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

Anonymous Author(s)
Affiliation
Address
email

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.

1 Introduction

1.1 Context and Motivation

Recent work on calibration for AI has focused on Bayesian learning, or related ensembling methods, as means to quantify epistemic uncertainty [Finn et al., 2018, Yoon et al., 2018, Ravi and Beatson, 2018, Jose et al., 2022]. However, recent studies have shown the limitations of Bayesian learning when the assumed model likelihood or prior distribution are *misspecified* [Masegosa, 2020]. Furthermore, exact Bayesian learning is computationally infeasible, calling for approximations such as Monte Carlo (MC) sampling [Robert et al., 1999] and variational inference (VI) [Blundell et al., 2015]. Overall, under practical conditions, Bayesian learning does not provide *formal guarantees* of calibration.

Conformal prediction (CP) [Vovk et al., 2005] 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. 1, CP produces *set predictors* that output a subset of the output space $\mathcal{Y}$ for each input x, with the property 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 τ, assume that we are given a data set $\mathcal{D}_\tau$ with N_τ samples, i.e., $\mathcal{D}_\tau = \{z_\tau[i]\}_{i=1}^{N_\tau}$, where the ith 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* $\Gamma(\cdot|\mathcal{D}_\tau, \xi) : \mathcal{X}_\tau \to 2^{\mathcal{Y}_\tau}$, specified by a hyperparameter

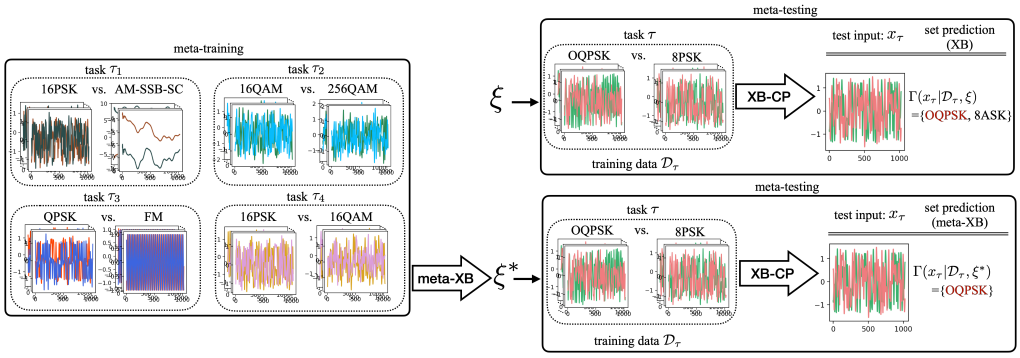

Figure 1: Illustration of proposed meta-learned cross-validation-based CP (XB-CP) scheme, referred to as meta-XB. The example refers to the problem of classifying received radio signals depending on the modulation scheme used to generate it, e.g., QPSK or FM [O'Shea et al., 2016, 2018]. Based on data from multiple tasks, meta-XB optimizes a hyperparameter vector ξ^* by minimizing the average set prediction size. As compared to conventional XB, shown on the top-right part of the figure, which uses a fixed hyperparameter vector ξ, meta-XB can achieve reduced set prediction size, while maintaining the per-task validity property (1).

vector ξ, 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 $\mathcal{D}_\tau$. Calibration amounts to the *per-task validity* condition

$$\mathbb{P}(\mathbf{y}_\tau \in \Gamma(\mathbf{x}_\tau|\boldsymbol{\mathcal{D}}_\tau, \xi)) \geq 1 - \alpha, \tag{1}$$

which indicates that the set predictor $\Gamma(\mathbf{x}_\tau|\boldsymbol{\mathcal{D}}_\tau, \xi)$ contains the true target $\mathbf{y}_\tau$ with probability at least $1 - \alpha$. In (1), the probability $\mathbb{P}(\cdot)$ is taken over the ground-truth, exchangeable, joint distribution $p(\mathcal{D}_\tau, z_\tau)$, 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 [Vovk et al., 2005]. The validation subset is used to calibrate the set prediction $\Gamma_\alpha^{\text{VB}}(x_\tau|\mathcal{D}_\tau, \xi)$ on a test example x_τ for a given desired miscoverage level α in (1). The drawback of this approach is that validation data is not used for training, resulting in inefficient set predictors $\Gamma_\alpha^{\text{VB}}(x_\tau|\mathcal{D}_\tau, \xi)$ in the presence of a limited number N_τ of data samples. The average size of a set predictor $\Gamma(x_\tau|\mathcal{D}_\tau, \xi)$, referred to as *inefficiency*, is defined as

$$\mathcal{L}_\tau(\xi) = \mathbb{E}\big|\Gamma(\mathbf{x}_\tau|\boldsymbol{\mathcal{D}}_\tau, \xi)\big|, \tag{2}$$

where the average is taken with respect to the ground-truth joint distribution $p(\mathcal{D}_\tau, z_\tau)$.

A more efficient CP set predictor was introduced by Barber et al. [2021] based on cross-validation. The *cross-validation-based CP* (XB-CP) set predictor $\Gamma_\alpha^{K\text{-XB}}(x_\tau|\mathcal{D}_\tau, \xi)$ splits the data set $\mathcal{D}_\tau$ 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)[1].

Further improvements in efficiency can be obtained via *meta-learning* [Thrun, 1998]. 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 ξ of the set predictor $\Gamma(\mathbf{x}_\tau|\boldsymbol{\mathcal{D}}_\tau, \xi)$ to be used on a new task $\tau \sim p(\tau)$. Specifically, reference [Fisch et al., 2021] introduced a meta-learning-based method that modifies VB-CP. The resulting *meta-VB* algorithm satisfies a looser validity condition with respect to the *per-task* inequality (1), in which the probability in (1) is no smaller than $1 - \alpha$ only *on average* with respect to the task distribution $p(\tau)$.

In this paper, we introduce a novel meta-learning approach, termed *meta-XB*, with the aim of reducing the inefficiency (2) of XB-CP, while preserving, unlike [Fisch et al., 2021], the per-task validity condition (1) for every task τ.

[1]We refer here in particular to the jacknife-mm scheme presented in Section 2.2 of [Barber et al., 2021].

2 Meta-Learning Algorithm for XB-CP (Meta-XB)

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

2.1 Meta-Learning

Meta-learning utilizes data from multiple tasks to enhance the efficiency of the learning procedure for new tasks. Following the standard meta-learning formulation [Baxter, 2000, Amit and Meir, 2018], as anticipated in Section 1, the learning environment is characterized by a task distribution $p(\tau)$ over the task identifier $\boldsymbol{\tau}$. Given T meta-training tasks realizations $\boldsymbol{\tau}_1 = \tau_1, \ldots, \boldsymbol{\tau}_T = \tau_T$ drawn i.i.d. from the task distribution $p(\tau)$, the *meta-training data set* $\mathcal{D}_{\tau_{1:T}} := \{\{\mathcal{D}_t^j, z_t^j\}_{j=1}^{M_t}\}_{t=1}^T$ consists of M_t realizations $\{\mathcal{D}_t^j, z_t^j\}_{j=1}^{M_t}$ of data sets $\mathcal{D}_{\tau_t}^j = \mathcal{D}_t^j$ with $N_{\tau_t} = N_t$ examples and test sample $z_{\tau_t}^j = z_t^j$ for each task τ_t. Pairs $\{\mathcal{D}_t^j, z_t^j\}_{j=1}^{M_t}$ are generated i.i.d. from the joint distribution $p(\mathcal{D}_{\tau_t}, z_{\tau_t})$, satisfying exchangeability assumption for all tasks t.

The goal of meta-learning for CP is to optimize the vector of hyperparameter ξ based on the meta-training data $\mathcal{D}_{\tau_{1:T}}$, so as to obtain a more efficient set predictor $\Gamma(x_\tau|\mathcal{D}_\tau, \xi)$. While reference [Fisch et al., 2021] proposed a meta-learning solution for VB-CP [Vovk et al., 2005], here we introduce a meta-learning method for XB-CP.

2.2 Cross-Validation-Based Conformal Prediction (XB-CP)

XB-CP leverages K-fold cross-validation [Barber et al., 2021]. K-fold cross-validation partitions the per-task data set $\mathcal{D}_\tau = \{z_\tau[i]\}_{i=1}^{N_\tau}$ into $K \geq 2$ disjoint subsets $\mathcal{D}_{\tau,1}, \ldots, \mathcal{D}_{\tau,K}$ such that the condition $\bigcup_{k=1}^K \mathcal{D}_{\tau,k} = \mathcal{D}_\tau$ is satisfied. We define the *leave-one-out* data set $\mathcal{D}_{\tau,\neg k} = \bigcup_{k'=1,k'\neq k}^K \mathcal{D}_{\tau,k'}$ that excludes the subset $\mathcal{D}_{\tau,k}$. We also introduce a mapping function $k : \{1, \ldots, N_\tau\} \to \{1, \ldots, K\}$ to identify the subset $\mathcal{D}_{\tau,k(i)}$ that includes the sample $z_\tau[i]$, i.e., $z_\tau[i] \in \mathcal{D}_{\tau,k(i)}$.

A *nonconformity (NC) score* is a function $\mathrm{NC}(z|\tilde{\mathcal{D}}_\tau, \xi)$ that maps a data set $\tilde{\mathcal{D}}_\tau$ 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 $\mathrm{NC}(z|\{\tilde{z}_\tau[1], \ldots, \tilde{z}_\tau[\tilde{N}]\}, \xi) = \mathrm{NC}(z|\{\tilde{z}_\tau[\pi(1)], \ldots, \tilde{z}_\tau[\pi(\tilde{N})]\}, \xi)$ for any permutation operator $\pi(\cdot)$. A good NC score should express how poorly the point (x_τ, y) "conforms" to the data set $\tilde{\mathcal{D}}_\tau$. 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 $p(\phi|\tilde{\mathcal{D}}_\tau, \xi)$, which describes the output $\boldsymbol{\phi}$ of the algorithm as a function of training data set $\tilde{\mathcal{D}}_\tau \subseteq \mathcal{D}_\tau$ and hyperparameter vector ξ.

