Neuromorphic Split Computing With Wake-Up Radios: Architecture and Design via Digital Twinning

Jiechen Chen, *Member, IEEE*, Sangwoo Park, *Member, IEEE*, Petar Popovski, *Fellow, IEEE*, H. Vincent Poor, *Life Fellow, IEEE*, and Osvaldo Simeone, *Fellow, IEEE*

***Abstract*—Neuromorphic computing leverages the sparsity of temporal data to reduce processing energy by activating a small subset of neurons and synapses at each time step. When deployed for split computing in edge-based systems, remote neuromorphic processing units (NPUs) can reduce the communication power budget by communicating asynchronously using sparse impulse radio (IR) waveforms. This way, the input signal sparsity translates directly into energy savings both in terms of computation and communication. However, with IR transmission, the main contributor to the overall energy consumption remains the power required to maintain the main radio on. This work proposes a novel architecture that integrates a wake-up radio mechanism within a split computing system consisting of remote, wirelessly connected, NPUs. A key challenge in the design of a wake-up radio-based neuromorphic split computing system is the selection of thresholds for sensing, wake-up signal detection, and decision making. To address this problem, as a second contribution, this work proposes a novel methodology that leverages the use of a digital twin (DT), i.e., a simulator, of the physical system, coupled with a sequential statistical testing approach known as Learn Then Test (LTT) to provide theoretical reliability guarantees. The proposed DT-LTT methodology is broadly applicable to other design problems, and is showcased here for neuromorphic communications. Experimental results validate the design and the analysis, confirming the theoretical reliability guarantees and illustrating trade-offs among reliability, energy consumption, and informativeness of the decisions.**

***Index Terms*—Neuromorphic computing, spiking neural networks, wake-up radios, neuromorphic wireless communications, reliability.**

Received 2 April 2024; revised 19 July 2024 and 4 September 2024; accepted 13 September 2024. Date of publication 18 September 2024; date of current version 23 October 2024. This work was supported in part by the European Union's Horizon Europe Project CENTRIC under Grant 101096379, in part by the Open Fellowship of the EPSRC under Grant EP/W024101/1, in part by the EPSRC Project under Grant EP/X011852/1, in part by the Villum Investigator Grant "WATER" from the Velux Foundations, Denmark, and in part by the U.S. National Science Foundation under Grant ECCS-2335876. The associate editor coordinating the review of this article and approving it for publication was Dr. Joao Mota. *(Corresponding author: Sangwoo Park.)*

Jiechen Chen and Sangwoo Park are with the King's Communications, Learning and Information Processing (KCLIP) Lab within the Centre for Intelligent Information Processing Systems (CIIPS), Department of Engineering, King's College London, WC2R 2LS London, U.K. (e-mail: jiechen.chen@kcl.ac.uk; sangwoo.park@kcl.ac.uk).

Petar Popovski is with the Department of Electronic Systems, Aalborg University, 9100 Aalborg, Denmark (e-mail: petarp@es.aau.dk).

H. Vincent Poor is with the Department of Electrical and Computer Engineering, Princeton University, Princeton, NJ 08544 USA (e-mail: poor@princeton.edu).

Osvaldo Simeone is with the King's Communications, Learning and Information Processing (KCLIP) Lab within the Centre for Intelligent Information Processing Systems (CIIPS), Department of Engineering, King's College London, WC2R 2LS London, U.K., and also with the Department of Electronic Systems, Aalborg University, 9100 Aalborg, Denmark (e-mail: osvaldo.simeone@kcl.ac.uk).

Digital Object Identifier 10.1109/TSP.2024.3463210

I. Introduction

A. Context and Motivation

NEUROMORPHIC processing units (NPUs), such as Intel's Loihi or BrainChip's Akida, leverage the sparsity of temporal data to reduce processing energy by activating a small subset of neurons and synapses at each time step [1], [2]. This mechanism implements *spike*-based signaling, whereby information is exchanged in the timing of the synaptic activation. The opportunistic activation of neurons and synapses distinguishes NPUs from conventional deep learning accelerators such as graphical processing units (GPUs) or tensor processing units (TPUs), making NPUs particularly attractive for time-series data.

As illustrated in Fig. 1, when deployed for *split computing* in edge-based systems [3], [4], remote NPUs, each carrying out part of the computation, can reduce the communication power budget by communicating asynchronously using sparse *impulse radio* (IR) waveforms [5], [6], [7], a form of ultra-wide bandwidth (UWB) spread-spectrum signaling. After extensive research activity in the early 2000s (see, e.g., [8]), UWB has recently re-emerged as a prominent solution for low-power, low-range connectivity. For example, Apple has incorporated a UWB radio in the most recent iPhone models, from iPhone 11 onwards, for precision ranging [9]. Furthermore, the IEEE 802.15.4z standard includes the items Enhanced UWB Physical Layers (PHYs) and Associated Ranging Techniques which cover the reliability, accuracy, and security of UWB communications [10]. Additionally, several high-profile academic proposals have advocated for the use of impulse radio in next-generation wireless interfaces [11].

Using IR waveforms, the input signal's sparsity, which depends on the semantics of the information processing task, translates directly into energy savings both in terms of

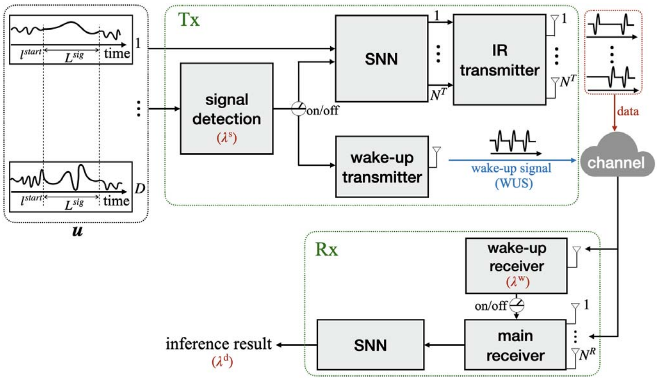


Fig. 1. In this work, we propose a low-power wake-up radio aided wireless split computing system, which operates through the following steps. (*i*) *Signal detection at the Tx*: the sensor captures a time-series data $\boldsymbol{u}$ for $L^{\max}$ time steps, containing meaningful information from an unknown time l^{start} for a duration of L^{sig}. A change detector is applied simultaneously to determine whether the sensed sequence contains a signal of interest. (*ii*) *Wake-up signal transmission*: If a signal of interest is detected at some specific time, the wake-up Tx and encoding SNN are turned on, and a WUS is transmitted by the wake-up Tx. (*iii*) *Data transmission*: after a fixed delay following the transmission of the WUS, the input signal $\{\boldsymbol{u}_l\}_{l=l^{\text{start}}}^{L^{\max}}$ is processed by the encoding NPU, and the output spikes are modulated using impulse radio (IR) and transmitted over the wireless channel to the main Rx. (*iv*) *Wake-up signal reception and activation of the main radio*: the WUS is detected by the wake-up Rx, leading to the activation of the main Rx. (*v*) *Decision Making*: upon waking up of the main Rx, the NPU at the receiver side processes the signal received by the main Rx to make an inference decision. Our goal is to optimize the threshold applied by signal detection, WUS detection, and decision making in order to provably control the average loss of the decision to a predetermined level, while minimizing the overall energy consumption.

computation and communication. This property has been leveraged in recent works such as [12] and [13] for innovative applications including the sensing of peripheral nerves and brain-computer interfaces. Both references above present hardware validations of the concept, with [13] reporting on a testbed involving 78 sensors (see also the news story[1]).

However, the power savings afforded by sparse transmitted signals are limited to the transmitter's side, which can transmit impulsive waveforms only at the times of synaptic activations. The main contributor to the overall energy consumption remains the power required to maintain the main radio on [14], [15], [16]. To address this architectural problem, as seen in Fig. 1, this work proposes a novel architecture that integrates a *wake-up radio* mechanism within a split computing system consisting of remote, wirelessly connected, NPUs.

Wake-up radios introduce a low-cost radio at the transmitter and at the receiver. The wake-up transmitter monitors the sensed signals, deciding when to transmit a *wake-up signal* (WUS) to the receiver. The wake-up receiver operates at a much reduced power as compared to the main receiver radio, and its sole purpose is detecting the WUS. Upon detection of the WUS, the main radio is activated [14], [15], [17], [18], [19].

A key challenge in the design of a wake-up radios is the selection of thresholds for sensing and WUS detection, and decision making. A conventional solution would be to calibrate the thresholds via *on-air* testing, trying out different thresholds via testing on the actual physical system. On-air calibration would be expensive in terms of spectral resources, and there is generally no guarantee that the selected thresholds would provide desirable performance levels for the end application.

To address this design problem, as illustrated in Fig. 2, this paper proposes a novel methodology that leverages the use of a *digital twin*, i.e., a simulator, of the physical system, coupled with a sequential statistical testing approach that provides theoretical reliability guarantees [20], [21].

B. Related Work

Neuromorphic communications: Neuromorphic communication, introduced in [5], integrates event-driven principles from neuromorphic computing into wireless communication systems for efficient sensing, communications, and decision-making. Reference [6] presented an architecture for wireless cognition that incorporates neuromorphic sensing, processing, and IR communications for multiple devices, leveraging time hopping for asynchronous multi-access. Motivated by the potential of IR for radar sensing [22], reference [7] introduced a neuromorphic integrated sensing and communication system, which targets simultaneous data transmission and target detection. In [23], a neuromorphic computing-based detector was implemented at a satellite receiver, whose goal was to detect Internet-of-Things signals in the presence of significant uplink interference. A hardware implementations of the system introduced in [6] was detailed in [13], showing the potential of the approach to scale to thousands of nodes. The solution presented in [13] leveraged energy harvesting.

Decentralized implementations of NPUs were studied in [24], while assuming conventional digital communications. A corresponding optimal resource allocation problem was investigated in [25].

Impulse radio for neuromorphic communications: IR has been proposed for wireless communication of digital packets between SNN chips in [26], and for transmitting time-encoded analog signals, similar to those measured by neuromorphic sensors, for biomedical applications in [27]. Additionally, a combination of neuromorphic sensing, time-based computing, and IR has been utilized in [28] to implement a consensus method based on device-to-device local communications for computing the maximum of scalar observations. In [5], [6], IR waveform were used to modulate the spiking signal for wireless transmission.

Wake-up radio: Wake-up radios can reduce energy consumption in wireless communication systems by keeping the main receiver radio off until an incoming signal of interest is detected [14]. In 3GPP Release 18, two wake-up receiver (WUR) architectures are introduced, using either a radio frequency envelope detector or an on-chip local oscillator approach [17]. The first type of architecture is characterized by low complexity, low cost, and extremely low energy consumption. In contrast, the second architecture requires more complex components, like on-chip local oscillators. This results in higher energy consumption, but the benefits include better sensitivity and robustness to interferers.

[1] https://spectrum.ieee.org/brain-machine-interface-2667619198

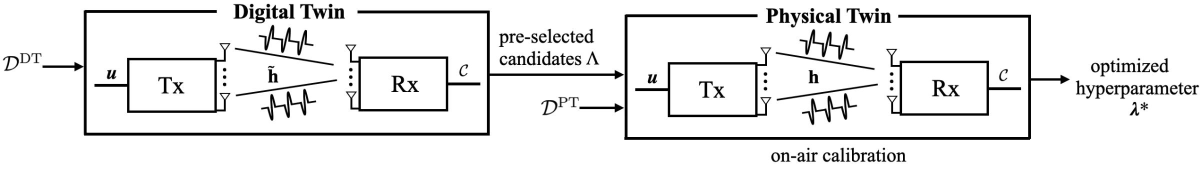


Fig. 2. Hyperparameters optimization is carried out by leveraging a dataset $\mathcal{D}$ of data examples, as well as access to a simulator of the channel implemented in a digital twin. The simulator produces channel variables $\tilde{\mathbf{h}}$ with a distribution $\tilde{p}(\mathbf{h})$ that is generally mismatched with respect to the true distribution $p(\mathbf{h})$. In a first phase, the digital twin uses the simulator to pre-select a subset Λ of candidate hyperparameters $\boldsymbol{\lambda}$. In a second phase, on-air calibration leverages transmission on the actual system (physical twin) to identify a solution $\boldsymbol{\lambda}^*$ that is guaranteed to satisfy the constraint in (25).

For the design of WUS, two main candidates in 3GPP Release 18 are on-off keying (OOK)-based WUS and OFDM-based WUS [17]. The OFDM-based signal structure does not require significant changes on the transmitter, while OOK-based WUS is an attractive choice for receivers with low complexity.

Wake-up radios have been integrated into a number of wireless systems. For example, in [29], a multi-access protocol was introduced that facilitates fully asynchronous communication among network devices, while reference [30] focused on WURs for wireless local area networks. Reference [15] proposed a neuromorphic enhanced WUR, tailored for brain-inspired applications using OOK-modulated WUSs.

Digital twins for wireless communication: Digital twinning is currently viewed as a promising enabling tool for the design and monitoring of next-generation wireless systems implementing machine learning modules [31]. For example, reference [32] proposed a Bayesian framework for the development of a DT platform aimed at the control, monitoring, and analysis of a multi-access communication system. The papers [33] and [34] proposed the use of digital twinning for the design of beam prediction and localization, respectively.

Guaranteed reliability for machine learning in wireless communications: Conformal prediction (CP) uses past experience to determine precise levels of confidence in new predictions [35]. This approach guarantees that, with a specified confidence level, future predictions will fall within the prediction regions, thereby providing reliable estimates of uncertainty. For the application of CP to wireless communication, [36] applied CP to the design of AI for communication systems in conjunction with both frequentist and Bayesian learning, focusing on the key tasks of demodulation, modulation classification, and channel prediction.

