Few-Shot Calibration of Set Predictors via Meta-Learned Cross-Validation-Based Conformal Prediction

Sangwoo Park, Member, IEEE, Kfir M. Cohen, Student Member, IEEE, and Osvaldo Simeone, Fellow, IEEE

Abstract—Conventional frequentist learning is known to yield poorly calibrated models that fail to reliably quantify the uncertainty of their decisions. Bayesian learning can improve calibration, but formal guarantees apply only under restrictive assumptions about correct model specification. Conformal prediction (CP) offers a general framework for the design of set predictors with calibration guarantees that hold regardless of the underlying data generation mechanism. However, when training data are limited, CP tends to produce large, and hence uninformative, predicted sets. This paper introduces a novel meta-learning solution that aims at reducing the set prediction size. Unlike prior work, the proposed meta-learning scheme, referred to as meta-XB, i) builds on cross-validation-based CP, rather than the less efficient validation-based CP; and ii) preserves formal per-task calibration guarantees, rather than less stringent task-marginal guarantees. Finally, meta-XB is extended to adaptive non-conformal scores, which are shown empirically to further enhance marginal per-input calibration.

Index Terms—Conformal prediction, cross-validation, distribution-free calibration, meta-learning, set prediction.

I. INTRODUCTION

A. Context and Motivation

In modern application of artificial intelligence (AI), calibration is often deemed as important as the standard criterion of (average) accuracy [1]. A well-calibrated model is one that can reliably quantify the uncertainty of its decisions [2], [3]. Information about uncertainty is critical when access to data is limited and AI decisions are to be acted on by human operators, machines, or other algorithms. Recent work on calibration for AI machines, or other algorithms. Recent work on calibration for AI

has focused on Bayesian learning, or related ensembling methods, as means to quantify epistemic uncertainty [4], [5], [6], [7]. However, recent studies have shown the limitations of Bayesian learning when the assumed model likelihood or prior distribution are misspecified [8]. Furthermore, exact Bayesian learning is computationally infeasible, calling for approximations such as Monte Carlo (MC) sampling [9] and variational inference (VI) [10]. Overall, under practical conditions, Bayesian learning does not provide formal guarantees of calibration.

Conformal Prediction (CP) [11] provides a general framework for the calibration of (frequentist or Bayesian) probabilistic models. The formal calibration guarantees provided by CP hold irrespective of the (unknown) data distribution, as long as the available data samples and the test samples are exchangeable — a weaker requirement than the standard i.i.d. assumption. As illustrated in Fig. 2, CP produces set predictors that output a subset of the output space \( \mathcal{Y} \) for each input \( x \), with the property

![Fig. 1. Conformal prediction (CP)-based set predictors in the presence of limited data samples: Validation-based CP (VB-CP) [11] and the more efficient cross-validation-based CP (XB-CP) [12] provide set predictors that satisfy the per-task validity condition (1); while previous works on meta-learning for VB-CP [13], [14], which aims at improving efficiency, do not offer validity guarantees when conditioning on a given task \( \tau \). In contrast, the proposed meta-XB algorithm outputs efficient set predictors with guaranteed per-task validity. The last column illustrates efficiency and per-task validity for a simple example with possible outputs \( y \) given by black dots, where the ground-truth outputs are given by the colored crosses and the corresponding set predictions by circles. Per-task validity (see (1)) holds if the set prediction includes the ground-truth output with high probability for each task \( \tau \), while inefficiency (see (2)) measures the average size of the prediction set.](image-url)
that the set contains the true output value with probability no smaller than a desired value $1 - \alpha$ for $\alpha \in [0, 1]$.

Mathematically, for a given learning task $\tau$, assume that we are given a data set $D_\tau$ with $N_\tau$ samples, i.e., $D_\tau = \{z_\tau[i]\}_{i=1}^{N_\tau}$, where the $i$th sample $z_\tau[i] = (x_\tau[i], y_\tau[i])$ contains input $x_\tau[i] \in \mathcal{X}_\tau$ and target $y_\tau[i] \in \mathcal{Y}_\tau$. CP provides a set predictor $\Gamma(\cdot|D_\tau, \xi) : \mathcal{X}_\tau \rightarrow 2^{\mathcal{Y}_\tau}$, specified by a hyperparameter vector $\xi$, that maps an input $x_\tau \in \mathcal{X}_\tau$ to a subset of the output domain $\mathcal{Y}_\tau$ based on a data set $D_\tau$. Calibration amounts to the per-task validity condition

$$\mathbb{P}(y_\tau \in \Gamma(x_\tau|D_\tau, \xi)) \geq 1 - \alpha,$$

(1)

which indicates that the set predictor $\Gamma(x_\tau|D_\tau, \xi)$ contains the true target $y_\tau$ with probability at least $1 - \alpha$. In (1), the probability $\mathbb{P}(\cdot)$ is taken over the ground-truth, exchangeable, joint distribution $p(D_\tau, z_\tau)$, where $z_\tau = (x_\tau, y_\tau)$ and bold letters represent random variables.

The most common form of CP, referred to as validation-based CP (VB-CP), splits the data set into training and validation subsets [11]. The validation subset is used to calibrate the set predictor $\Gamma^\text{VB}(x_\tau|D_\tau, \xi)$ on a test example $x_\tau$ for a given desired miscoverage level $\alpha$ in (1). The drawback of this approach is that validation data is not used for training, resulting in inefficient set predictors $\Gamma^\text{VB}(x_\tau|D_\tau, \xi)$ in the presence of a limited number $N_\tau$ of data samples. The average size of a set predictor $\Gamma(x_\tau|D_\tau, \xi)$, referred to as inefficiency, is defined as

$$\mathcal{L}_\tau(\xi) = \mathbb{E}[|\Gamma(x_\tau|D_\tau, \xi)|],$$

(2)

where the average is taken with respect to the ground-truth joint distribution $p(D_\tau, z_\tau)$.

A more efficient CP set predictor was introduced by [12] based on cross-validation. The cross-validation-based CP (XB-CP) set predictor $\Gamma^{K\text{-XB}}_\alpha(x_\tau|D_\tau, \xi)$ splits the data set $D_\tau$ into $K$ folds to effectively use the available data for both training and calibration. XB-CP can also satisfy the per-task validity condition (1).

An orthogonal approach for the reduction of the inefficiency of CP designs more effective score functions to evaluate the potential of a target value to be included in the predicted set. In particular, adaptive scores were introduced in [17] and were shown via experiments to have improved validity conditions when conditioned on individual inputs in [17, 18, 19, 20] (see Section II-D for further details).

Further improvements in efficiency can be obtained via meta-learning [21]. Meta-learning jointly processes data from multiple learning tasks, say $\tau_1, \ldots, \tau_T$, which are assumed to be drawn i.i.d. from a task distribution $p(\tau)$. These data are used to optimize the hyperparameter $\xi$ of the set predictor $\Gamma(x_\tau|D_\tau, \xi)$ to be used on a new task $\tau \sim p(\tau)$. Specifically, [13] introduced a meta-learning-based method that modifies VB-CP.

