Learning to Learn to Demodulate with Uncertainty Quantification via Bayesian Meta-Learning

Kfir M. Cohen\(^1\), Sangwoo Park\(^1\), Osvaldo Simeone\(^1\), and Shlomo Shamai (Shitz)\(^2\)

\(^1\) KCLIP, CTR, Department of Engineering, King’s College London, UK
\(^2\) Viterbi Electrical Engineering Department, Technion—Israel Institute of Technology, Haifa, Israel

\({\text{kfir.cohen, sangwoo.park, osvaldo.simeone}@kcl.ac.uk, sshlomo@ee.technion.ac.il}\)

Abstract—Meta-learning, or learning to learn, offers a principled framework for few-shot learning. It leverages data from multiple related learning tasks to infer an inductive bias that enables fast adaptation on a new task. The application of meta-learning was recently proposed for learning how to demodulate from few pilots. The idea is to use pilots received and stored for offline use from multiple devices in order to meta-learn an adaptation procedure with the aim of speeding up online training on new devices. Standard frequentist learning, which can yield relatively accurate “hard” classification decisions, is known to be poorly calibrated, particularly in the small-data regime. Poor calibration implies that the soft scores output by the demodulator are inaccurate estimates of the true probability of correct demodulation. In this work, we introduce the use of Bayesian meta-learning via variational inference for the purpose of obtaining well-calibrated few-pilot demodulators. In a Bayesian framework, each neural network weight is represented by a distribution, capturing epistemic uncertainty. Bayesian meta-learning optimizes over the prior distribution of the weights. The resulting Bayesian ensembles offer better calibrated soft decisions, at the computational cost of running multiple instances of the neural network for demodulation. Numerical results for single-input single-output Rayleigh fading channels with transmitter’s non-linearities are provided that compare symbol error rate and expected calibration error for both frequentist and Bayesian meta-learning, illustrating how the latter is both more accurate and better-calibrated.

I. INTRODUCTION

A. Background and Motivation

Consider a wireless packet-based Internet-of-Things (IoT)-like setting, in which multiple edge devices transmit to a single access point under dynamic wireless fading channels. Since IoT devices transmit sporadically using short packets, only a small number of symbols are available as pilots. Therefore, adapting the demodulator using conventional machine learning generally fails to produce effective solutions.

As demonstrated in \([1], [2]\), meta-learning, or learning to learn \([3]\), can alleviate this problem by leveraging pilot data received previously from other similar devices. The general principle behind meta-learning is to optimize an inductive bias \([4]–[8]\) based on the observation of data from multiple related problems, so as to facilitate training for new tasks using small amount of training data \([9]\).

Fig. 1. The meta-learning problem studied in this work for the example of 16-ary quadrature amplitude modulation (16-QAM): Given meta-training data sets \(\{\mathcal{D}_t\}_{t=1}^{T}\) of pilots from previously active devices, the demodulator adapts with few pilot data \(\mathcal{D}_t^*\) to a new device. The output of this adaptation is the device-specific parameter vector \(\phi_\ast\), that defines the soft demodulator \(p(x_t^*|y_t^*, \phi_\ast)\) acting over the newly received data symbols \(y_t^*\). In the Bayesian meta-learning framework, instead of a single demodulator \(\phi_\ast\), we have an ensemble of demodulator parameter vectors, that follows posterior distribution \(p(\phi_\ast|\mathcal{D}_t^*, \xi)\) defined with meta-learned hyperprior parameter \(\xi\).

Prior work on meta-learning for communication systems, including \([1], [2]\), as well as \([10]–[16]\), implements standard frequentist learning, hence focusing on point estimates of the model parameters. Frequentist learning is known to have poor calibration performance, providing over-confident decisions that fail to capture epistemic uncertainty \([17]\) in the regime of small data, i.e., of few pilots. In contrast, Bayesian learning \([18]\) captures epistemic uncertainty by optimizing over probability distributions, rather than point estimates in the model parameter space. The distribution over the neural network weights produced by Bayesian learning accounts for the disagreement of the decisions provided by different models on the training set, as well as for prior knowledge on the model parameters. As a result, Bayesian learning can effectively account for epistemic uncertainty due to limited data.

