45 & 0.05 \\ 0.3 & 0.2 & 0.4 & 0.1 \\ 0.35 & 0.15 & 0.45 & 0.05 \\ 0.3 & 0.2 & 0.4 & 0.1 \end{bmatrix}.$$

Fig. 9(a) shows the total rewards achieved by different algorithms across a span of 8000 time slots, with the Oracle providing a reference for optimal performance. In the initial 3000 slots, GNeuralUCB outperforms EXPNeuralUCB due to its selection of what is initially the optimal path (path 3), which faces the least attacks. During this period, GNeuralUCB frequently selects this path more often than EXPNeuralUCB. However, after the attacking strategy shifts post-slot 3000 and path 3 ceases to be the most favorable, GNeuralUCB struggles to adapt. In contrast, EXPNeuralUCB demonstrates superior performance in adjusting to the new attack pattern, eventually surpassing GNeuralUCB in total reward. Fig. 9(b) further illustrates the shift in path selection probabilities post-slot

3000, showcasing EXPNeuralUCB's ability to effectively adapt to changes in the attacker's strategy.

VII. Conclusion

In this paper, we studied the problem of online path selection and qubit allocation in QDNs, specifically under the presence of potential path attacks. Our goal is to optimize the long-term success rate of entanglement connections between two quantum nodes. We approach this challenge by formulating it as an adversarial group neural bandits problem, introducing the EXPNeuralUCB algorithm, which treats potential paths as groups and qubit allocation as arm selections. Additionally, we demonstrate that EXPNeuralUCB achieves a theoretical regret upper bound of $O(T^{3/4}\sqrt{\log T})$. To assess the effectiveness of our algorithm, we conducted a series of experiments in various simulation environments, confirming that EXPNeuralUCB outperforms other baseline algorithms.

Appendix

A. Proof of Lemma 2

Proof: This lemma follows Lemma 5.2 in [39] by considering the sub-sequence of rounds in which the learner selects group r and is not attacked by the adversary. Then we have: $\|\theta_r^t - \theta_r^*\|_{V_r^t} \leq \alpha_r^t\sqrt{m}$ and α_r^t is defined as

$$\begin{aligned}
&\alpha_r^t \\
&= \sqrt{1 + C_1 m^{-1/6}\sqrt{\log m}L^4 t^{7/6}\lambda^{-7/6}} \\
&\times \left(\nu\sqrt{\log\frac{\det V_r^t}{\det \lambda I} + C_2 m^{-1/6}\sqrt{\log m}L^4 t^{5/3}\lambda^{-1/6} - 2\log\delta}\right. \\
&\left. + \sqrt{\lambda}\right) + (\lambda + C_3 tL)\Big[(1-\zeta m\lambda)^{J/2}\sqrt{t/\lambda} \\
&+ m^{-1/6}\sqrt{\log m}L^{7/2}t^{5/3}\lambda^{-5/3}(1+\sqrt{t/\lambda})\Big]. \qquad (14)
\end{aligned}$$

for some constant C_1, C_2, C_3. λ is the regularization parameter, ν denotes the exploration parameter for UCB-like estimation; J is the number of gradient steps for each round of neural network training; when the network width m, regularization parameter λ, and step size ζ satisfy the same condition as Theorem 1, $\alpha_r^t = O(\sqrt{\tilde{d}\log(1+T)})$ and the group index can be dropped. □

Note Lemma 2 leads to an upper confidence bound (UCB) on the prediction error of the estimated parameter θ_r^t for each group, and thus the UCB-based arm selection rule in (11).

B. Proof of Lemma 3

Proof: This Lemma follows the Lemma 5.3 in [39] by combining the Lemma 2. □

C. Proof of Lemma 4

Proof: Consider the sub-sequence of rounds where group r is selected by the learner and not attacked by the adversary, we have

$$\begin{aligned}
&\sum_{i=b}^{B}\sqrt{g_r(x^i;\theta_r^{i-1})^\top (V_r^{i-1})^{-1} g_r(x^i;\theta_r^{i-1})/m} \\
&\leq \sqrt{B\sum_{b=1}^{B} g_r(x^b;\theta_r^{b-1})^\top (V_r^{b-1})^{-1} g_r(x^b;\theta_r^{b-1})/m} \\
&\leq \sqrt{B\tilde{d}\log(1+B)} \leq \sqrt{T\tilde{d}\log(1+T)}
\end{aligned}$$

where the first inequality is due to Jensen's inequality, the second inequality is due to Lemma 5.4 in [39], and the last inequality is because the length of the sub-sequence B is smaller than T. Further noticing that $\alpha^t = O(\sqrt{\tilde{d}\log(1+T)})$ yields the desired bound. □

