Timely and Massive Communication in 6G: Pragmatics, Learning, and Inference

Deniz Gündüz, *Fellow, IEEE*, Federico Chiariotti, *Member, IEEE*, Kaibin Huang, *Fellow, IEEE*, Anders E. Kalør, *Member, IEEE*, Szymon Kobus *Student Member, IEEE*, and Petar Popovski, *Fellow, IEEE*

***Abstract*—5G has expanded the traditional focus of wireless systems to embrace two new connectivity types: ultra-reliable low latency and massive communication. The technology context at the dawn of 6G is different from the past one for 5G, primarily due to the growing intelligence at the communicating nodes. This has driven the set of relevant communication problems beyond reliable transmission towards semantic and pragmatic communication. This paper puts the evolution of low-latency and massive communication towards 6G in the perspective of these new developments. At first, semantic/pragmatic communication problems are presented by drawing parallels to linguistics. We elaborate upon the relation of semantic communication to the information-theoretic problems of source/channel coding, while generalized real-time communication is put in the context of cyber-physical systems and real-time inference. The evolution of massive access towards massive closed-loop communication is elaborated upon, enabling interactive communication, learning, and cooperation among wireless sensors and actuators.**

***Index Terms*—6G, AI-based networking, massive access, machine-to-machine communications, pragmatic communication, semantic communication**

D. Gündüz and S. Kobus are with the Department of Electrical and Electronic Engineering, Imperial College London, U.K. (email: {d.gunduz,szymon.kobus17}@imperial.ac.uk). F. Chiariotti is with the Department of Information Engineering, University of Padova, Italy (email: chiariot@dei.unipd.it). K. Huang and A. E. Kalør are with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong (email: {huangkb,aekaloer}@eee.hku.hk). P. Popovski is with the Department of Electronic Systems, Aalborg University, Denmark (email: petarp@es.aau.dk).

D. Gündüz and S. Kobus received funding from the UKRI for the project "AIR" (ERC-Consolidator Grant, EP/X030806/1). F. Chiariotti's work was supported by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, under the Young Researchers grant SoE0000009 "REDIAL.". K. Huang's work was supported in part by the Research Grants Council of the Hong Kong Special Administrative Region, China under a fellowship award (HKU RFS2122-7S04), the Areas of Excellence scheme grant (AoE/E-601/22-R), and the Grant 17212423. The work of A. E. Kalør was supported by the Independent Research Fund Denmark (IRFD) under Grant 1056-00006B. The work of P. Popovski was supported by the Villum Investigator Grant "WATER" financed by the Villum Foundation, Denmark.

I. Introduction

The focus of wireless cellular evolution before 5G was on high-fidelity delivery of voice and reliable data transmission at increasing rates. 4G can be seen as a reliable broadband data pipe, geared towards human-operated mobile devices. 5G expanded the wireless landscape by considering autonomous machine-type devices, robots, as well as a new class of tactile, human-operated devices. This led to two new connectivity types in 5G: low latency, coupled with high reliability, and massive Internet of Things (IoT) communication. Tracing the further evolution of timely and massive communication needs to account for the 6G features that came to prominence via various technology visions. The premises in this article are:

- With the advances in hardware technology and machine learning (ML) and artificial intelligence (AI) algorithms, the intelligence available to communicating devices is steadily increasing, leading to potential synergies between communication, computation, and learning. Transmitted data can be used for the learning process and, vice versa, communication can be aided by the intelligence at the nodes and embedded in the protocols. As the vision of networked intelligence is emerging, the problem of communication as an accurate reproduction of data bits needs to be generalized to the problem of communicating the relevant information for the desired task, which can include computation, learning and inference, summarized in the phrase *semantic communication* [1]–[3].
- Recently, there has been a growing awareness that low latency is just one example of a variety of timing constraints in a communication system [4]. In practice, the timing requirements belong to a more general category in which the usefulness of communication is gauged through a goal accomplishment. This is the basis for *pragmatic communication*, often called goal-oriented or effective communication in the literature, where cyber-physical systems interact with each other and the environment.

- Massive IoT communication has been mainly considered as a mechanism to offload data collected by edge nodes to a central or Cloud server. Thus, massive access has traditionally been defined as a problem for uplink transmission. Nevertheless, new applications call for revising this assumption, as various real-time systems require closed-loop communication with a massive number of sensors and actuators. The data acquired by IoT devices can be used for distributed training of ML models, using both uplink and downlink communications.

Based on these observations, this paper will describe the evolution of 6G systems towards a *network of intelligence* augmenting low-latency and massive communications considered in 5G. We start with a discussion on semantic and pragmatic communications by drawing parallels to linguistics. Next, we describe the relation between semantic communication and data/knowledge representation seen through the lens of source coding. This is followed by generalized real-time requirements in 6G, including both cyber-physical systems that interact with the environment and real-time inference. Finally, we discuss closed-loop massive communication and its potential role in distributed and federated learning.

II. Semantic and Pragmatic Communication

A standard introduction to semantic communication refers to the preface that W. Weaver wrote in [5] to the original work of C. Shannon from 1948. Weaver defined three types of communication problems: *(Level A)* The technical problem: accurate transmission of bits from the transmitter to the receiver; *(Level B)* The semantic problem: how precisely do the transmitted symbols convey the desired meaning; and *(Level C)* The effectiveness problem: how the received meaning affects conduct in the desired way. Despite the initial intuitive appeal of this classification, the distinction between semantics and effectiveness and the corresponding communication models have remained blurry. This is also reflected in the often used phrase "semantic and goal-oriented communications," which treats the beyond-technical problems in a bulk, avoiding to make a clear distinction; yet, the term "goal-oriented" presumably covers the part of effectiveness.

As an attempt for a clearer distinction between Levels B and C, we resort to the classification in linguistics [6], as shown in Fig. 1, and relate them to problems in communication engineering, as well as to Weaver's classification. *Phonology* deals with the speech sounds and phonemes as basic ingredients of a language. A phoneme can be related to a physical layer symbol from

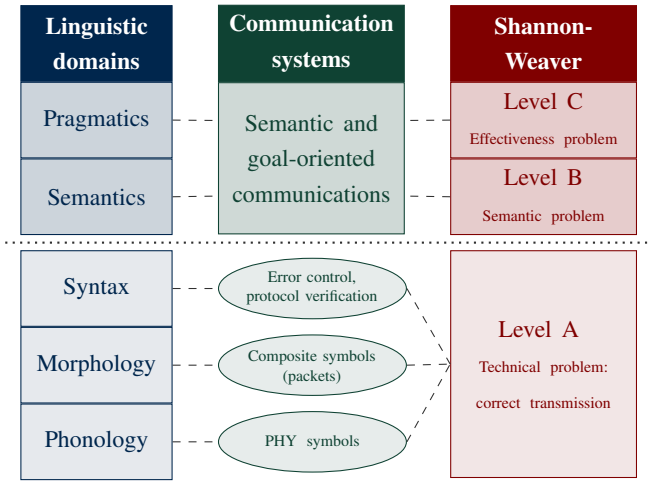

Fig. 1. The five linguistic domains, their relation to different elements of communication engineering and Weaver's classification.