In particular, reference [13] leverages data from multiple tasks in order to optimize a single few-shot model, as well as a quantile predictor. The single model is used to score candidate input-output pairs, while the quantile predictor is used to obtain a threshold on the scores that determines the labels to be included in the predicted set. Given its inherent limitation to the use of a single model, the approach is not applicable to XB-CP, which relies on the use of multiple models trained on different data shards.

Furthermore, as a result of its design, the method introduced in [13], referred to as meta-VB, obtains a validity guarantee that requires averaging the probability in (1) with respect to the task distribution $p(\tau)$. Hence, the resulting meta-VB algorithm satisfies a looser validity condition with respect to the per-task inequality (1).

### B. Main Contributions

In this paper, we introduce a novel meta-learning approach, termed meta-XB, with the aim of reducing the inefficiency (2). We refer here in particular to the jackknife-mm scheme presented in Section II-D of [12].
of XB-CP, while preserving, unlike [13], the per-task validity condition (1) for every task \( \tau \). Rather than meta-training a single scoring model and a quantile function as in [13], meta-XB meta-trains a learning procedure that can be applied simultaneously to multiple data shards as required by cross validation. To this end, meta-XB adopts a novel meta-learning criterion that evaluates the average inefficiency across different tasks via a differentiable functional. A summary of the considered CP schemes can be found in Fig. 1.

Overall, the contribution of this work can be summarized as follows:

- We introduce meta-XB, a meta-learning algorithm for XB-CP with the goal of improving over the efficiency of VB-based strategies. Meta-XB leverages data from multiple tasks, further reducing the average predicted set size produced by XB-CP on a new task.
- We prove that meta-XB satisfies the per-task validity condition (1). This is in contrast to the existing meta-VC algorithm [13], which only provides validity guarantees on average over the task distribution.
- Through experimental results, including demodulation [22], image classification [23], and modulation classification [16], we demonstrate that meta-XB can reduce the average prediction set size (2) as compared to VB-CP and XB-CP, while satisfying per-task validity constraints.
- We incorporate adaptive NC scores [17] in the design of meta-XB, demonstrating via experiments that adaptive NC scores can enhance validity requirements conditioned on the input.

The remainder of the paper is organized as follows. Section II summarizes necessary background material on CP, including VB-CP and XB-CP. The meta-XB algorithm is proposed in Section III, followed by Section IV that describes related works. Experimental setting and results are described in Section V. Finally, Section VI concludes the paper.

II. DEFINITIONS AND PRELIMINARIES

In this section, we describe necessary background material on CP [11], [24], VB-CP [11], XB-CP [12], and adaptive NC scores [17].

A. Nonconformity (NC) Scores

At a high level, given an input \( x_\tau \) for some learning task \( \tau \), CP outputs a prediction set \( \Gamma(x_\tau | D_\tau, \xi) \) that includes all outputs \( y \in \mathcal{Y}_\tau \); such that the pair \((x_\tau, y)\) conforms well with the examples in the available data set \( D_\tau = \{z_\tau[i] = (x_\tau[i], y_\tau[i])\}_{i=1}^{N_\tau} \). We recall from Section I that \( \xi \) represents a vector of hyperparameters.

The key underlying assumption is that data set \( D_\tau \) and test pair \( z_\tau = (x_\tau, y_\tau) \) are realizations of exchangeable random variables \( D_\tau \) and \( z_\tau \).

Assumption 1: For any learning task \( \tau \), data set \( D_\tau \) and a test data point \( x_\tau \) are exchangeable random variables, i.e., the joint distribution \( p(D_\tau, z_\tau) = p(z_\tau[1], \ldots, z_\tau[N_\tau], z_\tau) \) is invariant to any permutation of the variables \( \{z_\tau[1], \ldots, z_\tau[N_\tau], z_\tau\} \). Mathematically, we have the equality \( p(z_\tau[1], \ldots, z_\tau[N_\tau + 1]) = p(z_\tau[\pi(1)], \ldots, z_\tau[\pi(N_\tau + 1)]) \) with \( \pi = \pi(N_\tau + 1) \), for any permutation operator \( \pi(\cdot) \).

Note that the standard assumption of i.i.d. random variables satisfies exchangeability.

CP measures conformity via NC scores, which are generally functions of the hyperparameter vector \( \xi \), and are defined as follows.

Definition 1. (NC score): For a given learning task \( \tau \), given a data set \( D_\tau = \{z_\tau[i] = (x_\tau[i], y_\tau[i])\}_{i=1}^{N_\tau} \subseteq D_\tau \) with \( N_\tau \leq N_\tau \) samples, a nonconformity (NC) score is a function \( NC(z|D_\tau, \xi) \) that maps the data set \( D_\tau \) and any input-output pair \( z = (x, y) \) with \( x \in \mathcal{X}_\tau \) and \( y \in \mathcal{Y}_\tau \) to a real number while satisfying the permutation-invariance property \( NC(z|\{\tilde{z}_\tau[1], \ldots, \tilde{z}_\tau[N]\}, \xi) \) for any permutation operator \( \pi(\cdot) \).

A good NC score should express how poorly the point \((x_\tau, y)\) “conforms” to the data set \( D_\tau \). The most common way to obtain an NC score is via a parametric two-step approach. This involves a training algorithm defined by a conditional distribution \( p(\phi|D, \xi) \), which describes the output \( \phi \) of the algorithm as a function of training data set \( D \subseteq D_\tau \) and hyperparameter vector \( \xi \). This distribution may describe the output of a stochastic optimization algorithm, such as stochastic gradient descent (SGD), for frequentist learning, or of a Monte Carlo method for Bayesian learning [25], [26], [27]. The hyperparameter vector \( \xi \) may determine, e.g., learning rate schedule or initialization.

Definition 2. (Conventional two-step NC score): For a learning task \( \tau \), let \( \ell_\tau(\cdot|\phi) \) represent the loss of a machine learning model parametrized by vector \( \phi \) on an input-output pair \( z = (x, y) \) with \( x \in \mathcal{X}_\tau \) and \( y \in \mathcal{Y}_\tau \). Given a training algorithm \( p(\phi|D, \xi) \) that is invariant to permutation of the training set \( D_\tau \), a conventional two-step NC score for input-output pair \( z \) given data set \( D_\tau \) is defined as

\[
NC(z|D_\tau, \xi) := \mathbb{E}_{\phi \sim p(\phi|D, \xi)}\left[ \ell_\tau(z|\phi) \right].
\]

Due to the permutation-invariance of the training algorithm, it can be readily checked that (3) is a valid NC score as per Definition 1.

B. Validation-Based Conformal Prediction (VB-CP)

VB-CP [11] divides the data set \( D_\tau \) into a training data set \( D_\tau^u \) of \( N_\tau^u \) samples and a validation data set \( D_\tau^{val} \) of \( N_\tau^{val} \) samples with \( N_\tau^u + N_\tau^{val} = N_\tau \). It uses the training data set \( D_\tau^u \) to estimate the NC scores \( NC(z|D_\tau^{val}, \xi) \), while the validation data set \( D_\tau^{val} \) is leveraged to construct the set predictor \( \Gamma_{\alpha}^{VB}(x_\tau|D_\tau, \xi) \) as detailed next.