Now, we are ready to present the regret bound. *Proof:* Denote $w^t(r) = \exp(\eta S_r^{t-1})$, $W^t = \sum_{r'=1}^{R} w^t(r')$ and $I^t(r) = \frac{\mathbf{1}\{r^t = r\}}{P^t(r)}$. Then there exists a positive constant $\bar{C}$ such that for any $\delta \in (0,1)$, if η and m satisfy the same conditions as in Lemma 2, W^{T+1} in the expectation can be lowered bounded with probability at least $1-\delta$ as follows

$$\begin{aligned}
&\mathbb{E}\left[\log\left(\frac{W^{T+1}}{W^1}\right)\right] \\
&\geq \mathbb{E}\left[\log\left(\frac{\exp(\eta\sum_{t=1}^{T} I^t(\gamma)Y^t)}{W^1}\right)\right] \\
&= \mathbb{E}\left[\eta\sum_{t=1}^{T} I^t(\gamma)Y^t\right] - \log R = \eta\sum_{t=1}^{T} a^t(\gamma)h_\gamma(x^t) - \log R \\
&\geq \eta\sum_{t=1}^{T} a^t(\gamma)h_\gamma(\xi_\gamma) - \log R - \eta\sum_{t=1}^{T} a^t(\gamma) \\
&\quad\times\left(2\alpha^t\|g_\gamma(x^t;\theta_\gamma^{t-1}/\sqrt{m})\|_{(V_\gamma^{t-1})^{-1}} + 3O(m^{-1/6})\right), \qquad (15)
\end{aligned}$$

where the first inequality uses $W^{T+1} \geq w^{T+1}(\gamma)$, the second equality uses $W^1 = R$, the third equality uses the definition of Y^t and $\mathbb{E}[I^t(\gamma)] = 1$, and the last inequality is derived based on the Lemma 3.

On the other hand, we have the following upper-bound

$$\begin{aligned}
&\mathbb{E}\left[\log\left(\frac{W^{t+1}}{W^t}\right)\right] \\
&= \mathbb{E}\left[\log\left(\sum_{r=1}^{R}\frac{\exp(\eta\sum_{b=1}^{t} I^b(r)Y^b)}{W^t}\right)\right] \\
&= \mathbb{E}\left[\log\left(\sum_{r=1}^{R}\frac{\exp(\eta\sum_{b=1}^{t-1} I^b(r)Y^b)}{W^t}\exp(\eta I^t(r)s^t)\right)\right] \\
&= \log\left(\sum_{r=1}^{R}\frac{P^t(r)-\frac{\beta}{R}}{1-\beta}\exp(\eta I^t(r)s^t)\right) \\
&\leq \log\left(\sum_{r=1}^{R}\frac{P^t(r)-\frac{\beta}{R}}{1-\beta}(1+\eta I^t(r)s^t + (\eta I^t(r)s^t)^2)\right)
\end{aligned}$$

$$\leq \sum_{r=1}^{R} \frac{P^t(r) - \frac{\beta}{R}}{1-\beta}(1 + \eta I^t(r)s^t + (\eta I^t(r)s^t)^2) - 1$$

$$\leq \sum_{r=1}^{R} \frac{P^t(r)}{1-\beta}(\eta I^t(r)s^t + (\eta I^t(r)s^t)^2) - \frac{\eta\beta}{R(1-\beta)} \sum_{r=1}^{R} I^t(r)s^t,$$

where the first equality uses the definition of W^{t+1}, the second equality breaks the sum into two parts and uses $\mathbb{E}[Y^b] = s^b$, the third equality uses the definition of the sampling distribution P^t, the fourth inequality uses $e^z \leq 1 + z + z^2, \forall z \leq 1$, the fifth inequality uses $\log z \leq z - 1, \forall z \geq 0$, and the last inequality holds by canceling out terms and realizing that $-\sum_{r=1}^{R}(\eta I^t(r)s^t)^2 \leq 0$. Noticing that $\sum_{t=1}^{T} \log \frac{W^{t+1}}{W^t} = \log \frac{W^{T+1}}{W^1}$, we can sum both sides for $t = 1, \ldots, T$ and compare with the lower bound in (15) and obtain