Bayesian meta-learning aims at optimizing the prior distribution used for Bayesian inference, based on data from multiple tasks \([19], [20]\). Most methods implement empirical Bayes \([21]\), either via parametric variational inference (VI) \([22]–[24]\) or via particle-based VI \([25]\). Fully Bayesian meta-learning approaches that treat the hyperprior parameters as random variables were considered in \([26]–[28]\). No attempt to date...
has been presented to leverage Bayesian meta-learning for communication systems.

B. Contributions

Fig. 1 shows the considered scenario: By using available pilots from multiple devices, the meta-learner learns how to quickly adapt the demodulator matched to any new transmitting device based on few pilots. As discussed, reference [1] has demonstrated how frequentist meta-learning can leverage pilots from previous transmissions by other devices to enable training of an effective hard demodulator with few pilots at deployment time. However, in many scenarios, demodulation needs to output reliable soft estimates, e.g., for use in channel decoders. This requires the demodulator to be well calibrated.

In this paper, we propose to leverage the mentioned well-known calibration properties of Bayesian learning [29], along with the sample efficiency of meta-learning, to obtain well-calibrated soft predictors from few pilots. To this end, we adopt an empirical Bayesian model, which is addressed via VI, as illustrated in Fig. 2. The approach is akin to VAMPIRE [19] due to its reliance on Gaussian variational posteriors. From numerical results, we show that the proposed approach offers better accuracy (lower Symbol Error Rate (SER)) as well as improved calibration performance in terms of Expected Calibration Error (ECE) [30].

The rest of the paper is organized as follows. Section II introduces the channel model and notations of conventional learning. Section III briefly recalls frequentist meta-learning, which we take as benchmark for the main contribution in Section IV describing Bayesian meta-learning for demodulation. Numerical results are shown in Section V.

II. CHANNEL MODEL AND CONVENTIONAL-LEARNING

A. Channel Model

In this paper, we consider packet-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 $c$. The channel state $c$ is constant within each block, and it is independently and identically distributed (i.i.d.) across blocks according to a given unknown distribution $p(c)$. At block $\tau$, the transmitter sends a packet consisting of $N_t$ symbols $x_\tau = \{x_\tau[i]\}_{i=1}^{N_t}$, which are received as $y_\tau = \{y_\tau[i]\}_{i=1}^{N_t}$ with $y_\tau[i] \sim p(y_\tau[i]|x_\tau[i],c_\tau)$ and channel state $c_\tau$.

A soft demodulator is a conditional distribution $p(x|y)$ which maps channel outputs $y \in \mathcal{Y}$ to estimated probabilities of all symbols in $\mathcal{X}$. The demodulation is applied separately to each received sample $y_\tau[i]$ in a memoryless fashion.

B. Conventional Data-Driven Demodulators

Pilot-aided demodulation schemes utilize available pilots to adapt the demodulator to the instantaneous channel state $c_\tau$ in each block. A typical choice for a trainable soft demodulator is a fully-connected neural-network with ReLU activations and softmax at the last layer [31], e.g.,

$$p(x|y,\phi) = \frac{\exp((W_L \cdot f_{W_{L-1}, b_{L-1}} \circ \cdots \circ f_{W_1, b_1}(y) + b_L)_x)}{\sum_{x' \in \mathcal{X}} \exp((W_L \cdot f_{W_{L-1}, b_{L-1}} \circ \cdots \circ f_{W_1, b_1}(y) + b_L)_{x'})},$$

where $\circ$ is the composition operator; $W_l$ and $b_l$ are the weights and biases of the $l$-th layer; $f_{W_l,b_l}$ is a linear mapping followed by ReLU activation defined as $y_t = f_{W_l,b_l}(y_{t-1}) = \text{ReLU}(W_l \cdot y_{t-1} + b_l)$ with $y_0 = y_t$; the rectified linear unit activation is applied element-wise as $\text{ReLU}(y) = \max\{y,0\}$; and $[.]_x$ stands for the $x$-th element of a vector. All the weights and biases are stacked together to form the model parameters vector $\phi := \{W_l, b_l\}_{l=1}^L$, having a total of $D$ parameters.