Learn then Test (LTT) is a framework for the selection of hyperparameters in pre-trained machine learning models that satisfy finite-sample statistical guarantees [21]. Like CP and conformal risk control (CRC), it relies on the use of calibration data, but it does not require the monotonicity assumption of CRC. As a result, it applies to more general settings, such as problems with multiple hyperparameters. Being a generic framework, LTT requires a dedicated effort to be tailored to a specific problem setting. To the best of our knowledge, ours is the first work that proposes a methodology for the application of LTT to the design of communications system.

Regarding the comparison with artificial neural networks (ANNs) in the context of wireless communication, reference [5] has shown that split computing based on SNNs and impulse radio can outperform frame-based ANN-based solutions, thanks to the benefits of event-driven communication and processing. Reference [6] extended these benefits to multi-device scenarios, for which impulse radio transmission can facilitate energy-efficient multi-access protocols. This was also verified experimentally by [13] using a testbed involving 78 sensors, built to operate according to the principles of neuromorphic communications. Against this background, our work offers additional energy saving advantages on top of those already reported in these papers by implementing a wake-up radio receiver.

TABLE I
Power Consumption for Split Computing Strategies Operating on Temporally Sparse Signals, e.g., for Monitoring Applications

Scheme (Communication/Computation)	Low Transmit-Power Duty Cycle	Low Receive-Power Duty Cycle
frame-based/ANNs	✗	✗
event-driven/SNNs [5], [6], [37]	✓	✗
event-driven/SNNs with wake-up radio (this work)	✓	✓

C. Main Contributions

The contribution of this paper is twofold. First, as shown in Fig. 1, we introduce a low-power wake-up radio aided neuromorphic wireless split computing architecture, whose goal is to carry out a remote inference task in an energy efficient way. Second, we propose a novel design methodology that combines LTT with digital twinning. This methodology, dubbed DT-LTT, enhances the spectral efficiency of a direct application of LTT [21] via a digital twin-based pre-selection of candidate thresholds for sensing, detection, and decision making. The main contributions of this paper are summarized as follows.

Architecture: We introduce a wake-up radio aided neuromorphic wireless split computing architecture, which combines the energy savings resulting from event-driven computing at the transmitter and receiver, as well as from IR transmission, with the energy savings made possible at the receiver via the introduction of a WUR. We summarize the merits of different split computing schemes in Table I, highlighting the capacity of the architecture proposed in this work to attain low energy consumption duty cycle at both transmitter and receiver via the introduction not only of IR at the transmitter but also of a WUR at the receiver.

As illustrated in Fig. 1, in the proposed architecture, the NPU at the transmitter side remains idle until a signal of interest is detected by the signal detection module. Subsequently, a WUS is transmitted by the wake-up transmitter over the channel to the wake-up receiver, which activates the main receiver. The IR transmitter modulates the encoded signals from the NPU, and sends them to the main receiver. The NPU at the receiver side then decodes the received signals and make an inference decision.

Digital twin-aided design methodology with reliability guarantees: In order to select the thresholds used at transmitter and receiver for sensing, WUS detection, and decision making, we propose a novel design methodology that integrates the LTT framework [21] with digital twinning. The proposed methodology, dubbed DT-LTT, is of broader interest as it can be applied to any communication system requiring the selection of hyperparameters via on-air transmission.

To explain, consider any setting that requires the selection of hyperparameters affecting the operation of a wireless link, here the mentioned thresholds. A direct application of LTT [21] would sequentially test candidate hyperparameters via the estimation of the target performance metrics through transmissions on the wireless channel. This way, the designer would be limited to testing a few candidate hyperparameters, given the limited availability of spectral resources.

To reduce the spectral overhead caused by hyperparameter calibration, we propose executing LTT through digital twinning. Specifically, the digital twin is leveraged to pre-select a sequence of hyperparameters to be tested using on-air calibration via LTT. The proposed DT-LTT calibration procedure is proved to guarantee reliability of the receiver's decisions irrespective of the fidelity of digital twin and of the data distribution. Indeed, the fidelity of the digital twin only affects the energy consumption and the informativeness of the output produced by the calibrated system. In this regard, the proposed method also supports the optimization of a weighted criterion involving energy consumption and informativeness of the receiver's decision.

Numerical evaluations: Extensive numerical results are provided that demonstrate the advantages of the proposed digital twin-based design approach.

D. Organization

The remainder of the paper is organized as follows. Section III presents the system model for the proposed wake-up radios assisted neuromorphic split computing system. Section IV describes the neuromorphic receiver processing with wake-up radio and the problem of interest, while the reliable hyperparameters optimization algorithm is proposed in Section V. Experimental setting and results are described in Section VI. Finally, Section VII concludes the paper.

II. Background

In this section, we provide background material that will be used in this work to introduce the proposed neuromorphic split computing system. Specifically, we first review reliable decision-making via prediction sets and CP [35]; and then we discuss hyperparameter optimization via multiple hypothesis testing [21].

A. Reliable Decision-Making via Prediction Sets

Reliable decision-making in machine learning requires not only accurate predictions but also a quantification of the uncertainty associated with the predictions. Conventional models often provide point predictions, which, while useful, fail to convey the uncertainty inherent in the model's decision-making process. This subsection reviews CP as a statistical method to calibrate prediction sets to ensure finite-sample coverage guarantees.

In classification problems, a machine learning model is trained to map an input $\boldsymbol{u}$ into one out of a discrete set $\{1,\ldots,C\}$ of class labels. The goal is to predict the most likely class $\hat{c}$ given a new input $\boldsymbol{u}$, along with a confidence score. It is well known that machine learning models, particularly with larger and potentially more accurate architectures, tend to be overconfident, offering an unreliable estimate of their uncertainty [38], [39], [40].

CP addresses this limitation by providing a set of possible outcomes $\mathcal{C}$ that are statistically likely to contain the true class label c with a specified confidence level $1-\alpha$, i.e.,

$$\Pr(c \in \mathcal{C}) \geq 1-\alpha. \tag{1}$$

CP leverages the scores s_c associated by the underlying model to each class c. Scores s_c are assumed here to be negatively oriented, i.e., they are smaller for classes on which the model is most confident. An example is given by the standard log-loss [41]. Given the scores s_c for all classes $c \in \{1,\ldots,C\}$, CP constructs the predicted set by including all classes whose score is below a threshold λ^{d} as

$$\mathcal{C} = \{c \in \{1,\ldots,C\} : s_c \leq \lambda^{\mathrm{d}}\}, \tag{2}$$

where the threshold λ^{d} is obtained based on a held-out calibration set. As detailed in Section IV-C, in this work, we treat the threshold λ^{d} as one of the hyperparameters to be optimized by the system.

B. Reliable Hyperparameter Optimization via Multiple-Hypothesis Testing

In this subsection, we introduce LTT, a reliable hyperparameter optimization framework based on multiple-hypothesis testing. Consider a machine learning model whose operation is controlled by a hyperparameter vector $\boldsymbol{\lambda}$, such as the learning rate for fine-tuning or the temperature in generative models [42]. LTT searches through a pre-defined set of candidate hyperparameter vectors $\Lambda = \{\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \ldots, \boldsymbol{\lambda}_{|\Lambda|}\}$ to produce a subset of hyperparameters that are guaranteed to control the risk of the system.

To elaborate, define as $R(\boldsymbol{\lambda})$ a population risk measure that we wish to control, such as the probability of a classification error. LTT associates with each candidate hyperparameter $\boldsymbol{\lambda}_j \in \Lambda$, with $j = 1,\ldots,|\Lambda|$, the null hypothesis

$$\mathcal{H}(\boldsymbol{\lambda}_j) : R(\boldsymbol{\lambda}_j) > \alpha, \tag{3}$$

where α is the maximum tolerated risk. Accordingly, the null hypothesis $\mathcal{H}(\boldsymbol{\lambda}_j)$ posits that hyperparameter $\boldsymbol{\lambda}_j$ is unreliable. Rejecting this hypothesis hence entails a decision that hyperparameter $\boldsymbol{\lambda}_j$ is reliable, in the sense that it meets the reliability condition $R(\boldsymbol{\lambda}_j) \leq \alpha$.

The goal of LTT is to identify a subset $\Lambda^{\text{rel}} \subseteq \Lambda$ of hyperparameter vectors such that the condition

$$\Pr[\exists \boldsymbol{\lambda} \in \Lambda^{\text{rel}} \text{ s.t. } R(\boldsymbol{\lambda}) > \alpha] \leq 1 - \delta \tag{4}$$

is satisfied for some target outage probability δ. Accordingly, the identified set of hyperparameters Λ^{rel} contains no unreliable hyperparameter $\boldsymbol{\lambda}$ with probability at least $1 - \delta$.

LTT relies on the evaluation of a p-value $p(\boldsymbol{\lambda}_j)$ for each null hypothesis $\mathcal{H}(\boldsymbol{\lambda}_j)$ [43], and hence for each candidate hyperparameter $\boldsymbol{\lambda}_j$. To this end, an empirical estimate $\hat{R}(\boldsymbol{\lambda}_j)$ of the risk $R(\boldsymbol{\lambda}_j)$ is obtained by using existing data or real-world testing. The p-value measures the probability of obtaining an estimate at least as small as $\hat{R}(\boldsymbol{\lambda}_j)$ when assuming the validity of the null hypothesis $\mathcal{H}(\boldsymbol{\lambda}_j)$ that the hyperparameter $\boldsymbol{\lambda}_j$ is not reliable. The p-values are then combined using methods for the control of the family-wise error rate (FWER) such as Bonferroni or fixed sequence testing. In this work, we will leverage fixed sequence testing, which tests hyperparameters sequentially. As further detailed in Section V-C, the testing order is ideally selected to consider hyperparameters in order of decreasing expected reliability.

III. System Model

As shown in Fig. 1, we consider an end-to-end neuromorphic remote inference system, in which the receiver (Rx) collects information from a device in order to carry out a semantic task, such as segmentation.

At the device, also referred to as transmitter (Tx), the sensor monitors the environment continuously to detect the start of a signal of interest. When the Tx detects a semantically relevant signal, the wake-up Tx is turned on to transmit the WUS, and the encoding NPU is also activated to process the input signal. The output of the NPU is buffered, and subsequently modulated and transmitted by the IR Tx after a given delay. Upon detecting the WUS, the wake-up Rx activates the main Rx, which starts receiving after a given delay. The received signal is then processed by a decoding NPU, which produces a final decision.

In this way, the proposed architecture combines the energy savings resulting from event-driven computing at Tx and Rx, as well as from IR transmission, with the energy savings made possible at the Rx via the introduction of a WUR.

We observe that, throughout this study, the presence of NPUs at the transmitter and receiver is accounted for by considering neural models that are suitable for implementation on neuromorphic hardware. This is detailed in the next section, and it follows the approach adopted in most works in the field such as [6], [44]. Note that libraries such as Intel's Lava also simulate the operation of NPUs by implementing suitable spiking neural models. We leave it as future work to present a full implementation integrating software-defined radios, neuromorphic hardware, and neuromorphic sensors (see also [13], [45] for some initial work in this direction).

A. Sensing Model

We assume that the relevant discrete-time signal captured by the sensor has a duration of L^{sig} samples, with each sample $\boldsymbol{u}_l$ being a D-dimensional vector. The duration L^{sig} is assumed to be known and deterministic. The signal of interest is semantically associated with label information c. We assume that the labels take values in a finite discrete set, but extensions to continuous quantities are direct. Furthermore, the signal is produced by an information source after a random delay of l^{start} time instants. Specifically, during an initial random period of $l^{\text{start}} - 1$ samples, the device observes a signal containing semantically irrelevant information, e.g., noise. The samples of the signal of interest is presented to the device starting at time l^{start}. Subsequently, the device again records irrelevant signals.

The sensor is active for a period of time equal to $L^{\max} \geq L^{\text{sig}}$ samples. The choice of $L^{\max}$ entails a trade-off between energy consumption and probability of fully observing the signal of interest of duration L^{sig}.

The sensed samples $\boldsymbol{u}_l$ for $l = 1, 2, \ldots$, are processed continuously by a *signal detector* at the Tx to determine an estimate $\hat{l}^{\text{start}}$ of the time l^{start}. The signal detector updates a cumulative sum statistic S_l at each time l using the current sample $\boldsymbol{u}_l$ via an algorithm such as QUSUM [46] or non-parametric change detection [47]. A change is detected at time l if the statistics S_l exceeds a threshold λ^{s}, i.e., $S_l > \lambda^{\text{s}}$, and thus the wake-up Tx and encoding NPU are activated at time

$$\hat{l}^{\text{start}} = \min_{l \in \{1, \ldots, L^{\max}\}} \{S_l > \lambda^{\text{s}}\}, \tag{5}$$

where the threshold λ^{s} is subject to optimization.

B. Neuromorphic Encoding

Upon activation of the wake-up Tx at time $\hat{l}^{\text{start}}$ in (5), an OOK-based WUS is transmitted for duration of L^{w} time steps. Following standard practice [14], as shown in Fig. 3 (top panel), data is then transmitted L^{d} time steps after the end of the WUS by IR Tx. The delay L^{d} accommodates channel delay spread, detection time of the wake-up Rx, as well as the wake-up latency of the main Rx [14].

The encoding NPU processes samples $\boldsymbol{u}_l$ starting from time $\hat{l}^{\text{start}}$. For each time instant $l \in [\hat{l}^{\text{start}}, L^{\max}] = \hat{l}^{\text{start}}, \hat{l}^{\text{start}} + 1, \ldots, L^{\max}$, the encoding NPU produces an $N^{\text{T}} \times 1$ vector

$$\boldsymbol{x}_l = f_{\boldsymbol{\theta}^e}(\boldsymbol{u}_l) \tag{6}$$

from its N^{T} readout neurons. In (6), the vector $\boldsymbol{\theta}^e$ is the parameter vector of the encoding NPU. The output spiking vectors $\boldsymbol{x}_l$ for $l \in [\hat{l}^{\text{start}}, L^{\max}]$ are buffered and transmitted in a first-in-first-out manner starting at time $\hat{l}^{\text{start}} + L^{\text{w}} + L^{\text{d}}$, i.e., after the transmission of the WUS and the delay L^{d}.

