Rate-Constrained Remote Contextual Bandits

Francesco Pase®, Graduate Student Member, IEEE, Deniz Gündüz®, Fellow, IEEE, and Michele Zorzi®, Fellow, IEEE

Abstract—We consider a rate-constrained contextual multi-armed bandit (RC-CMAB) problem, in which a group of agents are solving the same contextual multi-armed bandit (CMAB) problem. However, the contexts are observed by a remotely connected entity, i.e., the decision-maker, that updates the policy to maximize the returned rewards, and communicates the arms to be sampled by the agents to a controller over a rate-limited communications channel. This framework can be applied to personalized ad placement, whenever the content owner observes the website visitors, and hence has the context, but needs to transmit the ads to be shown to a controller that is in charge of placing the marketing content. Consequently, the rate-constrained CMAB (RC-CMAB) problem requires the study of lossy compression schemes for the policy to be employed whenever the constraint on the channel rate does not allow the uncompressed transmission of the decision-maker’s intentions. We characterize the fundamental information theoretic limits of this problem by letting the number of agents go to infinity, and study the regret that can be achieved, identifying the two distinct rate regions leading to linear and sub-linear regrets respectively. We then analyze the optimal compression scheme achievable in the limit with infinite agents, when using the forward and reverse KL divergence as distortion metric. Based on this, we also propose a practical coding scheme, and provide numerical results.

Index Terms—Contextual multi-armed bandits, policy compression, rate-distortion theory.

I. INTRODUCTION

The past decade has seen a transition from centralized computing solutions, based on the cloud, to more distributed systems, also known as multi-access edge computing [1], mainly in response to the emerging paradigm of Internet of everything, where data is generated, processed and consumed by a network of connected nodes with diverse storage and computing capabilities. Indeed, today massive amounts of data are generated by edge devices, e.g., sensors, cameras, vehicles and drones, and need to be quickly processed to satisfy the latency constraints of the new vertical services, e.g., Industrial Internet of Things (IIoT) systems, autonomous driving scenarios or remote surgery. Moving data from distributed and personal devices to the cloud introduces several issues related to privacy, network resources (i.e., offloading data collected by edge devices to central entities may congest the communication networks) and latency.

Another paradigm shift involves the adoption of machine learning (ML) as a core technology for future services. ML techniques, mainly driven by deep neural networks (NNs), achieve state-of-the-art performance in many practical applications such as computer vision, speech recognition, and automatic control [2]. However, these impressive performance results require extremely large models, which, in turn, need a lot of data to be trained. This demands the careful design of the distributed learning systems [3], [4], that are foreseen to produce and process most of the data in future communications networks. Consequently, there is an urgent need for new methods to represent and compress the information generated and processed by distributed learning solutions to efficiently make use of the available physical network resources. To this end, many recent works have tried to reduce the amount of data needed to properly train a distributed model at the edge. One potential solution is NN compression, which reduces the amount of information needed to store or transmit an NN.

The focus of this work is to study the information-theoretic limits of distributed learning in the specific context of multi-agent contextual multi-armed bandits (CMABs). CMABs model decision-making problems in which an agent interacts with an environment in sequential rounds. At each round, the agent observes a context, which contains side information about the environment, and has to pull one out of K arms. Based on the observed context and pulled arm, the environment returns a reward, which is sampled according to an unknown distribution. The goal of the agent is to optimize an arm selection strategy to maximize the average sum of obtained rewards. In this scenario, the contexts of N agents are available to a central decision-maker that has to inform a remote entity, called the controller, on the arms the agents should pull. Depending on the scenario, the controller either directly pulls the arms, or provides the indices of the suggested arms to the agents, that can physically interact with the environment. However, we assume that there is a rate-limited communication channel between the decision-maker and the controller, which limits the number of bits the decision-maker can use.
to convey the intended arms at each round. This framework can model standard recommendation systems, where a content owner observes some client-dependent features on one side, and must communicate the recommendations to a separate entity, which in turn proposes the recommended items to the clients. Another related engineering problem is that of distributed training of reinforcement learning (RL) policies. Particularly in deep RL, complex policies are usually trained exploiting parallel simulations involving many agents, which can simultaneously collect data to speed up the policy training convergence, running at the central server. Consequently, the communication link between the decision-maker and the controller, over which potentially high-dimensional data, i.e., the policy/observation/actions, must be conveyed, may represent the bottleneck of the system. Further analysis may also consider limited communication budget in the feedback link, over which the rewards are transmitted.

The purpose of this work is to formulate the novel rate-constrained CMAB (RC-CMAB) framework, and to study the theoretical limits, i.e., in the regime of infinite agents, of this communication problem when the available rate is below what would be required to perfectly convey the decision-maker’s policy at each iteration. Then, we provide practical lossy policy compression schemes, that take into account the learning objective when specifying the distortion metric to be used, and can be applied in the more practical scenario with a finite number of agents. We then draw connections between our framework and the information bottleneck [5], [6] and maximum entropy reinforcement learning [7], [8] approaches. Finally, we report numerical results in support of our analyses.

II. RELATED WORK

Future communication networks should be designed with the aim of efficiently supporting edge artificial intelligence (AI) applications [3]. In particular, the data exchanged in such systems is correlated with some training process, and so new data compression and representation methods should be considered. The limits of lossy compression have been defined by rate-distortion theory [9], which provides the minimum number of bits needed to represent information given a target value for the maximum tolerable distortion between the original data, and the data reconstructed from the compressed version. However, in distributed ML, the goal at the receiver is not to perfectly reconstruct the input data, but to perform some inference or training task based on it. In the past literature, these problems have been formulated as distributed hypothesis testing [10], [11] and estimation [12], [13] with rate-constrained communication links.

In particular, this work focuses on multi-agent reinforcement learning (MARL) [14], where the goal is not to learn a function through supervised learning, but rather to optimize a policy, i.e., a map that, given an observation, provides a probability distribution over a set of feasible actions. Whenever agents take actions, they receive stochastic rewards distributed according to some unknown distributions. In general, the goal of the agents is to learn a policy that maximizes the returned rewards. In our case, the processes of collecting observations, training the policy and taking the actions are implemented by physically distributed entities [14], which require the design of compression and communication strategies to coordinate the training process. Another application of MARL is the parallel training of a single logical agent, with the aim of accelerating the training process [15], [16], [17].

To this end, an important line of research is the study of the most valuable data to be shared among the agents in order to achieve cooperation, and thus convergence to better policies. Frameworks in [18], [19], [20], [21] admit cooperation among the agents through the transmission of signals, whose effects on the common task are differentiable with respect to the signal transmission decisions. In this case, the emergence of languages and cooperative messages is analyzed, with perfect communications links. The authors in [22] consider noisy communications, and propose a scheme to jointly learn good policies and optimal ways to code and transmit the actions over the communications channels.

Other research efforts consider deep RL agents, and focus on compressions schemes of NNs, which are generally used to approximate value functions and target policies. For example, the authors in [23] adopt a knowledge distillation technique to reduce the complexity of a behavioral policy, while training a more complex target policy that serves as a teacher, in the context of parallel training [17]. In [24] and [25], the authors achieve model compression adopting pruning techniques [26], [27], [28] to reduce the size of an agent’s NN-based policy. Both papers provide empirical studies of the performance of a single agent when trained with different pruning levels. In [24], an iterative process is used to prune the model, which is reduced after training the original (full size) NN. In our framework, training is performed over rate-limited communications channels, and there could be no opportunity to transmit the whole policy. The authors in [25] consider performing pruning exploiting an initial offline dataset, that is not present in our case. Moreover the pruning level, and so the needed rate to transmit the models, has to be fixed at the beginning of the training process.

As mentioned above, the focus of our analyses is on a multi-agent version of CMAB, in which a decision-maker observes the contexts for a set of distributed agents, which can interact with the decision-maker through a remotely connected controller. The framework is similar to that of [29], where a server communicates actions to a pool of agents. However, in this case the analyzed link is the one between the agents and the server, and is used to send back the observed rewards. Moreover, no context (or state) is considered. In [30], the same multi-armed bandit (MAB) problem is solved cooperatively by a set of agents that can share some information about the best estimated actions. In this case communication is peer-to-peer with a deterministic rate, and again there is no contextual information. A related formulation has been proposed in [31], where the Batched Thompson Sampling algorithm is introduced. In this case the goal is to reduce the number of policy updates, thus reducing the computational complexity of the algorithm. Rounds are grouped into batches, and within one
batch arms are pulled without updating the sampling policy. Similarly, in our formulation one round can be considered as a batch, given that the \( N \) agents operate in parallel. However, in [31], the \textit{batch size}, i.e., the number of samples without policy updates, can be optimized by the algorithm and is not fixed during the training, i.e., different batches may have different sizes. On the contrary, in our case the batch size is fixed to \( N \), and is given by the environment. Moreover, in [31] the authors do not consider the contextual case, and there are no communications constraints when pulling arms.

Interestingly, the authors of [32] study a similar framework to justify the concept of information bottleneck in RL [33], [34] from a psychological perspective. In this work, the rate-constrained channel is compared to resource-limited policy storing systems, e.g., the brain and noisy storage devices, commenting on the trade-off between policy complexity, i.e., level of correlation between states and actions, and policy performance, i.e., obtained reward. In these studies the single-agent case is considered, and the formulation is somewhat generic with qualitative and empirical considerations, without properly framing the underlying learning problem. In this older study [35], the authors performed a set of behavioral experiments that resemble the CMAB problem, recording correlations between performance variance, i.e., action stochasticity, and model capacity, i.e., work memory, in humans.

With this work, the aim is to study this relation within the proper information-theoretic setting, relating it to the learning task of CMAB, and highlighting the theoretical trade-off between channel rate and learning performance. Moreover, we study practical ways to compress policies having limited impact in the training process.