Given a test input x_τ, XB-CP computes the NC score for a candidate pair $z = (x_\tau, y)$ with $y \in \mathcal{Y}_\tau$ by taking the minimum NC score $\mathrm{NC}(z|\mathcal{D}_{\tau,\neg k}, \xi)$ over all possible subsets $k \in \{1, \ldots, K\}$, i.e., as $\min_{k\in\{1,\ldots,K\}} \mathrm{NC}(z|\mathcal{D}_{\tau,\neg k}, \xi)$. Furthermore, for each data point $z_\tau[i] \in \mathcal{D}_\tau$, the NC score is evaluated by excluding the subset $\mathcal{D}_{\tau,k(i)}$ as $\mathrm{NC}(z_\tau[i]|\mathcal{D}_{\tau,\neg k(i)}, \xi)$. Note that evaluating the resulting $N_\tau + 1$ NC scores requires running the training algorithm $p(\phi|\mathcal{D}_{\tau,\neg k}, \xi)$ K times, once for each subset $\mathcal{D}_{\tau,\neg k}$. Finally, a candidate $y \in \mathcal{Y}_\tau$ is included in the prediction set $\Gamma_\alpha^{K\text{-XB}}(x_\tau|\mathcal{D}_\tau, \xi)$ if the NC score for $z = (x_\tau, y)$ is smaller (or equal) than for a fraction (at least) $\lfloor \alpha'(N_\tau + 1) \rfloor / N_\tau$ of the validation data points with $\alpha' = \alpha - \frac{1-K/N_\tau}{K+1}$. We refer to supplementary material for details.

2.3 Meta-XB

Meta-XB aims at finding a hyperparameter vector ξ that minimizes the average size of the prediction set $\Gamma_\alpha^{K\text{-XB}}(x_\tau|\mathcal{D}_\tau, \xi)$ for tasks τ that follow the distribution $p(\tau)$. To this end, it addresses the problem of minimizing the empirical average of the sizes of the prediction sets $\Gamma_\alpha^{K\text{-XB}}(x_{\tau_t}|\mathcal{D}_{\tau_t}, \xi)$ across the meta-training tasks $\tau_1, \ldots, \tau_T$ over the hyperparameter vector ξ. This amounts to the optimization

$$\xi^* = \arg\min_\xi \frac{1}{T} \sum_{t=1}^T \frac{1}{M_t} \sum_{j=1}^{M_t} \left|\Gamma_\alpha^{K\text{-XB}}(x_t^j|\mathcal{D}_t^j, \xi)\right|, \tag{3}$$

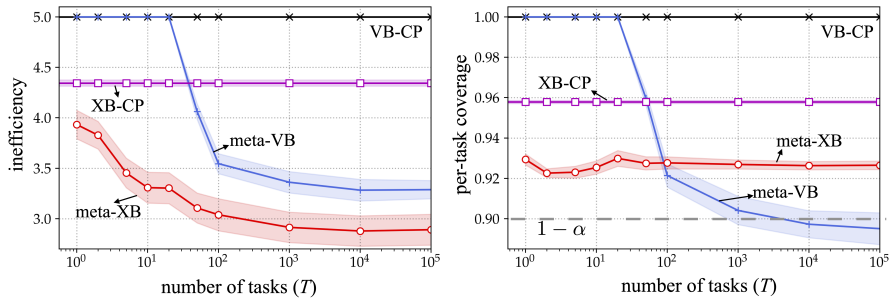

Figure 2: Per-task inefficiency and coverage for VB-CP, XB-CP, meta-VB, and meta-XB for the synthetic-data example in [Romano et al., 2020]. Each data set $\mathcal{D}_\tau$ contains $N_\tau = 9$ examples, while the meta-training data set $\mathcal{D}_{\tau_{1:T}}$ consists of 500 examples per task from $M_t = 50$ realizations. The shaded areas correspond to confidence intervals covering 95% of the realized values.

where the first sum is over the meta-training tasks and the second is over the available data for each task. Since it involves hard comparisons among NC scores, the size $|\Gamma_\alpha^{K\text{-XB}}(x|\mathcal{D},\xi)|$ is not a differentiable function of the hyperparameter vector ξ. Therefore, in order to adress (3) via gradient descent, we introduce a differentiable *soft inefficiency* criterion by replacing the indicator function with the sigmoid $\sigma(u) = (1+\exp(-u/c_\sigma))^{-1}$ for some $c_\sigma > 0$; the quantile $Q_{1-\alpha}(\cdot)$ with a differentiable soft empirical quantile $\hat{Q}_{1-\alpha}(\cdot)$ constructed via the *pinball loss* [Koenker and Bassett Jr, 1978]; and the minimum operator with the softmin function [Goodfellow et al., 2016]. Details can be found in the supplementary material.

3 Experiments

In this section, we provide experimental results to validate the performance of meta-XB in terms of *(i)* per-task coverage $\mathbb{P}(\mathbf{y}_\tau \in \Gamma(\mathbf{x}_\tau|\boldsymbol{\mathcal{D}}_\tau,\xi))$; and *(ii)* per-task inefficiency (2). As benchmark schemes, we consider *(i)* VB-CP, *(ii)* XB-CP, and *(iii)* meta-VB [Fisch et al., 2021]. We refer to the supplementary material for other details and additional experiments including *mini*Imagenet classification [Vinyals et al., 2016] and modulation classification [O'Shea et al., 2018].

3.1 Multinomial Model and Inhomogeneous Features

We consider here the synthetic-data experiment introduced in [Romano et al., 2020]. In Fig. 2, we demonstrate the performance of the considered set predictors as a function of number of tasks T. 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 [Yin et al., 2019, Jose and Simeone, 2020]. For example, meta-XB with $T = 100$ tasks obtain an average prediction set size of 3, while XB-CP has an inefficiency larger than 4. Furthermore, all schemes satisfy the validity condition (1), except for meta-VB for $T \gtrsim 10^4$, confirming the analytical results.

4 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. Through experimental results, 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 [Stutz et al., 2021, Einbinder et al., 2022], or with stochastic set predictors.

References

Ron Amit and Ron Meir. Meta-learning by adjusting priors based on extended PAC-Bayes theory. In *Proc. of Int. Conf. Machine Learning (ICML)*, pages 205–214, Jul 2018.

Rina Foygel Barber, Emmanuel J Candes, Aaditya Ramdas, and Ryan J Tibshirani. Predictive inference with the jackknife+. *The Annals of Statistics*, 49(1):486–507, 2021.

Jonathan Baxter. A model of inductive bias learning. *Journal of Artificial Intelligence Research*, 12: 149–198, March 2000.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International conference on machine learning*, pages 1613–1622. PMLR, 2015.

Bat-Sheva Einbinder, Yaniv Romano, Matteo Sesia, and Yanfei Zhou. Training uncertainty-aware classifiers with conformalized deep learning. *arXiv preprint arXiv:2205.05878*, 2022.

Chelsea Finn, Kelvin Xu, and Sergey Levine. Probabilistic model-agnostic meta-learning. *arXiv preprint arXiv:1806.02817*, 2018.

Adam Fisch, Tal Schuster, Tommi Jaakkola, and Regina Barzilay. Few-shot conformal prediction with auxiliary tasks. In *International Conference on Machine Learning*, pages 3329–3339. PMLR, 2021.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

Sharu Theresa Jose and Osvaldo Simeone. Information-theoretic bounds on transfer generalization gap based on Jensen-Shannon divergence. *arXiv preprint: arXiv: 2010.09484*, 2020.

Sharu Theresa Jose, Sangwoo Park, and Osvaldo Simeone. Information-theoretic analysis of epistemic uncertainty in bayesian meta-learning. In *International Conference on Artificial Intelligence and Statistics*, pages 9758–9775. PMLR, 2022.

Roger Koenker and Gilbert Bassett Jr. Regression quantiles. *Econometrica: journal of the Econometric Society*, pages 33–50, 1978.

Andres Masegosa. Learning under model misspecification: Applications to variational and ensemble methods. *Advances in Neural Information Processing Systems*, 33:5479–5491, 2020.

Timothy J O'Shea, Johnathan Corgan, and T Charles Clancy. Convolutional radio modulation recognition networks. In *International conference on engineering applications of neural networks*, pages 213–226. Springer, 2016.

Timothy James O'Shea, Tamoghna Roy, and T Charles Clancy. Over-the-air deep learning based radio signal classification. *IEEE Journal of Selected Topics in Signal Processing*, 12(1):168–179, 2018.

Sachin Ravi and Alex Beatson. Amortized bayesian meta-learning. In *International Conference on Learning Representations*, 2018.

Christian P Robert, George Casella, and George Casella. *Monte Carlo statistical methods*, volume 2. Springer, 1999.

Yaniv Romano, Matteo Sesia, and Emmanuel Candes. Classification with valid and adaptive coverage. *Advances in Neural Information Processing Systems*, 33:3581–3591, 2020.

David Stutz, Ali Taylan Cemgil, Arnaud Doucet, et al. Learning optimal conformal classifiers. *arXiv preprint arXiv:2110.09192*, 2021.

Sebastian Thrun. Lifelong learning algorithms. In *Learning to learn*, pages 181–209. Springer, 1998.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. *arXiv preprint arXiv:1606.04080*, 2016.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*. Springer Science & Business Media, 2005.

Mingzhang Yin, George Tucker, Mingyuan Zhou, Sergey Levine, and Chelsea Finn. Meta-Learning Without Memorization. *arXiv preprint arXiv:1912.03820*, 2019.

Jaesik Yoon, Taesup Kim, Ousmane Dia, Sungwoong Kim, Yoshua Bengio, and Sungjin Ahn. Bayesian model-agnostic meta-learning. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 7343–7353, 2018.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See supplementary material and conclusion.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] See supplementary material.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [N/A] We provide implementation details in supplementary material but could not upload the code as we cannot upload zip files through the system.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See supplementary material.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Fig. 2.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See supplementary material.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We used RadioML 2018.01A dataset [O'Shea et al., 2018] and *mini*Imagenet dataset [Vinyals et al., 2016] which are both available online.
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

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

1 Main Contributions

As summarized in the main document, this work introduces a novel meta-learning approach, termed *meta-XB*, with the aim of reducing the inefficiency (2) of XB-CP, while preserving the per-task validity condition (1) for every task τ unlike [Fisch et al., 2021]. In this supplementary material, we provide full details, and we also incorporate in the design of meta-XB the *adaptive nonconformity* (NC) scores introduced in [Romano et al., 2020]. As argued in [Romano et al., 2020] for conventional CP, adaptive NC scores are empirically known to improve the *per-task conditional validity* condition

$$\mathbb{P}(\mathbf{y}_\tau \in \Gamma(\mathbf{x}_\tau | \boldsymbol{\mathcal{D}}_\tau, \xi) | \mathbf{x}_\tau = x_\tau) \geq 1 - \alpha. \tag{4}$$

This condition is significantly stronger than (1) as it holds for any test input x_τ. A summary of the considered CP schemes can be found in Fig. 3.

Overall, the contribution of this work, including also the supplementary material can be summarized as follows:

- We introduce meta-XB, a meta-learning algorithm for XB-CP, that can reduce the average prediction set size (2) as compared to XB-CP, while satisfying the per-task validity condition (1), unlike existing meta-learning algorithms for CP;
- We incorporate adaptive NC scores [Romano et al., 2020] in the design of meta-XB, demonstrating via experiments that adaptive NC scores can enhance conditional validity as defined by condition (4).

