Learning a Decentralized Medium Access Control Protocol for Shared Message Transmission

Lorenzo Mario Amorosa, *Member, IEEE*, Zhan Gao, *Graduate Student Member, IEEE*, Roberto Verdone, *Senior Member, IEEE*, Petar Popovski, *Fellow, IEEE*, and Deniz Gündüz, *Fellow, IEEE*

***Abstract*—In large-scale Internet of things networks, efficient medium access control (MAC) is critical due to the growing number of devices competing for limited communication resources. In this work, we consider a new challenge in which a set of nodes must transmit a set of shared messages to a central controller, without inter-node communication or retransmissions. Messages are distributed among random subsets of nodes, which must implicitly coordinate their transmissions over shared communication opportunities. The objective is to guarantee the delivery of all shared messages, regardless of which nodes transmit them. We first prove the optimality of deterministic strategies, and characterize the success rate degradation of a deterministic strategy under dynamic message-transmission patterns. To solve this problem, we propose a decentralized learning-based framework that enables nodes to autonomously synthesize deterministic transmission strategies aiming to maximize message delivery success, together with an online adaptation mechanism that maintains stable performance in dynamic scenarios. Extensive simulations validate the framework's effectiveness, scalability, and adaptability, demonstrating its robustness to varying network sizes and fast adaptation to dynamic changes in transmission patterns, outperforming existing multi-armed bandit approaches.**

***Index Terms*—Decentralized Learning, Distributed Coordination, Internet of Things, Medium Access Control**

I. Introduction

With exponentially growing devices in modern wireless Internet of things (IoT) networks, medium access control (MAC) to coordinate multiple nodes contending for shared communication opportunities (in time, frequency, etc.) is increasingly challenging, especially under tight resources and large node counts [1–3]. Distributed random access strategies, such as slotted ALOHA, are generally preferred because they minimize signaling overhead and do not require centralized scheduling [4]. These advantages become particularly important as the network size grows, although the occurrence of collisions, i.e., the risk that two or more nodes attempt to access the same transmission resource, remains a significant challenge. This challenge becomes more complex in large-scale IoT deployments for tasks such as critical infrastructure monitoring. When multiple nodes report the same event, e.g., multiple sensors alarm on an industrial anomaly [5–7], handling resource contention in a collision-free manner becomes critical.

L.M. Amorosa and R. Verdone are with the Department of Electrical, Electronic and Information Engineering (DEI), "Guglielmo Marconi", University of Bologna & WiLab - National Wireless Communication Laboratory (CNIT), Bologna 40136, Italy. E-mail: {lorenzomario.amorosa, roberto.verdone}@unibo.it

Z. Gao is with the Department of Computer Science and Technology, University of Cambridge, Cambridge CB3 0FD, U.K. *(Corresponding author.)* E-mail: zg292@cam.ac.uk

P. Popovski is with the Department of Electronic Systems, Aalborg University, 9220 Aalborg, Denmark. E-mail: petarp@es.aau.dk

D. Gündüz is with the Department of Electrical and Electronic Engineering, Imperial College London, London SW7 2AZ, U.K. E-mail: d.gunduz@imperial.ac.uk

In this work, we consider a complex problem, in which a set of nodes must deliver a set of *shared* messages to a central controller, without inter-node communication or retransmissions. Unlike conventional random access scenarios, where each node transmits its own independent data, the presence of shared messages introduces additional complexity. Specifically, these messages are distributed among random subsets of nodes, indicating that multiple nodes may simultaneously attempt to transmit the same message over shared communication opportunities. We refer to the nodes that have to send one or more messages in the current time slot as the *active nodes*, and all the nodes that have to send the same message belong to the same *active set*. Complexity arises because nodes need to coordinate who should transmit a shared message and over which transmission opportunity without retransmissions or central coordination, and the network must rely on implicit mechanisms to ensure each shared message is delivered successfully. In this case, a practical mitigation is permitting *message repetition* across communication opportunities (e.g., IRSA-type schemes [8]) to increase the chance that at least one copy of each shared message finds a collision-free opportunity.

Fig. 1 demonstrates an example example where message repetition is crucial. Consider four nodes and two shared messages, l_1 (possibly available to nodes n_1, n_2) and l_2 (possibly available to nodes n_3, n_4). There are three shared transmission opportunities. We further assume that, at each time slot t, the set of active nodes can only be one of $\{n_2, n_3\}$, $\{n_2, n_4\}$, $\{n_1, n_2, n_4\}$, or $\{n_1, n_3, n_4\}$. We show by contradiction that, under such an activity pattern, any strategy that restricts each node to transmit its message on a single opportunity will fail, and message repetition is required to guarantee the sucessful delivery of all shared messages. Specifically, assume there exists a successful strategy in which every node transmits at most once at each t (i.e., each node picks exactly one opportunity). We consider different possible active sets step by step: (i) When the active set is $\{n_1, n_2, n_4\}$, nodes n_1 and n_2 must ensure l_1 is received. Because there is no retransmission or inter-node coordination, and neither one node knows whether the other holds l_1, n_1 and n_2 must both attempt to transmit, yet they cannot do so on the same opportunity (as that would cause a self-collision for l_1). Similarly, n_4 (which carries l_2) must avoid colliding with the transmissions of n_1 and n_2 to deliver l_2 successfully. With only three opportunities available, this forces n_1, n_2, and n_4 to occupy three distinct opportunities; (ii) Next consider the active set $\{n_1, n_3, n_4\}$. It follows the same reason as in (i) that n_3 and n_4 must attempt to transmit, but cannot use the same opportunity as that would result in a self-collision for l_2. Moreover, n_1 (carrying l_1) must use an opportunity different from those chosen by n_3 and n_4 to prevent an inter-message collision. Because n_1 and n_4 already had fixed (distinct) opportunities from step (i), these constraints force n_3 to choose the same opportunity that n_2 used in step (i); (iii) Now consider the

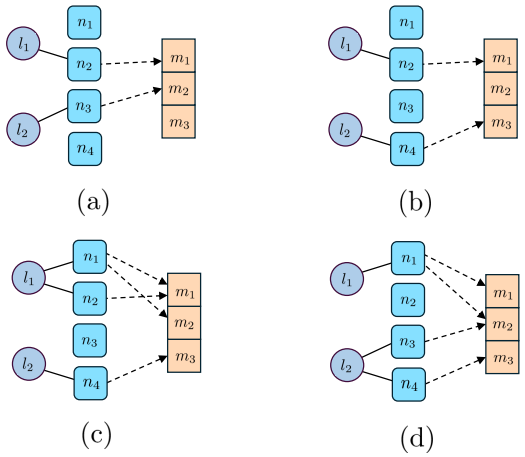


Fig. 1: Example scenario where message repetition is crucial. Panels (a)-(d) demonstrate how a strategy with message repetition handles four possible active sets of shared messages with limited communication opportunities, ensuring that for every active set at least one copy of each shared message can be delivered without collision.

active set $\{n_2, n_3\}$. From the assignments deduced above, n_2 and n_3 have been driven to the same opportunity, so they would collide when both are active. That collision would cause both messages to fail delivery at that time slot t, contradicting the assumption that the single-transmission strategy always succeeds. In contrast, as shown in Fig. 1, a strategy that allows controlled message repetition, i.e., assigning n_1 to transmit on both opportunity 1 and opportunity 2, n_2 on opportunity 1, n_3 on opportunity 2 and n_4 on opportunity 3, ensures that for every possible active set at least one copy of each shared message finds a collision-free opportunity.

Recognizing the significance of message repetition for transmission success, our goal is to maximize the probability of successful delivery of shared messages under strict resource limits. This objective is of practical importance in many IoT applications, particularly those involving critical monitoring tasks. For example, when multiple sensors detect the same fault in an industrial process, it is essential to ensure that the fault notification reaches its destination at least once. In these cases, traditional throughput-maximizing approaches are inefficient, as they assume each packet of individual node carries independent information; instead nodes must coordinate implicitly to guarantee at least one delivery per shared message. Achieving this goal becomes particularly challenging in dynamic environments where the set of nodes transmitting messages changes over time. Dynamic node activation patterns, which evolve probabilistically from one time slot to the next, result in constantly shifting active sets. This unpredictability is compounded by the fact that activation probabilities are neither known in advance nor static; they may depend on conditional relationships between nodes or emerge randomly. For example, a node may be more likely to be active if another correlated node is also active, or activations of two nodes may be independent of each other. Crucially, each node only knows its own activation status at any given moment, making real-time coordination with other nodes challenging.

We also address IoT networks operating in continuously evolving environments, where new sources of messages may emerge, nodes may move, and the phenomena being monitored could change [9, 10]. This requires the communication protocol adapting to time-varying activation probabilities and handling the inherent uncertainty in node activations. These challenges, i.e., the lack of global information and the dynamics of message transmission scenarios, make it essential to develop efficient and decentralized access control mechanisms that ensure reliable and timely message delivery, while allowing to tune transmission strategies in accordance with changes in node activation patterns. These scenarios are especially relevant for ultra-reliable low-latency communication (URLLC) and energy-constrained IoT deployments [11], where strict latency and power budgets rule out costly retransmissions and prolonged coordination.

A. Related Work

Several works in the literature address scenarios that share similarities with the one proposed in this paper. In [12], the authors introduce a framework for critical IoT applications by prioritizing *alarm messages* over *standard messages* for all nodes. They assume that the number of active nodes is random and unknown in each time slot, deriving a random-coding achievability bound on the misdetection and false positive probabilities of both message types over a Gaussian multiple access channel. Although their focus is on message type classification, the random nature of node activation aligns with some aspects of our problem. In [13], the authors focus on improving detection performance in grant-free random access systems where node activation patterns are correlated. This correlation aids to enhance system performance, but the work primarily emphasizes activity detection rather than message delivery coordination which are central to our approach.

The work in [14] is the closest one to the current paper. The authors consider a scenario where a single shared message must be sent by a set of nodes over shared resources, with signaling constraints necessitating a random access scheme. The nodes must coordinate implicitly without inter-node communication to ensure that at least one transmission avoids a collision. They propose a solution based on a distributed version of Thompson sampling for Bernoulli multi-armed bandit (MAB) algorithms that can operate independently on each node after an initial training phase. While effective, their approach does not fully address scalability concerns as the number of nodes grows. Our work differs in that we consider the transmission of multiple shared messages and explicitly address the need to adapt to dynamic activation probabilities that govern the active sets and can continuously evolve. Other works, such as [15] and [16], focus on throughput maximization by leveraging correlations in activation patterns. Similarly, in [17], the authors propose a resource allocation algorithm to design a random access scheme that maximizes both throughput and sum-rate by leveraging correlated activation patterns. These works are focused on throughput optimization, which contrasts with our goal of improving the reliability and success probability of transmitting shared messages.

Learning-based techniques have recently been explored in throughput maximization tasks [18–22]. In [18], multi-agent reinforcement learning (MARL) is used to design grant-free random access schemes suited for massive machine-type communication (mMTC) scenarios. Similarly, in [19], reinforcement learning (RL) is employed to develop a coordinated random access scheme for industrial IoT networks, where nodes generate sporadic, correlated traffic. However, both RL and MARL approaches tend to suffer from scalability issues, especially in large-scale random access schemes. A more lightweight approach is proposed in [21], where the authors develop a stochastic gradient descent-based algorithm

to optimize slot allocation probabilities. While promising, their method assumes knowledge of activity estimation errors, which may not always be available in real-world settings. The work in [23] propose a graph neural network (GNN)-based, distributed framework to optimize uplink scheduling requests in industrial IoT scenarios, reducing redundant scheduling messages and communication overhead; unlike our work, which targets reliable transmission of multiple shared messages, their focus is on scheduling-request reduction and efficient distributed coordination. Finally, a related problem is explored in [22], where the authors aim to reliably transmit a shared alarm message by superposing individual signals. Unlike the collision model we adopted, this approach does not require coordination between the nodes and relies on signal superposition for reliable transmission.