### III. PROBLEM FORMULATION

#### A. The Contextual Multi-Armed Bandit (CMAB) Problem

The standard single-agent CMAB problem considers an agent interacting with the environment by pulling arms upon the observation of some contextual information, and receiving a reward based on the observed context and pulled arm. Specifically, at each round \( t = 1, \ldots, T \), the environment samples a context \( s_t \in \mathcal{S} \) following distribution \( P_S \), where \( \mathcal{S} \) is a finite set containing all possible contexts. We assume that all contexts are observable, i.e., \( \forall s \in \mathcal{S} \). \( P_S(s) > 0 \). When observing \( s_t \), the agent chooses an arm \( a_t \in \mathcal{A} = \{1, \ldots, K\} \), with probability \( \pi_t(a_t|s_t) \). Given the pair \((s_t, a_t)\), the environment returns a stochastic reward \( R(s_t, a_t) \) sampled according to \( P_R(s_t, a_t) \), which is an unknown and stationary distribution that characterizes the reward statistics, and depends on the sampled context and the arm pulled by the agent. We then define \( \mu(s, a) = \mathbb{E}_{R}[R(s, a)] \), and further assume that \( R(s, a) \in [0,1] \), \( \forall s \in \mathcal{S} \) and \( \forall a \in \mathcal{A} \). Moreover, we assume that the reward distributions belong to the exponential family, as also detailed in [36], Assumption 1. The policy \( \pi_t(a_t|s_t) \) employed by the agent is a map \( \pi_t: \mathcal{H}^{t-1} \times \mathcal{S} \rightarrow \Delta_K \), where \( \Delta_K \) denotes the \((K-1)\)-simplex, containing all possible distributions over the set of \( K \) arms, and \( \mathcal{H}^{t-1} \) is the history, containing all possible observations, arms and rewards collected until round \( t-1 \), i.e., the element \( H(t-1) \in \mathcal{H}^{t-1} \) is \( H(t-1) = \{s_1, a_1, r_1, \ldots, s_{t-1}, a_{t-1}, r_{t-1}\} \). The goal for the agent is to optimize the policy \( \pi_t \) in order to maximize the average sum of received rewards or, equivalently, to minimize the Bayesian regret

\[
BR(\pi, T) = \mathbb{E} \left[ \sum_{t=1}^{T} \mu(s_t, a^*(s_t)) - \mu(s_t, A_t) \right],
\]

where \( A_t \) is the arm pulled by the agent in round \( t \) sampled according to \( \pi_t(a_t|s_t) \), and \( a^*(s_t) = \arg \max_{a \in \mathcal{A}} \mu(s_t, a) \) is the optimal arm in round \( t \), i.e., the one that maximizes the average reward in context \( s_t \). If \( a^*(s) \) is not unique, it represents an arbitrarily chosen arm among the optimal ones. Here the expectation is taken with respect to the state, action and problem instance distributions.

#### B. Rate-Constrained CMAB

We consider a system in which \( N \) agents have to solve the same realization of a CMAB problem, where each agent observes an independent context, distributed according to \( P_S \), and depicted in Fig. 1. The agents can only interact with the environment pulling arms, whereas the contexts are observed by a remote decision-maker. The decision-maker communicates with the agents through a controller, that is in charge of receiving instructions from the decision-maker, and informing the agents accordingly. However, the decision-maker can transmit information to the controller through a rate-constrained channel, which imposes a constraint on the number of bits per agent the decision-maker can transmit to the controller. Consequently, at each system round, the environment samples \( N \) contexts \( \{s_{j,n}\}_{n=1}^{N} \), i.e., one per agent, which are observed by the decision-maker, which, in turn, needs to decide and communicate to the controller the \( N \) arms \( \{a_{j,n}\}_{n=1}^{N} \) to be pulled by the agents.

Then, the decision-maker exploits the knowledge accumulated until system round \( j \), and encoded in the variable \( H(j-1) = \{\{s_{1,n}, a_{1,n}, r_{1,n}\}_{n=1}^{N}, \ldots, \{s_{j-1,n}, a_{j-1,n}, r_{j-1,n}\}_{n=1}^{N}\} \in \mathcal{H}^{(j-1)} \) to optimize its policy \( \pi_j \). However, in our setting, the decision-maker can interact with the controller only through a rate-constrained communication channel, which may not allow to transmit all the intended arms to the agents. Consequently, the problem is to communicate the arm distribution, i.e., the policy \( \pi_j(a|s) \), which depends on the specific context realizations observed in round \( j \), to the controller within the available communication resources while inducing the minimal impact on the performance of the learning algorithm. To this end, the decision-maker employs a function \( f_j^{(N)}: \mathcal{H}^{(j-1)} \times \mathcal{S}^{N} \rightarrow \)
mapping the knowledge acquired up to round $j - 1$, together with the agents’ contexts, to a message index to be transmitted over the channel. At the receiver, the controller adopts a function $g^{(N)}_j : \{1, 2, \ldots, B\} \rightarrow \mathcal{A}^N$ to decode from the received message the $N$ arms to be pulled by the agents. In general, both functions $f^{(N)}_j$ and $g^{(N)}_j$ can be stochastic. We then define the Bayesian system regret as

$$BR(J, \{f^{(N)}_j, g^{(N)}_j\})$$

$$= \mathbb{E}\left[\sum_{j=1}^{J} \sum_{n \in \mathcal{N}} r(s_{j,n}, a^*(s_{j,n}) - r(s_{j,n}, g_{j,n}(m_j))\right],$$

(2)

where $g_{j,n}(m_j)$ is the arm pulled by agent $n$ during round $j$ decoded from the message $m_j = f^{(N)}_j(H(j - 1), s^N_j)$, and $s^N_j \in S^N$ is the vector containing the contextual information in round $j$ for all the $N$ agents. The goal is to specify the encoding and decoding functions, $f^{(N)}_j$ and $g^{(N)}_j$, to minimize the Bayesian system regret in Eq. (2). More specifically, the goal is to obtain a regret which is sub-linear in $J$, possibly achieving the same performances obtained by standard CMAB solutions [46]. For a problem with $N$ agents, a rate $R$ is said to be achievable if there exist functions $\{f^{(N)}_j, g^{(N)}_j\}_{j=1}^J$ with rate $\frac{1}{N} \log_2 B \leq R$, and Bayesian system regret

$$\lim_{J \rightarrow \infty} \frac{BR(J, \{f^{(N)}_j, g^{(N)}_j\})}{J} = 0.$$  

Note that when there is no communication channel from the decision-maker to the controller, i.e., $B = 0$, sub-linear regret is not possible, since the agents cannot learn on their own without observing the contexts and rewards. On the other hand, when $B$ is sufficiently large, i.e., $B \geq N \log_2 K$, any desired arm sequence can be conveyed to the controller, and the problem becomes a distributed CMAB problem with $N$ parallel agents. Our goal is to identify the minimal communication needed from the decision-maker to the controller that makes sub-linear regret feasible. We would like to emphasize that the introduced RC-CMAB problem differs from the standard CMAB formulation in two aspects: First, at each round $j$, the decision-maker pulls $N$ parallel arms through the agents exploiting $\pi_j$, which is updated at the end of each round. This is similar to the batch MAB formulation presented in [31], in which the policy can be updated only every $N$ arm pulls. Second, given the available rate $R$, the decision-maker may not be able to convey the exact sequence of arms $[a_{j,n}]_{n=1}^N$ sampled from $\pi_j$, and instead must send a compressed version, which may result in some agents to pull sub-optimal arms. We highlight that this is a lossy compression problem; however, unlike classical lossy source coding problems, the goal is not to send a sequence of arms with highest average fidelity, but to enable the agents to pull the arms that would result in a sub-linear regret.

IV. THEORETICAL LIMITS

In this section, we provide a theoretical analysis of the regret bound achievable by the Thompson Sampling (TS) strategy, and the minimum rate required to achieve sub-linear regret.

A. TS Performance

We first provide the regret performance of the TS algorithm, when there is no constraint on the available rate to transmit the intended arms. In this work, we analyze the general case in which no prior structure is assumed between the optimal policies and the different contexts, and so we consider the simplest implementation of one MAB agent for each context $s \in \mathcal{S}$. The TS algorithm adopts a Bayesian strategy, estimating the distribution $p^{a,s}(\mu)$ of the reward mean $\mu(s, a) \in [0, 1]$ in each round $j$ with $p^{a,s}_j(\mu)$. When observing the context $s_j$, it samples $\tilde{\mu}_j(s_j, a) \sim p^{a,s}_j(\mu)$, $\forall a \in \mathcal{A}$, and pulls the arm $a_j = \arg \max_{a \in \mathcal{A}} \tilde{\mu}_j(s_j, a)$. In RC-CMAB the decision-maker adopts the described sampling strategy for each agent $n \in \{1, \ldots, N\}$. After receiving all the rewards $[r_{t,n}]_{n=1}^N$, the decision-maker updates its belief on $\mu(s, a)$ optimizing the posterior $p^{a,s}_j(\mu)$. The variance of the posterior distributions is exploited to perform exploration. This algorithm is well studied, and is known under the name of Thompson Sampling (TS) [37]. The TS algorithm implicitly induces a probability distribution $\pi_j(a|s)$ over the arms that can be computed as

$$\pi_j(a|s) = \int \prod_{k=1}^K p^{a,k}_j(\mu)d\mu,$$

where $p^{a,k}_j(\mu)$ is the cumulative distribution function (CDF) of $\mu(s, k)$, and the random variables $\mu(s, a)$ are independently distributed. We will call $\pi_j(a|s)$ the target policy, i.e., the one that the decision-maker would like to convey to the controller. However, in RC-CMAB the constraint on the rate of the communication channel may not allow to sample the arms according to $\pi_j(a|s)$, as explained in Section III. In this case, the problem is that the decision-maker updates the posteriors $p_j^{a,k}$, and so $\pi_j(a|s)$, using the TS algorithm, but can only sample with an approximate policy $Q_j(a|s)$, whenever the available rate is not sufficient to convey $\pi_j(a|s)$.

