Curriculum Learning for Goal-Oriented Semantic Communications With a Common Language

Mohammad Karimzadeh Farshbafan, Walid Saad, Fellow, IEEE, and Merouane Debbah, Fellow, IEEE

Abstract—Goal-oriented semantic communication will be a pillar of next-generation wireless networks. Despite significant recent efforts in this area, most prior works are focused on specific data types (e.g., image or audio), and they ignore the goal and effectiveness aspects of semantic transmissions. In contrast, in this paper, a holistic goal-oriented semantic communication framework is proposed to enable a speaker and a listener to cooperatively execute a set of sequential tasks in a dynamic environment. A common language based on a hierarchical belief set is proposed to enable semantic communications between speaker and listener. The speaker, acting as an observer of the environment, utilizes the beliefs to transmit an initial description of its observation (called event) to the listener. The listener is then able to infer on the transmitted description and complete it by adding related beliefs to the transmitted beliefs of the speaker. As such, the listener reconstructs the observed event based on the completed description, and it then takes appropriate action in the environment based on the reconstructed event. An optimization problem is defined to determine the perfect and abstract description of the events while minimizing the various communication costs with constraints on the task execution time and belief efficiency. Then, a novel bottom-up curriculum learning (CL) framework based on reinforcement learning is proposed to solve the optimization problem and enable the speaker and listener to gradually identify the structure of the belief set and the perfect and abstract description of the events. Simulation results show that the proposed CL method outperforms classical RL and CL without inference scheme in terms of convergence time, task execution cost and time, reliability, and belief efficiency.

Index Terms—Goal-oriented semantic communication, semantic optimization, curriculum learning, reinforcement learning.

I. INTRODUCTION

GOAL-ORIENTED communication will be a pillar of next-generation wireless networks due to the radical increase in the autonomy level of emerging communication services in the Internet of Everything (IoE) [1], [2], [3]. For example, remotely controlling high-precision manufacturing in automated factories can be a meaningful application of goal-oriented communication. However, maximizing the bit accuracy, which is the main goal of current networks, is not sufficient for goal-oriented communication. Goal-oriented communication requires the network to consider the semantics (meanings) and effectiveness [1] of the data transmission. Hence, current metrics and algorithms for optimizing wireless networks must be redesigned to consider semantics and effectiveness. Deploying semantic communications for goal-oriented networks requires overcoming many challenges [4], [5] such as modeling semantic information, defining semantic and effectiveness metrics, and enabling transmitter-receiver cooperation for goal-oriented communications.

A. Prior Works

Several recent works investigated the principles of semantic communications [6], [7], [8]. In [6] and [7], the authors studied the information-theoretic challenges of semantic communications. In [6], the authors proposed a cognitive semantic communication system using a knowledge graph for semantic extraction (SE) and enhanced error correction. The work in [7] investigated the rate-distortion function for a semantic source communication system using a knowledge graph for semantic communications. In [6], the authors proposed a cognitive semantic communication system based on a universal adaptive transformer for text transmission. In [10], the authors proposed a semantic communication framework for text transmission using deep learning (DL). The work in [11] introduced a reinforcement learning (RL) solution to capture the meanings of transmitted information by learning semantic similarities. The main drawback of [9], [10], and [11] is that their models are only applicable to specific data types and, like [6], [7], [8], they do not consider the effectiveness of the semantics on the system’s goal and, thus, they cannot be generalized to broader goal-oriented semantic communication scenarios.
The work in [12] investigated the security aspect of semantic communication for Internet-of-Things (IoT) and introduced two novel metrics named semantic secrecy outage probability and detection failure probability for the security of semantic-based IoT systems. The authors in [13] proposed a unified task-based semantic communication system using a DL method, which simultaneously supports tasks with different data modalities, including image, text, and speech. The work in [14] introduced a model for implementing semantic communications to address the reliability and latency requirements for drone networks. In [15], the authors proposed a semantic-aware collaborative deep RL (DRL) to select the best subset of semantically relevant heterogeneous DRL agents across a wireless cellular network. The works in [14] and [15] can be considered as task-oriented semantic communication approaches. However, their solutions are only applicable to the specific applications that they considered and, thus, they cannot be generalized. Meanwhile, the authors in [16] proposed a semantic-native communication structure to extract the most effective semantics of the transmitter for the receiver. However, their approach cannot capture the effectiveness of the transmitted semantics for task execution purposes.

B. Contributions

The main contribution of this paper is a novel semantic communication system that leverages speaker and listener cooperation to perform goal-oriented communications while optimizing task execution time and cost, reliability, and belief efficiency. Our main contributions include:

- We introduce a holistic model for goal-oriented communications, which includes a speaker and a listener who wish to jointly execute a set of tasks in a dynamic environment. Each task is defined as a chain of multiple sequential events named task event chain (TEC), whereby each event observed by the speaker captures the state of the environment at each time. The speaker must inform the listener about the observed events, and the listener is responsible for taking a proper action regarding the observed event to steer the system to its goal.

- For event description and the communication between the speaker and listener, we assume the existence of a basic common language set called belief set with a hierarchical structure, in which the beliefs can be categorized in different levels, based on the amount of semantic information that they can convey and their data type. For event description, the speaker transmits an abstract description of the observed event using the belief set. Then, the listener infers and completes the description based on the received information.

- We formulate an optimization problem to minimize the costs incurred to speaker and listener with constraints on the task execution time and belief efficiency. To efficiently solve this problem, we develop a novel bottom-up curriculum learning (CL) [17] framework based on RL. The main objective of the proposed CL is to gradually identifying the hierarchical structure of the belief set and the perfect description of the events. We analytically derive sufficient conditions under which the proposed CL method can find a perfect description of each event.

- Simulation results show that our CL method outperforms classical RL and CL without inference in all metrics. In particular, our CL method compared to classical RL and CL without inference yields around 89% and 65.6% reduction in the task execution time, and 94% and 80.3% reduction in the task execution cost, 45% and 1.2% improvement in task execution reliability, and 2.5-fold and 15.5% improvement in belief efficiency.

The rest of the paper is organized as follows. The proposed goal-oriented semantic communication system is presented in Section II. Then, we present the proposed CL framework in Section III. Simulation results are analyzed in Section IV, and conclusions are drawn in Section V.

II. SYSTEM MODEL

Consider a speaker and listener who want to cooperatively execute a set of sequential tasks in an environment observed by the speaker. Each task consists of a chain of sequential events called TEC. At a given time, an event occurs in the environment and is observed by the speaker. This observed event captures the state of the environment, as perceived by the speaker at the observation time. The distortion level in the observed event of the speaker is tolerable in the sense that the observed event does not misinform about the state of the environment [16]. The speaker must describe its observed event to the listener, who is responsible for taking appropriate action in the environment according to the event so as to continue executing the associated task. The action taken by the listener and the environment’s dynamics will determine the next observed event. For semantic communication between speaker and listener, we consider the existence of a common language that is built using a set of beliefs between the speaker and the listener, known to both speaker and listener. For each observed event, there exists at least one representation in the belief set which can perfectly describe it. The perfect description of an event means that, by using the beliefs in such a description, the listener can successfully reconstruct the observed event. Precisely, a perfect description of each event must have sufficient information (through the beliefs) about the event and, thus, it can be used for explaining the event to a remote listener thus allowing it to successfully reconstruct the event.

To improve the communication resource (belief in our model) efficiency, the speaker and listener must cooperate to describe each event observed by the speaker. In other words, the speaker should transmit an abstract description of the observed event which is a subset of the perfect descriptor of the event, and the listener should infer what was sent by completing the received description, in a way to obtain a perfect descriptor of the observed event. The listener can complete the received description by adding new and appropriate beliefs from the belief set. We call these added beliefs the inferred description of the listener. Therefore, the final,
completed description of each event will be, a combination of the speaker’s transmitted description and the listener’s inferred description. The listener uses the completed description to reconstruct the observed event. Given its reconstructed event, the listener must take specific actions to steer the system in the right direction to complete the ongoing task. When the listener can successfully reconstruct the observed event, it can take a proper action in the environment regarding the observed event and immediately steer the ongoing task to its goal without delay. However, if the reconstructed event does not match the observed event, the listener will take an incorrect action thus delaying the ongoing task. The speaker and the listener can perceive the correctness of the transmitted and inferred description for each event, according to the average task execution time.

We divide time into different episodes, each of which is divided into equal time slots. At each episode \( m \), a specific task \( V_m \) is executed by the speaker-listener pair. At the beginning of slot \( n \) in episode \( m \), the speaker observes event \( e_{m,n}^s \) as the state of the environment. We define \( e_{m,n}^l \) as the reconstructed event of the listener, at time slot \( n \) of episode \( m \). Let \( B_{m,n}^s \), \( B_{m,n}^l \), and \( B_{m,n} \) be, respectively, the transmitted description of the speaker, the inferred description of the listener, and the completed description on the listener, at slot \( n \) of episode \( m \). Fig. 1 illustrates the proposed goal-oriented semantic communication system. Here, the speaker observes an event \( e_{m,n}^s \) and must describe it using \( B_{m,n}^s \). Then, the listener attempts to complete the received description based on its own inference and, then, reconstruct the observed event as \( e_{m,n}^l \). Finally, the listener takes an action in the environment based on the reconstructed event. It is worth noting that \( e_{m,n}^l \) is considered to be equal to \( e_{m,n}^s \) if and only if the listener’s action based on \( e_{m,n}^l \) is equal to the listener’s action based on \( e_{m,n}^s \). Then, the environment transitions to the next event, which depends on the reconstructed event of the listener (which in fact determines its taken action) and the current observed event of the speaker.