Given an input \( x_\tau \), the prediction set \( \Gamma_{\alpha}^{VB}(x_\tau|D_\tau, \xi) \) of VB-CP includes all output values \( y \in \mathcal{Y}_\tau \) whose NC score \( NC(z|\pi^{val}(1), \xi) \) is smaller than (or equal to) a fraction (at least) \( \alpha \) of \( N_\tau^{val} + 1 \) of the NC scores \( \{NC(z_\tau[i]|D_\tau^{val}, \xi)\}_{i=1}^{N_\tau^{val}} \) for validation data points \( z_\tau[i] \in D_\tau^{val} \).

Definition 3: The \((1-\alpha)\)-empirical quantile \( Q_{1-\alpha}(M) \) of \( M \) real numbers \( a[1], \ldots, a[M] \), with \( a[i] \in \mathbb{R} \), is defined as the \( \left(1-\alpha\right)(M+1)\)th smallest value in the set \( \{a[1], \ldots, a[M], \infty\} \).

With this definition, the set predictor for VB-CP can be thus expressed as...
\[
\Gamma_{\alpha}^{\text{VB}}(x_r | D_r, \xi) = \left\{ y \in \mathcal{Y}_r : \text{NC}(z | D^p_r, \xi) \leq Q_{1-\alpha} \left( \{ \text{NC}(z_r[i] | D^p_r, \xi) \}_{i=1}^{N_r} \right) \right\}
\]

Intuitively, by the exchangeability condition, the empirical ordering condition among the NC scores used to define set (4) ensures the validity condition (1) [11].

Theorem 1: [11] Under Assumption 1, for any miscoverage level \( \alpha \in \{1/(N_r^{(1)} + 1), 1\} \), given any NC score as per Definition 1, the VB-CP set predictor (4) satisfies the validity condition (1).

C. Cross-Validation-Based Conformal Prediction (XB-CP)

In VB-CP, the validation data set is only used to compute the empirical quantile in (4), and is hence not leveraged by the training algorithm \( p(\phi | D_r, \xi) \). This generally causes the inefficiency (2) of VB-CP to be large if number of data points, \( N_r \), is small. XB-CP addresses this problem via \( K \)-fold cross-validation [12].

\[ K \] -fold cross-validation partitions the per-task data set \( D_r = \{ z_r[i] \}_{i=1}^{N_r} \) into \( K \geq 2 \) disjoint subsets \( D_{r,1}, \ldots, D_{r,K} \) such that the condition \( \bigcup_{k=1}^{K} D_{r,k} = D_r \) is satisfied. We define the leave-one-out data set \( D_{r,k} = \bigcup_{k'=1, k' \neq k}^{K} D_{r,k'} \) that excludes the subset \( D_{r,k} \). We also introduce a mapping function \( k : \{1, \ldots, N_r \} \rightarrow \{1, \ldots, K \} \) to identify the subset \( D_{r,k(i)} \) that includes the sample \( z_r[i] \), i.e., \( z_r[i] \in D_{r,k(i)} \).

We focus here on a variant of XB-CP that is referred to as min-max jackknife+ in [12]. This variant has stronger validity guarantees than the jackknife+ scheme also studied in [12]. Accordingly, given a test input \( x_r \), XB-CP computes the NC score for a candidate pair \( z = (x_r, y) \) with \( y \in \mathcal{Y}_r \) by taking the minimum NC score \( \text{NC}(z | D_{r,k}, \xi) \) over all possible subsets \( k \in \{1, \ldots, K\} \), i.e., as \( \min_{k \in \{1, \ldots, K\}} \text{NC}(z | D_{r,k}, \xi) \). Furthermore, for each data point \( z_r[i] \in D_r \), the NC score is evaluated by excluding the subset \( D_{r,k(i)} \) from \( \text{NC}(z_r[i] | D_{r,k(i)}, \xi) \). Note that evaluating the resulting \( N_r + 1 \) NC scores requires running the training algorithm \( p(\phi | D_{r,k}, \xi) \) \( K \) times, once for each subset \( D_{r,k} \). Finally, a candidate \( y \) is included in the prediction set if the NC score for \( z = (x_r, y) \) is smaller (or equal) than for a fraction (at least) \( \alpha'(N_r + 1)/N_r \) of the validation data points with \( \alpha' = \alpha - 1/K \).

Overall, given data set \( D_r = \{ z_r[i] = (x_r[i], y_r[i]) \}_{i=1}^{N_r} \) and test input \( x_r \in \mathcal{X}_r \), \( K \)-fold XB-CP produces the set predictor

\[
\Gamma_{\alpha}^{\text{K-XB}}(x_r | D_r, \xi) = \left\{ y \in \mathcal{Y}_r : \sum_{i=1}^{N_r} \min_{k \in \{1, \ldots, K\}} \text{NC}(z_r[i] | D_{r,k(i)}, \xi) \leq \text{NC}(z_r[i] | D_{r,k(i)}, \xi) \geq \alpha'(N_r + 1) \right\}
\]

where \( 1(\cdot) \) is the indicator function (1(true) = 1 and 1(false) = 0).

Theorem 2: [12] Under Assumption 1, for any miscoverage level \( \alpha \in \left[ \frac{1}{N_r + 1}, \frac{1-K/N_r}{K+1} \right] \), given any NC score as per Definition 1, the XB-CP set predictor (5) satisfies the validity condition (1).

While a proof of Theorem 2 for \( K = N_r \) can be found in [12], the general case for \( K < N_r \) follows from the same proof techniques in [12] and is included for completeness in Appendix A.1, available online.

D. Adaptive Parametric NC Score

The CP methods reviewed so far achieve the per-task validity condition (1). In contrast, per-input, per-task conditional validity condition, defined as

\[
\mathbb{P}(y_r \in \Gamma(x_r, D_r, \xi) | x_r = x_r \geq 1 - \alpha.
\]

is only attainable with strong additional assumptions on the joint distribution \( p(D_r, x_r) \) [28, 29]. However, as discussed in Section I, the adaptive NC score introduced by [17] is known to empirically improve the per-input conditional validity of VB-CP (4) and XB-CP (5). In this section, we review for reference the definition of the adaptive NC score.

In this subsection, we assume that a model class of probabilistic predictors \( p(y|x, \phi) \) is available, e.g., a neural network with a softmax activation in the last layer. To gain insight on the definition of adaptive NC scores, let us assume for the sake of argument that the ground-truth conditional distribution \( p(y_r | x_r) \) is known. The most efficient (deterministic) set predictor satisfying the conditional coverage condition (6) would then be obtained as the smallest-cardinality subset of target values in \( \mathcal{Y}_r \) that satisfies the conditional coverage condition (6), i.e.,

\[
\Gamma_{\alpha}^*(x_r) = \arg \min_{\Gamma \subseteq \mathcal{Y}_r} |\Gamma| \text{ s.t. } \sum_{y \in \Gamma} p(y | x_r) \geq 1 - \alpha.
\]

Note that set (7) can be obtained by adding values \( y \in \mathcal{Y}_r \) to set predictor \( \Gamma_{\alpha}^*(x_r) \) in order from largest to smallest value of \( p(y_r | x_r) \) until the constraint in (7) is satisfied.

