DOI:
10.1109/TSP.2022.3220035

Document Version
Peer reviewed version

Citation for published version (APA):
Cohen, K. M., Park, S., Simeone, O., & Shamai, S. (2022). Bayesian Active Meta-Learning for Reliable and Efficient AI-Based Demodulation. *IEEE Transactions on Signal Processing*, *70*, 5366-5380. Advance online publication. https://doi.org/10.1109/TSP.2022.3220035

Bayesian Active Meta-Learning for Reliable and Efficient AI-Based Demodulation

Kfir M. Cohen, *Student Member, IEEE*, Sangwoo Park, *Member, IEEE*,
Osvaldo Simeone, *Fellow, IEEE*, Shlomo Shamai (Shitz), *Life Fellow, IEEE*

***Abstract*—Two of the main principles underlying the life cycle of an artificial intelligence (AI) module in communication networks are *adaptation* and *monitoring*. Adaptation refers to the need to adjust the operation of an AI module depending on the current conditions; while monitoring requires measures of the reliability of an AI module's decisions. Classical frequentist learning methods for the design of AI modules fall short on both counts of adaptation and monitoring, catering to one-off training and providing overconfident decisions. This paper proposes a solution to address both challenges by integrating meta-learning with Bayesian learning. As a specific use case, the problems of demodulation and equalization over a fading channel based on the availability of few pilots are studied. Meta-learning processes pilot information from multiple frames in order to extract useful shared properties of effective demodulators across frames. The resulting trained demodulators are demonstrated, via experiments, to offer better calibrated soft decisions, at the computational cost of running an ensemble of networks at run time. The capacity to quantify uncertainty in the model parameter space is further leveraged by extending Bayesian meta-learning to an active setting. In it, the designer can select in a sequential fashion channel conditions under which to generate data for meta-learning from a channel simulator. Bayesian active meta-learning is seen in experiments to significantly reduce the number of frames required to obtain efficient adaptation procedure for new frames.**

***Index Terms*—Bayesian meta-learning, uncertainty quantification, Bayesian active meta-learning, demodulation.**

I. Introduction

A. Motivation

Artificial intelligence (AI) is seen as a key enabler for next-generation wireless systems [2]. Emerging solutions, such as Open-Radio Access Network (O-RAN), incorporate AI modules as native components of a modular architecture that can be fine-tuned to meet the requirements of specific deployments [3]. Two of the main principles underlying the life cycle of an AI module in communication networks are *adaptation* and *monitoring* [4]. Adaptation refers to the need to adjust the operation of an AI module depending on the current conditions, particularly for real-time applications at the frame level. At run time, an AI model should ideally enable monitoring of the quality of its outputs by providing measures of the reliability of its decisions. The availability of such reliability measures is instrumental in supporting several important functionalities, from the combination of multiple models to decisions about retraining [5].

Classical frequentist learning methods for the design of AI modules fall short on both counts of adaptation and monitoring (see, e.g., [6], [7]). First, conventional frequentist learning is well known to provide inaccurate measures of reliability, typically producing overconfident decisions [7]. Second, the standard learning approach prescribes the one-off optimization of an AI model, hence failing to capture the need for adaptation. This paper investigates the integration of meta-learning and Bayesian learning as a means to address both challenges. As we detail in the next section, Bayesian learning can provide well-calibrated, and hence reliable, measures of uncertainty of a model's decision; while meta-learning can reduce the amount of data required for adaptation to a new task, thus improving efficiency. As a specific use case, we focus on the problems of demodulation and equalization over a fading channel based on the availability of few pilots (see Fig. 1). The goal is to develop AI solutions that are capable of adapting the demodulator/equalizer to changing conditions based on few training symbols, while also being able to quantify the uncertainty of the AI model's output.

B. Background

As illustrated in Fig. 2, frequentist learning assigns a single value to each model parameter as a result of training. This neglects (epistemic) uncertainty that exists at the level of model parameters due to the limited availability of data. In contrast, *Bayesian learning* can express uncertainty about the true value of the model parameter vector by optimizing over a distribution, rather than over a single point value [8]. By averaging predictions over the distribution of the model parameters, Bayesian learning is known to be capable of providing decisions that are well calibrated [9]–[11]. Calibration

Part of this work was presented in WSA 2021 - 25th International ITG Workshop on Smart Antennas [1].

The work of K. M. Cohen, S. Park and O. Simeone has been supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme, grant agreement No. 725731. The work of O. Simeone has also been supported by an Open Fellowship of the EPSRC.

The work of S. Shamai has been supported by the European Union's Horizon 2020 Research And Innovation Programme, grant agreement No. 694630.

The authors acknowledge use of the research computing facility at King's College London (2022): King's Computational Research, Engineering and Technology Environment (CREATE). Retrieved October 25, 2022, from https://doi.org/10.18742/rnvf-m076; and Rosalind (https://rosalind.kcl.ac.uk)

Kfir M. Cohen, Sangwoo Park, and Osvaldo Simeone are with King's Communication, Learning, & Information Processing (KCLIP) lab, Department of Engineering, King's College London, London WC2R 2LS, U.K. (e-mail: kfir.cohen@kcl.ac.uk; sangwoo.park@kcl.ac.uk; osvaldo.simeone@kcl.ac.uk).

Shlomo Shamai (Shitz) is with the Viterbi Faculty of Electrical and Computing Engineering, Technion—Israel Institute of Technology, Haifa, Israel 3200003 (e-mail: sshlomo@ee.technion.ac.il).

Code for this work can be found in `https://github.com/kclip/bayesian_active_meta_learning`.

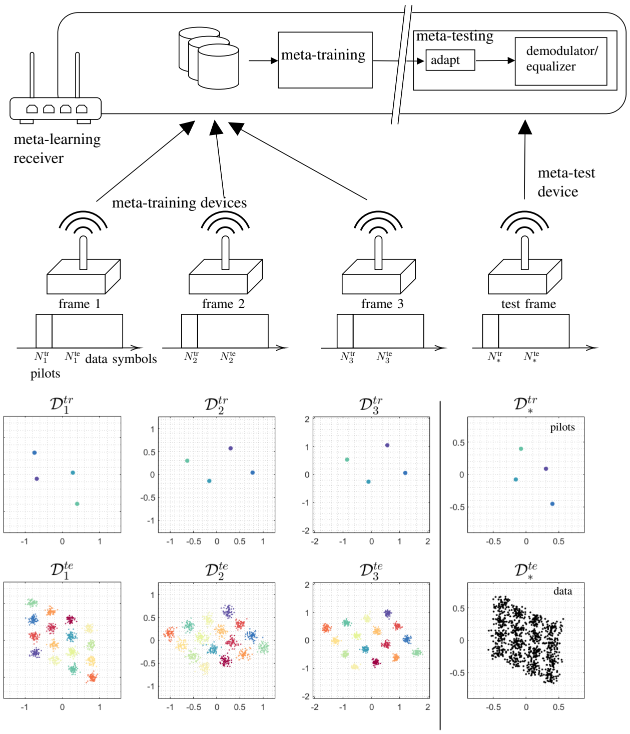


Fig. 1. Illustration of the meta-learning problem studied in this work for the example of 16-ary quadrature amplitude modulation (16-QAM). A receiver has available data corresponding to frames previously received from multiple devices, each possibly experiencing different channel conditions. Given meta-training data sets $\{\mathcal{D}_\tau\}_{\tau=1}^t$ of pilots from previous frames, partitioned into training data and test data, the demodulator optimizes a hyperparameter vector ξ. For a newly received frame, the receiver uses the few pilots therein to adapt the demodulator/equalizer parameter vector ϕ_*. In the Bayesian meta-learning framework, instead of a single parameter vector ϕ_*, the receiver optimizes over an ensemble of parameter vectors through the hyperparameter vector ξ of a posterior distribution $p(\phi_*|\mathcal{D}_*^{\rm tr}, \xi)$.

refers to the capacity of a model to produce confidence levels that reproduce well the actual accuracy of the decisions.

Meta-learning, also known as *learning to learn*, optimizes training strategies that can fine-tune a model based on few samples for a new task by transferring knowledge across different learning tasks [12]–[18]. Meta-learning is a natural tool to produce AI solutions that are optimized for adaptation. Prior work on meta-learning for communication systems, including [19]–[29], is limited to standard frequentist learning. Therefore, existing art is unable to produce models that provide well-calibrated estimations of reliability. Most related to our work is [19], which proposes to leverage pilot information from previous frames in order to optimize training procedures to be applied to the pilots of new frames (see Fig. 1).

Bayesian meta-learning aims at optimizing the procedure that produces the posterior distribution for new learning tasks. Accordingly, the goal of Bayesian meta-learning is to enhance the efficiency of Bayesian learning by reducing the number of training points needed to obtain accurate and well calibrated Bayesian models. The optimization of the Bayesian learning process is carried out by transferring knowledge from previously encountered tasks for which data are assumed to be available [30]–[32]. To the best of our knowledge, with the exception of the conference version of this paper [1], this is the first work to consider the application of Bayesian meta-learning to communication systems.

Beside meta-learning, another approach to reduce the number of required training data points is *active learning* [33]–[37]. Active learning amounts to the process of choosing which samples should be annotated next and incrementally added to the training set [38]. Through this process, active learning can select relevant samples at which the model is currently most uncertain in order to speed up the training process.

A much less studied area is *active meta-learning*, which aims at reducing the number of tasks a meta-learner must collect data from, before it can adapt efficiently to new tasks [37], [39]. Reference [37] proposes a method based on Bayesian meta-learning via empirical Bayes; while the paper [39] takes a hierarchical Bayesian approach, generalizing the Bayesian active learning by disagreements (BALD) criterion introduced in [33] to meta-learning. While [37] assumes labeled training sets, reference [39] considers unlabeled data during active meta-learning. As such, the setting in it is not applicable to the problem under study here in which data consists of supervised pairs of pilots and received signals (see Fig. 1). A summary of the relevant approaches built upon in this work is given in Table I.

C. Contributions

This paper introduces the use of Bayesian meta-learning to enable both adaptation and monitoring for the tasks of demodulation and equalization. Unlike prior works that considered either frequentist meta-learning [6], [19]–[26] or Bayesian learning [40]–[43], the proposed Bayesian meta-learning methodology enables both resource-efficient adaptation and a reliable quantification of uncertainty. To further improve the efficiency of Bayesian meta-learning we propose the use of active meta-learning, which reduces the number of required meta-training data from previously received frames. Specific contributions are as follows.

- We introduce Bayesian meta-learning for the problems of demodulation and equalization from few pilots. The proposed implementation is derived based on parametric VI.
- We introduce Bayesian active meta-learning as a solution to reduce the number of frames required by meta-learning. Active meta-learning selects in a sequential fashion channel conditions under which to generate data for meta-learning from a channel simulator.
- Extensive experimental results demonstrate that Bayesian meta-learning produces demodulators and equalizers that offer better calibrated soft decisions. Furthermore, they show that for a target meta-testing loss, active meta-learning can reduce the number of simulated meta-training frames required.

Part of this paper was presented in [1], which presented the idea of Bayesian meta-learning with some preliminary experiments. This journal version presents full technical details, new results and introduces for the first time Bayesian active meta-learning for communication systems.

TABLE I
A SUMMARY OF THE RELEVANT TECHNIQUES CONSIDERED IN THIS WORK

Approach	Goal	Methodology
frequentist learning	accurate data-driven predictions	minimize the training loss over the model parameter vector ϕ
Bayesian learning via variational inference	reliable and accurate data-driven predictions	minimize the free energy over the parameters φ of the variational distribution $q(\phi\|\varphi)$
frequentist meta-learning	sample efficiency	minimize the meta-training loss over hyperparameters ξ to be used for frequentist learning
Bayesian meta-learning	sample efficiency and reliability	minimize the meta-training loss over hyperparameters ξ to be used for Bayesian learning
Bayesian active meta-learning	task efficiency, sample efficiency, and reliability	minimize the meta-training loss over the hyperparameters ξ and over the sequential selection of meta-learning tasks

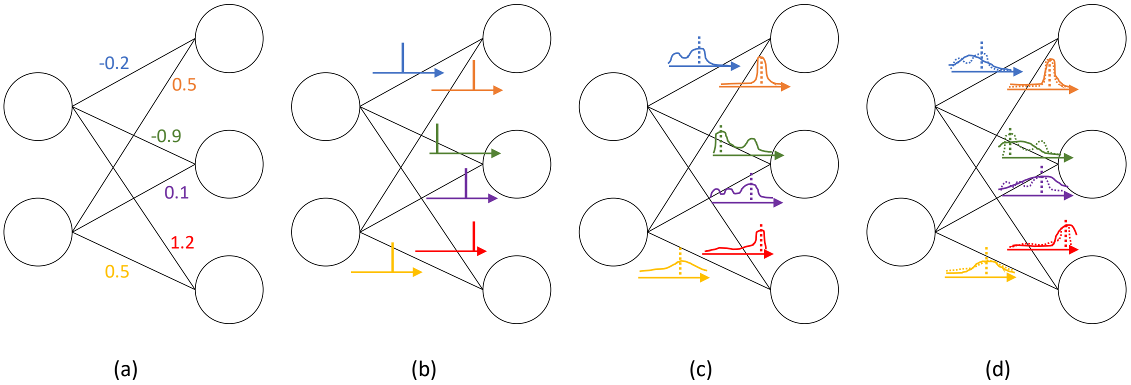


Fig. 2. Network weights in frequentist and Bayesian learning: (a) in frequentist learning, each weight is described by a scalar value; (b) the scalar value can be viewed as random variable having a degenerated probabilistic distribution concentrated at a simple prior; (c) in Bayesian learning, the weights are assigned a probability distribution, which, unlike the frequentist point estimate (dashed vertical line), provides information about the uncertainty on the weight; (d) in variational inference (VI), the posterior is approximated with a parameter distribution.

The rest of the paper is organized as follows. Section II introduces the channel model, along with background material on standard frequentist learning and frequentist meta-learning. Section III expands on Bayesian meta-learning. Then, we present Bayesian active meta-learning in Section IV. Numerical results are presented in Section V, and Section VI concludes the paper.

D. Related Work

For scalability, Bayesian learning can be implemented via approximate methods based on variational inference (VI) or Monte Carlo (MC). VI methods approximate the exact Bayesian posterior distribution with a tractable variational density [9], [44]–[47], while Monte Carlo techniques obtain approximate samples from the Bayesian posterior distribution [48]–[50]. Each class of methods comes with its own set of technical challenges and engineering choices. For instance, VI requires the selection of a variational distribution family, such as mean-field Gaussian models, and the specification of a stochastic optimization algorithm. There are also non-parametric VI methods such as Stein variational gradient descent (SVGD) [51], which optimize over deterministic and interacting particles. For MC techniques, solutions range from first-order Langevin dynamics techniques [49] to more complex methods such as Hamiltonian Monte Carlo (HMC) [48]. Implementing any of these schemes for a specific engineering application is a non-trivial task.