In each block $\tau$, conventional learning optimizes the model parameters $\phi_\tau$ using $N_t^\tau < N_t$ pilots $D_\tau^\tau = \{(y_\tau[i], x_\tau[i])\}_{i=1}^{N_t^\tau} \subset \mathcal{D}_\tau$ as training data. Optimization aims at minimizing the training log-loss

$$\mathcal{L}_{D_\tau^\tau}(\phi_\tau) := -\frac{1}{N_t^\tau} \sum_{i=1}^{N_t^\tau} \log p(x_\tau[i]|y_\tau[i],\phi_\tau)$$

via gradient descent (GD) [17]. GD updates parameter vector $\phi_\tau$ for $I$ iterations with learning rate $\eta > 0$ starting from an initialization $\xi$. Accordingly, the updated parameters $\phi_\tau := \phi_{GD}(D_\tau^\tau,\xi,\eta,I)$ are obtained via the iterations

$$\phi_\tau^{(0)} = \xi, \quad \forall i = 1, \ldots, I : \phi_\tau^{(i)} \leftarrow \phi_\tau^{(i-1)} - \eta \nabla_{\phi_\tau^{(i-1)}} \mathcal{L}_{D_\tau^\tau}(\phi_\tau^{(i-1)}),$$

$$\phi_{GD}(D_\tau^\tau,\xi,\eta,I) = \phi_\tau^{(I)}.$$
at each meta-iteration and have a total of $N_T = \sum_{\tau \in T} N_\tau$ samples. The outer loop addresses the outer optimization in Fig. 3 via the gradient step with learning-rate $\kappa > 0$

$$\xi \leftarrow \xi - \kappa \frac{1}{N_T} \sum_{\tau \in T} N_\tau \nabla \mathcal{L}_{D^\tau}(\phi^{GD}(D^\tau_\tau | \xi, \eta, I)).$$

For any new block, its $N^*_\tau$ pilots symbols $D^*_\tau = \{(y^*_\tau[i], x^*_\tau[i])\}_{i=1}^{N^*_\tau}$, together with the meta-learned initialization hyperparameter $\xi$, are used by applying the GD update (Fig. 3) yielding $\phi_* = \phi^{GD}(D^*_\tau | \xi, \eta, I)$. These model parameters are then used to demodulate the payload data symbols $\{y^*_\tau[i]\}_{i=1}^{N^*_\tau}$ of the current block via the soft demodulator $p(x^*_\tau | y^*_\tau[i], \phi_*)$.

**IV. BAYESIAN VARIATIONAL-INFERENCE META-LEARNING**

Instead of producing a single demodulator parameters $\phi_\tau = \phi^{GD}(D^\tau_\tau | \xi, \eta, I)$ in each block $\tau$, **Bayesian learning** optimizes a distribution $p(\phi_\tau | D^\tau_\tau, \xi)$ over the space of the demodulator parameters $\phi_\tau$ as a function of the training data $D^\tau_\tau$ and of hyperprior parameter vector $\xi$. Frequentist meta-learning, reviewed in the previous section, can be viewed as a special case for which we have the deterministic choice $p(\phi_\tau | D^\tau_\tau, \xi) = \delta(\phi_\tau - \phi^{GD}(D^\tau_\tau | \xi, \eta, I))$, where, with some abuse of notation, we have used the hyperprior vector $\xi$ to denote the initialization in Fig. 3. The frequentist approach is inherently limited in its capacity to express parameters uncertainty due to limited data.

Following empirical Bayes [21], we optimize the deterministic hyperprior parameter $\xi$ based on meta-training data $D_{1:t}$ as illustrated in Fig. 4. The hyperprior parameter $\xi$ defines the prior of the demodulator parameters, which we choose as

$$p(\phi_\tau | \xi) = \mathcal{N}(\phi_\tau | \nu, \text{Diag}(\exp(2\phi)))$$

with $\xi = [\nu^T, \phi^T]^T$, where $\nu \in \mathbb{R}^D$ and $\phi \in \mathbb{R}^D$ stand for the mean and exponent of the standard deviation, respectively, and the exponent function is applied element-wise.