a given constellation used in communication engineering or a channel use in an information-theoretic model. *Morphology* studies the minimal meaningful units of language called morphemes. In a communication system, such a meaningful unit can be a group of symbols, such as an OFDM symbol, a codeword, or a packet. *Syntax* deals with the rules to form sentences and can thus be related to the correct protocol operation. Some syntactic elements are also present in individual packets, as they are data units with integrity check that allow design of link-layer protocols with error detection. Regardless of that, the first three linguistic domains can be mapped, as a group, to various elements that constitute the technical communication problem, as illustrated in Fig. 1. *Semantics* studies the signs and their relation to the world[1]. Finally, *pragmatics* refers to the rules in conversation and social contexts, including implicit communication through external cues and shared context. Fig. 1 shows that these two domains, corresponding to Weaver's semantic and effectiveness levels, are not considered in the design of modern communication systems, which focus on the technical problem, and therefore, has motivated a flurry of recent research activity. Against this background, we will first attempt to define a distinction between semantic and pragmatic communications. Our choice to use the term "pragmatic" to refer to Level C problems is motivated by a closer match with the linguistic metaphor, which clarifies the distinction between the two.

Semantic communication refers to the the unidirectional transmission of a message or a signal x under a

[1]Following the argument of C. W. Morris on the ambiguity of "meaning" due to overuse, we have adopted his definition of semantics, without the word "meaning". See: C. W. Morris, Foundations of the Theory of Signs. University of Chicago Press, 1938.

prescribed measure of quality that relies on the semantics, along with the utilized source/channel coding. As such, it does not need to be related to the physical time, but can work with an abstract communication model, consisting of channel uses. Time is not explicitly a part of Shannon's mathematical model of communication; that model provides a causal relation between inputs and outputs, but says nothing about the time duration between two inputs, two outputs, or between an input and its corresponding output. Semantic communication can be put in a temporal context, such as "convey a certain meaning within a given time interval T" by, e. g., establishing a symbol duration T_s, a codeword duration T and codeword length of $n = \frac{T}{T_s}$ symbols.

Pragmatic communication captures the interaction between communicating entities, machines or humans, as well as interaction with their physical environment, such as movement or actuated command, which determines the dynamic context. Dealing with the physical time is thus necessary in pragmatic communication, as the interaction between the entities via communication and/or action occurs under the constraints of a physical time. Note that pragmatic communication is not identical to multi-user communication, as there are multi-user channels in which the transmitting users are not interacting with each other, such as the multiple access or broadcast channels, which can also be seen as setups for unidirectional semantic communication.

III. Semantic Communication

Semantic communication has recently become a catch-phrase to describe communication systems that cater for the content of the source signal rather than focusing solely on the reliable transmission of bits. Current communication protocols are designed to create the largest capacity reliable bit pipes with minimal resources, i.e., maximize the capacity in terms of bits per signal dimension, while achieving the highest reliability with the highest energy efficiency. In that sense, the codes and modulation schemes employed at the physical layer are independent of the delivered content and transfer the packets of bits to their destinations. Content is handled at the application layer, where different types of information sources, e.g., text, image, or video, are transformed into packets of bits, with the goal of obtaining the most efficient representation that allows the receiver to reconstruct the information with the desired fidelity. Thus, the semantic aspects of the source are currently handled at the application layer.

Semantic communication is commonly exemplified through the communication of a text or an image. Semantic-aware communication conveys the meaning of a text without sending the exact order of words. For images, semantics typically refers the objects in the image and their relative locations, without necessarily reconstructing them with a pixel-level fidelity. Extracting the meaning of a sentence or the semantic content of an image are highly challenging tasks, but data-driven ML algorithms has led to significant progress. Despite the impression that this type of semantic communication departs from the basic information theoretic principles, these problems can still be treated within the framework of rate-distortion theory, dealing with an efficient representation of the source signal within the desired fidelity. However, unlike the problems that one encounters in a first course on rate-distortion theory, many practical sources do not have independent and identically distributed (i.i.d.) samples, and we typically do not have a well-defined additive fidelity measure, such as mean square error (MSE). Indeed, a major challenge is to identify the fidelity measure itself, even before specifying a compression algorithm. Shannon was well-aware of the limitations of the assumptions in his theory, but these were necessary to obtain the succinct single-letter mathematical expressions that connect source compression with channel coding through the mutual information functional. However, the complexity of this setup can be addressed through ML algorithms. This approach to semantic communications that relies on generalized distortion measures has become relevant with the recent advances in ML algorithms and the intelligence of communicating devices.

In his seminal paper [5], after introducing the fidelity measure $\rho(x, y)$ between the input signal x and its reconstruction y, Shannon adds the following remark:

"The structure of the ear and brain determine implicitly a number of evaluations, appropriate in the case of speech or music transmission. There is, for example, an 'intelligibility' criterion in which $\rho(x, y)$ is equal to the relative frequency of incorrectly interpreted words when message $x(t)$ is received as $y(t)$. Although we cannot give an explicit representation of $\rho(x, y)$ in these cases, it could, in principle, be determined by sufficient experimentation. Some of its properties follow from well-known experimental results in hearing, e.g., the ear is relatively insensitive to phase and the sensitivity to amplitude and frequency is roughly logarithmic."

One can argue that, here, Shannon makes an early argument for a data-driven approach to measuring fidelity, as done in state-of-the-art ML algorithms.

It is possible to extend this framework to include other types of fidelity measures. For example, rather than reconstruction, the receiver may want to only estimate

certain statistics of the source signal; or infer another random variable correlated with the source signal, which can correspond to, e.g. the class of the input sample. Most of these problems can be considered in the context of remote source coding [7]: the encoder has access to correlated observation z with the underlying source x, which the receiver wants to reconstruct. For example, z may represent the class the input x belongs to, or its features. In [7], x and z come from a known joint distribution $p(x, z)$, and i.i.d. samples $z^n = (z_1, \ldots, z_n)$ of z are available at the encoder, while the decoder wants to decode x^n. This is a generalization of Shannon's rate-distortion problem, reducible to Shannon's original formulation by an appropriate change of the distortion measure. This formulation is quite general, and can cover many statistical problems under communication constraints. For example, when the fidelity measure between the reconstruction y^n and the source sequence x^n is the *log-loss*, we can obtain a single-letter expression for the remote rate distortion function, equivalent to the well-known *information bottleneck* function.