In this work, we extend the study of decentralized MAC scenarios by introducing the challenge of transmitting multiple shared messages under dynamic activation probabilities while maintaining low complexity and scalability, filling a gap not fully addressed by previous methods.

B. Our Contributions

In this work, we address a novel MAC problem where multiple shared messages must be transmitted by nodes over shared communication resources. Due to strict constraints on signaling overhead, nodes are required to rely on a local access scheme with no centralized coordination or explicit communication among nodes. The main challenge is to guarantee every shared message delivered at least once, which requires nodes implicitly coordinating their transmissions in a fully decentralized manner, under dynamic node activation patterns and without the possibility of retransmissions. More in detail, our contributions can be summarized as follows:

1) We propose a new MAC problem for coordinated transmission of multiple shared messages over limited resources. Analyzing its theoretical foundations, we prove that, if a unique optimal protocol exists for this task, it must be deterministic. Randomized (i.e., stochastic) solutions only exist as convex combinations of equally optimal deterministic strategies. Moreover, we show that this problem is NP-hard.
2) We develop an unsupervised and stateless learning-based method to solve the proposed problem. Our method is lightweight and decentralized, where each node deploys a local deep neural network (DNN) to select transmission opportunities. It eliminates the need of inter-node communication and central coordination, significantly simplifying the protocol design and system requirements.
3) We derive a theoretical upper bound on the degradation of transmission success probability when node activation probabilities change due to environment variations, indicating resilience of our solution to mild environment changes. When these changes are significant, we introduce an online learning mechanism to maintain solution robustness to time-varying environments.
4) We perform extensive experiments to show that our framework outperforms baselines, scales to large networks, and exhibits robustness to dynamic environment changes. All code is available on GitHub[1].

The remainder of this paper is structured as follows. Sec. II formulates the problem setting and Sec. III presents theoretical findings. In Sec. IV, we detail our proposed unsupervised learning framework and online learning mechanism, which are then evaluated through numerical simulations in Sec. V. Finally, conclusions are drawn in Sec. VI.

II. Problem Formulation

We consider a wireless network consisting of N nodes $\mathcal{N} = \{n_1, \dots, n_N\}$, which are tasked to transmit a set of L shared messages $\mathcal{L} = \{l_1 \dots, l_L\}$ to a central controller S using $M \geq L$ orthogonal transmission opportunities (in time, frequency, etc.). At any given time t,[2] message $l \in \mathcal{L}$ becomes available to a subset of *active nodes*, denoted as $\mathcal{A}_l$, where $\mathcal{A}_l \subseteq \mathcal{N}$ and $0 < |\mathcal{A}_l| \leq N$. These active nodes are responsible for transmitting message l to the controller. Importantly, nodes belonging to $\mathcal{A}_l$ are not aware of which other nodes are also responsible for transmitting the same message, nor do they have direct communication with each other. The objective of the system is to ensure that each message $l \in \mathcal{L}$ is successfully transmitted to S at least once over the M transmission opportunities. This must be achieved regardless of which specific node transmits the message, emphasizing the reliability of shared message delivery rather than individual node success. The inherent challenge lies in the absence of inter-node awareness, i.e., each node does not know which other nodes are responsible for transmitting the same message as itself, precluding explicit coordination. An overall depiction of the scenario is provided in Fig. 2.

The coordination problem highly depends on the pattern according to which the nodes become active. We model all active nodes at time t as $\mathcal{A} = \mathcal{A}_1 \parallel \mathcal{A}_2 \parallel \dots \parallel \mathcal{A}_L$, where $\cdot \parallel \cdot$ is the concatenation operator. Here, a node that is active for ℓ messages simultaneously is counted ℓ times. For the cardinality of A, we have $|\mathcal{A}| \leq L \cdot N = K$. We model $\mathcal{A}$ as a random variable with probability mass function $p(\mathcal{A})$. Note that $\mathcal{A}$ is sampled from all possible subsets of active nodes, i.e., $\mathcal{A} \in \mathcal{R}_{\mathcal{A}} = \{\mathcal{P}_1, \dots, \mathcal{P}_{2^K}\}$, where the probability of the set $\mathcal{P}_i$ is p_i for $i = 1, \dots, 2^K$. To account for possible correlations among nodes that are active for the same message (e.g., device groups that tend to co-activate), we allow $p(\mathcal{A})$ to exhibit certain correlation structure. A convenient and flexible way to encode this structure is to place a Dirichlet prior on the probability vector over the 2^K possible activation patterns. By definition, these probabilities belong to the standard $(2^K - 1)$-dimensional simplex

$$\Delta^{2^K} = \left\{ (p_1, \dots, p_{2^K}) : \sum_{i=1}^{2^K} p_i = 1,\ p_i \geq 0 \right\}. \quad (1)$$

Since Δ^{2^K} can be seen as a sample of a Dirichlet distribution, we can model it as

$$(p_1, \dots, p_{2^K}) \sim \text{Dir}(\alpha_1, \dots, \alpha_{2^K}), \quad (2)$$

where $\alpha_1, \dots, \alpha_{2^K}$ are the concentration parameters with each $\alpha_i > 0$. These parameters determine the shape of the Dirichlet distribution. If $\alpha_i > 1$, the distribution is more concentrated around the center of the $(2^K - 1)$-dimensional simplex. If $\alpha_i < 1$, the distribution tends to favor extreme values, inducing a strong preference for a few recurring patterns. When all parameters are $\alpha_i = 1$ for $i = 1, \dots, 2^K$, the Dirichlet distribution becomes the uniform distribution over the simplex.

[1]Code available at: https://github.com/Lostefra/Learning-Dec-MAC.

[2]For simplicity, we omit the subscript t in following expressions to improve readability.

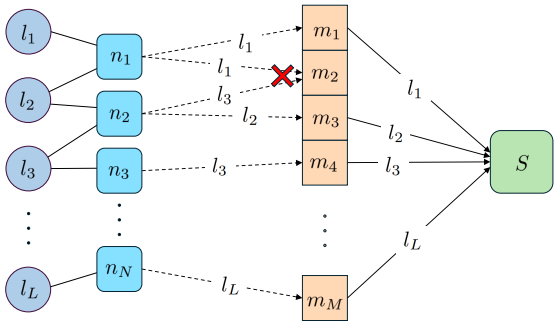


Fig. 2: Reference scenario comprising N nodes transmitting L messages over M communication opportunities to a central controller S. Nodes n_1 and n_2 transmit over the same communication opportunity m_2 simultaneously, resulting in a "collision".

Each active node $n \in \mathcal{A}_l$ needs to decide which communication opportunities it uses to transmit message l. It is allowed to use multiple communication opportunities to transmit the same message, the principle behind which is to leverage repetitions to improve throughput and reliability, as in irregular repetition slotted ALOHA (IRSA) schemes [8]. We model this decision as a vector $\mathbf{x}_{l,n} \in \{0,1\}^M$, referred to as a *move*, where $[\mathbf{x}_{l,n}]_m = x_{l,n,m} = 1$ if node n transmits message l over the m-th opportunity. We represent the moves of all nodes for all messages in a tensor $\mathbf{X} \in \{0,1\}^{L\times N\times M}$, hence we have a total of 2^{LNM} different possible moves. We say a transmission to be successful if the central controller receives all messages $\mathcal{L}$. Furthermore, a message is received successfully through the m-th opportunity only if at most one node transmits over communication opportunity m. In this context, we model the condition for a successful transmission as

$$\xi(\mathcal{A},\mathbf{X}) = \bigwedge_{l=1}^{L} I\Bigg(\exists m \in \{1,\dots,M\} \;:\; \sum_{n\in\mathcal{A}_l} x_{l,n,m} = 1 \;\wedge\; \sum_{\substack{\bar{l}=1\\ \bar{l}\neq l}}^{L}\sum_{\bar{n}\in\mathcal{A}_{\bar{l}}} x_{\bar{l},\bar{n},m} = 0\Bigg)\,, \tag{3}$$

where $\wedge$ represents the *logical and* operator and $I(\cdot)$ denotes the indicator function, which returns 1 if the condition is met and 0 otherwise.

For example in Fig. 2, consider nodes $n_1, n_2, \dots, n_N$ with message availabilities $(l_1, l_2), (l_2, l_3), (l_3), \dots, (l_L)$, respectively. Node n_1 sends message l_1 on communication opportunities m_1 and m_2; n_2 sends l_2 on m_3 and l_3 on m_2; n_3 sends l_3 on m_4; n_N sends l_L on m_M. The transmission is successful, i.e., the success indicator $\xi(\mathcal{A},\mathbf{X})$ in (3) is equal to 1, although n_1 and n_2 both transmit on m_2 resulting in a collision. This is because for every shared message l there exists at least one transmission opportunity m satisfying the two required conditions: (i) exactly one active node for l transmits on m, and (ii) no node transmitting any other message uses the same m. Concretely, l_1 is received via m_1 (only n_1 transmits l_1 on m_1), l_2 is received via m_3 (only n_2 transmits l_2 on m_3), l_3 is received via m_4 (only n_3 transmits l_3 on m_4), and so on up to l_L on m_M. Since each $l \in \mathcal{L}$ has at least one such interference-free opportunity, the conjunction over messages in (3) evaluates to 1 and the collective transmission is successful.

We define a *local strategy* of node n to transmit message l as a random variable $\Phi_{l,n}$, whose support is the set of all possible moves, and its probability mass function (PMF) $\boldsymbol{\phi}_{l,n}(\mathbf{x}_{l,n}) \in [0,1]$ represents the probability that node n chooses move $\mathbf{x}_{l,n} \in \{0,1\}^M$ when it is active to transmit message l. Moreover, we define the *joint strategy* of all nodes as a random variable $\boldsymbol{\Psi}$ whose PMF $\psi(\mathbf{X}) \in [0,1]$ is the joint PMF of $\{\Phi_{l,n}\}_{l=1,n=1}^{L,N}$, i.e.,

$$\psi(\mathbf{X}) = P\left(\boldsymbol{\Psi} = \mathbf{X}\right) = \prod_{l=1}^{L}\prod_{n=1}^{N} \boldsymbol{\phi}_{l,n}(\mathbf{x}_{l,n})\,, \tag{4}$$

where $P(\cdot)$ denotes a probability. We can then get the expected probability of successful transmission according to the law of total probability as

$$\mathbb{E}[\,\xi \mid \boldsymbol{\Psi}, p\,] = \sum_{\mathcal{A}\in\mathcal{R}_{\mathcal{A}}} p(\mathcal{A}) \sum_{\mathbf{X}\in\{0,1\}^{L\times N\times M}} \xi(\mathcal{A},\mathbf{X})\;\psi(\mathbf{X})\,, \tag{5}$$

where $\mathbb{E}[\cdot]$ is the expectation, and formulate an optimization problem to maximize the probability of successful transmission by finding the optimal joint strategy $\boldsymbol{\Psi}^*$ as

$$\mathbb{E}[\,\xi \mid \boldsymbol{\Psi}^*, p\,] = \max_{\boldsymbol{\Psi}}\; \mathbb{E}[\,\xi \mid \boldsymbol{\Psi}, p\,]\,, \tag{6}$$

referred to as the problem of shared messages' transmission. In Section III, we conduct theoretical analysis to provide interpretations for problem (6) and its optimal solution. In Section IV, we develop a decentralized learning-based method to solve problem (6) and leverage online learning that adapts our method to dynamic wireless scenarios. In Section V, we evaluate the proposed method in extensive numerical experiments to show superior performance.