A. Events and Tasks Definition

Let \( \mathcal{E} = \{ \mathcal{E}_{\text{init}}, \mathcal{E}_{\text{mid}}, \mathcal{E}_{\text{fin}} \} \) be the set of all possible observed events where \( \mathcal{E}_{\text{init}}, \mathcal{E}_{\text{mid}}, \) and \( \mathcal{E}_{\text{fin}} \) are the set of initial, intermediary, and final events. Hence, the speaker starts each episode by observing an event from \( \mathcal{E}_{\text{init}} \) and, then, it observes multiple events from \( \mathcal{E}_{\text{mid}} \) and ends its monitoring of a given task by observing an event from \( \mathcal{E}_{\text{fin}} \). Therefore, we can write \( e_{m,1}^s \in \mathcal{E}_{\text{init}}, \{ e_{m,2}^s, \ldots, e_{m,T_{\text{max}}-1}^s \} \in \mathcal{E}_{\text{mid}}, \) and \( e_{m,T_{\text{max}}}^s \in \mathcal{E}_{\text{fin}} \), where \( X_m \) is the length of episode \( m \) quantified in terms of number of time slots, and it is equal to the execution time of the task performed during episode \( m \). We can now define a task based on the different events of the event set. Let \( T \) be the set of all \( \nu \) tasks. Then, we can define \( \mathcal{O}^T \) as the TEC of task \( T \), as follows:

\[
\mathcal{O}^T = \left\{ (e_1^T, e_2^T, \ldots, e_{T_{\text{max}}}^T) \mid e_1^T \in \mathcal{E}_{\text{init}}, (e_2^T, \ldots, e_{T_{\text{max}}-1}^T) \in \mathcal{E}_{\text{mid}}, e_{T_{\text{max}}}^T \in \mathcal{E}_{\text{fin}} \right\},
\]

where \( e_j^T \) is the \( j \)th event in the TEC of task \( T \) and \( T_{\text{max}} \) is the maximum length of task \( T \) in number of time slots, which is fixed for each task but can vary from one task to another. Due to the randomness of the environment’s dynamics, the length of task \( T \) will vary between different episodes. The environment’s dynamic includes all unknown factors in the environment for the speaker and listener that affect how the observed event of the speaker evolves from the current time slot to the next time slot (in addition to the action taken by the listener). These unknown factors of the environment are out of control of the speaker and listener and modeled by the transition model of the environment (matrices \( P \) and \( \hat{P} \)). Let \( L_T \) be a discrete random variable that captures the length of task \( T \) across episodes. \( L_T \) is distributed according to a probability mass function (PMF) \( f_T(L) = \text{Prob}(L_T = L) \), where \( L \in \{3, 4, \ldots, L_{\text{max}}\} \). The minimum value for \( L \) is 3 because we assume that each task has at least one event from each of \( \mathcal{E}_{\text{init}}, \mathcal{E}_{\text{mid}}, \) and \( \mathcal{E}_{\text{fin}} \). Also, we assume that the initial event of each task is fixed to differentiate among different tasks. One simple example of our proposed model is the remote control of high-precision manufacturing in automated factories in which there are multiple devices (e.g., cameras and robots) distributed in the factory to observe the state of the production line. The devices should transmit their captured data (equivalent to the event in our model) to a coordinator.
(e.g., an aggregator or a base station). The coordinator is responsible for taking control action in the factory based on the received information. Examples of tasks in this application include the coordination of a group of robots and devices to build a product, real-time monitoring of the production line, and resolution of a deficiency in the production line.

Next, we explain the relation between $L_T$ and the length of episode $m$, $X_m$, which is the execution time of the performed task in episode $m$. Assume that the executed task in episode $m$ is task $T$. Due to the probabilistic nature of $L_T$, when the listener takes a proper action for each observed event during episode $m$, then we expect to have $X_m = \mathbb{E}[L_T]$, where $\mathbb{E}[L_T]$ is the expectation of $L_T$. For taking a proper action, the listener must successfully reconstruct each observed event during episode $m$. However, the speaker and listener do not know the perfect descriptions of the events, and they need to learn them. During learning, the speaker’s transmitted descriptions can be imperfect or incomplete, and the listener cannot perfectly infer and complete the received description, and thus, the completed description may not perfectly describe the observed events. In this case, the listener cannot successfully reconstruct the observed events, the action taken by the listener will not be accurate, and the ongoing task will be delayed. When executing task $T$ in episode $m$, we have $e_{m,1}^T = e_1^T$. Assume that at the first slot of episode $m$, we have $e_{m,1}^L = e_{m,1}^S$, then $e_{m,2}^S \in \{ e_2^T, \ldots, e_{L_T}^T \}$. However, if the listener cannot successfully reconstruct the observed event in the second slot, and $e_{m,2}^L \neq e_{m,2}^S$, then, for the next observed event, we have $e_{m,3}^S \in \mathcal{E}$ and $e_{m,3}^S \notin \{ e_2^T, \ldots, e_{L_T}^T \}$, which means that the task is delayed. Hence, in this case, during learning, we anticipate that the execution time of the executed task in episode $m$, will be greater than its expected length and, thus, $X_m > \mathbb{E}[L_T]$.

### B. Common Language Belief Set and Semantic Communication Costs

We consider a basic common language set between the speaker and listener for describing the observed events of the speaker. Let $B$ be the set of $B$ beliefs of this language. Each belief $b \in B$ is a unique feature that can be used for event description. Thus, the belief set can be different for different applications. Intuitively, the belief set for a specific application can be determined by performing a semantic extraction procedure that is beyond the scope of this work. We assume the speaker and listener have same perception and, thus, the beliefs in other levels convey different levels, based on their amount of semantic information, and their data type.\(^2\)

Here, we considered simple relations to combine the beliefs for event description. To model complicated relations between the beliefs to combine them for event description, we can use techniques such as factor graphs [22]. In a hierarchical structure, the beliefs in the first level convey general information about the events and, thus, the amount of semantic information conveyed by the beliefs in the first level is less than that of the beliefs in other levels. By going deeper in the hierarchical structure, the amount of semantic information conveyed by the beliefs increases as deeper beliefs contain more detailed information about the events. According to this structure, the belief set can be defined as follows:

\[
\mathcal{B} = \{ B_1, B_2, \ldots, B_K \}, \\
B_k = \{ b_{k,1}, \ldots, b_{k,B_k} \}, \quad B_k \cap B_{k'} = \emptyset \quad \forall 1 \leq k, k' \leq K,
\]

\(^2\)This hierarchical structure can also be seen as a way to reflect the causal relations between different beliefs [3].
where \( B_k \) is the belief set of level \( k \) of the hierarchy, \( B_k \) is the number of beliefs in level \( k \), and \( K \) is the total number of levels of the belief set hierarchy. The total number of beliefs will be \( B = \sum_{k=1}^{K} B_k \). The intersection of the belief sets of two different levels will be empty since each belief can only belong one level of the hierarchy. Let \( C_{k,u}^S \) be the cost incurred by the speaker for transmitting belief \( u \) of level \( k \) and \( C_{k,u}^L \) be the cost incurred by the listener for inferring and adding belief \( u \) of level \( k \) to a received description. The inference cost can be viewed as the computational cost required for the listener to add beliefs and compute the completed description. The cost of each belief is determined based on the amount of the semantic information that it can convey. As such, we consider \( C_{k,u}^S \) and \( C_{k,u}^L \) to be increasing functions of semantic information. The increasing costs with the hierarchy are further justified by the fact the more detailed beliefs require more bits for transmission in a classic communication system.

According to the definition of the belief set, which includes different features of the events, there could be multiple subsets of the belief set that can perfectly describe a given event. Since the belief set has a hierarchical structure, then, the perfect descriptor of each event is also hierarchical. Precisely, a perfect description of an event only contains one belief from each level of the hierarchy. Let \( B_e^P \) be the set of the perfect descriptors of event \( e \in E \), defined as follows:

\[
B_e^P = \{ B_h \subset B \mid \text{Listener can successfully reconstruct } e \text{ based on } B_h \},
\]

\[
B_h = \{ b_{1,m_1}, \ldots, b_{K_h,u_K_h} \}, \quad b_{k,u_k} \in B_k, \quad 2 \leq K_h \leq K, \quad (3)
\]

where \( B_h \) is a perfect, hierarchical descriptor of event \( e \), \( K_h \) is the number of the hierarchy levels, which is different for each event, and \( b_{k,u_k} \) is a belief in the level \( k \) of the hierarchy.

Fig. 2 shows an example of the hierarchical belief set, \( B \), that includes four levels, where there are 3, 4, 5, and 6 beliefs in the first, second, third, and fourth levels, respectively. The red edges represent incompatible combinations (i.e., not useful for event description) between the beliefs of the first and second levels. Meanwhile, the black, green, and blue edges represent several possible compatible combinations between the beliefs of different levels. Also, Fig. 2 illustrates the perfect description of two events \( e \in E \) and \( e' \in E \). Event \( e \) has two perfect descriptors, as captured by the green edges. The first and second perfect descriptors of \( e \) include two and three beliefs, respectively. Event \( e' \) has one perfect descriptor, which is shown using blue edges and includes four beliefs. Here, in contrast to [18], adding more beliefs to a perfect description of an event \( e \) can make that description imperfect for describing \( e \). For example, in Fig. 2, \( \{ b_{1,1}, b_{2,1} \} \) is considered as a perfect descriptor of event \( e \). In contrast, for this same figure, description \( \{ b_{1,1}, b_{2,1}, b_{k,u} \} \) for any values of \( k \) and \( u \) is not necessarily a perfect descriptor of \( e \).

We next explain the cost of each event reconstruction by the speaker and listener. Here, we have a dual-objective optimization problem, including minimizing the transmission cost of the speaker and the inference cost of the listener. This dual-objective optimization can be modeled in a variety of ways. One popular approach is to use scalarization [23] which we adopt here. The scalarization method incorporates multi-objective functions into a scalar fitness function [24], and it is known as a meaningful way to represent such problems. Let \( C_{m,n} \) be the total cost of the completed description used at slot \( n \) of episode \( m \), which can be computed as follows:

\[
C_{m,n}^T = \alpha \cdot C_{m,n}^S + (1 - \alpha) \cdot C_{m,n}^L, \quad (4)
\]

\[
C_{m,n}^S = \sum_{k=1}^{K} \sum_{u=1}^{B_k} C_{k,u}^S \cdot x_{m,n,k,u}^S, \quad (5)
\]

\[
C_{m,n}^L = \sum_{k=1}^{K} \sum_{u=1}^{B_k} C_{k,u}^L \cdot x_{m,n,k,u}^L, \quad (6)
\]

\[
x_{m,n,k,u}^S = \begin{cases} 
1, & b_{k,u} \in B_{m,n}^S, \\
0, & b_{k,u} \in B_{m,n}^L, 
\end{cases} \quad (7)
\]

\[
x_{m,n,k,u}^L = \begin{cases} 
1, & b_{k,u} \in B_{m,n}^L, \\
0, & b_{k,u} \in B_{m,n}^S, 
\end{cases} \quad (8)
\]

Here, \( x_{m,n,k,u}^S \) is a binary decision variable that indicates whether belief \( u \) of level \( k \) of the hierarchy is used in \( B_{m,n}^S \). Similarly, \( x_{m,n,k,u}^L \) is a binary decision variable that indicates...
whether belief \( u \) of level \( k \) of the hierarchy is used in \( B_{m,n}^L \).