$$\begin{aligned}
&\eta \sum_{t=1}^{T} a^t(\gamma)(h_\gamma(\xi_\gamma)) - \log R - \eta \sum_{t=1}^{T} a^t(\gamma) \\
&\quad \times \left(2\alpha^t \|g_\gamma(x^t; \theta_\gamma^{t-1}/\sqrt{m})\|_{(V_\gamma^{t-1})^{-1}} + 3O(m^{-1/6})\right) \\
&\leq \sum_{t=1}^{T}\Big(\sum_{r=1}^{R} \frac{P^t(r)}{1-\beta}(\eta I^t(r)s^t + (\eta I^t(r)s^t)^2) \\
&\quad - \frac{\eta\beta}{R(1-\beta)} \sum_{r=1}^{R} I^t(r)s^t\Big). \qquad (16)
\end{aligned}$$

Reordering and multiplying both sides by $\frac{1-\beta}{\eta}$ gives

$$\begin{aligned}
&\sum_{t=1}^{T}\left(a^t(\gamma)h_\gamma(\xi_\gamma) - \sum_{r=1}^{R} \mathbf{1}\{r^t = r\}s^t\right) \\
&\leq \frac{1-\beta}{\eta}\log R + \sum_{t=1}^{T}\sum_{r=1}^{R} \eta I_t(r)(s^t)^2 \\
&\quad + \beta \sum_{t=1}^{T}\left(a^t(\gamma)h_\gamma(\xi_\gamma) - \frac{1}{R}\sum_{r=1}^{R} I^t(r)s^t\right) + (1-\beta) \\
&\quad \times \sum_{t=1}^{T} a^t(\gamma)\left(2\alpha^t \|r_\gamma(x^t; \theta_\gamma^{t-1}/\sqrt{m})\|_{(V_\gamma^{t-1})^{-1}} + 3O(m^{-1/6})\right) \qquad (17)
\end{aligned}$$

Now, consider the definition of the regret in (5),

$$\begin{aligned}
&\textsc{Regret}(T) \\
&= \sum_{t=1}^{T}\left(s^t(\gamma, \xi_\gamma^t) - \sum_{r=1}^{R} \mathbf{1}\{r^t = r\}s^t\right) \\
&\leq \frac{1-\beta}{\eta}\log R + \eta RT + \beta T + \frac{1-\beta}{\beta}\sum_{t=1}^{T} a^t(\gamma) \\
&\quad \times \left(2\alpha^t \|r_\gamma(x^t; \theta_\gamma^{t-1}/\sqrt{m})\|_{(V_\gamma^{t-1})^{-1}} + 3O(m^{-1/6})\right) \\
&\leq \frac{1}{\eta}\log R + \eta RT + \beta T + \frac{1-\beta}{\beta} \\
&\quad \times \left[O(\sqrt{\tilde{d}\log(1+T)})O(\sqrt{T\tilde{d}\log(1+T)}) + 3O(m^{-1/6})\right], \qquad (18)
\end{aligned}$$

where the last inequality holds for Lemma 4 and sufficiently large m.

Finally, by setting $\beta = T^{-1/4}\sqrt{\log(T)}$ and $\eta = T^{-1/2}$, we have $\textsc{Regret}(T) = O(T^{3/4}\sqrt{\log(T)})$. □

REFERENCES

[1] C. Qiao, Y. Zhao, G. Zhao, and H. Xu, "Quantum data networking for distributed quantum computing: Opportunities and challenges," in *Proc. IEEE Conf. Comput. Commun. Workshops*, May 2022, pp. 1–6.

[2] A. Steane, "Quantum computing," *Rep. Progr. Phys.*, vol. 61, no. 2, p. 117, 1998.

[3] M. Mehic et al., "Quantum key distribution: A networking perspective," *ACM Comput. Surv.*, vol. 53, no. 5, pp. 1–41, 2020.

[4] W. K. Wootters and W. H. Zurek, "A single quantum cannot be cloned," *Nature*, vol. 299, no. 5886, pp. 802–803, 1982.