A. Joint Source-Channel Coding (JSCC)

Separation of source and channel coding leads to an architecture that is modular and universal, where bits coming from different sources can all be delivered using the same channel code. This simplifies the code design for each subproblem, for example, code design for compressing different sources, or reliable communication over different types of channels, and resulted in the specialization of engineers and researchers to specific subproblems. The separate architecture also allows encryption straightforwardly: as the communication system is oblivious to the data source, the random bits obtained as a result of the compression can be encrypted with no impact on the way they can be transmitted. The theoretical basis for the separated architecture is Shannon's separation theorem [5], showing that separation is asymptotically optimal for ergodic sources and channels.

Nevertheless, there are many scenarios in which a joint design can provide performance gains and simplify the code design at the expense of modularity. A prominent example is the transmission of an i.i.d. source sequence of Gaussian samples, x^n, over n uses of an additive white Gaussian channel under the MSE distortion measure. Here, the optimal scheme consists of transmission of each source sample over one channel use, by simple scaling according to the transmitter power constraint, and estimating each sample with a minimum mean square estimator (MMSE) at the decoder. This simple uncoded scheme offers the lowest possible latency, while achieving the same average MSE performance for any n that can be achieved by the separation scheme as $n \to \infty$. If we instead consider two receivers with different signal-to-noise ratios (SNRs), the uncoded scheme simultaneously achieves the optimal performance for both receivers as if each one is the sole receiver in the system; while this is not possible with standard channel codes that operate at the capacity region of the underlying broadcast channel. Yet, until recently, no practical joint source-channel coding (JSCC) design provided significant gains compared to separation based codes for practical sources such as images or videos. Most of the existing designs relied on separate source and channel codes, whose parameters are optimized jointly in a cross-layer fashion. While this approach retained some modularity, the gains are mainly through adapting the source/channel code parameters according to temporal variations of the underlying source and channel statistics. Despite certain performance gains are achieved for non-ergodic channels/sources, they are not truly joint code designs.

More recently, significant progress has been made by employing deep learning in JSCC problems. Pioneered in [8], these truly joint designs are known as DeepJSCC, where the encoder and decoder are parameterized as a pair of deep neural networks (DNNs), trained jointly on a given source dataset and channel distribution. DeepJSCC provides several significant advantages: First of all, even for a given fixed channel SNR, for which the right compression and channel code rates can be chosen to achieve reliable communication with high probability, DeepJSCC can achieve a better performance compared to the concatenation of state-of-the-art compression codecs (e.g., BPG or JPEG200) and channel codes, e.g., low-density parity check (LDPC). The results show that the gains of DeepJSCC are larger for low bandwidth ratio (average channel uses per source sample) and low SNR regimes. The gains also become more pronounced when a perceptually aligned distortion measure, such as multi-scale structural similarity index measure (MS-SSIM), is considered, or the model is trained and tested on data from a specific domain, such as satellite images. This is because DeepJSCC can be trained with a specific loss function or a dataset, while conventional compression algorithms are not adaptive. Moreover, DeepJSCC simplifies the coding process, and can provide significant reduction in complexity. The architecture in [8] is a simple 5-layer convolutional neural network (CNN), which can be parallelized and implemented efficiently on a dedicated hardware, whereas conventional compression codecs and iterative decoding algorithms are computationally much more demanding. On the other hand,

the original design in [8] requires training a separate encoder-decoder network pair for each channel condition (SNR/bandwidth ratio), which limits its practicality as it would require storing different network parameters to be used in different scenarios. This would impose significant memory complexity. Several adaptive architectures have been subsequently proposed, showing that a single DeepJSCC network can be deployed for a range of channel and bandwidth values [9].

Learning-based solutions to semantic communication bring about another unexplored dimension of compression and JSCC problems. In conventional information theoretic approaches to these problems, we assume perfectly known source and channel statistics, and provide certain performance guarantees under the ergodicity assumption. Alternatively, in universal schemes, the distribution is either learned from a first pass over the particular input sequence to be compressed, or the codebook is adapted dynamically to the sequence as it is being coded, as in the well-known Lempel-Ziv algorithm. In the aforementioned ML approaches to JSCC, we assumed an offline training stage that learns the code using a specific dataset of images and a channel distribution. On the other hand, if the source statistic changes over time, an online learning approach can be adapted, in which case the overhead of communicating the updated model should also be taken into account.

IV. Generalized 6G Real-time Communication

This section treats the role of timing in communication systems, which is an essential component of the engineering formulation of pragmatic communications. We will explain how it generalizes the technical communication problem and highlight its connections to other timing-based performance measures.

A. Pragmatic Communication

Formulating a communication problem of effectiveness/pragmatics (Level C) and favorably affecting an agent's conduct requires a quantifiable way to measure the effect of its actions. A general way to model series of actions and their outcomes is a Markov decision process (MDP), where an agent acts in a stochastic environment and aims to accumulate as much reward as possible. Specifically, the current state of the environment and the action of the agent affect the probability of the consequent state and the reward.

The effectiveness/pragmatic communication problem is formulated in [10] as an extension of the MDP framework to a multi-agent partially observable Markov decision process (MA-POMDP), in which two agents communicate to maximize their common reward. The agents represent different parties, such as transmitters, receivers, controllers, and robots, and partial observability means that some information is not known to all the parties in the system. The joint actions of the agents influence the evolution of the environment. A simple setup that captures the fundamental aspects is a *remote MDP*, as formulated in [10] and shown in Fig. 2. Here a controller, say a guide, observes the environment and transmits messages through a noisy channel to instruct and navigate the agent towards a target location. The actions of the agent can be possible movements in the environment, while the actions of the guide are the messages it transmits. A large positive reward is received when the agent reaches the target, while a unit of negative reward is accrued at each time. This incentivizes to reach the target as soon as possible, as the goal is to maximize the total cumulative reward. Note that the timing here is in reference to the interactions with the environment, while the channel can still be used multiple times for the transmission of each message. Differently from Level A or B problems, here the communication performance cannot be measured by the reliability of bits transmitted at each round, or the reconstruction quality of the receiver. What matters is the set of actions taken over the horizon until the target is reached, while maximizing the cumulative reward, and the context/state of the receiver is critical to determine the success/quality of each transmission, as in the definition of pragmatic communication in linguistics.

Moreover, this formulation includes Level A and B problems as special cases [10]. If the state of the environment is a randomly generated bit sequence, and the goal of the agent is to match this bit sequence, then we recover the classical channel coding problem, where maximizing the average reward over a fixed time horizon is equivalent to minimizing the error probability of a fixed-length code. If the state of the environment is a randomly generated sequence of i.i.d. symbols from a fixed distribution $p(x)$ and the goal of the agent is to match this sequence with the minimum distortion, then we recover the lossy source coding, or the semantic communication problem as formulated in Section III.