\( C_{m,n}^S \) is the incurred cost of the speaker for transmitting \( B_{m,n}^S \) and \( C_{m,n}^L \) is the incurred cost of the listener for inferring \( B_{m,n}^L \) at slot \( n \) of episode \( m \). \( \alpha \) is a design parameter for making a compromise between the costs of the speaker and the listener. Here, we assume that the listener can only complete the speaker’s transmitted description by adding new beliefs based on its own inference. In other words, the listener does not ignore any beliefs transmitted by the speaker, and it uses all of them for event reconstruction.

**C. Environment State Evolution**

The environment’s state at each time slot, captured by the observed event of the speaker \( e_{m,n}^S \), is dependent on the environment’s state and the action taken by the listener in the environment, at the previous slot. The speaker describes its observation by transmitting \( B_{m,n}^S \) to the listener. Then, the listener infers and completes \( B_{m,n}^S \) by adding new beliefs, captured by \( B_{m,n}^L \). Finally, the listener reconstructs the observed event of the speaker, based on the completed description. Thus, the reconstructed event \( e_{m,n}^L \) by the listener at each time slot, can be computed as follows:

\[
e_{m,n}^L = \left\{ \begin{array}{ll}
    e_{m,n}^S, & B_{m,n} \in B_e^L, \\
    e \in \{ E \setminus e_{m,n}^S \}, & B_{m,n} \notin B_e^L,
\end{array} \right.
\]

where \( B_{m,n} = \{ B_{m,n}^S, B_{m,n}^L \} \) is the set of beliefs at \( m \) and \( e_{m,n}^S \) is the set of all events except \( e_{m,n}^S \). To capture the environment evolution from one slot to another, we must define the transition probabilities between different events as functions of the reconstructed event by the listener, \( e_{m,n}^L \). Let \( p_{j_1,j_2} \) be the transition probability from \( e_{m,n}^S = e_{j_1} \) to \( e_{m,n+1}^S = e_{j_2} \), given by:

\[
p_{j_1,j_2} = \Pr(e_{m,n}^S = e_{j_1} \rightarrow e_{m,n+1}^S = e_{j_2} | e_{m,n}^L) = \left\{ \begin{array}{ll}
    P(j_1,j_2) e_{m,n}^L = e_{j_1}, \\
    \tilde{P}(j_1,j_2) e_{m,n}^L \neq e_{j_1},
\end{array} \right.
\]

where \( P \) is the transition probability matrix of the environment’s events when the listener can take proper action regarding the observed event. We expect \( P \) to be a sparse matrix because, when the listener can take proper action regarding the state of the environment, the next state is one of the events of the ongoing task’s TEC. \( \tilde{P} \) is the transition probability matrix when the speaker does not take the right action regarding the observed event because the listener cannot successfully reconstruct the observed event. Hence, we expect \( \tilde{P} \) to be a random matrix that has more nonzero elements compared to \( P \). This is due to the fact that, when the listener cannot take a proper action for a given state of the environment, the next state will have higher randomness. Precisely, we assume that, when the listener cannot take proper action, the probability distribution of the next observed event is completely random. In such scenarios, the transition probability can be uniform over all possible events. In this case, there is no difference between the taken actions in terms of the transition probability matrix in the second case of (10) and all of the improper actions that lead to a random behavior in terms of the transition probability matrix.

**D. Task Execution Metrics**

We consider three key performance metrics for our semantic communication system:

1) **Task Execution Cost**: We know that there can be multiple subsets of the belief set which can perfectly describe each event \( e \), as captured by \( B_e^P \) and defined in (3). To differentiate between the perfect descriptors of an event, in (4), we defined the cost of the used descriptor at each time slot. We now compute the cost \( C_m \) of the executed task in episode \( m \). This cost will be a function of the incurred cost of the speaker and listener during episode \( m \), as follows:

\[
C_m = \alpha \cdot C_{m}^{S} + (1 - \alpha) \cdot C_{m}^{L} = \alpha \cdot \sum_{m=1}^{X_m} C_{m}^{S} + X_m \cdot \sum_{n=1}^{K} B_{m,n} \cdot t_{m,n,k,u} + (1 - \alpha) \cdot \sum_{m=1}^{X_m} \sum_{u=1}^{K} B_{m,k,u} \cdot t_{m,n,k,u}.
\]

In our model, the listener is a remote node and it is not the observer of the environment. For example, the listener can be a base station or an aggregator. Therefore, we can assume that the listener is computationally capable, and it possesses a wide range of inference capabilities.

2) **Task Execution Time**: The length of each task \( T \) is a random variable. Thus, for each task \( T \), we should consider the mean task execution time over different episodes, \( T_\Gamma \). We consider the length of each episode, \( X_m \), as the task execution time for the task executed during episode \( m \). To compute \( T_\Gamma \), we should calculate the average of the episode length \( T_\Gamma = \mathbb{E}[X_m | V_m = T] \), during which task \( T \) is executed, where \( V_m \) is the index of the task executed in episode \( m \) and the expectation is over different episodes. The speaker and listener must learn the perfect descriptors of each event and, during the learning process, they cannot perfectly describe the observed event hence delaying the tasks. Thus, during learning, we have \( T_\Gamma > \mathbb{E}[L_T] \), where \( \mathbb{E}[L_T] \) is the expected length of task \( T \) calculated over \( f_T \). However, as the learning evolves over time and the speaker and listener gradually learn the perfect description of the events, we expect that \( \lim_{M \to \infty} [T_\Gamma] \to \mathbb{E}[L_T] \), where \( M \) is the number of episodes.

3) **Belief Efficiency**: In semantic communications, the main resource of the speaker is the beliefs, which can be considered as the equivalent of bandwidth in classical communication. Spectral efficiency is the main metric to optimize bandwidth usage in classical communication. We now optimize the belief usage by defining a similar metric. Let \( \mathbb{U} \) be the belief efficiency. Let \( U_{b,T} \) and \( U_{b,m} \) be the belief efficiency for task \( T \) and belief efficiency of episode \( m \), respectively. To compute \( U_{b,m} \), we should consider the total number (across all time slots) of beliefs that the speaker uses to describe the observed events during episode \( m \), and inverse it, as follows:

\[
U_{b,m} = \frac{1}{\sum_{n=1}^{X_m} W_{m,n}^S}, \quad W_{m,n}^S = \sum_{k=1}^{K} \sum_{u=1}^{B_{m,n,k,u}} f_{m,n,k,u}.
\]
computing the belief efficiency, we only consider the beliefs sent by the speaker while the inferred beliefs of the listener are ignored because they are not transmitted over a wireless channel. To compute \( U_{b,T} \), we should calculate the expectation of \( U_{b,m} \) in the episodes during which task \( T \) is executed, i.e., \( U_{b,T} = \mathbb{E}[U_{b,m} | V_m = T] \). This is because the required number of the beliefs for executing a task randomly vary across episodes since the number of events in each task’s TEC is a random variable. Finally, the belief efficiency \( U_b = \mathbb{E}[U_{b,T}] \) will be the expectation of \( U_{b,T} \) over different tasks.

III. PROBLEM FORMULATION AND PROPOSED CL ALGORITHM

A. Problem Formulation

Given the metrics defined in Section II, we can now formally pose the problem of determining the perfect and abstract description of the events, as follows:

\[
\begin{align*}
\min_{\mathbf{X}_{m,n}^S, \mathbf{X}_{m,n}^L} & \quad \mathbb{E}[N_c(t) | \mathbf{X}_{m,n}^S, \mathbf{X}_{m,n}^L] \\
\text{s. t.} & \quad \lim_{M \to \infty} \mathbb{E}[U_{b,m}] \geq U_{b,\text{min}}, \\
& \quad \sum_{u=1}^{B_k} \left[ \mathbf{X}_{m,n}^S[k,u] + \mathbf{X}_{m,n}^L[k,u] \right] \leq 1, \\
& \quad \sum_{u=1}^{B_k} \left[ \mathbf{X}_{m,n}^S[k+1,u] + \mathbf{X}_{m,n}^L[k+1,u] \right] \leq 1, \\
& \quad \sum_{u=1}^{B_k} \left[ \mathbf{X}_{m,n}^S[k,u] + \mathbf{X}_{m,n}^L[k,u] \right] \leq 1,
\end{align*}
\]

where the objective function is the discounted sum of the total costs of both the speaker and listener during an infinite horizon and \( \gamma \) is a discount factor. \( \mathbf{X}_{m,n}^S = [x_{m,n,k,u}]_{K \times B_K} \) and \( \mathbf{X}_{m,n}^L = [x_{m,n,k,u}]_{K \times B_K} \) are, respectively, the matrix format decision variables of the speaker and listener at slot \( n \) of episode \( m \), over different values \( k \) and \( u \). \( A \) is the set of all possible binary matrices of size \( K \times B_K \). \( C_m \) is a function of \( \mathbf{X}_{m,n}^S \) and \( \mathbf{X}_{m,n}^L \) according to (11), but, for notational simplicity, we do not show this dependence in (13a). (13b) is used to minimize the task execution time. More precisely, this constraint motivates the speaker and listener to find the perfect description of each event. (13c) is used to optimize the belief efficiency. Note that (13b) and (13c) depend on \( \mathbf{X}_{m,n}^S \) and \( \mathbf{X}_{m,n}^L \), since \( \mathbf{X}_{m,n}^S \) and \( \mathbf{X}_{m,n}^L \) can be determined using \( \mathbf{X}_{m,n}^S \) and \( \mathbf{X}_{m,n}^L \), respectively. \( \mathbf{X}_{m,n}^S \) and \( \mathbf{X}_{m,n}^L \) as the variables of problem (13) cooperatively construct the completed description, that is used by the listener for decision-making. The optimization problem (13) is solved if the speaker and listener can determine the transmitted and inferred descriptions in a way that the resulting completed description is the perfect and abstract description of each event. Therefore, solving problem (13) is equivalent to finding the perfect and abstract description of each event by the pair of the speaker and listener. (13d) indicates that a belief can only be transmitted by the speaker or can be inferred by the listener. (13e) indicates that, at each time slot, only one belief from each level of the hierarchy can be used. (13f) captures the fact that the existence of a belief from level \( k \) is necessary for the presence of a belief from level \( k+1 \).