C. IR Transmission Model

The wake-up Tx is equipped with one antenna, while the IR transmitter has N^{T} antennas. Both transmitters adopts

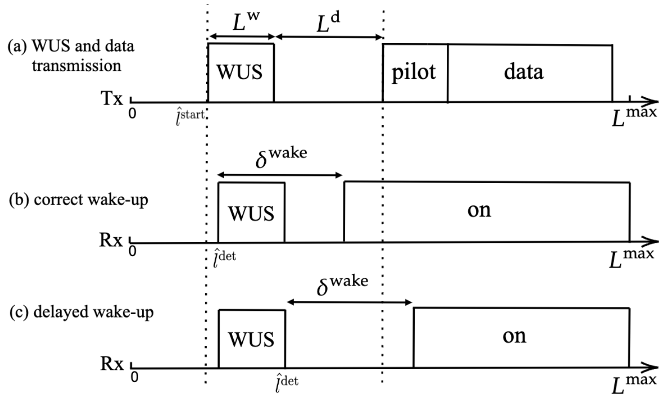


Fig. 3. Illustration of the working flow of the Tx and Rx. (a) WUS and data transmission: the WUS is sent by the wake-up Tx once the signal of interest is detected at time $\hat{l}^{\mathrm{start}}$, followed by the transmission of the pilot and the data after L^{d} delay. (b) Correct wake-up: the wake-up receiver detects the WUS at time $\hat{l}^{\mathrm{det}}$ and activates the main receiver. The main receiver takes δ^{wake} time to be fully activated. Importantly, the wake-up time of the main receiver precedes the commencement of data transmission. (c) Delayed wake-up: in this scenario, the main receiver wakes up after the data transmission has initiated, leading to data loss.

IR to modulate their respective transmitted signal $s_w(t)$ and $\{s_i(t)\}_{i=1}^{N^{\mathrm{T}}}$. Note that this is not a requirement for the wake-up radio, and is assumed here to facilitate a low-complexity implementation. Bandwidth expansion, leveraging time hopping (TH) [16], is utilized to manage interference between antennas of the transmitting device during data transmission.

Accordingly, each time step l of the sensed signal $\boldsymbol{u}_l$ comprises $L^{\mathrm{b}} \geq 1$ chips on the radio channel, with each chip having a duration of T_c seconds. Consequently, each time step l spans $L^{\mathrm{b}}T_c$ seconds, and hence L^{b} is referred to as *bandwidth expansion factor*. The bandwidth expansion factor L^{b} serves as a tradeoff between latency and interference mitigation. Using TH, each ith antenna modulates the corresponding mth entry of vector $\boldsymbol{x}_l$ in (6) using random time shifts across the L^{b} chips of the lth time period. This introduces temporal separation to reduce interference.

1) WUS Transmission: To elaborate, the antenna at the wake-up Tx modulates the OOK-based WUS using IR at each time step $l \in [\hat{l}^{\mathrm{start}}, \hat{l}^{\mathrm{start}} + L^{\mathrm{w}} - 1]$. The OOK-based WUS $s_{\mathrm{w}}(t)$ is defined as [16]

$$s_{\mathrm{w}}(t) = \sum_{j=\hat{l}^{\mathrm{start}}}^{\hat{l}^{\mathrm{start}}+L^{\mathrm{w}}-1} x_j^{\mathrm{w}} \phi(t - jL^{\mathrm{b}}T_c), \tag{7}$$

where x_j^{w} represents the jth OOK symbol in the set $\{0, 1\}$, and $\phi(t)$ denotes the OOK pulse waveform with bandwidth $1/T_c$. The WUS $s_{\mathrm{w}}(t)$ is received over a multi-path fading channel impulse response $h_w(t)$ by the wake-up Rx as

$$w(t) = s_{\mathrm{w}}(t) * h_w(t) + z(t), \tag{8}$$

where $*$ denotes the convolutional operation and $z(t)$ is the white Gaussian noise with noise power N_0.

2) Pilot Transmission: As shown in Fig. 3(a), following a pre-introduced delay of L^{d} after the WUS transmission, the IR transmitter is activated. To facilitate the main receiver's adaptation to the frequency-selective channel conditions, the IR transmitter transmits pilots prior to the data transmission. The pilot symbols sent from the ith antenna have a length of L^{p} and are defined as

$$s_i^{\mathrm{p}}(t) = \sum_{j=\hat{l}^{\mathrm{start}}+L^{\mathrm{w}}+L^{\mathrm{d}}}^{\hat{l}^{\mathrm{start}}+L^{\mathrm{w}}+L^{\mathrm{d}}+L^{\mathrm{p}}-1} \phi(t - jL^{\mathrm{b}}T_c - c_{j,i}^{\mathrm{p}}T_c), \tag{9}$$

where $c_{j,i}^{\mathrm{p}} \in \{0, 1, \ldots, L^{\mathrm{b}} - 1\}$ is an integer for the jth pilot symbol transmitted from the ith antenna, representing the TH position within L^{b} chips. The pilot is transmitted over the multi-path fading channel impulse response $h_{i,n}(t)$, and is received at the nth receive antenna as

$$v_n^{\mathrm{p}}(t) = \sum_{i=1}^{N^{\mathrm{T}}} s_i^{\mathrm{p}}(t) * h_{i,n}(t) + z_n(t), \tag{10}$$

where $z_n(t)$ represents the white Gaussian noise at the nth receive antenna.

3) Data Transmission: Data transmission commences once all pilot symbols have been transmitted. Each ith antenna at the IR transmitter modulates entry $x_{l,i}$ of the vector $\boldsymbol{x}_l = (x_{l,1}, \ldots, x_{l,N^{\mathrm{T}}})^T$ in (6) at time $l \in [\hat{l}^{\mathrm{start}} + L^{\mathrm{w}} + L^{\mathrm{d}} + L^{\mathrm{p}}, \ldots, L^{\max}]$, into a continuous-time signal $s_i(t)$, e.g., using Gaussian monopulses, and TH as

$$s_i(t) = \sum_{j=\hat{l}^{\mathrm{start}}+L^{\mathrm{w}}+L^{\mathrm{d}}+L^{\mathrm{p}}}^{L^{\max}} x_{j,i} \cdot \phi(t - jL^{\mathrm{b}}T_c - c_{j,i}T_c), \tag{11}$$

where $c_{j,i}$ is a random integer between 0 and $L^{\mathrm{b}} - 1$, representing TH position for the ith antenna at the jth time step.

The modulated signal $s_i(t)$ is then transmitted over the multi-path fading channel impulse response $h_{i,n}(t)$ to the Rx, where the received signal at the nth receive antenna is obtained as the superposition

$$v_n(t) = \sum_{i=1}^{N^{\mathrm{T}}} s_i(t) * h_{i,n}(t) + z_n(t). \tag{12}$$

Note that this assume the delay L^{d} to be longer than the channel spread to avoid interference with the WUS.

IV. Neuromorphic Receiver Processing With a Wake-Up Radio

To save energy at the Rx, instead of keeping the main radio on continuously, the proposed system incorporates an ultra low-power wake-up Rx that monitors the ambient radio frequency (RF) environment and listens for the WUS via the received signal (8). This approach allows the Rx to remain in a low-power state for extended periods, activating the main radio only when a WUS is detected. In this section, we start by introducing the WUS detection process operated by the wake-up Rx, and then we describe how the main Rx operates after it has been activated. Finally, we mathematically formulate the design problem of interest, which consists of minimizing the main Rx power consumption and the informativeness of the inference while guaranteeing the desired level of reliability for the decision made at the Rx.

A. WUS Detection

The wake-up Rx is always on, and it applies a correlator to detect the WUS $s_{\mathrm{w}}(t)$ in (7) from the received signal $w(t)$ in (8) [14]. This is done via matched filtering, i.e., by evaluating the convolution between $w(t)$ and the complex conjugate of the WUS $s^*_{\mathrm{w}}(t)$ as

$$d(\tau) = \int_{-\infty}^{+\infty} w(t) s^*_{\mathrm{w}}(t-\tau) dt, \tag{13}$$

and by detecting the WUS at time τ if the absolute value of the matched filter output $d(\tau)$ in (13) is larger than some threshold λ^w, i.e.,

$$\hat{l}^{\mathrm{det}} = \min_{l \in [1,\ldots,L^{\max}]} \{|d(lL^{\mathrm{b}}T_c)| \geq \lambda^{\mathrm{w}}\}, \tag{14}$$

with threshold λ^{w} being subject to optimization. As a result, the wake-up time of the main Rx is given by $\hat{l}^{\mathrm{det}} + \delta^{\mathrm{wake}}$, where $\delta^{\mathrm{wake}} \leq L^{\mathrm{d}}$ denotes the time required by the main Rx to be turned on upon the reception of WUS.

The main Rx does not miss the start of the data packet (see Fig. 3(b)) as long as we have the inequality

$$\hat{l}^{\mathrm{det}} + \delta^{\mathrm{wake}} \leq \hat{l}^{\mathrm{start}} + L^{\mathrm{w}} + L^{\mathrm{d}}. \tag{15}$$

Otherwise, the wake-up Rx misses at least some of the transmitted samples (Fig. 3(c)).

B. Main Radio Processing

The main radio is equipped with N^{R} antennas, and it stays idle until time $\hat{l}^{\mathrm{det}} + \delta^{\mathrm{wake}}$. Upon waking up, the main receiver samples the received pilot signals $\{v^{\mathrm{p}}_n(t)\}_{n=1}^{N^{\mathrm{R}}}$ and the received data signals $\{v_n(t)\}_{n=1}^{N^{\mathrm{R}}}$ at each time l, obtaining discrete-time pilots $\boldsymbol{v}^{\mathrm{p}}_l = [\boldsymbol{v}^{\mathrm{p}}_{l,1}, \ldots, \boldsymbol{v}^{\mathrm{p}}_{l,N^{\mathrm{R}}}]$ and discrete-time data $\boldsymbol{v}_l = [\boldsymbol{v}_{l,1}, \ldots, \boldsymbol{v}_{l,N^{\mathrm{R}}}]$, respectively. Here, the nth element represents the collection of signals by the nth antenna for L^{b} chips at time l, i.e., $\boldsymbol{v}^{\mathrm{p}}_{l,n} = \{v^{\mathrm{p}}_n(jT_c)\}_{j\in\mathcal{I}_l}$ and $\boldsymbol{v}_{l,n} = \{v_n(jT_c)\}_{j\in\mathcal{I}_l}$, where $\mathcal{I}_l = \{(l-1)L_b + 1, \ldots, lL_b\}$.

1) Pilot Processing via Hypernetwork: A hypernetwork is a type of neural network that generates the weights for another neural network, which can enhance the adaptability of the other neural network to the channel conditions [6]. The target network in our setting is the decoding NPU.

Provided that the main radio has woken up in time, we assume knowledge of the time of arrival of the pilots. Accordingly, we begin by collecting all the received pilot symbols as $\boldsymbol{v}^{\mathrm{p}} = \{\boldsymbol{v}^{\mathrm{p}}_l\}_{l=\hat{l}^{\mathrm{start}}+L^{\mathrm{w}}+L^{\mathrm{d}}}^{\hat{l}^{\mathrm{start}}+L^{\mathrm{w}}+L^{\mathrm{d}}+L^{\mathrm{p}}-1}$. To process the received pilot, we implement a pre-trained hypernetwork parameterized by ψ, such as a deep neural network (DNN). This hypernetwork takes the pilot $\boldsymbol{v}^{\mathrm{p}}$ as input, and produces a vector ω as

$$\boldsymbol{\omega} = f_{\boldsymbol{\psi}}(\boldsymbol{v}^{\mathrm{p}}), \tag{16}$$

in which each element is a scaling factor for each neuron in the decoding NPU. Effectively, the hypernetwork subsumes the task of channel estimation by directly mapping pilots to receiver's parameters.

Specifically, the vector ω is composed of N_d sub-vectors as $\boldsymbol{\omega} = \{\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_{N_d}\}$, where N_d is also the number of layers in the decoding NPU. Each element $\boldsymbol{\omega}_s$ has a length equal to the number of neurons in layer s of the decoding NPU. Thus, the weight matrix $\tilde{\boldsymbol{\theta}}^d_s$ for layer s in the decoding NPU can be adjusted by the hypernetwork as

$$\boldsymbol{\theta}^d_s = \tilde{\boldsymbol{\theta}}^d_s \cdot \mathrm{diag}\{\boldsymbol{\omega}_s\}, \tag{17}$$

where $\mathrm{diag}\{\boldsymbol{\omega}_s\}$ is a diagonal matrix with main diagonal given by the vector $\boldsymbol{\omega}_s$. We collect the updated weights of the decoding NPU as $\boldsymbol{\theta}^d = \{\boldsymbol{\theta}^d_1, \ldots, \boldsymbol{\theta}^d_{N_d}\}$.