The notion of pragmatic communication can be extended to source coding [11], where the standard objective is to minimize the expected length of the description of a random symbol drawn from an alphabet. To pose it as an effectiveness problem, let the alphabet represent the set of possible states the receiver can occupy: the aim is to communicate a randomly drawn state within the minimum average time; however, as in a remote MDP, the receiver's state cannot change arbitrarily but

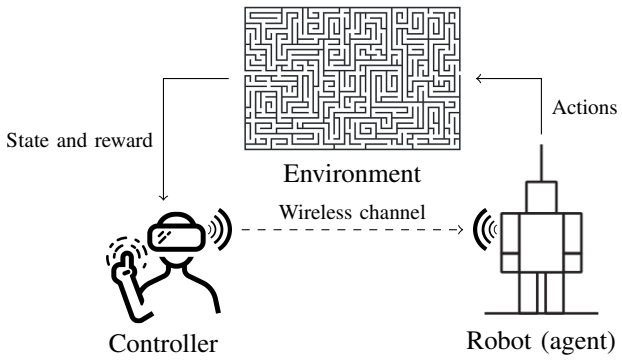

Fig. 2. Illustration of the remote MDP problem exemplifying level C pragmatic communications: an intelligent controller controls an agent over a wireless channel to maximize its reward through interactions with the environment.

there is a constrained set of possible transitions. This setting can be interpreted as guiding an agent through a graph of possible states from the initial to the goal state. At each time step, the agent receives a message and decides a state transition. The simplest strategy to communicate such a plan is to code each state transition sequentially; this has the benefit of immediate relevance to the controlled agent. The opposite approach is to only send the goal, which is more efficient in terms of total communicated information, but introduces a greater delay before actionable information. In general, a right balance needs to be achieved between prioritizing actionable information in the short term and information useful over a long term, bringing timing aspects into a simple compression problem.

B. Timing Constraints and Pragmatic Communication

The underlying assumption behind defining timing-related QoS is that the delay between the generation of a piece of information and its reception and processing by the receiver affects the control performance [4]. The nature of this assumption becomes clear for Level C problems: timing is useful because information about a process becomes less accurate over time, as the real environment drifts from the observed state.

Latency and age of information (AoI) are textbook Level A metrics [4], as they do not consider the content of updates, only how quickly they are delivered (latency) or how stale the information available at the receiver gets (AoI). If we consider a tracked signal $x(t)$, which is observed by the transmitter, the latency of packet i, generated at time g_i and received at time r_i, is simply $\tau_i = r_i - g_i$, while the AoI at time $t > r_i$ (but before any other, more recent, packet is received) is $\Delta(t) = t - g_i$. We can see that AoI $\Delta(t)$ is tied more closely to the tracked process than the latency: as the last update from the transmitter has an age $\Delta(t)$, the receiver must estimate the current value of the process, $\hat{x}(t)$, using only the historic data of the signal up until time g_i.

The value of information (VoI) considers the *content* of messages and not just their timing [12]. If the receiver has a known estimator, the value $\nu(t)$ of an update generated at time t is given by the difference its transmission would make in the error. The receiver gets a series of observations $h(t) = (y_1, g_1), \ldots, (y_i, g_i)$, which correspond to the packets transmitted up until time t by the sensor observing the process and their generation instants. The observations may be noisy, requiring further processing. The receiver keeps an estimate $\hat{x}(t|h(t))$ and the error function can be defined as $e(x(t), \hat{x}(t|h(t)))$. The VoI of an update is defined as:

$$\nu(t) = e(x(t), \hat{x}(t|h(t))) - e(x(t), \hat{x}(t|h(t), y(t))) \tag{1}$$

For well-defined $e(x(t), \hat{x}(t|h(t)))$ (e.g. ℓ_2 norm), and the receiver has a good estimator, VoI is always positive, as additional information will never increase the error.

We can further distinguish between *push-based* communication systems, where the sender decides autonomously when to send data, and *pull-based* systems, where the sender only sends updates in response to requests from the receiver. In the pull-based case, the VoI definition in (1) cannot use the value of $y(g_i)$, as the receiver does not know it before it is transmitted, but must estimate the VoI as well, based on its current knowledge [13]:

$$\begin{aligned}\nu'(t) = &\mathbb{E}\left[e(x(t), \hat{x}(t)|h(t))\right] \\ &- \mathbb{E}\left[e(x(t), \hat{x}(t|h(t), y(t))|h(t)\right].\end{aligned} \tag{2}$$

If the process is *memoryless*, i.e., the residual error of the estimator is a martingale (e.g., a Wiener process), the Level B problem of VoI maximization reduces to a simpler Level A AoI minimization. In this case, the estimate of the current state can be performed knowing only $\Delta(t)$ with the same precision as the estimate knowing the full history $h(t)$, and minimizing a (potentially non-linear) function of $\Delta(t)$ is equivalent to maximizing the receiver-computed VoI. We can contrast this with *stateful* processes, which include most Markov processes, where the value of $x(t)$ affects future changes in the process, and AoI is not sufficient to compute the expected VoI.

Naturally, we can extend this to multiple dimensions, and different error functions: we can easily imagine a system in which a multidimensional signal $\mathbf{x}(t)$ is reconstructed from multiple noisy measurements from a set of N different sensors, none of which can directly communicate with the others. In this case, computing VoI is more complex: the receiver has a better picture

TABLE I
EQUIVALENCES: AOI, SEMANTIC VOI, AND PRAGMATIC VOI

Task	Communication	Process	AoI	VoI (B)	VoI (C)
Tracking	Pull-based	Martingale	✓	✓	✓
		Stateful		✓	✓
	Push-based	Any		✓	✓
Control	Pull-based	Martingale	✓	✓	
		Stateful			
	Push-based	Any			

of the environment, i.e., can receive data from all the sensors, but its knowledge is statistical and partially outdated. On the other hand, each sensor can have relatively precise information about the part of the signal it observes, but a much narrower view of the overall system. Medium access is also a significant issue, and it is generally easier to avoid collisions in pull-based systems, as the receiver can act as a central coordinator and limit the pool of potential transmitters by requesting specific information, or even address nodes directly. A further complication comes from epistemic errors, i.e., events and perturbations not included in the receiver's model of the tracked signal; a purely pull-based system will often take a long time to detect these errors, and operate poorly on incorrect assumptions.

We can then move to the Level C problem by introducing the possibility of control, i.e., of changing the environment to achieve a specific goal [10]. Note that the environment is subject to natural evolution within the physical time, such that the timing requirements depend on the speed of the process relative to the AoI, as already indicated for pragmatic communication. In this case, the definition of VoI can be adapted to consider the performance of the controller instead of the accuracy of the estimator. Unless the control task is simply to track the input signal, the *pragmatic* VoI is always different from the *semantic* VoI, as it considers both the content of a sensory observation and its effect on control. There are three possible cases: *(1)* The update has a low semantic value and does not significantly affect the receiver's estimate of the environment. It can be subject to lossy compression with a limited loss in terms of performance, or even discarded entirely; *(2)* The update has a high semantic value, but it has a low pragmatic value: the change in the receiver's estimate leads to taking the same action, or another action with the same control performance. Thus, a semantic scheduler and compressor would consider this type of update highly important, while it has low relevance to the Level C problem; *(3)* The update significantly affects the receiver's choices, and receiving it would lead to greatly improved control performance; hence, the update has a high pragmatic VoI, and should be transmitted with a high priority.