To enable optimization, we introduce a variational distribution approximation $q(\phi_{\tau} | \phi_\tau) \approx p(\phi_\tau | D^\tau_\tau, \xi)$, which depends on a learnable vector of variational parameters $\varphi$. Following variational inference, this vector is optimized in each block $\tau$ as

$$\varphi_\tau = \arg\min_{\varphi} \mathcal{F}_{D^\tau_\tau}(\varphi, \xi)$$

Fig. 2. Evolution of network weights between two neuron layers, from deterministic to Bayesian amortized VI: (a) in a frequentist, deterministic, network, each weight is described by a scalar value; (b) which can be viewed as random variable having a degenerated probabilistic distribution Dirac’s delta function; (c) in Bayesian learning, the weights are with a posterior distribution given by the data, that generalizes the frequentist point estimate (dashed vertical line); (d) in VI, the posterior is approximated with a parameter distribution.

Fig. 3. Frequentist meta-testing for soft demodulation over channel with state $c_*$. Meta-learning optimizes an initialization $\xi$ from the pilots obtained in previous blocks $D_{1:t}$, and transfers this knowledge to assist meta-testing a new block. Conventional learning is a special case for which there is no access to meta-training pilots (e.g. random initialization $\xi$).

Fig. 4. Probabilistic graphical model [32] for Bayesian meta learning. Circles represent random variables; double-lined circles represent deterministic variables; gray shaded circles represent observations; and plaques indicate multiple instances.
by minimizing the variational free energy \[ \mathcal{F}_{D^T} (\varphi, \xi) = N^r_T \mathbb{E}_{q(\varphi) \mid \varphi} [\mathcal{L}_{D^T} (\varphi)] + \text{KL}(q(\varphi) \mid \Pi(\varphi)) \] \[ (8) \]
The first summand is a scaled expectation of the log-likelihood \[ \left( \text{2} \right) \] of the training data set over the distributions of model parameters, whereas the second summand is a regularizer that penalizes variational distributions that are dissimilar to the hyperprior parametrized by \( \xi \). Denoting \( \varphi_r = [\nu_r^T, \xi_r^T]^T \), the variational distribution is chosen as
\[ q(\varphi_r | \varphi_r) = N(\nu_r | \nu_r, \text{Diag}(\exp(2\varphi_r))) \]

(9)

Algorithm 1: Bayesian Meta-Training

Inputs: \( D_{\text{te}} \) = labelled data sets of \( t \) training devices
Parameters: \( B \) = number of devices per meta-update batch
\( I \) = number of local update steps
\( \eta, \kappa \) = device-specific/meta learning rates
\( R^\tau \) = train/test ensemble sizes
Output: \( \xi \) = Meta-learned hyperprior parameter

1. initialize \( \xi \)
2. while \( \xi \) not converged do
3. \( T \) = random batch of \( B \) blocks
4. for \( \tau \) \( \in \) \( T \) do
5. randomly divide \( D_{\tau} = \{ D^r_{\tau}, D^e_{\tau} \} \)
6. device-specific update
7. initialize variational parameter \( \varphi_r^{(0)} \) \( \leftarrow \) \( \xi \)
8. for \( i \in [1 : I] \) local update steps do
9. \( \varphi_r^{(i)} \leftarrow \varphi_r^{(i-1)} - \eta \nabla_{\varphi_r^{(i-1)}} \mathcal{F}_{D^T}(\varphi_r^{(i-1)}, \xi) \)
10. using (12)
11. set device-specific variational parameter \( \varphi_r \leftarrow \varphi_r^{(I)} \)
12. meta-update
13. \( \xi \leftarrow \kappa \nabla_{\xi} \mathcal{L}_{D^T}(\xi), \) based on (14)
14. return \( \xi \)

Bayesian meta-training tackles the bi-level problem
\[ \min_{\xi} \left\{ N_T \sum_{r=1}^t N^\text{te}_r \mathbb{E}_{q(\varphi) \mid \varphi} [\mathcal{L}_{D^T} (\varphi)] \mid \mathcal{L}_{D^T} (\varphi) \right\} \]

(10)
via an SGD based nested loop algorithm. To this end, as summarized in Algorithm 1, the inner loop updates the block-specific variational parameters \( \varphi_r \) by minimizing the free energy \( \mathcal{F}_{D^T} (\varphi, \xi) \) in (8) separately for each block \( \tau \) within a mini-batch \( T \) via GD. This yields the updates
\[ \varphi_r^{(0)} = \xi, \]