III. Theoretical Analysis

In the problem of shared message transmission (6), understanding the behaviors and properties of the optimal joint strategies is critical, as it provides valuable insights to efficiently guide the search of solutions. Before proceeding, we define deterministic and randomized joint strategies of problem (6) as follows.

Definition 1 (Deterministic joint strategy)**.** A joint strategy $\boldsymbol{\Psi}$ is said to be *deterministic* if its PMF takes binary values. That is, for every node $n \in \mathcal{N}$ and for every message $l \in \mathcal{L}$, there exists a unique move $\mathbf{x}^*_{l,n} \in \{0,1\}^M$ such that

$$\boldsymbol{\phi}_{l,n}(\mathbf{x}^*_{l,n}) = 1, \tag{7}$$

$$\boldsymbol{\phi}_{l,n}(\mathbf{x}_{l,n}) = 0, \quad \forall\, \mathbf{x}_{l,n} \neq \mathbf{x}^*_{l,n}\,. \tag{8}$$

Equivalently, the joint PMF $\psi(\mathbf{X})$ satisfies

$$\psi(\mathbf{X}) \in \{0,1\}, \quad \forall\; \mathbf{X} \in \{0,1\}^{L\times N\times M}\,, \tag{9}$$

implying that the strategy always selects the same moves.

Definition 2 (Randomized joint strategy)**.** A joint strategy $\boldsymbol{\Psi}$ is said to be *randomized* if there exists at least one node $n \in \mathcal{N}$ and one message $l \in \mathcal{L}$ for which the corresponding PMF $\boldsymbol{\phi}_{l,n}$ is non binary-valued. That is, there exists at least one move $\mathbf{x}_{l,n} \in \{0,1\}^M$ such that

$$0 < \boldsymbol{\phi}_{l,n}(\mathbf{x}_{l,n}) < 1\,. \tag{10}$$

That is, the joint strategy $\boldsymbol{\Psi}$ inherently involves probabilistic decision-making.

One of the key questions is whether deterministic strategies are sufficient to achieve optimality or randomization is needed. Deterministic strategies are easier to interpret and provide predictable behavior compared to randomized strategies, as they represent a specific subset within the broader class of

randomized strategies and each node makes always the same move given the same problem setting. This motivates us to characterize the deterministic / randomized property of the optimal strategies for problem (6) in the following theorem.

Theorem 1. *The set of optimal joint strategies for the optimization problem defined in (6) either consists of a single deterministic joint strategy or includes a finite number $D > 1$ of deterministic joint strategies along with infinitely many randomized ones, i.e., $\forall \mathbf{X} \in \{0,1\}^{L\times N\times M}$:*

$$\begin{aligned} &\exists!\ \mathbf{\Psi},\ \psi(\mathbf{X}) \in \{0,1\} : \mathbb{E}[\,\xi \mid \mathbf{\Psi}\,] = \mathbb{E}[\,\xi \mid \mathbf{\Psi}^*\,] \\ &\qquad\qquad \oplus \\ &\exists\ \{\mathbf{\Psi}'_i\}_{i=1}^{D},\ \psi'_i(\mathbf{X}) \in \{0,1\}, D > 1 \wedge \\ &\qquad \exists\ \{\mathbf{\Psi}''_j\}_{j=1}^{\infty},\ \psi''_j(\mathbf{X}) \in [0,1] : \\ &\mathbb{E}[\,\xi \mid \mathbf{\Psi}'_i\,] = \mathbb{E}[\,\xi \mid \mathbf{\Psi}''_j\,] = \mathbb{E}[\,\xi \mid \mathbf{\Psi}^*\,]\ , \end{aligned} \tag{11}$$

where $\oplus$ is the *exclusive or*, the existence of an element x is denoted by $\exists x$, while $\exists! x$ specifies that x exists uniquely.

Proof. See Appendix A. □

Theorem 1 reveals a fundamental property of the optimal joint strategies for problem (6). That is, the optimality landscape is structured in one of two distinct ways: (i) it is possible that there exists a unique deterministic joint strategy that achieves the maximum expected performance; (ii) if there are multiple optimal deterministic joint strategies (i.e., a finite set of them), there exists an infinite number of randomized strategies - constructed as convex combinations of those deterministic strategies - that also achieve the same optimal performance. Therefore, it suffices to focus only on deterministic joint strategies when solving problem (6) without loss of optimality. This result motivates to search for deterministic joint strategies, which offer a clear and unambiguous rule for decision-making, without resorting to probabilistic rules. While Theorem 1 provides interpretations for the optimal solution and narrows down the search space of transmission strategies, problem (6) remains significantly difficult to solve. In the following theorem, we formally show that (6) is a NP-hard problem.

Theorem 2. *The optimization problem in (6) is NP-hard.*

Proof. See Appendix A. □

Theorem 2 states that it is indeed challenging to solve problem (6), which corroborates our intuitive observations from its nature of limited resources, message sharing and local information availability. The result identifies the complexity of conventional heuristic methods when solving problem (6) and motivates to leverage learning-based approaches to develop efficient and scalable solutions in a decentralized manner.

In a general case, it is possible that node activation probabilities vary over time due to many different factors, such as new sources of messages, changes in the wireless environment, node mobility, and so on. This indicates that the distribution of active node sets $\mathcal{A}$, modeled by $p(\mathcal{A})$, can vary over time. As a consequence, an important property that requires attention is the robustness of a joint strategy $\mathbf{\Psi}$ to distribution changes of $\mathcal{A}$ caused by external environment factors, i.e., how much $\xi' = \mathbb{E}[\,\xi \mid \mathbf{\Psi}, p'\,]$ can degrade if the distribution shifts to p'' and the expected probability of successful transmission becomes $\xi'' = \mathbb{E}[\,\xi \mid \mathbf{\Psi}, p''\,]$. If we model p' and p'' as two independent and identically distributed (i.i.d.) realizations of the same Dirichlet distribution and consider $\mathbf{\Psi}$ to be deterministic, the following theorem derives an upper bound on the probability that the degradation exceeds a threshold error, i.e., $|\xi' - \xi''| \geq \eta$ with $\eta \in [0,1]$ the threshold error.

Theorem 3. *Given $p', p'' \sim \mathrm{Dir}(\alpha_1, \ldots, \alpha_{2^K})$ [cf. (2)], a deterministic joint strategy $\mathbf{\Psi}$, and defining $\xi' = \mathbb{E}[\,\xi \mid \mathbf{\Psi}, p'\,]$, $\xi'' = \mathbb{E}[\,\xi \mid \mathbf{\Psi}, p''\,]$, and $\alpha_\varepsilon = \min\{\alpha_i\}_{i=1}^{2^K}$, then*

$$P(|\xi' - \xi''| \geq \eta) < \frac{2}{\eta^2\ (1 + \alpha_\varepsilon\ 2^K)}\ , \tag{12}$$

for any threshold $\eta \in (0,1]$.

Proof. See Appendix A. □

Theorem 3 quantifies the performance degradation of the transmission strategy caused by environment changes, i.e., distribution shifts of active node sets $\mathcal{A}$. The result indicates that when distribution shifts are mild, the transmission strategy is able to maintain satisfactory performance without need of re-design. Moreover, the bound derived in (12) is particularly useful when we consider large scenarios comprising many nodes and messages. Having fixed η and α_ε, an increase in the number of nodes N or in the number of messages L results in a smaller upper bound, allowing for a lower degradation of the value of $\mathbb{E}[\,\xi \mid \mathbf{\Psi}, p'\,]$ when p' shifts to p''.

IV. MAC Protocol Learning

With the deterministic nature of the optimal transmission strategy [cf. Theorem 1] and the NP-hard complexity of the shared message transmission problem [cf. Theorem 2], we propose a decentralized, unsupervised, and stateless learning-based (DUSL) algorithm to solve problem (6). Firstly, DUSL is decentralized, where each node $n \in \mathcal{N}$ is associated with a distinct DNN and determines independently its strategy to transmit messages $\mathcal{L}$ over the M opportunities, without requiring communication among nodes. Secondly, DUSL is unsupervised because we update DNN parameters using only acknowledgments (ACKs) for successful transmissions as feedback, without relying on any oracle or ground truth. Thirdly, DUSL is stateless since it does not need explicit knowledge of the distribution of active node sets p, but instead learns to coordinate nodes' message transmission through trial and error during the training phase.

DUSL conducts centralized training with decentralized execution (CTDE). At training time, we leverage a centralized feedback mechanism to guide the learning process, and perform a stochastic training procedure that enables gradient backpropagation for optimization with non-differentiable ACK feedback. At inference time, each node deploys the learned policy locally in a decentralized manner, without the operational overhead of a central controller. Moreover, the learned policy is applied in a deterministic manner, where each node follows a fixed decision rule based on the trained DNN parameters to improve computational and energy efficiency, reduce inference latency, and provide interpretability. The overall training and inference procedures are presented in Algorithms 1-2 and depicted in Fig. 3. We provide details in the following subsections.

A. DNN parameterization

We parameterize the transmission strategy of each node n as a set of L DNNs [24, 25], each corresponding to one message. For each message l, the transmission decision is computed by a dedicated DNN that consists of a single input neuron, H hidden layers, and M output neurons as

$$\mathbf{Z}_n^{(l)} = \mathbf{W}_{n,H}^{(l)} \sigma\Big(\mathbf{W}_{n,H-1}^{(l)} \sigma\Big(\cdots \mathbf{W}_{n,1}^{(l)} s\Big)\Big), \tag{13}$$

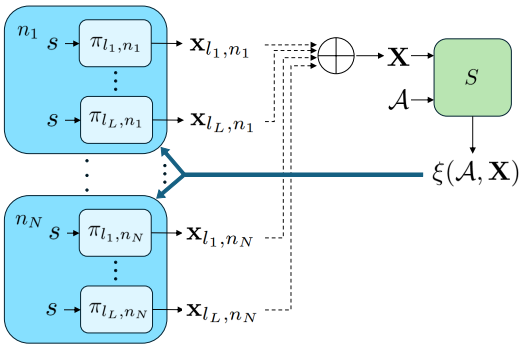


Fig. 3: DUSL learning framework: Each node independently determines communication strategies for messages using L local DNNs, where feedback ξ guides local training procedures.

where $\{\mathbf{W}_{n,h}^{(l)}\}_{h=1}^{H} \in \mathbb{R}^{F_h \times F_{h-1}}$ are the linear operation matrices for the l-th DNN of node n, F_h is the feature dimension at layer h, and $\sigma(\cdot)$ is the nonlinearity. The single neuron input s does not encode any node-specific state but is solely used to facilitate the exploration of new moves. Details are further discussed in Sec. IV-B. The M output neurons of each DNN are passed through a sigmoid activation function to obtain probabilities, i.e.,

$$[\tilde{\mathbf{Z}}_n^{(l)}]_m = \sigma([\mathbf{Z}_n^{(l)}]_m), \quad m = 1, \dots, M, \tag{14}$$

where $[\tilde{\mathbf{Z}}_n^{(l)}]_m$ represents the probability that the l-th message is transmitted utilizing the m-th available transmission opportunity by node n. Each DNN can be represented as a nonlinear mapping $\chi_{l,n}(s, \mathcal{W}_n^{(l)})$, where $\mathcal{W}_n^{(l)}$ collects the parameters $\{\mathbf{W}_{n,h}^{(l)}\}_{h=1}^{H}$ for the l-th network. To determine the transmission move of each active node $\{\mathbf{x}_{l,n}\}_{l=1}^{L}$, we independently sample from $L \cdot M$ Bernoulli distributions with the corresponding probabilities $\{[\tilde{\mathbf{Z}}_n^{(l)}]_m\}_{m=1,l=1}^{M,L}$[3].