2 Definitions and Preliminaries

In this section, we describe necessary background material on CP [Vovk et al., 2005, Balasubramanian et al., 2014], VB-CP [Vovk et al., 2005], XB-CP [Barber et al., 2021], and adaptive NC scores [Romano et al., 2020].

2.1 Nonconformity (NC) Scores

At a high level, given an input x_τ for some learning task τ, CP outputs a prediction set $\Gamma(x_\tau | \mathcal{D}_\tau, \xi)$ that includes all outputs $y \in \mathcal{Y}_\tau$ such that the pair (x_τ, y) *conforms* well with the examples in the available data set $\mathcal{D}_\tau = \{z_\tau[i] = (x_\tau[i], y_\tau[i])\}_{i=1}^{N_\tau}$. We recall from Section 1 that ξ represents a vector of hyperparameter. The key underlying assumption is that data set $\mathcal{D}_\tau$ and test pair $z_\tau = (x_\tau, y_\tau)$ are realizations of *exchangeable* random variables $\boldsymbol{\mathcal{D}}_\tau$ and $\mathbf{z}_\tau$.

Assumption 1 *For any learning task τ, data set $\boldsymbol{\mathcal{D}}_\tau$ and a test data point $\mathbf{z}_\tau$ are exchangeable random variables, i.e., the joint distribution $p(\mathcal{D}_\tau, z_\tau) = p(z_\tau[1], \ldots, z_\tau[N_\tau], z_\tau)$ is invariant to any permutation of the variables $\{\mathbf{z}_\tau[1], \ldots, \mathbf{z}_\tau[N_\tau], \mathbf{z}_\tau\}$. 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 $z_\tau = z_\tau[N_\tau + 1]$, for any permutation operator $\pi(\cdot)$. Note that the standard assumption of i.i.d. random variables satisfies exchangeability.*

CP method	nonconformity (NC) score	per-task validity (1)	efficiency (2)	per-task conditional validity* (3)	example
VB	conventional [Vovk et al., 2005]	☺	☹	☹	
	adaptive [Romano et al., 2020]	☺	☹	☺	
XB [Barber et al., 2021]	conventional	☺	😐	☹	
	adaptive	☺	😐	☺	
meta-VB [Fisch et al., 2021, Park et al., 2022]	conventional	☹	☺	☹	
meta-XB (this paper)	conventional	☺	☺	☹	
	adaptive	☺	☺	☺	

*empirical evaluations

Figure 3: Conformal prediction (CP)-based set predictors in the presence of limited data samples: Validation-based CP (VB-CP) [Vovk et al., 2005] and the more efficient cross-validation-based CP (XB-CP) [Barber et al., 2021] provide set predictors that satisfy the per-task validity condition (1); while previous works on meta-learning for VB-CP [Fisch et al., 2021, Park et al., 2022], which aims at improving efficiency, do not offer validity guarantees when conditioning on a given task τ. In contrast, the proposed meta-XB algorithm outputs efficient set predictors with guaranteed per-task validity. By incorporating adaptive NC scores [Romano et al., 2020], meta-XB can also empirically improve per-input conditional validity (see (4)). The last column illustrates efficiency, per-task validity, and per-task conditional 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 τ; while per-task conditional validity (see (4)) holds when the set predictor is valid for each input. Conditional validity typically results in prediction sets of different sizes depending on the input [Romano et al., 2019, Lin et al., 2021, LeRoy and Zhao, 2021, Izbicki et al., 2020]. Inefficiency (see (2)) measures the average size of the prediction set.

CP measures conformity via *NC scores*, which are generally functions of the hyperparameter vector ξ, and are defined as follows.

Definition 1 (NC score) *For a given learning task τ, given a data set $\tilde{\mathcal{D}}_\tau = \{\tilde{z}_\tau[i] = (\tilde{x}_\tau[i], \tilde{y}_\tau[i])\}_{i=1}^{\tilde{N}_\tau} \subseteq \mathcal{D}_\tau$ with $\tilde{N}_\tau \leq N_\tau$ samples, a nonconformity (NC) score is a function $NC(z|\tilde{\mathcal{D}}_\tau, \xi)$ that maps the data set $\tilde{\mathcal{D}}_\tau$ 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 $NC(z|\{\tilde{z}_\tau[1], \ldots, \tilde{z}_\tau[\tilde{N}]\}, \xi) = NC(z|\{\tilde{z}_\tau[\pi(1)], \ldots, \tilde{z}_\tau[\pi(\tilde{N})]\}, \xi)$ for any permutation operator $\pi(\cdot)$.*

A good NC score should express how poorly the point (x_τ, y) "conforms" to the data set $\tilde{\mathcal{D}}_\tau$. 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 $p(\phi|\tilde{\mathcal{D}}_\tau, \xi)$, which describes the output $\boldsymbol{\phi}$ of the algorithm as a function of training data set $\tilde{\mathcal{D}}_\tau \subseteq \mathcal{D}_\tau$ and hyperparameter vector ξ. 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 [Guedj, 2019, Angelino et al., 2016, Simeone, 2022]. The hyperparameter vector ξ may determine, e.g., learning rate schedule or initialization.

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

as

$$NC(z|\tilde{\mathcal{D}}_\tau, \xi) := \mathbb{E}_{\boldsymbol{\phi}\sim p(\phi|\tilde{\mathcal{D}}_\tau,\xi)}\big[\ell_\tau(z|\boldsymbol{\phi})\big]. \tag{5}$$

Due to the permutation-invariance of the training algorithm, it can be readily checked that (5) is a valid NC score as per Definition 1.

2.2 Validation-Based Conformal Prediction (VB-CP)

VB-CP [Vovk et al., 2005] divides the data set $\mathcal{D}_\tau$ into a training data set $\mathcal{D}_\tau^{\text{tr}}$ of N_τ^{tr} samples and a validation data set $\mathcal{D}_\tau^{\text{val}}$ of N_τ^{val} samples with $N_\tau^{\text{tr}} + N_\tau^{\text{val}} = N_\tau$. It uses the training data set $\mathcal{D}_\tau^{\text{tr}}$ to evaluate the NC scores $\text{NC}(z|\mathcal{D}_\tau^{\text{tr}}, \xi)$, while the validation data set $\mathcal{D}_\tau^{\text{val}}$ is leveraged to construct the set predictor $\Gamma_\alpha^{\text{VB}}(x_\tau|\mathcal{D}_\tau, \xi)$ as detailed next.

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

Definition 3 *The $(1-\alpha)$-empirical quantile $Q_{1-\alpha}(\{a[i]\}_{i=1}^M)$ of M real numbers $a[1], \ldots, a[M]$, with $a[i] \in \mathbb{R}$, is defined as the $\lceil (1-\alpha)(M+1)\rceil$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

$$\Gamma_\alpha^{\text{VB}}(x_\tau|\mathcal{D}_\tau, \xi) = \Big\{ y \in \mathcal{Y}_\tau \ : \ \text{NC}(z|\mathcal{D}_\tau^{\text{tr}}, \xi) \ \leq \ Q_{1-\alpha}\big(\{\text{NC}(z_\tau[i]|\mathcal{D}_\tau^{\text{tr}}, \xi)\}_{i=1}^{N_\tau^{\text{val}}}\big) \text{ with } z = (x_\tau, y) \Big\}. \tag{6}$$

Intuitively, by the exchangeability condition, the empirical ordering condition among the NC scores used to define set (6) ensures the validity condition (1) [Vovk et al., 2005].

Theorem 1 *[Vovk et al., 2005] Under Assumption 1, for any miscoverage level $\alpha \in [1/(N_\tau^{val}+1), 1)$, given any NC score as per Definition 1, the VB-CP set predictor (6) satisfies the validity condition (1).*

2.3 Cross-Validation-Based Conformal Prediction (XB-CP)

In VB-CP, the validation data set is only used to compute the empirical quantile in (6), and is hence not leveraged by the training algorithm $p(\phi|\mathcal{D}_\tau^{\text{tr}}, \xi)$. This generally causes the inefficiency (2) of VB-CP to be large if number of data points, N_τ, is small. XB-CP addresses this problem via K-fold cross-validation [Barber et al., 2021]. K-fold cross-validation partitions the per-task data set $\mathcal{D}_\tau = \{z_\tau[i]\}_{i=1}^{N_\tau}$ into $K \geq 2$ disjoint subsets $\mathcal{D}_{\tau,1}, \ldots, \mathcal{D}_{\tau,K}$ such that the condition $\bigcup_{k=1}^K \mathcal{D}_{\tau,k} = \mathcal{D}_\tau$ is satisfied. We define the *leave-one-out* data set $\mathcal{D}_{\tau,\neg k} = \bigcup_{k'=1,k'\neq k}^K \mathcal{D}_{\tau,k'}$ that excludes the subset $\mathcal{D}_{\tau,k}$. We also introduce a mapping function $k : \{1, \ldots, N_\tau\} \to \{1, \ldots, K\}$ to identify the subset $\mathcal{D}_{\tau,k(i)}$ that includes the sample $z_\tau[i]$, i.e., $z_\tau[i] \in \mathcal{D}_{\tau,k(i)}$.

We focus here on a variant of XB-CP that is referred to as min-max jacknife+ in [Barber et al., 2021]. This variant has stronger validity guarantees than the jacknife+ scheme also studied in [Barber et al., 2021]. Accordingly, given a test input x_τ, XB-CP computes the NC score for a candidate pair $z = (x_\tau, y)$ with $y \in \mathcal{Y}_\tau$ by taking the minimum NC score $\text{NC}(z|\mathcal{D}_{\tau,\neg k}, \xi)$ over all possible subsets $k \in \{1, \ldots, K\}$, i.e., as $\min_{k\in\{1,\ldots,K\}} \text{NC}(z|\mathcal{D}_{\tau,\neg k}, \xi)$. Furthermore, for each data point $z_\tau[i] \in \mathcal{D}_\tau$, the NC score is evaluated by excluding the subset $\mathcal{D}_{\tau,k(i)}$ as $\text{NC}(z_\tau[i]|\mathcal{D}_{\tau,\neg k(i)}, \xi)$. Note that evaluating the resulting $N_\tau + 1$ NC scores requires running the training algorithm $p(\phi|\mathcal{D}_{\tau,\neg k}, \xi)$ K times, once for each subset $\mathcal{D}_{\tau,\neg k}$. Finally, a candidate $y \in \mathcal{Y}_\tau$ is included in the prediction set if the NC score for $z = (x_\tau, y)$ is smaller (or equal) than for a fraction (at least) $\lfloor \alpha'(N_\tau + 1)\rfloor / N_\tau$ of the validation data points with $\alpha' = \alpha - \frac{1-K/N_\tau}{K+1}$.