(11a)
\[ \forall i = 1, \ldots, I : \varphi_r^{(i)} \leftarrow \varphi_r^{(i-1)} - \frac{\eta}{N^\text{te}_r} \nabla_{\varphi_r^{(i-1)}} \mathcal{F}_{D^T}(\varphi_r^{(i-1)}, \xi), \]

(11b)
\[ \varphi_r^{\text{GD}} (D^T_{\tau} | \xi, \eta, I) = \varphi_r^{(I)}, \]

(11c)
where \( \mathcal{F}_{D^T} \) is the estimate of the free energy \( \mathcal{F}_{D^T} \) obtained via the reparameterization trick \[ \text{33} \] as
\[ \mathcal{F}_{D^T} (\varphi, \xi) := \frac{1}{N^\text{te}_r} \sum_{r=1}^{R^e} \left( \mathcal{L}_{D^T} (\varphi_r, \varphi_r) \right) \]

(12)

which uses \( R^e \) demodulator parameters drawn from the current variational posterior \( q(\varphi_r | \varphi_r) \)
as
\[ \varphi_r (e_r, \varphi_r) = \nu_r + \exp (\varphi_r) \odot e_r, \]

(13)
with \( \odot \) being the element-wise multiplication and with an auxiliary random standard Gaussian vector \( \nu_r \sim N(0, I_D) \). The regularizer term is given in closed form as
\[ \text{KL}(q(\varphi_r | \varphi_r) || p(\varphi_r | \xi)) = \frac{1}{2} \sum_{d=1}^D \left[ 2(\varphi_r [d] - \nu_r [d]) + (\nu_r [d] - \nu_r)^2 \right] \]

The overall algorithm is detailed in Algorithm 1.

As illustrated in Fig. 5 for a new block, given pilots \( D^e_{\tau} \), Bayesian meta-testing obtains block-specific variational parameter \( \varphi_r = \varphi^{\text{GD}} (D^e_{\tau} | \xi, \eta, I) \) via (11c) to define the block-specific distribution \( q(\varphi_r | \varphi_r) \). Demodulation is done using \( R^e \) demodulator parameters drawn as \( \phi_r (e_r, \varphi_r) \sim q(\varphi_r | \varphi_r) \) with \( r = 1, \ldots, R^e \). Specifically, we obtain the ensemble demodulation for \( i = 1, \ldots, N^e_r \) as the ensemble predictor

\[ p(\phi^e_r [i] | y^e_r [i], D^e_{\tau}, \xi) := \frac{1}{R^e} \sum_{r=1}^{R^e} p(\phi^e_r [i] | y^e_r [i], \phi_r (e_r, \varphi_r)). \]

(15)
V. EXPERIMENTS

We consider a transmitter with I/Q imbalance [35], [36] followed by a block fading Rayleigh channel as in [1]. We further assume 16-QAM constellation $\mathcal{X} = 1/\sqrt{10} \{\pm1, \pm3\} + j\{\pm1, \pm3\}$ for all transmissions, and baseband received alphabet $\mathcal{Y} = \mathbb{C}$. For each block $\tau$, the channel-state $c_\tau$ consists of the tuple: (a) amplitude imbalance $\epsilon_\tau \in [0, 0.15]$, (b) phase imbalance $\delta_\tau \in [0, 15^\circ]$; (c) Rayleigh fading $h_\tau \in \mathbb{C}$. All of them are drawn i.i.d. across different blocks and are fixed during each block. Precisely, the transmitted symbol $x_{\tau}[i] \in \mathcal{X}$ is distorted by I/Q imbalance [37] as

$$\begin{bmatrix}
    x_{\tau,1}[i] \\
    x_{\tau,0}[i]
\end{bmatrix}
= \begin{bmatrix}
    1 + \epsilon_\tau & 0 \\
    0 & 1 - \epsilon_\tau
\end{bmatrix}
\begin{bmatrix}
    \cos \delta_\tau & -\sin \delta_\tau \\
    -\sin \delta_\tau & \cos \delta_\tau
\end{bmatrix}
\begin{bmatrix}
    x_{\tau,1}[i] \\
    x_{\tau,0}[i]
\end{bmatrix}$$