In practice, the conditional distribution \( p(y_r | x_r) \) is estimated via the model \( p(y_r | x_r, \phi) \) where the parameter vector \( \phi \) is produced by a training algorithm \( p(\phi | D_r, \xi) \) applied to some training data set \( D_r \). This yields the naive set predictor

\[
\Gamma_{\alpha, \text{naive}}^*(x_r | D_r, \xi) = \arg \min_{\Gamma \subseteq \mathcal{Y}_r} |\Gamma| \text{ s.t. } \sum_{y \in \Gamma} \mathbb{E}_{\phi \sim p(\phi | D_r, \xi)} p(y | x_r, \phi) \geq 1 - \alpha_{\text{naive}},
\]

where we have used for generality the ensemble predictor obtained by averaging over the output \( \phi \sim p(\phi | D_r, \xi) \) of the training algorithm. Unless the likelihood model is perfectly calibrated, i.e., unless the equality \( p(y_r | x_r) = \mathbb{E}_{\phi \sim p(\phi | D_r, \xi)} p(y_r | x_r, \phi) \) holds, there is no guarantee that the set predictor in (8) satisfies the conditional coverage condition (6) or the marginal coverage condition (1) with \( \alpha = \alpha_{\text{naive}} \).
To tackle this problem, [17] proposed to apply VB-CP or XB-CP with a modified set score inspired by the naïve prediction

\[ \text{Definition 4. (Adaptive NC score): For a learning task } \tau, \text{ given a training algorithm } \mathcal{p}(\phi|D_\tau, \xi) \text{ that is invariant to permutation in } \tilde{X}, \text{ the adaptive NC score for input-output pair } z = (x, y) \text{ with } x \in \mathcal{X}, \text{ and } y \in \mathcal{Y}, \text{ given data set } D_\tau, \text{ is defined as} \]

\[
\text{NC}_{\text{adv}}(z|\tilde{D}_\tau, \xi) = 1 - \max_{\alpha_{\text{naive}} \in [0, 1]} \alpha_{\text{naive}} ^{\text{naive}} \\
\text{s.t. } y \in \Gamma^{\text{naive}}(x|\tilde{D}_\tau, \xi). \tag{9}
\]

Intuitively, if the adaptive NC score is large, the pair \( z \) does not conform well with the probabilistic model \( \mathbb{E}_{\phi \sim \mathcal{p}(\phi|D_\tau, \xi)} \mathcal{p}(y|x, \phi) \) obtained by training on set \( D_\tau \). The adaptive NC score satisfies the condition in Definition 1, and hence by Theorems 1 and 2, the set predictors (4) and (5) for VB-CP and XB-CP, respectively, are both valid when the adaptive NC score is used. Furthermore, [17] demonstrated improved conditional empirical coverage performance as compared to the conventional two-step NC score in Definition 2. This may be seen as a consequence of the conditional validity of the naïve predictor (8) under the assumption of a well-calibrated model.

The adaptive NC score (9) can be equivalently expressed as

\[
\text{NC}_{\text{adv}}(z|\tilde{D}_\tau, \xi) = \sum_{y \in \mathcal{Y}} \mathbb{1}(p(y|x, \tilde{D}_\tau, \xi) > p(y|x, \tilde{D}_\tau, \xi)) \mathcal{p}(y|x, \tilde{D}_\tau, \xi), \tag{10}
\]

where we have used the notation \( p(y|x, \tilde{D}_\tau, \xi) := \mathbb{E}_{\phi \sim \mathcal{p}(\phi|D_\tau, \xi)} \mathcal{p}(y|x, \phi) \).

III. META-LEARNING ALGORITHM FOR XB-CP (META-XB)

In this section, we introduce the proposed meta-XB algorithm. We start by describing the meta-learning framework.

A. Meta-Learning

Up to now, we have focused on a single task \( \tau \). Meta-learning utilizes data from multiple tasks to enhance the efficiency of the learning procedure for new tasks. Following the standard meta-learning formulation [30, 31], as anticipated in Section 1, the learning environment is characterized by a task distribution \( \mathcal{p}(\tau) \) over the task identifier \( \tau \). Given \( T \) meta-training tasks realizations \( \tau_1 = \tau_1, \ldots, \tau_T = \tau_T \) drawn i.i.d. from the task distribution \( \mathcal{p}(\tau) \), the meta-training data set \( D_{\tau_1:T} := \{ (D_{\tau_1}^{i}, z_{\tau_1}^{i}) \}_{i=1}^{M_1} \) consists of \( M_1 \) realizations \( \{D_{\tau_1}^{i}, z_{\tau_1}^{i}\}_{i=1}^{M_1} \) of data sets \( D_{\tau_1} = D_\tau \) with \( N_{\tau_1} = N_1 \) examples and test sample \( z_{\tau_1}^{i} = z_{\tau_1} \) for each task \( \tau_1 \). Pairs \( \{D_{\tau_1}^{i}, z_{\tau_1}^{i}\}_{i=1}^{M_1} \) are generated i.i.d. from the joint distribution \( \mathcal{p}(\mathcal{D}_{\tau_1}, z_{\tau_1}) \), satisfying Assumption 1 for all tasks \( t \).

The goal of meta-learning for CP is to optimize the vector of hyperparameter \( \xi \) based on the meta-training data \( D_{\tau_1:T} \), so as to obtain a more efficient set predictor \( \Gamma(x_{\tau}|D_t, \xi) \). While reference [13] proposed a meta-learning solution for VB-CP [11], here we introduce a meta-learning method for XB-CP.

B. Meta-XB

This section first describes how to optimize the hyperparameter \( \xi \) based on data set \( D_{\tau_1:T} \) obtained from \( T \) meta-training tasks. This procedure takes place during an offline meta-training phase. Then, upon deployment of the system after meta-training, the system enters the meta-testing phase of operation. In it, a new task is presented for which a set predictor must be produced.

1) Meta-Training: Meta-XB aims at finding a hyperparameter vector \( \xi \) that minimizes the average size of the prediction set \( \Gamma^K_{\text{XB}}(x_{\tau}|D_t, \xi) \) in (5) for tasks \( \tau \) that follow the distribution \( \mathcal{p}(\tau) \). To this end, it addresses the problem of minimizing the empirical average of the sizes of the prediction sets \( \Gamma^K_{\text{XB}}(x_{\tau}|D_{\tau_1:T}, \xi) \) across the meta-training tasks \( \tau_1, \ldots, \tau_T \) over the hyperparameter vector \( \xi \). This amounts to the optimization

\[
\xi^* = \arg \min_{\xi} \frac{1}{T} \sum_{t=1}^{T} \frac{1}{M_t} \sum_{j=1}^{M_t} \left| \Gamma^K_{\text{XB}}(x_{\tau}^{i}|D_t^{i}, \xi) \right|, \tag{11}
\]

where the first sum is over the meta-training tasks and the second is over the available data for each task. By (5), the size of the prediction set \( \left| \Gamma^K_{\text{XB}}(x_{\tau}|D_t, \xi) \right| \) is not a differentiable function of the hyperparameter vector \( \xi \). Therefore, in order to address (11) via gradient descent, we introduce a differentiable soft inefficiency criterion by replacing the indicator function with the sigmoid \( \sigma(u) = (1 + \exp(-u/c_{S}))^{-1} \) for some \( c_{S} > 0 \); the quantile \( \tilde{Q}_{1-\alpha} \) with a differentiable soft empirical quantile \( \tilde{Q}_{1-\alpha} \); and the minimum operator with the softmax function [32].