The success of the m-th Bernoulli variable for the l-th DNN indicates that the node attempts to use the m-th opportunity to transmit the l-th message. We note that this parameterization does not prevent a node from selecting the same transmission opportunity for multiple messages. However, as detailed in the following section, the training procedure relies on centralized ACK feedback and thus, DUSL will implicitly coordinate nodes and disfavor self-colliding strategies over time to increase the number of successful ACKs. Extending the DNN parametrization to ensure selecting each resource at most once is left for future work.

B. Training Procedure

Our training algorithm is based on a reward-driven policy gradient approach, similar to the REINFORCE algorithm [26] but in an unsupervised manner and in a CTDE setting. At its core, the objective is to maximize the reward, i.e., expected success rate $\mathbb{E}[\,\xi \mid \mathbf{\Psi}, p\,]$ [cf. (5)], over the transmission decisions taken by all active nodes. The following steps break down the procedure:

1) *Initialization*. For each message $l \in \mathcal{L}$ and node $n \in \mathcal{N}$, we initialize a neural network policy $\pi_{l,n}$ with parameters $\boldsymbol{\theta}_{l,n}$. Each $\pi_{l,n}$ outputs M independent probabilities corresponding to the M transmission opportunities. We set the total number of training epochs T, a learning rate $\lambda \in \mathbb{R}^+$, and an exploration decay factor $\varepsilon \in \mathbb{R}^+$ that controls the injected randomness during training.
2) *Exploration and action selection*. At each epoch t, a set of active nodes $\mathcal{A}$ is sampled from the overall node set, and a scalar s is sampled from a Gaussian distribution $s \sim \text{Gaussian}(0, e^{-\varepsilon t})$. This noise s is injected to induce stochasticity in the action selection and to guide the policy exploring the action space during training. The decay factor $e^{-\varepsilon t}$ decreases over time, which enables a shift from exploration to exploitation. For each active node $n \in \mathcal{A}_l$ corresponding to message l, the policy $\pi_{l,n}(s)$ is queried to output a set of probabilities for the M opportunities
$$[\tilde{\mathbf{Z}}_n^{(l)}]_m = \pi_{l,n}(s). \tag{15}$$
The value of $\tilde{\mathbf{Z}}_{n,m}^{(l)}$ is interpreted as the probability of using the m-th transmission opportunity to transmit the l-th message. For each $\tilde{\mathbf{Z}}_{n,m}^{(l)}$, a Bernoulli random variable $m_{l,n}^m$ is sampled as
$$m_{l,n}^m \sim \text{Bernoulli}(\tilde{\mathbf{Z}}_{n,m}^{(l)}) \tag{16}$$
These sampled outcomes determine the move $\mathbf{x}_{l,n}$ of node n for message l.
3) *Reward computation*. The reward signal ξ is the only centralized element of the DUSL framework and is used exclusively for policy updates. Specifically, a central controller collects the messages resulting from the joint transmissions, computes a unique scalar reward ξ that measures overall transmission success, and broadcasts this feedback to all N nodes at each iteration. Specifically, the controller attempts to decode every communication opportunity after each joint transmission, i.e., it builds per-message ACKs $y_l \in \{0, 1\}$ (1 if message l was decoded on any opportunity), and broadcasts the scalar reward $\xi = \prod_{l=1}^{L} y_l$ for policy updates. The reward signal is positive if, and only if, the collective transmissions were successful, which indicates that every shared message was received by the controller at least once. Therefore, each node learns to optimize its behavior based on system-wide performance rather than individual transmission success, promoting coordinated strategies without direct inter-node communication. We remark that ξ is only needed for policy updates, but not required during deployment. This ensures that the trained system operates in a completely decentralized manner, i.e., each node can execute its learned policy locally without the operational overhead of a controller.
4) *Policy gradient update*. Each neural network policy $\pi_{l,n}$ is updated with gradient descent in an unsupervised manner. However, since the reward $\xi(\mathcal{A}, \mathbf{X})$ is non-differentiable [cf. (3)], the gradient cannot be computed directly. We leverage the policy gradient following the REINFORCE update rule, which is a stochastic approximation for the true gradient [27–30]. The update for the parameters $\boldsymbol{\theta}_{l,n}$ is then given by:
$$\boldsymbol{\theta}_{l,n} \leftarrow \boldsymbol{\theta}_{l,n} + \lambda \cdot \nabla_{\boldsymbol{\theta}_{l,n}} \log \pi_{l,n}\big(m_{l,n}^1, \dots, m_{l,n}^M \,|\, s\big) \cdot \xi(\mathcal{A}, \mathbf{X}) \tag{17}$$
The preceding update has three main components: (i) learning rate λ, which controls the step size; (ii) log-probability gradient $\nabla_{\boldsymbol{\theta}_{l,n}} \log \pi_{l,n}(\cdot)$, which computes the sensitivity of the chosen actions with respect to

[3]The probability $[\tilde{\mathbf{Z}}_n^{(l)}]_m = 0$ for $m = 1, \dots, M$ by default if the node does not belong to the active set of the l-th message.

Algorithm 1 DUSL training procedure

Initialize a DNN $\pi_{l,n}$ with parameters $\boldsymbol{\theta}_{l,n}$ $\forall(l,n) \in \mathcal{L} \times \mathcal{N}$, total number of training epochs T, an exploration decay factor $\varepsilon \in \mathbb{R}^+$, and a learning rate $\lambda \in \mathbb{R}^+$
for $t \in \{1, \ldots, T\}$ **do**
  Sample $\bigcup_{l=1}^{L} \mathcal{A}_l = \mathcal{A}$ and $s \sim \text{Gaussian}(0, e^{-\varepsilon t})$
  for $l \in \mathcal{L}, n \in \mathcal{A}_l$ **do**
    Get probabilities $[\tilde{\mathbf{Z}}_n^{(l)}]_m \leftarrow \pi_{l,n}(s)$
    Sample $m_{l,n}^m \sim \text{Bernoulli}(\tilde{\mathbf{Z}}_{n,m}^{(l)})$, $m \in \{1, \cdots, M\}$
    Determine move $\mathbf{x}_{l,n} \in \mathbf{X}$ according to $m_{l,n}^1, \ldots, m_{l,n}^M$
  end for
  Compute $\xi(\mathcal{A}, \mathbf{X})$ centrally and broadcast it to $\mathcal{N}$
  for $l \in \mathcal{L}, n \in \mathcal{A}_l$ **do**
    $\boldsymbol{\theta}_{l,n} \leftarrow \boldsymbol{\theta}_{l,n} + \lambda \cdot \nabla_{\boldsymbol{\theta}_{l,n}} \log \pi_{l,n} \left(m_{l,n}^1, \ldots, m_{l,n}^M \middle| s \right) \cdot \xi(\mathcal{A}, \mathbf{X})$
  end for
end for

Algorithm 2 Deterministic inference procedure

Precomputation Phase:
for all $l \in \mathcal{L}$ and $n \in \mathcal{N}$ **do**
  Compute probabilities $[\tilde{\mathbf{Z}}_n^{(l)}]_m \leftarrow \pi_{l,n}(0)$
  Round to binary values $m_{l,n}^1, \ldots, m_{l,n}^M \leftarrow [\tilde{\mathbf{Z}}_{n,1}^{(l)}], \ldots, [\tilde{\mathbf{Z}}_{n,M}^{(l)}]$
  Compute and store move $\mathbf{x}_{l,n} \in \mathbf{X}$ based on $m_{l,n}^1, \ldots, m_{l,n}^M$
end for
Inference Phase:
Sample the active node sets $\bigcup_{l=1}^{L} \mathcal{A}_l = \mathcal{A}$
for all $l \in \mathcal{L}$ and $n \in \mathcal{A}_l$ **do**
  Retrieve precomputed move $\mathbf{x}_{l,n}$
end for

the network parameters; (iii) reward weighting $\xi(\mathcal{A}, \mathbf{X})$, which reinforces actions resulting in higher rewards.

5) *Iteration*. The process repeats over T epochs iteratively, refining transmission policies through observed rewards. As the exploration noise s decays, the DNNs focus more on exploiting the learned strategies that maximize the transmission success.

C. Deterministic Inference

Unlike training - where exploration via a stochastic policy is essential for learning - we switch the trained stochastic DNN-based policy to a deterministic transmission strategy during inference to improve computational and energy efficiency, reduce inference latency, and enhance interpretability. From Theorem 1 in Section III, the optimal solution of the shared message transmission problem (6) is either a deterministic strategy or a convex combination of deterministic strategies; hence, an optimal policy can be taken deterministic during inference. Specifically, we set the exploration noise to zero $s = 0$ that fixes the input of DNNs, and precompute network outputs $\{\pi_{l,n}^*(0)\}_{l,n}$ to obtain decision vectors of nodes. Then, we round these output probabilities to the nearest integers (0 or 1) to determine transmissions for all nodes and messages. This is a deterministic strategy, which remains unchanged over time and depends only on the distribution of active nodes over shared messages p, not instantaneous realizations of active node sets. Alg. 2 summarizes the above inference procedure. This deterministic inference enjoys the following benefits:

- *Computational and energy efficiency*. Since input remains fixed and outputs can be precomputed, the system avoids querying the neural networks repeatedly during inference, reducing unnecessary energy consumption.
- *Reduced inference latency*. Precomputed decisions allow for immediate access to the optimal action, significantly reducing the time required for MAC decisions.
- *Interpretability*. The deterministic decisions provide a clear and unambiguous rule for communication, facilitating easier analysis of the MAC layer.

While the training procedure requires a stochastic policy to explore and learn effective strategies via policy gradients, the inference procedure leverages deterministic and precomputed decisions to provide efficient and interpretable MAC.

D. Online Learning Mechanism

While Theorem 3 guarantees that, under mild changes in the node activation distribution p, the trained transmission strategy $\{\pi_{l,n}^*\}_{l,n}$ is capable of maintaining satisfactory performance with bounded degradation $|\xi' - \xi''|$, practical scenarios may experience severe distribution shifts in p. In these cases, $\{\pi_{l,n}^*\}_{l,n}$ could suffer from significant performance degradation. To overcome this issue, we propose an online learning mechanism to tune the offline trained strategy and adapt it to the shifted activation statistics in a real-time manner. The online learning mechanism is based on a reward-driven policy gradient approach as well, where the objective remains to maximize the expected success rate $\mathbb{E}[\, \xi \mid \boldsymbol{\Psi}, p \,]$ over the transmission decisions taken by all active nodes. Similarly to the offline training procedure, it computes the instantaneous objective at each time step and leverages the latter to update neural networks' parameters with policy gradient, while requiring only feedback from the central controller for objective computation based on acknowledgments of successful transmissions.

Specifically, given the offline trained policies $\{\pi_{l,n}^*\}_{l,n}$ with parameters $\{\boldsymbol{\theta}_{l,n}^*\}_{l,n}$, we consider two online tuning strategies: (i) *full adaptation*, which updates all parameters of $\{\pi_{l,n}^*\}_{l,n}$, and (ii) *partial adaptation*, which updates only a subset of parameters while freezing the others. The former adapts more thoroughly to distribution shifts in time-varying environments and may yield better performance, while the latter reduces computational cost and speeds up the real-time online learning procedure, which is beneficial in resource-constrained settings.