Bayesian learning has been applied in reference [52] to the problem of predicting the number of active users in LTE system; papers [53], [54] applied MC-based Bayesian learning for MIMO detection; the works [55]–[57] addressed channel prediction/estimation for massive MIMO systems; reference [58] studied the identification of IoT transmitters; and the authors of [59] proposed the use of robust Bayesian learning for modulation classification, localization, and channel modeling.

As for active learning, applications to communication systems include paper [60], which proposed a sample-efficient retransmission protocol; reference [61], which tackled initial beam alignment for massive MIMO system; work [62], which aimed at mitigating the problem of scarce training data in wireless cyber-security attack; and reference [63], which addressed resource allocation problems in vehicular communication systems.

Like Bayesian learning, meta-learning also provides a general design principle, which can be implemented by following different approaches. Optimization-based methods design the hyperparameters used by training algorithms; model-based techniques optimize an additional neural network model to guide adaptation of the main AI model; and metric-based schemes identify metric spaces for non-parametric inference (see, e.g., [64] and references therein).

The integration of meta-learning and Bayesian learning is highly non-trivial, and is an active topic of research in the machine learning literature. References [65]–[69] addressed Bayesian meta-learning via empirical Bayes using parametric VI [65], [66], particle-based VI [67], deep-kernels [68], and expectation-maximization [69]; while the papers [70]–[73] studied full Bayesian meta-learning that treats also the hyperparameters as random variables. Lastly, the work [74] proposed the use of quantum machine learning models as parameterized variational distributions.

II. Channel Model and Background

A. Channel Model and Soft Demodulation or Equalization

In this paper, we consider frame-based transmission over a memoryless block fading channel model with constellation $\mathcal{X}$ and channel output's alphabet $\mathcal{Y}$. The channel is characterized by a conditional distribution $p(y|x,c)$ of received symbol $y \in \mathcal{Y}$ given transmitted symbol $x \in \mathcal{X}$ and channel state c. In the case of demodulation, we treat the set $\mathcal{X}$ as discrete; while for equalization we view it as the space of vectors of a certain size. In both cases, we will refer to channel input x as symbol. The channel state c is constant within each frame, and it is independently and identically distributed (i.i.d.) across frames according to an unknown distribution $p(c)$. At frame τ, the transmitter sends a packet consisting of N_τ symbols $x_\tau = \{x_\tau[i]\}_{i=1}^{N_\tau}$. Given the channel state c_τ and the transmitted symbols, collected in a vector x_τ, the received samples $y_\tau = \{y_\tau[i]\}_{i=1}^{N_\tau}$ are conditionally independent and each received i-th sample is distributed as $y_\tau[i] \sim p(y_\tau[i]|x_\tau[i], c_\tau)$.

A *soft demodulator/equalizer* is a conditional distribution $p(x|y,\phi)$ that maps channel outputs $y \in \mathcal{Y}$ to estimated probabilities for channel input symbol $x \in \mathcal{X}$. The demodulator/equalizer depends on a vector of parameters ϕ, and is applied separately to each received sample $y[i]$ in a memoryless fashion as $p(x|y_\tau[i],\phi)$. The ideal frame-specific parameter vector ϕ_τ for the frame τ is the one that best approximates the channel conditional distribution $p(x_\tau|y_\tau,c_\tau)$, within its model class, obtained from the Bayes rule as

$$p(x_\tau|y_\tau,\phi_\tau) \approx p(x_\tau|y_\tau,c_\tau) = \frac{p(y_\tau|x_\tau,c_\tau)p(x_\tau)}{\sum_{x'_\tau \in \mathcal{X}} p(y_\tau|x'_\tau,c_\tau)p(x'_\tau)}, \tag{1}$$

where $p(x_\tau)$ is the distribution of the input symbol vector x_τ. In practice, as we detail below, the demodulator/equalizer is optimized based on pilot symbols. To simplify the terminology, we will also refer to demodulation/equalization as *prediction* henceforth.

B. Conventional Data-Driven Demodulators/Equalizers

Pilot-aided schemes utilize available pilot symbols to adapt the predictor $p(x|y,\phi)$ to the unknown channel state c in each frame τ. A typical choice for a predictor is a multi-layer neural-network [75]. With L layers, given received sample y, this class of models produces a vector

$$a(y|\phi) = W_L \cdot f_{W_{L-1},b_{L-1}} \circ \cdots \circ f_{W_1,b_1}(y) + b_L, \tag{2}$$

where $\circ$ is the composition operator; the weights $\{W_l\}_{l=1}^L$ and biases $\{b_l\}_{l=1}^L$ define the model parameter vector $\phi := \{W_l, b_l\}_{l=1}^L$ for a total of D parameters; and the function for the l-th layer f_{W_l,b_l} is a linear mapping followed by an entry-wise activation function $h(\cdot)$, i.e., $y_l = f_{W_l,b_l}(y_{l-1}) = h(W_l \cdot y_{l-1} + b_l)$ with $y_0 = y$. In the last, L-th layer, a soft *demodulator* applies the softmax function to vector $a(y|\phi)$, producing the probability distribution

$$\begin{aligned} p(x|y,\phi) &= \big[\mathrm{softmax}(a(y|\phi))\big]_x \\ &= \frac{\exp([a(y|\phi)]_x)}{\sum_{x' \in \mathcal{X}} \exp([a(y|\phi)]_{x'})}, \end{aligned} \tag{3}$$

using $[\cdot]_x$ as the x-th element of the vector. In contrast, a soft *equalizer* typically defines the conditional distribution

$$p(x|y,\phi) = \mathcal{N}(x|a(y|\phi), \beta^{-1}), \tag{4}$$

where the precision β is fixed. Throughout this paper, we use $\mathcal{N}(x|\mu,\Sigma)$ to indicate the probability density function of a Gaussian vector with mean μ and covariance matrix Σ.

In each frame τ, *conventional learning* optimizes the model parameters ϕ_τ using N_τ^{tr} i.i.d. pilots $\mathcal{D}_\tau^{\mathrm{tr}} = \{(y_\tau^{\mathrm{tr}}[i], x_\tau^{\mathrm{tr}}[i])\}_{i=1}^{N_\tau^{\mathrm{tr}}}$ as training data. Optimization of the prediction aims at minimizing the *training log-loss*

$$\mathcal{L}_{\mathcal{D}_\tau^{\mathrm{tr}}}(\phi_\tau) := -\tfrac{1}{N_\tau^{\mathrm{tr}}} \sum_{i=1}^{N_\tau^{\mathrm{tr}}} \log p(x_\tau^{\mathrm{tr}}[i]|y_\tau^{\mathrm{tr}}[i],\phi_\tau), \tag{5}$$

which amounts to the cross entropy for demodulation (3) and the quadratic prediction loss for equalization (4). Minimization of (5) can be done via *gradient descent* (GD), or stochastic GD (SGD), a variant thereof [76].

GD updates model parameter vector ϕ_τ for I iterations with learning rate $\eta > 0$ starting from an initialization vector ξ. Accordingly, the updated parameters $\phi_\tau := \phi^{\mathrm{GD}}(\mathcal{D}_\tau^{\mathrm{tr}}|\xi)$ are obtained via the iterations

$$\begin{aligned} \phi_\tau^{(0)} &= \xi, \\ \forall i = 1,\ldots,I: \quad \phi_\tau^{(i)} &\leftarrow \phi_\tau^{(i-1)} - \eta \nabla_{\phi_\tau^{(i-1)}} \mathcal{L}_{\mathcal{D}_\tau^{\mathrm{tr}}}(\phi_\tau^{(i-1)}), \\ \phi^{\mathrm{GD}}(\mathcal{D}_\tau^{\mathrm{tr}}|\xi) &= \phi_\tau^{(I)}. \end{aligned} \tag{6}$$

The resulting prediction for a test input-output pair $(y_\tau^{\mathrm{te}}[i], x_\tau^{\mathrm{te}}[i])$ is given as $p(x_\tau^{\mathrm{te}}[i]|y_\tau^{\mathrm{te}}[i], \phi^{\mathrm{GD}}(\mathcal{D}_\tau^{\mathrm{tr}}|\xi))$.

C. Frequentist Meta-Learning

The most prominent shortcoming of conventional learning is its potentially high sample complexity, which translates into the need for a large number of pilots, N_τ^{tr}, per frame. Meta-learning addresses this issue by transferring knowledge acquired over previous frames. Specifically, frequentist meta-learning, as proposed in [19], treats the initialization vector ξ in (6) as a hyperparameter vector to be optimized based on the availability of pilots from t previous transmission frames.

As a preliminary step, we decompose the available pilots from each frame τ into a disjoint training set $\mathcal{D}_\tau^{\mathrm{tr}}$ and test set $\mathcal{D}_\tau^{\mathrm{te}}$ as $\mathcal{D}_\tau = \{\mathcal{D}_\tau^{\mathrm{tr}}, \mathcal{D}_\tau^{\mathrm{te}}\}$. Furthermore, the data sets for all previous t frames are stacked as $\mathcal{D}_{1:t} = \{\mathcal{D}_\tau\}_{\tau=1}^t$, and similarly for $\mathcal{D}_{1:t}^{\mathrm{te}} = \{\mathcal{D}_\tau^{\mathrm{te}}\}_{\tau=1}^t$, having a total of $N_{1:t}^{\mathrm{te}} = \sum_{\tau=1}^t N_\tau^{\mathrm{te}}$

samples. Meta-learning has two phases: meta-training and meta-testing. These are defined next by following the frequentist meta-learning strategy of [19].

Meta-training tackles the bi-level optimization problem

$$\min_{\xi} \quad \frac{1}{N^{\text{te}}_{1:t}} \sum_{\tau=1}^{t} N^{\text{te}}_{\tau} \mathcal{L}_{\mathcal{D}^{\text{te}}_{\tau}}\big(\phi_\tau(\mathcal{D}^{\text{tr}}_{\tau}|\xi)\big) \tag{7a}$$

$$\text{s.t.} \quad \phi_\tau(\mathcal{D}^{\text{tr}}_{\tau}|\xi) = \operatorname*{argmin}_{\phi(\xi)} \mathcal{L}_{\mathcal{D}^{\text{tr}}_{\tau}}(\phi), \quad \tau = 1, \ldots, t. \tag{7b}$$

The notation $\phi(\xi)$ in (7b) indicates the dependence of the optimizer on the initialization vector ξ. By (7), the goal of frequentist meta-training is to find a hyperparameter vector ξ such that for any frame τ, the optimized model parameter vector $\phi_\tau(\mathcal{D}^{\text{tr}}_{\tau}|\xi)$ fits well the test data set $\mathcal{D}^{\text{te}}_{\tau}$.

Problem (7) is addressed via a nested loop optimization involving SGD-based inner updates and SGD-based outer updates, which are also referred as meta-iterations. The inner loop tackles the inner optimization (7b) in a per-frame manner via (6) for a randomly selected subset $\mathcal{T} \subset \{1, \ldots, t\}$ of frames, which are redrawn independently at each meta-iteration. The outer loop addresses the outer optimization (7a) via an SGD step of the meta-loss with learning-rate $\kappa > 0$, i.e.,

$$\xi \leftarrow \xi - \kappa \frac{1}{N^{\text{te}}_{\mathcal{T}}} \sum_{\tau \in \mathcal{T}} N^{\text{te}}_{\tau} \nabla_{\xi} \mathcal{L}_{\mathcal{D}^{\text{te}}_{\tau}}\big(\phi^{\text{GD}}(\mathcal{D}^{\text{tr}}_{\tau}|\xi)\big), \tag{8}$$

based on data from the batch $\mathcal{T}$ of selected frames, and using the notation $N^{\text{te}}_{\mathcal{T}} = \sum_{\tau \in \mathcal{T}} N^{\text{te}}_{\tau}$ for the total samples within the batch of selected frames. Meta-training updates the initialization vector ξ across multiple meta-iterations. When meeting some stopping criterion, here determined by a predefined number of meta-iterations I_{meta}, meta-training stops, and the hyperparameter vector ξ is stored to be used for future learning tasks.

Upon deployment, i.e., during *meta-testing*, the meta-test frames also include pilots and data as the meta-training frames. Accordingly, each meta-test device loads the hyperparameter vector ξ for initialization, and produces the adapted model parameter vector $\phi_* = \phi^{\text{GD}}(\mathcal{D}^{\text{tr}}_{*}|\xi)$ as in (6) using N^{tr}_{*} pilots symbols $\mathcal{D}^{\text{tr}}_{*} = \{(y^{\text{tr}}_{*}[i], x^{\text{tr}}_{*}[i])\}_{i=1}^{N^{\text{tr}}_{*}}$. Then, it applies the learned model to the payload data symbols $\{y^{\text{te}}_{*}[i]\}_{i=1}^{N^{\text{te}}_{*}}$ to carry out demodulation or equalization

$$p(x^{\text{te}}_{*}[i]|y^{\text{te}}_{*}[i], \phi_*). \tag{9}$$

III. The Bayesian Framework

A. Bayesian Learning

Bayesian learning treats the model parameter vector ϕ_τ for some frame τ as a random vector, rather than as a deterministic optimization variable as in frequentist learning framework. As illustrated in Fig. 2, instead of producing a single demodulator parameters $\phi_\tau = \phi^{\text{GD}}(\mathcal{D}^{\text{tr}}_{\tau}|\xi)$ as in (6), Bayesian learning produces a distribution $p(\phi_\tau|\mathcal{D}^{\text{tr}}_{\tau}, \xi)$ over the space of the demodulator parameters ϕ_τ. This distribution is computed based on training data $\mathcal{D}^{\text{tr}}_{\tau}$ and on *predetermined* prior distribution $p(\phi_\tau|\xi)$, which depends in turn on the hyperparameter vector ξ, also fixed *a priori*.