The equivalences between AoI, semantic VoI, and pragmatic VoI, which are Level A, B, and C metrics, respectively, are listed in Table I. However, timing metrics such as AoI can still be extremely useful as a first approximation of higher-level metrics: in most scenarios, sharp variations are relatively rare, and can be treated as anomalies in a relatively simple weighted AoI minimization problem.

C. Real-time inference over communication links

Inference at the network edge will empower a broad spectrum of applications, such as industrial automation, extended reality, autonomous vehicles, and robots [14]. *Edge inference* refers to the process of offloading inference tasks, executed through large-scale AI models, from edge devices to edge servers, which are co-located and connected to base stations. By leveraging the powerful computational resources of these servers, edge devices can experience enhanced capabilities (e.g., vision, decision making, and natural language processing) and extended battery life. A real-time edge inference can support tactile applications, such as extended reality and robotic control. The major challenge to this paradigm is the communication bottleneck resulting from a massive number of mobile devices uploading high-dimensional data features to servers for inference.

A specific edge inference architecture is *split inference*, which divides a trained model into a low-complexity sub-model on a device and a deep sub-model on a server. The low-complexity sub-model extracts features from raw data, while the server sub-model performs inference on the uploaded features. This architecture enables resource-constrained edge devices to access large-scale AI models on servers while preserving data ownership. In split inference, the communication bottleneck is addressed by designing task-oriented techniques, aimed to optimize end-to-end (E2E) inference throughput, accuracy, or latency. To overcome communication constraints, the model's splitting point can be adapted according to available bandwidth and latency requirements [15]. For optimizing E2E system performance, JSCC can be employed in split inference to transmit features over the wireless edge. This method follows the same approach as DeepJSCC [8], with the objective of maximizing inference accuracy at the receiver rather than the fidelity of reconstructed features [16], which is an example of semantic communication.

1) Accelerating Edge Inference by Batching and Early Exiting: In the field of computing architecture, the well-known von Neumann bottleneck refers to the frequent shuttling of data between memory and processors, which

can account for up to 90% of total computation energy consumption and latency. One technique to address this issue is called batching, which consolidates tasks offloaded by multiple users into a single batch for parallel execution. This process increases the number of tasks executed per unit time by amortizing memory access time and enhancing the utilization of computational resources. Consequently, the latency per task is reduced, facilitating real-time edge inference. However, an excessively large batch size can result in increased queuing time for individual users, leading to reduced throughput.

Another technique that can reduce inference latency is called *early exiting*, which allows a task to exit a DNN once it meets the task-specific accuracy requirement. By avoiding the traversal of all network layers uniformly for all tasks, execution speeds are accelerated. Early exiting is characterized by a trade-off between accuracy and computation latency, which is based on a customized DNN architecture known as a backbone network [17]. This architecture consists of a conventional DNN augmented with multiple low-complexity intermediate classifiers, serving as candidate exit points. A task that requires lower accuracy traverses fewer network layers before exiting, essentially being diverted to an intermediate classifier for immediate inference. Recently, early exiting has been applied to edge inference, specifically in the local/server model partitioning for split inference. Exit points can be jointly optimized to maximize inference accuracy under a latency constraint, while layer skipping and early exiting are combined to facilitate edge inference with stringent resource constraints.

Joint design of batching and early exiting with radio resource management can further enhance E2E latency performance, which combines multiple access and inference latency. Specifically, the advantages of end-to-end design are twofold. First, the early-feedback effect, i.e., the instantaneous downloading of inference results upon early exits to intended users, thereby shortening their task latency—is further complemented by batched parallel processing. Second, early exits release computational resources to accelerate other ongoing tasks in the same batch, as reflected by a progressively shrinking batch size over blocks of model layers, which in turn accelerates computation over these blocks. However, the complex edge-inference process renders joint communication-and-computing (C2) resource management challenging in at least three ways. *(1)* The interweaving of batching and early exiting, where the latter results in a shrinking batch size over sequential layer blocks, causing variable computation time for different tasks. *(2)* The task executions of scheduled users have complex interdependencies, not least due to spectrum sharing for communication. Specifically, user's accuracy requirements are translated into a variable number of layer blocks to be traversed, while the computation time of each block depends on the random number of tasks passing through it. *(3)* The end-to-end latency constraints cause coupling of communication and computation, necessitating joint consideration of channel states and quality of service (QoS) requirements for batching/scheduling and bandwidth allocation. The optimization of C2 resource allocation requires solving an integer programming problem that is NP-complete. These challenges present numerous research opportunities in real-time edge inference, such as formulation of the joint optimal control of batching, early exiting, and radio resource management as a combinatorial optimization problem.

V. Massive Closed-Loop Communication in 6G

Massive communication, in which a very large number of devices communicate with a single base station, is typically associated to the uplink scenario, in which a large number of devices transmit messages to a single receiver. However, as we start to consider semantic and pragmatic communication with a massive number of devices, we move towards a setting in which uplink and downlink are of equal importance and operate in closed-loop. In this section, we will first explain the general role of massive downlink communication in 6G, and then provide an overview of distributed edge inference and learning as an example of massive closed-loop communication relying both on uplink and downlink.

A. The Role of Massive Downlink Communication

In 6G applications such as robotic control, distributed learning and edge inference, downlink communication appears as a central feature of massive communication alongside traditional massive uplink. In some cases, e.g., in distributed learning, the information transmitted in the downlink tends to be the same for a large group of devices and can be realized by a simple broadcast protocol. However, in the case of robotic control or a remote MDP, the edge server typically has a specific piece of information that it intends to deliver to a given agent, e.g., based on its own and other agents' states. In current systems, such individual downlink communication often comes at a high cost in terms of overhead, which can be prohibitive when only a small amount of information needs to be communicated (e.g., a robotic action).

This massive downlink setting with individual information leads to a two-way generalization of the uplink massive random access problem, in which the set of active devices is random. Towards this, we note that the

This article has been accepted for publication in IEEE BITS the Information Theory Magazine. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/MBITS.2023.3322667

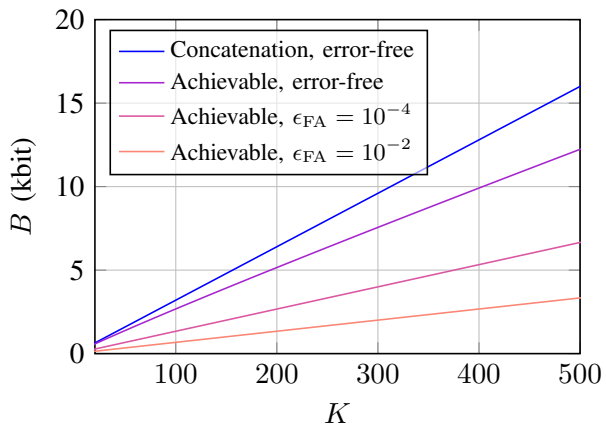

Fig. 3. Message length, B, required to encode acknowledgment feedback to K out of 2^{32} users, either error-free or with false alarm probability ϵ_{FA}.

fundamental characteristic of massive communication is the fact that the users are *uncoordinated*. Specifically, instead of considering only massive random access, we can think of *massive uncoordinated communication* as a generalized communication scenario involving both uplink and downlink in which there is high uncertainty about who is active (or relevant) at any given time.