Overall, given data set $\mathcal{D}_\tau = \{z_\tau[i] = (x_\tau[i], y_\tau[i])\}_{i=1}^{N_\tau}$ and test input $x_\tau \in \mathcal{X}_\tau$, *K-fold XB-CP* produces the set predictor

$$\Gamma_\alpha^{K\text{-XB}}(x_\tau|\mathcal{D}_\tau, \xi) = \Big\{ y \in \mathcal{Y}_\tau \ : \ \sum_{i=1}^{N_\tau} \mathbf{1}\Big(\min_{k\in\{1,\ldots,K\}} \text{NC}(z|\mathcal{D}_{\tau,\neg k}, \xi) \leq \text{NC}(z_\tau[i]|\mathcal{D}_{\tau,\neg k(i)}, \xi)\Big) \ \geq \ \lfloor \alpha'(N_\tau+1) \rfloor \text{ with } z = (x_\tau, y) \Big\}, \tag{7}$$

where $\mathbf{1}(\cdot)$ is the indicator function ($\mathbf{1}(\text{true}) = 1$ and $\mathbf{1}(\text{false}) = 0$).

Theorem 2 *[Barber et al., 2021] Under Assumption 1, for any miscoverage level $\alpha \in \big[\frac{1}{N_\tau+1} + \frac{1-K/N_\tau}{K+1}, 1\big)$, given any NC score as per Definition 1, the XB-CP set predictor (7) satisfies the validity condition (1).*

While a proof of Theorem 2 for $K = N_\tau$ can be found in [Barber et al., 2021], the general case for $K < N_\tau$ follows from the same proof techniques in [Barber et al., 2021] and is included for completeness in Appendix A.1.

2.4 Adaptive Parametric NC Score

The CP methods reviewed so far achieve the per-task validity condition (1). In contrast, the per-input conditional validity (4) is only attainable with strong additional assumptions on the joint distribution $p(\mathcal{D}_\tau, x_\tau)$ [Vovk, 2012, Lei and Wasserman, 2014]. However, the *adaptive* NC score introduced by Romano et al. [2020] is known to empirically improve the per-input conditional validity of VB-CP (6) and XB-CP (7).

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 $p(y_\tau|x_\tau)$ is known. The most efficient (deterministic) set predictor satisfying the conditional coverage condition (4) would then be obtained as the smallest-cardinality subset of target values in $\mathcal{Y}_\tau$ that satisfies the conditional coverage condition (4), i.e.,

$$\Gamma_\alpha^*(x_\tau) = \underset{\Gamma\subseteq\mathcal{Y}_\tau}{\text{argmin}}\, |\Gamma| \text{ s.t.} \sum_{y\in\Gamma} p(y|x_\tau) \geq 1-\alpha. \tag{8}$$

Note that set (8) can be obtained by adding values $y \in \mathcal{Y}_\tau$ to set predictor $\Gamma_\alpha^*(x_\tau)$ in order from largest to smallest value of $p(y|x_\tau)$ until the constraint in (8) is satisfied.

In practice, the conditional distribution $p(y_\tau|x_\tau)$ is estimated via the model $p(y_\tau|x_\tau, \phi)$ where the parameter vector ϕ is produced by a training algorithm $p(\phi|\tilde{\mathcal{D}}_\tau, \xi)$ applied to some training data set $\tilde{\mathcal{D}}_\tau$. This yields the *naïve* set predictor

$$\Gamma_{\alpha^{\text{naïve}}}^{\text{naïve}}(x_\tau|\tilde{\mathcal{D}}_\tau, \xi) = \underset{\Gamma\subseteq\mathcal{Y}_\tau}{\text{argmin}}\, |\Gamma| \text{ s.t.} \sum_{y\in\Gamma} \mathbb{E}_{\boldsymbol{\phi}\sim p(\phi|\tilde{\mathcal{D}}_\tau,\xi)} p(y|x_\tau, \boldsymbol{\phi}) \geq 1-\alpha^{\text{naïve}}, \tag{9}$$

where we have used for generality the ensemble predictor obtained by averaging over the output $\boldsymbol{\phi} \sim p(\phi|\tilde{\mathcal{D}}_\tau, \xi)$ of the training algorithm. Unless the likelihood model is perfectly calibrated, i.e., unless the equality $p(y_\tau|x_\tau) = \mathbb{E}_{\boldsymbol{\phi}\sim p(\phi|\tilde{\mathcal{D}}_\tau,\xi)}[p(y_\tau|x_\tau, \phi)]$ holds, there is no guarantee that the set predictor in (9) satisfies the conditional coverage condition (4) or the marginal coverage condition (1) with $\alpha = \alpha^{\text{naïve}}$.

To tackle this problem, Romano et al. [2020] proposed to apply VB-CP or XB-CP with a modified NC score inspired by the naïve prediction (9).

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

$$NC^{ada}(z|\tilde{\mathcal{D}}_\tau, \xi) = \max_{\alpha^{naïve}\in[0,1]} \alpha^{naïve} \text{ s.t. } y \in \Gamma_{\alpha^{naïve}}^{naïve}(x_\tau|\tilde{\mathcal{D}}_\tau, \xi). \tag{10}$$

Intuitively, if the adaptive NC score is large, the pair z does not conform well with the probabilistic model $\mathbb{E}_{\boldsymbol{\phi}\sim p(\phi|\tilde{\mathcal{D}}_\tau,\xi)}p(y|x,\boldsymbol{\phi})$ obtained by training on set $\tilde{\mathcal{D}}_\tau$. The adaptive NC score satisfies the condition in Definition 1, and hence by Theorems 1 and 2, the set predictors (6) and (7) for VB-CP and XB-CP, respectively, are both valid when the adaptive NC score is used. Furthermore, Romano et al. [2020] 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 (9) under the assumption of a well-calibrated model.

The adaptive NC score (10) can be equivalently expressed as

$$\text{NC}^{\text{ada}}(z|\tilde{\mathcal{D}}_\tau,\xi) = \sum_{y'\in\mathcal{Y}_\tau} \mathbf{1}\big(p(y'|x,\tilde{\mathcal{D}}_\tau,\xi) \geq p(y|x,\tilde{\mathcal{D}}_\tau,\xi)\big)p(y'|x,\tilde{\mathcal{D}}_\tau,\xi), \tag{11}$$

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

3 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.

3.1 Meta-Learning

Up to now, we have focused on a single task τ. Meta-learning utilizes data from multiple tasks to enhance the efficiency of the learning procedure for new tasks. Following the standard meta-learning formulation [Baxter, 2000, Amit and Meir, 2018], as anticipated in Section 1, the learning environment is characterized by a task distribution $p(\tau)$ over the task identifier $\boldsymbol{\tau}$. Given T meta-training tasks realizations $\boldsymbol{\tau}_1 = \tau_1, \ldots, \boldsymbol{\tau}_T = \tau_T$ drawn i.i.d. from the task distribution $p(\tau)$, the *meta-training data set* $\mathcal{D}_{\tau_{1:T}} := \{\{\mathcal{D}_t^j, z_t^j\}_{j=1}^{M_t}\}_{t=1}^T$ consists of M_t realizations $\{\mathcal{D}_t^j, z_t^j\}_{j=1}^{M_t}$ of data sets $\mathcal{D}_{\tau_t}^j = \mathcal{D}_t^j$ with $N_{\tau_t} = N_t$ examples and test sample $z_{\tau_t}^j = z_t^j$ for each task τ_t. Pairs $\{\mathcal{D}_t^j, z_t^j\}_{j=1}^{M_t}$ are generated i.i.d. from the joint distribution $p(\mathcal{D}_{\tau_t}, z_{\tau_t})$, satisfying Assumption 1 for all tasks t.

The goal of meta-learning for CP is to optimize the vector of hyperparameter ξ based on the meta-training data $\mathcal{D}_{\tau_{1:T}}$, so as to obtain a more efficient set predictor $\Gamma(x_\tau|\mathcal{D}_\tau,\xi)$. While reference [Fisch et al., 2021] proposed a meta-learning solution for VB-CP [Vovk et al., 2005], here we introduce a meta-learning method for XB-CP.

3.2 Meta-XB

Meta-XB aims at finding a hyperparameter vector ξ that minimizes the average size of the prediction set $\Gamma_\alpha^{K\text{-XB}}(x_\tau|\mathcal{D}_\tau,\xi)$ in (7) for tasks τ that follow the distribution $p(\tau)$. To this end, it addresses the problem of minimizing the empirical average of the sizes of the prediction sets $\Gamma_\alpha^{K\text{-XB}}(x_{\tau_t}|\mathcal{D}_{\tau_t},\xi)$ across the meta-training tasks $\tau_1,\ldots,\tau_T$ over the hyperparameter vector ξ. This amounts to the optimization

$$\xi^* = \arg\min_\xi \frac{1}{T}\sum_{t=1}^T \frac{1}{M_t}\sum_{j=1}^{M_t} \big|\Gamma_\alpha^{K\text{-XB}}(x_t^j|\mathcal{D}_t^j,\xi)\big|, \tag{12}$$

where the first sum is over the meta-training tasks and the second is over the available data for each task. By (7), the size of the prediction set $|\Gamma_\alpha^{K\text{-XB}}(x|\mathcal{D},\xi)|$ is not a differentiable function of the hyperparameter vector ξ. Therefore, in order to address (12) via gradient descent, we introduce a differentiable *soft inefficiency* criterion by replacing the indicator function with the sigmoid $\sigma(u) = (1+\exp(-u/c_\sigma))^{-1}$ for some $c_\sigma > 0$; the quantile $Q_{1-\alpha}(\cdot)$ with a differentiable soft empirical quantile $\hat{Q}_{1-\alpha}(\cdot)$; and the minimum operator with the softmin function [Goodfellow et al., 2016].

For an input set $\{a[j]\}_{j=1}^M$, the softmin function is defined as [Goodfellow et al., 2016, Section 6.2.2.3]

$$\text{softmin}\big(\{a[j]\}_{j=1}^M\big) = \sum_{j=1}^M a[j]\frac{\exp(-a[j]/c_S)}{\sum_{i=1}^M \exp(-a[i]/c_S)}, \tag{13}$$

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

$$\hat{Q}_{1-\alpha}\big(\{a[j]\}_{j=1}^{M}\big) = \sum_{j=1}^{M+1} a[j] \frac{\exp(-\rho_{1-\alpha}(a[j]|\{a[j]\}_{j=1}^{M+1})/c_Q)}{\sum_{i=1}^{M+1}\exp(-\rho_{1-\alpha}(a[i]|\{a[j]\}_{j=1}^{M+1})/c_Q)}, \tag{14}$$

for some $c_Q > 0$ and $a[M+1] = \max(\{a[j]\}_{j=1}^{M}) + \delta$ for some $\delta > 0$, where we have used the *pinball loss* $\rho_{1-\alpha}(a|\{a[1], \ldots, a[M]\})$ [Koenker and Bassett Jr, 1978]

$$\begin{aligned}&\rho_{1-\alpha}(a|\{a[j]\}_{j=1}^{M}) \\ &= \alpha \sum_{j=1}^{M} \mathrm{ReLU}(a - a[j]) + (1-\alpha)\sum_{j=1}^{M}\mathrm{ReLU}(a[j] - a),\end{aligned} \tag{15}$$

with $\mathrm{ReLU}(a) = \max(0, a)$. With these definitions, the soft inefficiency metric is derived from (7) as follows (see details in Appendix A.2).