B. Regret Bounds

We now report the performance of the TS algorithm, when the available rate is sufficient to perfectly convey the arms from the decision-maker to the controller in each round $j = 1, \ldots, J$. To this end, we align the parallel interactions between the agents and the environment in time, and consider virtual rounds $t \in \{1, \ldots, J \cdot N\} = \mathcal{T}$, as if a single agent, i.e., the decision-maker, were playing a sequential CMAB game, with the constraint that the policy can be updated only every $N$ steps, i.e., at the end of each round. We observe that the order used to align the agent interactions does not affect the analysis, as the agents all play the same policy, and interact with independent and identically distributed (i.i.d) replicas of the same environment, leading the two formulations to be mathematically equivalent. Consequently, by providing the results as a function of the virtual rounds $t$, we consider the total number of interactions the agents have with the environment, which is consistent with the usual MAB notation.

Theorem 1 (TS Bayesian System Regret): The finite-time Bayesian system regret of TS is upper bounded by

$$BR(\pi^{TS}, T) \leq 2SKN + 4\sqrt{(2 + 6 \log T)SKNT},$$

(4)
and the asymptotic regret is
\[
BR(\pi^T, T) \in \mathcal{O}\left(\sqrt{KST \log T}\right). \tag{5}
\]

**Proof:** See Appendix-A.

We observe that, in the finite-time analysis, an additional term \(\sqrt{N}\) appears, with respect to the standard single-agent performance [46]. This is a consequence of the fact that, in one round \(j\), \(N\) arms are pulled in parallel, without updating the policy. This effect has highest impact during the first rounds, as the policy has not converged yet, and so sub-optimal arms are pulled in parallel. In the long term, the effect vanishes. This result is consistent with the analysis in [31], with the difference that what the authors called batch, in our scenario is the parallel execution of the \(N\) agents, and so in our case the batch size is fixed to \(N\) and cannot be optimized. Consequently, in the finite-time upper bound we obtain a factor \(\sqrt{N}\), that replaces the factor \(\sqrt{N}\) obtained in [31], where \(\alpha\) is the so-called batch growing factor [31]. The factor \(S\) is introduced as we consider the CMAB problem.

### C. Rate-Distortion Function for Communicating Policies

We now present the minimum rate needed to transmit a policy, i.e., the arms \(a_N^j = (a_{j,1}, \ldots, a_{j,N})\) to be sampled for each agent according to \(\pi_j(a)\), and conditioned on the observed context vector \(z_N^j = (s_{j,1}, \ldots, s_{j,N})\), when a specific distortion function is adopted to measure the discrepancy between the sequence \(z_N^j = ((s_{j,1}, a_{j,1}), \ldots, (s_{j,N}, a_{j,N}))\), and the sequence \(z_N^{\hat{j}} = ((s_{j,1}, \hat{a}_{j,1}), \ldots, (s_{j,N}, \hat{a}_{j,N}))\), where \(\hat{a}\) indicates the arms decoded by the controller based on the received message \(m_j\), as indicated in Section III-B. In short, \(\hat{a}_N^j\) is the vector containing the arms actually pulled by the agents. Given the underlying learning problem, the quality metric for the vector \(\hat{a}_N^j\) should not be based on a per-symbol distance, but rather on the sampling probability distributions. Indeed, to obtain sub-linear regret, what interests us is the probability of sampling specific sequences. Consequently, the distortion function \(d(\tilde{Q}_N, P_{SA})\) compares the empirical distribution \(\tilde{Q}_N\) with \(P_{SA}\), which is the joint distribution \(P_{SA} = P_S(S)\pi(A|S)\). In the sequel, we omit to explicitly write the round index \(j\), as the analysis does not depend on it. We consider the particular case in which the distortion measure \(d(\tilde{Q}_N, P_{SA})\) respects the following properties: it is 1) nonnegative; 2) upper bounded by a constant \(D_{max}\); 3) continuous in \(P_{SA}\) at \(\tilde{Q}_N\); 4) convex in \(P_{SA}\), and such that 5) \(d(\tilde{Q}_N, P_{SA}) = 0 \iff \tilde{Q}_N = P_{SA}\). Given the assumptions above, the authors of [38] provide the rate-distortion function \(R(D)\), i.e., the minimum rate \(R = \log_2 B\) bits per symbol such that \(\mathbb{E}[Q_{SA}|d(\tilde{Q}_N, P_{SA})] \leq D\), in the limit when \(N\) is arbitrarily large. Here the expectation is taken with respect to the distribution \(Q_{SA} = P_S(S)Q(A|S)\), where \(Q(A|S)\) is the sampling policy decoded by the controller from the message sent by the decision-maker. The solution is given by
\[
R(D) = \min_{Q_{SA}: d(\tilde{Q}_N, P_{SA}) \leq D} I(S; A). \tag{6}
\]

As we can see, in the asymptotic limit when \(N \to \infty\), the problem admits a single-letter solution, which also serves as a lower bound for the finite agent scenario. Let \(R_{\pi_j}\) denote the rate required to perfectly convey the policy \(\pi_j\), i.e., with zero distortion. From Eq. (6), we can see that \(R_{\pi_j} = I(S; A)\), where the mutual information is computed under \(P_{SA}\) dictated by the policy.

### D. Achievable Rate

We now state the rate condition under which it is possible to achieve sub-linear regret. In the analysis, the available rate \(R\) is considered fixed in each round \(j\). First of all, we denote with \(H(A^*)\) the entropy of the arms under the marginal distribution \(\pi^*(a) = \sum_s P_S(s)\pi^*(a|s)\), where \(\pi^*\) is the optimal policy, i.e., the one that selects, \(\forall s \in S\), the arm \(a^* = \arg \max_{a \in A} \alpha(s, a)\). We start by stating the following Lemma, which provides a rate limit below which it is not possible to achieve sub-linear regret.

**Lemma 1:** If \(R < H(A^*)\), it is not possible to achieve sub-linear Bayesian system regret.

**Proof:** See Appendix-B.

The following Lemma provides the achievability part.

**Lemma 2:** If \(R > H(A^*)\), then it is possible to achieve sub-linear Bayesian system regret in the limit \(N \to \infty\).

**Proof:** See Appendix-B.

We can see that, due to Lemma 2, even if for some round \(j\), \(R_{\pi_j} > R\), as long as \(R > H(A^*)\), it is still possible to achieve sub-linear regret. According to the definition in Eq. (3), this implies that, as \(N \to \infty\), any rate \(R > H(A^*)\) is achievable, while any rate \(R < H(A^*)\) is not achievable. The entropy of the marginal \(\pi^*(a)\) is thus the fundamental information-theoretic limit of the problem to achieve sub-linear regret.

### V. Policy Compression

We are now ready to study compression strategies to deal with the case in which it is not always possible to convey the policy with zero distortion, i.e., \(\exists j\) s.t. \(R_{\pi_j} > R\). In such cases, it is not clear which is the message \(m_j\) the decision-maker should transmit to the controller. As a consequence of Eq. (6), when \(R_{\pi_j} > R\), the sampling policy \(Q_j\) adopted by the controller may differ from \(\pi_j\). In [36], the authors provide some theoretical guidelines to construct approximate sampling policies to make the posteriors, i.e., \(\pi(a|s)\), converge to the optimal one achieving sub-linear regret, even when an agent is sampling with a different policy \(Q(a|s)\). In particular, they studied the case in which the sampling distribution \(Q\) differs from the target posterior \(\pi\), using the \(\alpha\)-divergence \(D_\alpha(\pi, Q)\) as distortion measure, which is defined as
\[
D_\alpha(\pi, Q) = \frac{1 - \int x^\alpha Q(x)^{1-\alpha} dx}{\alpha(1-\alpha)}. \tag{7}
\]

We now provide two theoretical compression schemes that adopt the forward KL divergence \(D_{KL}(\pi, Q) = D_{KL}(\pi|Q)\), and the reverse KL divergence \(D_{KL}(Q|\pi)\), as distortion functions, which are the two cases considered also in [36]. We remember that, for two discrete distributions \(p\) and \(q\) such that \(p\) is absolutely continuous with respect to \(q\), i.e., if \(x \in X\) is such that \(q(x) = 0\) implies \(p(x) = 0\), the KL divergence is defined as \(D_{KL}(p||q) = \sum_{x \in X} p(x) \log \frac{p(x)}{q(x)}\).
Another reason to adopt these two metrics is related to the fact that it is possible to bound, in each round $j$, the gap between the expected reward of the target policy $\pi_j$, and that obtained by using the approximate policy $\hat{Q}$. To this end, we denote by $\mu^x(s,a)$ the average reward obtained in context $s$ when using policy $\pi$, i.e., $\mu^x(s,a) = E_{\pi(a|\cdot)}[\mu(s,a)]$, and find
\[
\mathbb{E}_S \left[ |\mu^x(S,A) - \mu^{\hat{Q}}(S,A)| \right] \\
= \sum_{s \in S} P_S(s) \sum_{a \in A} |\mu(s,a)| - Q_j(a|s)| \\
\leq \sum_{s \in S} P_S(s) |\mu(s,a) - Q_j(a|s)| \\
= \sum_{s \in S} P_S(s) |\mu(s,a) - Q_j(a|s)| \\
\leq C \cdot \mathbb{E}_S \left[ \sqrt{D_{KL}(\pi(S)|Q_j(S))} \right] ,
\]
where (a) holds because $\mu(s,a) \in [0,1]$ by assumption, (b) is the Pinsker’s inequality, and $C$ is a constant that depends on the base of the logarithm in the divergence, e.g., $C = \sqrt{\frac{1}{2\ln 2}}$ if in base 2. We notice that, by swapping the roles of the two distributions in the last inequality, it is possible to obtain the version with the reverse KL divergence.