followed by fading complex channel with gain $h_\tau \sim \mathcal{CN}(0, 1)$

$$y_{\tau}[i] = h_\tau (\bar{x}_{\tau,1}[i] + j\bar{x}_{\tau,0}[i]) + v_{\tau}[i]$$

and noise $v_{\tau}[i] \sim \mathcal{CN}(0, 1)$. The complex input $\mathcal{Y} = \mathbb{C}$ is treated as $\mathbb{R}^2$ when is fed as the network’s input, and is followed by fully connected neurons layers with $[10, 30, 30, 16]$ neurons, ending with a softmax function.

![Fig. 6](image1.png)

**Fig. 6.** Symbol error rate as a function of the number $t$ of meta-training blocks, with 16-QAM, Rayleigh fading, and I/Q imbalance for $N_0 = 4$, $N_1 = 8$. The symbol error rate is averaged over by $N_0 = 4000$ data symbols and 50 meta-test devices with all ensembles size of 100.

![Fig. 7](image2.png)

**Fig. 7.** Expected calibration error (ECE) over meta-test data $D_{te}^*$ as a function of the number $t$ of meta-training blocks, same setting as in Fig. 6.

![Fig. 8](image3.png)

**Fig. 8.** Reliability diagrams (top) for frequentist meta-learning (left) and Bayesian meta-learning (right) with $t = 16$ meta-training devices across 50 meta-test devices, using $N_0 = 4000$ data symbols and ensembles sized 100. 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 $|B_m|/N$ of prediction over $M = 10$ bins.
are roughly half those obtained by frequentist meta-learning provided enough meta-training blocks. Fig. [8] depicts reliability diagrams [30], which provide a way to show the gap between a predictor’s confidence and its actual prediction accuracy. Each bin $m = 1, \ldots, M = 10$ gathers all samples $B_m$ having predictions within the interval $[(m-1)/M, m/M]$. Both accuracy $acc(B_m)$ in (17b) and confidence $conf(B_m)$ in (17c) are plotted as bars. For an ideal predictor operating solely under aleatoric uncertainty, the confidence would match the actual accuracy. Therefore, any deviation from the line when accuracy equals confidence indicates over-confidence ($conf(B_m) > acc(B_m)$) or under-confidence ($conf(B_m) < acc(B_m)$). This ECE discussed previously is a scalar metric obtained by weighting the gaps $|17a|$. From Fig. [8] frequentist meta-learning shows over-confident behavior, while Bayesian meta-learning has reliable uncertainty quantification.

VI. CONCLUSIONS

In this paper, we have extended frequentist approach of meta-learning soft demodulator as proposed in [1] to a Bayesian meta-learning framework, enabling each weight and bias of the demodulator neural network’s parameters to represent uncertainty around its mean value. At the cost of learning and running multiple demodulators forming the ensemble predictor, the proposed Bayesian meta-learning based demodulator not only shows higher accuracy, but it also enjoys better-calibration performance than its frequentist counterpart. This better calibration ability can be practically utilized by post processing blocks like forward-error-correction decoders.