2) Information Decoding: The data signal $\boldsymbol{v}_l$ is fed to the NPU, which produces a $C \times 1$ vector

$$\boldsymbol{r}_l = f_{\boldsymbol{\theta}^d}(\boldsymbol{v}_l) \tag{18}$$

via C readout neurons. At the final time $L^{\max}$, the output of the decoding NPU is first processed to yield a decision variable. As a typical example, the $C \times 1$ spike count vector $\bar{\boldsymbol{r}}$ is obtained by first summing up all output signal $\{\boldsymbol{r}_l\}_{l=\hat{l}^{\mathrm{det}}+\delta^{\mathrm{wake}}}^{L^{\max}}$ from the C readout neurons as

$$\bar{\boldsymbol{r}} = \sum_{l'=\hat{l}^{\mathrm{det}}+\delta^{\mathrm{wake}}}^{L^{\max}} \boldsymbol{r}_{l'}. \tag{19}$$

Focusing on a classification problem, the decoding NPU applies softmax function to the spike count vector $\bar{\boldsymbol{r}}$ to obtain a probability vector $\boldsymbol{p} = [p_1, \ldots, p_C]$. A score is assigned to each class c using the log-loss as $s_c = -\log(p_c)$. The final decision is constructed in the form of a *decision set* that includes the classes whose scores are smaller than a given threshold λ^{d}, i.e., [48]

$$\mathcal{C} = \{c : s_c \leq \lambda^{\mathrm{d}}\}. \tag{20}$$

The use of a decision set supports reliable decision making, whereby the size of the decision set $\mathcal{C}$ can be determined as a function of the uncertainty of the decision [21], [49]. This way, in contrast to standard methods such as top-k prediction, the size $|\mathcal{C}|$ of the set is adapted to the difficulty of the input, providing a means to control the expected loss and to quantify the uncertainty.

C. Design Problem

Overall, the decision vector r in (19) produced by the decoding NPU at the receiver depends on the fading channels and noise experienced by WUS transmission as per (8) and by data transmission as per (12). We denote collectively all noise and channel variables as $\mathbf{h}$. While the variables in vector $\mathbf{h}$ cannot be controlled, the system can tune the hyperparameters $\boldsymbol{\lambda} = [\lambda^{\mathrm{s}}, \lambda^{\mathrm{w}}, \lambda^{\mathrm{d}}]$, dictating the threshold λ^{s} for input signal detection at the Tx as in (5); the threshold λ^{w} for WUS detection at the wake-up Rx as in (14); and the threshold λ^{d} for prediction (20).

As the predicted set $\mathcal{C}$ in (20) depends on the input data $\boldsymbol{u}$, the channel variables $\mathbf{h}$, and the hyperparameter vector $\boldsymbol{\lambda}$, we will explicitly denote it as $\mathcal{C}(\boldsymbol{u}, \mathbf{h}, \boldsymbol{\lambda})$. To define the problem of optimizing the hyperparameters $\boldsymbol{\lambda}$, we introduce a *loss function*

$\ell(c, \mathcal{C}(\boldsymbol{u}, \mathbf{h}, \boldsymbol{\lambda}))$ capturing the discrepancy between the true target variable c and the estimate $\mathcal{C}(\boldsymbol{u}, \mathbf{h}, \boldsymbol{\lambda})$. The corresponding *expected loss* is defined as

$$L(\boldsymbol{\lambda}) = \mathbb{E}[\ell(c, \mathcal{C}(\boldsymbol{u}, \mathbf{h}, \boldsymbol{\lambda}))], \tag{21}$$

where the expectation is taken with respect to the data distribution $p(\boldsymbol{u}, c)$ of the input-output pair $(\boldsymbol{u}, c)$, as well as over the distribution $p(\mathbf{h})$ of the channel variables $\mathbf{h}$.

Given pre-trained encoding and decoding NPUs, we wish to find hyperparameters $\boldsymbol{\lambda}$ that minimize the average energy consumption $E(\boldsymbol{\lambda})$ at the Rx main radio and the size of the predicted set $\mathcal{C}(\boldsymbol{u}, \mathbf{h}, \boldsymbol{\lambda})$, while controlling the expected loss $L(\boldsymbol{\lambda})$ at some predetermined level $\alpha \in [0, 1]$. Note that the focus on energy consumption of the main radio at the Rx is justified by the fact that it is typically the most significant contributor to the overall energy expenditure at the Rx [15].

The *average energy* $E(\boldsymbol{\lambda})$ consumed by the Rx main radio is evaluated as

$$E(\boldsymbol{\lambda}) = P^{\text{on}}(L^{\max} - \mathbb{E}[\hat{l}^{\text{det}}(\boldsymbol{u}, \mathbf{h}, \boldsymbol{\lambda})] - \delta^{\text{wake}} + 1), \tag{22}$$

with P^{on} being the per-time-step energy consumed by the main radio when it is on, and the expectation is computed with respect to the data distribution of the input $\boldsymbol{u}$ and the distribution of vector $\mathbf{h}$. In fact, as illustrated in Fig. 3, the Rx main radio is on for $L^{\max} - \hat{l}^{\text{det}}(\boldsymbol{u}, \mathbf{h}, \boldsymbol{\lambda}) - \delta^{\text{wake}} + 1$. The notation $\hat{l}^{\text{det}}(\boldsymbol{u}, \mathbf{h}, \boldsymbol{\lambda})$ is introduced in (22) to highlight the dependence of the detection time $\hat{l}^{\text{det}}$ on input $\boldsymbol{u}$, channel $\mathbf{h}$, and hyperparameter $\boldsymbol{\lambda}$.

A smaller energy consumption (22) can be obtained by waking up the main radio later, i.e., by maximizing the expected value $\mathbb{E}[\hat{l}^{\text{det}}(\boldsymbol{u}, \mathbf{h}, \boldsymbol{\lambda})]$, but this generally comes at the cost of an increased average loss $L(\boldsymbol{\lambda})$. To assess the informativeness of the predicted set $\mathcal{C}(\boldsymbol{u}, \mathbf{h}, \boldsymbol{\lambda})$, we evaluate the *average set size* as

$$I(\boldsymbol{\lambda}) = \mathbb{E}[|\mathcal{C}(\boldsymbol{u}, \mathbf{h}, \boldsymbol{\lambda})|], \tag{23}$$

where the expectation is taken with respect to the data distribution of the input $\boldsymbol{u}$ and the distribution of vector $\mathbf{h}$.

Overall, the design problem of interest is formulated as the constrained minimization

$$\begin{aligned} &\underset{\boldsymbol{\lambda}}{\text{minimize}}\ E(\boldsymbol{\lambda}) + \gamma I(\boldsymbol{\lambda}) \\ &\text{subject to } L(\boldsymbol{\lambda}) \leq \alpha, \end{aligned} \tag{24}$$

where $\gamma \geq 0$ is a weight factor determining the relative priority between the energy consumption $E(\boldsymbol{\lambda})$ and the set size $I(\boldsymbol{\lambda})$, while the parameter $\alpha > 0$ specifies the desired reliability level, with a smaller α indicating a stricter reliability requirement. Regarding the choice of parameter γ in (24), note that there is generally a tension between energy $E(\boldsymbol{\lambda})$, and set size $I(\boldsymbol{\lambda})$. In fact, reducing the set size $I(\boldsymbol{\lambda})$, while maintaining the desired target reliability α, generally requires a larger energy expenditure $E(\boldsymbol{\lambda})$.

V. DT-LTT: Hyperparameters Optimization For Energy-Efficient Risk Control

As discussed in the last section, the goal of this work is to introduce a methodology for the selection of hyperparameters $\boldsymbol{\lambda}$ by addressing problem (24). In this section, we describe the proposed solution based on digital twinning and LTT [21], a method recently introduced in statistics.

A. Digital Twin-Based Optimization

Addressing problem (24) is made complicated by the fact that we do not assume knowledge of the distribution $p(\boldsymbol{u}, c)$ of each data pair $(\boldsymbol{u}, c)$, consisting of sensed signal $\boldsymbol{u}$ and label c, and we also do not have access to the distribution $p(\mathbf{h})$ of the channel variables $\mathbf{h}$. To obtain information about the data distribution $p(\boldsymbol{u}, c)$, we make the common assumption that a dataset $\mathcal{D} = \{(\boldsymbol{u}_n, c_n)\}_{n=1}^{|\mathcal{D}|}$ is available, where each pair $(\boldsymbol{u}_n, c_n)$ of signal $\boldsymbol{u}_n$ and label c_n is generated in an independent and identically distributed (i.i.d.) manner from the distribution $p(\boldsymbol{u}, c)$. Note that each pair is thus produced under an independent channel realization from distribution $p(\mathbf{h})$. Furthermore, to facilitate the collection of information about the distribution $p(\mathbf{h})$ of the channel variables, we assume access to a simulator in a *digital twin* of the system. As illustrated in Fig. 2, the simulator can produce samples $\tilde{\mathbf{h}}$ from a distribution $\tilde{p}(\mathbf{h})$ that is generally different from the true distribution $p(\mathbf{h})$. The *fidelity* of the simulator depends on how similar the distribution $p(\mathbf{h})$ and $\tilde{p}(\mathbf{h})$ are.

With this information, DT-LTT aims at solving a relaxation of problem (24), in which the constraint is required to be satisfied with a user-determined probability $1 - \delta$ with $\delta \in (0, 1)$. The resulting problem is defined as

$$\begin{aligned} &\underset{\lambda}{\text{minimize}}\ E(\boldsymbol{\lambda}) + \gamma I(\boldsymbol{\lambda}) \\ &\text{subject to } \Pr\left[L(\boldsymbol{\lambda}) \leq \alpha\right] \geq 1 - \delta, \end{aligned} \tag{25}$$

where the probability $\Pr[\cdot]$ is taken with respect to the random realization of the dataset $\mathcal{D}$ and the channel $\mathbf{h}$. Note that the probability in (25) cannot be evaluated given that the distribution $p(\boldsymbol{u}, c)$ and $p(\mathbf{h})$ are unknown.

B. Digital Twin-Based Pre-Selection of Candidate Solutions

In order to address problem (25), we follow a two-stage approach illustrated in Fig. 2. In the first phase, the digital twin pre-selects a subset Λ of candidate hyperparameter vectors $\boldsymbol{\lambda}$. The pre-selected candidates in set Λ are then tested in the following phase of *on-air calibration* to identify a hyperparameter vector $\boldsymbol{\lambda}^*$ that provably satisfies the constraint in (25). Reducing the size of the candidate solutions via the use of the digital twin supports a more efficient use of the physical channel resources during on-air calibration, as fewer options need to be evaluated using transmission on the wireless channel.

At a technical level, as detailed in the Appendix, the proposed approach leverages the freedom in the LTT scheme reviewd in Section II to choose any fixed sequence of hyperparameter vectors for testing of the reliability condition (25). Our proposed method, DT-LTT, determines the sequence of hyperparameter vectors by leveraging a digital twin model.

To start, the dataset $\mathcal{D}$ is randomly partitioned into two subsets, namely the dataset $\mathcal{D}^{\text{DT}}$ to be used with the simulator produced by the digital twin and the dataset $\mathcal{D}^{\text{PT}}$ to be leveraged

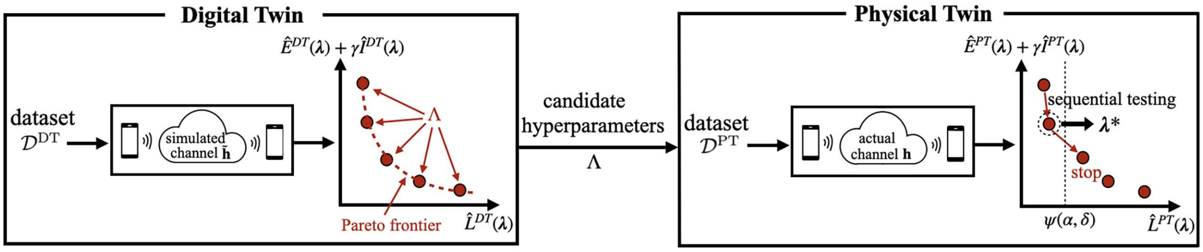

Digital Twin
$\hat{E}^{DT}(\boldsymbol{\lambda})+\gamma\hat{I}^{DT}(\boldsymbol{\lambda})$
dataset $\mathcal{D}^{\mathrm{DT}}$
simulated channel $\tilde{\mathbf{h}}$
Λ
Pareto frontier
$\hat{L}^{DT}(\boldsymbol{\lambda})$
candidate hyperparameters
Λ
Physical Twin
$\hat{E}^{PT}(\boldsymbol{\lambda})+\gamma\hat{I}^{PT}(\boldsymbol{\lambda})$
dataset $\mathcal{D}^{\mathrm{PT}}$
actual channel $\mathbf{h}$
sequential testing
$\boldsymbol{\lambda}^*$
stop
$\psi(\alpha,\delta)$
$\hat{L}^{PT}(\boldsymbol{\lambda})$

Fig. 4. Illustration of the proposed DT-LTT design strategy: during the first phase of pre-selection, the digital twin determines a subset of candidate hyperparameters that yield estimated $\hat{E}^{\mathrm{DT}}(\boldsymbol{\lambda})+\gamma\hat{I}^{\mathrm{DT}}(\boldsymbol{\lambda})$ and loss $\hat{L}^{\mathrm{DT}}(\boldsymbol{\lambda})$ on the Pareto frontier. Then, during on-air calibration, the physical twin transmits on the actual channel to test the candidates in set Λ sequentially, stopping when the estimated loss crosses a threshold $\psi(\alpha,\delta)$. The solution $\boldsymbol{\lambda}^*$ is then obtained by choosing the value of $\boldsymbol{\lambda}$ that yields the minimum estimated objective $\hat{E}^{\mathrm{PT}}(\boldsymbol{\lambda})+\gamma\hat{I}^{\mathrm{PT}}(\boldsymbol{\lambda})$, while guaranteeing the inequality $\hat{L}^{\mathrm{PT}}(\boldsymbol{\lambda})<\psi(\alpha,\delta)$.