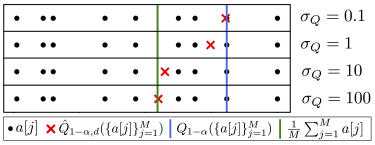

Figure 4: Trade-off between smoothness and accuracy of soft quantile. Dots represent the M input values, the blue line is the true empirical quantile $Q_{1-\alpha}(\cdot)$, and the green line is the mean of the M input values.

Definition 5 *Given a data set $\mathcal{D}_\tau$ and a test input x_τ, the soft inefficiency for the K-fold XB-CP predictor (7) is defined as*

$$\begin{aligned}&|\hat{\Gamma}_{\alpha}^{K\text{-}XB}(x_\tau|\mathcal{D}_\tau, \xi)| \\ &= \sum_{y\in\mathcal{Y}} \sigma\Big(\hat{Q}_{1-\alpha'}\Big(\Big\{NC(z_\tau[i]|\mathcal{D}_{\tau,\neg k(i)}, \xi) - softmin\big(\{NC((x_\tau, y)|\mathcal{D}_{\tau,\neg k}, \xi)\}_{k=1}^{K}\big)\Big\}_{i=1}^{N}\Big)\Big),\end{aligned} \tag{16}$$

where $\alpha' = \alpha - \frac{1-K/N_\tau}{K+1}$ and $c_\sigma, c_S, c_Q > 0$.

The parameters c_σ, c_S, and c_Q dictate the trade-off between smoothness and accuracy of the approximation $|\hat{\Gamma}_{\alpha}^{K\text{-XB}}(x_\tau|\mathcal{D}_\tau, \xi)|$ with respect to the true inefficiency $|\Gamma_{\alpha}^{K\text{-XB}}(x_\tau|\mathcal{D}_\tau, \xi)|$: As $c_\sigma, c_S, c_Q \to 0$, the approximation becomes increasingly accurate for any $\delta > 0$, as long as we have $\alpha \in \big[\frac{1}{N_\tau+1} + \frac{1-K/N_\tau}{K+1}, 1\big)$, but the function $|\hat{\Gamma}_{\alpha}^{K\text{-XB}}(x_\tau|\mathcal{D}_\tau, \xi)|$ is increasingly less smooth (see Fig. 4 for an illustration of the accuracy of the soft quantile).

Replacing the soft inefficiency (16) into problem (12) yields a differentiable program when conventional two-step NC scores (Definition 2) are used. We address the corresponding problem via stochastic gradient descent (SGD), whereby at each iteration a batch of tasks and examples per task are sampled. The overall meta-learning procedure is summarized in Algorithm 1.

3.3 Meta-XB with Adaptive NC Scores

Adaptive NC scores are not differentiable. Therefore, in order to enable the optimization of problem (12) with the soft inefficiency (16), we propose to replace the indicator function $\mathbf{1}(\cdot)$ in (11) with the sigmoid function $\sigma(\cdot)$. We also have found that approximating the number of outputs $y' \in \mathcal{Y}_\tau$ that satisfy (11) rather than direct application of sigmoid function empirically improves per-input coverage performance. This yields the *soft adaptive NC score* $\hat{\mathrm{NC}}^{\mathrm{ada}}(z|\tilde{\mathcal{D}}_\tau, \xi)$, which is detailed in Appendix B. With the soft adaptive NC score, meta-XB is then applied as in Algorithm 1.

3.4 Per-Task Validity of Meta-XB

As mentioned in Section 1, existing meta-learning schemes for CP cannot achieve the per-task validity condition in (1), requiring an additional marginalization over distribution $p(\tau)$ [Fisch et al., 2021] or achieving looser validity guarantees formulated as probably approximately correct (PAC)-bounds [Park et al., 2022]. In contrast, meta-XB has the following property.

Theorem 3 *Under Assumption 1, for any miscoverage level $\alpha \in \left[\frac{1}{N_\tau+1} + \frac{1-K/N_\tau}{K+1}, 1\right)$, given any NC score (Definition 1), the XB-CP set predictor (7) with $\xi = \xi^*$ in (12) 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 $p(\phi|\tilde{\mathcal{D}}_\tau, \xi^*)$ as required by Definition 2.

Algorithm 1: Meta-XB

Input: meta-training set $\mathcal{D}_{1:T} = \{\mathcal{D}_t\}_{t=1}^T$; number of examples $\{N_t\}_{t=1}^T$ to be used for set prediction; step size hyperparameter κ; approximation parameter c_S for softmin, c_Q for soft quantile, and c_σ for sigmoid; minibatch size for tasks $\tilde{T}$ and minibatch size for realization pairs $\tilde{M}_t$

Output: meta-learned hyperparameter vector ξ^*

initialize hyperparameter vector ξ
while convergence criterion not met **do**
  choose $\tilde{T}$ tasks randomly from set $\{1, \ldots, T\}$ and denote the corresponding task set as $\tilde{\mathcal{T}}$
  for each sampled task $t \in \tilde{\mathcal{T}}$ **do**
    randomly sample $\tilde{M}_t$ pairs from the data set $\mathcal{D}_{\tau_t}$, i.e., $\{\mathcal{D}_t^j, z_t^j\}_{j \in \tilde{\mathcal{J}}_t}$ denoting the corresponding index set as $\tilde{\mathcal{J}}_t$, and compute the soft inefficiency

$$\hat{\mathcal{L}}_t(\xi) = \frac{1}{\tilde{M}_t} \sum_{j \in \tilde{\mathcal{J}}_t} |\hat{\Gamma}_\alpha^{K\text{-XB}}(x_t^j | \mathcal{D}_t^j, \xi)|. \tag{17}$$

  end
  update hyperparameter vector $\xi \leftarrow \xi - \kappa \sum_{t \in \tilde{\mathcal{T}}} \nabla_\xi \hat{\mathcal{L}}_t(\xi)$
end
return the optimized hyperparameter vector ξ

4 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 [Masegosa, 2020], Bayesian learning may yield poor generalization performance [Masegosa, 2020, Morningstar et al., 2022, Wenzel et al., 2020]. Downweighting the prior distribution and/or the likelihood, as done in generalized Bayesian learning [Knoblauch et al., 2019, Simeone, 2022] or in "cold" posteriors [Wenzel et al., 2020], improve the generalization performance. In order to mitigate the model likelihood misspecification, alternative variational free energy metrics were introduced by Masegosa [2020] via second-order PAC-Bayes bounds, and by Morningstar et al. [2022] 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 [MacKay, 2003].

Bayesian meta-learning While frequentist meta-learning has shown remarkable success in few-shot learning tasks in terms of accuracy [Finn et al., 2017, Snell et al., 2017], improvements in terms of calibration can be obtained by Bayesian meta-learning that optimizes over a hyper-posterior distribution from multiple tasks [Amit and Meir, 2018, Finn et al., 2018, Yoon et al., 2018, Ravi and Beatson, 2018, Nguyen et al., 2020, Jose et al., 2022]. The hyper-prior can also be modelled as a stochastic process to avoid the bias caused by parametric models [Rothfuss et al., 2021].

CP-aware loss. Stutz et al. [2021] and Einbinder et al. [2022] proposed CP-aware loss functions to enhance the efficiency or per-input validity (4) of VB-CP. The drawback of these solutions is that they require a large amount of data samples, i.e., $N_\tau \gg 1$, unlike the meta-learning methods studied here.

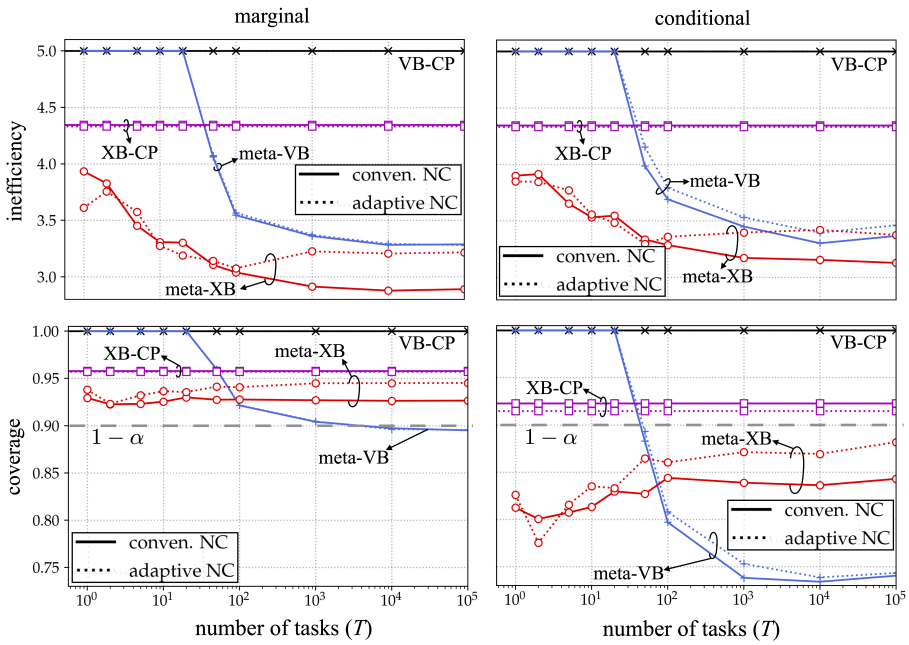

Figure 5: Per-task inefficiency and coverage (left) and per-task conditional inefficiency and coverage (right) for VB-CP, XB-CP, meta-VB, and meta-XB for the synthetic-data example in [Romano et al., 2020]. Each data set $\mathcal{D}_\tau$ contains $N_\tau = 9$ examples, while the meta-training data set $\mathcal{D}_{\tau_{1:T}}$ consists of 500 examples per task from $M_t = 50$ realizations.