For an input set \( \{a[j]\}_{j=1}^{M} \), the softmax function is defined as [33]

\[
\text{softmax} (\{a[j]\}_{j=1}^{M}) = \sum_{j=1}^{M} a[j] \exp(-a[j]/c_{S}) \sum_{i=1}^{M} \exp(-a[i]/c_{S}), \tag{12}
\]

for some \( c_{S} > 0 \). Finally, given an input set \( \{a[1], \ldots, a[M]\} \), the soft empirical quantile \( \tilde{Q}_{1-\alpha} \) is defined as

\[
\tilde{Q}_{1-\alpha} (\{a[j]\}_{j=1}^{M}) = \sum_{j=1}^{M+1} a[j] \exp(-\rho_1-\alpha(a[j])\{a[j]\}_{j=1}^{M+1}/c_{Q}) \sum_{i=1}^{M+1} \exp(-\rho_1-\alpha(a[i])\{a[i]\}_{j=1}^{M+1}/c_{Q}), \tag{13}
\]

for some \( c_{Q} > 0 \) and \( c_{M+1} = \max(\{a[j]\}_{j=1}^{M}) + \delta \) for some \( \delta > 0 \), where we have used the pinball loss \( \rho_1-\alpha(a[1], \ldots, a[M]) \) [34]

\[
\rho_1-\alpha(a[j])\{a[j]\}_{j=1}^{M} = \alpha \sum_{j=1}^{M} \text{ReLU}(a-a[j]) + (1 - \alpha) \sum_{j=1}^{M} \text{ReLU}(a[j] - a). \tag{14}
\]
with ReLU\((a) = \max(0, a)\). With these definitions, the soft inefficiency metric is derived from (5) as follows (see details in Appendix A.2, available online).

Definition 5: Given a data set \(\mathcal{D}_\tau\) and a test input \(x_\tau\), the soft inefficiency for the \(K\)-fold XB-CP predictor (5) is defined as

\[
\hat{\Gamma}_{\alpha, K}^{XB}(x_\tau | \mathcal{D}_\tau, \xi) = \sum_{y \in \mathcal{Y}} \{Q_{1-\delta - \alpha'} (\{NC((z_\tau, y) | \mathcal{D}_\tau, -k(i)) \})_{K} K_{i=1}^{K/N} \}^{N_{z}} - \text{softmax} \{\{NC((x_\tau, y) | \mathcal{D}_\tau, -k(i)) \})_{K} K_{i=1}^{K/N} \}^{N_{z}} \}
\]

(15)

where \(\alpha' = \alpha - \frac{1-K/N}{K+1}\) and \(c_\sigma, c_\xi, c_Q > 0\).

The parameters \(c_\sigma, c_\xi, c_Q\) dictate the trade-off between smoothness and accuracy of the approximation \(\hat{\Gamma}_{\alpha, K}^{XB}(x_\tau | \mathcal{D}_\tau, \xi)\) with respect to the true inefficiency \(\Gamma_{\alpha, K}^{XB}(x_\tau | \mathcal{D}_\tau, \xi)\). As \(c_\sigma, c_\xi, c_Q \to 0\), the approximation becomes increasingly accurate for any \(\delta > 0\), as long as we have \(\alpha \in \left[\frac{1}{N_{\tau}+1} + \frac{1-K/N}{K+1}, 1\right]\), but the function \(\hat{\Gamma}_{\alpha, K}^{XB}(x_\tau | \mathcal{D}_\tau, \xi)\) is increasingly less smooth (see Fig. 3 for an illustration of the accuracy of the soft quantile).

Replacing the soft inefficiency (15) in problem (11) yields a differentiable program when conventional two-step NC scores (Definition 2) are used. We address the corresponding problem via SGD-based optimizers, such as Adam [35], whereby at each iteration a batch of tasks and examples per task are sampled. The overall meta-learning procedure is summarized in Algorithm 1.

2) Meta-Testing: After meta-training (Algorithm 1), during the meta-testing phase, the optimized hyperparameter \(\xi^{*}\) is fixed. Accordingly, for any new task \(\tau\), meta-XB constructs set predictor using the XB-CP set predictor (7) by using the optimized hyperparameter \(\xi^{*}\) in lieu of \(\xi\).

C. Meta-XB With Adaptive NC Scores

Adaptive NC scores are not differentiable. Therefore, in order to enable the optimization of problem (11) with the soft inefficiency (15), we propose to replace the indicator function \(1(\cdot)\) in (10) with the sigmoid function \(\sigma(\cdot)\). We also have found that approximating the number of outputs \(y' \in \mathcal{Y}_\tau\) that satisfy (10) rather than direct application of sigmoid function empirically improves per-input coverage performance. This yields the soft adaptive NC score \(NC^{\text{ada}}(z|\mathcal{D}_\tau, \xi)\), which is detailed in Appendix B, available online. With the soft adaptive NC score, meta-XB is then applied as in Algorithm 1.

D. Per-Task Validity of Meta-XB

As mentioned in Section I, existing meta-learning schemes for CP cannot achieve the per-task validity condition in (1), requiring an additional marginalization over distribution \(p(\tau)\) [13] or achieving looser validity guarantees formulated as probably approximately correct (PAC)-bounds [14]. In contrast, meta-XB has the following property.

Theorem 3: Under Assumption 1, for any miscoverage level \(\alpha \in \left[\frac{1}{N_{\tau}+1} + \frac{1-K/N}{K+1}, 1\right]\), given any NC score (Definition 1), the XB-CP set predictor (5) with \(\xi = \xi^{*}\) in (11) satisfies the validity condition (1).

Theorem 3 is a direct consequence of Theorem 2, since meta-XB maintains the permutation-invariance of the training algorithm \(p(\phi|\mathcal{D}_\tau, \xi^{*})\) as required by Definition 2.

E. Application of the Proposed Meta-Learning Framework to VB-CP

We conclude this section by elaborating on the application of the proposed meta-learning framework to VB-CP. As explained in Section I, while there exist meta-learning approaches for VB-CP [13, 14], these techniques do not guarantee per-task validity.
(1). In contrast, as we briefly discuss next, the proposed method achieves this goal.

By rewriting the objective function of meta-XB (11) with the VB-CP set predictor \( \Gamma^\text{VB}_\alpha(x_t^j | D^j_t, \xi) \) in lieu of the XB-CP set predictor \( \Gamma^\text{XB}_\alpha(x_t^j | D^j_t, \xi) \), a meta-learning objective for VB-CP can be defined as

\[
\xi^* = \arg \min_\xi \frac{1}{T} \sum_{t=1}^{T} \frac{1}{M_t} \sum_{j=1}^{M_t} |\hat{\Gamma}^\text{VB}_\alpha(x_t^j | D^j_t, \xi)|,
\]

in which we have denoted as \( |\hat{\Gamma}^\text{VB}_\alpha(x_t^j | D^j_t, \xi)| \) the soft inefficiency for VB-CP. Following the same smoothing approximations in Section III-B, this is defined as

\[
|\hat{\Gamma}^\text{VB}_\alpha(x_t^j | D^j_t, \xi)| = \sum_{y \in \mathcal{Y}} \sigma \left( \hat{Q}_{1-\alpha} \left( \{ \text{NC}(z_\tau[i] | D^\tau, \xi) \}_{i=1}^{\mathcal{N}^j} \right) \right).
\]

Addressing this problem as explained in Section III-B during the meta-training phase, the obtained meta-trained hyperparameter vector \( \xi^* \) is then used during meta-testing for a new task \( \tau \).