for on-air calibration in the physical system. To carry out the pre-selection of a subset Λ of hyperparameter, the digital twin addresses the *multi-objective problem*

$$\underset{\boldsymbol{\lambda}}{\text{minimize}}\ \{\hat{L}^{\mathrm{DT}}(\boldsymbol{\lambda}),\hat{E}^{\mathrm{DT}}(\boldsymbol{\lambda})+\gamma\hat{I}^{\mathrm{DT}}(\boldsymbol{\lambda})\},\tag{26}$$

where the objectives $\hat{L}^{\mathrm{DT}}(\boldsymbol{\lambda})$, $\hat{E}^{\mathrm{DT}}(\boldsymbol{\lambda})$ and $\hat{I}^{\mathrm{DT}}(\boldsymbol{\lambda})$ are empirical estimates obtained at the digital twin for the expected loss [41]

$$\hat{L}^{\mathrm{DT}}(\boldsymbol{\lambda})=\frac{1}{|\mathcal{D}^{\mathrm{DT}}|}\sum_{n=1}^{|\mathcal{D}^{\mathrm{DT}}|}\ell\big(c,\mathcal{C}(\boldsymbol{u}_n,\tilde{\mathbf{h}}_n,\boldsymbol{\lambda})\big),\tag{27}$$

the average energy consumption

$$\begin{aligned}&\hat{E}^{\mathrm{DT}}(\boldsymbol{\lambda})\\&=P^{\mathrm{on}}\left(L^{\max}-\frac{1}{|\mathcal{D}^{\mathrm{DT}}|}\sum_{n=1}^{|\mathcal{D}^{\mathrm{DT}}|}\hat{l}^{\mathrm{det}}(\boldsymbol{u}_n,\tilde{\mathbf{h}}_n,\boldsymbol{\lambda})-\delta^{\mathrm{wake}}+1\right),\end{aligned}\tag{28}$$

and the average set size

$$\hat{I}^{\mathrm{DT}}(\boldsymbol{\lambda})=\frac{1}{|\mathcal{D}^{\mathrm{DT}}|}\sum_{n=1}^{|\mathcal{D}^{\mathrm{DT}}|}|\mathcal{C}(\boldsymbol{u}_n,\tilde{\mathbf{h}}_n,\boldsymbol{\lambda})|.\tag{29}$$

The empirical estimates (27), (28) and (29) are obtained by using the dataset $\mathcal{D}^{\mathrm{DT}}$ and transmission simulated using channels $\tilde{\mathbf{h}}_n\sim\tilde{p}(\mathbf{h})$ generated by digital twin. As shown in Fig. 4, the digital twin uses an arbitrary multi-objective optimization algorithm to identify a discrete subset Λ of values of the hyperparameter $\boldsymbol{\lambda}$ such that the resulting estimates $\big(\hat{L}^{\mathrm{DT}}(\boldsymbol{\lambda}),\hat{E}^{\mathrm{DT}}(\boldsymbol{\lambda})+\gamma\hat{I}^{\mathrm{DT}}(\boldsymbol{\lambda})\big)$ lie on the Pareto front of the set of achievable values for the pair $\big(\hat{L}^{\mathrm{DT}}(\boldsymbol{\lambda}),\hat{E}^{\mathrm{DT}}(\boldsymbol{\lambda})+\gamma\hat{I}^{\mathrm{DT}}(\boldsymbol{\lambda})\big)$. Mathematically, each vector $\boldsymbol{\lambda}$ included in the candidate set Λ satisfies the condition

$$\begin{aligned}&\nexists\boldsymbol{\lambda}'\text{ such that }\hat{L}^{\mathrm{DT}}(\boldsymbol{\lambda}')<\hat{L}^{\mathrm{DT}}(\boldsymbol{\lambda})\text{ and}\\&\quad\hat{E}^{\mathrm{DT}}(\boldsymbol{\lambda}')+\gamma\hat{I}^{\mathrm{DT}}(\boldsymbol{\lambda}')<\hat{E}^{\mathrm{DT}}(\boldsymbol{\lambda})+\gamma\hat{I}^{\mathrm{DT}}(\boldsymbol{\lambda})\end{aligned}\tag{30}$$

that no other hyperparameter $\boldsymbol{\lambda}'$ improves both empirical loss and empirical energy consumption plus the weighted set size.

C. On-Air Calibration

Given the pre-selected candidate solutions in set Λ, on-air calibration aims at selecting a value $\boldsymbol{\lambda}$ that approximately solves problem (26), ensuring the validity of the reliability constraint in (25). To this end, the solutions in set Λ are first ordered with respect to the loss value $\hat{L}^{\mathrm{DT}}(\boldsymbol{\lambda})$ in (27) as

$$\hat{L}^{\mathrm{DT}}(\boldsymbol{\lambda}_1)\le\hat{L}^{\mathrm{DT}}(\boldsymbol{\lambda}_2)\le\ldots\le\hat{L}^{\mathrm{DT}}(\boldsymbol{\lambda}_{|\Lambda|}).\tag{31}$$

On-air calibration evaluates the solutions in set Λ in the order $\boldsymbol{\lambda}_1,\boldsymbol{\lambda}_2,\ldots$, selecting a value $\boldsymbol{\lambda}^*$ that is guaranteed to satisfy constraint (25), while reducing as much as possible the weighted sum of energy consumption and set size.

For any hyperparameter $\boldsymbol{\lambda}_j$ being tested, using transmission on the actual physical channel, the physical twin evaluates empirical expected loss

$$\hat{L}^{\mathrm{PT}}(\boldsymbol{\lambda}_j)=\frac{1}{|\mathcal{D}^{\mathrm{PT}}|}\sum_{n=1}^{|\mathcal{D}^{\mathrm{PT}}|}\ell(c,\mathcal{C}(\boldsymbol{u}_n,\mathbf{h}_n,\boldsymbol{\lambda}_j)),\tag{32}$$

the empirical energy consumption

$$\begin{aligned}\hat{E}^{\mathrm{PT}}(\boldsymbol{\lambda}_j)=P^{\mathrm{on}}\Bigg(&L^{\max}-\frac{1}{|\mathcal{D}^{\mathrm{PT}}|}\sum_{n=1}^{|\mathcal{D}^{\mathrm{PT}}|}\hat{l}^{\mathrm{det}}(\boldsymbol{u}_n,\mathbf{h}_n,\boldsymbol{\lambda}_j)\\&-\delta^{\mathrm{wake}}+1\Bigg)\end{aligned}\tag{33}$$

and the empirical set size

$$\hat{I}^{\mathrm{PT}}(\boldsymbol{\lambda}_j)=\frac{1}{|\mathcal{D}^{\mathrm{PT}}|}\sum_{n=1}^{|\mathcal{D}^{\mathrm{PT}}|}|\mathcal{C}(\boldsymbol{u}_n,\mathbf{h}_n,\boldsymbol{\lambda}_j)|\tag{34}$$

by transmitting on actual channel realizations $\mathbf{h}_n\sim p(\mathbf{h})$. Note that the channel realization $\mathbf{h}_n$ is not known and not required to evaluate the estimates (32), (33) and (34). The estimates (32), (33) and (34) are evaluated successively for the candidate solutions $\boldsymbol{\lambda}_1,\boldsymbol{\lambda}_2,\ldots$, until a stopping criterion is satisfied.

Specifically, as illustrated in Fig. 4, the evaluation of candidate solutions $\boldsymbol{\lambda}_1,\boldsymbol{\lambda}_2,\ldots$ stops at the first value j^{stop} for which the estimated loss $\hat{L}^{\mathrm{PT}}(\boldsymbol{\lambda}_{j^{\mathrm{stop}}})$ in (32) exceeds the threshold

$$\psi(\alpha,\delta)=\alpha-\sqrt{\frac{-\ln(\delta)}{2|\mathcal{D}^{\mathrm{PT}}|}},\tag{35}$$

Algorithm 1: Digital Twin-Based Learn-then-Test (DT-LTT) Calibration

```
Initialization: Dataset $\mathcal{D}^{\rm DT}$, dataset $\mathcal{D}^{\rm PT}$, risk tolerance $\alpha \in [0,1]$, and error level $\delta \in [0,1]$
Digital Twin-based Pre-selection of Candidate Solutions:
Using the simulated channel $\tilde{\mathbf{h}} \sim p(\tilde{\mathbf{h}})$, identify a subset $\Lambda$ of the candidate solutions $\boldsymbol{\lambda}$ such that each $\boldsymbol{\lambda} \in \Lambda$ returns estimates $\big(\hat{L}^{\rm DT}(\boldsymbol{\lambda}), \hat{E}^{\rm DT}(\boldsymbol{\lambda}) + \gamma \hat{I}^{\rm DT}(\boldsymbol{\lambda})\big)$ in (27), (28) and (29) on the Pareto frontier.
On-Air Calibration:
Order the solutions in set $\Lambda$ as $\hat{L}^{\rm DT}(\boldsymbol{\lambda}_1) \leq \hat{L}^{\rm DT}(\boldsymbol{\lambda}_2) \leq \ldots \leq \hat{L}^{\rm DT}(\boldsymbol{\lambda}_{|\Lambda|})$.
for $j = 1, 2, \ldots, |\Lambda|$ do
    Estimate expected loss $\hat{L}^{\rm PT}(\boldsymbol{\lambda}_j)$, energy consumption $\hat{E}^{\rm PT}(\boldsymbol{\lambda}_j)$ and set size $\hat{I}^{\rm PT}(\boldsymbol{\lambda}_j)$ in (32), (33) and (34) using the actual channel $\mathbf{h} \sim p(\mathbf{h})$.
    if $j = 1$ and $\hat{L}^{\rm PT}(\boldsymbol{\lambda}_j) > \psi(\alpha, \delta)$
        Set $\boldsymbol{\lambda}^* = [\lambda^{\rm s} = \infty, \lambda^{\rm w} = \infty, \lambda^{\rm d} = \infty]^T$ (secure solution).
    else if $j > 1$ and $\hat{L}^{\rm PT}(\boldsymbol{\lambda}_j) > \psi(\alpha, \delta)$
        Set $\boldsymbol{\lambda}^*$ using (37).
    end if
end for
```

which is a function of the dataset size $|\mathcal{D}^{\rm PT}|$, of the target expected loss α in (25), and of the probability bound δ in (25). For the optimal hyperparameter $\boldsymbol{\lambda}^*$ to be well defined, one needs to ensure the condition

$$\hat{L}^{\rm PT}(\boldsymbol{\lambda}_1) < \psi(\alpha, \delta). \tag{36}$$

If condition (36) is not met, the decoding NPU makes a *secure* decision by including all classes in the predicted set $\mathcal{C}$ in (20), while saving energy by keeping the main receiver off. This amounts to the choice $\boldsymbol{\lambda}^* = [\lambda^{\rm s} = \infty, \lambda^{\rm w} = \infty, \lambda^{\rm d} = \infty]^T$.

Assuming that such value exists, finally, the selected value $\boldsymbol{\lambda}^*$ is obtained by choosing the value $\boldsymbol{\lambda}_j$ with $j \in \{1, \ldots, j^{\rm stop}\}$ that returns the smallest estimated sum $\hat{E}^{\rm PT}(\boldsymbol{\lambda}_j) + \gamma \hat{I}^{\rm PT}(\boldsymbol{\lambda}_j)$, i.e.,

$$\boldsymbol{\lambda}^* = \boldsymbol{\lambda}_{j^*}, \text{ with } j^* = \underset{j \in \{1, \ldots, j^{\rm stop}\}}{\arg\min} \{\hat{E}^{\rm PT}(\boldsymbol{\lambda}_j) + \gamma \hat{I}^{\rm PT}(\boldsymbol{\lambda}_j)\}. \tag{37}$$

The overall proposed calibration procedure is described in Algorithm 1. As proved next, by the properties of LTT [21], DT-LTT guarantees the constraint (25) irrespective of the true, unknown, distributions $p(\boldsymbol{u}, c)$ and $p(\mathbf{h})$, and irrespective of the fidelity of the digital twin.

Theorem 1 (Reliability of DT-LTT): By setting the hyperparameter vector $\boldsymbol{\lambda}^*$ as in Algorithm 1, DT-LTT satisfies the inequality

$$\Pr[L(\boldsymbol{\lambda}^*) \leq \alpha] \geq 1 - \delta \tag{38}$$

holds for any realizations of dataset $\mathcal{D}^{\rm DT}$, simulated channels $\{\tilde{\mathbf{h}}_n \sim \tilde{p}(\mathbf{h})\}_{n=1}^{|\mathcal{D}^{\rm DT}|}$, with probability in (38) evaluated with respect to the randomness of the dataset $\mathcal{D}^{\rm PT}$ and the true channels $\{\mathbf{h}_n \sim p(\mathbf{h})\}_{n=1}^{|\mathcal{D}^{\rm PT}|}$.

Proof: The proof is provided in the Appendix. □

VI. Experiments

In this section, we present numerical results that validate the proposed design and analysis.

A. Setting

To test the proposed DT-LTT calibration method, we consider a neuromorphic wireless communication link over a multi-path fading channel, whose goal is to support reliable image classification at the receiver. The transmitter is equipped with $N^{\rm T} = 10$ antennas, each modulating the spiking signal produced by the corresponding neuron of the encoding NPU, while the receiver has $N^{\rm R} = 2$ antennas. All antennas share the same multipath delays, with delay of the ith path equal to the ith chip time. The signal-to-noise ratio (SNR) per time step is defined as the ratio of the transmission power, which is assumed to be the same for WUS, pilots, and data transmission, over the noise power. We set the SNR to 10 dB.

As in [6], the encoding NPU is a fully-connected SNN featuring one hidden layer comprising 600 neurons and an output layer with 10 neurons, while the decoding NPU is designed as an SNN with a single hidden layer containing 200 neurons and an output layer consisting of 10 neurons, each representing one of the 10 classes. The hypernetwork is implemented as an ANN with two hidden layers, containing 800 and 500 neurons, respectively.