**Observation:** It is known that by minimizing the forward KL divergence $D_{KL}(\pi||Q)$, we obtain a more “spread” solution $Q\sigma$ that puts mass around the peaks of $\pi$, leading to a solution that puts mass around the peaks of $\pi$. Consequently, the exploration-exploitation trade-off is usually biased towards a more exploration-seeking solution when minimizing $D_{KL}(\pi||Q)$, and to a more exploitation-seeking solution when minimizing $D_{KL}(Q||\pi)$.

### A. Reverse KL Divergence

In this case, the adopted distortion function is
\[
d(Q_{SA}, \pi_{SA}) = \mathbb{E}_P [D_{KL}(\pi(S)||Q(S))] .
\]
In the following lemma, we provide the shape of the policy $Q(a|s)$ that can achieve the minimum in Eq. (6), i.e., the policy that minimizes the required rate while meeting the constraint on the reverse KL divergence.

**Lemma 3:** Given the constraint on the reverse KL divergence $D_{KL}(Q||\pi) \leq \delta$, the policy that achieves the minimum in Eq. (6) is
\[
Q_j(a|s) = \frac{\hat{Q}(a|s) \cdot \pi(a|s)^{1-\lambda}}{Z} ,
\]
where $\lambda \in [0,1]$ is such that $D_{KL}(Q_j||\pi) = \delta$, $\hat{Q}(a)$ is the marginal, i.e., $\hat{Q}(a) = \sum_{s \in S} P_S(s)Q_j(a|s)$, and $Z$ is the normalization factor.

**Proof:** See Appendix-C1.

We can see that, when $\lambda = 0$, we have $Q(a|s) = \pi(a|s)$, and so $R_Q = R_\pi = I(S;A)$, where the mutual information is computed with respect to $P_{SA}$. As $\lambda$ increases from 0 to 1, the policy $Q$ tends to be more similar to the marginal $\pi(a) = \sum_{s \in S} P_S(s)\pi(a|s)$, which requires zero rate. Indeed, when $\lambda = 1$, the marginal $\pi(a)$ does not require context information, and so the agents would sample arms regardless of the contexts $s_j,n$.

### B. Forward KL Divergence

We consider $d(Q_{SA}, \pi_{SA}) = \mathbb{E}_P [D_{KL}(\pi(S)||Q(S))]$. As for the reverse case, we report the shape of the optimal compressed policy.

**Lemma 4:** Given the constraint on the forward KL divergence $D_{KL}(\pi||Q) \leq \delta$, the policy that achieves the minimum in Eq. (6) is
\[
Q_j(a|s) = \frac{\lambda \cdot \pi(a|s) \cdot W_0(\frac{\lambda \cdot \pi(a|s)}{Q_j(a|s)\cdot Z})}{Z} ,
\]
where $\lambda$ is such that the maximum distortion is achieved with equality $D_{KL}(\pi||Q_j) = \delta$, $W_0(\cdot)$ is the Lambert function [39], $\hat{Q}(a)$ is the marginal, i.e., $\hat{Q}(a) = \sum_{s \in S} P_S(s)Q_j(a|s)$, and $Z$ the normalization factor.

**Proof:** See Appendix-C2.

We observe that the above compressed policies can achieve the minimum in Eq. (6) as $N \rightarrow \infty$, characterizing the information-theoretic limit and serving as a lower bound for more practical schemes that can work for finite $N$.

**Remark:** We notice that, in general, $D_{KL}(P||Q)$ always satisfies 1) and 3) — 5) defined in Section IV-C, but not condition 2), i.e., it may be unbounded. We now define $\sigma_q := \min_{x \in \mathcal{X}} q(x)$, if $p$ is absolutely continuous with respect to $q$, by the inverse Pinsker’s inequality we have
\[
D_{KL}(p||q) \leq \frac{2TV(p,q)^2}{\sigma_q} \leq \frac{8}{\sigma_q} = D_{\max} ,
\]
to which $TV(p,q)$ indicates the total variation distance between the distributions $p$ and $q$. To meet this requirement, we further assume $\exists \epsilon > 0 : \pi(a|s) \geq \epsilon$, $\forall s \in S$ and $a \in A$, whenever the compression scheme has to be applied. This additional assumption is reasonable in our context, as the target policy $\pi$ needs exploration during training. Once $\pi_j$ has converged to the optimal policy (see Appendix-B), further compression would lead to sub-linear regret.

### C. Practical Coding Scheme

To find practical coding schemes, we propose a solution that is based on the idea of context reduction, and computes compact context representations. In essence, the decision-maker constructs a message containing the new context representations $\hat{s}(s) \in \hat{S}$ of $s$, one for each agent, and sends it over the channel. Once the agents have received the message, they can sample the arms according to a common policy $Q_j(a|\hat{s})$, which is defined on the compressed context space $\hat{S}$. If the rate constraint imposes $B$ bits per agent, it means that it is possible to transmit at most $2^B$ different contexts to each agent. The idea is to group the contexts into $2^B = M$ clusters $\hat{s}_1, \ldots, \hat{s}_M$, minimizing $d(Q_{SA}, P_{SA})$, where $Q_{SA}$ is the new policy defined on the compressed contexts $\hat{s}(s) \in \hat{S}$. Again, we avoid to explicitly write the round index $j$, as the scheme does not depend on it.
To find the clusters and relative policies we employ the well-known Lloyd algorithm [40], which is an iterative process to group states into clusters. First of all, knowing the policy $\pi$, the decision-maker maps each state $s \in S$ into a $K$-dimensional point $\tilde{\alpha}_s = \pi(s) \in \Delta_K$, finding $|S| = S$ different points $\tilde{\alpha}_1, \ldots, \tilde{\alpha}_S$. Then, it generates $2^B = M$ random points $\mu_1, \ldots, \mu_M \in \Delta_K$ as initial centroids, i.e., representative policies, and iterates over the following two steps:

1) Assign to each point $\tilde{\alpha}_s$ the class $c_s \in \{1, \ldots, M\}$ such that $c_s = \arg \min_{c \in \Delta_K} E_{p_s} D_{KL}(\mu_c || \pi(s))$, i.e., minimizing the average $\alpha$-divergence between the representative $\mu_c$ and the original policy, which is the point $\tilde{\alpha}_s$. For each cluster $c$, we now denote by $S_c$ the set containing the contexts associated to the policies in that cluster.

2) Update $\mu_1, \ldots, \mu_M$ such that $\mu_c = \arg \min_{\mu \in \Delta_K} \sum_{s \in S_c} p(s) D_{KL}(\mu || \pi(s))$, which is still a convex optimization problem when using $\alpha = 0$ or $\alpha = 1$, and can be solved applying the Lagrangian multipliers. The solution is

$$\mu^c = \frac{\prod_{s \in S_c} \pi(s) || P(s) \mu^c || Z}{Z},$$

(10)

when using $D_{H\to0}(\mu^c, \pi(s)) = D_{KL}(\mu^c || \pi(s))$, and

$$\mu^c = \frac{\sum_{s \in S_c} P(s) \pi(s) || Z}{Z},$$

(11)

when using $D_{H\to1}(\mu^c, \pi(s)) = D_{KL}(\pi(s) || \mu^c)$. Here, the product is to be considered element-wise, $A(S_c)$ is the sum of the contexts probabilities in $S_c$, i.e., $A(S_c) = \sum_{s \in S_c} P(s)$, and $Z$ is the normalizing factor. After computing the new centroids, we go back to step 1). The derivations of Eq. (10) and Eq. (11) are provided in Appendix-C1 and Appendix-C2, respectively.

The process continues until the new solution does not decrease the average distortion between the cluster policies and the target ones.

**Observation:** Note that the controller is assumed to know the $M$ policies from which it samples the arms of the agents. This can be transmitted at the beginning of each round. In this case, the scheme is efficient as long as $N \log_2 K \gg BP \log_2 K$, where $P$ is the number of bits used to represent the values of the Probability Mass Function (PMF) $Q_{\pi}(s)$. For this reason, we provide a scheme where the new policy is updated not at every transmission, but only when the new target $\pi$ has changed considerably. In particular, if we denote with $\pi^{\text{old}}$ the policy defined over the compressed context representation, with $\pi^{\text{last}}$ the last policy used to compute $\pi^{\text{old}}$, and with $\pi$ the updated target policy, we compute and transmit $\pi^{\text{old}}$ every time $D_{H}(\pi^{\text{last}}, \pi) > \xi$ exceeds a threshold $\xi$.

**Algorithm Complexity:** The complexity of the algorithm is related to solving the k-means problem, which is known to be NP-hard. Consequently, the solution relies on state-of-the-art heuristic clustering schemes, which have complexity $O(SKM)$ [41], where $S$ is the number of states, $K$ is the number of total actions, $M$ is the number of clusters, which in turn depends on the available rate, and $I$ is the total number of iterations needed to converge to an acceptable solution, i.e., when

the performance of two consecutive solutions is within a sensitivity threshold $\epsilon$. Specifically, for our simulations reported in Section VI, we chose the Lloyd’s algorithm [40], and we notice that $\epsilon$ sensitivity target is always achieved within 10 iterations, i.e., $I < 10$.