The resulting algorithm provably guarantees per-task validity condition (1). More specifically, by the same arguments used to prove Theorem 3, the statement of the theorem applies verbatim to this scheme, which we will refer to as soft meta-VB.

IV. RELATED WORK

Bayesian Learning and Model Misspecification: When the model is misspecified, i.e., when the assumed model likelihood or prior distribution cannot express the ground-truth data generating distribution [8], Bayesian learning may yield poor generalization performance [8], [36], [37]. Downweighting the prior distribution and/or the likelihood, as done in generalized Bayesian learning [27], [38] or in “cold” posterior [37], improve the generalization performance. In order to mitigate the model likelihood misspecification, alternative variational free energy metrics were introduced by [8] via second-order PAC-Bayes bounds, and by [36] via multi-sample PAC-Bayes bounds. Misspecification of the prior distribution can be also addressed via Bayesian meta-learning, which optimizes the prior from data in a manner similar to empirical Bayes [39].

Bayesian Meta-Learning: While frequentist meta-learning has shown remarkable success in few-shot learning tasks in terms of accuracy [40], [41], improvements in terms of calibration can be obtained by Bayesian meta-learning that optimizes over a hyper-posterior distribution from multiple tasks [4], [5], [6], [7], [31], [42]. The hyper-prior can also be modelled as a stochastic process to avoid the bias caused by parametric models [43].

CP-Aware Loss: References [44] and [18] proposed CP-aware loss functions to enhance the efficiency or per-input validity (6) of VB-CP for single-task learning problems. The main idea is to adopt the soft inefficiency of the VB-CP set predictor (15) as the objective of the training algorithm \( p(\phi | D^\tau, \xi) \), which is practically approximated by mimicking validation and test data with disjoint subsets \( D^\tau_{\text{val}}, D^\tau_{\text{te}} \) of the training data \( D^\tau \). More precisely, defining the NC score of an input-output pair \( z = (x, y) \) given a fixed model parameter vector \( \phi \) as NC\( (z | \phi) = \ell_\tau(z | \phi) \) (cf. Definition 2), the CP-aware training algorithm aims at finding a parameter vector \( \phi^CA \) that satisfies

\[
\phi^CA = \arg \min_\phi \sum_{(x,y) \in D^\tau_{\text{val}}} |\hat{\Gamma}^\text{VB}_\alpha(x | D^\tau_{\text{val}}, \phi)|,
\]

in which \( |\hat{\Gamma}^\text{VB}_\alpha(x | D^\tau_{\text{val}}, \phi)| \) is the soft inefficiency of the VB-CP set predictor \( \Gamma^\text{VB}_\alpha(x | D^\tau_{\text{val}}, \phi) \) that uses the auxiliary validation data \( D^\tau_{\text{val}} \) in lieu of \( D^\tau \). The CP-aware training algorithm can hence be expressed as \( p(\phi | D^\tau_{\text{val}}, \xi) = \delta(\phi - \phi^CA) \), where \( \delta(\cdot) \) is the Dirac delta function. Multiple splits of training data \( (D^\tau_{\text{val}}, D^\tau_{\text{te}}) \) can further be taken into account when solving (19) [44]. These methods typically require a large number of data samples, unlike the meta-learning methods studied here. Furthermore, the corresponding training algorithm does not preserve the permutation-invariance property and hence it cannot be applied to XB-CP (see Definition 1).

Per-Input Validity and Local Validity: As discussed in Section II-D, the per-input validity condition (6) cannot be satisfied without strong assumptions on the joint distribution \( p(D_{\tau}, z_{\tau}) \) [28], [29]. Given the importance of adapting the prediction set size to the input to capture heteroscedasticity [45], [46], a looser local validity condition, which conditions on a subset of the input data space \( A_{\tau_x} \subset X_{\tau} \) containing the input \( x_{\tau} \) of interest, i.e., \( x_{\tau} \in A_{\tau_x} \), has been considered in [29], [47]. Choosing a proper subset \( A_{\tau_x} \) becomes problematic especially in high-dimensional input space [46], [48], and [49], [50] proposed to reweight the samples outside the subset \( A_{\tau_x} \) by treating the problem as distribution-shift between the data set \( D_{\tau} \) and the test input \( x_{\tau} \). More recently, reference [51] proposed to use input-dependent empirical quantile for VB-CP via a generalized definition of the pinball loss with the goal of enhancing conditional validity; while [52] provided a theoretical lower bound on the local coverage via information-theoretic measures.

V. EXPERIMENTS

In this section, we provide experimental results to validate the performance of meta-XB in terms of i) per-task coverage \( \mathbb{P}(y_{\tau} \in \Gamma(x_{\tau} | D_{\tau}, \xi)) \); ii) per-task inefficiency (2); iii) per-task conditional coverage \( \mathbb{P}(y_{\tau} \in \Gamma(x_{\tau} | D_{\tau}, \xi) | x_{\tau} = x_{\tau}) \); and iv) per-task conditional inefficiency \( \mathbb{E}[|\Gamma(x_{\tau} | D_{\tau}, \xi)| | x_{\tau} = x_{\tau}] \). To evaluate input-conditional quantities, we follow the approach in [17, Section S1.2]. As benchmark schemes, we consider i) VB-CP, ii) XB-CP, and iii) meta-VB [13], with either the conventional NC score (Definition 2 with log-loss \( \ell_\alpha(z | \phi) \) or adaptive NC score with Definition 4). Note that meta-VB was described in [13] only for the conventional NC score, but the application of the adaptive NC score is direct.

For all the experiments, unless specified otherwise, we consider a number of examples \( N_{\tau} = 9 \) for the data set \( D_{\tau} \) and the desired miscoverage level \( \alpha = 0.1 \). For the cross-validation-based set predictors XB-CP and meta-XB, we set number of

\[\text{Code is available at https://github.com/kclip/meta-XB.}\]
folds to $K = N_τ$; while for VB-CP, we adopt equal split for training and validation data, i.e., $N_τ^v = N_τ^t = N_τ/2$ [17]. The aforementioned performance measures are estimated by averaging over 1000 realizations of data set $D_τ$ and over 500 realizations for the test sample $z_τ$ of each task $τ$. We report in this section the 100 different per-task quantities which are computed from 100 different tasks. During meta-training, for $T$ different tasks, we assume availability of $M_t(N_τ + 1)$ i.i.d. examples, from which we sample $M_t$ pairs $\{D_τ^{(i)}, z_τ^{(i)}\}_{i=1}^{M_t}$ when computing inefficiency (16), with which we use Adam optimizer [35] to update the hyperparameter vector $ξ$ via SGD. Lastly, we set the value of the approximation parameters $c_ρ$, $c_S$, and $c_Q$ to be one.