Unless stated otherwise, the maximum observation period for each data $\boldsymbol{u}$ is $L^{\max} = 60$ time steps, with the duration for the signal of interest fixed at $L^{\rm sig} = 40$. During this period, we repetitively present an input image to be classified for 40 time steps. The initial time $l^{\rm start}$ is determined by drawing from a discrete uniform distribution in the set $\{1, L^{\max} - L^{\rm sig}\}$. Subsequently, the initial $l^{\rm start}$ and the last $L^{\max} - L^{\rm sig} - l^{\rm start}$ time samples of $\boldsymbol{u}$ are generated independently using a Bernoulli distribution.

To implement the QUSUM algorithm, the irrelevant signals are modelled as Bernoulli i.i.d. samples with probability $p^{\rm noise}$, while relevant signals are also modelled as Bernoulli i.i.d. variables with a spiking probability $p^{\rm sig}$ estimated from the training data.

For IR transmission, the duration of the WUS is set to $L^{\rm w} = 2$, and the duration for the pilot is also set to $L^{\rm p} = 2$. The delay added by the transmitter is $L^{\rm d} = 3$ time steps, and the wake-up time $\delta^{\rm wake} = 2$. The power for keeping the main radio on is set to a normalized value $P^{\rm on} = 1$.

Decision are made via set prediction as in (20), and the loss function $\ell(c, \mathcal{C})$ is a 0-1 loss that indicates whether the true label c is included in the predicted set $\mathcal{C}$ or not, i.e., $\ell(c, \mathcal{C}) = \mathbb{1}(c \notin \mathcal{C})$, where $\mathbb{1}(\cdot)$ is an indicator function. Accordingly, the average loss represents the *probability of miscoverage* for the decision set $\mathcal{C}$. To evaluate the *informativeness* of the set prediction, we also compute the normalized average set size $|\mathcal{C}|/C$ of the prediction set [50].

Since our focus is on the optimization of the thresholds, rather than on training, we adopt *pre-trained* SNNs. Pre-training, testing, and calibration use the N-MNIST dataset, a neuromorphic dataset that comprises 60,000 training samples and 10,000 test samples. Each sample in the dataset represents a handwritten digit ranging from 0 to 9, and is presented as a 34×34 pixel image. We partition the training dataset by drawing $6,000$ samples for the dataset $\mathcal{D}^{\mathrm{DT}}$ and $6,000$ samples for the dataset $\mathcal{D}^{\mathrm{PT}}$, with the remaining data points used for pre-training. Pre-training is done in an end-to-end manner without considering the wake-up radio as in [6].

B. Benchmarks

For comparison, we consider the following benchmarks. For all the schemes using LTT, the grid contains all threshold tuples $(\lambda^{\mathrm{s}}, \lambda^{\mathrm{w}}, \lambda^{\mathrm{d}})$ with $\lambda^{\mathrm{s}} \in \{0, 1, \ldots, 4\}$, $\lambda^{\mathrm{w}} \in \{0.1, 0.2, \ldots, 0.6\}$, and $\lambda^{\mathrm{d}} \in \{1, 3, \ldots, 9\}$.

- *Conventional neuromorphic wireless communications:* The conventional system is designed without signal detection and wake-up radio modules, which amounts to setting the corresponding thresholds as $\lambda^{\mathrm{s}} = 0$ and $\lambda^{\mathrm{w}} = 0$. With this conventional setup, the NPUs are continuously on. Furthermore, rather than relying on the proposed adaptive set prediction strategy, in this conventional strategy, the NPU at the receiver side applies top-2 prediction to generate a prediction set, which is constructed by including the top two predicted classes with the highest spike count in the output vector (19).
- *LTT:* To evaluate the performance of a basic version of the LTT algorithm, we consider a scheme that implements LTT without the use of digital twinning. This approach follows Algorithm 1, with two caveats: (*i*) the step 1 of pre-selection via a digital twin is not carried out; and (*ii*) the number of on-air calibration transmissions, i.e., the number of iterations of the for cycle in line 4 of Algorithm 1, is limited by the average number of Pareto points in set Λ used by the proposed DT-LTT scheme. This way, the use of spectral resources for calibration is not increased as compared to DT-LTT. Note that this modification violates the assumptions in Theorem 1, and thus this scheme may not satisfy the reliability condition (38). This approach uses a fixed test sequence within the mentioned grid of hyperparameters considering first all option with the highest threshold, and then exploring other options decreasing first $\lambda^{\mathrm{s}} \in \{0, 1, \ldots, 4\}$, then $\lambda^{\mathrm{w}} \in \{0.1, 0.3\}$, and finally $\lambda^{\mathrm{d}} \in \{1, 5, 9\}$.
- *DT-LTT with an always-on main radio:* We also consider an *always-on* variant of DT-LTT, which keeps the main receiver radio on for all time instants. In this case, the hyperparameter vector $\boldsymbol{\lambda}$ to be optimized contains only the threshold λ^{s} for signal detection and the threshold λ^{d} for set prediction. As for LTT, we limit the number of on-air calibration rounds to be at most equal to the number of Pareto points in set Λ of DT-LTT. Furthermore, we set $\lambda^{\mathrm{w}} = 0$. Note that, for this strategy, the resulting calibration output does not depend on the parameter γ, since the energy consumption at the receiver is constant, irrespective of the selected hyperparameters λ^{s} and λ^{d}.

C. High-Fidelity Digital Twin

We first consider a scenario in which the digital twin implements an accurate model of the channel so that the simulated channel $\tilde{\mathbf{h}}$ follows the same distribution $p(\mathbf{h})$ as the true channel $\mathbf{h}$. For both simulated and real channels, we adopt here the standard 3GPP TR 38.901 channel model generated by Sionna, an open-source library for simulating the physical layer of wireless communication systems [51]. We use a tapped delay line channel model from the 3GPP TR38901 specification with six paths.

To illustrate the operation of DT-LTT, Fig. 5(a) presents as black and red dots the expected loss and the energy consumption plus the weighted set size estimated by the digital twin via (27), (28) and (29) for a given realization of dataset $\mathcal{D}^{\mathrm{DT}}$ and realization of the simulated channels, when the hyperparameters $\boldsymbol{\lambda}$ are chosen within the mentioned grid of values.

As seen in the figure, the expected loss and energy consumption plus weighted set size are conflicting objectives, since no hyperparameter vector $\boldsymbol{\lambda}$ exists that yields simultaneously the smallest loss and the smallest energy or the smallest set size. The Pareto optimal points, within the set of chosen options, are depicted as red points, constituting the set Λ of candidates produced by the digital twin. During on-air calibration, the candidates in set Λ are further evaluated in order of the value of the loss estimated at the digital twin.

To elaborate, in Fig. 5(b), we show weighed sum of energy consumption and set size estimated during on-air calibration using one realization of the dataset $\mathcal{D}^{\mathrm{PT}}$ and channel transmissions for hyperparameters within the set Λ. As detailed in Algorithm 1, the on-air calibration estimates the loss, energy and set size using (32), (33) and (34), starting from the candidate yielding the smallest value of the loss estimated at the digital twin, and stopping once the loss estimated on the physical system exceeds the threshold $\psi(\alpha, \delta)$. Here we set $\alpha = 0.2$ and $\delta = 0.05$. The final solution selected by the PT is represented by the star. Note that the PT does not need to evaluate hyperparameters that result in an expected loss larger than the threshold $\psi(\alpha, \delta)$.

In Fig. 6, we validate the reliability, energy consumption and informativeness of the decisions produced by the calibrated system as a function of the target miscoverage loss α with $\delta = 0.05$. The ground-truth expected loss $L(\boldsymbol{\lambda}^*)$, energy consumption $E(\boldsymbol{\lambda}^*)$ and set size $I(\boldsymbol{\lambda}^*)$ are obtained by averaging over the test set. In Fig. 6(a) and 6(b), the shaded area corresponds to average miscoverage losses that do not satisfy the average constraint (25). In a manner consistent with Theorem 1, we fix a single realization of dataset $\mathcal{D}^{\mathrm{DT}}$, simulated channels at the digital twin, and real channels, and evaluate the variability of expected loss, energy consumption, and normalized set size with respect to the realization of dataset $\mathcal{D}^{\mathrm{PT}}$. Specifically, each box spans the interquartile range of the corresponding random quantity, with a line indicating the median, while the

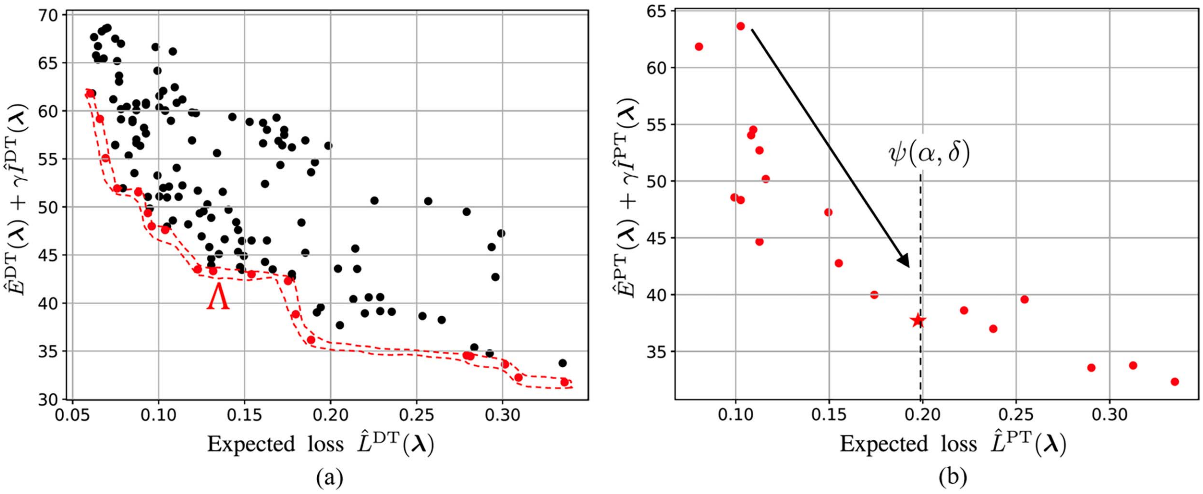


Fig. 5. Illustration of the operation of DT-LTT: (a) *Digital twin-based pre-selection*: expected loss $\hat{L}^{\mathrm{DT}}(\boldsymbol{\lambda})$ versus weighted sum $\hat{E}^{\mathrm{DT}}(\boldsymbol{\lambda}) + \gamma \hat{I}^{\mathrm{DT}}(\boldsymbol{\lambda})$ estimated at the digital twin using dataset $\mathcal{D}^{\mathrm{DT}}$ and channel simulators via (27), (28) and (29), with each point corresponding to the evaluation of a hyperparameter $\boldsymbol{\lambda}$ in a grid of options. The red points represent the selected candidates, which lie on the Pareto frontier Λ. (b) *On-air calibration*: expected loss $\hat{L}^{\mathrm{PT}}(\boldsymbol{\lambda})$ versus weighted sum $\hat{E}^{\mathrm{PT}}(\boldsymbol{\lambda}) + \gamma \hat{I}^{\mathrm{PT}}(\boldsymbol{\lambda})$ estimated using actual wireless transmissions with each point representing the evaluation for one of the hyperparameters $\boldsymbol{\lambda}$ in the set Λ. The star is the hyperparameter selected by on-air calibration with $\alpha = 0.2$, $\delta = 0.05$, $\gamma = 10$ and $L^{\max} = 60$.

whiskers extend from the box to show the overall range of the observed values.

From Fig. 6(a) and 6(b), the conventional calibration scheme fails to meet the reliability requirement, while the basic LTT scheme selects conservative hyperparameters for $\alpha = 0.1$, $\alpha = 0.15$ and $\alpha = 0.2$, by including all classes in the predicted set, leading to zero expected loss. In contrast, the proposed DT-LTT schemes are guaranteed to meet the probabilistic reliability requirement (25) as per Theorem 1. Furthermore, as the allowed miscoverage probability α increases, the expected loss obtained with DT-LTT also grows accordingly.

Looking now at the bottom part of Fig. 6, it is observed that the DT-LTT scheme with an always-on receiver is over-conservative, yielding a large energy consumption, which does not adapt to varying reliability requirements α (Fig. 6(c)). This is because this scheme is not given the freedom to keep the main radio of the receiver off in an adaptive manner. In contrast, DT-LTT is able to adjust the energy consumption to the tolerated unreliability level α, reducing the energy consumption accordingly.

The reduction in energy consumption afforded by a larger value of α depends on the design parameter γ, which dictates the relative importance of decreasing the predicted set size. In particular, increasing γ cause the DT-LTT calibration schemes to further reduce the set size as α increases, as a smaller set can support a larger miscoverage rate α. In this regard, for DT-LTT with $\gamma = 10$, the set size initially decreases and then increases with α. This is due to the importance attributed by calibration to lowering energy consumption, which calls for a larger predicted set to meet the reliability condition. Conversely, with $\gamma = 20$, the set size consistently decreases with α, as the primary objective is to minimize the set size.

We have also carried out experiments with the DVS128 Gesture dataset and the performance results are qualitatively very similar to Fig. 6, and thus we have decided not to include them due to lack of space.

D. Impact of Digital Twin Fidelity

In practice, the digital twin may employ simplified or approximated models of the physical system due to computational limitations or modeling errors. In this subsection, we evaluate the impact of a mismatch between the ground-truth physical system and the digital twin model. To this end, in this experiment, the true channel is generated by using ray tracing in a street canyon scene with cars by following Nvidia's Sionna [51]. In contrast, the digital twin model assumes the standard tapped delay line channel model from the 3GPP TR38901 specification with a variable number of paths $N_{\mathrm{DT}}^{\mathrm{P}}$ [51]. Consequently, the digital twin uses a mismatched simulator, which follows a statistical model, rather than one that is adapted to the geometry under which the real channels are generated via ray tracing. The level of real-to-simulation mismatch can be partly controlled via the choice of the number of paths $N_{\mathrm{DT}}^{\mathrm{P}}$. Furthermore, we also show the performance of DT-LTT when using a channel model matched to the real channels. We set $\alpha = 0.2$, $\delta = 0.05$, and $\gamma = 10$.