Per-input validity and local validity. As discussed in Section 2.4, the per-input validity condition (4) cannot be satisfied without strong assumptions on the joint distribution $p(\mathcal{D}_\tau, z_\tau)$ [Vovk, 2012, Lei and Wasserman, 2014]. Given the importance of adapting the prediction set size to the input to capture heteroscedasticity [Romano et al., 2019, Izbicki et al., 2020], a looser **local validity** condition, which conditions on a subset of the input data space $A_{x_\tau} \subset \mathcal{X}_\tau$ containing the input x_τ of interest, i.e., $x_\tau \in A_{x_\tau}$, has been considered in [Lei and Wasserman, 2014, Foygel Barber et al., 2021]. Choosing a proper subset A_{x_τ} becomes problematic especially in high-dimensional input space [Izbicki et al., 2020, LeRoy and Zhao, 2021], and Tibshirani et al. [2019], Lin et al. [2021] proposed to reweight the samples outside the subset A_{x_τ} by treating the problem as *distribution-shift* between the data set $\mathcal{D}_\tau$ and the test input x_τ.

5 Experiments

In this section, we provide experimental results to validate the performance of meta-XB in terms of *(i)* per-task coverage $\mathbb{P}(\mathbf{y}_\tau \in \Gamma(\mathbf{x}_\tau|\boldsymbol{\mathcal{D}}_\tau, \xi))$; *(ii)* per-task inefficiency (2); *(iii)* per-task conditional coverage $\mathbb{P}(\mathbf{y}_\tau \in \Gamma(\mathbf{x}_\tau|\boldsymbol{\mathcal{D}}_\tau, \xi)|\mathbf{x}_\tau = x_\tau)$; and *(iv)* per-task conditional inefficiency $\mathbb{E}[|\Gamma(\mathbf{x}_\tau|\boldsymbol{\mathcal{D}}_\tau, \xi)||\mathbf{x}_\tau = x_\tau]$. To evaluate input-conditional quantities, we follow the approach in [Romano et al., 2020, Section S1.2]. As benchmark schemes, we consider *(i)* VB-CP, *(ii)* XB-CP, and *(iii)* meta-VB [Fisch et al., 2021], with either the conventional NC score (Definition 2 with log-loss $\ell_\tau(z|\phi)$) or adaptive NC score with Definition 4). Note that meta-VB was described in [Fisch et al., 2021] 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 $\mathcal{D}_\tau$ and the desired miscoverage level $\alpha = 0.1$. For the cross-validation-based set predictors XB-CP and meta-XB, we set number of folds to $K = N_\tau$. The aforementioned performance measures are estimated by averaging over 1000 realizations of data set $\mathcal{D}_\tau$ 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 $M_t(N_\tau + 1)$ i.i.d. examples, from which we sample $\tilde{M}_t$ pairs $\{\mathcal{D}_t^j, z_t^j\}_{j \in \tilde{\mathcal{J}}_t}$ when

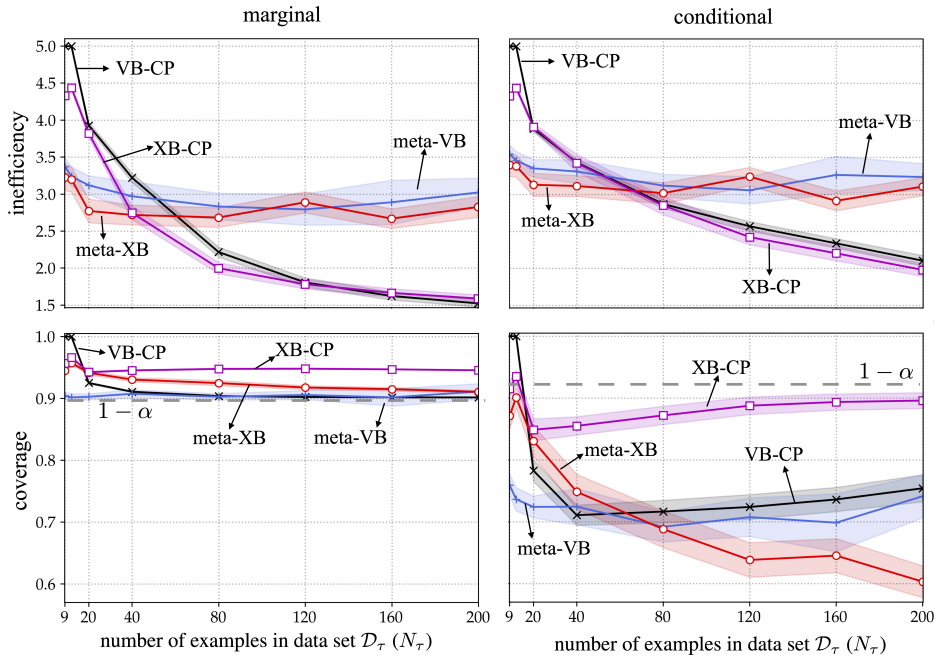

Figure 6: Per-task inefficiency and coverage (left) and per-task conditional inefficiency and coverage (right) for VB-CP, XB-CP, meta-VB, and meta-XB for the synthetic-data example in [Romano et al., 2020] using adaptive NC scores. Different numbers of examples N_τ for each data set $\mathcal{D}_\tau$ is considered here, while $T = 1000$ tasks are used to generate the meta-training data set $\mathcal{D}_{\tau_{1:T}}$ consists of $(N_\tau + 1)M_t$ examples per task from $M_t = 50$ realizations. The shaded areas correspond to confidence intervals covering 95% of the realized values.

computing inefficiency (17), with which we use Adam optimizer [Kingma and Ba, 2014] to update the hyperparameter vector ξ via SGD. Lastly, we set the value of the approximation parameters c_σ, c_S, and c_Q to be one.

Following [Romano et al., 2020], for VB-CP and XB-CP, we adopt a support vector classifier as training algorithm $p(\phi|\tilde{\mathcal{D}}_\tau, \xi)$ as it does not require any tuning of the hyperparameter vector ξ. In contrast, for meta-VB and meta-XB, we adopt a neural network classifier [Romano et al., 2019], and set the training algorithm $p(\phi|\tilde{\mathcal{D}}_\tau, \xi)$ to output the last iterate of a pre-defined number of steps of GD (1, unless specified otherwise) with initialization given by the hyperparameter vector ξ [Finn et al., 2017]. Note that using full-batch GD ensures the permutation-invariance of the training algorithm as required by Definition 2.

All the experiments are implemented by PyTorch [Paszke et al., 2019] and ran over a GPU server with single NVIDIA A100 card.

5.1 Multinomial Model and Inhomogeneous Features

We start with the synthetic-data experiment introduced in [Romano et al., 2020] in which the input $x \in \mathbb{R}^{10}$ 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 $\tau \in \mathbb{R}^{10 \times |\mathcal{Y}_\tau|}$ is sampled with i.i.d. standard Gaussian entries and the ground-truth conditional distribution $p(y_\tau|x_\tau)$ is defined as the categorical distribution

$$p(y_\tau = y|x_\tau) = \frac{\exp(x_\tau^\top \tau_y)}{\sum_{y'=1}^{|\mathcal{Y}_\tau|} \exp(x_\tau^\top \tau_{y'})}, \tag{18}$$

for $y \in \{1, \ldots, |\mathcal{Y}_\tau|\}$, where $\tau_y \in \mathbb{R}^{|\mathcal{Y}_\tau|}$ is the yth column of the task information matrix τ. The number of classes is $|\mathcal{Y}_\tau| = 5$ and neural network classifier consists of two hidden layers with

Exponential Linear Unit (ELU) activation [Clevert et al., 2015] in the hidden layers and a softmax activation in the last layer.

In Fig. 5, we demonstrate the performance of the considered set predictors as a function of number of tasks T. 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 [Yin et al., 2019, Jose and Simeone, 2020]. For example, meta-XB with $T = 100$ tasks obtain an average prediction set size of 3, while XB-CP has an inefficiency larger than 4. Furthermore, all schemes satisfy the validity condition (1), except for meta-VB for $T \gtrsim 10^4$, confirming the analytical results. Adaptive NC scores are seen to be instrumental in improving the conditional validity (4) when used with meta-XB, although this comes at the cost of a larger inefficiency.

Next, we investigate the impact of number of per-task examples N_τ in data set $\mathcal{D}_\tau$ using adaptive NC scores. As shown in Fig. 6, the average size of the set predictors decreases as N_τ grows larger. In the few-examples regime, i.e., with $N_\tau \leq 40$, the meta-learned set predictors meta-VB and meta-XB outperform the conventional set predictors VB-CP and XB-CP in terms of inefficiency. However, when N_τ is large enough, i.e., when $N_\tau \geq 80$, conventional set predictors are preferable, as transfer of knowledege across tasks becomes unnecessary, and possibly deleterious [Amit and Meir, 2018] (see also [Park et al., 2020] for related discussions). In terms of conditional coverage, Fig. 6 shows that cross-validation-based CP methods are preferable as compared to validation-based CP approaches.

5.2 Modulation Classification

We now consider the real-world *modulation classification* example illustrated in Fig. 1, in which the goal is classifying received radio signals depending on the modulation scheme used to generate it [O'Shea et al., 2016, 2018]. The RadioML 2018.01A data set consists $98,304$ inputs with dimension 2×1024, accounting for complex baseband signals sampled over 1024 time instants, generated from 24 different modulation types [O'Shea et al., 2018]. Each task τ 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 [Lake et al., 2011, Ravi and Larochelle, 2016]. We adopt VGG16 [Simonyan and Zisserman, 2014] as the neural network classifier as in [O'Shea et al., 2018]. Furthermore, for meta-VB and meta-XB, we apply a single GD step during meta-training and five GD steps during meta-testing [Finn et al., 2017, Ravi and Beatson, 2018].

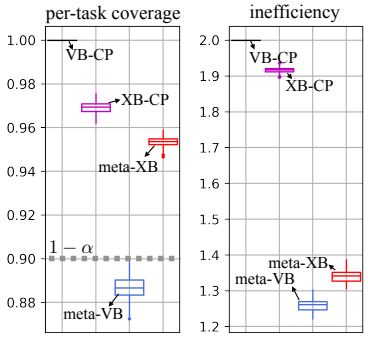

Figure 7: Per-task coverage and inefficiency of VB-CP, XB-CP, meta-VB, and meta-XB for modulation classification [O'Shea et al., 2018]. We consider $N_\tau = 9$ examples for each data set $\mathcal{D}_\tau$. The boxes represent the 25% (lower edge), 50% (line within the box), and 75% (upper edge) percentiles of the per-task performance metrics evaluated over 100 different meta-test tasks.

Fig. 7 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.

5.3 Image Classification

Lastly, we consider image classification problem with the *mini*Imagenet dataset [Vinyals et al., 2016] 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 [Finn et al., 2017]. 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. 8 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.

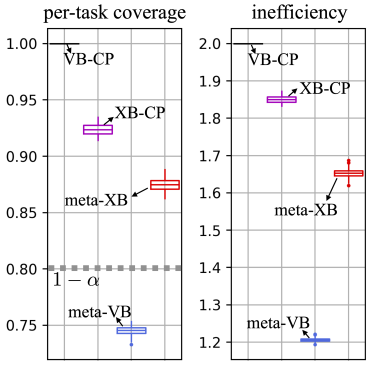

Figure 8: Per-task coverage and inefficiency of VB-CP, XB-CP, meta-VB, and meta-XB for *mini*Imagenet [Vinyals et al., 2016]. We consider $N_\tau = 4$ examples for each data set $\mathcal{D}_\tau$. The boxes represent the 25% (lower edge), 50% (line within the box), and 75% (upper edge) percentiles of the per-task performance metrics evaluated over 100 different meta-test tasks.