Since the environment’s dynamic nature affects the observed event’s transition, the speaker and listener need to interact with the environment to experience different events and descriptions. We first note that classical stochastic optimization techniques cannot be used to solve (13) because they require knowing the transition functions of the system, which are unknown. Second, model-based RL methods also cannot be used because the speaker and listener are unaware of the transition model of the environment (matrices \( P \) and \( \tilde{P} \)) [25]. Finally, model-free methods will not be efficient due to the specific characteristics of our problem including (a) the massive size of action space, \( 2^B \), which is exponential in the number of beliefs, and (b) the task execution structure of the problem that exhibits a sparse reward signal.

These challenges of using classical RL motivate us to introduce a curriculum framework [17] combined with multi-agent RL to solve problem (13). The main goal of the speaker and listener is to cooperatively find the perfect and abstract description of each event to solve (13). Here, the main responsibility of the speaker is finding useful beliefs for the description of each event, and the listener wants to find useful beliefs to complete the speaker’s transmitted description. However, the listener is completely blind about the usefulness of each belief for the different events, and, thus, they must gain this knowledge by experiencing various tasks with different events. However, the random exploration process of classical RL requires a long training to converge because of the massive action space size. In such problems, using a curriculum framework can be a suitable solution. The main idea behind CL is to design simple tasks based on a main problem which requires long training, perform these simple tasks, and use their gained experience to solve the main problem more efficiently and more quickly [17]. Here, we propose a CL method that enables the speaker and listener to identify, step-by-step, the hierarchical structure of the belief set. In each step, the speaker and listener identify a portion of the hierarchical structure and learn the perfect description of the events that can be described using the identified portion of the belief set. Note that the identified portion of the belief set in each step is used to initialize the algorithm in the next step.

B. Bottom-to-Up Curriculum Learning Algorithm

The hierarchical structure of the belief set plays a key role in determining the perfect description of each event. However, the speaker and listener are unaware of the exact hierarchical structure of the belief set. An interesting solution here is to gradually identify the hierarchical structure of the belief set and the perfect descriptions of the events that can be perfectly described using the identified portion of the hierarchical structure. We thus propose a linear sequence-based task-level CL, defined by a sequence of problems \( \{z_1, z_2, \ldots, z_l\} \) [17]. Here, \( z_l \) refers to the problem that should be solved in CL step \( l \) and whose gained knowledge will then be used to initialize and solve \( z_{l+1} \). The term “task” is used here in
the context of the CL paradigm, and it is not related to our system’s task. In $z_l$, the CL method determines the structure of the first $l + 1$ levels of the belief set hierarchy as well as the perfect descriptors of events using these $l + 1$ levels of the belief set. We consider two outputs for each CL step. The first one is related to the perfect description of each event and the second is related to the hierarchical belief structure. Note that the minimum number of beliefs for having a perfect description of an event is two. Thus, during the first CL step, the speaker and listener will learn the structure of the first and second hierarchy levels. Let $B_{z_1,e}^\text{Out} = \left\{ (b_{i_1}, b_{i_2}) \mid b_{i_1}, b_{i_2} \in B_{z_1,b_1}, (b_{i_1}, b_{i_2}) \in B_{z_1}^P \right\}$, be the output of the first CL step for the perfect description of event $e$, where $B_{z_1}$ and $B_2$ are the belief sets of the first and second hierarchy levels.

For some event $e'$, we may have $B_{z_1,e'}^\text{Out} = \emptyset$, if this event does not have a perfect descriptor with 2 beliefs. In other words, the first CL step determines the perfect description of events that have $B_h \in B_{z_1}^P$ with $K_h = 2$. Now, let $B_{z_1,H}^\text{Out}$ be the second output of $z_1$ that is related to the hierarchical structure of the belief set, as follows:

$$B_{z_1,H}^\text{Out} = \left\{ (b_{i_1}, b_{i_2}) \mid \forall e \in E : (b_{i_1}, b_{i_2}) \in B_{z_1,e}^\text{Out} \right\}. \quad (14)$$

In the second CL step $z_2$, the speaker and listener want to find the beliefs in the third level of the belief set hierarchy while also determining the compatible combination between the beliefs in the first and second level of the belief set hierarchy and the beliefs in the third level of the hierarchy. For this purpose, they use the output of the first CL step as the compatible combinations between the first and second levels of the belief set hierarchy. Let $B_{z_2,e}^\text{Out}$ be the output of the second CL step that is related to perfect possible descriptions of the events and defined as follows:

$$B_{z_2,e}^\text{Out} = \left\{ (b_{i_1}, b_{i_2}, b_{i_3}) \mid (b_{i_1}, b_{i_2}), (b_{i_1}, b_{i_3}), (b_{i_2}, b_{i_3}) \in B_{z_1,H}^\text{Out} \right\}, \quad (15)$$

where $B_{z_1,H}^\text{Out}$ is defined in (14) and $B_3$ is the belief set of the third level of the hierarchy. As seen in (15), the gained knowledge $B_{z_2,H}^\text{Out}$ at the first step is directly used for initializing $z_2$, which showcases how we leverage the CL concept in our approach. Let $B_{z_2,H}^\text{Out}$ be the second output of $z_2$, which is related to the hierarchical structure of the belief set, which is $B_{z_2,H}^\text{Out} = \left\{ (b_{i_1}, b_{i_2}, b_{i_3}) \mid \exists e \in E : (b_{i_1}, b_{i_2}, b_{i_3}) \in B_{z_2,e}^\text{Out} \right\}$. By following the procedure used for defining the outputs of $z_1$ and $z_2$, we can determine the two outputs of CL step $z_l$ for $l \geq K - 1$, as follows:

$$B_{z_l,e}^\text{Out} = \left\{ (b_{i_1}, \ldots, b_{i_{l+1}}) \mid (b_{i_1}, \ldots, b_{i_l}) \in B_{z_{l-1},H}^\text{Out} \right\}, \quad (16)$$

$$B_{z_l,H}^\text{Out} = \left\{ (b_{i_1}, \ldots, b_{i_{l+1}}) \mid \exists e \in E : (b_{i_1}, \ldots, b_{i_{l+1}}) \in B_{z_l,e}^\text{Out} \right\}, \quad (17)$$

where $B_{z_l,H}^\text{Out}$ is the set of compatible combinations between the beliefs in the first $l$ levels of the hierarchy and the beliefs in level $l + 1$. Also, $B_{z_l,e}^\text{Out}$ is the set of perfect descriptors of event $e$ that have $(l + 1)$ beliefs. The main objective of step $l$ of the CL algorithm is determining $B_{z_l,e}^\text{Out}$ and $B_{z_l,H}^\text{Out}$. For this purpose, we use the output of the CL algorithm at step $l - 1$, $B_{z_{l-1},H}^\text{Out}$, which is the hierarchical structure of the first $l$ levels of the belief set. The CL method proceeds until we have $l = K - 1$, at which point the algorithm determines the compatible combinations in the last belief hierarchy level and the $K$-beliefs perfect descriptors of each event $e$. Fig. 3 shows an example of the proposed algorithm for determining the structure of a belief set that has 4 belief levels. Fig. 3 shows how the output $B_{z_l,H}^\text{Out}$ of each step of the CL is used in the next step.

**C. Reinforcement Learning at Each CL Step**

For solving each CL step, we now introduce a multi-agent RL method (the agents are the speaker and listener) using Q-Learning. In each step of CL, a multi-agent RL problem is defined. As the CL algorithm evolves, the defined multi-agent RL problem in each CL step becomes more difficult to solve. However, our CL approach initializes the multi-agent RL problem of each CL step based on the output of the previous step thus reducing this difficulty and making the multi-agent RL problem tractable. The CL method continues until the main problem is completely and tractably solved. For our system, in each event, we want to find a single perfect description, which minimizes the incurred cost of the speaker and listener, minimizes task execution time, and maximizes the belief efficiency. The speaker and listener will cooperate to solve our multi-agent RL problem, in each step. Here, the speaker will learn how to transmit an abstract description, i.e., a subset of the perfect description of the observed event. Then, the listener learns how to infer the beliefs based on the speaker’s transmitted description such that the completed description is a perfect description of the observed event. In the first CL step, $z_1$, the speaker and listener are unaware of the hierarchical structure of the belief set. In $z_1$, where $l \geq 2$, the speaker and listener are aware of the first $l$ hierarchy levels. Hence, we first introduce the learning method for the first CL step. Then, we present the learning method for any step $l \geq 2$ of the CL method.

1) **Learning During the First CL Step:** In the first CL step, we define the state space and action space of the speaker as follows:

$$\Omega_\Lambda^{S,z_1} = \left\{ e \mid e \in E \right\}, \quad \Omega_\Lambda^{S,z_1} = \left\{ b_i \mid b_i \in B \right\}, \quad (18)$$

where $\Omega_\Lambda^{S,z_1}$ and $\Omega_\Lambda^{S,z_1}$ are the state space and action space of the speaker in the first CL step. The action space is all possible subsets of $B$ with cardinality 1. We define $\lambda^{S,m,n,z_1}_i \in \Omega_\Lambda^{S,z_1}$ and $a^{S,m,n,z_1}_i \in \Omega_\Lambda^{S,z_1}$ as the state and action of the speaker at slot $n$ of episode $m$, in the first CL step. The state space and action spaces of the listener in the first CL step are defined as follows:

$$\Omega_\Lambda^{L,z_1} = \left\{ b_i \mid b_i \in B \right\}, \quad \Omega_\Lambda^{L,z_1}(\lambda^{L,m,n,z_1}_i) = \left\{ b_i \mid b_i \in B \setminus \lambda^{L,m,n,z_1}_i \right\}, \quad (19)$$
where $\Omega^L_{m,z_1}$ is the state space of the listener in the first step of CL and $\lambda^L_{m,n,z_1} \in \Omega^L_{A;z_1}$ represents the state of the listener at slot $n$ of episode $m$, in the CL first step. Also, $\Omega^L_{A;z_1}(\lambda^L_{m,n,z_1})$ is the action space for state $\lambda^L_{m,n,z_1}$. For the listener, we should define a specific action space for each state in the CL first step. For state $\lambda^L_{m,n,z_1}$, the action space is all possible subsets of $\{B \setminus \lambda^L_{m,n,z_1}\}$ with cardinality 1. We define $a^L_{m,n,z_1} \in \Omega^L_{A;z_1}(\lambda^L_{m,n,z_1})$ as the action of the listener at slot $n$ of episode $m$, in the first CL step. We can now connect the state and action of the speaker and listener to the variables of the original problem (13), as follows:

$$3\lambda^S_{m,n,z_1} = e^S_{m,n}, \quad B^S_{m,n} = a^S_{m,n,z_1}, \quad (20)$$

$$\lambda^L_{m,n,z_1} = B^L_{m,n}, \quad B^L_{m,n} = a^L_{m,n,z_1}, \quad (21)$$

where $e^S_{m,n}$ is the observed event of the speaker. $B^S_{m,n}$ and $B^L_{m,n}$ are the transmitted and inferred description of the speaker and listener, respectively. As seen in (21), the RL state of the listener is the transmitted description of the speaker because the listener is not aware of the observed event and, thus, it should make decisions based on the speaker’s transmitted description. The completed description, $B_{m,n}$, used by the listener for reconstructing the observed event, will be $B_{m,n} = \{B^S_{m,n}, B^L_{m,n}\} = \{a^S_{m,n,z_1}, a^L_{m,n,z_1}\}$.