In Fig. 7, we present the expected loss, energy consumption, and the normalized set size as a function of the number of paths $N_{\mathrm{DT}}^{\mathrm{P}}$ in the simulated channel in digital twin. As shown in Fig. 7(a), DT-LTT ensures the reliability condition (25) irrespective of the fidelity of the digital twin. Furthermore, as seen in Fig. 7(b), higher energy is required for mismatched DT model in order to achieve the reliability condition. Finally, as illustrated in Fig. 7(c), a richer DT model, with a larger number of paths, supports the selection of hyperparameters that reduce the set size, improving the informativeness of the decision at the receiver.

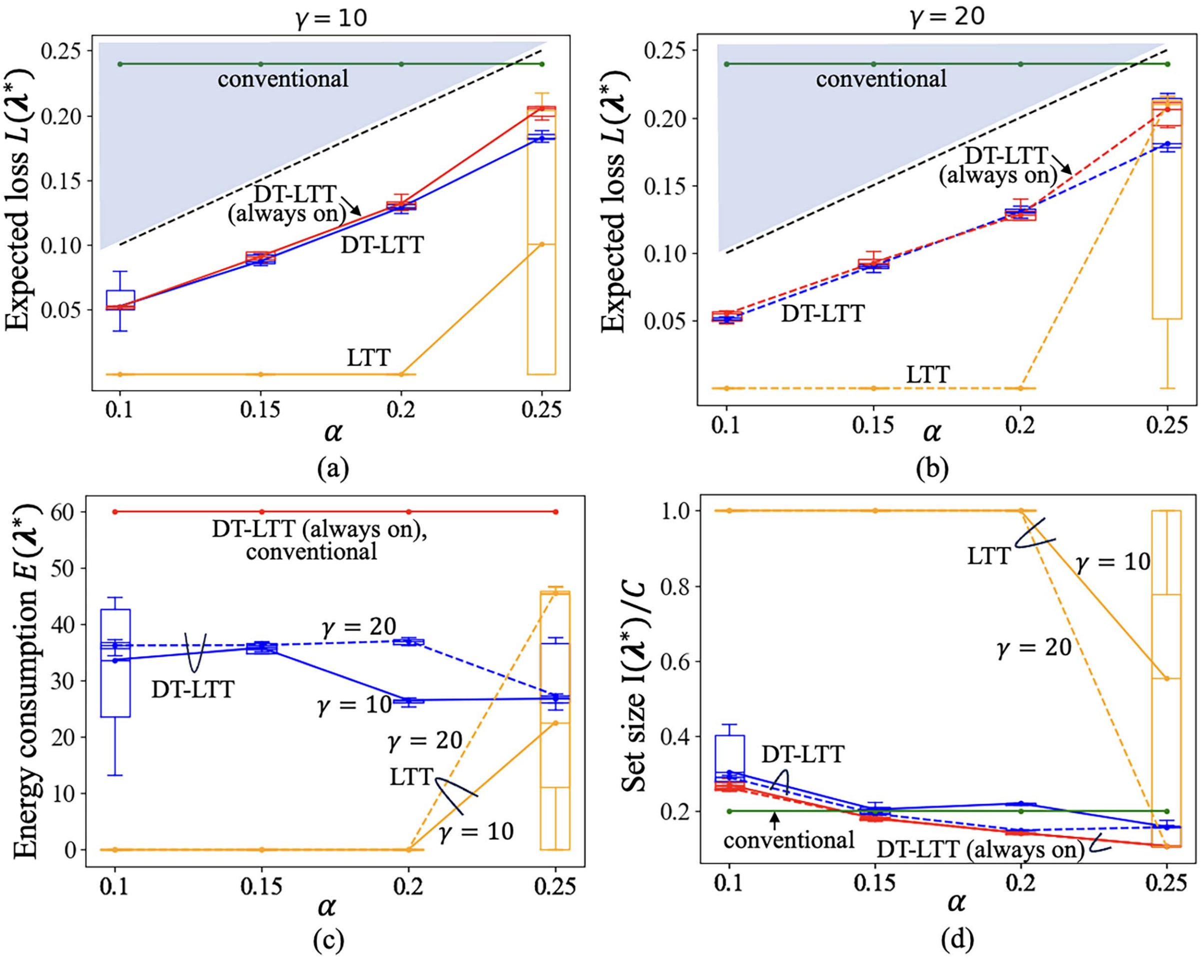


Fig. 6. (a)–(b) Expected loss $L(\boldsymbol{\lambda}^*)$ versus the reliability target α. (c) Energy consumption $E(\boldsymbol{\lambda}^*)$ versus the reliability target α. (d) Average normalized predicted set size $I(\boldsymbol{\lambda}^*)/C$ as a function of the reliability target α (with $L^{\max} = 60$ and $\delta = 0.05$).

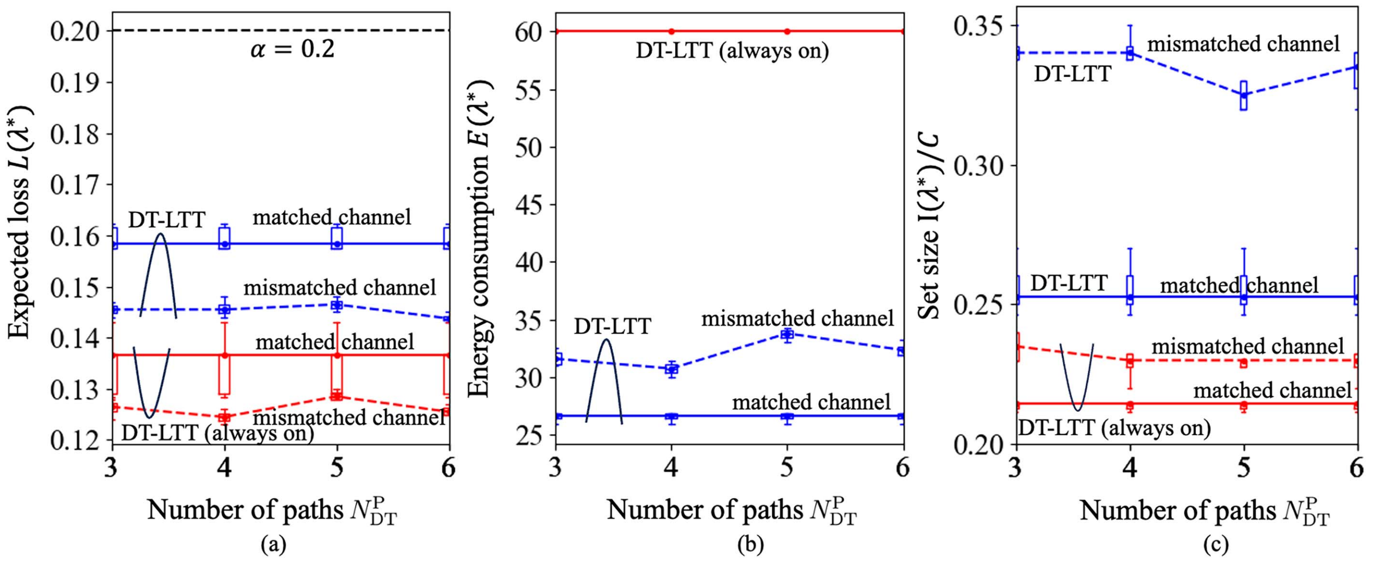


Fig. 7. (a) Expected loss $L(\boldsymbol{\lambda}^*)$ versus the number of paths $N_{\mathrm{DT}}^{\mathrm{P}}$ of the channel simulated at digital twin. (b) Energy consumption $E(\boldsymbol{\lambda}^*)$ versus the number of paths $N_{\mathrm{DT}}^{\mathrm{P}}$ of the channel simulated at digital twin. (c) Average normalized predicted set size as a function of the number of paths $N_{\mathrm{DT}}^{\mathrm{P}}$ of the channel simulated at digital twin with $\alpha = 0.2$, $\delta = 0.05$ and $L^{\max} = 60$.

VII. Conclusion

This paper has introduced a novel architecture that integrates wake-up radios into a split neuromorphic computing system. A key challenge in this integration lies in determining thresholds for sensing, WUS detection, and decision-making processes so that the system maintains an expected decision-making loss below a pre-defined target level. To tackle this problem, we have proposed a digital twin-based calibration algorithm that ensures the reliability of the receiver's decision, while also optimizing a desired trade-off between energy consumption and informativeness of the decision. By leveraging a digital twin of the system, the use of on-air resources for calibration is reduced. Experimental results demonstrated the effectiveness of the proposed algorithm, confirming the theoretical guarantees on reliability, which hold irrespective of the data distribution and of the fidelity of the digital twin.

Future research may explore a hardware-based evaluation of the proposed solution, encompassing integrated sensing, computation, and communication [45]. In terms of algorithm extensions, future work may consider incorporating delay-adaptive decision making by producing an early output once the system is confident in the inference results [48], [52].

Appendix
Proof of Theorem 1

The reliability condition (38) is a consequence of the properties of LTT [21], which is leveraged by DT-LTT via the Pareto testing method introduced in [20]. As detailed next, LTT formulates the problem of hyperparameters selection in the framework of multiple-hypothesis testing.

Consider first a single hyperparameter vector $\boldsymbol{\lambda}$, and define the null hypothesis

$$\mathcal{H}(\boldsymbol{\lambda}) : L^{\rm PT}(\boldsymbol{\lambda}) > \alpha \tag{39}$$

that the hyperparameter vector $\boldsymbol{\lambda}$ does not guarantee the desired reliability level α, where $L^{\rm PT}(\boldsymbol{\lambda}) \in [0, 1]$ is assumed to be bounded. Rejecting hypothesis $\mathcal{H}(\boldsymbol{\lambda})$ implies that the calibration algorithms deems that the hyperparameter vector $\boldsymbol{\lambda}$ ensures the reliability condition $L^{\rm PT}(\boldsymbol{\lambda}) \leq \alpha$ in (25).

To decide whether to accept or reject the null hypothesis $\mathcal{H}(\boldsymbol{\lambda})$, one can evaluate a p-value associated with hypothesis $\mathcal{H}(\boldsymbol{\lambda})$, such as

$$p(\boldsymbol{\lambda}) = e^{-2|\mathcal{D}^{\rm PT}|(\alpha - \hat{L}^{\rm PT}(\boldsymbol{\lambda}))_+^2}. \tag{40}$$

The quantity (40) is indeed a valid p-value for the null hypothesis $\mathcal{H}(\boldsymbol{\lambda})$ since the probability

$$\Pr[p(\boldsymbol{\lambda}) \leq \delta] \leq \delta \tag{41}$$

holds for $\delta \in [0, 1]$, with the probability $\Pr[\cdot]$ evaluated with respect to the distribution of dataset $\mathcal{D}^{\rm PT}$ and the true channels $\{\mathbf{h}_n \sim p(\mathbf{h})\}_{n=1}^{|\mathcal{D}^{\rm PT}|}$ under the null hypothesis $\mathcal{H}(\boldsymbol{\lambda})$. The inequality (41) is verified by Hoeffding's inequality due to the boundedness of the assumed loss [21].

Plugging (40) into (41), the inequality (41) is equivalent to the condition $\Pr[\hat{L}^{\rm PT}(\boldsymbol{\lambda}) \leq \psi(\alpha, \delta)] \leq \delta$ for any fixed hyperparameter $\boldsymbol{\lambda}$. Therefore, if the inequality $\hat{L}^{\rm PT}(\boldsymbol{\lambda}) \leq \psi(\alpha, \delta)$ is verified, so is the required reliability condition (25).

The discussion so far has focused on a single hyperparameter $\boldsymbol{\lambda}$. However, DT-LTT considers multiple hypotheses $\mathcal{H}(\boldsymbol{\lambda})$ corresponding to different candidate hyperparameter vectors $\boldsymbol{\lambda}$. To this end, DT-LTT follows fixed sequence testing via Pareto testing [20]. Accordingly, the hyperparameter vectors are tested sequentially stopping as soon as the first hyperparameter vector $\boldsymbol{\lambda}$ is found for which hypothesis $\mathcal{H}(\boldsymbol{\lambda})$ is accepted. By [21, Algorithm 1], this guarantees that all the hyperparameters associated with the rejected hypotheses ensure the reliability condition $L^{\rm PT}(\boldsymbol{\lambda}) \leq \alpha$ with probability at least $1 - \delta$. Finally, the conservative hyperparameter $\boldsymbol{\lambda} = [\lambda^{\rm s} = \infty, \lambda^{\rm w} = \infty, \lambda^{\rm d} = \infty]$ also satisfies the reliability condition (25), since the predicted set $\mathcal{C}$ always includes the true label, concluding the proof.

References

[1] M. Davies et al., "Advancing neuromorphic computing with Loihi: A survey of results and outlook," *Proc. IEEE*, vol. 109, no. 5, pp. 911–934, May 2021.

[2] H. Jang, O. Simeone, B. Gardner, and A. Gruning, "An introduction to probabilistic spiking neural networks: Probabilistic models, learning rules, and applications," *IEEE Signal Process. Mag.*, vol. 36, no. 6, pp. 64–77, Nov. 2019.

[3] Y. Matsubara and M. Levorato, "Split computing for complex object detectors: Challenges and preliminary results," in *Proc. Int. Workshop Embedded Mobile Deep Learn.*, 2020, pp. 7–12.

[4] Y. Matsubara et al., "Head network distillation: Splitting distilled deep neural networks for resource-constrained edge computing systems," *IEEE Access*, vol. 8, pp. 212177–212193, Nov. 2020.