To illustrate the idea and trade-offs for the massive downlink, it is instructive to consider the simple task of communicating 1-bit message acknowledments to a subset of the devices as part of an automatic repeat request (ARQ) feedback scheme. In this case, while the base station knows which devices it successfully received a message from, it does not know the subset of users that it failed to decode, e.g., due to outage. This is in sharp contrast to traditional ARQ in coordinated (scheduled) communication, where the base station knows if it failed to decode a message from a specific user. A consequence of this uncertainty is that the simple task of communicating what used to be a single bit of information, indicating whether the packet was received or not, becomes a highly non-trivial problem. As it turns out, simply concatenating the information for each user with their identifiers is sub-optimal, as illustrated in Fig. 3 [18]. This sub-optimality of concatenation in the massive downlink scenario holds not only for the ARQ problem, but also in the case of transmitting general messages to a small subset of a large population of users [19].

The previous example illustrates the challenge of massive uncoordinated communication at Level A, where the aim is to increase the reliability of the individual messages. Semantic and pragmatic communication point towards alternative ways to make use of massive downlink. For instance, one could construct a semantic ARQ scheme in which the receiver must keep requesting messages until the intended meaning has been correctly received. This requires an acknowledgment message that does not simply inform whether a given message is received or not, but instead tells what is known and unknown to the receiver. The prototypical example could be a surveillance scenario, in which the goal of the receiver is to detect all the objects observed by a number of wireless cameras, e.g., as part of an inference task. By acknowledging the detection of specific objects, as opposed to the individual captured images, a single message can acknowledge all the cameras that capture that specific object. At the pragmatic level, massive downlink is, for instance, relevant for controlling a large fleet of remote MDPs, where joint encoding of individual actions might reduce the communication cost at the expense of deviating from the optimal policy. Since the encoding schemes need to be tailored to the specific application and goal, their construction represents an opportunity for future research.

B. Distributed Inference and Learning at the Edge

The massive access problem was originally perceived for a massively large number of sensory nodes offloading their occasional bursty measurements in an efficient manner. These measurements are often employed at an edge or cloud server for certain downstream tasks involving inference or learning. However, as the number of sensors and the amount of data collected by them increases, there is a trend towards distributing the learning and inference tasks due to the increasing communication load and privacy concerns.

6G is likely to give rise to a widespread deployment of edge AI to enhance IoT applications [14] and large-scale distributed sensing enabled by cross-network collaboration among edge devices. The natural integration of edge AI and network sensing, known as AI of Things (AIoT) sensing, leverages the advantages of multi-view observations and/or distributed models across many devices, and the powerful prediction capabilities of DNN models to enable accurate and intelligent sensing at the edge. This will require combining features/inferences of multiple edge devices transmitted to an edge server over a multiple access channel, or enabling distributed collaborative training across wireless devices.

Both distributed learning and inference problems require the receiver to carry out computations based on the received features or models. Both scenarios can be considered as examples of semantic communication where the goal is not to convey the underlying features or models reliably, but to enable the receiver to achieve high accuracy in the desired learning task. Over-the-air computing (AirComp) has emerged as a promising simultaneous-access technique that enables ultra-fast

computations over-the-air by leveraging the waveform-superposition property of a multiple access channel [20], [21]. The primary motivation behind AirComp is to overcome channel distortion and noise, allowing for accurate implementation of data averaging or other computational functions. Next we briefly highlight how AirComp can be used to achieve efficient distributed learning and inference in the presence of massive number of devices.

1) Distributed Learning in Massive Access Scenarios: Federated learning (FL) has emerged as a natural framework for collaborative learning, where an iterative learning algorithm is carried out in a distributed fashion without offloading the data to a central server (see Fig. 4 for an illustration). The devices participating in FL run the necessary iterations locally on their private data, and share only the computed model updates with the parameter server (PS) at each iteration. The PS, which is the orchestrator of the learning process, aggregates the model updates from the participating devices, and updates the global model, which is then shared with the devices for the next round. This iterative learning process continues for a fixed number of rounds, or until a certain convergence condition is met.

FL can be implemented across hundreds or even thousands of devices. In certain scenarios, those devices may be in the same physical environment orchestrated by a single PS, which may be a macro base station or a WiFi access point, called federated edge learning (FEEL). This can be the case for: sensors deployed in the same environment learning a task based on their measurements; mobile devices in the same environment collaboratively learning a communication protocol; or a communication-related task, fine-tuned to the characteristics of the physical environment, e.g., millimeter-wave beam alignment based on LIDAR measurements.

2) FEEL with AirComp: While FEEL with a large number of sensors can be seen as a massive access problem, there is an important distinction. In the classical massive access problem, the goal of the receiver is to decode all the transmitted messages individually, even though the receiver may not be interested in the source of these messages. On the other hand, in the case of FL, the PS does not need individual model updates, as it only needs to recover their averages.

The computation problem is often treated separately from communication, although this is known to be suboptimal information-theoretically. This was first shown by Körner and Marton in [22] on a simple example, where the receiver wants to compute the parity of two correlated symmetric binary random variables. They were able to characterize the optimal rate region for this toy example, which was sufficient to illustrate

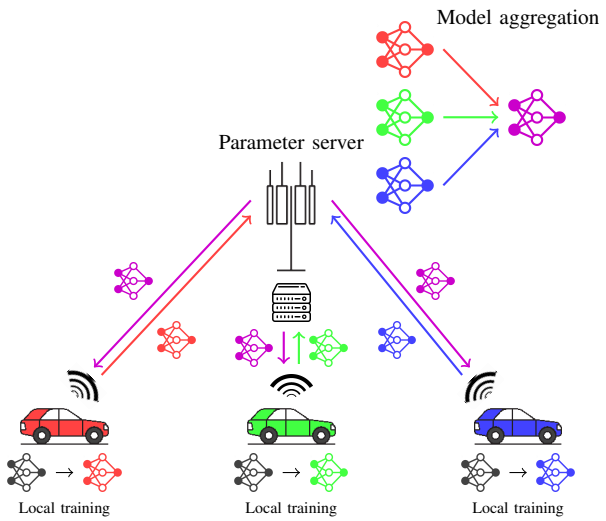

Fig. 4. Illustration of FEEL, where the participating devices upload their local updates to the PS through a multiple access channel, while the goal of the PS is to compute their average.

the suboptimality of first sending the sources to the receiver and then carrying out the computation task. Later, Orlitsky showed that the function computation problem is inherently different from conventional source coding by focusing on a point-to-point scenario in [23], where the receiver wants to compute a function of the source available at the transmitter and correlated side information available at the receiver. Furthermore, the optimal rate required for computing any function in a lossless fashion is given by the conditional G-entropy, where G is the characteristic graph of the two sources. This is more efficient than first sending the source to the receiver at a rate equal to its conditional entropy given the side information, and then computing the function.