Having obtained the distribution $p(\phi_\tau|\mathcal{D}^{\text{tr}}_{\tau}, \xi)$, the *ensemble prediction* of a test point $(y^{\text{te}}_{\tau}[i], x^{\text{te}}_{\tau}[i])$ is given by the ensemble average of the predictions $p(x^{\text{te}}_{\tau}[i]|y^{\text{te}}_{\tau}[i], \phi_\tau)$ with random vector ϕ_τ having distribution $p(\phi_\tau|\mathcal{D}^{\text{tr}}_{\tau}, \xi)$, i.e.,

$$p\big(x^{\text{te}}_{\tau}[i]\big|y^{\text{te}}_{\tau}[i], \mathcal{D}^{\text{tr}}_{\tau}, \xi\big) = \mathrm{E}_{p(\phi_\tau|\mathcal{D}^{\text{tr}}_{\tau}, \xi)}\big[p\big(x^{\text{te}}_{\tau}[i]\big|y^{\text{te}}_{\tau}[i], \phi_\tau\big)\big]. \tag{10}$$

The frequentist prediction (9), reviewed in the previous section, can be viewed as a special case in which one is limited to the choice $p(\phi_\tau|\mathcal{D}^{\text{tr}}_{\tau}, \xi) = \delta(\phi_\tau - \phi^{\text{GD}}(\mathcal{D}^{\text{tr}}_{\tau}|\xi))$, with $\delta(\cdot)$ indicating the Dirac Delta. With this choice, the distribution $p(\phi_\tau|\mathcal{D}^{\text{tr}}_{\tau}, \xi)$ is concentrated at one point, namely the GD solution (6). The frequentist approach is therefore inherently limited in its capacity to express uncertainty on the model parameters due to limited data.

Ideally, the distribution $p(\phi_\tau|\mathcal{D}^{\text{tr}}_{\tau}, \xi)$ should be obtained as the posterior distribution

$$p(\phi_\tau|\mathcal{D}^{\text{tr}}_{\tau}, \xi) \propto p(\phi_\tau|\xi) p(\mathcal{D}^{\text{tr}}_{\tau}|\phi_\tau), \tag{11}$$

where $p(\mathcal{D}^{\text{tr}}_{\tau}|\phi_\tau) = \prod_{i=1}^{N^{\text{tr}}_{\tau}} p(x^{\text{tr}}_{\tau}[i]|y^{\text{tr}}_{\tau}[i], \phi_\tau)$ is the likelihood function for the training data. However, computing the posterior $p(\phi_\tau|\mathcal{D}^{\text{tr}}_{\tau}, \xi)$ in (11) is generally intractable for high dimensional vector ϕ_τ.

To address this challenge, we follow VI and introduce a *variational distribution* approximation

$$q(\phi_\tau|\varphi_\tau) \approx p(\phi_\tau|\mathcal{D}^{\text{tr}}_{\tau}, \xi), \tag{12}$$

which depends on a variational parameter vector φ_τ. A typical choice is given by the Gaussian mean-field approximation [77] which can be expressed as

$$q(\phi_\tau|\varphi_\tau) = \mathcal{N}(\phi_\tau|\nu_\tau, \mathrm{Diag}(\exp(2\varrho_\tau))), \tag{13}$$

with variational parameter vector $\varphi_\tau = [\nu_\tau^\top, \varrho_\tau^\top]^\top$, and the exponent function is applied element-wise. The variational parameter vector includes the mean vector $\nu_\tau \in \mathbb{R}^D$ and the vector of the logarithm of the standard deviations $\varrho_\tau \in \mathbb{R}^D$ for the Gaussian random vector ϕ_τ. Note that vector ϱ_τ models uncertainty in the model parameter space.

To describe VI, we will use the Kullback-Liebler (KL) divergence $\mathrm{KL}(p(z)||q(z))$ [78], which is a measure of the distance between two distributions $p(z)$ and $q(z)$. It is defined as the average of the log-likelihood ratio $\log(p(z)/q(z))$ as

$$\mathrm{KL}(p(z)||q(z)) = \mathrm{E}_{p(z)}\Big[\log\Big(\frac{p(z)}{q(z)}\Big)\Big]. \tag{14}$$

VI-based Bayesian learning prescribes that the variational parameter vectors ϕ_τ be obtained via the minimization of the KL divergence $\mathrm{KL}(q(\phi_\tau|\varphi)||p(\phi_\tau|\mathcal{D}^{\text{tr}}_{\tau}, \xi))$ between the variational distribution $q(\phi_\tau|\varphi)$ and the posterior distribution $p(\phi_\tau|\mathcal{D}^{\text{tr}}_{\tau}, \xi)$. This problem can be equivalently formulated as the minimization [76], [77]

$$\varphi_\tau = \operatorname*{argmin}_{\varphi} F_{\mathcal{D}^{\text{tr}}_{\tau}}(\varphi|\xi), \tag{15}$$

where the *variational free energy* [79] is defined as

$$\begin{aligned} F_{\mathcal{D}^{\text{tr}}_{\tau}}(\varphi_\tau|\xi) &= N^{\text{tr}}_{\tau} \mathrm{E}_{q(\phi_\tau|\varphi_\tau)}[\mathcal{L}_{\mathcal{D}^{\text{tr}}_{\tau}}(\phi_\tau)] + \mathrm{KL}(q(\phi_\tau|\varphi_\tau)||p(\phi_\tau|\xi)) \\ &= N^{\text{tr}}_{\tau} L_{\mathcal{D}^{\text{tr}}_{\tau}}(\varphi_\tau) + \mathrm{KL}(q(\phi_\tau|\varphi_\tau)||p(\phi_\tau|\xi)). \end{aligned} \tag{16}$$

In (16), we have defined as $L_{\mathcal{D}^{\mathrm{tr}}_\tau}(\varphi_\tau)$ the expectation of loss function $\mathcal{L}_{\mathcal{D}^{\mathrm{tr}}_\tau}(\phi_\tau)$ (5) over variational distribution $q(\phi_\tau|\varphi_\tau)$, i.e.,

$$L_{\mathcal{D}^{\mathrm{tr}}_\tau}(\varphi_\tau) = \mathrm{E}_{q(\phi_\tau|\varphi_\tau)}[\mathcal{L}_{\mathcal{D}^{\mathrm{tr}}_\tau}(\phi_\tau)]. \tag{17}$$

In (16), the second summand is a regularizer that restricts the variational distribution to be close to the prior distribution. Note that, if the variational distribution has ability to express the posterior distribution in (11), the minimizer of the problem (15) becomes the Bayesian posterior $p(\phi_\tau|\mathcal{D}^{\mathrm{tr}}_\tau, \xi)$, since the KL divergence $\mathrm{KL}(q(\phi_\tau|\varphi)||p(\phi_\tau|\mathcal{D}^{\mathrm{tr}}_\tau, \xi))$ is minimized (and it equals zero) when the two distributions are the same.

A typical choice for the prior distribution $p(\phi_\tau|\xi)$ is the Gaussian distribution. In this case, we have

$$p(\phi_\tau|\xi) = \mathcal{N}(\phi_\tau|\nu, \mathrm{Diag}(\exp(2\varrho))), \tag{18}$$

which is defined by the hyperparameter vector $\xi = [\nu^\top, \varrho^\top]^\top$, where $\nu \in \mathbb{R}^D$ and $\varrho \in \mathbb{R}^D$ stand for the mean and logarithm of the standard deviation vector of the Gaussian random vector ϕ_τ.

Assuming the Gaussian variational distribution in (13) and the Gaussian prior (18), the regularizer term in (16) can be computed in closed-form as

$$\mathrm{KL}(q(\phi_\tau|\varphi_\tau)||p(\phi_\tau|\xi)) =$$

$$\frac{1}{2}\sum_{d=1}^{D}\left(2(\varrho[d]-\varrho_\tau[d]) + \frac{\exp(2\varrho_\tau[d]) + (\nu_\tau[d]-\nu[d])^2}{\exp(2\varrho[d])} - 1\right),$$

which is a differentiable function for φ_τ.

With these choices of variational posterior and prior, problem (15) can be addressed via gradient-descent methods by using the reparametrization trick [80]. This is done by writing the random model parameter vector $\phi_\tau \sim q(\phi_\tau|\varphi_\tau)$ as $\phi_\tau = \nu_\tau + \exp(\varrho_\tau) \odot \mathrm{e}$, with random vector $\mathrm{e} \sim \mathcal{N}(0, I_D)$ and $\odot$ being the element-wise multiplication. An estimate of the gradient of the objective (17) using the reparametrization trick is done with the aid of R drawn independently samples of the standard normal Gaussian random vector e, and differentiating the resulting empirical estimate of (17).

Algorithm 1: `Reparametrization Trick` [80]

```
Inputs     : $\mathcal{G}(\cdot)$ = a function over vector $\phi_\tau$
             $\varphi_\tau$ = variational parameter
Parameters : $R$ = ensemble size
Output     : $\hat{G}(\varphi_\tau)$ = approximation of $\mathrm{E}_{q(\phi_\tau|\varphi_\tau)}[\mathcal{G}(\phi_\tau)]$
for $r = 1, \dots, R$ do
    Draw $\mathrm{e}_{\tau,r} \sim \mathcal{N}(0, I_D)$
    $\phi_{\tau,r}(\varphi_\tau, \mathrm{e}_{\tau,r}) \leftarrow \nu_\tau + \exp(\varrho_\tau) \odot \mathrm{e}_{\tau,r}$
return $\hat{G}(\varphi_\tau) \leftarrow \frac{1}{R}\sum_{r=1}^{R} \mathcal{G}(\phi_{\tau,r}(\varphi_\tau, \mathrm{e}_{\tau,r}))$
```

Specifically, we estimate the free energy in (16) by replacing the training loss $L_{\mathcal{D}^{\mathrm{tr}}_\tau}(\varphi_\tau)$ with the empirical estimate

$$\hat{L}_{\mathcal{D}^{\mathrm{tr}}_\tau}(\varphi_\tau) = \frac{1}{R}\sum_{r=1}^{R} \mathcal{L}_{\mathcal{D}^{\mathrm{tr}}_\tau}\big(\nu_\tau + \exp(\varrho_\tau) \odot \mathrm{e}_{\tau,r}\big), \tag{19}$$

obtained by drawing samples $\mathrm{e}_{\tau,r} \sim \mathcal{N}(0, I_D)$ for $r = 1, 2, \dots, R$. This yields the estimated free energy

$$\hat{F}_{\mathcal{D}^{\mathrm{tr}}_\tau}(\varphi_\tau|\xi) = N^{\mathrm{tr}}_\tau \hat{L}_{\mathcal{D}^{\mathrm{tr}}_\tau}(\varphi_\tau) + \mathrm{KL}(q(\phi_\tau|\varphi)||p(\phi_\tau|\xi)). \tag{20}$$

This is a special case of Algorithm 1 with input $\mathcal{G}(\phi_\tau) = \mathcal{L}_{\mathcal{D}^{\mathrm{tr}}_\tau}(\phi_\tau)$. The function (20) can be directly differentiated and used in SGD updates.

Once the variational parameter φ_τ is inferred using Bayesian training, ensemble prediction for a payload data symbol $(y^{\mathrm{te}}_\tau[i], x^{\mathrm{te}}_\tau[i])$ can be obtained via (10) by replacing $p(\phi_\tau|\mathcal{D}^{\mathrm{tr}}_\tau, \xi)$ with $q(\phi_\tau|\varphi_\tau)$ to yield the ensemble predictor

$$p(x^{\mathrm{te}}_\tau[i]|y^{\mathrm{te}}_\tau[i], \varphi_\tau) = \mathrm{E}_{q(\phi_\tau|\varphi_\tau)}\big[p(x^{\mathrm{te}}_\tau[i]|y^{\mathrm{te}}_\tau[i], \phi_\tau)\big]. \tag{21}$$

Practically, it uses Monte Carlo sampling with R model vectors, producing the approximated soft predictor $\hat{p}(x^{\mathrm{te}}_\tau[i]|y^{\mathrm{te}}_\tau[i], \varphi_\tau)$ via Algorithm 1 with $\mathcal{G}(\phi_\tau) = p(x^{\mathrm{te}}_\tau[i]|y^{\mathrm{te}}_\tau[i], \phi_\tau)$.

B. Bayesian Meta-Learning

While conventional Bayesian learning assumes that the random model parameter vector ϕ_τ has a fixed prior distribution $p(\phi_\tau|\xi)$ parametrized by a predefined hyperparameter vector ξ, Bayesian meta-learning leverages the stronger assumption that there is a shared prior distribution $p(\phi_\tau|\xi)$ across all frames that can be optimized through a hyperparameter vector ξ.

In this section, we formulate Bayesian meta-learning by following *empirical Bayes* [81], with the aim of selecting a distribution $p(\phi_\tau|\xi)$ that provides a useful prior for the design of the predictor on new frames. Mathematically, Bayesian meta-training optimizes over the hyperparameter vector ξ by addressing the bi-level problem

$$\min_{\xi} \quad \frac{1}{N^{\mathrm{te}}_{1:t}}\sum_{\tau=1}^{t} N^{\mathrm{te}}_\tau \mathrm{E}_{q(\phi_\tau|\varphi_\tau(\mathcal{D}^{\mathrm{tr}}_\tau|\xi))}\big[\mathcal{L}_{\mathcal{D}^{\mathrm{te}}_\tau}(\phi_\tau)\big] \tag{22a}$$

$$\text{s.t.} \quad \varphi_\tau(\mathcal{D}^{\mathrm{tr}}_\tau|\xi) = \operatorname*{argmin}_{\varphi} F_{\mathcal{D}^{\mathrm{tr}}_\tau}(\varphi|\xi), \tau = 1, \dots, t. \tag{22b}$$

Problem (22) chooses the hyperparameter vector ξ that minimizes the average test loss on the meta-training frames $\tau \in \{1, \dots, t\}$ that is obtained with the variational posterior via (15). The subproblems in (22b) correspond to Bayesian learning applied separately to each frame as explained in Section III-A. An illustration of all the quantities involved in problem (22) can be found in Fig. 3 by using the formalism of Bayesian networks [82].

To address problem (22) in a tractable manner, we apply the reparametrization trick for both outer (22a) and inner optimization (22b) by following the same steps described in Section III-A. Details on the optimization can be found in Algorithm 2. In short, the inner loop updates the frame-specific variational parameters φ_τ by minimizing the approximated free energy (20) separately for each frame τ within a mini-batch $\mathcal{T}$ via GD (dashed blue line in Fig. 3b). Following [30], [66], the prior's parameter vector ξ plays two roles in the inner loop, namely *(i)* as the initialization for the inner GD update in Algorithm 2 line 9; and *(ii)* as the regularizer for the same update via the prior $p(\phi_\tau|\xi)$. The outer optimization (22a) is addressed via SGD to minimize the average log-likelihood for test set using Algorithm 1 with $\mathcal{G}(\phi_\tau) = \mathcal{L}_{\mathcal{D}^{\mathrm{te}}_\tau}(\phi_\tau)$, shown as dashed green line in Fig. 3b.