Following [17], for all the schemes, we adopt a neural network classifier [17], and set the training algorithm $p(θ|D_τ, ξ)$ to output the last iterate of a pre-defined number of steps of GD, with initialization given by the hyperparameter vector $ξ$ [40]. The use of full-batch GD ensures the permutation-invariance of the training algorithm as required by Definition 2. As originally suggested by [40] and validated in many follow-up studies [53], [54], [55], [56], we set the number of GD steps to five, unless specified otherwise.

All the experiments are implemented by PyTorch [57] and ran over a GPU server with single NVIDIA A100 card.

### A. Multinomial Model and Inhomogeneous Features

We start with the synthetic-data experiment introduced in [17] in which the input $x \in \mathbb{R}^1$ is such that the first element equals $x_1 = 1$ with probability 1/5 and $x_1 = -8$ otherwise, while the other elements $x_2, \ldots, x_{10}$ are i.i.d. standard Gaussian variables. For each task $τ$, matrix $\tau \in \mathbb{R}^{10 \times |Y_τ|}$ is sampled with i.i.d. standard Gaussian entries and the ground-truth conditional distribution $p(y_τ|x_τ)$ is defined as the categorical distribution

$$p(y_τ = y|x_τ) = \frac{\exp(x_τ^\top τ_y)}{\sum_{y' = 1}^{|Y_τ|} \exp(x_τ^\top τ_{y'})},$$

for $y \in \{1, \ldots, |Y_τ|\}$, where $τ_y \in \mathbb{R}^{|Y_τ|}$ is the $y$th column of the task information matrix $τ$. The number of classes is $|Y_τ| = 5$ and neural network classifier consists of two hidden layers with Exponential Linear Unit (ELU) activation [58] in the hidden layers and a softmax activation in the last layer.

In Fig. 4, we demonstrate the performance of the considered set predictors as a function of number of tasks $T$ under $N_τ = 9$. Both meta-VB and meta-XB achieve lower inefficiency (2) as compared to the conventional set predictors VB-CP and XB-CP, as soon as the number of meta-training tasks is sufficiently large to ensure successful generalization across tasks [59], [60]. For example, meta-XB with $T = 100$ tasks obtain an average prediction set size of 2.85 on average, while XB-CP has an inefficiency of 3.35. Furthermore, all schemes satisfy the validity condition (1), except for meta-VB for $T \approx 10^4$, confirming the analytical results. It can also be seen from Fig. 4 that, irrespective of the choice of the training algorithm – standard training [17] or CP-aware training (19) – VB-CP yields uninformative set predictions, reaching the maximum average set size of 5.
and with CP-aware training examples per task from via a meta-learned quantile predictor, but it does (1) using adaptive NC scores.

φx ≤ for each data set \[17\] = 5\ 0 ∈ Y equals the set of neigh-

\[7\] = 1 grows larger. In particular, when \[40\] is large \[7\] consists of 500 examples per task from \[e\] : \[τ\]

τ for \[z\] (6) and (see also \[Y\] using conventional NC ≥ in the same

when used with meta-XB, although this comes at the cost of a larger inefficiency.

Furthermore, in Fig. 6, we investigate the impact of number of per-task examples \[N\] τ in data set \[D\] τ using conventional NC scores. As shown in Fig. 6, the average size of the set predictors decreases as \[N\] τ grows larger. In particular, when \[N\] τ is large enough, i.e., when \[N\] τ ≥ 40, conventional set predictors become preferable, as in this regime transfer of knowledge across tasks is unnecessary, and possibly deleterious [31] (see also [61] for related discussions). In contrast, in the few-examples regime, i.e., with \[N\] τ ≤ 30, the meta-learned set predictors meta-VB and meta-XB significantly outperform the conventional set predictors VB-CP and XB-CP in terms of inefficiency. With few data points, even the use of CP-aware training yields maximally un-

informative set predictors, highlighting the importance of lever-

aging data from multiple tasks when test-task data are limited.

Lastly, in Fig 7, we demonstrate the performance of the soft meta-VB scheme described in Section III-E in the same setting as in Fig. 6. Being based on VB-CP, this approach is outperformed by meta-XB in terms of inefficiency. Furthermore, like meta-XB, it provably preserves the per-task validity condition. This is unlike meta-VB [13], which can reduce the inefficiency of VB-CP in the extremely few-examples regime (\[N\] τ < 20) via a meta-learned quantile predictor, but it does not guarantee the per-task validity condition (1). Moreover, meta-VB has a larger inefficiency as compared to meta-XB, which leverages every data example for both training and validation purpose via cross-validation.

B. Demodulation

To elaborate further on the last column in Fig. 1, here we present another toy example that allows us to visualize the set predictors obtained by XB-CP and meta-XB, both with conventional and adaptive NC scores. To this end, we implement XB-CP with the same neural network used for meta-XB but with the hyperparameter vector \(\xi\) defining the initialization of GD set to a random vector [62]. Given a learning task \(\tau\), the input and output space \(X_\tau\) and \(Y_\tau\) are given by the set of complex points, i.e., by the two-dimensional real vectors [22]

\[X_\tau = Y_\tau = \left\{ \sqrt{\frac{2z}{M + 1}} \cdot e^{j2\pi(1 - \frac{x}{M + 1})} \cdot e^{j\phi_\tau} \mid z = 1, 2, \ldots, M \right\}, \]

for some task-specific phase shift \(\phi_\tau \in [0, 2\pi]\). Denoting as \(N_\tau(x) = \{ y \in Y_\tau : |x - y| \leq r \and y \neq x \}\) the set of neighboring points within some radius \(r\), the ground-truth distribution \(p(\{y_\tau | x_\tau\})\) is such that \(y_\tau\) equals \(x_\tau\) with probability \(1 - p\), and
it equals any neighboring point \( y \in \mathcal{N}_r(x) \) with probability \( p/|\mathcal{N}_r(x)| \). We set \( M = 6, r = 1.3, \) and \( p = 0.2 \). We design neural network classifier to consist of two hidden layers with Exponential Linear Unit (ELU) activation [58] in the hidden layers and a softmax activation in the last layer.

Fig. 8 visualizes the set predictors for XB-CP, i.e., with a random hyperparameter vector \( \xi \), and for meta-XB, after meta-training with 1000 tasks, by focusing on a specific realizations of phase shift that follows the distribution \( \phi_r \sim \text{Unif}(0, 2\pi) \). By transferring knowledge from multiple tasks, meta-XB is seen to yield more efficient set predictors. Furthermore, by using adaptive NC scores, meta-XB can adjust the prediction set size depending on the “difficulty” of classifying the given input, while a conventional NC score tends to produce set predictors of similar sizes across all inputs. Note, in fact, that inputs close to the center of the set, as the green example in Fig. 8, have more neighbors as compared to points at the edge, as the red points in Fig. 8, making it harder to identify the true value of \( y \) given \( x \).