### D. Policy Compression, Information Bottleneck and Maximum Entropy Reinforcement Learning

We discuss here the connection of the compression schemes presented above with two popular training strategies in RL: the information bottleneck (IB) approach [5], and maximum entropy reinforcement learning (MERL) [8].

The IB method is a popular way to train an RL agent to maximize a target cumulative reward, with an additional regularization term which accounts for mutual information $I(S; A)$ between the states and actions. As done in [6], [33], [34], the target function $J(\pi)$ to be maximized with respect to policy $\pi$ is

$$J^{\text{IB}}(\pi) = \mathbb{E}[R^\pi] - \beta I(S; A) = \mathbb{E}[R^\pi] - \beta H(A) + \beta H(A|S),$$

(12)

inducing the policy $\pi$ to forget task-independent state information, achieving a better generalization performance.

A different but similar concept is that of MERL, in which the RL agent is trained to maximize the reward, together with a regularization term that is the entropy of the conditional policy given the state [7], [8], i.e.,

$$J^{\text{MERL}}(\pi) = \mathbb{E}[R^\pi] + \beta H(A|S),$$

(13)

inducing the policy $\pi$ to more exploration. As proved in [7], MERL naturally appears in the Control as Inference framework, in which a probabilistic view of the RL problem is adopted. Specifically, when using a uniform prior distribution over the actions, inferring the optimal posteriors leads to the optimization object in Eq. (13).

If we now write the distortion-rate function of the RC-CMAB problem, and solve it using the Lagrangian multiplier method, we end up minimizing the objective

$$\ell^{\text{RC-CMAB}}(Q) = d(Q_{\pi^*}, \pi_{\pi^*}) + \lambda J(S, A),$$

(14)

where the term $d(Q_{\pi^*}, \pi_{\pi^*})$ plays the role of $\mathbb{E}[R^\pi]$, i.e., it is related to reward maximization by playing according to the optimized posterior, and $\lambda J(S, A)$ is implied by the constraint on the rate. However the difference is that, in our analysis, the parameter $\lambda$ is optimized to meet the rate constraint in each round, and so it is determined by the problem, rather than being a hyper-parameter to be fine tuned as in the other two formulations. Moreover, in Section IV-D we proved that, if $\lambda$ is such that $R_{Q_{\pi^*}} = I_{Q_{\pi^*}}(S; A) < R_{\pi^*} = H(A^*)$, then the learning algorithm cannot achieve sub-linear regret. Consequently, it is important to carefully tune the coefficient $\beta$ when optimizing the policy with the IB and MERL methods. This connection will be also highlighted by the results of the experiments in Section VI.
VI. NUMERICAL RESULTS

We now analyze the RC-CMAB problem presented in Section III, and apply both the theoretical and practical policy compression schemes to solve it. In particular, we compare the performance of the Perfect agent, which applies TS without any rate constraint, and thus admits samples from the true posterior $\pi$, with the performance of the rate-constrained algorithms Comm R-KL, Comm F-KL, Cluster R-KL, and the Cluster F-KL agents. The rate-constrained agents adopt TS at the decision-maker and, when the rate is not sufficient, transmit a compressed policy $Q$ with the different schemes. Specifically, the Comm R-KL agent uses the optimal scheme with reverse KL divergence, provided in Section V-A; the Comm F-KL agent uses the forward KL divergence as explained in Section V-B; the Cluster R-KL and Cluster F-KL agents implement the practical coding scheme provided in Section V-C. In every experiment, the context distribution $P_S$ is uniform over the $S = 16$ contexts, and there are $K = 16$ feasible arms, $N = 50$ agents, and the total number of system rounds is $J = 400$. In all the experiments the environment is such that, for each context $s \in S = \{0, \ldots , 15\}$, the best average reward is given by the arm $a \in A = \{0, \ldots , 15\}$ such that $a = \lceil \frac{s}{G} \rceil$, where $G$ is an experiment parameter. In particular, the reward behind arm $a$ with context $s$ is a Bernoulli random variable with parameter $\mu(s,a) = 0.8$ if $a = \lceil \frac{s}{G} \rceil$, and $\mu(s,a)$ is sampled uniformly in $[0, 0.65]$ otherwise. The average rewards for sub-optimal arms are randomly generated at the beginning of each experiment's run.

This set of parameters allows us to study the performance of the different compression schemes, as the degree of correlation between the optimal action distributions and contexts varies. We do not expect notable changes by varying the number of arms, states, and/or reward distributions. For example, increasing the number of arms, or decreasing the gaps between optimal and sub-optimal arms’ average rewards, would lead to longer training processes for any algorithms [46], without affecting the performance of compressed policies in comparison to the uncompressed ones.

A. Optimal Rate Constraint

In the first experiment, we set $G = 8$, meaning that the arm with the highest expected reward is $a = 0$ in the first 8 contexts, and $a = 1$ with $s \in \{8, \ldots , 15\}$. Consequently, the rate of the optimal policy is $R_{\pi^*} = H(A^*) = 1$ bit, given that $P_S$ is uniform. The maximum available rate $R$ is set to 1 bit for all the agents but the Perfect one. Specifically, the agents Comm R-KL and Comm F-KL adopt a rate $R_{\text{Comm}} = \min\{1, R_{\pi}\}$. Whenever $R_{\pi} > 1$, the compression schemes explained in Section V are adopted. The cluster agents use 1 bit in all the rounds, whereas the Perfect agent can always use $R_{\pi}$. The results are presented in Fig. 2.

As we can see from Fig. 2(a), as the learning process goes on, the required rate to transmit the TS policy increases, as the mutual information between $S$ and $A$ increases. Moreover, the rate $R_{\pi}$ converges to 1. However, it is interesting to observe Fig. 2(b), that reports the average state regret as a function of the virtual round index $t = 1, \ldots, JN$. The agents Comm R-KL, Comm F-KL and Perfect all achieve similar performance. However, agent Cluster F-KL is not able to converge to the optimal policy. Surprisingly, the best performing agent is the more practical Cluster R-KL. We argue that this is because it is the agent that makes better use of the IB trick, as 16 different policies are not necessary to represent the optimal responses, given that 2 different distributions are sufficient. We show thus that, when the hyper-parameter $\beta$ in Eq. (12) is optimally tuned, i.e., such that $I(S; A) = H(A^*)$, it is possible to gain in performance when adding the regularization term.

B. Training With Rate Constraint

In this second experiment, we set $G = 2$, and $R = 2$ for the first 200 system rounds, and $R = 3$ for the remaining 200 rounds. Given $G = 2$, we now have $R_{\pi} = 3$ bits, and so the constraint can potentially damage the training process of the rate-constrained agents. As we can see from Fig. 3(b), the Perfect agent can easily converge obtaining sub-linear regret. The regret of the rate-constrained agents grows linearly as their rate is imposed to $R = 1$. However, we can see that the two agents that theoretically-optimally trade rate with policy distortion, present the smallest slope in the regret curve. Again, the Cluster R-KL agent outperforms Cluster F-KL, and achieves almost zero per-round regret as soon as $R$ jumps to 3, meaning that the posteriors converged to the optimal ones.
Fig. 3. Rate $R_{\pi}$ needed to reliably transmit the policy $\pi$ as a function of the system round $j$, together with the imposed rate $R$ (a); average state regret obtained by the different algorithms as a function of the virtual round $t$ (b). In this case, $G = 2$. Curves indicates average rewards ± one standard deviation over 5 runs.

even when sampling with $R = 1$. This is not true for Cluster $F$-$KL$, as it presents larger slope in the regret curve, and the regret keeps growing even when $R = 3$. These observations are consistent with the analysis in [36].

C. Practical Coding Scheme

This last experiment (Fig. 4) presents the effectiveness of the practical coding schemes described in Section V-C, in the most complex case $G = 1$, i.e., the association between context and optimal arm is a one-to-one map, and the optimal policy $\pi^*$ needs $R_{\pi^*} = 4$, which is also equal to the context entropy. Consequently, in this case the maximum-complexity policy, i.e., a different arm distribution for each context, is needed [32], and so the IB should be carefully used, and $\beta = 0$ in the end is needed, as no compression scheme can achieve sub-linear regret with $R < 4$ bits. In this case, the number of bits the Cluster agents can use is set to $R_{\text{Cluster}} = \lceil R_{\pi^*} \rceil$. What is interesting to notice here is that, in the long run, the Cluster $F$-$KL$ agent outperforms Cluster $R$-$KL$, as it is able to converge to the optimal scheme by introducing more exploration. Indeed, even if in the first 400 virtual rounds the Cluster $R$-$KL$ regret curve is below that for Cluster $F$-$KL$, from $t \sim 600$ the Cluster $F$-$KL$ agent presents better performance. The reason is that, in this case, more exploration should be encouraged in the first steps to discover optimal policies, as done in MERL, by sampling sub-optimal arms more frequently.

Fig. 4. Rate $R_{\pi}$ needed to reliably transmit the policy $\pi$ as a function of the system round $j$, together with the imposed rate $R$ (a); average state regret obtained by Perfect and Cluster algorithms, as a function of the virtual round $t$ (b). In this case, $G = 1$. Curves indicates average rewards ± one standard deviation over 5 runs.

VII. CONCLUSION

In this paper, we have proposed and studied a novel rate-constrained remote RL problem, called RC-CMAB. We first proved the performance of the TS strategy when no constraint on the rate is imposed, and the information-theoretic limit for the rate needed to achieve sub-linear regret. We then studied schemes to compress the decision-maker’s policy whenever the available rate is not sufficient to reliably transmit the intended actions to the controller. We considered $\alpha$-divergence as the distortion metric in the rate-distortion problem, and provided the shape of the policies that theoretically optimize the rate-distortion trade-off in close-form, in the cases $\alpha \to 0$, and $\alpha \to 1$, which lead to the reverse and forward KL divergences, respectively. We further proposed a practical compression scheme that relies on the idea of context clustering, and can be adopted to minimize the two analyzed divergence functions. The numerical results confirmed the limit on the achievable rate, and the performance gain that can be achieved when using proper compression strategies in the rate-constrained cases. We further connected and discussed with experiments the relation between our policy compression schemes and the information bottleneck approach in RL.