After obtaining meta-trained hyperparameter ξ, meta-testing takes place, starting with the adaptation of the variational

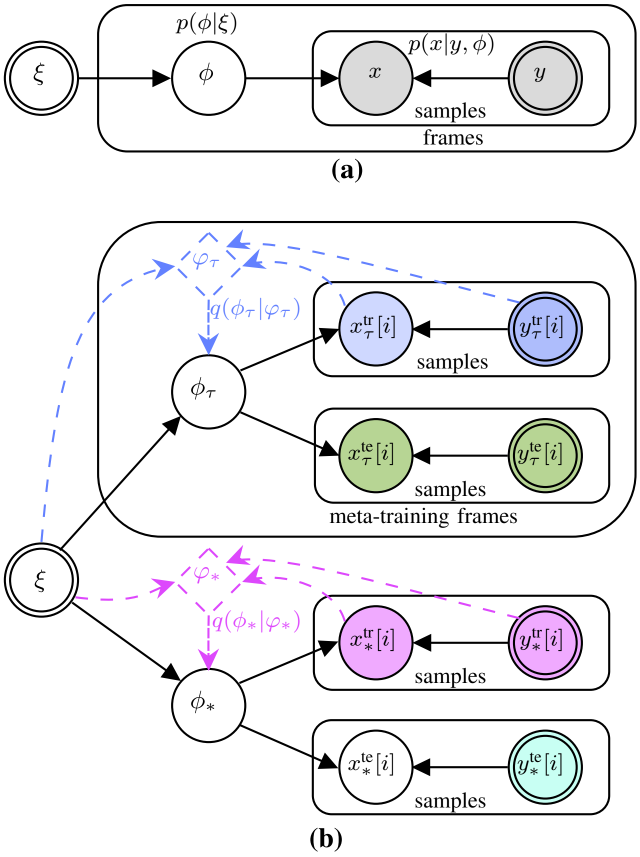


Fig. 3. Probabilistic graphical model (Bayesian network) [82] for Bayesian meta-learning. Circles represent random variables; double-lined circles represent deterministic variables or (hyper)parameters; shaded circles represent observations; dashed diamonds represent variational parameter vectors; and plaques indicate multiple instances (the outer plaques represent frames, whereas the inner represent multiple sample, e.g., symbols across time): (a) High level representation, assuming a prior $p(\phi|\xi)$ and predictor $p(x|y,\phi)$; (b) Model using the train/test splits, with variational inference $q(\phi_\tau|\varphi_\tau) \approx p(\phi_\tau|\mathcal{D}_\tau^{\mathrm{tr}},\xi)$ indicated as dashed arrows.

```
Algorithm 2: Bayesian Meta-Training
Inputs     : $\mathcal{D}_{1:t}$ = labeled data sets of $t$ meta-training frames
Parameters : $B$ = number of frames per meta-update batch
             $I$ = number of inner update steps
             $\eta, \kappa$ = inner/outer updates learning rates
Output     : $\xi$ = learned hyperparameter vector
initialize $\xi$
while meta-learning not done do
    $\mathcal{T} \leftarrow$ random batch of $B$ frames
    for $\tau \in \mathcal{T}$ do
        randomly divide $\mathcal{D}_\tau = \{\mathcal{D}_\tau^{\mathrm{tr}}, \mathcal{D}_\tau^{\mathrm{te}}\}$
        ◁ frame-specific update ▷
        $\varphi_\tau^{(0)} \leftarrow \xi$
        for $i = 1, 2, \ldots, I$ inner update steps do
            $\varphi_\tau^{(i)} \leftarrow \varphi_\tau^{(i-1)} - \frac{\eta}{N_\tau^{\mathrm{tr}}} \nabla_{\varphi_\tau} \hat{F}_{\mathcal{D}_\tau^{\mathrm{tr}}}\left(\varphi_\tau^{(i-1)} \middle| \xi\right)$ using (20)
        $\varphi^{\mathrm{GD}}(\mathcal{D}_\tau^{\mathrm{tr}}|\xi) \leftarrow \varphi_\tau^{(I)}$
    ◁ meta-update ▷
    $\xi \leftarrow \xi - \kappa \frac{1}{N_{\mathcal{T}}^{\mathrm{te}}} \sum_{t \in \mathcal{T}} N_\tau^{\mathrm{te}} \nabla_\xi \hat{L}_{\mathcal{D}_\tau^{\mathrm{te}}}\left(\varphi^{\mathrm{GD}}(\mathcal{D}_\tau^{\mathrm{tr}}|\xi)\right)$
return $\xi$
```

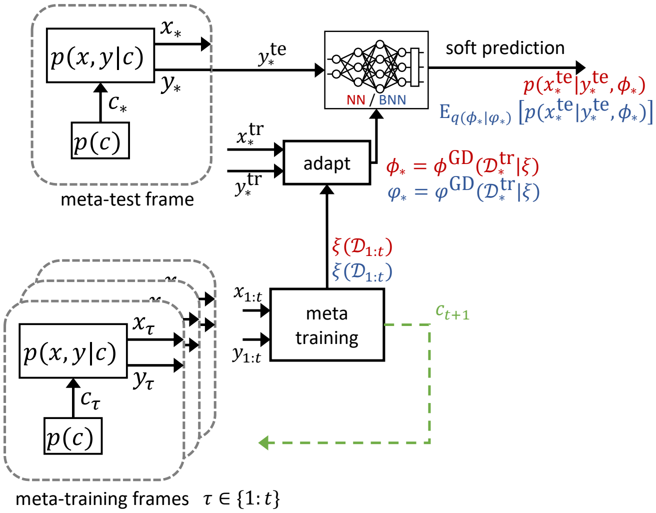

$p(x,y|c)$
x_*
y_*
c_*
$p(c)$
meta-test frame
y_*^{te}
NN / BNN
soft prediction
$p(x_*^{\mathrm{te}}|y_*^{\mathrm{te}},\phi_*)$
$\mathrm{E}_{q(\phi_*|\varphi_*)}[p(x_*^{\mathrm{te}}|y_*^{\mathrm{te}},\phi_*)]$
x_*^{tr}
y_*^{tr}
adapt
$\phi_* = \phi^{\mathrm{GD}}(\mathcal{D}_*^{\mathrm{tr}}|\xi)$
$\varphi_* = \varphi^{\mathrm{GD}}(\mathcal{D}_*^{\mathrm{tr}}|\xi)$
$\xi(\mathcal{D}_{1:t})$
$\xi(\mathcal{D}_{1:t})$
$x_{1:t}$
$y_{1:t}$
meta training
c_{t+1}
$p(x,y|c)$
x_τ
y_τ
c_τ
$p(c)$
meta-training frames $\tau \in \{1\!:t\}$

Fig. 4. Bayesian meta-learning (blue) as compared to frequentist meta-learning (red). The frequentist predictor uses a single predictor, depicted as a neural network (NN), whereas Bayesian meta-learning uses an ensemble of predictors, e.g., a Bayesian NN (BNN). The dashed line represents the operation of the active meta-learning introduced in Section IV. The data for each frame is generated by following the distribution $p(x,y|c) = p(x)p(y|x,c)$, with input distribution $p(x)$ and conditional distribution $p(y|x,c)$ for channel state c.

parameter $\varphi_*(\mathcal{D}_*^{\mathrm{tr}}|\xi)$ via (22b) using the available pilot data $\mathcal{D}_*^{\mathrm{tr}}$ at the current frame, to obtain ensemble prediction

$$p(x_*^{\mathrm{te}}[i]|y_*^{\mathrm{te}}[i],\varphi_*) = \mathrm{E}_{q(\phi_*|\varphi_*)}\left[p(x_*^{\mathrm{te}}[i]|y_*^{\mathrm{te}}[i],\phi_*)\right], \tag{23}$$

as done in (21). Bayesian meta-learning is illustrated comparatively to meta-learning in Fig. 4.

C. Computational Complexity

We now briefly elaborate on the complexity of meta-learning by analyzing the complexity of meta-training and of meta-testing. To this end, let us define as C the complexity of obtaining the probability $p(x|y,\phi)$ for a data sample (y,x). This baseline complexity depends on the model dimensionality, and it accounts for the amount of time needed to carry out the forward pass on the neural network implementing the model $p(x|y,\phi)$. Accordingly, as seen in Table II, the per-data point complexity of meta-testing equals C for frequentist learning, and CR^{te} for Bayesian learning, where R^{te} is the size of the ensemble used for inference.

The complexity of computing the first-order gradient via backpropagation per-sample is given by G_1C, with G_1 being a constant in the range between 2 and 5 [83], [84]. Furthermore, computing the Hessian-vector product (HVP) has a complexity of the order G_2G_1C, where constant G_2 is also between 2 to 5 [19, Appendix A], [85, Appendix C]. Assume that all tasks have data sets of equal size, i.e., $N_\tau^{\mathrm{tr}} = N^{\mathrm{tr}}$ and $N_\tau^{\mathrm{te}} = N^{\mathrm{te}}$ for any task τ. Therefore, for each meta-training iteration, for a batch of B tasks with I local updates, the complexity of the frequentist meta-update (8) is of the order

$$B\Big(\underbrace{IN^{\text{tr}}G_1C}_{\text{frame-specific update}} + \underbrace{IN^{\text{tr}}G_2G_1C}_{\text{HVPs in meta-update}} + \underbrace{N^{\text{te}}G_1C}_{\text{gradient in meta-update}} \Big). \tag{24}$$

For Bayesian meta-learning, the complexity increases linearly with the training ensemble size that is used for estimating the loss functions in (22a) and (22b). Note that the impact of the size R^{tr} of the training ensemble used for meta-training is different from the size R^{te} used for inference, as the first determines the variance of the stochastic loss functions, while the latter determines the quality of Bayesian prediction (see, e.g., [59] and references therein). Ignoring the constant cost of differentiating the KL term in the free energy and for sampling from the Gaussian distribution, the complexity analysis is summarized in Table II.

TABLE II
COMPUTATIONAL COMPLEXITY OF FREQUENTIST AND BAYESIAN META-LEARNING. (SEE TEXT IN SEC. III-C FOR DETAILS)

	inference [per-test sample]	meta-training [per-meta-iteration]
frequentist	C	$BG_1C\big(IN^{\text{tr}}(G_2+1)+N^{\text{te}}\big)$
Bayesian	CR^{te}	$BG_1C\big(IN^{\text{tr}}(G_2+1)R^{\text{tr}}+N^{\text{te}}R^{\text{te}}\big)$

IV. BAYESIAN ACTIVE META-LEARNING

In the previous sections, we have considered a passive meta-learning setting in which the meta-learner is given a number of meta-training data sets, each corresponding to a different channel state c. In this section, we study the situation in which the meta-learner has access to a simulator that can be used to generate random data sets for any channel state c via the channel $p(y|x,c)$. The problem of interest is to minimize the use of the simulator by actively selecting the channels $\{c_\tau\}$ for which meta-training data is generated. To this end, we devise a sequential approach, whereby the meta-learner optimizes the next channel state c_{t+1}, given all t meta-training data sets of frames $\tau = 1,\ldots,t$.

At the core of the proposed active meta-learning strategy, are mechanisms used by the meta-learner to discover model parameter vectors ϕ that have been underexplored so far, and to relate model parameter vector ϕ to a channel state.

A. Active Selection of Channel States

After having collected t meta-training data sets $\mathcal{D}_{1:t} = \{\mathcal{D}_\tau\}_{\tau=1}^t$, the proposed active meta-learning scheme selects the next channel state, c_{t+1}, to use for the generation of the $(t+1)$-th meta-training data set $\mathcal{D}_{t+1}$. We adopt the general principle of maximizing the amount of "knowledge" that can be extracted from the data set associated with selected channel c_{t+1}, when added to the t available data sets $\mathcal{D}_{1:t}$. This is done via the following three steps: *(i)* searching in the space of model parameter vectors for a vector ϕ_{t+1} that is most "surprising" given the available meta-training data $\mathcal{D}_{1:t}$; *(ii)* translating the selected model parameter vector ϕ_{t+1} into a channel c_{t+1}; and *(iii)* generating data set $\mathcal{D}_{t+1}$ by using the simulator with input c_{t+1}.

As illustrated in Fig. 5, in step *(i)*, we adopt the scoring function introduced in [37], i.e.,

$$s_t(\phi|\varphi_{1:t}) := -\log\left(\tfrac{1}{t}\sum_{\tau=1}^{t} q(\phi|\varphi_\tau)\right) \tag{25}$$

in order to select the next model parameter vector as

$$\phi_{t+1} = \underset{\phi}{\operatorname{argmax}}\, s_t(\phi|\varphi_{1:t}). \tag{26}$$

The criterion (25) measures how incompatible model parameter vector ϕ is with the available data $\mathcal{D}_{1:t}$. In fact, by the derivations in the previous section: the mixture of variational distributions $\frac{1}{t}\sum_{\tau=1}^{t} q(\phi|\varphi_\tau)$ quantifies how likely a vector ϕ is on the basis of the data $\mathcal{D}_{1:t}$ (Fig. 5b); and the negative logarithm in (25) evaluates the information-theoretic "surprise" associated with that mixture. Problem (26) can be addressed either by grid search for low-dimensional model parameter space, or by using gradient ascent due to the differentiability nature of the scoring function (25), as illustrated in Fig. 5c.