For either full adaptation or partial adaptation, let $\{\pi_{l,n,t}^*\}_{l,n}$ be the transmission policies at each time step t during inference. The nodes deploy their policies $\{\pi_{l,n,t}^*\}_{l,n}$ with the exploration variable s to determine which opportunities are used and, therefore, the complete set of moves to transmit messages. Consequently, the remote server computes the transmission success rate ξ and distributes it to nodes as feedback. Based on this feedback, each node updates full or partial parameters of its own policy $\pi_{l,n,t}^*$ with gradient ascent locally, in an unsupervised manner through policy gradient.

E. Learning Complexity

The proposed DUSL directly outputs M probabilities - one per communication opportunity - thereby framing the problem as a structured prediction task where each binary decision determines whether to use a specific opportunity for transmission. However, the distributed MAB approach based on Thompson sampling presented in [14] requires each active node to choose from 2^M possible arms, each corresponding to a unique combination of the M opportunities. This exponential increase not only escalates the memory requirements but also imposes a severe burden on the learning process.

From a learning-theoretic perspective, the hypothesis class for selecting among 2^M actions entails a multi-class classification problem with an exponentially large label space because each unique combination of communication opportunities is treated as a distinct class label, which leads to a Vapnik–Chervonenkis (VC)-dimension [31] equal to 2^M. As the VC-dimension grows, the sample complexity - the number of training examples required to achieve a desired generalization

error - scales poorly. Under standard assumptions[4], the sample complexity for learning the distributed MAB using Thompson sampling approach scales as

$$\mathcal{O}\left(\frac{2^M + \log(1/\delta)}{\epsilon^2}\right),$$

where ϵ is the target generalization error, δ is the confidence level, and the exponential factor is due to VC-dimension given by the need to choose among 2^M actions. In contrast, by outputting M independent probabilities, the proposed DUSL effectively decomposes the task into M binary classification problems, each corresponding a binary decision. This factorization reduces the hypothesis class complexity, decreases its effective VC-dimension to M, and consequently lowers the sample complexity. This allows DUSL to reduce the sample complexity for learning to

$$\mathcal{O}\left(\frac{M + \log(1/\delta)}{\epsilon^2}\right),$$

thereby providing an exponential-to-linear improvement compared to distributed MAB. This improvement in sample efficiency translates into faster convergence and more reliable learning performance, particularly in scenarios where M is large.

V. Numerical Results

In this section, we present the simulation results for the proposed algorithm. As discussed in Sec. II, a general way to model the distribution p of active node sets $\mathcal{A}$ is by sampling its PMF from a Dirichlet distribution. However, it is challenging to define concentration parameters that capture correlated node activation patterns, which may be due to physical proximity of nodes, or to bound the minimum and maximum number of active nodes. Moreover, as the number of nodes and messages increases, generating the distribution p completely becomes impractical because it requires defining a PMF for $|\mathcal{R}_{\mathcal{A}}| = 2^K$ possible events [cf. (1)]. Therefore, we define two feasible methods to determine p that allow for the expression of possible pattern correlations and boundaries:

1) *Conditional activation*: In this setting, we characterize the activation of a node conditionally to the others. We consider L Bayesian networks (BNs) to determine the active node set $\mathcal{A}_l$ of message $l \in \mathcal{L}$ for $l = 1, \ldots, L$. Each BN is modeled by a directed acyclic graph comprising a set of nodes $\mathcal{V} = \{V_l^1, \ldots, V_l^{N_l}\}$ representing N_l variables. These variables are arranged in a linear sequence, where each has at most one incoming and one outgoing edge, forming a path from a source to a sink. Specifically, the variable V_l^i in the sequence determines the i-th active node in $\mathcal{A}_l$ and the conditional dependencies between node activations are modeled by conditional probability tables. We get N_l active nodes transmitting message l by sampling from the joint probability function defined by the BN, which is given by the chain rule of probability as

$$P(V_l^1, V_l^2, \ldots, V_l^{N_l}) = \prod_{i=1}^{N_l} P(V_l^i \mid V_l^{i-1}, ..., V_l^1). \quad (18)$$

2) *General activation*: In this setting, each node in the active node set $\mathcal{A}_l$ of each message l is selected independently from the others. We consider L multinomial distributions to determine $\mathcal{A}_l$ for $l = 1, \ldots, L$. Specifically, we sample $\mathcal{A}_l$ through N_l independent draws:

$$\mathcal{A}_l \sim \text{Multinomial}(N_l, d_l^1, \ldots, d_l^N), \quad (19)$$

where d_l^i is the probability that node i is selected in the active set $\mathcal{A}_l$. Consequently, we can bound the cardinality of the active set transmitting l as $|\mathcal{A}_l| \leq N_l$.

For both cases, we evaluate the proposed algorithm for both static and dynamic scenarios. The former considers a fixed distribution of activation patterns p over time, while the latter includes time-varying shifts in p. Additional experiments investigating unbalanced message availability, temporal correlation, and learned policy statistics are provided in Appendix B.

A. Implementation Details and Baselines

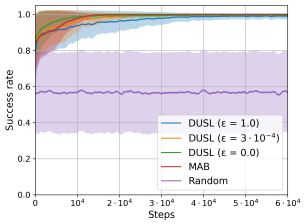
Each DNN used for testing DUSL consists of three layers with layer dimensions [256, 128, M]. The model is trained using the Adam optimizer with a learning rate $\lambda = 10^{-3}$, and employs ReLU activations. We also investigate a variant of our DUSL algorithm, where each node deploys a single DNN that generates $L\,M$ output neurons to simultaneously determine its transmission choices for all messages. For baselines, the most relevant one is the distributed MAB approach leveraging Thompson sampling proposed in [14], where each active node chooses which opportunities to use for message transmission in a distributed manner. In this approach, the action space consists of all possible 2^M subsets of communication opportunities, and each node learns a policy over this combinatorial space through bandit feedback, resulting in a higher learning complexity as discussed in Sec. IV-E. Another baseline to compare against our method is a random strategy, where each node selects a random subset of opportunities for transmission of each message $l \in \mathcal{L}$ at each time step, with each opportunity being independently chosen with probability $\frac{1}{|\mathcal{A}|}$. While this naïve approach does not leverage any learning or coordination, it serves as a useful lower-bound reference. Across multiple runs, fixed random seeds are used to ensure reproducibility.

B. Conditional Activation Patterns

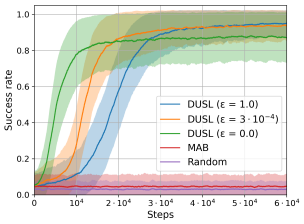
In this section, we first evaluate our method in static scenarios for different numbers of nodes N, channels M, and messages L under conditional activation patterns, and then consider dynamic scenarios in which the distribution of active node sets p changes over time, resulting in different conditional activation patterns. In each of the following cases, we average performance curves over 15 different independent runs to ensure the robustness of the methodology.

Scenario definition. To construct the BN for each message $l \in \mathcal{L}$, we employ a hierarchical sampling process to define the structure of the active node set $\mathcal{A}_l$. We start by randomly selecting four root nodes, each of which can activate up to four child nodes. This branching process continues recursively, forming a tree-like structure with directed edges representing conditional dependencies. To maintain computational tractability and avoid an excessively large set of active nodes, we introduce constraints on the tree's expansion at deeper levels. Specifically: (i) Up to level 4, each node can activate four child nodes; (ii) From level 5 to level 8, each node activates at most three child nodes; (iii) Beyond level 8, the branching is limited to two child nodes per node. This controlled growth ensures that the resulting BN remains compact and the sampled

[4]These sample complexity estimates hold under standard learning theory assumptions [32]: (i) training samples are drawn i.i.d. from a fixed distribution, (ii) the loss function is bounded, (iii) the hypothesis class has finite complexity (e.g., VC-dimension), and (iv) the output structure is fixed, either as a single 2^M-way output or M binary outputs.

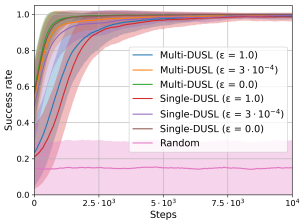


(a) $N = 16$, $M = 4$, $|\mathcal{A}_l| = 4$.

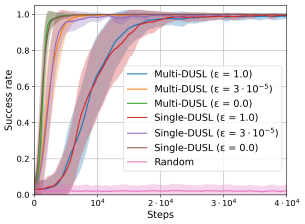


(b) $N = 64$, $M = 16$, $|\mathcal{A}_l| = 12$.

Fig. 4: Training in static scenarios with conditional activation patterns and $L = 1$.



(a) $N = 64$, $M = 32$, $|\mathcal{A}_l| = 4$, $L = 2$.



(b) $N = 64$, $M = 64$, $|\mathcal{A}_l| = 2$, $L = 4$.

Fig. 5: Training in static scenarios with conditional activation patterns and multiple messages.

active sets $\mathcal{A}_l$ are of manageable size while preserving a rich structure of conditional dependencies.

Static scenarios. Fig. 4 presents the performance of our algorithm for a single message, i.e., $L = 1$, under two different setups. The first one in Fig. 4a comprises a small scenario with $N = 16$ and $M = 4$, where the size of active node set is $|\mathcal{A}_1| = 4$. We see that the proposed DUSL and the baseline MAB achieve similar performance, as both of them prove to be effective in this simple scenario. However, as shown in Fig. 4b, if we increase the number of nodes to 64, the MAB does not scale and cannot solve the optimization problem. Conversely, the proposed DUSL exhibits good performance, being able to converge in these more complicated setting too. In Fig. 4b, it is possible to evaluate the impact of the exploration decay factor ε: if $\varepsilon = 1$, i.e., almost no exploration noise is fed to network, the learning process is slower, but in the end, achieves better performance than the case with $\varepsilon = 0$. In the latter case, the input fed to the neural network is always noisy, facilitating exploration at early training phases, but also preventing to converge to better values at the end of the training. A good trade off is to set a decaying ε, such that the training is faster at the beginning and the final convergence is not compromised.

In Fig. 5, we extend the scenario to multiple messages ($L = 2$ and $L = 4$). The configurations of the first case in Fig. 5a include $N = 64$ nodes and $M = 32$ opportunities with $|A_l| = 4$ active nodes per message, and that of the second case in Fig. 5 include $N = 64$ nodes and $M = 64$ opportunities with $|A_l| = 2$. With multiple messages, the complexity of the problem increases as the nodes must coordinate to successfully send all the messages. In these large-scale settings, we omit to present the distributed MAB approach as it is already proved to be ineffective for $L = 1$ and $N = 64$, and it becomes more computational and memory inefficient for greater M. For instance, with $M = 32$, each node would need to consider 2^{32} possible arms, requiring at least 2^{32} steps to sample each arm even once, and maintaining at least $2 \cdot 2^{32}$ counters to track successes and failures, which is clearly impractical in terms of both convergence time and memory requirements.