6 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 [O'Shea et al., 2016, 2018], 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 [Stutz et al., 2021, Einbinder et al., 2022], or with stochastic set predictors.

References

Ron Amit and Ron Meir. Meta-learning by adjusting priors based on extended PAC-Bayes theory. In *Proc. of Int. Conf. Machine Learning (ICML)*, pages 205–214, Jul 2018.

Elaine Angelino, Matthew James Johnson, Ryan P Adams, et al. Patterns of scalable bayesian inference. *Foundations and Trends® in Machine Learning*, 9(2-3):119–247, 2016.

Vineeth Balasubramanian, Shen-Shyang Ho, and Vladimir Vovk. *Conformal prediction for reliable machine learning: theory, adaptations and applications*. Newnes, 2014.

Rina Foygel Barber, Emmanuel J Candes, Aaditya Ramdas, and Ryan J Tibshirani. Predictive inference with the jackknife+. *The Annals of Statistics*, 49(1):486–507, 2021.

Jonathan Baxter. A model of inductive bias learning. *Journal of Artificial Intelligence Research*, 12: 149–198, March 2000.

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

Bat-Sheva Einbinder, Yaniv Romano, Matteo Sesia, and Yanfei Zhou. Training uncertainty-aware classifiers with conformalized deep learning. *arXiv preprint arXiv:2205.05878*, 2022.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proc. of Int. Conf. Machine Learning-Volume 70*, pages 1126–1135, Aug. 2017.

Chelsea Finn, Kelvin Xu, and Sergey Levine. Probabilistic model-agnostic meta-learning. *arXiv preprint arXiv:1806.02817*, 2018.

Adam Fisch, Tal Schuster, Tommi Jaakkola, and Regina Barzilay. Few-shot conformal prediction with auxiliary tasks. In *International Conference on Machine Learning*, pages 3329–3339. PMLR, 2021.

Rina Foygel Barber, Emmanuel J Candes, Aaditya Ramdas, and Ryan J Tibshirani. The limits of distribution-free conditional predictive inference. *Information and Inference: A Journal of the IMA*, 10(2):455–482, 2021.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

Benjamin Guedj. A primer on PAC-Bayesian learning. *arXiv preprint arXiv:1901.05353*, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

Rafael Izbicki, Gilson Shimizu, and Rafael B Stern. Cd-split and hpd-split: efficient conformal regions in high dimensions. *arXiv preprint arXiv:2007.12778*, 2020.

Sharu Theresa Jose and Osvaldo Simeone. Information-theoretic bounds on transfer generalization gap based on Jensen-Shannon divergence. *arXiv preprint: arXiv: 2010.09484*, 2020.

Sharu Theresa Jose, Sangwoo Park, and Osvaldo Simeone. Information-theoretic analysis of epistemic uncertainty in bayesian meta-learning. In *International Conference on Artificial Intelligence and Statistics*, pages 9758–9775. PMLR, 2022.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Jeremias Knoblauch, Jack Jewson, and Theodoros Damoulas. Generalized Variational Inference. *arXiv preprint arXiv:1904.02063*, 2019.

Roger Koenker and Gilbert Bassett Jr. Regression quantiles. *Econometrica: journal of the Econometric Society*, pages 33–50, 1978.

Brenden Lake, Ruslan Salakhutdinov, Jason Gross, and Joshua Tenenbaum. One shot learning of simple visual concepts. In *Proceedings of the annual meeting of the cognitive science society*, volume 33, 2011.

Peter Larsson. Golden angle modulation. *IEEE Wireless Communications Letters*, 7(1):98–101, 2017.

Jing Lei and Larry Wasserman. Distribution-free prediction bands for non-parametric regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(1):71–96, 2014.

Benjamin LeRoy and David Zhao. Md-split+: Practical local conformal inference in high dimensions. *arXiv preprint arXiv:2107.03280*, 2021.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. Locally valid and discriminative prediction intervals for deep learning models. *Advances in Neural Information Processing Systems*, 34:8378–8391, 2021.

David JC MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

Andres Masegosa. Learning under model misspecification: Applications to variational and ensemble methods. *Advances in Neural Information Processing Systems*, 33:5479–5491, 2020.

Warren R Morningstar, Alex Alemi, and Joshua V Dillon. Pacm-bayes: Narrowing the empirical risk gap in the misspecified bayesian regime. In *International Conference on Artificial Intelligence and Statistics*, pages 8270–8298. PMLR, 2022.

Cuong Nguyen, Thanh-Toan Do, and Gustavo Carneiro. Uncertainty in model-agnostic meta-learning using variational inference. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3090–3100, 2020.

Timothy J O'Shea, Johnathan Corgan, and T Charles Clancy. Convolutional radio modulation recognition networks. In *International conference on engineering applications of neural networks*, pages 213–226. Springer, 2016.

Timothy James O'Shea, Tamoghna Roy, and T Charles Clancy. Over-the-air deep learning based radio signal classification. *IEEE Journal of Selected Topics in Signal Processing*, 12(1):168–179, 2018.

Sangdon Park, Edgar Dobriban, Insup Lee, and Osbert Bastani. Pac prediction sets for meta-learning. *arXiv preprint arXiv:2207.02440*, 2022.

Sangwoo Park, Hyeryung Jang, Osvaldo Simeone, and Joonhyuk Kang. Learning to demodulate from few pilots via offline and online meta-learning. *IEEE Transactions on Signal Processing*, 69: 226–239, 2020.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Sachin Ravi and Alex Beatson. Amortized bayesian meta-learning. In *International Conference on Learning Representations*, 2018.

Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. 2016.

Yaniv Romano, Evan Patterson, and Emmanuel Candes. Conformalized quantile regression. *Advances in neural information processing systems*, 32, 2019.

Yaniv Romano, Matteo Sesia, and Emmanuel Candes. Classification with valid and adaptive coverage. *Advances in Neural Information Processing Systems*, 33:3581–3591, 2020.

Jonas Rothfuss, Dominique Heyn, Andreas Krause, et al. Meta-learning reliable priors in the function space. *Advances in Neural Information Processing Systems*, 34:280–293, 2021.

Osvaldo Simeone. *Machine Learning for Engineers*. Cambridge University Press, 2022.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.

David Stutz, Ali Taylan Cemgil, Arnaud Doucet, et al. Learning optimal conformal classifiers. *arXiv preprint arXiv:2110.09192*, 2021.

Ryan J Tibshirani, Rina Foygel Barber, Emmanuel Candes, and Aaditya Ramdas. Conformal prediction under covariate shift. *Advances in neural information processing systems*, 32, 2019.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. *arXiv preprint arXiv:1606.04080*, 2016.

Vladimir Vovk. Conditional validity of inductive conformal predictors. In *Asian conference on machine learning*, pages 475–490. PMLR, 2012.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*. Springer Science & Business Media, 2005.

Florian Wenzel, Kevin Roth, Bastiaan S Veeling, Jakub Świątkowski, Linh Tran, Stephan Mandt, Jasper Snoek, Tim Salimans, Rodolphe Jenatton, and Sebastian Nowozin. How good is the bayes posterior in deep neural networks really? *arXiv preprint arXiv:2002.02405*, 2020.

Mingzhang Yin, George Tucker, Mingyuan Zhou, Sergey Levine, and Chelsea Finn. Meta-Learning Without Memorization. *arXiv preprint arXiv:1912.03820*, 2019.

Jaesik Yoon, Taesup Kim, Ousmane Dia, Sungwoong Kim, Yoshua Bengio, and Sungjin Ahn. Bayesian model-agnostic meta-learning. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 7343–7353, 2018.

Appendix

A Proofs

A.1 Proof of Theorem 2

The proof mainly follows [Barber et al., 2021, Section B.1] and [Barber et al., 2021, Section B.2.2] with the following changes.

We first extend the result for regression problems in [Barber et al., 2021] to classification, starting with the case $K = N_\tau$, for which the mapping function $k(\cdot)$ is the identity $k(i) = i$. Unlike [Barber et al., 2021], which defined "comparison matrix" of residuals for the regression problem, we consider a more general comparison matrix defined in terms of NC scores that can be applied for both classification and regression problems in a manner similar to [Romano et al., 2020]. Accordingly, we define the comparison matrix $A \in \{0,1\}^{(N_\tau+1)\times(N_\tau+1)}$

$$A(i,j) = \left\{ \begin{array}{ll} \mathbf{1}\Big(\min_{k\in\{1,\ldots,K+1\}} \mathrm{NC}(z_\tau[i]|\mathcal{D}_{\tau,\neg(k(i),k)},\xi) > \mathrm{NC}(z_\tau[j]|\mathcal{D}_{\tau,\neg(k(i),k(j))},\xi)\Big) & \\ & \text{for } k(i) \neq k(j) \\ 0 & \\ & \text{for } k(i) = k(j) \end{array} \right\}, \tag{19}$$

for a fixed vector of hyperparameter ξ. The cardinality of the set $S(A)$ of "strange" points

$$S(A) = \left\{ i \in \{1,2,\ldots,N_\tau+1\} : \sum_{j'=1}^{N_\tau+1} A(i,j') \geq (1-\alpha')(N_\tau+1) \right\} \tag{20}$$

can be bounded as $|S(A)| \leq N_\tau + 1 - (1-\alpha')(N_\tau+1)$ [Barber et al., 2021, Romano et al., 2020]. Therefore, theorem 2 holds for $K = N_\tau$, since any $N_\tau + 1$ points can be "strange points" with equal probability thanks to Assumption 1.

To address the case $K < N_\tau$, we follow [Barber et al., 2021, Section B.2.2] by drawing $N_\tau/K - 1$ additional test examples that are all assigned to the $(K+1)$th fold. This way, the actual $(N_\tau+1)$th test point is equally likely to be in any of the $K+1$ folds. Now, taking the augmented data set $\bar{\mathcal{D}}$ that contains all the $N_\tau + N_\tau/K$ examples in lieu of $\mathcal{D}$ in (19), we can bound the number of "strange points" in set (20) as

$$|S(A)| \leq N_\tau + N_\tau/K - (1-\alpha')(N_\tau+1). \tag{21}$$

Finally, by using the same proof technique in [Barber et al., 2021, Section B.2.2], we have the inequality

$$\mathbb{P}\big(\mathbf{y}_\tau \in \Gamma_\alpha^{K\text{-XB}}(\mathbf{x}_\tau|\boldsymbol{\mathcal{D}}_\tau,\xi)\big) \geq 1 - \alpha' - \frac{1-K/N_\tau}{K+1}. \tag{22}$$

In Theorem 2, we choose $\alpha' = \frac{1-K/N_\tau}{K+1}$, which satisfies per-task validity condition (1) from (22).