In step *(ii)*, we need to convert the selected model parameter vector ϕ_{t+1}, i.e., the outcome of (26), into channel state c_{t+1}. We choose the channel state c_{t+1} that minimizes the cross entropy loss when evaluated at ϕ_{t+1}, i.e.,

$$c_{t+1} \in \underset{c}{\operatorname{argmin}} \big\{\mathcal{L}_{\text{p}}(\phi_{t+1}|c) = \text{E}_{p(x,y|c)}[-\log p(x|y,\phi_{t+1})]\big\}, \tag{27}$$

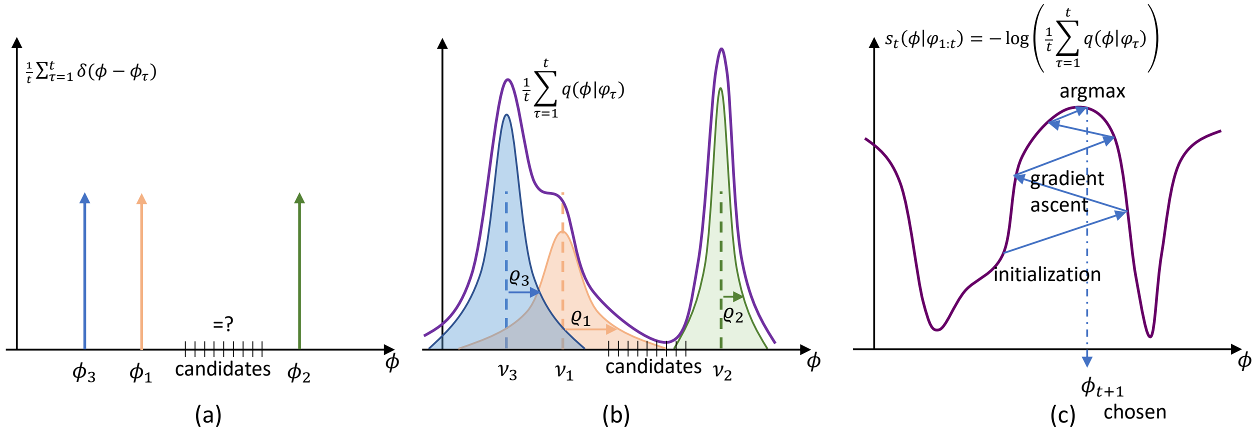
where we set $p(x,y|c) = p(x)p(y|x,c)$, with $p(x)$ being some fixed distribution and $p(y|x,c)$ being the distribution of the output of the simulator. In (27), we have emphasized that there may be more than one solution to the problem. The rational behind problem (27) is that data generated from the distribution $p(x,y|c_{t+1})$ can be interpreted as being the most compatible with the demodulator $p(x|y,\phi_{t+1})$, where compatibility is measured by the average of the cross entropy $\text{E}_{p(y|c_{t+1})}\big[\text{H}\big(p(x|y,c_{t+1}),p(x|y,\phi_{t+1})\big)\big]$.

We emphasize that the proposed approach is different from the methodology introduced by [37], which uses another variational distribution in problem (22). In our experiments, we found the method in [37] to be ineffective and complex for the problem under study here. The main issue appears to be overfitting for the additional variational distribution, which is overcome by leveraging the availability of the channel simulator implementing the model $p(y|x,c)$.

In some models, problem (27) can be solved analytically. For more complex models, SGD-based approaches can be used, either by differentiating an estimate of the loss in a manner similar to the discussion in Sec. III $\big($i.e., Algorithm 1 with $\mathcal{G}(\phi_{t+1}) = \mathcal{L}_{\text{p}}(\phi_{t+1}|c)\big)$, or by directly estimating its gradient [86].

Finally, in step *(iii)*, meta-training data set $\mathcal{D}_{t+1} = \{(y_{t+1}[i],x_{t+1}[i])\}_{i=1}^{N_{t+1}}$ is generated using the simulator in an i.i.d. fashion following the distribution

$$\prod_{i=1}^{N_{t+1}} p(x_{t+1}[i])p\big(y_{t+1}[i]\big|x_{t+1}[i],c_{t+1}\big). \tag{28}$$


$\frac{1}{t}\sum_{\tau=1}^{t}\delta(\phi-\phi_\tau)$
=?
ϕ_3
ϕ_1
candidates
ϕ_2
ϕ
$\frac{1}{t}\sum_{\tau=1}^{t} q(\phi|\varphi_\tau)$
ϱ_3
ϱ_1
ϱ_2
ν_3
ν_1
candidates
ν_2
ϕ
$s_t(\phi|\varphi_{1:t}) = -\log\left(\frac{1}{t}\sum_{\tau=1}^{t} q(\phi|\varphi_\tau)\right)$
argmax
gradient ascent
initialization
ϕ_{t+1}
chosen
ϕ

(a) (b) (c)

Fig. 5. Illustration of how model parameter vectors are scored to enable active meta-learning provided $t = 3$ meta-training sets. (a) Frequentist meta-learning relies on point estimates, and is hence unable to score as-of-yet unexplored model parameters; (b) Bayesian meta-learning can associate a score to each model parameter vector ϕ based on the variational distributions $\{q(\phi|\varphi_\tau)\}$ evaluated in the previously observed frames $\tau = 1, \dots, t$; (c) The scoring function can be maximized to obtain the next model parameter vector ϕ_{t+1} as the most "surprising" one.

As a final note, we adopt the proposal in [37] of implementing active selection only after $t_{\text{init}} > 1$ channel states that are generated at random, as a means to avoid being overconfident at early stages. The overall proposed Bayesian active meta-learning scheme is summarized in Algorithm 3.

Algorithm 3:
`Bayesian Active Meta-Training`

```
Inputs     : $p(y|x,c)$ = channel model
             $p(x)$ = generative symbols distribution
Parameters : $t_{\text{init}}$ = number of prior-based first frames
Output     : $\xi$ = shared hyperparameter vector

◁ Generate initial experience ▷
for $t = 1, 2, \dots, t_{init}$ do
    Draw using the prior $c_t \sim p(c_t)$
    Acquire data $\mathcal{D}_t \sim \prod_{i=1}^{N_t} p(x_t[i])p(y_t[i]|x_t[i], c_t)$
while data acquisition not done do
    $\xi \leftarrow$ BayesianMetaTraining($\mathcal{D}_{1:t}$) using Algorithm 2
    ◁ frame-specific update with updated $\xi$ ▷
    for $\tau = 1, 2, \dots, t$ do
        $\varphi_\tau \leftarrow \varphi^{\text{GD}}(\mathcal{D}_\tau^{\text{tr}}|\xi)$ using (20)
    ◁ step (i), choose surprising model parameter ▷
    $\phi_{t+1} = \text{argmax}_\phi\, s_t(\phi|\varphi_{1:t})$ using (25)
    ◁ step (ii), choose next channel ▷
    $c_{t+1} \in \text{argmin}_c\, \mathcal{L}_{\text{p}}(\phi_{t+1}|c)$ as in (27)
    ◁ step (iii), generate data set ▷
    $\mathcal{D}_{t+1} \sim \prod_{i=1}^{N_{t+1}} p(x_{t+1}[i])p(y_{t+1}[i]|x_{t+1}[i], c_{t+1})$
    $t \leftarrow t + 1$
return $\xi$
```

V. Experiments

In this section, we present experimental results to evaluate the performance of Bayesian meta-learning for demodulation/equalization.

A. Performance Metrics

Apart from the standard measures of symbol error rate (SER) and mean squared error (MSE), we will also evaluate metrics quantifying the performance in terms of the reliability of the confidence measures provided by the predictor. While such measures can be defined for both classification and regression problems, we will focus here on uncertainty quantification for demodulation via calibration metrics (see [87] for discussion on regression).

As discussed in the previous sections, for a new frame, we need to make a prediction for the payload symbols $\{y_*^{\text{te}}[i]\}_{i=1}^{N_*^{\text{te}}}$ via the demodulator $p(x_*^{\text{te}}[i]|y_*^{\text{te}}[i], \phi_*)$ for frequentist meta-learning (9), or $p(x_*^{\text{te}}[i]|y_*^{\text{te}}[i], \varphi_*)$ for Bayesian meta-learning (21). The *confidence level* assigned by the model to the hard predicted symbol

$$\hat{x}_*^{\text{te}}[i] = \underset{x\in\mathcal{X}}{\text{argmax}}\, p(x|y_*^{\text{te}}[i], \theta) \tag{29}$$

given the received symbol $y_*^{\text{te}}[i]$, can be defined as the corresponding probability [7]

$$\hat{p}[i] = \max_{x\in\mathcal{X}} p(x|y_*^{\text{te}}[i], \theta) = p(\hat{x}_*^{\text{te}}[i]|y_*^{\text{te}}[i], \theta), \tag{30}$$

where we have $\theta = \phi_*$ for frequentist meta-learning and $\theta = \varphi_*$ for Bayesian meta-learning. *Perfect calibration* [7] can be defined as the condition where symbols that are assigned a confidence level $\hat{p}[i]$ are also characterized by a probability of correct detection equal to p.

Two standard means of quantifying the extent to which the perfect calibration is satisfied are reliability diagrams [88] and expected calibration error (ECE) [7]. To introduce them, the probability interval $[0, 1]$ is first divided into M equal length intervals, with the m-th interval $(\frac{m-1}{M}, \frac{m}{M}]$ referred to as the m-th bin henceforth. Let us denote as $\mathcal{B}_m$ the subset of the payload data symbol indices whose associated confidence level $\hat{p}[i]$ lie within the m-th bin, i.e.,

$$\mathcal{B}_m = \left\{i \middle| \hat{p}[i] \in \left(\tfrac{m-1}{M}, \tfrac{m}{M}\right],\ \text{with } i = 1, 2, \dots, N_*^{\text{te}}\right\}. \tag{31}$$

Note this is a partition of the data set $\mathcal{D}_*^{\text{te}}$ since we have $\bigcup_{m=1}^{M}\mathcal{B}_m = \{i = 1, 2, \dots, N_*^{\text{te}}\}$ and $\mathcal{B}_m \cap \mathcal{B}_{m'} = \emptyset$ for any $m' \neq m$.

The within-bin empirical average accuracy of the predictor for the m-th bin is defined as

$$\mathrm{acc}(\mathcal{B}_m) = \frac{1}{|\mathcal{B}_m|} \sum_{i \in \mathcal{B}_m} \mathbf{1}(\hat{x}_*^{\mathrm{te}}[i] = x_*^{\mathrm{te}}[i]), \tag{32}$$

with $\mathbf{1}(\cdot)$ being indicator function and $|\mathcal{B}_m|$ denoting the number of total samples in $\mathcal{B}_m$. The within-bin empirical average confidence of the predictor for the m-th bin is

$$\mathrm{conf}(\mathcal{B}_m) = \frac{1}{|\mathcal{B}_m|} \sum_{i \in \mathcal{B}_m} \hat{p}[i]. \tag{33}$$

A perfectly calibrated demodulator $p(x|y,\theta)$ would have $\mathrm{acc}(\mathcal{B}_m) = \mathrm{conf}(\mathcal{B}_m)$ for all $m \in \{1,\dots,M\}$ in the limit of a sufficiently large payload data set, i.e., $N_*^{\mathrm{te}} \to \infty$.

Reliability diagrams plot the accuracy $\mathrm{acc}(\mathcal{B}_m)$ and the confidence $\mathrm{conf}(\mathcal{B}_m)$ over the binned probability interval $[0,1]$. Ideal calibration would yield $\mathrm{acc}(\mathcal{B}_m) = \mathrm{conf}(\mathcal{B}_m)$ in a reliability plot. If in the m-th bin, the empirical accuracy and empirical confidence are different, the predictor is considered to be over-confident when $\mathrm{conf}(\mathcal{B}_m) > \mathrm{acc}(\mathcal{B}_m)$, and under-confident when $\mathrm{conf}(\mathcal{B}_m) < \mathrm{acc}(\mathcal{B}_m)$.

The ECE quantifies the overall amount of miscalibration by computing the weighted average of the differences between within-bin accuracy and within-bin confidence levels across all M bins, i.e.,

$$\mathrm{ECE} = \frac{1}{N_*^{\mathrm{te}}} \sum_{m=1}^{M} |\mathcal{B}_m| \Big|\mathrm{acc}(\mathcal{B}_m) - \mathrm{conf}(\mathcal{B}_m)\Big|. \tag{34}$$

B. Frequentist and Bayesian Meta-Learning for Demodulation

For the first set of experiments, we focus on a demodulation problem at the symbol level in the presence of transmitter I/Q imbalance [89], [90], as considered also in [19]. The main reason for this choice is that channel decoding typically requires a hard decision on the transmitted codeword, whose accuracy can be validated via a cyclic redundancy check. In contrast, demodulation is usually a preliminary step at the receiver side, and downstream blocks, such as channel decoding, expect soft inputs that are well calibrated. For each frame τ, the transmitted symbols $x_\tau[i]$ are drawn uniformly at random from the 16-QAM constellation $\mathcal{X} = 1/\sqrt{10}(\{\pm 1, \pm 3\} + \jmath\{\pm 1, \pm 3\})$. The received symbol $y_\tau[i] \in \mathcal{Y} = \mathbb{C}$ is given as

$$y_\tau[i] = h_\tau f_{\mathrm{IQ},\tau}(x_\tau[i]) + z_\tau[i], \tag{35}$$

for a unit energy fading channel coefficient h_τ, where the additive noise is $z_\tau[i] \sim \mathcal{CN}(0, \mathrm{SNR}^{-1})$ for some signal-to-noise ratio (SNR) level SNR, and the I/Q imbalance function [91] $f_{\mathrm{IQ},\tau} : \mathcal{X} \to \bar{\mathcal{X}}_\tau$ is

$$f_{\mathrm{IQ},\tau}(x_\tau[i]) = \bar{x}_{\mathrm{I},\tau}[i] + \jmath \bar{x}_{\mathrm{Q},\tau}[i] \tag{36}$$

$$\begin{bmatrix} \bar{x}_{\mathrm{I},\tau}[i] \\ \bar{x}_{\mathrm{Q},\tau}[i] \end{bmatrix} = \begin{bmatrix} 1+\epsilon_\tau & 0 \\ 0 & 1-\epsilon_\tau \end{bmatrix} \begin{bmatrix} \cos\delta_\tau & -\sin\delta_\tau \\ -\sin\delta_\tau & \cos\delta_\tau \end{bmatrix} \begin{bmatrix} x_{\mathrm{I},\tau}[i] \\ x_{\mathrm{Q},\tau}[i] \end{bmatrix},$$

which depends on the imbalance parameters ϵ_τ and δ_τ. In (35), $x_{\mathrm{I},\tau}[i]$ and $x_{\mathrm{Q},\tau}[i]$ refer to the real and imaginary parts of the modulated symbol $x_\tau[i]$; and $\bar{x}_{\mathrm{I},\tau}[i]$ and $\bar{x}_{\mathrm{Q},\tau}[i]$ stand for the real and imaginary parts of the transmitted symbol $f_{\mathrm{IQ},\tau}(x_\tau[i])$. Note that the constellation $\bar{\mathcal{X}}_\tau$ of the transmitted symbols $\bar{x}_\tau[i]$ is also composed of 16 points via (36).

By (35) and (36), the channel state c_τ consists of the tuple: (a) amplitude imbalance factor $\epsilon_\tau \in [0, 0.15]$; (b) phase imbalance factor $\delta_\tau \in [0, 15^\circ]$; and (c) channel realization $h_\tau \in \mathbb{C}$. All of the variables are drawn i.i.d. across different frames and are fixed during each frame. We consider the channel state distribution for frame τ as

$$p(c_\tau) = \mathrm{Beta}\Big(\tfrac{\epsilon_\tau}{0.15}\Big|5,2\Big)\mathrm{Beta}\Big(\tfrac{\delta_\tau}{0.15^\circ}\Big|5,2\Big)\mathcal{CN}(h_\tau|0,1). \tag{37}$$

We set our base learner to be a multi-layer fully-connected neural network (3) with $L=5$ layers. The real and imaginary parts of the input $y[i] \in \mathbb{C}$ are treated as a vector in $\mathbb{R}^2$, which is fed to layers with 10, 30, and 30 neurons, all with ReLU activations, while the last linear layer implements a softmax function that produces probabilities for the 16QAM constellation points.