In this scenario, we also investigate a variant of our DUSL architecture in which each node deploys a single DNN with LM outputs determining the joint assignment of all L messages to the M transmission resources at that node. In Fig. 5, we explicitly compare the two designs: *Single-DUSL* refers to the single-DNN-per-node architecture, while *Multi-DUSL* denotes the original design with L independent DNNs per node (one per message). Empirically, Single-DUSL delivers comparable final success rates but consistently exhibits slower convergence than Multi-DUSL. We attribute this behaviour to the stronger coupling induced by joint assignment: a single network must learn to discriminate among $L\,M$ decision targets and its neurons jointly contribute to all messages, which increases the learning complexity compared to having one network dedicated to each message. Although the single-DNN design reduces the number of distinct parameter sets, in our large-scale settings the reduced per-message expressivity and the more challenging optimization outweighed those advantages. For the remainder of the experiments we focus on Multi-DUSL and, for brevity, simply refer to it as *DUSL*.

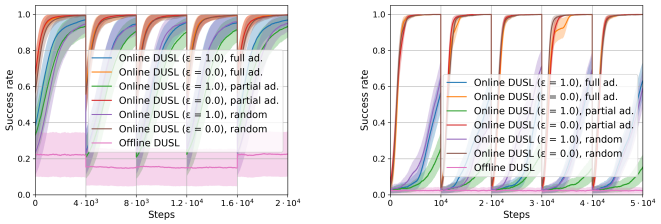
Dynamic scenarios. In Fig. 6, we examine adaptation capabilities of the proposed online learning mechanism in dynamic scenarios, where nodes are required to quickly adjust their transmission strategies according to variations in the activation distribution p. We set N, M, and L as that in Fig. 5, and change the environment periodically, i.e., allow the model to adjust within a very limited amount of time after the environment changes. We test the following four variants: i) DUSL with full adaptation, where all parameters of the offline trained model are updated online; ii) DUSL with partial adaptation, where the first hidden layer of the offline trained model is frozen and the rest parameters are updated online; iii) DUSL without offline training, where all parameters are randomly initialized and updated online; iv) Offline trained DUSL without online learning. For each of the first three cases including online learning, we consider two learning schemes with different exploration variables $\varepsilon = 1$ and $\varepsilon = 0$.

We see that the proposed online learning mechanism adapts the offline trained policy quickly to distribution shifts and maintains the performance of DUSL in dynamic scenarios. Online DUSL initialized at the offline trained model achieves the best performance, compared to the other variants, where partial adaption reduces the learning complexity and improves the adaption rate because it updates only a subset of model parameters online. We remark that the changes in p are harsh and each change is highly uncorrelated to the other in this scenario [cf. (18)], as the sampling form the BN is uniform and there is a limited number of possible active sets. This highlights the robustness of the online learning mechanism in challenging dynamic scenarios, which leads to significant performance improvements for each change of p.

C. General Activation Patterns

In this section, we evaluate the proposed method in general activation patterns, where there are no significant dependencies between the nodes' activations. We consider both static and dynamic scenarios, and average performance over 15 different independent runs to ensure robustness.

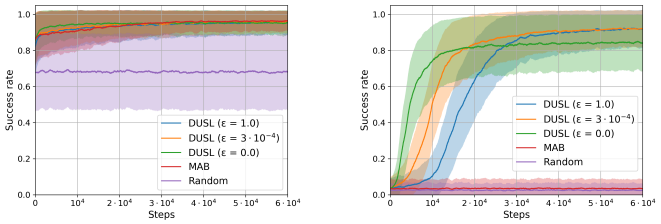
Scenario definition. As discussed, we use L multinomial distributions to model the activation patterns for messages $\mathcal{L}$. For each l, the probabilities of the multinomial distribution



(a) $N = 64$, $M = 32$, $|\mathcal{A}_l| = 4$, $L = 2$.

(b) $N = 64$, $M = 64$, $|\mathcal{A}_l| = 2$, $L = 4$.

Fig. 6: Adaptation to dynamic scenarios with conditional activation patterns and multiple messages, where 5 changes of p_A occur.



(a) $N = 16$, $M = 4$, $|\mathcal{A}_l| = 4$.

(b) $N = 64$, $M = 16$, $|\mathcal{A}_l| = 16$.

Fig. 7: Training in static scenarios with general activation patterns and $L = 1$.

$(d_l^1, \ldots, d_l^N)$ are sampled from a Dirichlet distribution:

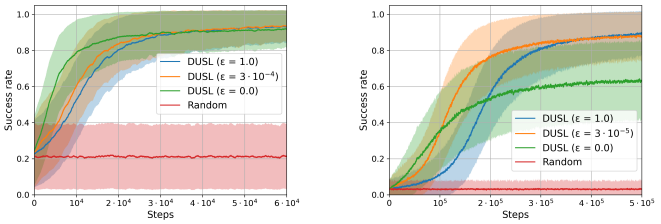
$$(d_l^1, \ldots, d_l^N) \sim \text{Dir}(\mathbf{1}_N) \ , \tag{20}$$

where $\mathbf{1}_N$ is a vector of ones of length N. This is a common case for a Dirichlet distribution, where all of the concentration parameters have the same value, resulting in no prior knowledge favoring one node activation over another. Particularly, when all concentration parameters are equal to one, the Dirichlet distribution is equivalent to a uniform distribution over the open standard $(K-1)$-simplex, i.e., it is uniform over all points in its support.

Static scenarios. In Fig. 7, we show the performance of our algorithm in general activation setting for $L = 1$ for static scenarios. The simpler scenario shown in Fig. 7a is easily solved by both DUSL and the distributed MAB baseline. However, in the more complex scenario shown in Fig. 7b where we increase the number of nodes from 16 to 64, DUSL is the only successful approach to reach convergence. The exploration decay factor ε has a notable impact; with reduced exploration, i.e., $\varepsilon = 1$, the training is slower but in the end achieves better performance than the case in which $\varepsilon = 0$.

In Fig. 8, we extend the analysis to multiple messages, considering $L = 2$ and $L = 4$. The first scenario in Fig. 8a features $N = 64$, $M = 32$, and $|A_l| = 4$ active nodes per message, while the second in Fig. 8b presents $N = 64$, $M = 64$, and $|A_l| = 2$. Unlike the conditional case, the nodes must coordinate their transmissions without any structured dependencies to rely on. Consequently, the general activation case is more difficult, and convergence is slower. In these cases, we omit results for the distributed MAB approach due to similar reasons as those discussed in Sec. V-B, because MAB already proves to be ineffective for $L = 1$ and $N = 64$, and computationally and memory inefficient for greater M too.

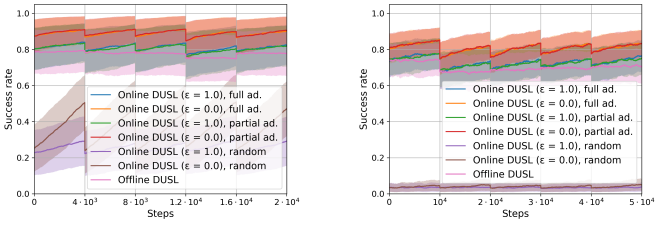
Dynamic scenarios. Finally, Fig. 9 shows the performance of DUSL in dynamic environments for general activation



(a) $N = 64$, $M = 32$, $|\mathcal{A}_l| = 4$, $L = 2$.

(b) $N = 64$, $M = 64$, $|\mathcal{A}_l| = 2$, $L = 4$.

Fig. 8: Training in static scenarios with general activation patterns and multiple messages.



(a) $N = 64$, $M = 32$, $|\mathcal{A}_l| = 4$, $L = 2$.

(b) $N = 64$, $M = 64$, $|\mathcal{A}_l| = 2$, $L = 4$.

Fig. 9: Adaptation to dynamic scenarios with general activation patterns and multiple messages, where 5 changes of p_A occur.

patterns using the proposed online learning mechanism. We set N and M as in Fig. 8, and evaluate whether the online learning mechanism adapts quickly to shifts in the probability distribution p. In these tests, the online learning mechanism proves to be effective to transfer knowledge. The online DUSL initialized from offline trained models, either with full adaptation or partial adaptation, exhibits the best adaption performance. The online DUSL initialized from scratch requires significantly more time to converge, whereas the offline DUSL without online learning mechanism performs worst with much lower success rates. In addition, we note from these results that fine-tuning through partial adaptation is generally the most computationally efficient solution, as it requires less operations to be performed online due to partial layer freezing in DNNs and achieves comparable performance to full adaptation.

VI. Conclusion

In this work, we introduced a novel decentralized MAC protocol learning framework considering the transmission of multiple shared messages under limited transmission opportunities and dynamic node activation patterns. We proposed DUSL, a deep learning-based, unsupervised framework, to address the inherent coordination challenge due to the lack of inter-node communication. We provided theoretical insights into the problem of decentralized coordination, proving the optimality of deterministic strategies. We also derived bounds on the degradation of success probability due to changes in node activation probabilities, which is crucial for understanding system robustness in dynamic environments. Through extensive simulations, we demonstrated the scalability and adaptability of our proposed framework, and its superiority to baselines like distributed MAB. These results validate our approach as an efficient solution for learning low-complexity, scalable, and reliable MAC protocols, particularly for URLLC scenarios and energy-constrained IoT applications.

APPENDIX A
PROOF OF THEOREMS 1, 2, AND 3

Proof of Theorem 1. The optimization problem in (6) is standard linear programming problem

$$\max_{\boldsymbol{\Psi}} \ \mathbb{E}[\,\xi \mid \boldsymbol{\Psi}\,] = \max_{\boldsymbol{\psi}} \ \boldsymbol{\omega}^T \boldsymbol{\psi} \ , \tag{21}$$

subject to simplex constraints $\boldsymbol{\psi} \geq \mathbf{0}$ and $\mathbf{1}^T \boldsymbol{\psi} = 1$, since

$$\begin{aligned} \mathbb{E}[\,\xi \mid \boldsymbol{\Psi}\,] &= \sum_{\mathbf{X} \in \{0,1\}^{L \times N \times M}} \mathbb{E}[\,\xi \mid \boldsymbol{\Psi} = \mathbf{X}\,] \\ &= \sum_{\mathbf{X} \in \{0,1\}^{L \times N \times M}} \psi(\mathbf{X}) \sum_{\mathcal{A} \in \mathcal{R}_A} p_A(\mathcal{A}) \ \xi(\mathcal{A}, \mathbf{X}) \\ &= \sum_{\mathbf{X} \in \{0,1\}^{L \times N \times M}} \psi(\mathbf{X}) \ \omega(\mathbf{X}) = \boldsymbol{\omega}^T \boldsymbol{\psi} \ , \end{aligned} \tag{22}$$

where $\omega(\mathbf{X}) = \sum_{\mathcal{A} \in \mathcal{R}_A} p_A(\mathcal{A}) \ \xi(\mathcal{A}, \mathbf{X}) \in [0, 1]$ is the probability of successful transmission given $\mathbf{X}$. The Lagrangian of (21) is

$$\begin{aligned} L(\boldsymbol{\psi}, \boldsymbol{\lambda}, \nu) &= \boldsymbol{\omega}^T \boldsymbol{\psi} - \boldsymbol{\lambda}^T \boldsymbol{\psi} + \nu(\mathbf{1}^T \boldsymbol{\psi} - 1) \\ &= -\nu + (\boldsymbol{\omega} - \boldsymbol{\lambda} + \nu \mathbf{1})^T \boldsymbol{\psi} \ , \end{aligned} \tag{23}$$

where $\boldsymbol{\lambda}$ and ν are the Lagrangian multipliers, the former for the 2^{KM} inequality constraints, the latter for the equality constraint. Maximizing with respect to $\boldsymbol{\psi}$ gives the dual function

$$g(\boldsymbol{\lambda}, \nu) = \sup_{\boldsymbol{\psi}} L(\boldsymbol{\psi}, \boldsymbol{\lambda}, \nu) = -\nu \ , \tag{24}$$

where $\sup_{\boldsymbol{\psi}}$ is the supremum of $\boldsymbol{\psi}$ and $\boldsymbol{\omega} - \boldsymbol{\lambda} + \nu \mathbf{1} = \mathbf{0}$ is the dual feasibility condition, otherwise $g(\boldsymbol{\lambda}, \nu)$ is not bounded from above as L is a linear function of $\boldsymbol{\psi}$. By considering the dual feasibility condition, the dual problem is therefore

$$\min_{\boldsymbol{\lambda}, \nu} g(\boldsymbol{\lambda}, \nu) = -\nu \ , \ \text{s.t.} \ \boldsymbol{\lambda} = \boldsymbol{\omega} + \nu \mathbf{1} \leq \mathbf{0} \ . \tag{25}$$