[5] N. Skatchkovsky, H. Jang, and O. Simeone, "End-to-end learning of neuromorphic wireless systems for low-power edge artificial intelligence," in *Proc. Asilomar Conf. Signals, Syst., Comput.*, 2020, pp. 166–173.

[6] J. Chen, N. Skatchkovsky, and O. Simeone, "Neuromorphic wireless cognition: Event-driven semantic communications for remote inference," *IEEE Trans. Cogn. Commun. Netw.*, vol. 9, no. 2, pp. 252–265, Apr. 2023.

[7] J. Chen, N. Skatchkovsky, and O. Simeone, "Neuromorphic integrated sensing and communications," *IEEE Wireless Commun. Lett.*, vol. 12, no. 3, pp. 476–480, Mar. 2023.

[8] L. Zhao and A. M. Haimovich, "Performance of ultra-wideband communications in the presence of interference," *IEEE J. Sel. Areas Commun.*, vol. 20, no. 9, pp. 1684–1691, Dec. 2002.

[9] "iphone 11," Apple. Accessed: Jul. 15, 2024. [Online]. Available: https://www.apple.com/iphone-11/

[10] "IEEE standard for low-rate wireless networks–amendment 1: Enhanced ultra wideband (UWB) physical layers (PHYs) and associated ranging techniques," *IEEE Std 802.15.4z-2020 (Amendment to IEEE Std 802.15.4-2020)*, 2020, pp. 1–174.

[11] G. P. Fettweis and H. Boche, "6G: The personal tactile internet—and open questions for information theory," *IEEE BITS Inf. Theory Mag.*, vol. 1, no. 1, pp. 71–82, 2021.

[12] Y. He et al., "An implantable neuromorphic sensing system featuring near-sensor computation and send-on-delta transmission for wireless neural sensing of peripheral nerves," *IEEE J. Solid-State Circuits*, vol. 57, no. 10, pp. 3058–3070, 2022.

[13] J. Lee et al., "An asynchronous wireless network for capturing event-driven data from large populations of autonomous sensors," *Nature Electron.*, vol. 7, pp. 1–12, Mar. 2024.

[14] R. Piyare et al., "Ultra low power wake-up radios: A hardware and networking survey," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2117–2157, Jul. 2017.

[15] Z. Jouni et al., "1.2 nw neuromorphic enhanced wake-up radio," in *Proc. SBC/SBMicro/IEEE/ACM Symp. Integr. Circuits Syst. Des. (SBCCI)*, 2022, pp. 1–6.

[16] M. Z. Win and R. A. Scholtz, "Impulse radio: How it works," *IEEE Commun. Lett.*, vol. 2, no. 2, pp. 36–38, Feb. 1998.

[17] A. Hoglund et al., "3GPP release 18 wake-up receiver: Feature overview and evaluations," 2021, *arXiv:2401.03333*.

[18] H. Yomo et al., "Wake-up ID and protocol design for radio-on-demand wireless LAN," in *Proc. IEEE Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, 2012, pp. 419–424.

[19] J. Shiraishi et al., "Content-based wake-up for top-k query in wireless sensor networks," *IEEE Trans. Green Commun. Netw.*, vol. 5, no. 1, pp. 362–377, Mar. 2021.

[20] B. Laufer-Goldshtein, A. Fisch, R. Barzilay, and T. Jaakkola, "Efficiently controlling multiple risks with pareto testing," 2022, *arXiv:2210.07913*.

[21] A. N. Angelopoulos et al., "Learn then test: Calibrating predictive algorithms to achieve risk control," 2021, *arXiv:2110.01052*.

[22] F. Bozorgi, P. Sen, A. N. Barreto, and G. Fettweis, "RF front-end challenges for joint communication and radar sensing," in *Proc. IEEE Int. Online Symp. Joint Commun. Sens. (JC&S)*, 2021, pp. 1–6.

[23] K. Dakic, B. Al Homssi, S. Walia, and A. Al-Hourani, "Spiking neural networks for detecting satellite internet-of-things signals," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 60, no. 1, pp. 1224–1238, Nov. 2023.

[24] T. Borsos et al., "Resilience analysis of distributed wireless spiking neural networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, 2022, pp. 2375–2380.

[25] Y. Liu, Z. Qin, and G. Y. Li, "Energy-efficient distributed spiking neural network for wireless edge intelligence," *IEEE Trans. Wireless Commun.*, vol. 23, no. 9, pp. 10683–10697, Sep. 2024.

[26] A. Cassidy, Z. Zhang, and A. G. Andreou, "Impulse radio address event interconnects for body area networks and neural prostheses," in *Proc. Argentine School Micro-Nanoelectron., Technol. Appl.*, 2008, pp. 87–92.

[27] A. Shahshahani et al., "An all-digital spike-based ultra-low-power IR-UWB dynamic average threshold crossing scheme for muscle force wireless transmission," in *Proc. Des., Automat. Test Europe Conf. Exhib. (DATE)*, 2015, pp. 1479–1484.

[28] F. Peper, K. Leibnitz, J.-N. Teramae, T. Shimokawa, and N. Wakamiya, "Low-complexity nanosensor networking through spike-encoded signaling," *IEEE Internet Things J.*, vol. 3, no. 1, pp. 49–58, Feb. 2016.

[29] A. Pegatoquet, T. N. Le, and M. Magno, "A wake-up radio-based MAC protocol for autonomous wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 27, no. 1, pp. 56–70, Feb. 2019.

[30] S. Tang and S. Obana, "Tight integration of wake-up radio in wireless LANs and the impact of wake-up latency," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2016, pp. 1–6.

[31] L. U. Khan et al., "Digital-twin-enabled 6G: Vision, architectural trends, and future directions," *IEEE Commun. Mag.*, vol. 60, no. 1, pp. 74–80, Jan. 2022.

[32] C. Ruah, O. Simeone, and B. Al-Hashimi, "A Bayesian framework for digital twin-based control, monitoring, and data collection in wireless systems," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 10, pp. 3146–3160, Aug. 2023.

[33] S. Jiang and A. Alkhateeb, "Digital twin based beam prediction: Can we train in the digital world and deploy in reality?" in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, 2023, pp. 36–41.

[34] J. Morais and A. Alkhateeb, "Localization in digital twin MIMO networks: A case for massive fingerprinting," 2024, *arXiv:2403.09614*.

[35] G. Shafer and V. Vovk, "A tutorial on conformal prediction," *J. Mach. Learn. Res.*, vol. 9, no. 3, pp. 371–421, Aug. 2008.

[36] K. M. Cohen, S. Park, O. Simeone, and S. Shamai, "Calibrating AI models for wireless communications via conformal prediction," *IEEE Trans. Mach. Learn. Commun. Netw.*, vol. 1, pp. 296–312, Sep. 2023.

[37] A. Rácz, A. Veres, P. Hága, T. Borsos, and Z. Kenesi, "A full-stack neuromorphic prototype architecture for low-power wireless sensors," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, 2022, pp. 353–358.

[38] J. Achiam et al., "GPT-4 technical report," 2023, *arXiv: 2303.08774*.

[39] J. Huang, S. Park, and O. Simeone, "Calibrating Bayesian learning via regularization, confidence minimization, and selective inference," 2024, *arXiv:2404.11350*.

[40] M. P. Vadera, A. D. Cobb, B. Jalaian, and B. M. Marlin, "URSABench: Comprehensive benchmarking of approximate Bayesian inference methods for deep neural networks," 2020, *arXiv:2007.04466*.

[41] O. Simeone, *Machine Learning for Engineers*. Cambridge, U.K.: Cambridge Univ. Press, 2022.

[42] M. Feurer and F. Hutter, "Hyperparameter optimization," in *Automated Machine* Learning: Methods, Systems, Challenges, F. Hutter, L. Kotthoff, and J. Vanschoren, Eds., Cham, Switzerland: Springer, 2019, pp. 3–33.

[43] J. A. Rice and J. A. Rice, *Mathematical Statistics and Data Analysis*, vol. 371. Belmont, CA, USA: Thomson/Brooks/Cole, 2007.

[44] D. Wu, X. Yi, and X. Huang, "A little energy goes a long way: Build an energy-efficient, accurate spiking neural network from convolutional neural network," *Frontiers Neurosci.*, vol. 16, 2022, Art. no. 759900.

[45] Y. Ke, Z. Utkovski, M. Heshmati, O. Simeone, J. Dommel, and S. Stanczak, "Neuromorphic wireless device-edge co-inference via the directed information bottleneck," 2024, *arXiv:2404.01804*.

[46] D. Seo, Lim, and S. Hoon, "On the fundamental tradeoff of joint communication and quickest change detection," 2023, *arXiv:2401.12499*.

[47] J. Shin, A. Ramdas, and A. Rinaldo, "E-detectors: A nonparametric framework for sequential change detection," 2022, *arXiv:2203.03532*.

[48] J. Chen, S. Park, and O. Simeone, "Knowing when to stop: Delay-adaptive spiking neural network classifiers with reliability guarantees," *IEEE J. Sel. Topics Signal Process.*, early access, Jul. 22, 2024.

[49] V. Vovk, A. Gammerman, and G. Shafer, *Algorithmic Learning in a Random World*. New York, NY, USA: Springer-Verlag, 2022.

[50] M. Zecchin, S. Park, O. Simeone, and F. Hellström, "Generalization and informativeness of conformal prediction," 2024, *arXiv:2401.11810*.

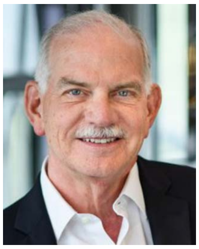
[51] J. Hoydis et al., "Sionna: An open-source library for next-generation physical layer research," 2022, *arXiv:2203.11854*.

[52] J. Chen, S. Park, and O. Simeone, "Agreeing to stop: Reliable latency-adaptive decision making via ensembles of spiking neural networks," *Entropy*, vol. 26, no. 2, Jan. 2024, Art. no. 126.

Jiechen Chen (Member, IEEE) received the Ph.D. degree from King's College London, U.K., in 2024. He is currently a Research Associate with King's Communications, Learning and Information Processing Lab, Department of Engineering, King's College London, U.K. His research interests include neuromorphic computing, signal processing, and their applications to wireless communications systems.

Sangwoo Park (Member, IEEE) received the B.Sc. degree in physics, and the M.S.E. and Ph.D. degrees in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2014, 2016, and 2020, respectively. From April to December 2019, he was a visiting Ph.D. Student with King's Communications, Learning and Information Processing Lab (KCLIP), King's College London (KCL), and United Kingdom (UK). He is currently a Research Associate with KCLIP, KCL, and UK. His research interests include practical, reliable AI, and its application to engineering systems including wireless communications and quantum information processing.

Petar Popovski (Fellow, IEEE) received the Dipl.-Ing. and M.Sc. degrees in communication engineering from the University of Sts. Cyril and Methodius in Skopje, and the Ph.D. degree from Aalborg University, in 2005. He is a Professor with Aalborg University, where he heads the section on Connectivity and a Visiting Excellence Chair with the University of Bremen. He is the author of the book titled *Wireless Connectivity: An Intuitive and Fundamental Guide*. His research interests include wireless communication and communication theory. He received an ERC Consolidator Grant (2015), the Danish Elite Researcher award (2016), the IEEE Fred W. Ellersick Prize (2016), the IEEE Stephen O. Rice Prize (2018), the Technical Achievement Award from the IEEE Technical Committee on Smart Grid Communications (2019), the Danish Telecommunication Prize (2020), and Villum Investigator Grant (2021). He was a Member at Large at the Board of Governors in IEEE Communication Society 2019–2021. He was the General Chair for IEEE SmartGridComm 2018 and IEEE Communication Theory Workshop 2019. He is currently the Editor-in-Chief of IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS and the Chair of IEEE Communication Theory Technical Committee.



H. Vincent Poor (Life Fellow, IEEE) received the Ph.D. degree in EECS from Princeton University, in 1977. From 1977 to 1990, he was on the faculty of the University of Illinois at Urbana-Champaign. Since 1990, he has been on the faculty at Princeton, where he is currently the Michael Henry Strater University Professor. From 2006 to 2016, he served as the Dean of Princeton's School of Engineering and Applied Science, and he has also held visiting appointments at several other universities, including most recently at Berkeley and Caltech. His research interests include information theory, machine learning, and network science, and their applications in wireless networks, energy systems and related fields. Among his publications in these areas is the book *Machine Learning and Wireless Communications* (Cambridge University Press, 2022). He is a Member of the National Academy of Engineering and the National Academy of Sciences and is a Foreign Member of the Royal Society and other national and international academies. He received the IEEE Alexander Graham Bell Medal in 2017.

Osvaldo Simeone (Fellow, IEEE) is a Professor of information engineering. He co-directs the Centre for Intelligent Information Processing Systems within the Department of Engineering of King's College London, where he runs the King's Communications, Learning and Information Processing Lab. He is also a Visiting Professor with the Connectivity Section within the Department of Electronic Systems, Aalborg University. From 2006 to 2017, he was a Faculty Member of the Electrical and Computer Engineering (ECE) Department, New Jersey Institute of Technology (NJIT). He is the author of the book titled *Machine Learning for Engineers* (Cambridge University Press), four monographs, two edited books, and more than 200 research journal and magazine papers. He is a co-recipient of the 2022 IEEE Communications Society Outstanding Paper Award, the 2021 IEEE Vehicular Technology Society Jack Neubauer Memorial Award, the 2019 IEEE Communication Society Best Tutorial Paper Award, the 2018 IEEE Signal Processing Best Paper Award, the 2017 JCN Best Paper Award, and the 2015 IEEE Communication Society Best Tutorial Paper Award. He is a Fellow of the IET and EPSRC.