To address the ability of meta-learning to adapt the demodulator using only few pilots, we set the number of pilots as $N_\tau^{\mathrm{tr}} = 4$ during meta-training and $N_*^{\mathrm{tr}} = 8$ for meta-testing [19]. Fig. 6 shows the SER as a function of the number of total meta-training frames t. Since only half of the constellation points are available as pilots during meta-test ($N_*^{\mathrm{tr}} = 8$ different symbols out of 16), conventional learning cannot obtain a SER lower than of 0.5. In fact, conventional learning performs worse than a standard model-based receiver applying linear minimal mean square error (LMMSE), followed by maximum likelihood (ML) demodulation, while disregarding the presence of I/Q imbalance function f_{IQ}. Both meta-learning schemes are clearly superior to conventional learning and to the mentioned model-based solution, showing that useful knowledge has been transferred from previous frames to a new frame. Furthermore, Bayesian meta-learning obtains a slightly lower SER as compared to frequentist meta-learning. This advantage stems from the capacity of ensemble predictors to implement more complex decision boundaries [59].

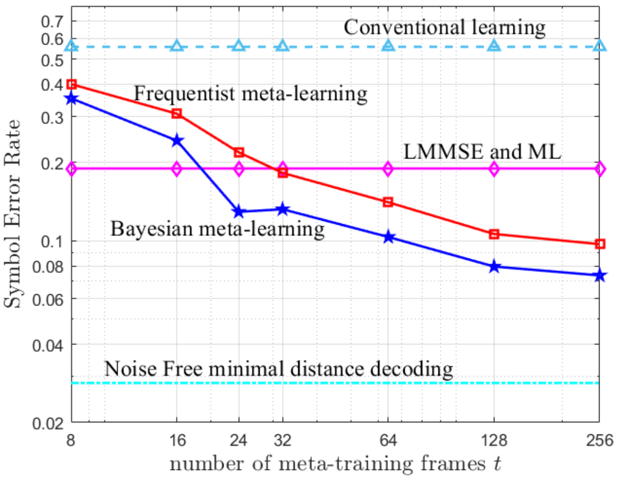


Fig. 6. Symbol error rate (SER) as a function of the number t of meta-training frames with 16-QAM, Rayleigh fading, and I/Q imbalance for $N_\tau^{\mathrm{tr}} = 4$, $N_*^{\mathrm{tr}} = 8$. The symbol error rate is averaged over by $N_*^{\mathrm{te}} = 4000$ data symbols and 50 meta-test frames with ensemble of size 100.

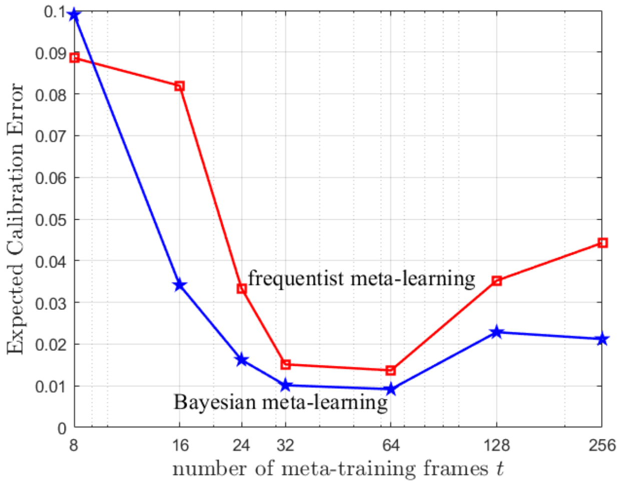


Fig. 7. Expected calibration error (ECE) over meta-test data $\mathcal{D}_*^{\text{te}}$ as a function of the number t of meta-training frames, for the same setting as in Fig. 6.

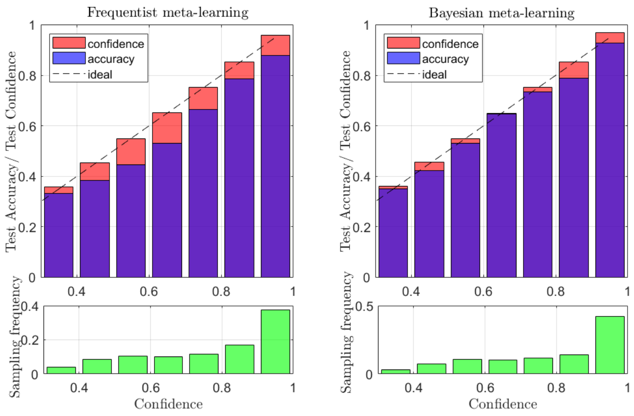


Fig. 8. Reliability diagrams (top) for frequentist meta-learning (left) and Bayesian meta-learning (right) with SNR $=18$ dB, using $t=16$ meta-training frames and predictions averaged over 50 meta-test frames. Frequentist meta-learning tends to be over-confident, whereas the Bayesian soft predictions are better matched to the true accuracy. The bottom figure shows the histogram of $|\mathcal{B}_m|/N$ of prediction over $M=10$ bins. Full details in Appendix A

To gain insights into the reliability of the uncertainty quantification provided by the demodulator, we use the metrics defined in Sec. V-A by setting the total number of bins to $M=10$. We plot the ECE as a function of the number of total meta-training frames t in Fig. 7. Bayesian meta-learning is seen to achieve a lower ECE than frequentist meta-learning, indicating that Bayesian meta-learning provides more reliable estimates of uncertainty. Furthermore, the increase in ECE as the number t of available meta-training frames increases may be interpreted as a consequence of meta-overfitting [92]. This suggests that meta-learning may be considered as complete after a number of frames that depends on the complexity of the propagation environment. In practice, this can be assessed by evaluating the performance of the demodulator on pilots (see the online strategy in [19] for further discussion on this point).

To further elaborate on the quality of uncertainty quantification, Fig. 8 depicts reliability diagrams for frequentist and Bayesian meta-learning. The within-bin accuracy levels $\text{acc}(\mathcal{B}_m)$ in (32) and the within-bin empirical confidence $\text{conf}(\mathcal{B}_m)$ in (33) are depicted as dark (blue) and light (red) bars, respectively. Frequentist meta-learning is observed to produce generally over-confident predictions, while Bayesian meta-learning provides better calibrated predictions with well-matching confidence and accuracy levels.

C. Bayesian Active Meta-Learning for Equalization

In this sub section, we illustrate the operation of active meta-learning by investigating a single-input multiple-output (SIMO) Rayleigh block fading real channel model. At frame τ, the modulator uses a 4-PAM to produce symbols $x_\tau[i]$, $i=1,2,\ldots,N_\tau$, taken uniformly from the set $\mathcal{X}\in 1/\sqrt{5}\{-3,-1,+1,+3\}$. Given channel state $c_\tau=[c_\tau^0,c_\tau^1]^\top\in\mathbb{R}^2$, the i-th channel output symbol $y_\tau[i]\in\mathbb{R}^2$ for $i=1,2,\ldots,N_\tau$ is defined as the two-dimensional real vector

$$y_\tau[i]=c_\tau x_\tau[i]+z_\tau[i], \tag{38}$$

where both the additive noise $z_\tau[i]\sim\mathcal{N}\big(0,\frac{1}{2\text{SNR}}I_2\big)$ and the normalized real block fading coefficients $c_\tau\sim p(c)=\mathcal{N}(c|0,I_2)$ are i.i.d. We adopt the linear equalizer

$$\hat{x}_\tau[i]=\phi_\tau^\top\cdot y_\tau[i] \tag{39}$$

with linear equalizer weight vector $\phi_\tau=[\phi_\tau^0,\phi_\tau^1]^\top\in\mathbb{R}^2$. To obtain a soft equalization, we account for a precision level β via the conditional distribution

$$p(x_\tau[i]|y_\tau[i],\phi_\tau)=\mathcal{N}(\phi_\tau^\top\cdot y_\tau[i],\beta^{-1}). \tag{40}$$

The next model parameter ϕ_{t+1} is chosen to maximize the scoring function as in (26) by restricting the optimization to the domain $||\phi||\leq 1$. This restricted optimization domain is selected in order to match the circular symmetry of the problem. Furthermore, the corresponding next channel state c_{t+1} is selected by tackling problem (27), which amounts to the minimization

$$\begin{aligned} c_{t+1}(\phi) &\in \operatorname*{argmin}_c \mathrm{E}_{p(x)p(y|x,c)}[-\log p(x|y,\phi)] && \text{(41a)}\\ &= \operatorname*{argmin}_c \mathrm{E}_{p(x)p(z)}[\tfrac{\beta}{2}(x-\phi^\top\cdot(cx+z))^2]\\ &= \operatorname*{argmin}_c \mathrm{E}_{p(x)p(z)}[\tfrac{\beta}{2}\big((1-\phi^\top\cdot c)x-\phi^\top z\big)^2]\\ &= \big\{c\big|\phi^\top\cdot c=1\big\}. && \text{(41b)}\end{aligned}$$

In the set of solutions of problem (41b), we select the minimum-norm solution $c_{t+1}=\phi_{t+1}/\|\phi_{t+1}\|^2$. This way, the selected channel focuses on the more challenging low-SNR regime. Details of this experiment are provided in Appendix A.

Fig. 9 illustrates the scoring function (25) used to select the next model parameter ϕ_{t+1} as a heat map in the space of model parameter ϕ. Specifically, the figure shows the scoring functions after observing $t=4$ and $t=5$ meta-training frames. The optimized next model parameter vector ϕ_{t+1} (26) is shown as a star, while the previously selected model parameter vectors $\phi_{1:t}$ are shown as squares. Fig. 9 illustrates how active meta-learning efficiently explores the model parameter space. It does so by avoiding the inclusion of channel states that are similar

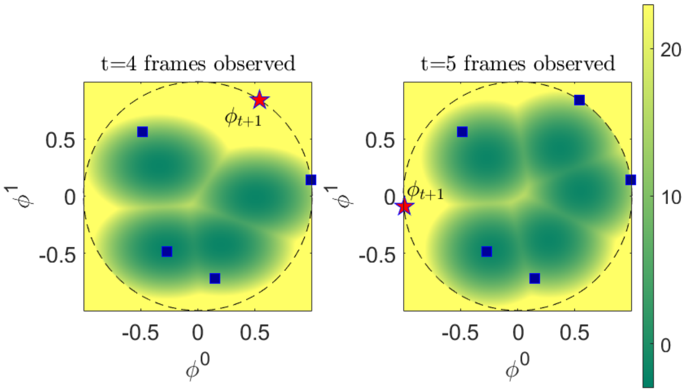


Fig. 9. Scoring function (25) used by Bayesian active meta-learning to select the next model parameter vector ϕ_{t+1} at the fourth and fifth iterations. The scoring function is shown as a heat map over the two dimensional space of the model parameter vector ϕ for the example detailed in Sec. V-C.

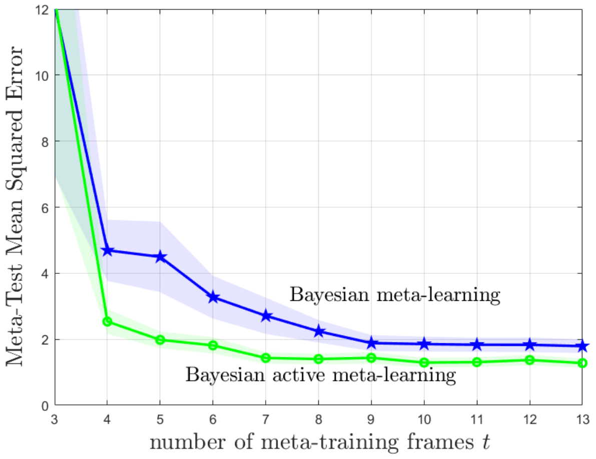


Fig. 10. Meta-test mean squared error (MSE) loss as function of the number of frames t. Bayesian active meta-training is able to achieve lower meta-test loss levels by using fewer meta-training tasks t. Solid lines are the mean test loss over 100 channel states. The confidence levels account for one standard deviation.

to those already considered (i.e., the squares in the figure). This way, the model parameter space can be covered with fewer meta-training frames t, leading to a larger frame efficiency of active meta-learning.

Finally, to numerically validate the advantage of active meta-learning, we plot the meta-test MSE loss in Fig. 10 for both passive and active Bayesian meta-learning versus the number of frames t. For passive meta-learning, we have generated random channel realizations by drawing from the distribution $p(c) = \mathcal{N}(c|0, I_2)$. We have repeated the experiment 100 times, and show the confidence interval of one standard deviation for the meta-test loss. The results in the figure confirm that active meta-learning requires far fewer meta-training frames. Furthermore, the increased randomness of passive meta-learning is due to the random selection of channel states at each iteration.

VI. Conclusions

In this paper, we have introduced tools for reliable and efficient AI in communication systems via Bayesian meta-learning. Bayesian learning has the advantage of producing well-calibrated decisions whose confidence levels are a close match for the corresponding test accuracy. This property facilitates monitoring of the quality of the outputs of an AI module. Meta-learning optimizes models that can quickly adapt based on few pilots, producing sample-efficient AI solutions. This paper has focused on the application of Bayesian meta-learning to the basic problems of demodulation/equalization from few pilots. We have demonstrated via experiments that the demodulator/equalizer obtained via Bayesian meta-learning not only achieves a higher accuracy, but it also enjoys better calibration performance than its standard frequentist counterpart. Furthermore, thanks to meta-learning, such performance levels can be obtained based on a limited number of pilots per frame.

To reduce the number of past frames required by meta-learning, we have also introduced Bayesian active meta-learning, which leverages the uncertainty estimates produced by Bayesian learning to actively explore the space of channel conditions. We have shown via numerical results that active meta-learning can indeed significantly speed up meta-training in terms of number of frames.

Future work may consider a fully Bayesian meta-learning implementation that also accounts for uncertainty at the level of hyperparameters (see, e.g., [73] and references therein). This may be particularly useful in the regime of low number of frames. Another direction for research would be to investigate different scoring functions for active meta-learning (see, e.g., [39]). A study on the impact of well-calibrated decisions obtained via Bayesian learning on downstream blocks at the receiver, such as channel decoding, is also of interest. Finally, the proposed tools may find applications to other problems in communications, such as power control [25] and channel coding [28], [93].