A.2 Proof for Definition 5

From the definition of the XB-CP set predictor (7), the inefficiency can be obtained as

$$\begin{aligned} &|\Gamma_\alpha^{K\text{-XB}}(x_\tau|\mathcal{D}_\tau,\xi)| \\ &= \sum_{y\in\mathcal{Y}_\tau} \mathbf{1}\Bigg(\sum_{i=1}^{N_\tau} \mathbf{1}\Big(\min_{k\in\{1,\ldots,K\}} \mathrm{NC}((x_\tau,y)|\mathcal{D}_{\tau,\neg k},\xi) \leq \mathrm{NC}(z_\tau[i]|\mathcal{D}_{\tau,\neg k(i)},\xi)\Big) \;\geq\; \lfloor \alpha'(N_\tau+1)\rfloor \Bigg) \\ &= \sum_{y\in\mathcal{Y}_\tau} \mathbf{1}\Big(Q^-_{1-\alpha'}\Big(\Big\{ \min_{k\in\{1,\ldots,K\}} \mathrm{NC}((x_\tau,y)|\mathcal{D}_{\tau,\neg k},\xi) - \mathrm{NC}(z_\tau[i]|\mathcal{D}_{\tau,\neg k(i)},\xi) \Big\}_{i=1}^{N_\tau} \Big) \leq 0 \Big) \qquad (23) \\ &= \sum_{y\in\mathcal{Y}_\tau} \mathbf{1}\Big(Q_{1-\alpha'}\Big(\Big\{ \mathrm{NC}(z_\tau[i]|\mathcal{D}_{\tau,\neg k(i)},\xi) - \min_{k\in\{1,\ldots,K\}} \mathrm{NC}((x_\tau,y)|\mathcal{D}_{\tau,\neg k},\xi) \Big\}_{i=1}^{N_\tau} \Big) \geq 0 \Big), \qquad (24) \end{aligned}$$

with $Q^-_{1-\alpha}(\{a[i]\}_{i=1}^M) := -Q_{1-\alpha}(\{-a[i]\}_{i=1}^M)$ being the $\lfloor \alpha(M+1) \rfloor$th smallest value in the set $\{a[1], \ldots, a[M], \infty\}$. The equality in (23) is proved as follows. Defining $g(z_\tau) := \min_{k \in \{1,\ldots,K\}} \text{NC}((x_\tau, y)|\mathcal{D}_{\tau,\neg k}, \xi)$ with $z_\tau = (x_\tau, y)$ and $f(z_\tau[i]) := \text{NC}(z_\tau[i]|\mathcal{D}_{\tau,\neg k(i)}, \xi)$, we show that the inequality $\sum_{i=1}^{N_\tau} \mathbf{1}\big(g(z_\tau) \leq f(z_\tau[i])\big) \geq \lfloor \alpha'(N_\tau + 1) \rfloor$ is equivalent to $Q^-_{1-\alpha'}\big(\{g(z_\tau) - f(z_\tau[i])\}_{i=1}^{N_\tau}\big) \leq 0$. This is a consequence of the following equivalence relations:

$$
\begin{aligned}
&\sum_{i=1}^{N_\tau} \mathbf{1}\big(g(z_\tau) \leq f(z_\tau[i])\big) \geq \lfloor \alpha'(N_\tau + 1) \rfloor \\
\Leftrightarrow &\sum_{i=1}^{N_\tau} \mathbf{1}\big(g(z_\tau) - f(z_\tau[i]) \leq 0\big) \geq \lfloor \alpha'(N_\tau + 1) \rfloor \\
\Leftrightarrow &\text{ at least } \lfloor \alpha'(N_\tau + 1) \rfloor \text{ values of } g(z_\tau) - f(z_\tau[i]) \text{ are smaller than or equal to } 0 \\
\Leftrightarrow &\lfloor \alpha'(N_\tau + 1) \rfloor \text{th smallest value of } g(z_\tau) - f(z_\tau[i]) \text{ is smaller than or equal to } 0 \\
\Leftrightarrow &Q^-_{\alpha'}\big(\{g(z_\tau) - f(z_\tau[i])\}_{i=1}^{N_\tau}\big) \leq 0 \\
\Leftrightarrow &Q_{\alpha'}\big(\{f(z_\tau[i]) - g(z_\tau)\}_{i=1}^{N_\tau}\big) \geq 0.
\end{aligned} \tag{25}
$$

By replacing $\mathbf{1}(\cdot)$ with the sigmoid $\sigma(\cdot)$, $\min(\cdot)$ with $\text{softmin}(\cdot)$ (13), and the quantile $Q_{\alpha'}(\cdot)$ with $\hat{Q}_{\alpha'}(\cdot)$ (14), we finally obtain the soft inefficiency of K-fold XB-CP predictor in (16) from (23).

B Details on Soft Adaptive NC Scores

Recalling (11), while denoting $p_{y'} := p(y'|x, \tilde{\mathcal{D}}_\tau, \xi)$ and $p_y := p(y|x, \tilde{\mathcal{D}}_\tau, \xi)$, the adaptive NC score for input-output pair $z = (x, y)$ with $x \in \mathcal{X}_\tau$ and $y \in \mathcal{Y}_\tau$ can be computed as

$$
\begin{aligned}
\text{NC}^{\text{ada}}(z|\tilde{\mathcal{D}}_\tau, \xi) &= \sum_{y' \in \mathcal{Y}_\tau} \mathbf{1}\big(p_{y'} \geq p_y\big) p_{y'} \\
&= \sum_{y' \in \mathcal{Y}_\tau} \text{ReLU}(p_{y'} - p_y) + p_y \sum_{y' \in \mathcal{Y}_\tau} \mathbf{1}(p_{y'} \geq p_y) \\
&= 1 + \sum_{y' \in \mathcal{Y}_\tau} \text{ReLU}(p_y - p_{y'}) - p_y \sum_{y' \in \mathcal{Y}_\tau} \mathbf{1}(p_{y'} < p_y).
\end{aligned} \tag{26}
$$

We define the soft adaptive NC score by approximating the indicator function $\mathbf{1}(\cdot)$ with the sigmoid $\sigma(\cdot)$ as

$$
\text{NC}^{\text{ada}}(z|\tilde{\mathcal{D}}_\tau, \xi) = 1 + \sum_{y' \in \mathcal{Y}_\tau} \text{ReLU}(p_y - p_{y'}) - p_y \sum_{y' \in \mathcal{Y}_\tau} \sigma(p_{y'} < p_y). \tag{27}
$$

Note that, we have found that the preprocessing (26) yields better empirical per-input coverage as compared to the direct approximation of (11) that replaces the indicator function $\mathbf{1}(\cdot)$ with sigmoid $\sigma(\cdot)$, i.e., $\sum_{y' \in \mathcal{Y}_\tau} \sigma\big(p_{y'} \geq p_y\big) p_{y'}$.

C Additional Experiments

C.1 Demodulation

To elaborate further on the last column in Fig. 3, here we present a toy example that allows us to visualize the set predictors obtained by XB-CP and meta-XB, both with conventional and adaptive NC. To this end, we implement XB-CP with the same neural network used for meta-XB but with the hyperparameter vector ξ defining the initialization of GD set to a random vector [He et al., 2015]. Given a learning task τ, the input and output space $\mathcal{X}_\tau$ and $\mathcal{Y}_\tau$ are given by the set of complex points, i.e., by the two-dimensional real vectors [Larsson, 2017]

$$
\mathcal{X}_\tau = \mathcal{Y}_\tau = \left\{ \sqrt{\frac{2z}{M+1}} e^{j2\pi\left(1 - \frac{\sqrt{5}-1}{2}\right)z} e^{j\phi_\tau} \text{ for } z = 1, 2, \ldots, M \right\}, \tag{28}
$$

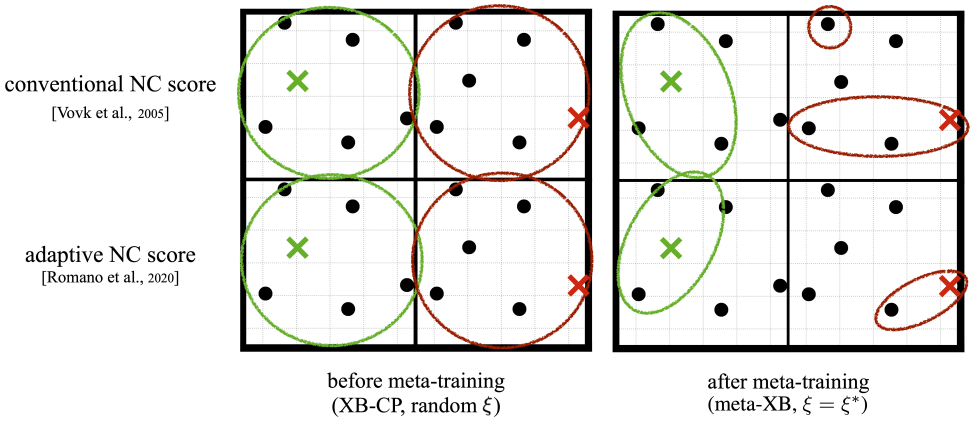

Figure 9: Illustration of set predictions for the demodulation problem described in Appendix C.1. Colored crosses represent ground-truth outputs and the correspondingly colored circles depict the predicted sets. For visualization purpose, the figure is generated based on a single realization of task τ, data set $\mathcal{D}_\tau$, and test input x_τ. $T = 1000$ different tasks are used for meta-training.

for some task-specific phase shift $\phi_\tau \in [0, 2\pi]$. Denoting as $\mathcal{N}_\tau(x) = \{y \in \mathcal{Y}_\tau : |x-y| \leq r \text{ and } y \neq x\}$ the set of neighboring points within some radius r, the ground-truth distribution $p(y_\tau|x_\tau)$ is such that y_τ equals x_τ with probability $1-p$, and it equals any neighboring point $y \in \mathcal{N}_\tau(x)$ with probability $p/|\mathcal{N}_\tau(x)|$. 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 [Clevert et al., 2015] in the hidden layers and a softmax activation in the last layer.

Fig. 9 visualizes the set predictors for XB-CP, i.e., with a random hyperparameter vector ξ, and for meta-XB, after meta-training with 1000 tasks, by focusing on a specific realizations of phase shift that follows the distribution $\phi_\tau \sim \text{Unif}[0, 2\pi)$. 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. 9, have more neighbors as compared to points at the edge, as the red points in Fig. 9, making it harder to identify the true value of y given x.