The reward function of the speaker at the first CL step is $R^S_{z_1}(\lambda^S_{m,n,z_1}, a^S_{m,n,z_1}, \lambda^L_{m,n,z_1}, a^L_{m,n,z_1})$ as the reward gained by the speaker for taking action $a^S_{m,n,z_1}$ at state $\lambda^S_{m,n,z_1}$, while the listener takes action $a^L_{m,n,z_1}$ at state $\lambda^L_{m,n,z_1}$. To define the reward, we recall that the speaker’s goal is to determine the perfect description of each event that simultaneously optimizes the belief transmission cost, task execution time, and belief efficiency. Thus, we should consider how each action can affect the aforementioned metrics for the speaker. The speaker’s incurred cost for taken action and the effect of the taken action on the belief efficiency can be easily derived. However, the effect of each action on the task execution procedure is determined based on the current and next observed events. Thus, we have:

$$R^S_{z_1}(\lambda^S_{m,n,z_1}, a^S_{m,n,z_1}, \lambda^L_{m,n,z_1}, a^L_{m,n,z_1}) = \begin{cases} -C^S_{m,n} - W^S_{m,n} + R_T, & \lambda^S_{m,n+1,z_1} \in \mathcal{E}_{\text{fin}} \\ -C^S_{m,n} - W^S_{m,n} - C_T, & \lambda^S_{m,n+1,z_1} \in \mathcal{E}_{\text{init}} \\ -C^S_{m,n} - W^S_{m,n}, & \lambda^S_{m,n+1,z_1} \in \mathcal{E}_{\text{mid}}, \end{cases} \quad (22)$$

where $C^S_{m,n}$ and $W^S_{m,n}$ are defined in (6) and (12), respectively. $T$ is the executed task in episode $m$. $R_T$ is the reward of executing task $T$ and $C_T$ is the cost of delaying task $T$. $\lambda^S_{m,n+1,z_1}$ is the speaker’s next state, which is equal to the observed event of the speaker at slot $n + 1$, which is determined using (10). The reward function in (22) is defined in a way that solve (13). Specifically, $C^S_{m,n}$, which contributes to the objective function of (13), is considered as a cost in the reward function to minimize the transmission cost of the speaker. $W^S_{m,n}$ which contributes to (13c) is considered as a cost to maximize the belief efficiency according to (12). Finally, $R_T$ as the reward of executing task $T$, and $C_T$ as the cost of delaying task $T$ are used in the RL reward function to satisfy the task execution time constraint in (13b).

Similarly, the listener’s reward function at the first CL step is defined as $R^L_{z_1}(\lambda^L_{m,n,z_1}, a^L_{m,n,z_1}, \lambda^S_{m,n,z_1}, a^S_{m,n,z_1})$ as the reward gained by the listener for taking action $a^L_{m,n,z_1}$ at state $\lambda^L_{m,n,z_1}$, while the listener takes action $a^S_{m,n,z_1}$ at state $\lambda^S_{m,n,z_1}$. To define the reward, we recall that the speaker’s goal is to determine the perfect description of each event that simultaneously optimizes the belief transmission cost, task execution time, and belief efficiency. Thus, we should consider how each action can affect the aforementioned metrics for the speaker. The speaker’s incurred cost for taken action and the effect of the taken action on the belief efficiency can be easily derived. However, the effect of each action on the task execution procedure is determined based on the current and next observed events. Thus, we have:

$$R^L_{z_1}(\lambda^L_{m,n,z_1}, a^L_{m,n,z_1}, \lambda^S_{m,n,z_1}, a^S_{m,n,z_1}) = \begin{cases} -C^L_{m,n} + R_T, & \lambda^S_{m,n+1,z_1} \in \mathcal{E}_{\text{fin}} \\ -C^L_{m,n} - C_T, & \lambda^S_{m,n+1,z_1} \in \mathcal{E}_{\text{init}} \\ -C^L_{m,n}, & \lambda^S_{m,n+1,z_1} \in \mathcal{E}_{\text{mid}}, \end{cases} \quad (23)$$

where $C^L_{m,n}$ is the listener’s incurred cost for the inferred description at each slot, as per (6). Similar to speaker’s case, the reward function in (23) is defined in a way to solve problem (13). The completed description in each time slot, $B_{m,n}$, can include multiple beliefs from a single level. However, such a description cannot perfectly describe an event. Therefore, the reward of such descriptions would be calculated using the second or third case of (20) and (21). On the other hand, the reward of the perfect descriptions would be calculated using the first or second case of (20) and (21). Hence, the value of the perfect descriptions in each state at the end of the learning time will be greater than the imperfect descriptions.

2) Learning at CL Step $l$: For CL step $l \geq 2$, the state and action spaces are as follows:

$$\Omega^S_{A;z_1} = \{e | e \in \mathcal{E}\},$$

$$\Omega^S_{A;z_1} = \{(b_{i_1}, \ldots, b_{i_l}) | (b_{i_1}, \ldots, b_{i_l}) \in B^{Q_{l-1},H}\}, \quad (24)$$

Fig. 3. An example of how the proposed CL method determines the structure of the belief set.
where $\Omega_{\Lambda,z_i}^S$ and $\Omega_{\Lambda,z_i}^A$ are the state and action space of the speaker in CL step $l$. We define $\Lambda_{m,n,z_i}^S \in \Omega_{\Lambda,z_i}^S$ and $a_{m,n,z_i}^S \in \Omega_{A,z_i}^S$ as the state and action of the speaker at slot $n$ of episode $m$ in CL step $l$. According to this definition, the state space is the same as the first CL step. However, we define the action space based on the output of the previous CL step. More precisely, in CL step $l$, the speaker can only use the compatible combinations between the beliefs, $B_{z_{l-1},H}^l$, in the first $l$ levels of the hierarchical structure of the belief set for the description of the observed events. Now, we define the state space and action space of the listener in CL step $l$, as follows:

$$\Omega_{\Lambda,z_i}^L = \left\{ (b_1, \ldots, b_{l_i}) \mid (b_1, \ldots, b_{l_i}) \in B_{z_{l-1},H}^l \right\}, \quad (25)$$

$$\Omega_{A,z_i}^L = \left\{ b_i \mid b_i \in \left\{ B \setminus \bigcup_{k=1}^{l} B_k \right\} \right\}, \quad \bigcup_{k=1}^{l} B_k = \bigcup_{b_i \in B_{z_{l-1},H}^l} B_{l_i}, \quad (26)$$

where $\Omega_{\Lambda,z_i}^L$ and $\Omega_{A,z_i}^L$ are the state and action spaces of the listener at step $l$ of the CL method. We define $\Lambda_{m,z_i}^L \in \Omega_{\Lambda,z_i}^L$ and $a_{m,z_i}^L \in \Omega_{A,z_i}^L$ as the state and action of the listener at slot $n$ of episode $m$. For defining the action set of the listener at CL step $l \geq 2$, we consider the fact that the speaker and listener are aware of the first $l$ levels of the hierarchical structure of the belief set, based on the output of the previous step of CL ($l-1$). We have the same connection between the state and action of the speaker and listener to the variables of the original problem at each time slot of step $l$ of the CL, as defined in (20) and (21). Also, the reward functions of the speaker and listener are defined in a similar way to the first step, as per (22) and (23).

### D. Proposed Curriculum Learning Algorithm Review

Algorithm 1 summarizes our CL framework for solving (11). In this algorithm, in each step of the CL method, a multi-agent RL problem which is initialized based on the output of the previous step of CL is solved. At the end of CL, the structure of the hierarchical belief set and the abstract and perfect description of each event are determined, which is the solution of (11). To our best knowledge, this algorithm is the first implementation of a CL-based reinforcement learning process that can be used to solve semantic communication problems. Our CL method includes $K - 1$ steps, where the value of $K$ is unknown at the beginning of CL. In step 1, we determine the input variables of the algorithm. In step 2, we initialize two main parameters of the algorithm, where $B_{\text{Remain}}$ is the set of beliefs whose levels in the hierarchical structure are not yet determined. $B_{e}^l$ is the set of all possible perfect descriptors of event $e$. In steps 4-9, we show the first CL step, while in steps 10-16 we show CL steps $l \geq 2$. For the first CL step, the state and action spaces of the speaker and listener are determined in step 5. In step 6, the multi-agent RL problem of the first CL step is performed. In step 7, the first output of the CL method which is related to event description is calculated and the value of $B_{e}^l$ is updated. Then, in step 8, the second output of the CL method related to the hierarchical structure of the belief set is calculated and the value of $B_{\text{Remain}}$ is updated. We have the same procedure for steps $l \geq 2$ of the CL methods, which are captured by steps 10-16 of Algorithm 1. The CL output is the set of all possible perfect descriptors of event $e$, captured by $B_{e}^l$.

### E. Algorithm Analysis

The main output of problem (13) is determining the perfect description of each event. Next, we derive a sufficient condition for guaranteeing that our CL method can determine the possible perfect descriptors of each event, at the end of each step of the CL when the multi-agent RL problem of that step converges. Note that, it is completely possible that for some events $e \in \mathcal{E}$, we have $B_{z_{l-1},H}^l = \emptyset$, where $B_{z_{l-1},H}^l$ is the set of $l$-beliefs perfect descriptors of $e$. This because, for such events, there is no $l$-beliefs perfect descriptor.