Appendix A
Experiments Details

Table III summarizes the parameters used for the numerical experiments in Sec. V for demodulation and equalization. Throughout the simulations, we used `PyTorch` [94] adopting `autograd`'s option `create_graph` = `True` to allow the computational graph to calculate second-order derivatives.

For the demodulation problem in Sec. V-B (Figs. 6 – 8), the complex input space $\mathcal{Y} = \mathbb{C}$ is treated as a two-dimensional real vector space $\mathbb{R}^2$ when is fed into the neural network demodulator. The KL term in (20) is suppressed by a multiplicative coefficient of 0.1, as a means to emphasize the average log-likelihood term should have over the prior. This is an approach known as generalized Bayesian inference [79], [95]. To handle the discrepancy in the number of pilots for adaptation during meta-training and meta-testing, i.e., $N_*^{\text{tr}} > N_\tau^{\text{tr}}$, we consider the following strategy akin to burn-in phase [49] during meta-testing as done in [19]: *(i)* start with I updates using learning rate η utilizing N_τ^{tr} pilots among the available N_*^{tr} pilots; *(ii)* then, additional $I_* - I$ updates are performed with reduced learning rate (5% of the original learning rate) with all available N_*^{tr} pilots. This strategy becomes particularly useful in practical scalable systems in which the number of pilots may change depending on the deployment environments.

TABLE III
PARAMETERS FOR THE DEMODULATION AND EQUALIZATION META-LEARNING.

Description	Symbol	Demodulation (Sec. V-B)	Equalization (Sec. V-C)
Signal to noise ratio [dB]	SNR	18	6
Modulation	-	16-QAM	4-PAM
Neural network input dimension	integer	$\dim(\mathbb{C}) = 2$	$\dim(\mathbb{R}^2) = 2$
Neural network layers size [{hidden},output]	neurons per layer	$[10, 30, 30, 16]$	$[, 1]$
Activation of hidden layers	-	ReLU	NA
Activation of last layer	-	softmax	no activation
Meta-training frames mini-batch size	B	16	full batch (t)
Frame-specific learning rate	η	10^{-1}	$2 \cdot 10^{-3}$
Meta-learning rate	κ	10^{-3}	$5 \cdot 10^{-2}$
Number of pilots for frame-specific updates while meta-training	$N_{\mathcal{T}}^{\mathrm{tr}}$	4	4
Number of pilots for meta-updates while meta-training	$N_{\mathcal{T}}^{\mathrm{te}}$	3000	4
Number of pilots for frame-specific updates while meta-testing	N_{*}^{tr}	8	4
Number of symbols with each channel states while meta-testing	N_{*}^{te}	4000	1000
Number of inner SGD updates while meta-training	I	2	2
Number of inner SGD updates while meta-testing	I_*	200	2
Number of meta-updates while meta-training	I_{meta}	200	no. of tasks t
Ensemble size (for Bayesian framework only)	R	100	100
Assumed precision of equalizer	β	-	150
Number of frames forming the initial experience	t_{init}	-	3
Number of meta-training iterations	-	200	100
Number of meta-testing frames averaged over	-	50	100

As for the equalization setting in Sec. V-C (Figs. 9 – 10), we observe that reinitializing the hyperparameter ξ to a random value at each data acquisition iteration benefits meta-training in practice. While using the previous iteration's optimized hyperparameter vector ξ as the starting point for the current iteration is useful in reducing the computational complexity [19], [96], we found it beneficial not to do so in our equalization problem to avoid meta-overfitting especially in the few-frames (e.g., 10 frames) regime of interest.

REFERENCES

[1] K. M. Cohen, S. Park, O. Simeone, and S. Shamai, "Learning to Learn to Demodulate with Uncertainty Quantification via Bayesian Meta-Learning," in *Proc. WSA 2021; 25th International ITG Workshop on Smart Antennas in EURECOM, France*. VDE, 2021, pp. 202–207.

[2] K. B. Letaief, W. Chen, Y. Shi, J. Zhang, and Y.-J. A. Zhang, "The roadmap to 6G: AI Empowered Wireless Networks," *IEEE Communications Magazine*, vol. 57, no. 8, pp. 84–90, 2019.

[3] L. Bonati, S. D'Oro, M. Polese, S. Basagni, and T. Melodia, "Intelligence and learning in O-RAN for data-driven NextG cellular networks," *IEEE Communications Magazine*, vol. 59, no. 10, pp. 21–27, 2021.

[4] P. H. Masur, J. H. Reed, and N. Tripathi, "Artificial Intelligence in Open-Radio Access Network," *IEEE Aerospace and Electronic Systems Magazine*, 2022.

[5] O-RAN Alliance, "O-RAN Working Group 2 AI/ML Workflow Description and Requirements," *ORAN-WG2. AIML. v01.02.02*, vol. 1, 2020.

[6] O. Simeone, S. Park, and J. Kang, "From Learning to Meta-Learning: Reduced Training Overhead and Complexity for Communication Systems," in *Proc. 2020 2nd 6G Wireless Summit (6G SUMMIT) in Levi, Finland*. IEEE, 2020, pp. 1–5.

[7] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On Calibration of Modern Neural Networks," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1321–1330.

[8] D. Barber, *Bayesian Reasoning and Machine Learning*. USA: Cambridge University Press, 2012.

[9] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight Uncertainty in Neural Network," in *International Conference on Machine Learning*. PMLR, 2015, pp. 1613–1622.

[10] H. Wang and D.-Y. Yeung, "A Survey on Bayesian Deep Learning," *ACM Computing Surveys (CSUR)*, vol. 53, no. 5, pp. 1–37, 2020.

[11] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning," in *International Conference on Machine Learning*. PMLR, 2016, pp. 1050–1059.

[12] S. Thrun, "Lifelong Learning Algorithms," in *Learning to Learn*. Springer, 1998, pp. 181–209.

[13] C. Finn, P. Abbeel, and S. Levine, "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks," in *Proceedings of the 34th International Conference on Machine Learning*, vol. 70. PMLR, 06–11 Aug 2017, pp. 1126–1135.

[14] L. Zintgraf, K. Shiarli, V. Kurin, K. Hofmann, and S. Whiteson, "Fast Context Adaptation via Meta-Learning," in *International Conference on Machine Learning*. PMLR, 2019, pp. 7693–7702.

[15] D. Maclaurin, D. Duvenaud, and R. Adams, "Gradient-Based Hyperparameter Optimization Through Reversible Learning," in *International conference on machine learning*. PMLR, 2015, pp. 2113–2122.

[16] Z. Li, F. Zhou, F. Chen, and H. Li, "Meta-SGD: Learning to Learn Quickly for Few-Shot Learning," *arXiv preprint arXiv:1707.09835*, 2017.

[17] H. S. Behl, A. G. Baydin, and P. H. Torr, "Alpha MAML: Adaptive Model-Agnostic Meta-Learning," *arXiv preprint arXiv:1905.07435*, 2019.

[18] J. Baxter, "Theoretical Models of Learning to Learn," in *Learning to learn*. Springer, 1998, pp. 71–94.

[19] S. Park, H. Jang, O. Simeone, and J. Kang, "Learning to Demodulate From Few Pilots via Offline and Online Meta-Learning," *IEEE Transactions on Signal Processing*, vol. 69, pp. 226–239, 2021.

[20] M. Goutay, F. Ait Aoudia, and J. Hoydis, "Deep HyperNetwork-Based MIMO Detection," in *Proc. IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC) in Atlanta, USA*, 2020, pp. 1–5.

[21] Y. Yuan, G. Zheng, K.-K. Wong, B. Ottersten, and Z.-Q. Luo, "Transfer Learning and Meta Learning-Based Fast Downlink Beamforming Adaptation," *IEEE Transactions on Wireless Communications*, vol. 20, no. 3, pp. 1742–1755, 2021.

[22] Y. Hu, M. Chen, W. Saad, H. V. Poor, and S. Cui, "Meta-Reinforcement Learning for Trajectory Design in Wireless UAV Networks," in *Proc. GLOBECOM 2020- IEEE Global Communications Conference, in Taipei, Taiwan*. IEEE, 2020, pp. 1–6.

[23] T. Raviv, S. Park, N. Shlezinger, O. Simeone, Y. C. Eldar, and J. Kang, "Meta-ViterbiNet: Online Meta-Learned Viterbi Equalization for Non-Stationary Channels," in *Proc. 2021 IEEE International Conference on Communications Workshops (ICC Workshops) in Montreal, Canada*. IEEE, 2021, pp. 1–6.

[24] A. E. Kalør, O. Simeone, and P. Popovski, "Latency-Constrained Prediction of mmWave/THz Link Blockages through Meta-Learning," *arXiv preprint arXiv:2106.07442*, 2021.

[25] I. Nikoloska and O. Simeone, "Fast Power Control Adaptation via Meta-Learning for Random Edge Graph Neural Networks," in *Proc. 2021 IEEE 22nd International Workshop on Signal Processing Advances in Wireless Communications (SPAWC) in Lucca, Italy*. IEEE, 2021, pp. 146–150.

[26] J. Zhang, Y. Yuan, G. Zheng, I. Krikidis, and K.-K. Wong, "Embedding Model-Based Fast Meta Learning for Downlink Beamforming Adaptation," *IEEE Transactions on Wireless Communications*, vol. 21, no. 1, pp. 149–162, 2021.

[27] A. E. Kalør, O. Simeone, and P. Popovski, "Prediction of mmWave/THz Link Blockages through Meta-Learning and Recurrent Neural Networks," *IEEE Wireless Communications Letters*, vol. 10, no. 12, pp. 2815–2819, 2021.

[28] Y. Jiang, H. Kim, H. Asnani, and S. Kannan, "Mind: Model Independent Neural Decoder," in *Proc. 2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC) in Cannes, France*. IEEE, 2019, pp. 1–5.

[29] J. Zhang, Y. He, Y.-W. Li, C.-K. Wen, and S. Jin, "Meta Learning-Based MIMO Detectors: Design, Simulation, and Experimental Test," *IEEE Transactions on Wireless Communications*, vol. 20, no. 2, pp. 1122–1137, 2020.

[30] C. Nguyen, T.-T. Do, and G. Carneiro, "Uncertainty in Model-Agnostic Meta-Learning using Variational Inference," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 3090–3100.

[31] K. Posch, J. Steinbrener, and J. Pilz, "Variational Inference to Measure Model Uncertainty in Deep Neural Networks," *arXiv preprint arXiv:1902.10189*, 2019.

[32] Z. Sun, J. Wu, X. Li, W. Yang, and J.-H. Xue, "Amortized Bayesian Prototype Meta-learning: A New Probabilistic Meta-learning Approach to Few-shot Image Classification," in *Proc. International Conference on Artificial Intelligence and Statistics*. PMLR, 2021, pp. 1414–1422.

[33] N. Houlsby, F. Huszár, Z. Ghahramani, and M. Lengyel, "Bayesian Active Learning for Classification and Preference Learning," *arXiv preprint arXiv:1112.5745*, 2011.

[34] Y. Gal, R. Islam, and Z. Ghahramani, "Deep Bayesian Active Learning with Image Data," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1183–1192.

[35] F. Sohrabi, T. Jiang, W. Cui, and W. Yu, "Active Sensing for Communications by Learning," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 6, pp. 1780–1794, 2022.

[36] H. Sahbi, S. Deschamps, and A. Stoian, "Active Learning for Interactive Satellite Image Change Detection," *arXiv preprint arXiv:2110.04250*, 2021.

[37] J. Kaddour, S. Sæmundsson *et al.*, "Probabilistic Active Meta-Learning," *Proc. Advances in Neural Information Processing Systems (NIPS) as Virtual-only Conference*, vol. 33, pp. 20 813–20 822, 2020.

[38] A. Kirsch, J. Van Amersfoort, and Y. Gal, "BatchBALD: Efficient and Diverse Batch Acquisition for Deep Bayesian Active Learning," *Proc. Advances in Neural Information Processing Systems (NIPS) in Vancouver, Canada*, vol. 32, 2019.

[39] I. Nikoloska and O. Simeone, "Bayesian Active Meta-Learning for Black-Box Optimization," in *Proc. 2022 IEEE 23rd International Workshop on Signal Processing Advances in Wireless Communications (SPAWC) in Oulu, Finland*. IEEE, 2022.

[40] X. Zhang, Y.-C. Liang, and J. Fang, "Bayesian Learning Based Multiuser Detection for M2M Communications with Time-Varying User Activities," in *2017 IEEE International Conference on Communications (ICC) in Paris, France*. IEEE, 2017, pp. 1–6.

[41] ——, "Novel Bayesian Inference Algorithms for Multiuser Detection in M2M Communications," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 9, pp. 7833–7848, 2017.

[42] R. Prasad, C. R. Murthy, and B. D. Rao, "Joint Channel Estimation and Data Detection in MIMO-OFDM Systems: A Sparse Bayesian Learning Approach," *IEEE Transactions on Signal Processing*, vol. 63, no. 20, pp. 5369–5382, 2015.

[43] X. Lv, Y. Li, Y. Wu, X. Wang, and H. Liang, "Joint Channel Estimation and Impulsive Noise Mitigation Method for OFDM Systems Using Sparse Bayesian Learning," *IEEE Access*, vol. 7, pp. 74 500–74 510, 2019.

[44] A. Graves, "Practical Variational Inference for Neural Networks," *Proc. Advances in Neural Information Processing Systems (NIPS) in Granada, Spain*, vol. 24, 2011.

[45] M. Dusenberry, G. Jerfel, Y. Wen, Y. Ma, J. Snoek, K. Heller, B. Lakshminarayanan, and D. Tran, "Efficient and Scalable Bayesian Neural Nets with Rank-1 Factors," in *Proc. International Conference on Machine Learning in Baltimore , USA*. PMLR, 2020, pp. 2782–2792.

[46] E. Daxberger, A. Kristiadi, A. Immer, R. Eschenhagen, M. Bauer, and P. Hennig, "Laplace Redux-Effortless Bayesian Deep Learning," *Proc. Advances in Neural Information Processing Systems (NIPS) as Virtual-only Conference*, vol. 34, pp. 20 089–20 103, 2021.

[47] S. Farquhar, L. Smith, and Y. Gal, "Liberty or Depth: Deep Bayesian Neural Nets Do Not Need Complex Weight Posterior Approximations," *Proc. Advances in Neural Information Processing Systems (NIPS) as Virtual-only Conference*, vol. 33, pp. 4346–4357, 2020.

[48] R. M. Neal *et al.*, "MCMC using Hamiltonian Dynamics," *Handbook of Markov Chain Monte Carlo*, vol. 2, no. 11, 2011.