Future steps include the adoption of more advanced practical compression algorithms that reduce the gap between the cluster and optimal schemes. Moreover, the extension of the problem in the context of more general remote RL problems,
where the next state depends on the current state and the action taken, is an interesting and challenging direction, as modern algorithms involve the parallelization of the training process exploiting a multitude of agents interacting with many replicas of the same environment.

**APPENDIX**

A. Regret Analysis

In this section we prove Theorem 1. As explained Section IV-B, we align the interactions between agents and environment in time, and consider virtual rounds \( t \in \{1, \ldots, J \} \), \( N \) = \( T \), and rewrite the Bayesian system regret as follows:

\[
\begin{align*}
BR(\pi^{TS}, J) &= \mathbb{E} \left[ \sum_{t \in \mathcal{T}} \mu \left( s_{j,n}, a^* \left( s_{j,n} \right) \right) - \mu \left( s_{j,n}, A_{j,n} \right) \right] \\
&= \mathbb{E} \left[ \sum_{t=1}^{J} \mu \left( s_1, a^* \left( s_1 \right) \right) - \mu \left( s_1, A_1 \right) \right] \\
&= \mathbb{E} \left[ \sum_{t \in \mathcal{T} \cap \mathcal{S} \cap \mathcal{T}^a} \mu \left( s_1, a^* \left( s_1 \right) \right) - \mu \left( s_1, A_1 \right) \right]
\end{align*}
\]

where \( \mathcal{T}_s = \{ t \in \mathcal{T} : s_1 = s \} \), i.e., all time-steps where context \( s \) was sampled, and \( |\mathcal{T}_s| = T_s \). We now define the upper and lower bounds for the reward average \( \mu(s, a) \), which hold with high probability, and are used to bound the average per-round regret

\[
\begin{align*}
U_t(s, a) &= \mu_{-1}(s, a) + \sqrt{\frac{2 + 6 \log T_s}{\phi_{t-1}(s, a)}} \\
L_t(s, a) &= \mu_{-1}(s, a) - \sqrt{\frac{2 + 6 \log T_s}{\phi_{t-1}(s, a)}}
\end{align*}
\]

where \( \mu_t(s, a) \) is the empirical average return of \( a \) with context \( s \) at time \( t \), and \( \phi_t(s, a) \) is related to the number of times arm \( a \) has been pulled until time \( t \) with context \( s \), and will be better explained later. We now use Proposition 1 from [42], which allows us to write

\[
BR(\pi^{TS}, T) = \mathbb{E} \left[ \sum_{t \in \mathcal{T}} U_t(s_t, A_t) - \mu(s_t, A_t) \right] + \mathbb{E} \left[ \sum_{t \in \mathcal{T}} \mu(s_t, a^* \left( s_t \right)) - U_t(s_t, a^* \left( s_t \right)) \right].
\]

and apply [42, Proposition 2] obtaining, whenever \( T > SKN \),

\[
\begin{align*}
BR(\pi^{TS}, T) &\leq \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} \mathbb{E} \left[ U_t(s_t, A_t) - L_t(s_t, A_{t_0}) \right] + SKN.
\end{align*}
\]

We now consider the single context regret. For every \( s \in \mathcal{S} \), we define \( T^a_s = \{ t_s \in \mathcal{T}_s : A_{t_s} = a \} \) and \( |T^a_s| = T^a_s \), bounding the quantity

\[
\begin{align*}
&\sum_{t \in \mathcal{T}^a_s} U_t(s_t, A_t) - L_t(s_t, A_{t_0}) \\
&= \sum_{a \in \mathcal{A}} \sum_{t \in \mathcal{T}^a_s} U_t^a(s, a) - L_t^a(s, a) \\
&\leq \sum_{a \in \mathcal{A}} \left[ 1 + 2 \sqrt{2 + 6 \log T_s^a} \sum_{t \in \mathcal{T}^a_s} \left( 1 + \phi_{t-1}^a(s, a) \right) \right]^{-\frac{1}{2}}
\end{align*}
\]

is the last time the policy has been updated, i.e., at the end of the previous round. Now, \( \phi_t(s, a) \geq C_t(s, a) - T \), since it is not possible to sample more than \( T \) times the same arm in one round, given that the number of agents is \( N \). However, the quantity \( 1 + C_t(s, a) - T \) is not defined for \( C_t(s, a) < T \). If \( C_t(s, a) < T \), we can always write \( \phi_t(s, a) \geq \frac{C(a, a_0) + 1}{2} \).

1) Finite-Time Analysis: With the observations above, we can now prove an upper bound for the finite-time regret. We first notice that

\[
\sum_{t \in \mathcal{T}^a_s} \left( 1 + \phi_{t-1}^a(s, a) \right)^{-\frac{1}{2}} \leq \sum_{t \in \mathcal{T}^a_s} \left( 2 + \frac{C(a, a_0) + 1}{N} \right)^{-\frac{1}{2}}
\]

where \( a \) is a consequence of the bounds defined in Eq. (15). In standard MAB analysis, as in [42], \( \phi_t(s, a) \) is basically the number of times arm \( a \) has been sampled up to time \( t \) with context \( s \), that we denote with \( C_t(s, a) \). This quantity is critical to bound the intervals \( U_t(s, a) - L_t(s, a) \). However, in our formulation, we define \( \phi_t(s, a) = C_{t(i)}(s, a) \), where

\[
j(t) = [t/N] \cdot N
\]

Here (a) is a consequence of the bounds defined in Eq. (15). The regret analysis is based on the number of agents is \( N \) and the number of times arm \( a \) has been pulled until time \( t \) with context \( s \).
The equalities (a) and (b) come from the Cauchy-Schwarz inequality, and the definitions of $T^\pi_s$, $T_s$, and $T$.

2) Asymptotic Regret Analysis: However, in the asymptotic case $N \to \infty$, we get rid of the first constant terms when arms are pulled less than $N$ times. Consequently, we can use the bound $\phi_t(s, a) \geq C_t(s, a) - N$, and follow the same analysis for the finite-time case. We can see that the factor $T$ does not scale with $N$, obtaining

$$BR(\pi, T) \in O\left(\sqrt{KTN\log T}\right).$$

B. Achievable Rate

In this section we provide the detailed proofs of Lemma 1 and Lemma 2. To this end, we denote by $H_q(X)$ and $I_q(X; Y)$ the entropy and the mutual information computed with respect to the probability distribution $q$. Again, as introduced in Section IV-D, $H(\pi^*)$ is the entropy of the optimal actions, i.e., computed with respect to $\pi^*(a) = \sum_{s \in S} P(s)\pi(a^*|s)$, where $\pi^*(a|s)$ is the optimal policy in context $s$.

We start by proving the following lemma.

**Lemma 5:** Assuming that Thompson Sampling policy $\pi_j(a|s)$ achieves sub-linear Bayesian system regret, then

$$\lim_{N \to \infty} I_q(S; A) = \lim_{N \to \infty} H_q(A) = H(\pi^*).$$

**Proof:** First of all we notice that, in order to achieve sub-linear Bayesian system regret, it is necessary to achieve sub-linear regret in all contexts $s \in S$, given the assumption that $\forall s \in S$, $P_\pi(s) > 0$. We then write $I_q(S; A) = H_q(A) - H_q(A|S)$. Following [43, Th. 2], if $\pi_j(a|s)$ achieves sub-linear regret, then $\forall s \in S$,

$$\lim_{j \to \infty} \pi_j(a = a^*|s) = 1 \quad \text{when } a^* \text{ is the optimal arm},$$

$$\lim_{j \to \infty} \pi_j(a = a'|s) = 0 \quad \text{when } a' \neq a^*.$$

Consequently, in the limit, $\pi_j(a|s)$ is a deterministic function, thus

$$\lim_{j \to \infty} H_q(A^*|S) = 0,$$

which concludes our proof.

We start by proving Lemma 2, which we repeat below.

**Lemma 2** If $R > H(\pi^*)$, then it is possible to achieve sub-linear Bayesian system regret, in the limit $N \to \infty$.

**Proof:** We denote with $R_n$ the rate needed to perfectly convey the TS policy to the controller at round $j$, and let $\epsilon > 0$ s.t. $R = H(\pi^*) + \epsilon$, where $R$ is the available communication rate. We now provide a scheme that guarantees sub-linear regret.

If the policy $\pi_j$ generated from TS has $R_n \leq H(\pi^*) \forall j = 1, \ldots, J$, then Theorem 1 ensures sub-linear Bayesian system regret if sampling with $\pi_j$. If $\exists j$ such that $R_n > R$, generate parameters $\rho_j$ such that $\rho_j \in (0, 1)$ and $\sum_{j=1}^J \rho_j = \rho$, as explained in [36]. Then, with probability $\rho_j$ play $a$ uniformly at random, and with probability $1 - \rho_j$, play according to a policy $Q_j(a|s)$, which satisfies the rate-distortion constraint $I_{Q_j(a|s)}(S; A) \leq R$, which can be transmitted to the controller, using as distortion measure the reverse KL divergence $d(Q_{SA}, \pi_{SA}) = E_p[D_{KL}(\tilde{Q}(\cdot|S)||\pi(\cdot|S))]$. Following [44, Lemma 14], with this strategy enough exploration is guaranteed for the posterior policy, i.e., the one stored by the decision-maker, to concentrate. Consequently, by Lemma 5, there exists a finite $j_0$ s.t. $\forall j > j_0$, $R_n < H(\pi^*) + \epsilon$, in the limit $N \to \infty$. This means that, for the first $j_0$ rounds, both the target and the approximating policies would play sub-optimal arms with non-zero probabilities. Then, $\forall j > j_0$, it is possible to play the exact TS policy, leading to the optimal decisions for all future steps, and hence, to a sub-linear regret. The above procedure holds $\forall \epsilon > 0$, and so $VR > H(\pi^*)$.