[5] L. J. Stephenson et al., "High-rate, high-fidelity entanglement of qubits across an elementary quantum network," *Phys. Rev. Lett.*, vol. 124, no. 11, Mar. 2020, Art. no. 110501.

[6] K. Mattle, H. Weinfurter, P. G. Kwiat, and A. Zeilinger, "Dense coding in experimental quantum communication," *Phys. Rev. Lett.*, vol. 76, no. 25, pp. 4656–4659, Jun. 1996.

[7] T. Jennewein, G. Weihs, J.-W. Pan, and A. Zeilinger, "Experimental nonlocality proof of quantum teleportation and entanglement swapping," *Phys. Rev. Lett.*, vol. 88, no. 1, Dec. 2001, Art. no. 017903.

[8] S. Shi and C. Qian, "Concurrent entanglement routing for quantum networks: Model and designs," in *Proc. Annu. Conf. ACM Special Interest Group Data Commun. Appl., Technol., Archit., Protocols Comput. Commun.*, 2020, pp. 62–75.

[9] L. Yang, Y. Zhao, H. Xu, and C. Qiao, "Online entanglement routing in quantum networks," in *Proc. IEEE/ACM 30th Int. Symp. Quality Service (IWQoS)*, Jun. 2022, pp. 1–10.

[10] L. Yang, Y. Zhao, L. Huang, and C. Qiao, "Asynchronous entanglement provisioning and routing for distributed quantum computing," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, May 2023, pp. 1–10.

[11] Y. Zeng, J. Zhang, J. Liu, Z. Liu, and Y. Yang, "Multi-entanglement routing design over quantum networks," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, May 2022, pp. 510–519.

[12] Y. Zhao and C. Qiao, "Redundant entanglement provisioning and selection for throughput maximization in quantum networks," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, May 2021, pp. 1–10.

[13] G. Zhao, J. Wang, Y. Zhao, H. Xu, and C. Qiao, "Segmented entanglement establishment for throughput maximization in quantum networks," in *Proc. IEEE 42nd Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2022, pp. 45–55.

[14] Y. Zhao, G. Zhao, and C. Qiao, "E2E fidelity aware routing and purification for throughput maximization in quantum networks," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, May 2022, pp. 480–489.

[15] Y. Mao, Y. Liu, and Y. Yang, "Qubit allocation for distributed quantum computing," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, May 2023, pp. 1–10.

[16] A. Farahbakhsh and C. Feng, "Opportunistic routing in quantum networks," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, May 2022, pp. 490–499.

[17] P. H. Che, M. Chen, T. Ho, S. Jaggi, and M. Langberg, "Routing for security in networks with adversarial nodes," in *Proc. Int. Symp. Netw. Coding (NetCod)*, Jun. 2013, pp. 1–6.

[18] M. I. Khan, "Resource-aware task scheduling by an adversarial bandit solver method in wireless sensor networks," *EURASIP J. Wireless Commun. Netw.*, vol. 2016, no. 1, pp. 1–17, Dec. 2016.

[19] P. Zhou, J. Xu, W. Wang, Y. Hu, D. O. Wu, and S. Ji, "Toward optimal adaptive online shortest path routing with acceleration under jamming attack," *IEEE/ACM Trans. Netw.*, vol. 27, no. 5, pp. 1815–1829, Oct. 2019.

[20] D. Bouwmeester, J.-W. Pan, K. Mattle, M. Eibl, H. Weinfurter, and A. Zeilinger, "Experimental quantum teleportation," *Nature*, vol. 390, no. 6660, pp. 575–579, Dec. 1997.

[21] E. Schoute, L. Mancinska, T. Islam, I. Kerenidis, and S. Wehner, "Shortcuts to quantum network routing," 2016, *arXiv:1610.05238*.

[22] M. Pant et al., "Routing entanglement in the quantum Internet," *npj Quantum Inf.*, vol. 5, no. 1, p. 25, Mar. 2019.

[23] K. Chakraborty, F. Rozpedek, A. Dahlberg, and S. Wehner, "Distributed routing in a quantum Internet," 2019, *arXiv:1907.11630*.

[24] G. Vardoyan, S. Guha, P. Nain, and D. Towsley, "On the stochastic analysis of a quantum entanglement switch," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 47, no. 2, pp. 27–29, Dec. 2019.