**Theorem 1:** A sufficient condition for finding the possible perfect descriptors of each event at the end of each step of the proposed CL method is that the reward, $R_T$, satisfies the following inequality:

$$R_T \geq \max \left( D_1, D_2 \right), \quad (27)$$

$$D_1 = \max_{T \in T} \left( \frac{\max \left( \Delta^S, \Delta^L \right)}{\min_{1 \leq j \leq L^\text{max}_{m-1}} \left[ P(j, L^\text{max}_{m}) + \tilde{P}(j, 1) \right]} \right), \quad (28)$$

$$D_2 = \max_{T \in T} \left( \frac{\max \left( \Delta^S, \Delta^L \right)}{\min_{2 \leq j \leq L^\text{max}_{m-1}} \left[ P(j, L^\text{max}_{m}) - P(j-1, L^\text{max}_{m}) \right]} \right), \quad (29)$$

$$\Delta^S = \max_{B_1, B_2 \subset B} \left| C^S(B_1) - C^S(B_2) \right|, \quad (30)$$

Note that deriving a necessary and sufficient condition here is analytically challenging.
\[
\Delta^L = \max_{B_1, B_2 \in B} \left| C^L(B_1) - C^L(B_2) \right|
\]

where \( B_1 \) and \( B_2 \) are different subsets of the belief set. \( C^S(B_1) \) and \( C^L(B_1) \) are the incurred cost of the speaker for transmitting \( B_1 \) and incurred cost of the listener for inferring \( B_2 \), respectively. The denominator of \( D_1 \) captures the summation of the probability of transiting to the initial and final event from event \( e_j \). The denominator of \( D_2 \) implies that, by going forward in a TEC, the probability of transiting to the final event must increase.

**Proof:** See Appendix V-A.

Theorem 1’s condition states that the gained reward of executing a task by using a perfect description should be greater than the additional incurred cost for using that perfect description compared to imperfect descriptions (\( \max(\Delta^S, \Delta^L) \)). To guarantee (27), the speaker and listener must know \( \Delta^S, \Delta^L \), and the characteristics of matrices \( P \) and \( \tilde{P} \) to determine the denominators of (28) and (29). In practice, the speaker and listener can use their initial interaction with the environment to compute these parameters and satisfy the condition. In particular, the interaction with the environment gives the speaker and listener insight into the parameters that contribute to \( D_1 \) and \( D_2 \). Then, they can estimate an upper bound for \( D_1 \) and \( D_2 \), thereby satisfying (27). Theorem 1’s condition can also help improve the convergence rate of the multi-agent RL problem of each CL step. Without meeting (27), the multi-agent RL problem of CL steps can still determine the possible perfect descriptors of each event. However, the convergence time will be larger than the case in which (27) is satisfied.

\[\text{F. Complexity Analysis}\]

We next compare the total complexity of the CL method with classical RL by comparing the product of the sizes of the state and action spaces size for both methods. This product reflects the number of \( Q \) values. In classical RL, the size of the state space is equal to the number of events, \( |\mathcal{E}| \), and the size of the action space is equal to the total number of possible descriptions, \( 2^B \), where \( B \) is the total number of the beliefs. The total number of \( Q \) values in classical RL will be \( |\mathcal{E}| \times 2^B \), and the complexity of classical RL will be \( O(|\mathcal{E}| \times 2^B) \). For CL, we have \( K - 1 \) steps, where \( K \) is the number of hierarchy levels. In each CL step, we have a multi-agent RL problem including the speaker and listener as the agents. The total size of the state space is equal to the product of the speaker and listener state space sizes. The total size of the action space is equal to the product of the speaker and listener action space sizes. Thus, in the first CL step, the total size of the state and action spaces will be \( |\mathcal{E}| \times B \) and \( B \times B - 1 \), respectively. The complexity of the first step of the CL method will be \( O(|\mathcal{E}| \times B^3) \). For CL step \( l \), the total size of the state space is \( |\mathcal{E}| \times |\Omega_{A,z_l}^L| \), where \( |\Omega_{A,z_l}^L| \) is the size of the listener’s state space. In CL step \( l \), the total size of the action space is \( |\Omega_{A,z_l}^S| \times |\Omega_{L_A,z_l}^L| \), where \( |\Omega_{A,z_l}^S| \) and \( |\Omega_{L_A,z_l}^L| \) are the sizes of the speaker and listener action spaces. According to (24) and (25), the maximum value of \( |\Omega_{A,z_l}^S| \) and \( |\Omega_{L_A,z_l}^L| \) is \( \binom{B}{l} \), which captures all possibilities of selecting \( l \) beliefs. Also, from (26), the maximum value of \( |\Omega_{A,z_l}^L| \) is \( \binom{B}{l} \). Then, the maximum value of the product of the size of total state space and size of total action space at step \( l \) is \( |\mathcal{E}| \times \left( \binom{B}{1} \times \binom{B}{2} \right) \) and the complexity of the learning method at step \( l \) is \( O(|\mathcal{E}| \times B^{2l+1}) \). We thus observe that the complexity of the learning in each step of the CL method is a polynomial function of \( B \). However, the complexity of the classical RL algorithm, \( O(|\mathcal{E}| \times 2^B) \), is an exponential function of \( B \). Clearly, the proposed CL method has a significantly lower complexity compared to classical RL.

**IV. Simulation Results and Analysis**

For our simulations, we consider a hierarchical belief set with 22 beliefs divided across 4 levels with 4 beliefs in level 1, 5 beliefs in level 2, 6 beliefs in level 3, and 7 beliefs in level 4. The number of beliefs increases with the hierarchy level because higher hierarchy levels have more semantic information. Also, the transmission and inference costs of the beliefs increase by going deeper in the hierarchy. We randomly generate the transmission cost of the beliefs in the first level with values between 1 and 2, the beliefs in the second level with values between 2 and 3, and we proceed similarly for the remaining levels. The listener’s inference cost for each belief is assumed to be half of the transmission cost of that belief. We consider 120 events, including initial, intermediary, and final events. The perfect description of each event can include 2 to 4 beliefs. We consider 30 tasks and assume the minimum and maximum number of events in each task to be 3 and 6. Matrices \( P \) and \( \tilde{P} \) are of size \( 120 \times 120 \), and they are generated according to the TEC of each task. For comparison, we use a classical RL solution based on Q-learning in which, to solve (13), the speaker explores different subsets of the belief set using the \( \epsilon \)-greedy policy and without the cooperation of the listener and also a CL without inference scheme.

**A. Convergence of the Proposed CL Method**

For investigating the convergence of the CL method, we can separately evaluate the convergence of the CL’s steps. In Fig. 4, the \( Q \)-values of the optimal action for initial and intermediary
events at the speaker side and for different steps of the CL method are presented. Fig. 4 shows that the multi-agent RL problem of each step converges in 10,000 episodes. The required number of episodes depends on the total number of beliefs, events, and tasks. Fig. 4 shows that the convergence of intermediary events is faster than the convergence of initial events. This is because, in the intermediary events, by transmitting a perfect description, the system can transit to a final event and gain an immediate reward. However, by transmitting a perfect description in the initial event, the system can only transit to an intermediary event with no immediate reward. Fig. 4 also shows a decrease in the $Q$-values at the beginning of the learning since the listener needs some episodes to find an appropriate inference for each action of the speaker. When the listener finds the appropriate inferred description, the $Q$-value of the optimal action of the speaker begins to increase until it finally converges. Fig. 4 shows that the convergence rate in the higher steps of the CL method is slower because of the increase in the size of the action space in the higher steps.

Now, we shed light on the result of Theorem 1 and the effect of the sufficient condition on the performance of the system. For this purpose, Fig. 5 evaluates the effect of the introduced constraint on the task execution time in the first CL step. Fig. 5 includes 10,000 episodes of the first step of the proposed CL method for both cases of meeting and not meeting the condition of Theorem 1. Fig. 5 shows that, by meeting the condition in Theorem 1, the convergence time of the proposed algorithm will be faster, and the task execution time will be smaller. Precisely, when the system can meet the condition of Theorem 1, it can achieve about 45.4% and 44% improvement in task execution time compared to not meeting the Theorem 1 condition during the learning and at the end of learning, respectively.

B. Performance Comparison of Proposed CL Method and Classical RL

Fig. 6 shows the task execution time and cost resulting from the proposed CL method and classical RL for different tasks. The tasks of step $l$ are the tasks that are executable in step $l$ of the CL method. Precisely, the TEC of the tasks of step $l$ includes events that can be described in step $l$ of the CL method. The task execution time is equal to the length of each episode. Figs. 6a-6b include 30,000 episodes for both methods, which is a sufficient time for the convergence of the CL method. Figs. 6a-6b show that classical RL performs better than the CL method for the tasks of the first step, and convergence of the CL method in the first step is slower than classical RL. This is because our approach has two agents, while classical RL uses a single agent. However, for the tasks of the second and third steps, the performance of the CL method is significantly better than classical RL in terms of task execution time and cost. In particular, Figs. 6a-6b show that the proposed CL method can minimize the task execution time and cost in each step by determining the perfect description of each event. The convergence rate of classical RL is significantly reduced in the second step. Meanwhile, at the third step, classical RL cannot converge within a reasonable time. This is because of the exponential increase in the size of the action space. Thus, the task execution time and cost of classical RL will increase significantly for the tasks of the third step. From Figs. 6a-6b, we observe that the proposed CL method yields 13.8% and 49.7% improvement compared to classical RL during the learning of the tasks of all steps, in terms of task execution time and cost, respectively.

Fig. 7 shows the task execution time and cost resulting from our CL method and CL without inference for different
Fig. 7. Comparison of the proposed CL and CL without inference in terms of task execution time and cost. 

Fig. 7a shows that CL without inference performs better than the proposed CL scheme during learning of the tasks of the first and second steps. However, after convergence, the proposed approach is slightly superior to the CL without inference. Also, for the task of the third step, our CL approach performs significantly better than the CL without inference during learning and after convergence. Fig. 7b shows that our CL method significantly outperforms the CL without inference scheme in terms of task execution cost for the task of the second and third steps. This significant advantage of the tasks of higher steps stems from the fact that, for such tasks, the action space size becomes larger and, thus, for CL without inference, the speaker cannot individually determine the hierarchical structure of the belief set and the perfect descriptions of the events. However, the proposed CL method can efficiently find the perfect and abstract descriptions of the events in the tasks of higher steps due to cooperation between speaker and listener. Specifically, the task execution cost of our CL scheme is significantly better than CL without inference, because our approach allows the speaker to only transmit an initial description of each event and the listener can complete the received description using its inference capability.

From Fig. 7a, we observe that the task execution time of the CL without inference scheme is 3.6% smaller than the proposed CL only during the learning of the tasks of all steps. Meanwhile, according to Fig. 7b, the proposed CL method improves the task execution time by about 32.6% compared to the CL without inference case for the tasks of all steps. Here, we note that the CL without inference scheme slightly outperforms the proposed CL method in terms of task execution time only during learning. When both methods converge, the proposed approach significantly outperforms the CL without inference scheme, as will be evident next.