Here, from the dual constraint we get $-\nu \mathbf{1} \geq \boldsymbol{\omega}$, that is a vector inequality, equivalent to $\forall\, \mathbf{X} \in \{0,1\}^{L \times N \times M}, -\nu \geq \omega(\mathbf{X})$, and consequently $-\nu \geq \max \omega(\mathbf{X})$, so the dual solution is found at $\max \omega(\mathbf{X})$. We note that since the primal problem (21) is a linear optimization problem, strong duality holds, therefore the primal and the dual optimal objectives are equal:

$$\max_{\boldsymbol{\Psi}} \mathbb{E}[\,\xi \mid \boldsymbol{\Psi}\,] = \max_{\boldsymbol{\psi}} \boldsymbol{\omega}^T \boldsymbol{\psi} = \min_{\boldsymbol{\lambda}, \nu} g(\boldsymbol{\lambda}, \nu) = \max_{\mathbf{X}} \omega(\mathbf{X}) \ . \tag{26}$$

If $\mathbf{X}^* = \operatorname{argmax} \omega(\mathbf{X})$ is unique, then

$$\max_{\boldsymbol{\Psi}} \ \mathbb{E}[\,\xi \mid \boldsymbol{\Psi}\,] = \omega(\mathbf{X}^*) \ \ \text{and} \ \ \psi(\mathbf{X}^*) = 1 \ , \tag{27}$$

i.e., $\boldsymbol{\Psi}$ is the only optimal joint strategy and it is deterministic. On the other hand, if there is a set $\{\mathbf{X}_i^*\}_{i=1}^D$ such that $\mathbf{X}_i^* = \operatorname{argmax} \omega(\mathbf{X}) \ \forall i \in \{1, \ldots, D\}$, then

$$\max_{\boldsymbol{\Psi}} \ \mathbb{E}[\,\xi \mid \boldsymbol{\Psi}\,] = \omega(\mathbf{X}_i^*) \ \ \text{and} \ \ \sum_{i=1}^{D} \psi(\mathbf{X}_i^*) = 1 \ , \tag{28}$$

i.e., there exists a finite set $\{\boldsymbol{\Psi}_i'\}_{i=1}^D$ of optimal deterministic joint strategies, where $\psi_i'(\mathbf{X}_i^*) = 1$ and $\psi_i'(\mathbf{X}_k^*) = 0 \ \forall i, k \in \{1, \ldots, D\}, i \neq k$, and an infinite set $\{\boldsymbol{\Psi}_j''\}_{j=1}^{\infty}$ of optimal randomized joint strategies, where $\psi_j''(\mathbf{X}_k^*) < 1 \ \forall j, k \in \{1, \ldots, D\}$.

□

Proof of Theorem 2. The proof is straightforward since Theorem 1 allows to consider deterministic joint strategies only to solve (6) without loss of optimality. When considering only deterministic joint strategies, the problem in (6), framed as standard linear programming problem in (21), can be reduced to an instance of a NP-hard problem, i.e., 0-1 knapsack problem [33], that is

$$\begin{aligned} &\max_{\mathbf{x}} \sum_{i=1}^{n} c_i \ x_i \ , \\ &\text{s.t.} \ \sum_{i=1}^{n} w_i \ x_i \leq C \ , \ x_i \in \{0, 1\} \ \forall \ i \in \{1, \ldots, n\} \ , \end{aligned} \tag{29}$$

where $\mathbf{x} = \boldsymbol{\psi}$, $\mathbf{c} = \boldsymbol{\omega}$, $\mathbf{w} = \mathbf{1}$, and $C = 1$. To conclude the proof, we note that in our setup the inequality constraint in (29) is equivalent to an equality constraint, as $\mathbf{w} = \mathbf{1}$ and exactly one $x_i = 1, \ i \in \{1, \ldots, n\}$, is needed to solve the maximization problem. In fact, we can have at most one $x_i = 1$ to do not violate the inequality constraint, and we need at least one $x_i = 1$, as if $x_i = 0 \ \ \forall \ i \in \{1, \ldots, n\}$ then $\sum_{i=1}^n c_i \ x_i = 0$, which is indeed the global minimum.

□

Proof of Theorem 3. Defining by $\mathbf{X}$ the moves of all the nodes according to $\boldsymbol{\Psi}$, we can represent the set of values of A for which the condition for a successful transmission holds as

$$\mathcal{S} = \{\mathcal{A} \ : \ \xi(\mathcal{A}, \mathbf{X}) = 1, \ \forall \mathcal{A} \in \mathcal{R}_A\} \ , \tag{30}$$

and consequently we get the expected probability of successful transmission as $\xi' = \sum_{\mathcal{A} \in \mathcal{S}} p_A'(\mathcal{A})$ and $\xi'' = \sum_{\mathcal{A} \in \mathcal{S}} p_A''(\mathcal{A})$. Using the aggregation property of the Dirichlet distribution, we can model $(\xi', 1 - \xi')$ and $(\xi'', 1 - \xi'')$ as samples of a Dirichlet distribution of order two:

$$\begin{aligned} (\xi', 1 - \xi') \, , \, (\xi'', 1 - \xi'') &\sim \text{Dir} \left(\sum_{\mathcal{A} \in \mathcal{S}} \alpha_{\mathcal{A}} \, , \sum_{\mathcal{A} \in \mathcal{R}_A \setminus \mathcal{S}} \alpha_{\mathcal{A}} \right) \\ &= \text{Dir} \left(\alpha_S \, , \alpha_F \right) \ , \end{aligned} \tag{31}$$

where $\alpha_{\mathcal{A}}$ is the concentration parameter associated to both $p_A'(\mathcal{A})$ and $p_A''(\mathcal{A}) \ \forall \mathcal{A} \in \mathcal{R}_A$, and $\alpha_S = \sum_{\mathcal{A} \in \mathcal{S}} \alpha_{\mathcal{A}}$ and $\alpha_F = \sum_{\mathcal{A} \in \mathcal{R}_A \setminus \mathcal{S}} \alpha_{\mathcal{A}}$ are concentration parameters of a Dirichlet distribution of order two. Since the marginal distributions of a Dirichlet distribution are Beta distributions, we know that ξ' and ξ'' follow a Beta distribution, that is $\xi', \xi'' \sim \text{Beta} \left(\alpha_S \, , \alpha_F \right)$. We can now compute $\mathbb{E}[|\xi' - \xi''|]$ and a bound for $\text{Var}(|\xi' - \xi''|)$ which are helpful for the proof, where $\text{Var}(\cdot)$ is the variance. We can obtain the former as

$$\begin{aligned} \mathbb{E}[|\xi' - \xi''|] = \mathbb{E}[\xi' - \xi''] \ P\left(\xi' \geq \xi''\right) \\ + \ \mathbb{E}[\xi'' - \xi'] \ P\left(\xi' < \xi''\right) = 0 \ , \end{aligned} \tag{32}$$

since we can apply linearity of expectation and get

$$\mathbb{E}\left[\xi' - \xi''\right] = \mathbb{E}\left[\xi'' - \xi'\right] = \frac{\alpha_S}{\alpha_S + \alpha_F} - \frac{\alpha_S}{\alpha_S + \alpha_F} = 0 \ . \tag{33}$$

On the other hand, $\mathrm{Var}(|\xi' - \xi''|)$ can be written as

$$\begin{aligned}\mathrm{Var}(|\xi' - \xi''|) &= \mathbb{E}[|\xi' - \xi''|^2] - \mathbb{E}[|\xi' - \xi''|]^2 \\ &= \mathrm{Var}(\xi' - \xi'') + \overbrace{\mathbb{E}[\xi' - \xi'']^2}^{0} - \overbrace{\mathbb{E}[|\xi' - \xi''|]^2}^{0} \\ &= \mathrm{Var}(\xi') + \mathrm{Var}(\xi'') - \overbrace{2\ \mathrm{Cov}(\xi', \xi'')}^{0} \\ &= \frac{2\ \alpha_S\ \alpha_F}{(\alpha_S + \alpha_F)^2\ (1 + \alpha_S + \alpha_F)}\ ,\end{aligned} \tag{34}$$

where the expected values are zero because of (32, 33) and the covariance is zero given the i.i.d. assumption on p'_A and p''_A. By setting $\alpha = \alpha_S + \alpha_F$, we can bound $\mathrm{Var}(|\xi' - \xi''|)$ as

$$\begin{aligned}\mathrm{Var}(|\xi' - \xi''|) &= \frac{2\ \alpha_S\ \alpha_F}{\alpha^2\ (1 + \alpha)} \\ &< \frac{2\ \alpha^2}{\alpha^2\ (1 + \alpha)} = \frac{2}{1 + \alpha} \leq \frac{2}{1 + \alpha_\varepsilon\ 2^K}\ .\end{aligned} \tag{35}$$

In the first inequality, we leverage the fact that concentration parameters are non-negative, hence $\alpha > \alpha_S$ and $\alpha > \alpha_F$. On the other hand, the second inequality holds as $\alpha = \sum_{i=1}^{2^K} \alpha_i \geq \alpha_\varepsilon \cdot 2^K$, where $\alpha_\varepsilon = \min\{\alpha_i\}_{i=1}^{2^K}$. We can then obtain (12) by applying Chebyshev's inequality:

$$\begin{aligned}P(|\xi' - \xi''| \geq \eta) &= P\left(||\xi' - \xi''| - \mathbb{E}[|\xi' - \xi''|]| \geq \eta\right) \\ &\leq \frac{\mathrm{Var}(|\xi' - \xi''|)}{\eta^2} \\ &< \frac{2}{\eta^2\ (1 + \alpha_\varepsilon\ 2^K)}\ .\end{aligned} \tag{36}$$

Finally, since ξ' and ξ'', and consequently $|\xi' - \xi''|$, are bounded in $[0, 1]$, it suffices to consider $\eta \in (0, 1]$.

□

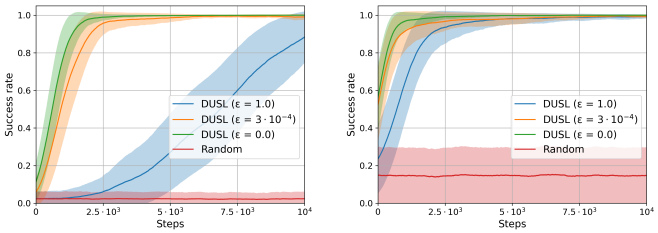
Appendix B
Additional Experiments

A. Unbalanced Message Availability

First, we investigate the performance of DUSL in a scenario with unbalanced message availability. We adopt the conditional activation pattern setting with $N = 64$ nodes, $M = 32$ communication opportunities, and $L = 2$ messages. The imbalance is introduced in the cardinality of the active sets: at each time step, the first message (l_1) is available to a set of $|\mathcal{A}_1| = 8$ nodes, while the second message (l_2) is available to only a single node, i.e., $|\mathcal{A}_2| = 1$. The results are presented in Fig. 10a. As shown, the DUSL framework successfully learns an effective transmission protocol, with all variants of DUSL significantly outperforming the random baseline. In particular, the high-exploration case ($\varepsilon = 0.0$) converges the fastest, while the low-exploration case ($\varepsilon = 1.0$) learns more slowly but still reaches a comparable performance level, indicating that DUSL is robust to asymmetries in message distribution.