REFERENCES

[1] 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.
[2] O. Simeone, S. Park, and J. Kang, “From learning to meta-learning: Reduced training overhead and complexity for communication systems,” in 2020 2nd 6G Wireless Summit (6G SUMMIT). IEEE, 2020, pp. 1–3.
[3] S. Thrun, “Lifelong learning algorithms,” in Learning to learn. Springer, 1998, pp. 181–209.
[4] 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, ser. Proceedings of Machine Learning Research, vol. 70. PMLR, 06–11 Aug 2017, pp. 1126–1135.
[5] L. Zintgraf, K. Shiarli, V. Kuri, K. Hofmann, and S. Whiteson, “Fast context adaptation via meta-learning,” in International Conference on Machine Learning. PMLR, 2019, pp. 7693–7702.
[6] 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.
[7] 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.
[8] H. S. Behl, A. G. Baydin, and P. H. Torr, “Alpha mam: Adaptive model-agnostic meta-learning,” arXiv preprint arXiv:1905.07435, 2019.
[9] J. Baxter, “Theoretical models of learning to learn,” in Learning to learn. Springer, 1998, pp. 71–94.
[10] M. Goutay, F. Ait Aoudia, and I. Hodeyis, “Deep HyperNetwork-Based MIMO Detection,” in IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), 2020, pp. 1–5.
[11] 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.
[12] Y. Hu, M. Chen, W. Saad, H. V. Poor, and S. Cui, “Meta-reinforcement learning for trajectory design in wireless uav networks,” in GLOBECOM 2020 - IEEE Global Communications Conference. IEEE, 2020, pp. 1–6.
[13] 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,” arXiv preprint arXiv:2103.13483, 2021.
[14] A. E. Kalr, O. Simeone, and P. Popovski, “Latency-constrained prediction of mmwave/htz link blockages through meta-learning,” arXiv preprint arXiv:2106.07442, 2021.
[15] I. Nikoloska and O. Simeone, “Fast power control adaptation via meta-learning for random edge graph neural networks,” arXiv preprint arXiv:2105.00459, 2021.
[16] 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, 2021.
[17] O. Simeone, “A brief introduction to machine learning for engineers,” Foundation and Trends in Signal Processing, vol. 13, no. 12, Aug. 2018. [Online]. Available: https://nms.kcl.ac.uk/osvaldo.simeone/ml4eng.html
[18] D. Barber, Bayesian Reasoning and Machine Learning. USA: Cambridge University Press, 2012.
[19] C. Nguyen, T.-T. Do, and G. Carneiro, “Uncertainty in Model-Agnostic Meta-Learning using Variational Inference,” in IEEE Winter Conference on Applications of Computer Vision (WACV), 2020, pp. 3079–3089.
[20] K. Posch, J. Steinbrener, and J. Pilz, “Variational inference to measure model uncertainty in deep neural networks,” arXiv preprint arXiv:1902.10189, 2019.
[21] E. Grant, C. Finn, S. Levine, T. Darrell, and T. Griffiths, “Recasting Gradient-Based Meta-Learning as Hierarchical Bayes,” ArXiv, vol. abs/1801.08930, 2018.
[22] C. Finn, K. Xu, and S. Levine, “Probabilistic model-agnostic meta-learning,” arXiv preprint arXiv:1806.02817, 2018.
[23] K. Shridhar, F. Laumann, and M. Liwicki, “Uncertainty estimations by softplus normalization in Bayesian convolutional neural networks with variational inference,” arXiv preprint arXiv:1806.05978, 2018.
[24] S. Ravi and A. A. Bartoson, “Amortized Bayesian Meta-Learning,” in ICLR (Poster), 2019.
[25] J. Yoon, T. Kim, O. Dia, S. Kim, Y. Bengio, and S. Ahn, “Bayesian Model-Agnostic Meta-Learning,” in NeurIPS, 2018, pp. 7343–7353.
[26] S. T. Jose, S. Park, and O. Simeone, “Information-Theoretic Analysis of Epistemic Uncertainty in Bayesian Meta-learning,” arXiv preprint arXiv:2106.00252, 2021.
[27] R. Amit and R. Meir, “Meta-Learning by Adjusting Priors Based on Extended PAC-Bayes Theory,” 2019.
[28] J. Rothfuss, V. Fortuin, M. Josifoski, and A. Krause, “Pach: Bayes-optimal meta-learning with pac-guarantees,” in International Conference on Machine Learning. PMLR, 2021, pp. 9116–9126.
[29] 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 International Conference on Artificial Intelligence and Statistics. PMLR, 2021, pp. 1414–1422.
[30] 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.
[31] I. Goodfellow, Y. Bengio, and A. Courville, Deep learning. MIT Press, 2016.
[32] D. Koller and N. Friedman, Probabilistic graphical models: principles and techniques. MIT Press, 2009.
[33] 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.
[34] D. P. Kingma and M. Welling, “Auto-encoding variational bays,” arXiv preprint arXiv:1312.6114, 2013.
[35] 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.
[36] 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.
[37] D. Tandur and M. Moonen, “Joint adaptive compensation of transmitter and receiver i/q imbalance under carrier frequency offset in ofdm-based systems,” IEEE Transactions on Signal Processing, vol. 55, no. 11, pp. 5246–5252, 2007.