To further highlight the advantage of our CL method, we consider a new metric called task execution reliability that quantifies the percentage of successfully executed tasks during a limited number of time slots. In Fig. 8, we present the task execution reliability resulting from the proposed CL method, classical RL, and CL without inference during different steps of the CL method. Fig. 8 shows that the proposed CL method can achieve 100% task execution reliability when the algorithm converges and determines the perfect descriptor of all events at the end of the third step. In contrast, the CL without inference achieves 98.8% task execution reliability when the algorithm converges. Meanwhile, since classical RL does not converge in the third step, its task execution reliability does not improve during the third step. Thus, the task execution reliability of the CL method significantly outperforms the classical RL at the end of learning. The proposed CL method yields around 45% improvement in task execution reliability, compared to classical RL, at the end of learning. Note that if the classical RL algorithm is given sufficient time to converge, it may reach a reliability of 100%. However, the required convergence time increases exponentially by going forward in the CL steps and, thus, the classical RL approach cannot achieve maximum reliability with minimal delay as done by our approach.

Figs. 9a-9b show the task execution time and cost resulting from our CL method, classical RL, and CL without inference.
Fig. 9. Benefit of proposed CL compared to classical RL and CL without inference for different events per task.

Fig. 10. Benefit of proposed CL compared to classical RL and CL without inference for different events per task.

as a function of the number of events per task at the end of learning. Fig. 9a shows that the task execution time resulting from the CL method increases proportionally with the number of events. However, the task execution time resulting from classical RL is significantly higher than that of the CL method for all number of events per task because classical RL cannot converge in reasonable time to the perfect event description. Also, our CL method significantly outperforms the CL without inference in terms of task execution time, specifically as the number of events per task increases. Fig. 9a shows that the CL method yields an average of 89% and 65.6% improvement in task execution time compared to classical RL and CL without inference, respectively. In Fig. 9b, we can see a similar improvement in task execution cost. Precisely, according to Fig. 9b, our CL scheme yields average improvements of 94% and 80.3% compared to classical RL and CL without inference, respectively.

Figs. 10a-10b show the task execution reliability and belief efficiency for the proposed CL method, classical RL, and CL without inference as a function of the number of events per task at the end of learning. Fig. 10a shows that the task execution reliability of the CL method is approximately independent of the number of events per task. However, the task execution reliability of classical RL decreases as the number of events per task increases. This is because, when the number of events per task increases, determining the perfect description becomes more challenging. From Fig. 10a, we observe that the proposed CL method yields, on average, about 45% improvement in task execution reliability compared to classical RL. Also, Fig. 10a shows that both the proposed CL method and CL without inference perform similarly in terms of task execution reliability. Fig. 10b shows that the CL method outperforms the classical RL and CL without inference in belief efficiency for all task sizes because, in the CL method, the speaker and listener jointly determine the perfect description of each event. Fig. 10b shows that when the number of events per task increases, the belief efficiency will decrease because the system will require more beliefs for task execution. However, the superiority of the CL method is maintained for different task sizes. Particularly, Fig. 10b shows that the proposed CL method yields, on average, around 2.5-fold and 15.5% improvement in belief efficiency compared to the classical RL and CL without inference, respectively.

C. Channel Error Effect on Proposed CL Method

In this work, we assumed that the wireless channel between the speaker and listener is perfect and, thus, the listener can receive the transmitted description of the speaker without error. This is a common assumption in the semantic communication literature [16]. However, it is useful to investigate the channel errors. In order to take into account the effect of wireless channel error in the proposed CL method, we can consider
Fig. 11. Effect of wireless channel error on the performance of the proposed CL method.

the existence of an error probability in the transmission of each belief, named belief transmission error and indicated by \( p(\text{error}) \). The value of \( p(\text{error}) \) indicates the probability that an error occurs in a transmitted belief from the speaker to the listener. For simplicity, we assumed that all beliefs of the belief set have an equal error probability. Extension to the case in which each belief has a specific error probability is straightforward. For simplicity, we assume that even if only one of the beliefs in the description \( B_{m,n}^S \) experiences error during transmission, the listener cannot utilize that description.

Figs. 11a-11b show the effect of different values of belief transmission error in the proposed CL method, in terms of task execution time and reliability during the learning, respectively. Precisely, Fig. 11a shows the task execution time of the proposed CL method in two cases of \( p(\text{error}) = 0 \) and \( p(\text{error}) = 0.1 \), respectively. We know that \( p(\text{error}) = 0.1 \) is a relatively large value for error probability of each transmitted belief. However, for such a large value of \( p(\text{error}) \), the performance of the proposed CL method is only slightly degraded. According to Fig. 11a, for \( p(\text{error}) = 0 \), the proposed CL method reaches 100% task execution reliability and for \( p(\text{error}) = 0.1 \), the proposed CL method reaches 96.6% task execution reliability. We can observe that for such a large value of \( p(\text{error}) \), the performance of the proposed CL method is only slightly degraded.

V. CONCLUSION

In this paper, we have investigated the problem of semantic communication for goal-oriented networks. We have introduced a novel model that allows a speaker and listener to jointly execute system tasks based on observed environmental events. In the introduced model, the speaker sends an abstract description of its own observation of the events. Then, the listener infers and completes the transmitted description of the speaker. For the communication between the speaker and listener, we have considered the existence of a common language consisting of a hierarchical belief structure that is based on the amount of semantic information that can be conveyed and the data type. Then, we have introduced an optimization problem to find the abstract and perfect description of each event to minimize the task execution cost with constraints on the task execution time and belief efficiency. To solve the introduced optimization problem, we have developed a novel CL framework that can gradually identify the structure of the hierarchical belief set and the perfect description of each event using the identified portion of the belief set. Simulation results show that our CL solution significantly outperforms the classical RL and CL without inference scheme in terms of task execution time, cost, reliability, and belief efficiency.

APPENDIX

A. Proof of Theorem 1

To solve the multi-agent RL problem at each CL step, we use the agent-based decomposition method from Q-learning [26]. In this method, for each state-action pair of each agent, we have a local \( Q \)-function that we update with interaction with the environment. For example, in the first CL step, for each action-state pair \((e_j, b_{i_1})\) of the speaker, where \( e_j \in \Omega_{A,z_1}^S \) and \( b_{i_1} \in \Omega_{A,z_2}^S \), we have a local \( Q \)-function \( Q^L(e_j, b_{i_1}) \), updated as follows:

\[
Q^L(e_j, b_{i_1}) := Q^S(e_j, b_{i_1}) + \beta \cdot \left[ R^S(e_j, b_{i_1}) + \gamma \cdot \max_{b_{i_1}'} Q^S(e_{j'}, b_{i_1}' \mid e_{i_1}') - Q^S(e_j, b_{i_1}) \right],
\]

(32)

where \( \beta \) is the learning rate, \( R^S(e_j, b_{i_1}) \) is the gained reward of the speaker for taking action \( b_{i_1} \) in state \( e_j \), and \( e_{i_1}' \) is the speaker’s next state. For each pair of state and action \((b_{i_1}, b_{i_2})\) of the listener, where \( b_{i_1} \in \Omega_{A,z_1}^L \) and \( b_{i_2} \in \Omega_{A,z_2}^L \), the local \( Q \)-function \( Q^L(b_{i_1}, b_{i_2}) \), updated as:

\[
Q^L(b_{i_1}, b_{i_2}) := Q^L(b_{i_1}, b_{i_2}) + \beta \cdot \left[ R^L(b_{i_1}, b_{i_2}) + \gamma \cdot \max_{b_{i_2}' \mid b_{i_2}} Q^L(b_{i_1}', b_{i_2}' \mid b_{i_1}') - Q^L(b_{i_1}, b_{i_2}) \right],
\]

(33)

where \( R^L(b_{i_1}, b_{i_2}) \) is the gained reward of the listener for taking action \( b_{i_2} \) in state \( b_{i_1} \), and \( b_{i_1}' \) is the next state.
of the listener. Now, consider a task $T$ with TEC $O'T = \{e^j_1, e^j_2, \ldots, e^j_{L_{\max} - 1}, e^j_{L_{\max}}\}$. For simplicity, we drop index $T$ and consider $O'T = \{e_1, e_2, \ldots, e_{L_{\max} - 1}, e_{L_{\max}}\}$. By initializing $Q^S(e_j, b_{i1}) = 0$ and $Q^L(b_{i1}, b_{i2}) = 0$, for all state-action pairs of the speaker and listener, according to the update formula of (32) and (33), the updated $Q$-values of the speaker and listener, after at least one visit of all state and action, can be written as in (40)-(41), shown at the top of the page, where $C^S(b_{i1})$ and $C^L(b_{i2})$ are the incurred cost of the speaker for transmitting belief $b_{i1}$ and the incurred cost of the listener for inferring belief $b_{i2}$. Considering (40) and (41), the sufficient conditions to converge to the perfect descriptor of each event at the end of each CL step, are:

$$R_T \geq \max\left(\frac{C^S(b_{i1}) - C^S(b'_{i1})}{|P(j, L_{\max}) + \tilde{P}(j, 1)|}, \frac{C^L(b_{i2}) - C^L(b'_{i2})}{|P(j, L_{\max}) + \tilde{P}(j, 1)|}\right)$$

where $b_{i1}, b_{i2} \in B^P_{e_j}$, $b'_i$ and $b''_i$ are beliefs that $\{b'_i, b''_i\} \notin B^P_{e_j}$. We can assume $R_T = C_T$, which is not a restricting assumption because both $R_T$ and $C_T$ are used to motivate the speaker and listener for minimizing the task execution time. Now, by simplifying (34) and (35), we have:

$$R_T \geq \frac{C^S(b_{i1}) - C^S(b'_{i1})}{|P(j, L_{\max}) + \tilde{P}(j, 1)|}$$

$$R_T \geq \frac{C^L(b_{i2}) - C^L(b'_{i2})}{|P(j, L_{\max}) + \tilde{P}(j, 1)|}$$

By combining these two constraints, and the inequalities $C^S(b_{i1}) - C^S(b'_{i1}) \leq \max_{b'_{i1}, b''_{i1}} |C^S(b_{i1}) - C^S(b'_{i1})|$ and $C^L(b_{i2}) - C^L(b'_{i2}) \leq \max_{b'_{i2}, b''_{i2}} |C^L(b_{i2}) - C^L(b'_{i2})|$, we have (42), where inequality (42) should hold for each $T \in T$ and, thus, we must have $R_T \geq D_1$, where $D_1$ is defined in (28). Similarly, combining the constraints in (36) and (37), $R_T \geq D_2$, where $D_2$ is defined in (29). Therefore, the sufficient condition would be $R_T \geq \max(D_1, D_2)$.