C. Modulation Classification

We now consider the real-world modulation classification example illustrated in Fig. 2, in which the goal is classifying received radio signals depending on the modulation scheme used to generate it [15], [16]. The RadioML 2018.01 A data set consists 98,304 inputs with dimension \( 2 \times 1024 \), accounting for complex baseband signals sampled over 1024 time instants, generated from 24 different modulation types [16]. Each task \( \tau \) amounts to the binary classification of signals from two randomly selected modulation types. Specifically, we divide the 24 modulations types into 16 classes used to generate meta-training tasks, and 8 classes used to produce meta-testing tasks, following the standard data generation approach in few-shot classifications [63], [64]. We adopt VGG16 [65] as the neural network classifier as in [16]. Furthermore, for meta-VB and meta-XB, we apply a single GD step during meta-training and five GD steps during meta-testing [6], [40].

Fig. 9 shows per-task coverage and inefficiency for all schemes assuming conventional NC scores. While the conventional set predictors VB-CP and XB-CP produce large, uninformative set predictors that encompass the entire target data space \( \mathcal{Y}_\tau \) of dimension \( |\mathcal{Y}_\tau| = 2 \), the meta-learned set predictors meta-VB and meta-XB can significantly improve the prediction efficiency. However, meta-VB fails to achieve per-task validity condition (1), while the proposed meta-XB is valid as proved by Theorem 3.

D. Image Classification

Lastly, we consider image classification problem with the miniImagenet dataset [23] considering \( N_\tau = 4 \) data points per task with desired miscoverage level \( \alpha = 0.2 \). We consider binary
classification with tasks being defined by randomly selecting two classes of images, and drawing training data sets by choosing among all examples belonging to the two chosen classes. Conventional NC scores are used, and the neural network classifier consists of the convolutional neural network (CNN) used in [40]. For meta-VB and meta-XB, a single step GD update is used during meta-training, while five GD update steps are applied during meta-testing. Fig. 10 shows that meta-learning-based set predictors outperform conventional schemes. Furthermore, meta-VB fails to meet per-task coverage in contrast to the proposed meta-XB.

VI. CONCLUSION

This paper has introduced meta-XB, a meta-learning solution for cross-validation-based conformal prediction that aims at reducing the average prediction set size, while formally guaranteeing per-task calibration. The approach is based on the use of soft quantiles, and it integrates adaptive nonconformity scores for improved input-conditional calibration. Through experimental results, including for modulation classification [15], [16], meta-XB was shown to outperform both conventional conformal prediction-based solutions and meta-learning conformal prediction schemes. Future work may integrate meta-learning with CP-aware training criteria [18], [44], or with stochastic set predictors.

REFERENCES

[1] A. Thelen et al., “A comprehensive review of digital twin–part 2: Roles of uncertainty quantification and optimization, a battery digital twin, and perspectives,” 2022, arXiv:2208.12004.
[2] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, “On calibration of modern neural networks,” in Proc. Int. Conf. Mach. Learn., PMLR, 2017, pp. 1321–1330.
[3] J. Hermans, A. Delaunoy, F. Rozet, A. Wehenkel, and G. Louppe, “Averting a crisis in simulation-based inference,” 2021, arXiv:2110.06581.
[4] C. Finn, K. Xu, and S. Levine, “Probabilistic model-agnostic meta-learning,” in Proc. Adv. Neural Inf. Process. Syst., 2018, pp. 9537–9548.
[5] J. Yoon, T. Kim, O. Dia, S. Kim, Y. Bengio, and S. Ahn, “Bayesian model-agnostic meta-learning,” in Proc. Adv. Neural Inf. Process. Syst., 2018, pp. 7343–7353.
[6] S. Ravi and A. Beatson, “Amortized Bayesian meta-learning,” in Proc. Int. Conf. Learn. Representations, 2018.
[7] S. T. Jose, S. Park, and O. Simeone, “Information-theoretic analysis of epistemic uncertainty in Bayesian meta-learning,” in Proc. Int. Conf. Artif. Intell., Statist., PMLR, 2022, pp. 9758–9775.
[8] A. Masegosa, “Learning under model misspecification: Applications to variational and ensemble methods,” Adv. Neural Inf. Process. Syst., vol. 33, pp. 5479–5491, 2020.
[9] C. P. Robert, G. Casella, and G. Casella, Monte Carlo Statistical Methods, vol. 2. Berlin, Germany: Springer, 1999.
[10] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, “Weight uncertainty in neural network,” in Proc. Int. Conf. Mach. Learn., PMLR, 2015, pp. 1613–1622.
[11] R. F. Barber, E. J. Candes, A. Ramdas, and R. J. Tibshirani, “Predictive inference with the jackknife+,” Ann. Statist., vol. 49, no. 1, pp. 486–507, 2021.
[12] A. Fisch, T. Schuster, T. Jaakkola, and R. Barzilay, “Few-shot conformal prediction with auxiliary tasks,” in Proc. Int. Conf. Mach. Learn., PMLR, 2021, pp. 3329–3339.
[13] S. Park, E. Dobriban, I. Lee, and O. Bastani, “PAC prediction sets for meta-learning,” 2022, arXiv:2207.02440.
Kfir M. Cohen (Student Member, IEEE) received the BSc (summa cum laude) and MSc degrees in the electrical engineering from the Faculty of the Technion – Israel Institute for Technology, Haifa, Israel, in 2006 and 2013, respectively. He is currently working toward the PhD degree with King’s Communications, Learning and Information Processing lab at the Department of Engineering of King’s College London (KCL). 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, conformal prediction, signal processing, as well as their applications to communications systems.

Osvaldo Simeone (Fellow, IEEE) received the MSc (with honors) and PhD degrees in information engineering from Politecnico di Milano, Milan, Italy, in 2001 and 2005, respectively. He is a professor of information engineering. He co-directs the Centre for Intelligent Information Processing Systems within the Department of Engineering of King’s College London, where he also runs the King’s Communications, Learning and Information Processing lab. From 2006 to 2017, he was a faculty member of the Electrical and Computer Engineering (ECE) Department, New York Institute of Technology (NIIT), 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. He is a co-recipient of the 2022 IEEE Communications Society Outstanding Paper Award, the 2021 IEEE Vehicular Technology Society Jack Neubauer Memorial Award, the 2019 IEEE Communication Society Best Tutorial Paper Award, the 2018 IEEE Signal Processing Best Paper Award, the 2017 JCN Best Paper Award, 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 was the chair of the Signal Processing for Communications and Networking Technical Committee of the IEEE Signal Processing Society in 2022, as well as of the U.K. & Ireland Chapter of the IEEE Information Theory Society from 2017 to 2022. He was a distinguished lecturer of the IEEE Communications Society in 2021 and 2022, and he was a distinguished lecturer of the IEEE Information Theory Society in 2017 and 2018. He is the author of the textbook “Machine Learning for Engineers” published by Cambridge University Press, four monographs, two edited books, and more than 200 research journal and magazine papers. He is a fellow of the IET, EPSRC.

Sangwoo Park (Member, IEEE) received the BSc degree in physics from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea in 2014, the MSE and PhD degrees in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea in 2016 and 2020. From April to December 2019, he was a visiting PhD student with King’s Communications, Learning and Information Processing lab (KCLIP), King’s College London (KCL), United Kingdom (U.K.). He is currently a research associate with KCLIP, KCL, U.K.. His research interests lie in practical, reliable AI and its application to engineering systems including wireless communications and quantum information processing.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.