As we can see, the strategy that achieves sub-linear regret consists of using TS at the decision-maker, which updates the posteriors according to the Bayes rule, and from which a target policy $\pi^*$ is computed. Then, the decision-maker computes an approximate policy $Q_j(a|s)$ as indicated in the proof, and it samples the arms according to $Q_j(a|s)$. We further notice that the limit $H(\pi^*)$ serves as a lower bound for practical schemes, as to achieve it we need $N \to \infty$.

We observe that [43, Th. 2] states that, if the TS strategy achieves sub-linear regret, then the policy converges to the deterministic policy selecting with probability one the optimal action. However, the converse is not always true in general, i.e., there could exist policies that play sub-optimal arms infinitely many times as $j \to \infty$, and still achieve sub-linear regret. The point is that such policies must play the optimal arms for most of the rounds, and could pull sub-optimal arms for a sub-linear amount of rounds. However, to play optimally in one round, the decision-maker needs $R \geq R_{\pi^*} = H(\pi^*)$.

We are now ready to prove Lemma 1.

**Lemma 6:** If $R < H(\pi^*)$, it is not possible to achieve sub-linear Bayesian system regret.

**Proof:** As explained above, to play optimally in one round, the decision-maker needs $R \geq R_{\pi^*} = H(\pi^*)$, in the limit $N \to \infty$. If $R < H(\pi^*)$, as a consequence of Eq. (6), the policy $Q(a|s)$ conveyed to the controller has non-zero distortion $d(Q_{SA}, \pi_{SA}) = D > 0$. If we take the $L_1$ norm as distortion measure, $Q(a|s)$ would sample a sub-optimal arm with constant probability of at least $D$ in every round $j = 1, \ldots, J$. Consequently, sub-linear regret cannot be achieved.

C. Policy Compression

To compute the optimal policy that solves Eq. (6) with a specific distortion function, we applied the well known Blahut-Arimoto iterative algorithm [9] that, given the considered distortion functions, is guaranteed to converge to the solution [45].

We rewrite the optimization objective of Eq. (6) as a double minimization problem ([9, Sec. 10.8])

$$R(D) = \min_{Q(a)} \min_{Q_j(a)} \min_{d((\tilde{Q}_{SA}, \pi_{SA}) \leq D} \sum_{s,a} P_S(s)Q_j(a|s) \log_2 \frac{Q_j(a|s)}{Q(a)}.$$ (19)

Following ([9, Lemma 10.8.1]), the marginal $\tilde{Q}(y) = \sum_{x} P(x)Q_j(x|y)$ has the property

$$\tilde{Q}(y) = \arg \min_{Q_j} D_{KL}(P(x|y|x)) \left| P(x)Q_j(y|y) \right|,$$ (20)
that is, it minimizes the KL-divergence between the joint and the product \( P(x)Q(y) \). This means that \( \tilde{Q}(a) \) obtained by solving Eq. (19) is indeed the marginal over the arms induced by \( Q(a) \). Exploiting this formulation, it is possible to apply the iterative Blahut-Arimoto algorithm to solve the problem and find the solution [9]. The process is initialized by setting a random \( \tilde{Q}_0(a) \), which is used as a fixed point to compute

\[
\tilde{Q}_1^t(a) = \arg\min_{\tilde{Q} : (Q_{sa}, P_{sa}) \leq D} \sum_s P(s) \sum_a Q(a|s) \log_2 \frac{Q(a|s)}{\tilde{Q}_0(a)}.
\]  

(21)

From \( \tilde{Q}_1^t(a) \), we compute the optimal \( \tilde{Q}_1(a) \) by solving Eq. (20), which is simply the marginal \( \tilde{Q}_1(a) = \sum_s P(s) \tilde{Q}_1^t(a|s) \). The process is iterated until convergence. We now solve the inner minimization problem, i.e., Eq. (21) with fixed \( \tilde{Q}(a) \) and distortion \( \mathbb{E}_{P_s[D_a(Q, \pi)]]} \), for \( \alpha \to 1 \), and \( \alpha \to 0 \).

1) Reverse KL Divergence (\( \alpha \to 0 \)): To solve this problem, we solve the related Lagrangian

\[
\mathcal{L}(Q(a|s), \lambda, \mu) = \sum_s P(s) \sum_a Q(a|s) \log \frac{Q(a|s)}{\pi(a|s)} + \lambda \left( \sum_s P(s) \sum_a Q(a|s) \log \frac{Q(a|s)}{\pi(a|s)} - D \right) + \mu \left( \sum_s P(s) \sum_a Q(a|s) - 1 \right)
\]

where the Lagrangian multiplier \( \lambda \) has to be optimized to meet the constraints on the divergence, whereas \( \mu \) ensures that the solution is a probability distribution, i.e., the elements sum to one. The positivity constraints on the terms are already satisfied by the fact that the solution has an exponential shape. We first take the derivative of the Lagrangian w.r.t. to the terms \( Q(a|s) \):

\[
\frac{\partial \mathcal{L}(Q(a|s), \lambda, \mu)}{\partial Q(a)} = P(s) \log \frac{Q(a|s)}{\pi(a|s)} + P(s) - \sum_s P(s') Q(a|s') \frac{P(s)}{Q(a)} + \lambda \left( \sum_s P(s) \sum_a Q(a|s) \log \frac{Q(a|s)}{\pi(a|s)} + 1 \right) + \mu P(s)
\]

and set it to zero, finding

\[
\log \frac{Q(a|s)}{\pi(a|s)} = - (\lambda + \mu)
\]

\[
Q(a|s) = e^{- (\lambda + \mu) 
\frac{Q(a|s)}{\pi(a|s)}}
\]

(22)

We now define \( \gamma := \frac{1}{1+\mu} \), \( \gamma \in [0, 1] \), and obtain the distribution

\[
Q_\gamma(a|s) = \tilde{Q}(a) \gamma \pi(a|s)^{1-\gamma} \frac{Q(a|s)^{\gamma} \pi(a|s)^{1-\gamma}}{\sum_{a'\in A} \tilde{Q}(a') \gamma \pi(a'|s)^{1-\gamma}} \forall s \in S, a \in A.
\]

By the convexity of KL-Divergence and its triangular inequality, we know the solution lies on the boundary of the constraints, i.e., when \( \mathbb{E}_{P_s[D_{KL}(Q_\gamma||\pi)]} = \delta \).

2) Forward KL Divergence (\( \alpha \to 1 \)): The derivative of the Lagrangian is (here the normalization factor is added in the end)

\[
\frac{\partial \mathcal{L}(Q(a|s), \lambda, \mu)}{\partial Q(a)} = P(s) \log \frac{Q(a|s)}{\pi(a|s)} - \lambda P(s) \pi(a|s) Q(a|s)
\]

and setting it to zero leads to

\[
\frac{\partial \mathcal{L}(Q(a|s), \lambda)}{\partial Q(a)} = 0
\]

\[
\log \frac{Q(a|s)}{\pi(a|s)} - \lambda \pi(a|s) Q(a|s) = 0
\]

\[
\log \frac{\tilde{Q}(a)}{Q(a)} + \lambda \pi(a|s) Q(a|s) = 0
\]

We now define \( x := \frac{1}{Q(a|s)} \), \( \alpha := \lambda \pi(a|s) \), \( \beta := 1 \), and \( \gamma := \log \frac{Q(a)}{\pi(a)} \), obtaining

\[
\alpha x + \beta \log x + \gamma = 0
\]

\[
\alpha x + \log \alpha x + \gamma - \log \alpha = 0
\]

\[
e^{-\alpha x} x = ae^{-\gamma}
\]

\[
x = \frac{1}{\alpha} W_0(\alpha e^{-\gamma})
\]

where \( W_0() \) is the Lambert function [39]. We can now replace the introduced variables with the original terms and normalize, obtaining

\[
Q_\lambda(a|s) = \frac{\lambda \pi(a|s) W_0 \left( \frac{\lambda \pi(a|s)}{Q(a)} \right)}{\sum_{a'} \lambda \pi(a'|s) W_0 \left( \frac{\lambda \pi(a'|s)}{Q(a')} \right)}
\]

where \( \lambda \) such that \( \mathbb{E}_{P_s[D_{KL}(\pi||Q_\lambda)]} = \delta \).