In the case of FEEL, the goal is to compute the average rather than an arbitrary function. This is a good match for the wireless multiple access, as the wireless channel superposes the transmitted signals. Averaging is a nomographic function amenable for AirComp. Therefore, in the context of FEEL, wireless interference can be used favorably [21], rather than being mitigated, leading to improvement of both the speed and the final accuracy of learning tasks under a bandwidth constraint.

3) FEEL with UMA: Massive access problem is a generalization of the traditional multiple access problem to a massively large number of transmitters. This formulation adopts the common assumption of the multiple access problem, that the messages are chosen independently at the transmitters, and hence, two simultaneously active users are highly unlikely to transmit the same message. As a consequence, the goal at the receiver at each point in time reduces to estimating the number of active users, and decoding those many messages with-

This article has been accepted for publication in IEEE BITS the Information Theory Magazine. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/MBITS.2023.3322667

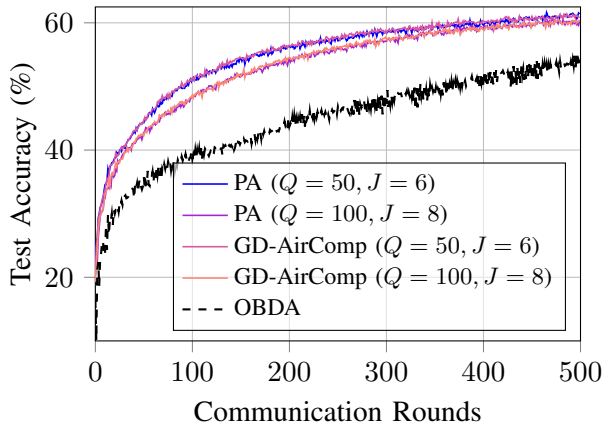

Fig. 5. Comparison of GD-AirComp with PA and OBDA schemes in terms of convergence speed and final test accuracy.

out necessarily associating them to particular devices. However, when devices are collaborating for a common task, such as distributed learning, their messages can be highly correlated. In this case, the assumption of each transmitter sending a different message is not valid any more, and the receiver must not only decide which messages were transmitted by the active transmitters, but also by how many transmitters.

For example, in the AirComp solution for the FEEL problem, the model updates are directly modulated to the amplitude of the transmitted symbols. This creates several problems. First, it requires strict synchronisation among devices. Second, it assumes continuous amplitude modulation, which is not possible in most practical communication systems, and may lead to problems such as power amplifier saturation or high peak-to-average power ratio. Last, but not least, AirComp requires sending one channel symbol for each model dimension, which corresponds to a very large bandwidth when training large models. In [24], an alternative digital scheme is proposed for efficient FEEL, based on unsourced massive access (UMA) [25]. Each device first vector quantizes its model update, which is then mapped to a channel codeword. Only a small subset of the devices may be active at each round, but this is not a problem since the PS is not interested in the identity of the active devices, as long as each device is active sufficiently often to contribute to the learning process. Therefore, UMA techniques can be used here to detect which updates are transmitted by the devices at each round, to be aggregated by the PS. However, due to correlation of the model updates, several active devices can transmit the same codeword. Hence, in addition to detecting the active messages, the PS should also count by how many active transmitters each message is transmitted.

In Fig. 5, we plot the convergence behaviour of test accuracy achieved by perfect accumulation (PA), which assumes perfect noise-free channel, one-bit digital accumulation (OBDA) scheme introduced in [26], and the UMA-based GD-AirComp scheme, proposed above, for the CIFAR-10 image classification task. Here, Q denotes the size of the compression blocks, where $Q = 1$ corresponds to scalar quantization, while $Q = 50, 100$ represent vector quantization over a block of symbols, and J refers to J-bit quantization; that is, a block of Q symbols is mapped to one of 2^J quantization codewords. For simplicity, we also set the size of the channel codewords transmitted for each block in GD-AirComp to Q. OBDA employs $Q = J = 1$ with AirComp rather than UMA. We can observe that GD-AirComp not only significantly improves the test accuracy compared to OBDA, but also meets the PA baseline in performance.

4) Distributed inference with AirComp: AirComp can also be used in edge inference to combine features and/or inferences of views/models distributed across multiple devices, which can provide privacy as well as accurate and efficient computation capability [27], [28].

In a multi-view scenario [29], features extracted from different sensors' views need to be combined into a global feature map, called multi-view pooling, which is then fed into the server's inference model. AirComp-based multi-view pooling is referred to as AirPooling. Implementing average AirPooling using AirComp is relatively straightforward; however, realizing max-pooling with AirComp is considerably more challenging. This is because the class of air-computable functions, known as nomographic functions, are characterized by a summation form with different pre-processing of summation terms and post-processing of the summation [30]. Examples include averaging and geometric mean. The max function in Max AirPooling is not nomographic and does not have a direct AirComp implementation. In [28], maximization is approximated by using the properties of the generalized p-norm:

$$\|x\|_p = \left(\sum_{n=1}^{N} |x_n|^p\right)^{\frac{1}{p}} \begin{cases} = \sum_{n=1}^{N} |x_n|, & p = 1, \\ \to \max\limits_{n} |x_n|, & p \to \infty. \end{cases} \quad (3)$$

One can see that when the configuration parameter, p, is set as one, the function implements averaging; when the parameter grows to infinity, the norm approximates maximization. Then a dual-mode AirPooling can be realized by decomposing the air-interface function into pre-processing at devices and post-processing at the server on top of the conventional AirComp and switching the pooling function by controlling p.

Consider the popular E2E sensing task of classification for object and pattern recognition. The configuration parameter of generated AirPooling should be optimised as

it regulates a trade-off between functional approximation and channel-noise suppression. On one hand, the max-function approximation error monotonically decreases as p grows. On the other hand, a larger p tends to amplify channel noise by making the transmitted features, x_n, highly skewed. Specifically, features with small magnitudes are suppressed and their submission is prone to channel distortion. The main difficulty in the parametric optimisation is the lack of closed-form expression for the E2E performance metric, e.g., sensing accuracy. One sub-optimal method as adopted in [28] is to use a surrogate metric such as the AirComp error.