B. Temporal Correlation in Activation Patterns

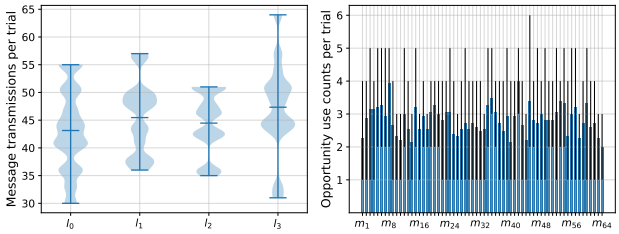
In the second experiment, we study the impact of temporal correlation in the activation patterns. We extend the conditional activation scenario by introducing a mechanism where the active sets can persist over multiple time steps. Specifically, for each message l at each time step t, after a new active set $\mathcal{A}_l(t)$ is sampled, we also sample an integer persistence duration $b_{t,l}$ uniformly from $\{0, \ldots, b_{\max}\}$. The active set $\mathcal{A}_l(t)$ then remains unchanged for the next $b_{t,l}$ time steps, i.e., $\mathcal{A}_l(t) = \mathcal{A}_l(t+1) = \cdots = \mathcal{A}_l(t + b_{t,l})$. For this experiment, we set $N = 64$, $M = 32$, $L = 2$, $|\mathcal{A}_l| = 4$ for both messages, and $b_{\max} = 10$. The results, shown in Fig. 10b, demonstrate that DUSL is robust to such temporal correlation as well. DUSL quickly adapts and converges to a high success rate, whereas the random policy's performance remains low. This suggests that the DUSL framework can effectively operate in environments where the underlying state exhibits temporal dependencies.



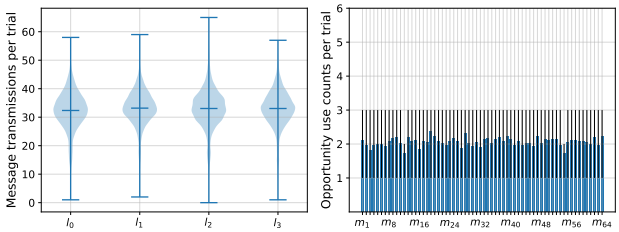
(a) $N = 64$, $M = 32$, $|\mathcal{A}_1| = 8$, $|\mathcal{A}_2| = 1$.

(b) $N = 64$, $M = 32$, $|\mathcal{A}_l| = 4$, $L = 2$, $b_{\max} = 10$.

Fig. 10: (a) Training in a static scenario with conditional activation patterns and unbalanced message availability, and (b) training in a static scenario with temporally correlated conditional activation patterns.



(a) Message transmissions and opportunity use counts per trial in conditional activation scenario.



(b) Message transmissions and opportunity use counts per trial in general activation scenario.

Fig. 11: Statistical analysis of learned policies for $N = 64, M = 64, L = 4, |\mathcal{A}_l| = 2$.

C. Learned Policy Statistics

To better understand the strategies developed by DUSL, we now investigate the statistics of the learned policies for two representative scenarios: the conditional activation case from Fig. 5b and the general activation case from Fig. 8b. Both scenarios use $N = 64, M = 64, L = 4$, and $|\mathcal{A}_l| = 2$. Figure 11 presents these statistics. The left plots of Fig. 11a and Fig. 11b show the distribution of the total number of message transmissions per trial for each of the $L = 4$ messages, presented as violin plots. In both scenarios, the policies leverage message repetitions to ensure delivery. This can be observed from the wide distributions and that the expected transmission counts significantly greater than the

number of active nodes ($|\mathcal{A}_l| = 2$). This demonstrates that the algorithm learns to build significant robustness into its strategy to maximize the success probability.

The right plots in Fig. 11 show the use counts of communication opportunities, displaying the mean usage (blue bar) and the 25-75 percentile range (black bars) for the $M = 64$ opportunities. A key finding is that for both learned policies, the 25th percentile for opportunity usage is in most cases 1. This is a critical outcome: it signifies that in at least 25% of the trials, these opportunities experience exactly one transmission. As a use count of 1 represents a successful, collision-free transmission, the fact that the learned policies consistently ensure this condition is met across a wide range of opportunities is a primary reason for their high success rates. Importantly, successful decoding does not require every opportunity to be collision-free, but only requires the L messages being delivered at least once. In other words, it suffices that there exist at least L out of the M opportunities that are used correctly (i.e., produce a collision-free transmission of the corresponding message), while the remaining $M - L$ slots may contain collisions without harming overall success, which explains why the policies can tolerate a certain number of collisions yet still achieve high success rates.

REFERENCES

[1] P. Z. Sotenga, K. Djouani, and A. M. Kurien, "Media access control in large-scale internet of things: A review," *IEEE Access*, vol. 8, pp. 55 834–55 859, 2020.

[2] J. Cheng, W. Chen, F. Tao, and C.-L. Lin, "Industrial IoT in 5G environment towards smart manufacturing," *Journal of Industrial Information Integration*, vol. 10, pp. 10–19, 2018.

[3] M. Vilgelm, H. Murat Gürsu, and W. Kellerer, "Random access protocols for industrial internet of things: Enablers, challenges, and research directions," *Wireless Networks and Industrial IoT: Applications, Challenges and Enablers*, pp. 55–76, 2021.

[4] X. Jian, Y. Liu, Y. Wei, X. Zeng, and X. Tan, "Random access delay distribution of multichannel slotted aloha with its applications for machine type communications," *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 21–28, 2016.

[5] M. Compare, P. Baraldi, and E. Zio, "Challenges to IoT-enabled predictive maintenance for industry 4.0," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4585–4597, 2020.

[6] Y. Liu, W. Yu, T. Dillon, W. Rahayu, and M. Li, "Empowering IoT predictive maintenance solutions with AI: A distributed system for manufacturing plant-wide monitoring," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 2, pp. 1345–1354, 2022.

[7] N. Longhi, L. M. Amorosa, S. Cavallero, E. Buracchini, and R. Verdone, "5G architectures enabling remaining useful life estimation for industrial iot: A holistic study," *IEEE Open Journal of the Communications Society*, vol. 6, pp. 1016–1029, 2025.

[8] A. Munari, "Modern random access: An age of information perspective on irregular repetition slotted aloha," *IEEE Transactions on Communications*, vol. 69, no. 6, pp. 3572–3585, 2021.

[9] L. Yang and A. Shami, "A lightweight concept drift detection and adaptation framework for IoT data streams," *IEEE Internet of Things Magazine*, vol. 4, no. 2, pp. 96–101, 2021.

[10] ——, "IoT data analytics in dynamic environments: From an automated machine learning perspective," *Engineering Applications of Artificial Intelligence*, vol. 116, p. 105366, 2022.

[11] M. A. Siddiqi, H. Yu, and J. Joung, "5G ultra-reliable low-latency communication implementation challenges and operational issues with IoT devices," *Electronics*, 2019.

[12] K.-H. Ngo, G. Durisi, A. Graell i Amat, P. Popovski, A. E. Kalør, and B. Soret, "Unsourced multiple access with common alarm messages: Network slicing for massive and critical IoT," *IEEE Transactions on Communications*, vol. 72, no. 2, pp. 907–923, 2024.

[13] E. Becirovic, E. Björnson, and E. G. Larsson, "Activity detection in distributed massive MIMO with pilot-hopping and activity correlation," *IEEE Wireless Communications Letters*, vol. 12, no. 2, pp. 272–276, 2023.

[14] S. u. Haque, S. Chandak, F. Chiariotti, D. Günduz, and P. Popovski, "Learning to speak on behalf of a group: Medium access control for sending a shared message," *IEEE Communications Letters*, vol. 26, no. 8, pp. 1843–1847, 2022.

[15] A. E. Kalor, O. A. Hanna, and P. Popovski, "Random access schemes in wireless systems with correlated user activity," in *2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, 2018, pp. 1–5.

[16] S. Ali, A. Ferdowsi, W. Saad, N. Rajatheva, and J. Haapola, "Sleeping multi-armed bandit learning for fast uplink grant allocation in machine type communications," *IEEE Transactions on Communications*, vol. 68, pp. 5072–5086, 2018.

[17] C. Zheng, M. Egan, L. Clavier, A. E. Kalør, and P. Popovski, "Stochastic resource optimization of random access for transmitters with correlated activation," *IEEE Communications Letters*, vol. 25, no. 9, pp. 3055–3059, 2021.

[18] M. A. Jadoon, A. Pastore, M. Navarro, and A. Valcarce, "Learning random access schemes for massive machine-type communication with MARL," *IEEE Transactions on Machine Learning in Communications and Networking*, vol. 2, pp. 95–109, 2024.

[19] A. Rech and S. Tomasin, "Coordinated random access for industrial IoT with correlated traffic by reinforcement-learning," in *2021 IEEE Globecom Workshops (GC Wkshps)*, 2021, pp. 1–6.

[20] Z. Gao, M. Eisen, and A. Ribeiro, "Optimal WDM power allocation via deep learning for radio on free space optics systems," in *IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.

[21] A. Jeannerot, M. Egan, and J.-M. Gorce, "Exploiting device heterogeneity in grant-free random access: A data-driven approach," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 9, pp. 13 440–13 450, 2024.

[22] K. Stern, A. E. Kalør, B. Soret, and P. Popovski, "Massive random access with common alarm messages," in *2019 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2019, pp. 1–5.

[23] L. M. Amorosa, L. Spampinato, C. Buratti, and R. Verdone, "Goal-oriented uplink scheduling requests in wireless networks via graph neural networks," in *IEEE EUROCON - International Conference on Smart Technologies*, 2025, pp. 1–6.

[24] V. Sze, Y. Chen, T. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2295–2329, 2017.

[25] Z. Gao, M. Eisen, and A. Ribeiro, "Resource allocation via model-free deep learning in free space optical communications," *IEEE Transactions on Communications*, vol. 70, no. 2, pp. 920–934, 2022.

[26] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine Learning*, vol. 8, no. 3–4, p. 229–256, May 1992.

[27] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 12, 1999.

[28] Z. Gao, M. Eisen, and A. Ribeiro, "Resource allocation via graph neural networks in free space optical fronthaul networks," in *IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2020, pp. 1–6.

[29] Z. Gao, Y. Shao, D. Gündüz, and A. Prorok, "Decentralized channel management in wlans with graph neural networks," in *IEEE International Conference on Communications (ICC)*, 2023, pp. 3072–3077.

[30] L. M. Amorosa, M. Skocaj, R. Verdone, and D. Gündüz, "Multi-agent reinforcement learning for power control in wireless networks via adaptive graphs," in *IEEE International Conference on Communications (ICC)*, 2024, pp. 2968–2973.

[31] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth, "Learnability and the vapnik-chervonenkis dimension," *Journal of the ACM (JACM)*, vol. 36, no. 4, pp. 929–965, 1989.

[32] S. Shalev-Shwartz and S. Ben-David, *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[33] S. Martello and P. Toth, *Knapsack problems: algorithms and computer implementations*. USA: John Wiley & Sons, Inc., 1990.