[25] L. Wang, J. Bian, and J. Xu, "Adaptive user-centric entanglement routing in quantum data networks," in *Proc. IEEE 44th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2024, pp. 1202–1212.
[26] S. Bubeck and N. Cesa-Bianchi, "Regret analysis of stochastic and nonstochastic multi-armed bandit problems," *Mach. Learn.*, vol. 5, no. 1, pp. 1–122, 2012.
[27] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Mach. Learn.*, vol. 47, no. 2, pp. 235–256, 2002.
[28] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "The non-stochastic multiarmed bandit problem," *SIAM J. Comput.*, vol. 32, no. 1, pp. 48–77, 2002.
[29] S. Bubeck and A. Slivkins, "The best of both worlds: Stochastic and adversarial bandits," in *Proc. Conf. Learn. Theory*, Jun. 2012, pp. 1–42.
[30] Y. Seldin and A. Slivkins, "One practical algorithm for both stochastic and adversarial bandits," in *Proc. Int. Conf. Mach. Learn.*, Jun. 2014, pp. 1287–1295.
[31] T. Lykouris, V. Mirrokni, and R. Paes Leme, "Stochastic bandits robust to adversarial corruptions," in *Proc. 50th Annu. ACM SIGACT Symp. Theory Comput.*, Jun. 2018, pp. 114–122.
[32] A. Gupta, T. Koren, and K. Talwar, "Better algorithms for stochastic bandits with adversarial corruptions," in *Proc. Conf. Learn. Theory*, Jun. 2019, pp. 1562–1578.
[33] Y. Huang, L. Zhang, and J. Xu, "Adversarial group linear bandits and its application to collaborative edge inference," in *Proc. IEEE Conf. Comput. Commun.*, May 2023, pp. 1–10.
[34] C. A. Micchelli, Y. Xu, and H. Zhang, "Universal kernels," *J. Mach. Learn. Res.*, vol. 7, no. 12, pp. 1–17, 2006.
[35] S. Liu and M. Zhu, "Distributed inverse constrained reinforcement learning for multi-agent systems," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, Dec. 2022, pp. 33444–33456.
[36] S. Liu and M. Zhu, "Learning multi-agent behaviors from distributed and streaming demonstrations," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 36, Feb. 2024, pp. 1–13.
[37] M. Valko, N. Korda, R. Munos, I. Flaounas, and N. Cristianini, "Finite-time analysis of kernelised contextual bandits," 2013, *arXiv:1309.6869*.
[38] A. Jacot, F. Gabriel, and C. Hongler, "Neural tangent kernel: Convergence and generalization in neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Jun. 2021, pp. 1–10.
[39] D. Zhou, L. Li, and Q. Gu, "Neural contextual bandits with UCB-based exploration," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 11492–11502.
[40] Y. Ban, Y. Yan, A. Banerjee, and J. He, "EE-Net: Exploitation-exploration neural networks in contextual bandits," 2021, *arXiv:2110.03177*.
[41] S. Salgia, "Provably and practically efficient neural contextual bandits," in *Proc. Int. Conf. Mach. Learn.*, Jul. 2023, pp. 29800–29844.
[42] A. Dahlberg et al., "A link layer protocol for quantum networks," in *Proc. ACM Special Interest Group Data Commun.*, Aug. 2019, pp. 159–173.
[43] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

Yin Huang received the B.E. degree in communication engineering from Beijing Jiaotong University in 2020. He is currently pursuing the Ph.D. degree with the Electrical and Computer Engineering Department, University of Florida. His research interests include multi-armed bandits, edge computing, and quantum networks.

Lei Wang (Graduate Student Member, IEEE) received the B.E. degree in electronic information engineering from the University of Electronic Science and Technology of China in 2020 and the M.S. degree in electrical and computer engineering from the University of California at Los Angeles in 2022. He is currently pursuing the Ph.D. degree with the Electrical and Computer Engineering Department, University of Florida. His research interests include federated foundation models, heterogeneous federated learning, and distributed quantum networks.

Jie Xu (Senior Member, IEEE) received the B.S. and M.S. degrees in electronic engineering from Tsinghua University, Beijing, China, in 2008 and 2010, respectively, and the Ph.D. degree in electrical engineering from UCLA in 2015. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, University of Florida. His research interests include mobile edge computing/intelligence, machine learning for networks, and network security. He received the NSF CAREER Award in 2021.