D. Clustering Compression Schemes

1) Reverse KL Divergence: Again, we compute the optimal centroids by solving the Lagrangian

\[
\mathcal{L}(\mu_a, \lambda) = \sum_{s \in S_c} P(s) \sum_{a \in A} \mu_a \log \frac{\mu_a}{\pi(a|s)} + \lambda \left( \sum_{a \in A} \mu_a - 1 \right)
\]

taking its derivative and solving the equality

\[
\frac{\partial \mathcal{L}(\mu_a, \lambda)}{\partial \mu_a} = \sum_{s \in S_c} P(s) \left( \log \frac{\mu_a}{\pi(a|s)} + 1 \right) + \lambda = 0
\]

finding

\[
\log \mu_a A(S_c) = \sum_{s \in S_c} P(s) \log \pi(a|s) + A(S_c) + \lambda
\]

\[
\log \mu_a = \log \frac{A(S_c)}{\mu_a} + \frac{\lambda}{A(S_c)}
\]

\[
\mu_a = \frac{\prod_{s \in S_c} \pi(a|s)^{\frac{P(s)}{A(S_c)}}}{Z}
\]

where \( Z \) is the normalizing factor, obtaining the shape expressed in Eq. (11).
2) **Forward KL Divergence:** In this case, the Lagrangian is

\[
\mathcal{L}(\mu^c_a, \lambda) = \sum_{s \in S_c} P(s) \sum_{a \in A} \pi(a | s) \log \frac{\pi(a | s)}{\mu_a} + \lambda \left( \sum_{a \in A} \mu_a^c - 1 \right).
\]

We take the derivative, and set it equal to zero

\[
\frac{\partial \mathcal{L}(\mu^c_a, \lambda)}{\partial \mu^c_a} = \sum_{s \in S_c} P(s) \left( -\frac{\pi(a | s)}{\mu_a} \right) + \lambda = 0
\]

finding

\[
\mu^*_a = \sum_{s \in S_c} P(s) \pi(a | s) Z
\]

where \( Z \) is the normalizing factor.

**References**

[1] M. Satyanarayanan, “The emergence of edge computing,” *Computer*, vol. 50, no. 1, pp. 30–39, Jan. 2017.

[2] J. Chen and Y. Pan, “Deep learning with edge computing: A review,” *Proc. IEEE*, vol. 107, no. 8, pp. 1655–1674, Aug. 2019.

[3] M. Chen et al., “Distributed learning in wireless networks: Recent progress and future challenges,” 2021, arXiv:2104.02151.

[4] J. Park, S. Samarakoon, M. Bennis, and M. Debbah, “Wireless network intelligence at the edge,” *Proc. IEEE*, vol. 107, no. 11, pp. 2204–2239, Nov. 2019.

[5] N. Tishby, F. C. Pereira, and W. Bialek, “The information bottleneck method,” 2000, arXiv:physics/0004057.

[6] A. Goyal et al., “InfoBot: Transfer and exploration via the information bottleneck,” in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–21.

[7] S. Levine, “Reinforcement learning and control as probabilistic inference: Tutorial and review,” 2018, arXiv:1805.09091.

[8] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, “Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic policy,” 2017, arXiv:1801.01290.

[9] S. Levine, “Reinforcement learning and control as probabilistic inference,” 2019, pp. 1–21.

[10] M. Chen et al., “Distributed learning in wireless networks: Recent progress and future challenges,” 2021, arXiv:2102.08329.

[11] S. Han, H. Mao, and W. J. Dally, “Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding,” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2016, pp. 1–10.

[12] O. A. Hanna, L. F. Yang, and C. Frascoli, “Solving multi-arm bandit using a few bits of communication,” in *Proc. 58th Int. Conf. Mach. Learn.*, 2021, pp. 11215–11236.

[13] M. Agarwal, V. Aggarwal, and K. Azizzadenesheli, “Multi-agent multi-armed bandits with limited communication,” 2021, arXiv:2102.08462.

[14] C. Kalkanli and A. Ozgur, “Batched Thompson sampling,” in *Advances in Neural Information Processing Systems*, vol. 34, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds. Red Hook, NY, USA: Curran Assoc., Inc., 2021, pp. 29984–29994. [Online]. Available: https://proceedings.neurips.cc/paper/2021/file/fb674ca6672b0930e9b0d0c384d8b1f6-Paper.pdf

[15] L. Lai and S. J. Gershman, *Psychology of Learning and Motivation—Advances in Research and Theory*, vol. 74. Amsterdam, The Netherlands: Elsevier, Inc., May 2021.

[16] M. Federici, A. Datta, P. Zhang, N. Kushman, and Z. Akata, “Learning robust representations via multi-view information bottleneck,” in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–26.

[17] M. Igli et al., “Generalization in reinforcement learning with selective noise injection and information bottleneck,” in *Advances in Neural Information Processing Systems*, vol. 32. Red Hook, NY, USA: Curran, 2019, pp. 13956–13966.

[18] A. G. E. Collins and M. J. Frank, “How much of reinforcement learning is working memory, not reinforcement learning? A behavioral, computational, and neurogenetic analysis,” *Eur. J. Neurosci.*, vol. 35, no. 7, pp. 1024–1035, Apr. 2012.

[19] M. Phan, Y. A. Yaidkori, and J. Domke, “Thompson sampling and approximate inference,” in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran, 2019

[20] W. R. Thompson, “On the theory of apportionment,” *Amer. J. Math.*, vol. 57, no. 2, pp. 450–456, Apr. 1935.

[21] G. Kramer and S. A. Savari, “Communicating probability distributions,” *IEEE Trans. Inf. Theory*, vol. 53, no. 2, pp. 518–525, Feb. 2007.

[22] J. Lambert, “Observations analytiques,” in *Nouveaux Mémoires de l’Académie Royale Des Sciences et Belles-Lettres*, vol. 1. Berlin, Germany: C.F. Voss, 1770.

[23] S. Lloyd, “Least squares quantization in PCM,” *IEEE Trans. Inf. Theory*, vol. IT-28, no. 2, pp. 129–137, Mar. 1982.

[24] M. Song and S. Rajasekaran, “Fast k-means algorithms with constant approximation ratio,” in *Approximation, Randomization, and Computation (ISAAC)* (Lecture Notes in Computer Science, 3827), X. Deng and D. Z. Du, Eds. Berlin, Germany: Springer, 2005. [Online]. Available: https://doi.org/10.1007/11602613_102

[25] D. Russo and B. Van Roy, “Learning to optimize via posterior sampling,” *Math. Oper. Res.*, vol. 39, no. 4, pp. 1221–1243, Apr. 2014.

[26] C. Kalkanli and A. G. E. Collins, “Asymptotic convergence of Thompson sampling,” 2020, arXiv:2011.03917.

[27] D. Russo, “Simple Bayesian algorithms for best arm identification,” in *Proc. 29th Annua. Conf. Learn. Theory*, Jun. 2016, pp. 1417–1418. [Online]. Available: https://proceedings.mlr.press/v49/russo16.html

[28] I. Csiszar and G. Tusnady, “Information geometry and alternating minimization procedures,” *Stat. Decis. Supplement*, vol. 1, pp. 205–237, Jan. 1984.

[29] T. Lattimore and C. Szepesvári, *Bandit Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2020, doi: 10.1017/9781108571401.
Francesco Pase (Graduate Student Member, IEEE) received the B.S. degree in information engineering and the M.S. degree in telecommunications engineering from the University of Padua in 2017 and 2019, respectively, where he is currently pursuing the Ph.D. degree.

His master’s thesis was on graph representation, and it was developed at the EPFL in collaboration with Learn to Forecast, Lausanne, Switzerland. He spent the summer 2020, working as a Research Intern with InstaDeep Ltd., London, U.K., conducting research on deep reinforcement learning. He spent six months as a Visiting student with the Imperial College London in 2022, under the supervision of Prof. D. Gündüz. His current research interests are in the fields of information theory, machine learning, and wireless communications. Specifically, his current research investigates the intersections between information theory and distributed/reinforcement learning, with applications to wireless communications systems optimization.

Deniz Gündüz (Fellow, IEEE) received the B.S. degree in electrical and electronics engineering from the Middle East Technical University, Turkey, in 2002, and the M.S. and Ph.D. degrees in electrical engineering from the NYU Tandon School of Engineering (formerly Polytechnic University) in 2004 and 2007, respectively.

He is currently a Professor of Information Processing with the Electrical and Electronic Engineering Department, Imperial College London, U.K., where he also serves as the Deputy Head of the Intelligent Systems and Networks Group. He has held visiting/part-time positions with the University of Modena and Reggio Emilia, University of Padua, Princeton University, Stanford University, and CTTC. His research interests lie in the areas of communications and information theory, machine learning, and privacy. He is the recipient of the IEEE Communications Society—Communication Theory Technical Committee Early Achievement Award in 2017, Starting in 2016, and Consolidator in 2022 Grants of the European Research Council, and several best paper awards. He serves in editorial roles for the IEEE TRANSACTIONS ON INFORMATION THEORY, IEEE TRANSACTIONS ON COMMUNICATIONS, IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, and the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS. He is a Distinguished Lecturer for the IEEE Information Theory Society from 2020 to 2022.

Michele Zorzi (Fellow, IEEE) received the Laurea and Ph.D. degrees in electrical engineering from the University of Padua, Italy, in 1990 and 1994, respectively. From 1992 to 1993, he was on leave with the University of California at San Diego (UCSD). In 1993, he joined the Faculty of the Dipartimento di Elettronica e Informazione Politecnico di Milano, Italy. After spending three years with the Center for Wireless Communications, UCSD, he joined the School of Engineering, University of Ferrara, Italy, in 1998, where he became a Professor in 2000. Since November 2003, he has been with the Faculty of the Information Engineering Department, University of Padua. His current research interests include performance evaluation in mobile communications systems, the Internet of Things, cognitive communications and networking, 5G mmWave cellular systems, vehicular networks, and underwater communications and networks. He received several awards from the IEEE Communications Society, including the Best Tutorial Paper Award in 2008 and 2019, the Education Award in 2016, the Stephen O. Rice Best Paper Award in 2018, and the Joseph LoCicero Award for Exemplary Service to Publications in 2020. He has served the IEEE Communications Society as a Member-at-Large of the Board of Governors from 2009 to 2011 and from 2021 to 2023, as the Director of Education from 2014 to 2015, and as the Director of Journals from 2020 to 2021. He was the Editor-in-Chief of the IEEE Wireless Communications Magazine from 2003 to 2005, the IEEE TRANSACTIONS ON COMMUNICATIONS from 2008 to 2011, and the IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING from 2014 to 2018.