[49] M. Welling and Y. W. Teh, "Bayesian Learning via Stochastic Gradient Langevin Dynamics," in *Proceedings of the 28th International Conference on Machine learning (ICML-11) in Bellevue, Washington, USA*. Citeseer, 2011, pp. 681–688.

[50] R. Zhang, C. Li, J. Zhang, C. Chen, and A. G. Wilson, "Cyclical Stochastic Gradient MCMC for Bayesian Deep Learning," *arXiv preprint arXiv:1902.03932*, 2019.

[51] Q. Liu and D. Wang, "Stein Variational Gradient Descent: A General Purpose Bayesian Inference Algorithm," *Proc. Advances in Neural Information Processing Systems (NIPS) in Barcelona, Spain*, vol. 29, 2016.

[52] O. Narmanlioglu, E. Zeydan, M. Kandemir, and T. Kranda, "Prediction of active UE number with Bayesian neural networks for self-organizing LTE networks," in *2017 8th International Conference on the Network of the Future (NOF)*. IEEE, 2017, pp. 73–78.

[53] Z. Wu and H. Li, "Stochastic Gradient Langevin Dynamics for Massive MIMO Detection," *IEEE Communications Letters*, vol. 26, no. 5, pp. 1062–1065, 2022.

[54] N. Zilberstein, C. Dick, R. Doost-Mohammady, A. Sabharwal, and S. Segarra, "Annealed Langevin Dynamics for Massive MIMO Detection," *arXiv preprint arXiv:2205.05776*, 2022.

[55] Z. Tao and S. Wang, "Improved Downlink Rates for FDD Massive MIMO Systems through Bayesian Neural Networks-Based Channel Prediction," *IEEE Transactions on Wireless Communications*, vol. 21, no. 3, pp. 2122–2134, 2021.

[56] N. K. Jha and V. K. Lau, "Transformer-Based Online Bayesian Neural Networks for Grant-Free Uplink Access in CRAN With Streaming Variational Inference," *IEEE Internet of Things Journal*, vol. 9, no. 9, pp. 7051–7064, 2021.

[57] ——, "Online Downlink Multi-User Channel Estimation for mmWave Systems using Bayesian Neural Network," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 8, pp. 2374–2387, 2021.

[58] J. Xu, Y. Shen, E. Chen, and V. Chen, "Bayesian Neural Networks for Identification and Classification of Radio Frequency Transmitters using Power Amplifiers' Nonlinearity Signatures," *IEEE Open Journal of Circuits and Systems*, vol. 2, pp. 457–471, 2021.

[59] M. Zecchin, S. Park, O. Simeone, M. Kountouris, and D. Gesbert, "Robust Bayesian Learning for Reliable Wireless AI: Framework and Applications," *arXiv preprint arXiv:2207.00300*, 2022.

[60] D. Liu, G. Zhu, Q. Zeng, J. Zhang, and K. Huang, "Wireless Data Acquisition for Edge Learning: Data-Importance Aware Retransmission," *IEEE Transactions on Wireless Communications*, vol. 20, no. 1, pp. 406–420, 2020.

[61] S.-E. Chiu, N. Ronquillo, and T. Javidi, "Active Learning and CSI Acquisition for mmWave Initial Alignment," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 11, pp. 2474–2489, 2019.

[62] K. Yang, J. Ren, Y. Zhu, and W. Zhang, "Active Learning for Wireless IoT Intrusion Detection," *IEEE Wireless Communications*, vol. 25, no. 6, pp. 19–25, 2018.

[63] M. K. Abdel-Aziz, S. Samarakoon, M. Bennis, and W. Saad, "Ultra-Reliable and Low-Latency Vehicular Communication: An Active Learning Approach," *IEEE Communications Letters*, vol. 24, no. 2, pp. 367–370, 2019.

[64] T. M. Hospedales, A. Antoniou, P. Micaelli, and A. J. Storkey, "Meta-Learning in Neural Networks: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[65] C. Finn, K. Xu, and S. Levine, "Probabilistic Model-Agnostic Meta-Learning," *Proc. Advances in Neural Information Processing Systems (NIPS) in Montreal, Canada*, vol. 31, 2018.

[66] S. Ravi and A. Beatson, "Amortized bayesian meta-learning," in *Proc. International Conference on Learning Representations (ICLR), in Vancouver, Canada*, 2018.

[67] J. Yoon, T. Kim, O. Dia, S. Kim, Y. Bengio, and S. Ahn, "Bayesian Model-Agnostic Meta-Learning," *Proc. Advances in neural information processing systems (NIPS), in Montreal, Canada*, vol. 31, 2018.

[68] M. Patacchiola, J. Turner, E. J. Crowley, M. O'Boyle, and A. J. Storkey, "Bayesian Meta-Learning for the Few-Shot Setting via Deep Kernels," *Proc. Advances in Neural Information Processing Systems (NIPS) as Virtual-only Conference*, vol. 33, pp. 16 108–16 118, 2020.

[69] Y. Zou and X. Lu, "Gradient-EM Bayesian Meta-Learning," *Proc. Advances in Neural Information Processing Systems (NIPS) as Virtual-only Conference*, vol. 33, pp. 20 865–20 875, 2020.

[70] R. Amit and R. Meir, "Meta-Learning by Adjusting Priors Based on Extended PAC-Bayes Theory," in *Proc. International Conference on Machine Learning (ICML) in Stockholm, Sweden*. PMLR, 2018, pp. 205–214.

[71] J. Rothfuss, V. Fortuin, M. Josifoski, and A. Krause, "PACOH: Bayes-Optimal Meta-Learning with PAC-Guarantees," in *(ICML), as Virtual-only Conference*. PMLR, 2021, pp. 9116–9126.

[72] J. Rothfuss, D. Heyn, A. Krause *et al.*, "Meta-Learning Reliable Priors in the Function Space," *Proc. Advances in Neural Information Processing Systems (NIPS) as Virtual-only Conference*, vol. 34, pp. 280–293, 2021.

[73] S. T. Jose, S. Park, and O. Simeone, "Information-Theoretic Analysis of Epistemic Uncertainty in Bayesian Meta-learning," in *Proc. International Conference on Artificial Intelligence and Statistics (AISTATS) as a Virtual-only Conference*. PMLR, 2022, pp. 9758–9775.

[74] I. Nikoloska and O. Simeone, "Quantum-Aided Meta-Learning for Bayesian Binary Neural Networks via Born Machines," *arXiv preprint arXiv:2203.17089*, 2022.

[75] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT Press, 2016.

[76] O. Simeone, "A Brief Introduction to Machine Learning for Engineers," *Foundation and Trends in Signal Processing*, vol. 13, no. 12, Aug. 2018.

[77] E. Angelino, M. J. Johnson, R. P. Adams *et al.*, "Patterns of Scalable Bayesian Inference," *Foundations and Trends® in Machine Learning*, vol. 9, no. 2-3, pp. 119–247, 2016.

[78] T. M. Cover and J. A. Thomas, "Information Theory and Statistics," *Elements of Information Theory*, vol. 1, no. 1, pp. 279–335, 1991.

[79] S. T. Jose and O. Simeone, "Free Energy Minimization: A Unified Framework for Modeling, Inference, Learning, and Optimization [lecture notes] ," *IEEE Signal Processing Magazine*, vol. 38, no. 2, pp. 120–125, 2021.

[80] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[81] E. Grant, C. Finn, S. Levine, T. Darrell, and T. Griffiths, "Recasting Gradient-Based Meta-Learning as Hierarchical Bayes," *arXiv preprint arXiv:1801.08930*, 2018.

[82] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[83] O. Simeone, *Machine Learning for Engineers*. Cambridge University Press, 2022.

[84] A. Griewank, "Some Bounds on the Complexity of Gradients, Jacobians, and Hessians," in *Complexity in Numerical Optimization*. World Scientific, 1993, pp. 128–162.

[85] A. Rajeswaran, C. Finn, S. M. Kakade, and S. Levine, "Meta-Learning with Implicit Gradients," in *Proc. Neural Information Processing Systems (NIPS), in Vancouver, Canada*, 2019, pp. 113–124.

[86] S. Mohamed, M. Rosca, M. Figurnov, and A. Mnih, "Monte Carlo Gradient Estimation in Machine Learning," *Journal of Machine Learning Research (JMLR)*, vol. 21, no. 132, pp. 1–62, 2020.

[87] T. Melluish, C. Saunders, I. Nouretdinov, and V. Vovk, "Comparing the Bayes and Typicalness Frameworks," in *European Conference on Machine Learning (ECML) in Freiburg, Germany*. Springer, 2001, pp. 360–371.

[88] M. H. DeGroot and S. E. Fienberg, "The Comparison and Evaluation of Forecasters," *Journal of the Royal Statistical Society: Series D (The Statistician)*, vol. 32, no. 1-2, pp. 12–22, 1983.

[89] Y. Zhang, A. Doshi, R. Liston, W.-t. Tan, X. Zhu, J. G. Andrews, and R. W. Heath, "DeepWiPHY: Deep Learning-Based Receiver Design and Dataset for IEEE 802.11 ax systems," *IEEE Transactions on Wireless Communications*, vol. 20, no. 3, pp. 1596–1611, 2020.

[90] A. G. Helmy, M. Di Renzo, and N. Al-Dhahir, "On the Robustness of Spatial Modulation to I/Q Imbalance," *IEEE Communications Letters*, vol. 21, no. 7, pp. 1485–1488, 2017.

[91] D. Tandur and M. Moonen, "Joint Adaptive Compensation of Transmitter and Receiver IQ Imbalance under Carrier Frequency Offset in OFDM-Based Systems," *IEEE Transactions on Signal Processing*, vol. 55, no. 11, pp. 5246–5252, 2007.

[92] M. Yin, G. Tucker, M. Zhou, S. Levine, and C. Finn, "Meta-Learning without Memorization," *arXiv preprint arXiv:1912.03820*, 2019.

[93] R. Li, O. Bohdal, R. Mishra, H. Kim, D. Li, N. Lane, and T. Hospedales, "A Channel Coding Benchmark for Meta-Learning," in *Proc. Advances in Neural Information Processing Systems (NIPS) as a Virtual-only Conference, Track on Datasets and Benchmarks*, 2021.

[94] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[95] J. Knoblauch, J. Jewson, and T. Damoulas, "Generalized Variational Inference: Three Arguments for Deriving New Posteriors," *arXiv preprint arXiv:1904.02063*, 2019.

[96] C. Finn, A. Rajeswaran, S. Kakade, and S. Levine, "Online Meta-Learning," in *International Conference on Machine Learning (ICML) in Long Beach, CA, USA*. PMLR, 2019, pp. 1920–1930.

Kfir M. Cohen (Student Member, IEEE) is pursuing his Ph.D. in the King's Communications, Learning and Information Processing lab at the Department of Engineering of King's College London (KCL). He received B.Sc. (summa cum laude) and M.Sc. degrees in 2006 and 2013, respectively, both in the Electrical Engineering Faculty of the Technion – Israel Institute for Technology, Haifa, Israel. He served for 15 years in different R&D roles. Before joining KCL, his last position was as a communication signal processing engineer. His research interests are Bayesian and reliable machine learning, signal processing, as well as their applications to communications systems.

Sangwoo Park (Member, IEEE) received his B.S. degree in physics in 2014; M.S.E and Ph.D. degrees in electrical engineering in 2016 and 2020, all from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea. He is currently a research associate in the Department of Engineering, King's Communications, Learning and Information Processing lab, King's College London, United Kingdom. His research interests lie in practical, reliable AI and its application for wireless communication systems and quantum information processing.

Osvaldo Simeone (Fellow, IEEE) is a Professor of Information Engineering with the Centre for Telecommunications Research at the Department of Engineering of King's College London, where he directs the King's Communications, Learning and Information Processing lab. He received an M.Sc. degree (with honors) and a Ph.D. degree in information engineering from Politecnico di Milano, Milan, Italy, in 2001 and 2005, respectively. From 2006 to 2017, he was a faculty member of the Electrical and Computer Engineering (ECE) Department at New Jersey Institute of Technology (NJIT), where he was affiliated with the Center for Wireless Information Processing (CWiP). His research interests include information theory, machine learning, wireless communications, neuromorphic computing, and quantum machine learning. Dr Simeone 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, the 2015 IEEE Communication Society Best Tutorial Paper Award and of the Best Paper Awards of IEEE SPAWC 2007 and IEEE WRECOM 2007. He was awarded an Open Fellowship by the EPSRC in 2022 and a Consolidator grant by the European Research Council (ERC) in 2016. His research has been also supported by the U.S. National Science Foundation, the European Commission, the European Research Council, the Vienna Science and Technology Fund, the European Space Agency, as well as by a number of industrial collaborations including with Intel Labs and InterDigital. He is the Chair of the Signal Processing for Communications and Networking Technical Committee of the IEEE Signal Processing Society and of the UK & Ireland Chapter of the IEEE Information Theory Society. He is currently a Distinguished Lecturer of the IEEE Communications Society, and he was a Distinguished Lecturer of the IEEE Information Theory Society in 2017 and 2018. Dr Simeone is the author of the textbook "Machine Learning for Engineers" published by Cambridge University Press, four monographs, two edited books, and more than 180 research journal and magazine papers. He is a Fellow of the IET, EPSRC, and IEEE.

Shlomo Shamai (Shitz) (Life Fellow, IEEE) is with the Department of Electrical Engineering, Technion—Israel Institute of Technology, where he is a Technion Distinguished Professor, and holds the William Fondiller Chair of telecommunications. Dr. Shamai is an URSI Fellow, a member of the Israeli Academy of Sciences and Humanities, and a Foreign Member of the U.S. National Academy of Engineering. He was a recipient of the 2011 Claude E. Shannon Award, the 2014 Rothschild Prize in Mathematics/Computer Sciences and Engineering, and the 2017 IEEE Richard W. Hamming Medal. He was a co-recipient of the 2018 Third Bell Labs Prize for Shaping the Future of Information and Communications Technology. He was also a recipient of numerous technical and paper awards and recognitions of the IEEE (Donald G. Fink Prize Paper Award), Information Theory, Communications and Signal Processing Societies and EURASIP. He is listed as a Highly Cited Researcher (computer science) for the years 2013/4/5/6/7/8. He has served as an Associate Editor for the Shannon Theory of the IEEE Transactions on Information Theory and has also served twice on the Board of Governors of the Information Theory Society. He has also served on the Executive Editorial Board of the IEEE Transactions on Information Theory, the IEEE Information Theory Society Nominations and Appointments Committee, and the IEEE Information Theory Society (Shannon Award Committee).