References

[1] P. Popovski, O. Simeone, F. Boccardi, D. Gündüz, and O. Sahin, “Semantic-effectiveness filtering and control for post-5G wireless connectivity,” *J. Indian Inst. Sci.*, vol. 100, no. 2, pp. 435–443, Apr. 2020.

[2] W. Saad, M. Bennis, and M. Chen, “A vision of 6G wireless systems: Applications, trends, technologies, and open research problems,” *IEEE Netw.*, vol. 34, no. 3, pp. 134–142, May/Jun. 2019.

[3] C. Chaccour, W. Saad, M. Debbah, Z. Han, and H. V.Poor, “Less data, more knowledge: Building next generation semantic communication networks,” 2022, arXiv:2211.14343.

[4] E. C. Strinati and S. Barbarossa, “6G networks: Beyond Shannon towards semantic and goal-oriented communications,” *Comput. Netw.*, vol. 190, May 2021, Art. no. 107930.

[5] X. Luo, H.-H. Chen, and Q. Guo, “Semantic communications: Overview, open issues, and future research directions,” *IEEE Wireless Commun.*, vol. 29, no. 1, pp. 210–219, Feb. 2022.

[6] F. Zhou, Y. Li, X. Zhang, Q. Wu, X. Lei, and R. Q. Hu, “Cognitive semantic communication systems driven by knowledge graph,” 2022, arXiv:2202.11958.

[7] J. Liu, S. Shao, W. Zhang, and H. V. Poor, “An indirect rate-distortion characterization for semantic sources: General model and the case of Gaussian observation,” 2022, arXiv:2201.12477.

[8] W. Yang et al., “Semantic communication meets edge intelligence,” 2022, arXiv:2202.06471.

[9] Q. Zhou, R. Li, Z. Zhao, C. Peng, and H. Zhang, “Semantic communication with adaptive universal transformer,” *IEEE Wireless Commun. Lett.*, vol. 11, no. 3, pp. 453–457, Mar. 2022.

[10] N. Farsad, M. Rao, and A. Goldsmith, “Deep learning for joint source-channel coding of text,” in *Proc. IEEE Int. Conf. Acoust.*, Speech Signal Process. (ICASSP), Calgary, AB, Canada, Apr. 2018, pp. 2326–2330.

[11] K. Lu, R. Li, X. Chen, Z. Zhao, and H. Zhang, “Reinforcement learning-powered semantic communication via semantic similarity,” 2021, arXiv:2108.12121.

[12] H. Du et al., “Rethinking wireless communication security in semantic Internet of Things,” 2022, arXiv:2210.04474.

[13] G. Zhang et al., “A unified multi-task semantic communication system for multimodal data,” 2022, arXiv:2209.07689.

[14] W. J. Yun et al., “Attention-based reinforcement learning for real-time UAV semantic communication,” in *Proc. 17th IEEE Int. Symp. Wireless Commun. Syst. (ISWCS)*, Sep. 2021, pp. 1–6.

[15] F. Lotti, O. Semiari, and W. Saad, “Semantic-aware collaborative deep reinforcement learning over wireless cellular networks,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, Seoul, South Korea, May 2022, pp. 5256–5261.

[16] H. Seo, J. Park, M. Bennis, and M. Debbah, “Semantics-native communication with contextual reasoning,” 2021, arXiv:2108.05861.

[17] S. Narvekar, B. Peng, M. Leonetti, J. Sinapov, M. E. Taylor, and P. Stone, “Curriculum learning for reinforcement learning domains: A framework and survey,” *J. Mach. Learn. Res.*, vol. 21, pp. 1–50, Jul. 2020. [Online]. Available: http://jmlr.org/papers/v21/20-212.html

[18] M. K. Farshbafan, W. Saad, and M. Debbah, “Common language for goal-oriented semantic communications: A curriculum learning framework,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, Seoul, South Korea, May 2022, pp. 1710–1715.

[19] H. Xie, Z. Qin, and G. Y. Li, “Task-oriented multi-user semantic communications for VQA,” *IEEE Wireless Commun. Lett.*, vol. 11, no. 3, pp. 553–557, Mar. 2022.

[20] H. Xie, Z. Qin, X. Tao, and K. B. Letaief, “Task-oriented multi-user semantic communications,” *IEEE J. Sel. Areas Commun.*, vol. 40, no. 9, pp. 2584–2597, Sep. 2022.
Mohammad Karimzadeh Farshbafan received the B.S. degree in electrical engineering from the Amirkabir University of Technology, Iran, in 2012, the M.S. degree in electrical engineering from the Sharif University of Technology, Iran, in 2014, and the Ph.D. degree in electrical engineering from the University of Tehran, Iran, in 2020. He is currently pursuing the Ph.D. degree with the Bradley Department of Electrical and Computer Engineering, Virginia Tech. His research interests include wireless networking, semantic communication, machine learning, and network softwarization.

Walid Saad (Fellow, IEEE) received the Ph.D. degree from the University of Oslo, Oslo, Norway, in 2010. He is currently a Professor with the Bradley Department of Electrical and Computer Engineering, Virginia Tech, where he leads the Network Science, Wireless, and Security (NEWS) Laboratory. He is also the Next-G Wireless Research Leader at Virginia Tech’s Innovation Campus. His research interests include wireless networks, game theory, security, unmanned aerial vehicles, semantic communications, and wireless physical systems, and network science. He was a recipient of the NSF CAREER Award in 2013, the AFOSR Summer Faculty Fellowship in 2014, and the Young Investigator Award from the Office of Naval Research (ONR) in 2015. He was the author/coauthor of 11 Conferences Best Paper Awards from WiOpt in 2009, ICIMP in 2010, IEEE WCNC in 2012, IEEE PIMRC in 2015, IEEE SmartGridComm in 2015, EucNC in 2017, IEEE GLOBECOM in 2018, IFIP NTMS in 2019, IEEE ICC in 2020 and 2022, and IEEE GLOBECOM in 2020. He was also a recipient of the 2015 and 2022 Fred W. Ellersick Prize from the IEEE Communications Society, the 2017 IEEE ComSoc Best Young Professional in Academia Award, the 2018 IEEE ComSoc Radio Communications Committee Early Achievement Award, and the 2019 IEEE ComSoc Communication Theory Technical Committee. He was the co-chair of the 2019 IEEE Communications Society Young Author Best Paper and the 2021 IEEE Communications Society Young Author Best Paper. From 2015 to 2017, he was named as the Stephen O. Lane Junior Faculty Fellow at Virginia Tech. In 2017, he was named as a College of Engineering Faculty Fellow at Virginia Tech. He received the Dean’s Award for Research Excellence from Virginia Tech in 2019. He was also an IEEE Distinguished Lecturer from 2019 to 2020. He currently serves as an Editor for the IEEE TRANSACTIONS ON MOBILE COMPUTING and the IEEE TRANSACTIONS ON SIGNAL PROCESSING.

Merouane Debbah (Fellow, IEEE) received the M.Sc. and Ph.D. degrees from the Ecole Normale Supérieure Paris-Saclay, France. From 1999 to 2002, he was with Motorola Labs, Saclay, France. He was with the Vienna Research Center for Telecommunications, Vienna, Austria, until 2003. From 2003 to 2007, he was an Assistant Professor with the Department of Mobile Communications, Institut Eurecom, Sophia Antipolis, France. In 2007, he was appointed as a Full Professor at the CentraleSupélec, Gif-sur-Yvette, France. From 2007 to 2014, he was the Director of the Alcatel-Lucent Chair on Flexible Radio. From 2014 to 2021, he was the Vice-President of the Huawei France Research Center. He was jointly the Director of the Mathematical and Algorithmic Sciences Laboratory and the Lagrange Mathematical and Computing Research Center. Since 2021, he has been leading the AI and Digital Sciences Research Center, Technology Innovation Institute, Abu Dhabi. He is currently the Chief Researcher with the Technology Innovation Institute, a Professor with the CentraleSupélec, and an Adjunct Professor with the Department of Machine Learning, Mohamned Bin Zayed University of Artificial Intelligence, Abu Dhabi. He has managed eight EU projects and more than 24 national and international projects. He held more than 40 patents. His research interests include fundamental mathematics, algorithms, statistics, information, and communication sciences research. He is a WWRF Fellow, a Eurasisip Fellow, an AAIA Fellow, an Institut Louis Bachelier Fellow, and a Membre émérite SEE. From 2021 to 2022, he has served as an IEEE Signal Processing Society Distinguished Industry Speaker. He was a recipient of the ERC Grant MORE (Advanced Mathematical Tools for Complex Network Engineering) from 2012 to 2017. He was also a recipient of the Mario Boella Award in 2005, the IEEE Glavieux Prize Award in 2011, the Qualcomm Innovation Prize Award in 2012, the IEEE Radio Communications Committee Technical Recognition Award in 2019, and the SEE Blondel Medal in 2020. He received more than 25 Best Paper Awards, among which the 2007 IEEE GLOBECOM Best Paper Award, the WiOpt 2009 Best Paper Award, the 2010 Newcom++ Best Paper Award, the WIN CogCom Best Paper Award in 2012 and 2013, the 2014 WCNC Best Paper Award, the 2015 ICC Best Paper Award, the 2015 IEEE Communications Society Leonard G. Abraham Prize, the 2015 IEEE Communications Society Best Tutorial Paper Award, the 2016 European Wireless Best Paper Award, the 2017 Eurasip Best Paper Award, the 2018 IEEE Marconi Prize Paper Award, the 2019 IEEE Communications Society Young Author Best Paper Award, the 2021 Eurasip Best Paper Award, the 2021 IEEE Marconi Prize Paper Award, the 2022 IEEE Communications Society Outstanding Paper Award, the 2022 ICC Best Paper Award, the 2022 IEEE GLOBECOM Best Paper Award, the 2022 IEEE TCOS TC Best GCSN Paper Award, the 2022 IEEE International Conference on Metaverse Best Paper Award, the Valuetools 2007, the Valuetools 2008, the ValueTools 2009, the ValueTools 2012, the SAM 2014, and the 2017 IEEE Sweden VT-COM-IT Joint Chapter Best Student Paper Awards. He was an Associate Area Editor and a Senior Area Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING from 2011 to 2013 and from 2013 to 2014, respectively. He is an Associate Editor-in-Chief of the Random Matrix: Theory and Applications journal.