VI. Conclusions and Outlook

We have presented a view on the wireless 6G systems and its associated technologies by extrapolating the evolution of massive and low-latency connectivity, which were two of the focal points in 5G. This evolution is put in the context of semantic and pragmatic communication in contrast to the technical problem of communication, which was the sole subject of the previous wireless generations. The paper argues that semantic and pragmatic communications have become relevant due to the increasing role that ML/AI play in communication systems. We presented how the low-latency requirement is expected to evolve towards generalized real-time requirements, coupled with cyber-physical systems and real-time inference. Finally, massive access is considered to evolve towards massive closed-loop communication used in interactive communication with wireless sensors and actuators, as well as in distributed learning. There are other aspects of 6G that were not covered in this article; for instance, in 3GPP standards there are already efforts for tighter coupling of the medium access layer performance to the real-time goals of the VR/AR traffic. Nevertheless, the principled discussion of semantic and pragmatic communication is receptive to be expanded with those other 6G aspects, such as integrated communication and sensing or non-terrestrial networks (NTN).

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, pp. 435–443, 2020.

[2] Q. Lan, D. Wen, Z. Zhang, Q. Zeng, X. Chen, P. Popovski, and K. Huang, "What is semantic communication? a view on conveying meaning in the era of machine intelligence," *J. Comm. Inf. Netw.*, vol. 6, no. 4, pp. 336–371, 2021.

[3] D. Gündüz, Z. Qin, I. E. Aguerri, H. S. Dhillon, Z. Yang, A. Yener, K. K. Wong, and C.-B. Chae, "Beyond transmitting bits: Context, semantics, and task-oriented communications," *IEEE J. Sel. Areas Comm.*, vol. 41, no. 1, pp. 5–41, 2022.

[4] P. Popovski, F. Chiariotti, K. Huang, A. E. Kalør, M. Kountouris, N. Pappas, and B. Soret, "A perspective on time toward wireless 6G," *Proc. IEEE*, vol. 110, no. 8, pp. 1116–1146, 2022.

[5] C. E. Shannon and W. Weaver, *The Mathematical Theory of Communication*. Urbana, IL: University of Illinois Press, 1949.

[6] J. B. Gleason and N. B. Ratner, *The Development of Language*. Plural Publishing, 2022.

[7] R. Dobrushin and B. Tsybakov, "Information transmission with additional noise," *IRE Trans. Inf. Theory*, vol. 8, no. 5, pp. 293–304, 1962.

[8] E. Bourtsoulatze, D. B. Kurka, and D. Gündüz, "Deep joint source-channel coding for wireless image transmission," *IEEE Trans. Cogn. Comm. Netw.*, vol. 5, no. 3, pp. 567–579, 2019.

[9] D. B. Kurka and D. Gündüz, "Bandwidth-agile image transmission with deep joint source-channel coding," *IEEE Trans. Wireless Comm.*, vol. 20, no. 12, pp. 8081–8095, 2021.

[10] T.-Y. Tung, S. Kobus, J. P. Roig, and D. Gündüz, "Effective communications: A joint learning and communication framework for multi-agent reinforcement learning over noisy channels," *IEEE J. Sel. Areas Comm.*, vol. 39, no. 8, pp. 2590–2603, 2021.

[11] S. Kobus, T.-Y. Tung, and D. Gündüz, "Goal-oriented compression with a constrained decoder," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2023.

[12] F. Alawad and F. A. Kraemer, "Value of information in wireless sensor network applications and the IoT: A review," *IEEE Sensors J.*, vol. 22, no. 10, pp. 9228–9245, 2022.

[13] J. Holm, F. Chiariotti, A. E. Kalør, B. Soret, T. B. Pedersen, and P. Popovski, "Goal-oriented scheduling in sensor networks with application timing awareness," *IEEE Trans. Comm.*, vol. 71, pp. 4513–4527, 2023.

[14] G. Zhu, D. Liu, Y. Du, C. You, J. Zhang, and K. Huang, "Toward an intelligent edge: Wireless communication meets machine learning," *IEEE Comm. Mag.*, vol. 58, no. 1, p. 19–25, 2020.

[15] E. Li, L. Zeng, Z. Zhou, and X. Chen, "Edge AI: On-demand accelerating deep neural network inference via edge computing," *IEEE Trans. Wireless Comm.*, vol. 19, no. 1, p. 447–457, 2020.

[16] M. Jankowski, D. Gündüz, and K. Mikolajczyk, "Wireless image retrieval at the edge," *IEEE J. Sel. Areas Comm.*, vol. 39, no. 1, pp. 89–100, 2021.

[17] T. Bolukbasi, J. Wang, O. Dekel, and V. Saligrama, "Adaptive neural networks for efficient inference," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2017.

[18] A. E. Kalør, R. Kotaba, and P. Popovski, "Common message acknowledgments: Massive ARQ protocols for wireless access," *IEEE Trans. Comm.*, vol. 70, no. 8, pp. 5258–5270, 2022.

[19] R. Song, K. M. Attiah, and W. Yu, "Coded downlink massive random access," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2023.

[20] G. Zhu, J. Xu, K. Huang, and S. Cui, "Over-the-air computing for wireless data aggregation in massive IoT," *IEEE Wireless Comm.*, vol. 28, no. 4, pp. 57–65, 2021.

[21] M. M. Amiri and D. Gündüz, "Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air," *IEEE Trans. Signal Process.*, vol. 68, pp. 2155–2169, 2020.

[22] J. Körner and K. Marton, "How to encode the modulo-two sum of binary sources (corresp.)," *IEEE Trans. Inf. Theory*, vol. 25, no. 2, pp. 219–221, 1979.

[23] A. Orlitsky and J. Roche, "Coding for computing," *IEEE Trans. Inf. Theory*, vol. 47, no. 3, pp. 903–917, 2001.

[24] L. Qiao, Z. Gao, Z. Li, and D. Gündüz, "Unsourced massive access-based digital over-the-air computation for efficient federated edge learning," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2023.

[25] Y. Polyanskiy, "A perspective on massive random-access," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2017, pp. 2523–2527.
[26] G. Zhu, Y. Du, D. Gündüz, and K. Huang, "One-bit over-the-air aggregation for communication-efficient federated edge learning: Design and convergence analysis," *IEEE Trans. Wireless Comm.*, vol. 20, no. 3, pp. 2120–2135, 2021.
[27] S. F. Yilmaz, B. Hasırcıoğlu, and D. Gündüz, "Over-the-air ensemble inference with model privacy," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2022.
[28] Z. Liu, Q. Lan, A. E. Kalør, P. Popovski, and K. Huang, "Over-the-air multi-view pooling for distributed sensing," *arXiv preprint arXiv:2302.09771*, 2023.
[29] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3D shape recognition," *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2015.
[30] M. Goldenbaum, H. Boche, and S. Stańczak, "Harnessing interference for analog function computation in wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 61, no. 20, pp. 4893–4906, 2013.